

**ESCUELA TÉCNICA SUPERIOR DE
INGENIERÍA DE TELECOMUNICACIÓN**
UNIVERSIDAD DE MÁLAGA



PROYECTO FIN DE CARRERA

*MONITORIZACIÓN DE VARIABLES AMBIENTALES
UTILIZADAS EN EL DISEÑO DE ESTRUCTURAS
BIOCLIMÁTICAS*

INGENIERÍA DE TELECOMUNICACIÓN

MÁLAGA, 2011

FRANCISCO ORTEGA ZAMORANO

**ESCUELA TÉCNICA SUPERIOR DE
INGENIERÍA DE TELECOMUNICACIÓN**

UNIVERSIDAD DE MÁLAGA

Titulación: Ingeniería de Telecomunicación

Reunido el tribunal examinador en el día de la fecha, constituido por:

D./D^a. _____

D./D^a. _____

D./D^a. _____

para juzgar el Proyecto Fin de Carrera titulado:

**MONITORIZACIÓN DE VARIABLES AMBIENTALES UTILIZADAS
EN EL DISEÑO DE ESTRUCTURAS BIOCLIMÁTICAS**

del alumno D./D^a. Francisco Ortega Zamorano

dirigido por D./D^a. Ignacio Molina Conde

ACORDÓ POR _____ OTORGAR LA
CALIFICACIÓN DE _____

Y, para que conste, se extiende firmada por los componentes del tribunal, la presente diligencia

Málaga, a _____ de _____ de _____

El Presidente

El Vocal

El Secretario

Fdo.: _____ Fdo.: _____ Fdo.: _____

**ESCUELA TÉCNICA SUPERIOR DE
INGENIERÍA DE TELECOMUNICACIÓN**

UNIVERSIDAD DE MÁLAGA

TÍTULO DEL PFC

*MONITORIZACIÓN DE VARIABLES AMBIENTALES UTILIZADAS EN EL
DISEÑO DE ESTRUCTURAS BIOCLIMÁTICAS*

REALIZADO POR:

Francisco Ortega Zamorano

DIRIGIDO POR:

Ignacio Molina Conde

DEPARTAMENTO DE: Tecnología Electrónica.

TITULACIÓN: Ingeniería de Telecomunicación.

PALABRAS CLAVE: Arquitectura bioclimática, Confort, Arduino, Anemómetro y Veleta, Piranómetro, Comandos AT, Matlab.

RESUMEN:

La generación de electricidad para el consumo en el ámbito doméstico está provocando un incremento de la cantidad de gases efecto invernadero que se expulsan a la atmósfera; esto produce efectos nocivos en el planeta; para paliar tales efectos se pueden construir todas las edificaciones bajo la premisa de arquitecturas bioclimáticas, que reducen el consumo hasta en un 50%.

En el presente proyecto se ha creado una herramienta hardware para poder captar las diferentes variables ambientales relacionadas con el análisis de las arquitecturas bioclimáticas, así como una aplicación software encargada de la visualización de los diferentes datos para el posterior estudio de la zona a edificar.

Málaga, Mayo de 2011

Dedico este proyecto a mi familia por confiar en mí y apoyarme incluso cuando no lo merecía.

A mi Padre por aguantar.

A mi Madre porque siempre está ahí.

A mi Hermana por levantarme cada vez que caigo.

A mis niñas por su alegría.

Y a Sandra por ser ella.

Agradecimientos,

A mi tutor, Ignacio Molina, por la paciencia que ha tenido conmigo.

A todos los compañeros (y amigos) que me ayudaron en el camino, solo ellos saben lo difícil que ha sido: Antonio (más tiempo), Arturo, Chemi (si, lo nuestro es de estudio), Pips, mi Rake, Maca, Esteban, Casi, Yaiza, Porcu, Pablillo, Tabo, Jorge,... Todos ellos, y los que por desgracia se me olvidan, han hecho que esto merezca la pena.

A los otros amigos, que no todo es teleco. A mi Concu (Sin él las salidas no son lo mismo); a Dami (taco!!); a Clara (Soportó los malos momentos); Juanjo(de cuñao a representante), Alberto, Laura, Agu, Nacho, Jose, Xino...

Al Conil team, porque vaya fines de semana; Kemariaaa!!!! A la MAE!. A los buenos momentos con los amigos, porque eso es lo más importante; a mi clan, Siempre Vendetta!! A los partidos de futbol. A mi sensei y al karate, Oss.

A Sandra por ayudarme a hacerlo, sin ella no hubiera podido con todo.

A ti, que lees este proyecto, porque si lo estás haciendo, seguro que has sido, eres o serás alguien importante en mi vida.

A toda y cada una de las personas que han pasado por mi vida, porque supongo que sin ellas, yo no sería el que soy. GRACIAS por todo.

Índice

Introducción	1
1.1 Motivación	2
1.2 Objetivos	4
1.3 Organización	5
Introducción al diseño de edificaciones.....	7
2.1 El confort en el acondicionamiento bioclimático	8
2.1.1 Confort Térmico	10
2.1.1.1 Técnicas para evaluar un ambiente Térmico	12
2.1.2 Confort Visual.....	15
2.1.3 Confort Acústico.....	16
2.2 El clima como característica extrínseca de la vivienda.....	17
2.2.1 Factores del clima	18
2.2.2 Elementos climáticos o parámetros ambientales	20
2.3 Arquitectura bioclimática	21
2.3.1 Principales actuaciones para una Arquitectura bioclimática.....	22
2.3.2.1 Climatización pasiva	25
2.3.2.2 Producción de energías alternativas en edificios.....	26
2.3.3 Comentarios sobre la arquitectura bioclimática	27
2.3.4 Ejemplos de Arquitecturas bioclimática	28
Herramienta hardware.....	29
3.1 Arduino	30
3.1.1 Esquema y pines	32
3.1.1.1 Alimentación.....	32
3.1.1.2 Pines Digitales.....	33
3.1.1.3 Pins de Entrada Analógica	35
3.1.1.4 PWM	36
3.1.2 Memoria	37
3.1.3 Programación.....	38
3.1.4 Comunicaciones	40
3.1.5 Reinicio Automático (Software)	41
3.1.6 Protección contra sobretensiones en USB	41
3.1.7 Características Físicas	41
3.1.8 ATmega328	42
3.1.8.1 Características	42
3.1.8.2 Configuración de pines.....	43
3.1.8.3 Diagrama de bloques	45
3.2 Memoria Micro SD	47
3.2.1 Consideraciones previas.....	48
3.2.2 Lectura Tarjeta MicroSD	49
3.2.3 Escritura tarjeta MicroSD	50
3.2.4 Resultado escritura Tarjeta MicroSD	51
3.3 GPRS	52
3.3.1 Servidor FTP.....	55
3.3.2 Problemas	56
3.3.3 Comandos AT	61
3.3.3.1 Comandos específicos de GPRS	61
3.3.3.2 Comandos específicos de FTP	62
3.3.4 Programación GPRS	63
3.4 Tiempo real (RTC).....	65
3.4.1 DS1307	65

3.4.2 BUS I²C.....	66
3.4.2.1 Descripción de las señales	66
3.4.2.2 Definición de términos.....	67
3.4.2.3 Protocolo de comunicación del bus I ² C	68
3.4.2.4 La comunicación en más detalle	69
3.4.2.5 Direcciónamiento de dispositivos en el bus I ² C.....	70
3.4.3 Realización física:	70
3.4.4 Programación para el RTC.....	71
3.5 Anemómetro y Veleta	72
 3.5.1 Anemómetro.....	72
3.5.1.1 Especificaciones.....	73
3.5.1.2 Esquema de funcionamiento del anemómetro	74
3.5.1.3 Programación anemómetro	75
 3.5.2 Veleta	76
3.5.2.1 Especificaciones.....	76
3.5.2.2 Esquema eléctrico	77
 3.5.3 Montaje.....	77
3.6 Piranómetro.....	78
 3.6.1 TAOS TSL230R	78
3.6.1.1 Configuración de pines TAOS TSL230.....	79
3.6.1.2 Características de funcionamiento (a V _{DD} = 5 V, TA = 25° C)	80
3.6.1.3 Características eléctricas (a 25°C y V _{DD} = 5V)	81
3.6.1.4 Las curvas características TSL230.....	81
 3.6.2 Espectro Solar	82
 3.6.3 Radiación solar partiendo de la frecuencia del fotodiodo.....	84
 3.6.4 Programación TAOS TLS 230	86
 3.6.5 Realización física piranómetro	87
 3.6.6 Mapa de sombras.....	89
3.7 Sensor de temperatura y Sensor de Humedad.....	91
 3.7.1 Sensor de Temperatura. MAXIM DS18B20	92
3.7.1.1 Características	92
3.7.1.2 Configuración de pines	93
3.7.1.3 Resumen	93
3.7.1.4 Especificaciones Eléctricas. (-55°C a 125°C, V _{DD} =3.0V a 5.5V)	94
3.7.1.5 Comunicaciones 1-Wire.....	95
3.7.1.6 Programación DS18B20	102
 3.7.2 Sensor de humedad.....	105
3.7.2.1 Descripción	105
3.7.2.2 Configuración de pines	105
3.7.2.3 Especificaciones.....	106
3.7.2.4 Programación sensor humedad	107
 3.7.3 Realización física del sensor de temperatura y humedad	107
3.8 Alimentación.....	109
 3.8.1 Consumo diario.....	109
 3.8.2 Dimensionado sistema de alimentación	110
 3.8.3 Realización física sistema de alimentación	112
3.9 Herramienta hardware completo.....	114
Verificación del sistema hardware	115
4.1 Instrumentación externa	116
4.1.1 Equipo multifunción 4 en 1 PCE-EM882	116
4.1.2 Anemómetro PCE-AM81	117
4.2 Verificación de los sensores.....	118
4.2.1 Verificación sensor de temperatura	118
4.2.2 Verificación del sensor de Humedad.....	119
4.2.3 Verificación Anemómetro	119
4.2.4 Verificación Veleta.....	120
4.2.5 Verificación Piranómetro.....	121

4.3 Verificación GPRS	124
4.4 Verificación módulo MicroSD	124
4.5 Verificación sistema de alimentación.....	125
4.6 Verificación sistema hardware completo	126
Herramienta software.....	127
5.1 Matlab	128
5.2 Programación software	129
5.2.1 Código Matlab	131
5.2.2 Creación de ejecutables.....	135
5.3 Interfaz de la aplicación	135
5.4 Verificación de la aplicación software	140
Ejemplo de aplicación	143
6.1 Datos de la zona.....	144
6.2 Días medidos.....	146
6.2.1 Día de agosto (6-8-10).....	146
6.2.2 Día de octubre (4-10-10).....	148
6.2.3 Día de diciembre (3-12-10).....	150
6.3 Conclusiones	152
Conclusiones y líneas futuras	155
7.1 Conclusiones	156
7.2 Líneas futuras.....	157

Índice de figuras

Figura 1.1.1 Emisiones de GEI por sectores en 2008 en España.....	3
Figura 1.2.1 Esquema del sistema.....	4
Figura 2.1.1 Principales factores que afectan al confort térmico	10
Figura 2.1.2 Diagrama psicométrico	13
Figura 2.1.3 Diagrama Psicométrico de Givoni	14
Figura 2.1.4 Carta de Olgayay	15
Figura 2.1.5 Grafica de Kruithof	16
Figura 2.1.6 Grafica ruido.....	17
Figura 2.2.1 Interrelación del clima con el confort personal	18
Figura 2.3.1 Ganancia directa	23
Figura 2.3.2 Muro de acumulación	23
Figura 2.3.3 Pared de captación solar.....	23
Figura 2.3.4 Invernadero.....	23
Figura 2.3.5 Muro de acumulación ventilado	24
Figura 2.3.6 Chimenea solar.....	25
Figura 2.3.7 Diferentes arquitecturas bioclimáticas.	28
Figura 3.1.1 Esquema herramienta hardware	31
Figura 3.1.2 Arduino Duemilanove	31
Figura 3.1.3 Esquema de componentes del Arduino Duemilanove.....	32
Figura 3.1.4 Señal de onda cuadrada de amplitud acotada, mostrando el ciclo de trabajo D.	36
Figura 3.1.5 Relaciones entre el ancho del pulso y el valor analógico con la técnica PWM	37
Figura 3.1.6 Chip ATMega 328	42
Figura 3.1.7 Configuración de pines.....	43
Figura 3.1.8 Diagrama de bloques ATmega328.	45
Figura 3.2.1 Módulo de memoria MicroSD.....	47
Figura 3.2.2 Componentes del módulo MicroSD.....	47
Figura 3.3.1 Módulo GPRS.	52
Figura 3.3.2 Esquema de los elementos módulos GPRS.....	53
Figura 3.3.3 Módulo GPRS acoplado al Arduino	54
Figura 3.3.4 Email recibido de Libelium 1	56
Figura 3.3.5 Email enviado a Libelium 1.....	57
Figura 3.3.6 Lista de emails intercambiados con Libelium 1.....	57
Figura 3.3.7 Email enviado a Libelium 2.....	58
Figura 3.3.8 Email recibido de Libelium 2.	58
Figura 3.3.9 Email enviado a Libelium 3.....	59
Figura 3.3.10 Email recibido de Libelium 3	59
Figura 3.3.11 Lista de emails intercambiados con Libelium 2.....	60
Figura 3.3.12 Email recibido de Libelium 4.	60
Figura 3.4.1 Pines del Chip DS1307	65
Figura 3.4.2 Estructura de un circuito típico para montar un DS1307.....	66
Figura 3.4.3 Diagrama de la conexión de un bus I ² C.....	67
Figura 3.4.4 Condición de inicio Bus I ² C.	68
Figura 3.4.5 Secuencia de inicio de bus I ² C.	69
Figura 3.4.6 Secuencia deparada del bus I ² C.	69
Figura 3.4.7 Realización física de un reloj de tiempo real (RTC).	71
Figura 3.5.1 Copas del anemómetro de inspeed.	72
Figura 3.5.2 Esquema de funcionamiento del anemómetro cuando su salida es un 1 lógico.	74
Figura 3.5.3 Esquema de funcionamiento del anemómetro cuando su salida es un 0 lógico.	74
Figura 3.5.4 Veleta de Inspeed.....	76
Figura 3.5.5 Esquema eléctrico de la veleta de Inspeed	77
Figura 3.5.6 Fotografía de la veleta.....	77
Figura 3.5.7 Fotografía anemómetro	77
Figura 3.6.1 Fotodiodo TSL230.....	78
Figura 3.6.2 Esquema de pines TAOS TSL230.	79
Figura 3.6.3 Respuesta espectral del fotodiodo.....	81
Figura 3.6.4 Coeficiente de temperatura de la frecuencia de salida.	81

Figura 3.6.5 Frecuencia de salida en función de la intensidad irradiada.....	82
Figura 3.6.6 Respuesta espectral solar a nivel del mar y en la atmósfera.....	82
Figura 3.6.7 Función de transferencia de la radiación solar.....	84
Figura 3.6.8 Función de transferencia del fotodiodo.	84
Figura 3.6.9 Función de transferencia del coeficiente de temperatura.	85
Figura 3.6.10 Función de transferencia del coeficiente de temperatura.	85
Figura 3.6.11 Vector tridimensional.	87
Figura 3.6.12 Cubo de fotodiodos, vista desde arriba.....	88
Figura 3.6.13 Cubo de fotodiodos, vista desde abajo.	88
Figura 3.6.14 Piranómetro con la cubierta de metacrilato.	88
Figura 3.6.15 Piranómetro.....	89
Figura 3.6.16 Mapa de zona.	90
Figura 3.6.17 Fotodiodo encapsulado.	90
Figura 3.6.18 Fotodiodos necesarios para realizar mapa de sombras.	90
Figura 3.7.1 Sensor DS18B20.....	92
Figura 3.7.2 Diagrama de pines DS18B20.....	93
Figura 3.7.3 Diagrama de bloques del sensor DS18B20.	93
Figura 3.7.4 Diagrama de flujo para la operación con los comandos ROM.....	97
Figura 3.7.5 Flujo para la operación con los comandos de la función DS18B20.	99
Figura 3.7.6 Sensor de humedad de Honeywell.	105
Figura 3.7.7 Configuraciones pines del sensor de humedad de honeywell.	105
Figura 3.7.8 Entorno operativo sensor HIH-4021-004.....	106
Figura 3.7.9 Circuito típico del sensor HIH-4021-004.	106
Figura 3.7.10 Sensores DS18B20 y HIH-4021.	107
Figura 3.7.11 Representación de la estructura de cobertura de los sensores.	107
Figura 3.7.12 Vista, desde debajo, de la estructura protectora de sensores.	108
Figura 3.7.13 Abertura lateral de la estructura protectora de sensores.	108
Figura 3.7.14 Sensor de Temperatura y Humedad.....	108
Figura 3.8.1 Instalación fotovoltaica aislada.	109
Figura 3.8.2 Consumo del sistema a lo largo de una hora.....	110
Figura 3.8.3 Batería 7,2Ah/20H.	110
Figura 3.8.4 Estructura de soporte.	111
Figura 3.8.5 Panel solar	111
Figura 3.8.6 Regulador phocos.....	112
Figura 3.8.7 Caja Protectora.	112
Figura 3.8.8 Visión frontal del ventilador.	112
Figura 3.8.9 Sistema alimentación.	113
Figura 3.9.1 Herramienta hardware completo.....	114
Figura 4.1.1 Equipo PCE-EM882.	116
Figura 4.1.2 Equipo PCE-AM81.....	117
Figura 4.2.1 Medidas de distintos escenarios del sensor de temperatura y del PCE-EM882.	118
Figura 4.2.2 Medidas en diferentes días del sensor de Humedad respecto al PCE-EM882.	119
Figura 4.2.3 Medidas en diferentes condiciones del anemómetro respecto al PCE-AM81.	120
Figura 4.2.4 Radiación diaria, teórica y Medida, de un día de agosto.....	122
Figura 4.2.5 Radiación perpendicular diaria, teórica y Medida, de un día de agosto.	123
Figura 5.1.1 Logotipo del Matlab.....	128
Figura 5.2.1 Pantalla de inicio de la aplicación GUI del Matlab.....	129
Figura 5.2.2 Pantalla de la aplicación GUI.	130
Figura 5.2.3 Puntos de representación de la zona	134
Figura 5.3.1 Pantalla inicial del programa en Matlab.	135
Figura 5.3.2 Representación de los valores de temperatura y humedad.....	136
Figura 5.3.3 Representación de los valores de temperatura y humedad.....	137
Figura 5.3.4 Representación de la radiación solar perpendicular.	138
Figura 5.3.5 Representación del mapa de sombras.	139
Figura 5.3.6 Evolución temporal del mapa de sombras.	139
Figura 5.4.1 Situación 1 en archivo incorrecto.	140
Figura 5.4.2 Situación 2 en archivo incorrecto.	140
Figura 6.1.1 Coordenadas GPS de la zona a estudiar.	144
Figura 6.1.2 Mapa topográfico de la zona a estudiar.	145

Figura 6.2.1 Representación de la temperatura y humedad del 6-8-10	146
Figura 6.2.2 Representación de la velocidad y dirección del viento del 6-8-10.....	146
Figura 6.2.3 Representación de la radiación solar del 6-8-10.....	147
Figura 6.2.4 Parte de la evolución del mapa de sombras del 6-8-10.....	147
Figura 6.2.5 Representación de la temperatura y humedad del 4-10-10.....	148
Figura 6.2.6 Representación de la velocidad y dirección del viento del 4-10-10.....	148
Figura 6.2.7 Representación de la radiación solar del 4-10-10.....	149
Figura 6.2.8 Parte de la evolución del mapa de sombras del 6-8-10.....	149
Figura 6.2.9 Representación de la temperatura y humedad del 3-12-10.....	150
Figura 6.2.10 Representación de la velocidad y dirección del viento del 3-12-10.....	150
Figura 6.2.11 Representación de la radiación solar del 3-12-10.....	151
Figura 6.2.12 Parte de la evolución del mapa de sombras del 3-12-10.....	151

Índice de tablas

Tabla 2.1.1 Parámetros del confort.....	9
Tabla 2.1.2 Factores del confort.....	9
Tabla 2.1.3 Sensación térmica dependiendo de la temperatura y la humedad relativa.....	11
Tabla 2.1.4 Sensación térmica dependiendo de la temperatura y velocidad del viento.....	12
Tabla 2.1.5 Niveles de iluminancias acorde con el esfuerzo visual.....	15
Tabla 3.1.1 Resumen de las características del Arduino duemilanove.....	31
Tabla 3.5.1 Especificaciones entregadas por el fabricante.....	73
Tabla 3.5.2 Especificaciones de la veleta dadas por el fabricante.....	76
Tabla 3.6.1 Esquema de pines TAOS TSL230.....	79
Tabla 3.6.2 Sensibilidad y escalado de frecuencia TAOS TSL230.....	79
Tabla 3.6.3 Diferentes características de funcionamiento del fotodiodo TLS230.....	80
Tabla 3.6.4 Diferentes características eléctricas del fotodiodo TLS230.....	81
Tabla 3.7.1 Descripción de pines DS18B20.....	93
Tabla 3.7.2 Especificaciones eléctricas DS18B20.....	94
Tabla 3.7.3 Descripción pines Sensor humedad.....	105
Tabla 3.7.4 Especificaciones sensor HIH-4021-004.....	106
Tabla 3.8.1 Potencia media mensual según inclinación.....	111
Tabla 4.2.1 Diferentes medidas del anemómetro respecto al PCE-AM81	120

Capítulo 1

Introducción

Este capítulo es la introducción al proyecto. Se pretende dar una idea global de la realización, justificando su ejecución, así como explicar la motivación para realizarlo. Además se analizan y explican los objetivos que se quieren alcanzar con la realización del mismo. Y por último, en este capítulo, se explica cual va a ser la estructuración (división en capítulos) del proyecto.

1.1 Motivación

Las consecuencias del cambio climático provocado por los gases GEI (Gases de Efecto Invernadero) se estudian en modelos de proyecciones realizados por varios institutos meteorológicos. Algunas de las previsiones realizadas por el IPCC (*Intergovernmental Panel on Climate Change*) son:

- Un calentamiento global de 0,2º C por decenio en los próximos veinte años.
- La disminución de la superficie de hielos y de nieve, lo que provocará un aumento del nivel del mar entre 4–6 m.
- Habrá impactos en los ecosistemas de tundra, bosques boreales y regiones montañosas, por su sensibilidad al incremento de temperatura.
- Disminuirán los recursos hídricos de regiones secas y latitudes medias, debido al descenso en las precipitaciones de lluvia y en la evapotranspiración; también ocurrirá en áreas surtidas por la nieve y el deshielo.
- Las proyecciones estiman una bajada del pH del océano entre 0,14 y 0,35 en este siglo. Esta acidificación progresiva tendrá efectos negativos sobre los ecosistemas marinos.

Según [1], está admitido que se producirá un daño generalizado por el aumento del nivel del mar, olas de calor, inundaciones y sequías más frecuentes e intensas. Su incidencia podría controlarse significativamente si se emprendiera una acción conjunta mundial de reducción de emisiones.

Para poder controlar las emisiones de gases GEI primero hay que saber de donde provienen dichas emisiones, producidas principalmente por los siguientes motivos:

- Transporte por carreteras.
- Electricidad.
- Aviación nacional.
- Transporte Marítimo nacional.
- Refino de petróleo.
- Industria.
- Cemento (uso no energético).
- Agricultura y ganadería.
- Residuos.
- Otros.

Cada sector provoca un porcentaje de emisiones siendo la electricidad, el transporte por carretera y la industria los que más producen. En la figura 1.1.2 se puede observar el porcentaje de cada sector en España en 2008, siendo la generación de la electricidad para uso doméstico de un 22%. Los datos han sido proporcionados por [2]

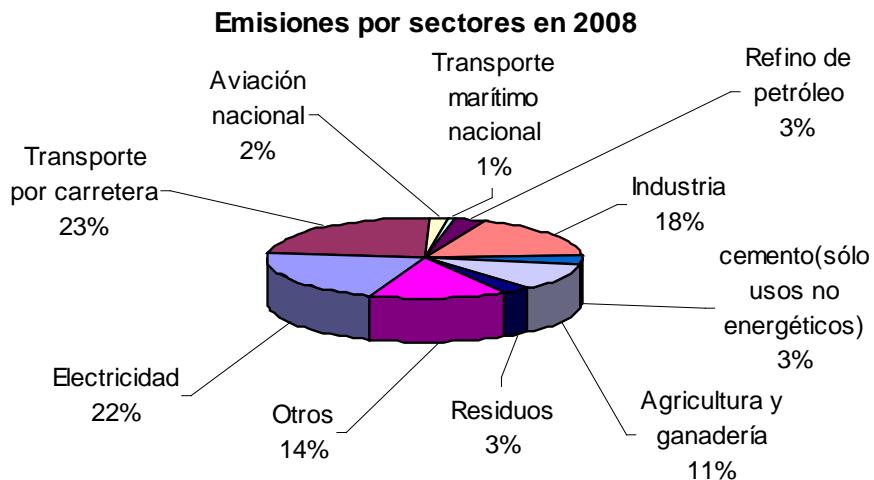


Figura 1.1.1 Emisiones de GEI por sectores en 2008 en España.

Las emisiones que son controlables por la población son las eléctricas y de transporte por carreteras, el 50% del total. La disminución de la emisión por transporte de carretera depende de que la industria de la automoción experimente un cambio radical en cuanto al concepto de fabricación de automóviles con nuevos tipos de combustible, también del aumento del uso de transporte público.

Para la reducción de las emisiones de consumo eléctrico hay que introducir un nuevo concepto en el colectivo social, el concepto de Arquitecturas o edificaciones bioclimáticas; éstas buscan el ahorro pasivo del consumo eléctrico mediante el diseño previo y el tipo de construcción. Se estima que pueden llegar a ahorrar un 50%, si aplicamos este tipo de edificación a casas o edificios. No sólo se pueden realizar en nuevas edificaciones, también se puede realizar en rehabilitaciones de las arquitecturas ya construidas, siendo las viviendas en las que más deberían tenerse en cuenta.

Por todo ello se intenta realizar algún proyecto que facilite o ayude a este tipo de arquitecturas. Se ha escogido analizar la zona donde se van a construir para que resulten más eficaces y eficientes.

1.2 Objetivos

El objetivo de este proyecto es el diseño y realización de un sistema integral, para poder hacer el análisis ambiental de una zona geográfica, para esto se requiere una serie de objetivos parciales.

El primer objetivo es la realización de una herramienta hardware, que capte las diferentes variables medioambientales y poder enviarlas a un servidor. El sistema deberá constar de un módulo central, que lo formarán un microcontrolador para poder realizar las diferentes funciones y procedimientos, un modulo de memoria, para poder almacenar los diferentes datos recogidos, además de un módulo adicional dedicado a proporcionar conectividad radio (mediante un módem GSM/GPRS) al sistema. También habrá que desarrollar diferentes sensores que capten las distintas variables. Ver en la figura 1.2.1.

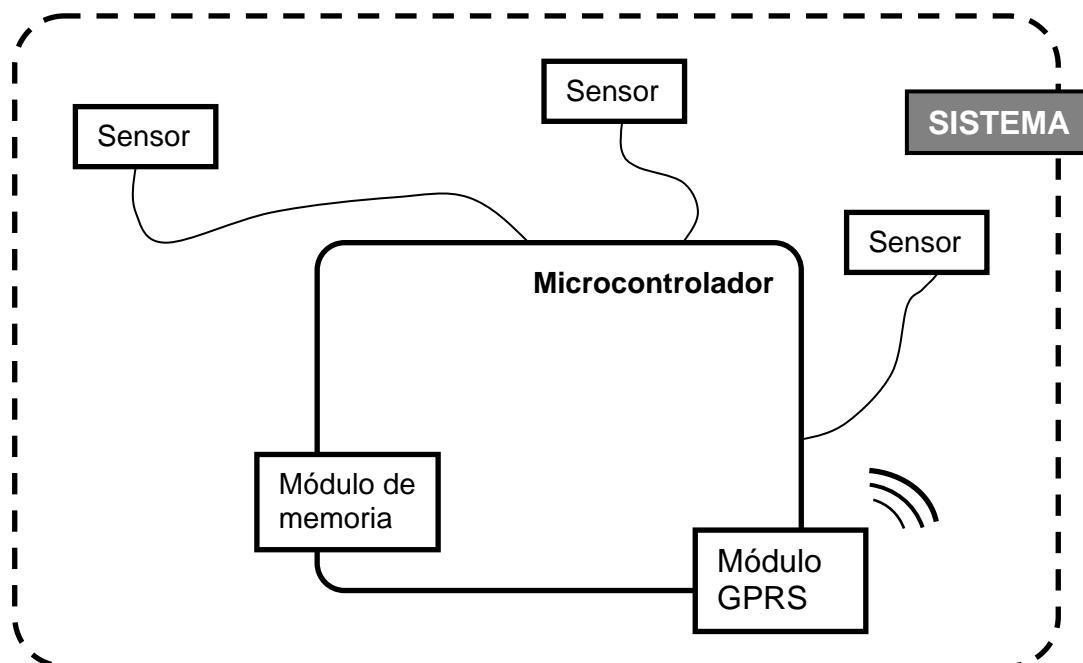


Figura 1.2.1 Esquema del sistema.

Otro objetivo es implementar una aplicación software para poder visualizar e interpretar las diferentes variables captadas por el sistema. Todo ello con el objetivo de poder hacer un estudio de las posibles actuaciones de una edificación que permita la monitorización in situ (entorno en el que se va a construir una edificación) y en tiempo real, de las variables climáticas más importantes de ese lugar.

El último objetivo será analizar una zona en concreto para poder decidir cuales son las actuaciones más eficientes a realizar sobre la futura construcción de una arquitectura o para la rehabilitación de una ya construida.

1.3 Organización

El proyecto está distribuido de la siguiente manera:

Capítulo 1: En la introducción del proyecto quedan reflejadas las motivaciones, objetivos y estructura del mismo.

Capítulo 2: Los conceptos más importantes de la arquitectura bioclimática. El confort y el clima como elementos más importantes a la hora de diseñar este tipo de arquitecturas.

Capítulo 3: El sistema como elemento de diseño. Los diferentes componentes para la realización física del sistema.

Capítulo 4: Verificación y pruebas que se realizan a cada componente del sistema y la integración de todos ellos para conformarlo.

Capítulo 5: Los resultados obtenidos por el sistema en una zona determinada y las posibles actuaciones para la edificación.

Capítulo 6: Las conclusiones por parte del autor así como las posibles líneas futuras que se pueden seguir a raíz del proyecto.

Capítulo 2

Introducción al diseño de edificaciones

En este capítulo se pretende dar una idea de cuales deberían ser los parámetros de planificación de las edificaciones, y como se deberían construir. Éstas, actualmente, se conciben de tal manera que se tienen en cuenta otros factores que no son los adecuados. Según [3], los esfuerzos han de orientarse al diseño de espacios más habitables, en el sentido de buscar el incremento de la calidad de vida de los individuos, favoreciendo el relax, la tranquilidad y el contacto con la naturaleza, a la vez que se mejoren todos los aspectos que intervienen en la funcionalidad y organización urbana.

Sin embargo los factores más usuales en el diseño de cualquier tipo de estructura, ya sea una vivienda o un edificio público o de cualquier otro tipo, son los factores económicos o factores comerciales. Es decir influyen más el precio y la venta de la

edificación que factores tan importantes como el confort, la eficiencia energética, la integración en el hábitat u otros factores importantes.

En este proyecto se han considerados cuales deben ser los aspectos más importante a la hora de diseñar edificaciones. El confort y el clima son los aspectos que se han tenido en cuenta y como estos influyen en la nueva concepción del tipo de edificaciones, las arquitecturas bioclimáticas

2.1 El confort en el acondicionamiento bioclimático

Según [16] el confort es aquello que produce bienestar y comodidades. Las sensaciones percibidas por el ser humano, ya sean agradables o desagradables, le impiden concentrarse en una tarea a realizar; por lo que se debe aspirar a la ausencia de cualquier sensación, siendo esto el confort. Al fin y al cabo, para realizar una actividad, el ser humano debe ignorar el ambiente, debe tener confort.

Organismos internacionales y especialistas se han encargado del estudio de este tema. No sólo han conceptualizado el término confort sino que además lo han clasificado en función de las variables implicadas, como pueden ser confort lumínico, térmico, acústico, etc. Se analizan los diferentes parámetros además de los factores que inciden en las sensaciones de bienestar, elaborándose tablas, fórmulas e incluso han marcado algunas pautas de diseño, tomando en cuenta los niveles de confort que se deben alcanzar para satisfacer a los usuarios.

Cuando se desea realizar algún tipo de edificación o a la hora de empezar una rehabilitación, es muy importante el confort a la hora de acometer el diseño, más aún cuando se vaya a tratar de una arquitectura bioclimática. Sin duda, el estudio de los niveles de confort permite tener en cuenta los parámetros y factores que intervienen en la edificación con el objetivo de lograr que se encuentre dentro de la llamada “zona de bienestar” a través del diseño adecuado.

Los parámetros y factores del confort son las condiciones de tipo ambiental, arquitectónico, personal y sociocultural que pueden afectar a la sensación de confort de un individuo determinado. Estos pueden influir en los distintos tipos de confort, afectando a las sensaciones térmicas, lumínicas, acústicas y/o visuales de una persona.

Por tanto su estudio resulta importante a la hora de realizar una edificación habitable o a la hora de determinar que grado de confort tiene una edificación, para poder ser restaurada o acondicionada con unos valores de confort determinados. Los parámetros de confort son aquellas condiciones propias del lugar que inciden en las sensaciones de los ocupantes.

Se sostiene que estas condiciones pueden variar con el tiempo y el espacio; se pueden clasificar en:

- Los **parámetros ambientales** son muy importantes y quizás son lo más estudiados. Pueden ser medidos y se han determinado rangos y valores dentro de los cuales operan las condiciones de confort de un individuo.
- Los **parámetros arquitectónicos** están directamente relacionados con las características de las edificaciones y la adaptabilidad del espacio, el contacto visual y auditivo que le permiten a sus ocupantes.

Parámetros del Confort		
Parámetros Ambientales	Temperatura del aire Humedad relativa Velocidad del aire Temperatura radiante Radiación solar Niveles de ruido	Todos tienen variabilidad temporal y espacial
Parámetros Arquitectónicos	Adaptabilidad del espacio Contacto visual y auditivo	

Tabla 2.1.1 Parámetros del confort.

Por otra parte, distinguimos los factores de confort como aquellas condiciones propias de una persona que determinan su respuesta al ambiente. Son independientes de las condiciones exteriores y se relacionan con las condiciones biológicas, fisiológicas, sociológicas o psicológicas de los individuos. Se pueden clasificar del modo siguiente:

Factores del confort		
Factores Personales	Metabolismo (Alimentación, Actividad)	Base o Basal De trabajo o Muscular
	Ropa. Grado de aislamiento	
	Tiempo de permanencia (Aclimatación)	
	Salud y color de piel	
	Historial térmico, lumínico, visual y acústico	Inmediato Mediato (situación geográfica, época del año)
	Sexo, edad, peso (constitución corporal)	
Factores Socioculturales	Educación Expectativas para el momento y lugar considerados	

Tabla 2.1.2 Factores del confort.

Los más analizados han sido los factores personales, ya que son más fáciles de cuantificar. Se establecen fórmulas y formas de medición que parametrizan estos factores, con el objetivo de evaluar las condiciones de un lugar en función de las características del usuario y de las actividades que lleva a cabo. Por su parte, los socio-culturales, por ser parámetros subjetivos, son difíciles de medir y solamente se evalúan de forma cualitativa.

2.1.1 Confort Térmico

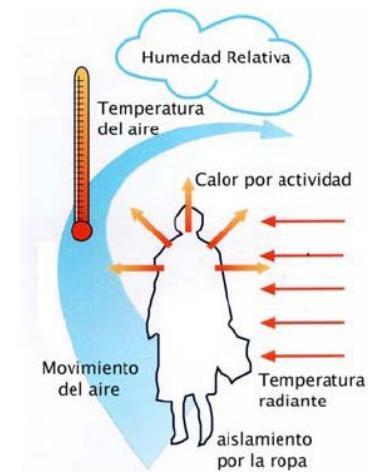


Figura 2.1.1 Principales factores que afectan al confort térmico.

Según [4], el confort Térmico puede definirse como la ausencia de malestar térmico. En fisiología se dice que hay confort térmico cuando no tienen que intervenir los mecanismos termorreguladores del cuerpo para una actividad sedentaria y con un ligero aislamiento de ropa. Esta situación puede registrarse mediante índices que no deben ser sobre pasados para que no se pongan en funcionamiento los sistemas termorreguladores (metabolismo, sudoración y otros). El confort térmico es una de las variables más importantes a la hora del diseño de estructuras y espacios urbanos.

Los factores y parámetros más importantes que influyen directamente en el confort térmico son los siguientes:

A. Temperatura del Aire: Es la Temperatura del aire a la sombra., supone uno de los parámetros más importantes a la hora de determinar en grado de confort térmico, puesto que determina si un individuo siente frío o calor. Ésta sensación térmica depende de la relación entre el calor que produce el metabolismo del cuerpo y el que disipa hacia el entorno. Si es mayor el primero, la sensación es de calor; si es mayor el segundo, la sensación es de frío.

B. Temperatura radiante: Tiene en cuenta el calor emitido por radiación de los elementos del entorno. Se mide con un termómetro de bulbo, que tiene el depósito de mercurio encerrado en una esfera o *bulbo* metálico de color negro, para asemejarlo lo más posible a un cuerpo negro y así absorber la máxima radiación. Para anular en lo posible el efecto de la temperatura del aire, el bulbo negro se aísla en otro bulbo en el que se ha hecho el vacío. Sirve para dar una idea de la sensación térmica.

C. La humedad relativa: Otro parámetros de importancia, ya que afecta en gran medida a la sensación térmica. La sensación térmica es la combinación de la temperatura del aire y la humedad relativa para determinar la temperatura que el cuerpo humano percibe.

A temperaturas altas, el nivel de humedad relativa necesario para que la temperatura percibida sea superior a la temperatura real, es más bajo que a temperaturas más bajas. Por ejemplo, a 27°C, la temperatura percibida y la real serán iguales si la humedad relativa es del 45%, pero a 43°C, la humedad relativa será sólo del 17%. A una

temperatura inferior a 20°C la humedad no eleva la temperatura percibida. Todo esto lo podemos observar en la tabla 2.1.3, la cual relaciona la temperatura y la humedad relativa con la temperatura que percibimos.

TEMPERATURA (° C)	HUMEDAD RELATIVA (%)																				
	0	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95	100
20	16	16	17	17	17	18	18	19	19	19	19	19	20	20	20	21	21	21	21	21	21
21	18	18	18	19	19	19	19	19	20	20	20	20	21	21	21	22	22	22	22	22	23
22	19	19	19	20	20	20	20	20	21	21	21	21	22	22	22	23	23	23	23	23	24
23	20	20	20	20	21	21	22	22	23	23	23	23	24	24	24	24	24	24	24	25	25
24	21	21	22	22	22	23	23	23	24	24	24	24	25	25	25	25	26	26	26	26	26
25	22	23	23	23	24	24	24	24	24	25	25	25	26	26	26	27	27	27	27	28	28
26	24	24	24	24	25	25	25	26	26	26	26	27	27	27	27	28	28	29	29	29	30
27	25	25	25	25	26	26	26	27	27	27	27	28	28	29	29	30	30	31	31	31	33
28	26	26	26	26	27	27	27	28	28	28	29	29	29	30	31	32	32	33	34	34	36
29	26	26	27	27	27	28	29	29	29	29	30	30	31	33	33	34	35	35	37	38	40
30	27	27	28	28	28	28	29	29	30	30	31	32	33	34	35	36	37	39	40	41	45
31	28	28	29	29	29	29	30	31	31	31	33	34	35	36	37	39	40	41	45	45	50
32	29	29	29	29	30	31	31	33	33	34	35	35	37	39	40	42	44	45	51	51	55
33	29	29	30	30	31	31	33	33	34	34	35	36	38	39	42	43	45	49	53	54	55
34	30	30	31	31	32	34	34	35	36	37	38	41	42	44	47	48	50	52	55		
35	31	32	32	32	33	35	35	37	37	40	40	44	45	47	51	52	55				
36	32	33	33	34	35	36	37	39	39	42	43	46	49	50	54	55					
37	32	33	34	35	36	38	38	41	41	44	46	49	51	55							
38	33	34	35	36	37	39	40	43	44	47	49	51	55								
39	34	35	36	37	38	41	41	44	46	50	50	55									
40	35	36	37	39	40	43	43	47	49	53	55										
41	35	36	38	40	41	44	45	49	50	55											
42	36	37	39	41	42	45	47	50	52	55											
43	37	38	40	42	44	47	49	53	55												
44	38	39	41	44	45	49	52	55													
45	38	40	42	45	47	50	54	55													
46	39	41	43	45	49	51	55														
47	40	42	44	47	51	54	55														
48	41	43	45	49	53	55															
49	42	45	47	50	54	55															
50	42	45	48	50	55																

Tabla 2.1.3 Sensación térmica dependiendo de la temperatura y la humedad relativa.

D. Velocidad del aire: Constituye un parámetro a examinar puesto que el aire puede considerarse como un elemento agradable que puede refrescar o calentar, pero dependiendo de la velocidad alcanzadas por éste pueden estimarse más como un inconveniente que una ventaja. Además habrá que considerar la temperatura del aire, ya que en el caso de que la temperatura del aire esté por debajo de la temperatura de la piel, se generará una sensación de frescor, pero si es al revés, el cuerpo sentirá calor. La temperatura de sensación es la temperatura que nota un cuerpo humano debido a la combinación de la temperatura y la velocidad del viento. Excepto a temperaturas altas, el viento sirve para aumentar la sensación de frío, ya que el viento favorece la evaporación a través de la piel y esto provoca que se disipe calor del cuerpo.

El movimiento del aire aumenta la velocidad a la que la temperatura de un objeto alcanza la temperatura del aire ambiente. Los humanos perciben o "sienten" que este aumento de la velocidad de enfriamiento del cuerpo es como el frío causado por el viento. Se puede observar en la tabla 2.1.4 como se relaciona la velocidad del viento y la sensación térmica.

Viento km/h	Temperatura (°C)																																				
	10	7,5	5	2,5	0	-2,5	-5	-7,5	-10	-12,5	-15	-17,5	-20	-22,5	-25	-27,5	-30	-32,5	-35	-37,5	-40	-42,5	-45	-47,5	-50	-52,5	-55										
8	7,5	5	2,5	0	-2,5	-5	-7,5	-10	-12,5	-15	-17,5	-20	-22,5	-25	-27,5	-30	-32,5	-35	-37,5	-40	-45	-47,5	-50	-52,5	-55	-57,5	-60	-62,5	-65	-67,5							
16	5	2,5	-2,5	-5	-7,5	-10	-12,5	-15	-17,5	-20	-22,5	-25	-27,5	-30	-32,5	-35	-37,5	-40	-45	-47,5	-50	-52,5	-55	-57,5	-60	-62,5	-65	-67,5	-70	-72,5	-75	-77,5					
24	2,5	0	-5	-7,5	-10	-12,5	-17,5	-20	-25	-27,5	-32,5	-35	-37,5	-42,5	-45	-47,5	-52,5	-55	-57,5	-60	-65	-67,5	-72,5	-75	-77,5	-80	-82,5	-85	-87,5	-90	-92,5	-95					
32	0	-2,5	-7,5	-10	-12,5	-17,5	-22,5	-25	-30	-35	-37,5	-42,5	-45	-47,5	-50	-52,5	-55	-57,5	-60	-65	-67,5	-70	-72,5	-75	-77,5	-80	-82,5	-85	-87,5	-90	-92,5	-95					
40	0	-5	-7,5	-10	-15	-17,5	-22,5	-25	-30	-32,5	-37,5	-40	-45	-47,5	-52,5	-55	-60	-62,5	-67,5	-70	-75	-77,5	-82,5	-85	-90	-95	-98	-100	-102	-104	-106	-108	-110				
48	-2,5	-5	-10	-12,5	-17,5	-20	-25	-27,5	-32,5	-35	-40	-42,5	-47,5	-50	-55	-57,5	-62,5	-67,5	-72,5	-75	-77,5	-80	-85	-90	-95	-100	-102	-104	-106	-108	-110	-112	-114	-116	-118	-120	
56	-2,5	-7,5	-10	-12,5	-17,5	-20	-25	-30	-32,5	-37,5	-42,5	-45	-50	-52,5	-57,5	-60	-65	-67,5	-72,5	-75	-80	-82,5	-85	-90	-95	-100	-102	-104	-106	-108	-110	-112	-114	-116	-118	-120	
64	-2,5	-7,5	-10	-15	-20	-22,5	-27,5	-30	-35	-37,5	-42,5	-45	-50	-55	-60	-62,5	-65	-70	-75	-75,5	-82,5	-85	-90	-92,5	-95	-97,5	-100	-102	-104	-106	-108	-110	-112	-114	-116	-118	-120
Vientos superior es a los 64 km/h	PELIGROSO								MUY PELIGROSO Las partes del cuerpo expuestas al viento se pueden congelar en 1 minuto								EXTREMADAMENTE PELIGROSO Las partes del cuerpo expuestas al viento se pueden congelar en 30 segundos												PELIGRO DE CONGELACIÓN DEL CUERPO HUMANO EXPUESTO AL VIENTO SIN LA APROPIADA VESTIMENTA								

Tabla 2.1.4 Sensación térmica dependiendo de la temperatura y velocidad del viento.

Todos estos factores determinan el confort térmico de una persona, por lo que habrá que tenerlos en cuenta a la hora de diseñar cualquier tipo de espacio urbano, ya sea una edificación o una estructura, o a la hora de realizar una rehabilitación de éstos.

En el presente proyecto se generarán diferentes herramientas para poder determinar cada parámetro de confort y así poder emplear las diferentes técnicas que existen para evaluar un espacio desde el punto de vista térmico y definir que actuaciones se pueden llevar a cabo.

2.1.1.1 Técnicas para evaluar un ambiente Térmico

El estudio de confort térmico ha permitido el desarrollo de diferentes índices térmicos subjetivos y objetivos, así como gráficas que pretenden identificar el modo de incidencia de los diferentes factores y parámetros sobre el equilibrio térmico. De estos, se pueden destacar algunos como:

A. El Diagrama Psicométrico: Es uno de los esquemas más utilizados para la evaluación del confort térmico debido probablemente a la sencillez en su manejo.

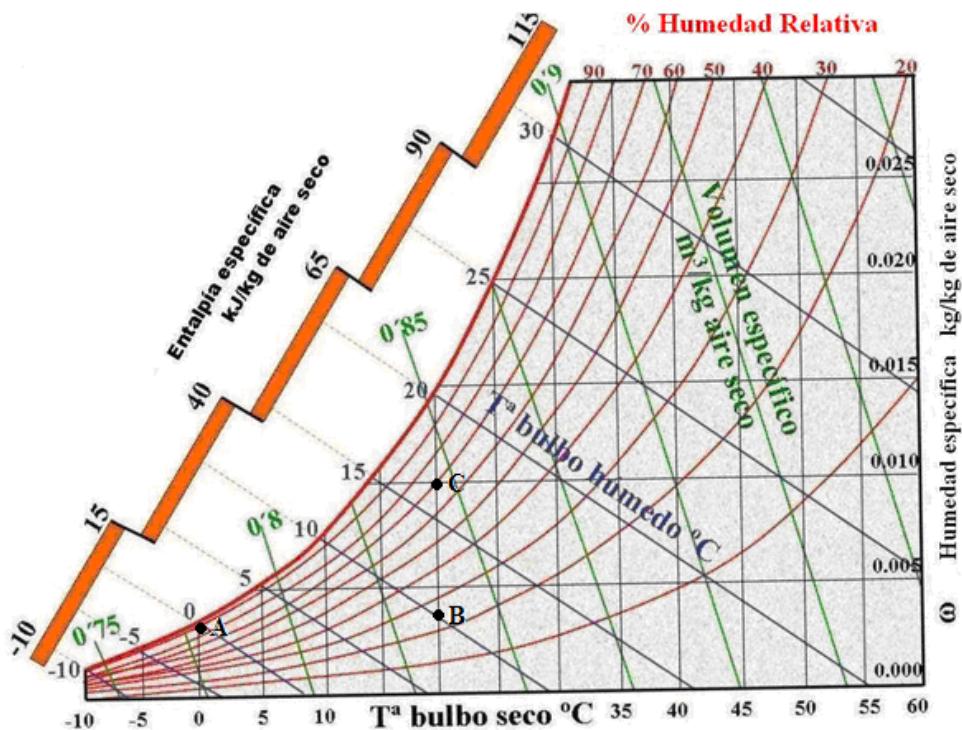


Figura 2.1.2 Diagrama psicométrico.

El diagrama psicométrico muestra la interdependencia de la humedad relativa (escala de la izquierda), la temperatura seca (escala horizontal) y la masa de vapor de agua por masa de aire seco con su equivalencia en presión de vapor, en mbar (escala de la derecha). Ejemplo de utilización:

- Si la temperatura seca exterior del aire es 0°C y el aire contiene $3,4 \text{ g/kg}$ de aire seco, la humedad relativa es del 90 %, y existe una presión de vapor de $5,4 \text{ mbar}$, lo que podría estar representando una típica condición del aire en invierno. **Punto A del diagrama.**
 - Este mismo aire, con la misma cantidad de agua por masa de aire seco, calentado a 20°C pasa a tener una humedad relativa del 23 %, lo que demuestra lo que sucede cuando introducimos este aire exterior para ventilación y lo calentamos. Punto B del diagrama.
 - Si a este aire le aportamos 7 g/kg como resultados de actividades normales en un edificio, a la misma temperatura, su humedad relativa ascenderá al 70 % con una presión de vapor de $16,5 \text{ mbar}$, y un contenido de $10,4 \text{ g/kg}$. **Punto C del diagrama.**
 - Este mismo aire alcanzará la saturación para temperaturas menores a $14,5^{\circ}\text{C}$.

B. Análisis del índice y del Diagrama Psicométrico de Givoni: Se determina la llamada zona de confort la que, coincide con aquella zona dentro de la cual se mantienen unos rangos climáticos en los cuales una persona manifiesta estar térmicamente confortable. Para esto Givoni llega a la síntesis en un climograma realizado sobre un diagrama psicométrico donde se traza una zona de confort higrotérmico para invierno y verano. Luego propone otras zonas donde es posible alcanzar el confort mediante la incorporación y/o aplicación de estrategia de diseño pasivo.

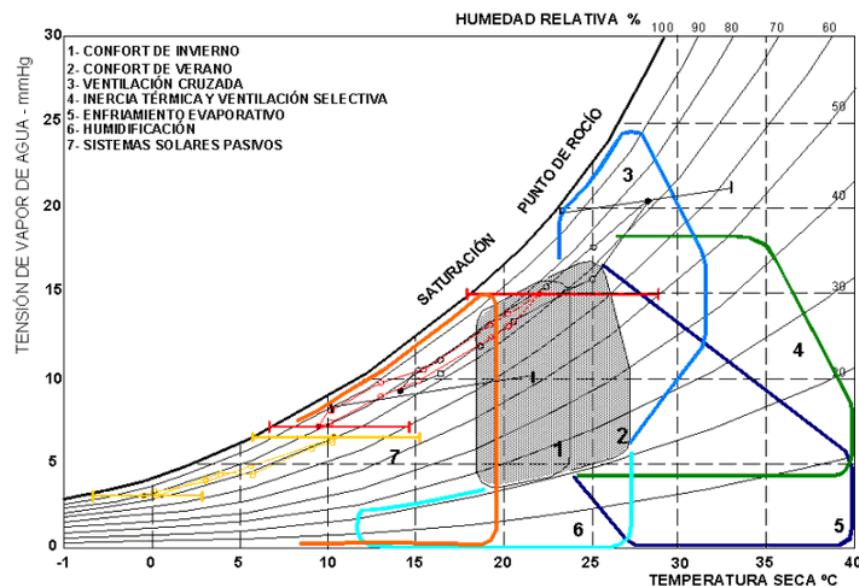


Figura 2.1.3 Diagrama Psicométrico de Givoni.

Fuera de estas zonas se vuelve necesario el uso de sistemas termomecánicos de acondicionamiento ambiental ya sea para calefacción como para refrigeración. Su modelo permite, mediante la inserción en el climograma de valores de temperatura y humedad medios mensuales, trazar las características bioclimáticas de un sitio.

De su interpretación, surgen estrategias de diseño con las que resolver un proyecto de una edificación a fin de mantenerla en confort sin uso de energía adicional a la del sol, el viento, las temperaturas día-noche y la humedad ambiente.

Según Givoni, la fórmula general para hallar el punto de confort es:

$$S = [(M - W) \pm C \pm R] \cdot (1/re)$$

Donde:

S: Grado de sudación requerido, en equivalente kcal/h.

M: Metabolismo. kcal/h.

W: Energía metabólica transformada en trabajo mecánico kcal/h.

C: Intercambio de calor por convección. kcal/h.

R: Intercambio de calor por radiación. kcal/h.

C. Carta de Olgyay: Se trata de una carta bioclimática en la cual se define la zona de confort, las variables que la afectan y los mecanismos correctores. Para esto, se señalan los valores medios de temperatura, humedad relativa, temperatura radiante, potencia de radiación y velocidad del viento que estarían dentro o fuera de esta zona. Para trabajar con ella se deben introducir los valores medios de los parámetros climáticos de cada mes del año y unir con líneas para ver en que parte de la gráfica se encuentran.

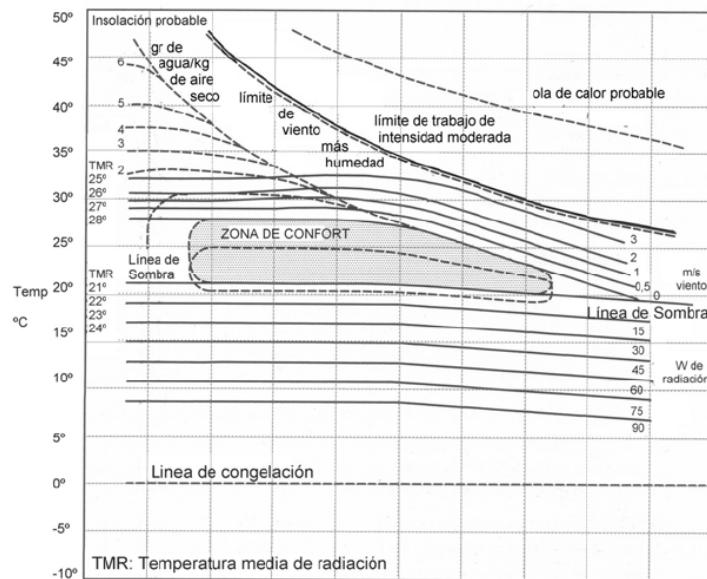


Figura 2.1.4 Carta de Olgyay.

Es importante señalar que estas tablas determinan las decisiones a tomar en el diseño para responder adecuadamente al contexto desde el punto de vista térmico. Para ello se han desarrollado las herramientas necesarias para el análisis y la posterior edificación o rehabilitación de una edificación.

2.1.2 Confort Visual

Teniendo en cuenta la definición del confort, el confort visual es el conjunto de parámetros que hacen ver sin ningún esfuerzo. Según [5] el primer parámetro es el nivel de iluminación o iluminancia, (unidad: Lux).

Esfuerzo Visual	Lux recomendados
Esfuerzo muy alto	1,000 lux
Esfuerzo visual alto o muy alto de poca duración	750 lux
Esfuerzo visual medio o alto poca duración	500 lux
Esfuerzo visual bajo o medio poca duración	250 lux

Tabla 2.1.5 Niveles de iluminancias acorde con el esfuerzo visual.

Otro parámetro del confort visual es el color de la luz, basándose en dos conceptos: La temperatura de color (T_c), que expresa el color de una fuente de luz, comparándola con la luz emitida por un cuerpo negro a una temperatura (en $^{\circ}\text{K}$) determinada, el índice de rendimiento de color (IRC), relaciona porcentualmente por comparación con una fuente de luz perfecta, la reproducción del color de los objetos. A mayor porcentaje de IRC, mayor será el nivel de reproducción de color.

Valores superiores al 85% del IRC son adecuados para espacios donde el color es importante concordando con T_c superiores a $4,500^{\circ}\text{K}$. Mientras que niveles inferiores de 70% de IRC se utilizan para espacios donde el reconocimiento cromático es poco importe. En la figura 2.1.5 podemos ver esta relación.

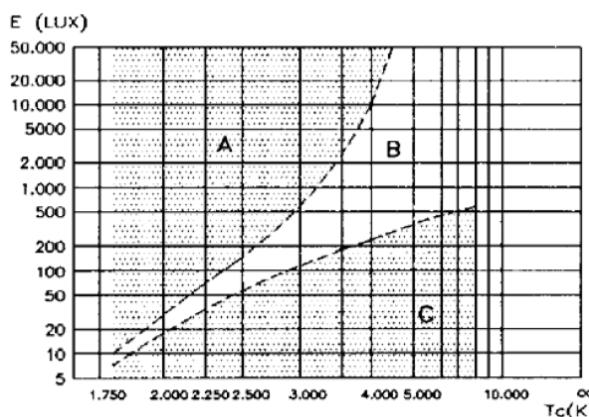


Figura 2.1.5 Grafica de Kruithof.

2.1.3 Confort Acústico

El nivel de confort acústico es el nivel de ruido a partir del cual el sonido provocado por las actividades humanas, las infraestructuras o las industrias resulta pernicioso para el descanso, la comunicación y la salud de las personas. En el ámbito del confort, se entiende ruido como el exceso de sonido que altere las condiciones normales del ambiente de una zona o edificación. Es lo que se conoce como contaminación acústica.

Un informe de la Organización Mundial de la Salud (OMS), considera los 50dB, como el límite superior deseable. En España, se establece como nivel de confort acústico los 55 dB. Por encima de este nivel, el sonido resulta perjudicial para el descanso y la comunicación.

Según estudios de la Unión Europea (2005): 80 millones de personas están expuestas diariamente a niveles de ruido ambiental superiores a 65dB y otras 170 millones, lo están a niveles entre 55-65 dB. Algunos efectos del ruido sobre la salud:

- Enfermedades fisiológicas: Se pueden producir en el trabajo o ambientes sonoros en torno a los 100 decibelios, algunas tan importantes como la pérdida parcial o total de la audición.

- Enfermedades psíquicas: Producidas por exceso de ruido, se pueden citar el estrés, las alteraciones del sueño, disminución de la atención, depresión, falta de rendimiento o agresividad.
- Enfermedades sociológicas: Alteraciones en la comunicación, el rendimiento, etc.

La percepción de un sonido depende de la frecuencia de dicho sonido. Por lo tanto la percepción de las personas sobre un ruido también dependerá de la frecuencia; en la figura 2.1.6, se puede observar si el sonido es perjudicial, audible o apto para la comunicación según su nivel y frecuencia.

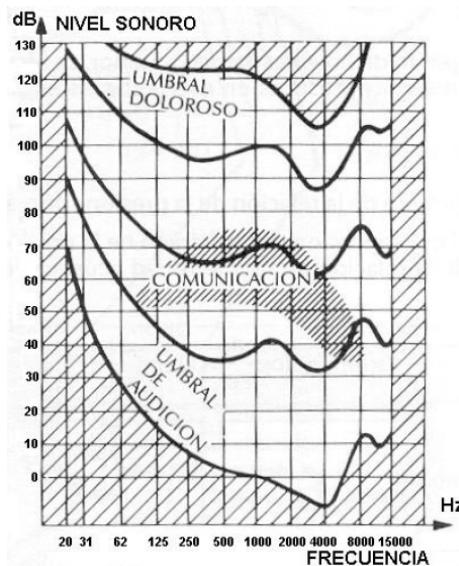


Figura 2.1.6 Grafica ruido

2.2 El clima como característica extrínseca de la vivienda.

El clima, a lo largo de la historia se ha definido de diferentes formas dependiendo de las diferentes tipologías de personas implicadas en el estudio, ya fueran geógrafos, arquitectos, climatólogos, etc. Aunque con diferentes matices todos afirman que consiste en una serie de factores o fenómenos atmosféricos periódicos que ocurren en una región determinada y que están relacionados con un conjunto de elementos geográficos y estadísticos, tales como la latitud, altitud, relieve, etc. El clima, en la actualidad, se define como un conjunto de factores o fenómenos atmosféricos y meteorológicos que caracterizan una región y determinan las condiciones ecológicas propias del lugar.

Según [6], todas aquellas características relacionadas con el medio ambiente constituyen una serie de variables dependientes e independientes que no solamente influyen en la edificación y su comportamiento, sino sobre el hombre y su confort.

El clima se entiende como parte del medio ambiente, puede ser analizado como variable independiente si se toman en cuenta los factores climáticos, o bien como una variable dependiente, si se consideran los diferentes parámetros ambientales.

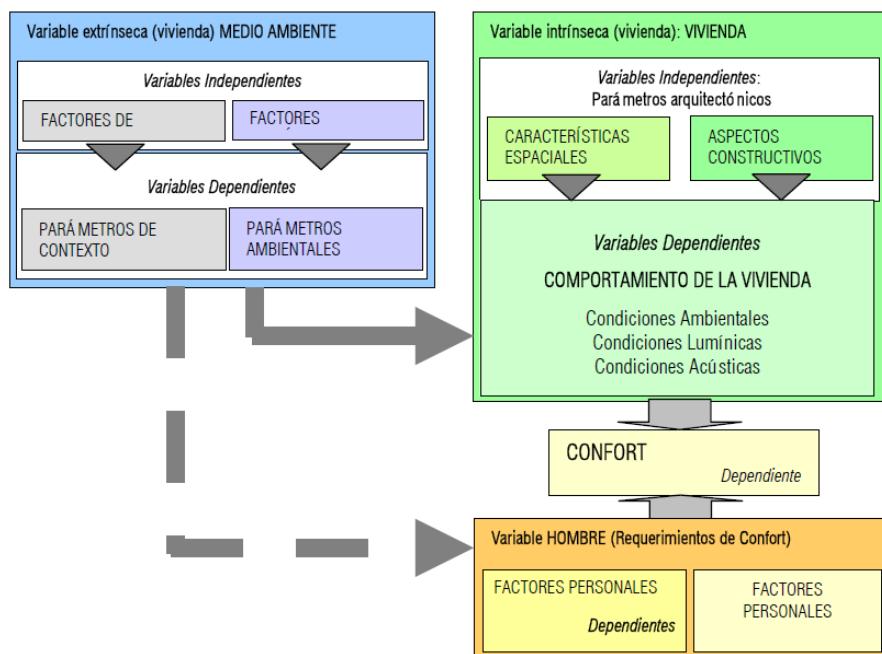


Figura 2.2.1 Interrelación del clima con el confort personal.

En la figura 2.2.1 se puede observar la relación de los parámetros ambientales que conforman el clima, el confort y la edificación, llegando a la conclusión de que en el diseño y acondicionamiento arquitectónico resulta imprescindible tener en cuenta los parámetros ambientales, cada uno de sus valores, junto con las diferentes variables del contexto y los parámetros del confort, nos proporciona las bases y criterios a seguir para establecer un correcto intercambio energético entre el interior y el exterior de la edificación y, así, generar bienestar en sus ocupantes.

2.2.1 Factores del clima

Los factores del clima son las propiedades físicas, no variables, de una zona que condicionan el clima. Habrá que tenerlos en consideración ya que determinan cualquier tipo de estructura desde el punto de vista medioambiental y de confort. Algunos parámetros son:

A. Situación geográfica o latitud: Distancia angular desde cualquier punto sobre la superficie terrestre hasta el ecuador. Su análisis es importante porque determina los procesos térmicos como la incidencia solar y el movimiento de rotación de la tierra que genera la aparición de sistemas de vientos variables y zonas de calma. La situación geográfica es tomada en consideración porque es de gran ayuda en la observación de la incidencia de los rayos solares sobre el lugar donde se ubican las viviendas y, además, porque afecta a la temperatura, el asoleamiento de muros, ventanas y cubiertas, así como la posible ubicación e inclinación de colectores solares.

B. Altura sobre el nivel del mar o altitud: Influyen en el clima de un lugar, de modo que los valores de la temperatura se reducen en la medida que se aumenta la altitud. No obstante, hay que recordar que ésta actúa en conjunto con los otros factores climáticos.

C. El factor continental: Situación de un lugar en función de la presencia o ausencia de masas de agua, la cual condiciona las variaciones de temperatura y humedad de la edificación. Las masas de agua actúan como reguladores térmicos, enfriando el ambiente de la noche y templándolo de día. Además, influyen en el movimiento de las masas de aire que generan brisas o vientos.

D. El factor orográfico: Factor de gran relevancia por su influencia en el clima de una zona, por la presencia o ausencia de montañas que generen obstrucción o incidencia de los rayos solares y el viento. Modifica las características microclimáticas de una zona con respecto a las de la región donde se encuentran, si la orografía es relevante.

E. La topografía, el relieve y la naturaleza de la superficie terrestre: La forma del relieve determinará la incidencia de los vientos, la radiación solar recibida y el % de reflexión de sus superficies, en función de los materiales que componen el suelo, así como el tipo de vegetación y la humedad del lugar. Estos factores, debido a la posibilidad de variación en una misma zona geográfica, son capaces de modificar las condiciones microclimáticas generando una variedad de subtipos climáticos dentro de una zona geográfica.

F. Vegetación y fauna: Se conciben como factores biológicos del clima, ya que dependiendo del tipo de plantas y de animales de una región podemos determinar el tipo climático. La vegetación influye directamente en la temperatura y humedad, así como en la radiación solar que es recibida por la superficie terrestre y en % de energía reflejada.

G. Urbanización o modificaciones del entorno: Este factor es analizado como un factor de tipo tecnológico, ya que las modificaciones del entorno se deben principalmente a la presencia de edificios, pavimentos, vehículos, industrias, falta de árboles u otros elementos propios de la urbanización, los cuales modifican el clima disminuyendo o aumentando la temperatura, la humedad, incrementando los niveles de ruido y de contaminación del agua, aire o del suelo e, incluso, permitiendo o no el paso directo de los rayos solares. Esta es probablemente una de las variables más determinantes de las condiciones climáticas de una estructura, ya que al comparar las condiciones de las viviendas ubicadas en entornos totalmente urbanizados con otras situadas en zonas rurales, probablemente se evidenciarán diferencias en los parámetros ambientales, a pesar de encontrarse dentro de la misma zona climática.

2.2.2 Elementos climáticos o parámetros ambientales

Según [6], los elementos del clima son las condiciones, variables o propiedades físicas de la atmósfera utilizadas para medir y describir el clima en un momento dado, lo cual quiere decir que afectan al tiempo atmosférico. Son de gran importancia para el diseño arquitectónico y pueden ordenarse siguiendo una secuencia aproximada de causa y efecto. Entre ellos tenemos:

- **La radiación solar.**
- **La temperatura.**
- **La humedad.**
- **El viento.**
- **Las precipitaciones.**
- **La radiación de onda larga.**
- **La presión atmosférica.**

El análisis de estos elementos es fundamental para el diseño bioclimático, ya que sirven para generar diferentes actuaciones que resuelven los problemas de acondicionamiento ambiental. Su conocimiento proporciona las características generales del sitio donde se situará la arquitectura. Así mismo, permite establecer las características de la edificación en función de responder positivamente al medio ambiente y prever el posible comportamiento de la misma. Estos parámetros serán estudiados con más detenimientos en el capítulo 3 puesto que son en los que se centra este proyecto.

2.2.3 Zonas climáticas

Cálidos

- **Clima ecuatorial:** Temperaturas altas y casi constante durante todo el año, además de lluvias abundantes y regulares. Se localiza en las zonas cercanas al ecuador terrestre, en latitudes bajas.
- **Clima tropical:** Se definen dos estaciones: una seca y una húmeda. Las temperaturas medias son elevadas y bastante uniformes a lo largo del año. No conoce el invierno. La humedad relativa del aire es muy elevada y existen abundantes precipitaciones.
- **Clima desértico y semidesértico:** Pocas precipitaciones. La temperatura media anual supera los 18°C, y las temperaturas diarias presentan fuertes oscilaciones.

Templados

- **Clima subtropical húmedo:** Se da en el interior de continentes o en los litorales al este de tales continentes, entre latitudes de 25° y 40°. Los veranos son calurosos, pero en cambio son húmedos debido a las masas tropicales inestables.

- **Clima mediterráneo:** Inviernos templados; y los veranos secos y calurosos. Pluviosidad bastante escasa y concentrada en las estaciones intermedias con temperaturas muy calurosas en verano y relativamente suaves en invierno.
- **Clima oceánico:** Temperaturas suaves, y abundantes precipitaciones. Los inviernos son suaves y los veranos frescos con oscilación térmica pequeña. Las precipitaciones son abundantes y están bien distribuidas aunque con un máximo hibernal.
- **Clima continental:** Diferencias de temperaturas entre invierno y verano, así como entre día y noche. Se producen por localizarse en el interior de los continentes o por regiones aisladas por cadenas montañosas que impiden la influencia marítima.

Muy fríos

- **Clima polar:** Temperaturas por debajo de 0°C, las precipitaciones son muy escasas. La humedad en el aire es muy alta y el viento suele ser bastante intenso.

2.3 Arquitectura bioclimática

La arquitectura bioclimática consiste en diseñar edificios teniendo en cuenta las condiciones climáticas, aprovechando los recursos disponibles (sol, vegetación, lluvia, vientos) para disminuir los impactos ambientales, intentando reducir los consumos de energía. Pueden conseguir un gran ahorro e incluso llegar a ser sostenible en su totalidad. Aunque el coste de construcción puede ser mayor, tiende a ser rentable, ya que el precio de la vivienda se compensa con la disminución de los recibos de energía.

La arquitectura bioclimática está muy relacionada con la arquitectura pasiva, la arquitectura energéticamente eficiente y la arquitectura sostenible; de hecho en muchos aspectos se considera a la arquitectura bioclimática un compendio de estas arquitecturas.

La arquitectura pasiva diseña edificios que minimizan el uso de equipos convencionales para la calefacción y refrigeración. La sostenible busca aprovechar los recursos naturales de tal modo que minimicen el impacto ambiental de los edificios sobre el medio ambiente y sus habitantes. La energéticamente eficiente minimiza el uso de las energías convencionales. La arquitectura bioclimática busca todas estas consideraciones a partir del clima del lugar y de las consideraciones del terreno (en particular la energía no renovable), a fin de ahorrar y hacer un uso racional de la misma.

Según [4], el hecho de que la construcción hoy en día no tenga en cuenta los aspectos bioclimáticos, se une al poco respeto por el ambiente que inunda a los países desarrollados y en vías de desarrollo, que no ponen los suficientes medios para frenar el desastre ecológico que dejamos a nuestro paso.

2.3.1 Principales actuaciones para una Arquitectura bioclimática.

A. Ubicación: Determina las condiciones climáticas (macro- y micro-climáticas) con las que tiene que "relacionarse" la vivienda. Las condiciones microclimáticas vienen determinadas por la latitud y la región en la que se ubique la vivienda. Estas condiciones vienen definidas por las temperaturas (máxima, media y mínima anual), pluviometría, radiación solar incidente, y dirección del viento dominante. Las condiciones microclimáticas están condicionada por la presencia de accidentes geográficos locales que pueden modificar de forma significativa las condiciones macroclimáticas.

A la hora de elegir la ubicación de una vivienda, además de tener en cuenta las condiciones macro- y micro-climáticas, es importante analizar también la pendiente del terreno, la existencia de relieves orográficos, la presencia de corrientes de agua, la presencia de masas boscosas y la existencia de otros edificios.

B. Orientación: Con una orientación de los huecos acristalados al sur en el Hemisferio Norte, o al norte en el Hemisferio sur, esto es, hacia el ecuador, se capta más radiación solar en invierno y menos en verano, aunque para las zonas más cálidas (con temperaturas promedio superiores a los 25°C) es sustancialmente más conveniente colocar los acristalamientos en el sentido opuesto, esto es, dándole la espalda al ecuador; de esta forma en verano, la cara acristalada sólo será irradiada por el sol en los primeros instantes del alba y en los últimos momentos del ocaso, y en Invierno el sol nunca bañará esta fachada, reduciendo el flujo calorífico al mínimo y permitiendo utilizar conceptos de diseño arquitectónico propios del uso del cristal.

C. Adaptación a la temperatura: Es quizá en este punto donde es más común incidir cuando se habla de arquitectura bioclimática. Lo más habitual, es aprovechar al máximo la energía térmica del sol cuando el clima es frío, por ejemplo, para calefacción y agua caliente sanitaria. Se aprovechar el efecto invernadero de los cristales y procurar tener las mínimas pérdidas de calor (buen aislamiento térmico) si hay algún elemento calefactor.

Sistemas solares pasivos

Los sistemas solares pasivos se utilizan, principalmente, para captar y acumular el calor proveniente de la energía solar. Se los llama pasivos ya que no se utilizan otros dispositivos electromecánicos para recircular el calor. Esto sucede por principios físicos básicos como la conducción, radiación y convección del calor.

- **Ganancia directa:** es el sistema más sencillo e implica la captación de la energía del sol por superficies vidriadas, que son dimensionadas para cada orientación y en función de las necesidades de calor del edificio o local a climatizar.

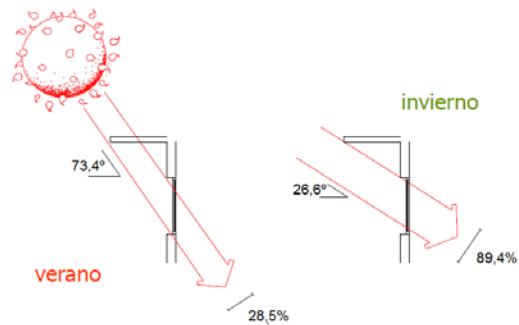


Figura 2.3.1 Ganancia directa.

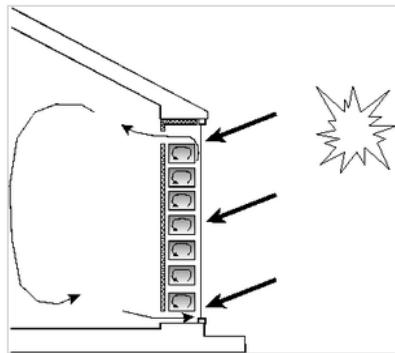


Figura 2.3.2 Muro de acumulación

- **Muro de acumulación no ventilado:** Es un muro construido con piedra, ladrillos, hormigón o incluso agua, pintado de negro o color muy oscuro por la cara exterior. Para mejorar la captación se aprovecha una propiedad del vidrio que es generar efecto invernadero, por el cual la luz visible ingresa y al tocar el muro lo calienta, emitiendo radiación infrarroja, la cual no puede atravesar el vidrio.

Por este motivo se eleva la temperatura de la superficie oscura y de la cámara de aire existente entre el muro y el vidrio.

- **Captación solar y acumulación calor:** es un sistema más complejo y permite combinar la *ganancia directa por ventanas* con colectores solares de aire o agua caliente para acumularlo debajo del piso. Luego, de modo similar al muro acumulador ventilado, se lleva el calor al ambiente interior. Adecuadamente dimensionado permite acumular calor para más de siete días.

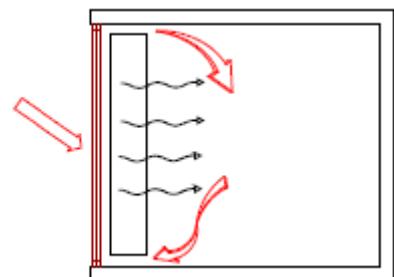


Figura 2.3.3 Pared de captación solar.

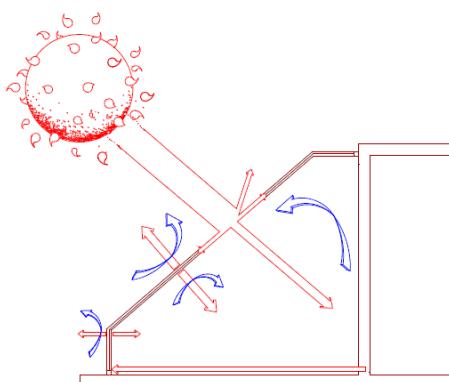


Figura 2.3.4 Invernadero.

- **Invernadero adosado:** en este caso al muro que da al mediodía se le incorpora un espacio vidriado, que puede ser habitable, mejorando la captación de calor durante el día, reduciendo las pérdidas de calor hacia al exterior.

- **Muro de acumulación ventilado o muro trombe:** es un muro o pared orientada al sol, construida con materiales que puedan acumular calor bajo el efecto de masa térmica (tales como piedra, hormigón o agua), combinado con un espacio de aire, una lámina de vidrio y ventilaciones formando un colector solar térmico. Su funcionamiento se basa en la diferencia de densidad del aire caliente y el aire frío, que provoca corrientes en una u otra dirección dependiendo de las trampillas que estén abiertas.

Estas corrientes de aire caliente o templado calientan o refrescan introduciendo o extrayendo el aire caliente. En la figura 2.3.5 podemos ver como circulan las corrientes de aire frío (flechas azules) y aire caliente (flechas rojas), dependiendo de la posición de las trampillas que hay en el muro del edificio.

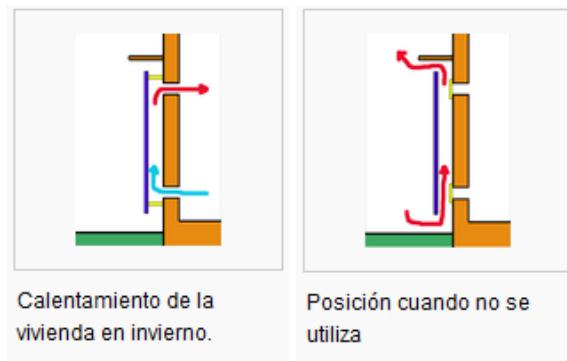


Figura 2.3.5 Muro de acumulación ventilado.

Durante el día la radiación solar incide en la lámina de vidrio calentando la superficie oscura y almacenando el calor. Por la noche, el calor se escapa del muro tendiendo a enfriarse principalmente hacia el exterior, pero como se encuentra con la lámina de vidrio el calor es devuelto al interior.

La elección de estas actuaciones para realizar una edificación o una rehabilitación se realizan dependiendo del confort y del clima. Normalmente cada tipo de clima tiene una serie de actuaciones más adecuadas. Aunque las condiciones microclimáticas hará que las actuaciones puedan cambiar o ser modificadas. El presente proyecto pretende dar los datos necesarios para poder seleccionar un clima y/o un microclima de la zona para luego decidir que tipo de actuaciones bioclimáticas se pueden realizar sobre una edificación.

2.3.2 Energía y arquitectura

La eficiencia energética es un compromiso en la arquitectura bioclimática. Los arquitectos utilizan diversas técnicas para reducir las necesidades energéticas de edificios mediante el ahorro de energía y para aumentar su capacidad de capturar la energía del sol o de generar su propia energía. Entre estas estrategias de diseño sostenible se encuentran la calefacción solar activa y pasiva, el calentamiento solar de agua (activo o pasivo), la generación eléctrica solar, la acumulación freática o la calefacción geotérmica, y más recientemente la incorporación en los edificios de generadores eólicos.

2.3.2.1 Climatización pasiva

En climas muy cálidos donde es necesario el refrescamiento, el diseño solar pasivo proporciona soluciones eficaces, como por ejemplo, una chimenea solar. Consiste en una chimenea pintada de negro. Durante el día la energía solar calienta la chimenea y el aire dentro de ella, creando una corriente de aire ascendente en la chimenea. La succión creada en la base de la chimenea se puede utilizar para ventilar y para refrescar el edificio.

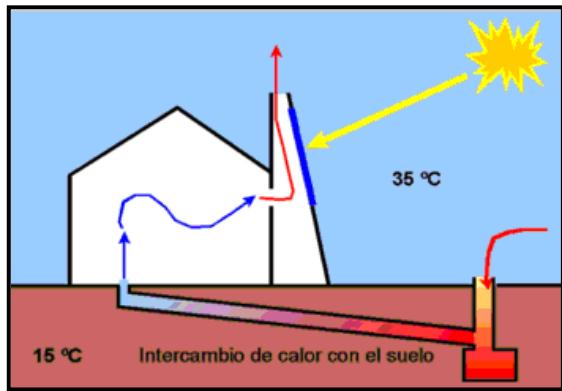


Figura 2.3.6 Chimenea solar.

Hay sin embargo un número de variaciones solares de la chimenea. Los elementos básicos del diseño de una chimenea solar son:

- El área del colector solar: esta superficie se puede situar en la parte superior de la chimenea o puede incluir el eje del tiro entero.
- La orientación, el tipo de pintura, el aislamiento y las características térmicas de este elemento son cruciales para captar, conservar y utilizar la energía solar.
- El eje principal de la ventilación: la localización, la altura, la sección representativa y las características térmicas de esta estructura son también muy importantes.
- Los orificios de entrada y salida: las dimensiones, localización así como aspectos aerodinámicos de estos elementos son también significativos en el rendimiento.

Cuando por condiciones particulares sea imposible el uso del refrescamiento pasivo, como por ejemplo, edificios en sectores urbanos muy densos, será necesario utilizar fuertes y activas estrategias de diseño sostenible. Entre otras:

- Adecuada protección solar en todas las superficies vidriadas.
- Evitar el uso de vidriados en techos.
- Buen aislamiento térmico en muros, techos y vidriados.
- Concentrar los espacios de gran emisión de calor (ejemplo: computadoras, cocinas, etc.) y darles buena ventilación.
- Sectorizar los espacios según usos.

- Utilizar sistemas de aire acondicionado con certificación energética a fin de conocer cuán eficientes son.
- Ventilar los edificios durante la noche.

Las edificaciones que utilizan la calefacción pasiva incorporan la inercia térmica mediante el uso de materiales de construcción que permiten la acumulación de calor en su masa térmica. Además es necesario utilizar aislamiento térmico para conservar el calor acumulado durante un día soleado. Además, para minimizar la pérdida de calor se busca que los edificios sean compactos y se logra mediante una superficie de muros, techos y ventanas bajas respecto del volumen que contienen. Esto significa que los diseños muy abiertos de múltiples alas o con forma de espina deben ser evitados eligiendo estructuras más compactas y centralizadas.

2.3.2.2 Producción de energías alternativas en edificios

Las energías alternativas en la arquitectura implican el uso de dispositivos, tales como paneles fotovoltaicos o generadores eólicos que ayudan a proporcionar electricidad para cualquier uso. Si los techos tienen pendientes hay que tratar de ubicarlas hacia el mediodía solar con una pendiente tal que optimice la captación de la energía solar a fin que los paneles fotovoltaicos generen con la eficacia máxima.

Se han construido edificios que incluso se mueven a través del día para seguir al sol. Los generadores eólicos se están utilizando cada vez más en zonas donde la velocidad del viento es suficiente con tamaños menores a 8m de diámetro. Los sistemas de calefacción solar activos mediante agua cubren total o parcialmente las necesidades de calefacción a lo largo del año de una manera sustentable. Los edificios que utilizan una combinación de estos métodos alcanzan la meta más alta que consiste en una demanda de energía cero.

Otras formas de generación de energía basadas en fuentes renovables son la energía solar térmica (para calefacción, agua caliente sanitaria y aire acondicionado), biomasa o incluso la geotérmica. Lo ideal para garantizar el suministro energético durante todo el año, bajo condiciones climáticas y ambientales cambiantes, es combinar las diferentes fuentes.

Para poder hacer las diferentes actuaciones sobre una edificación en relación a la energía utilizada para la climatización de una arquitectura, el sistema implementado dispone de una serie de sensores para medir la radiación solar y así saber la potencia que va a recibir la estructura, además se podrá determinar si sobre la zona y por consiguiente la

arquitectura va a recibir sombras de elementos físicos cercanos y a que hora, para poder determinar si son convenientes, y en que partes, las actuaciones bioclimáticas. Además por la misma razón se decide la ubicación de elementos de generación de energías alternativas para compensar el consumo de la arquitectura diseñada.

2.3.3 Comentarios sobre la arquitectura bioclimática

La arquitectura bioclimática es una concepción de diseño de las edificaciones que busca el confort y el ahorro energético. Este tipo de edificaciones pueden llegar a suponer un ahorro energético superior al 50% de la energía consumida, por lo que su implantación debe ser una constante a la hora de hacer nuevas edificaciones o de rehabilitar las ya existentes.

Como se ha visto el clima del emplazamiento es muy importante a la hora de este tipo de diseño, puesto que el clima influirá en gran medida en las actuaciones que se pueden llevar a cabo en una edificación. Pero no sólo hay que tener en cuenta el clima de la región puesto que como ya se ha visto el microclima de una zona influye sobremanera sobre las posibles actuaciones que se pueden realizar.

Por ello las variables climáticas más importantes a la hora de definir el tipo de microclima de una zona serán: la radiación solar, la temperatura, la humedad, el viento, las precipitaciones, la radiación de onda larga, la presión atmosférica, siendo éste el orden de importancia.

Para la elaboración de éste proyecto se ha decidido que las variables a captar por parte del sistema sean las más importantes quedando excluidas las precipitaciones, por no ser una variable microclimática sino una variable de una región y la radiación de onda larga y la presión atmosférica por ser las variables de menor importancia.

2.3.4 Ejemplos de Arquitecturas bioclimática



Figura 2.3.7 Diferentes arquitecturas bioclimáticas.

Capítulo 3

Herramienta hardware

En este capítulo se pretende describir la herramienta hardware realizada para la consecución del objetivo principal del proyecto, que es la obtención de los datos climáticos más importantes. En primer lugar se necesita un microcontrolador para poder actuar sobre los diferentes elementos que vamos a integrar en nuestro sistema. Para ello, se ha elegido el microcontrolador Atmega328 porque forma parte de una plataforma de hardware libre llamado Arduino, dicha plataforma dispone de un entorno de programación basado en el lenguaje de programación C/C++.

Además tenemos que disponer de diferentes sensores para la medición de las diferentes variables climáticas que se quieren registrar. Como la radiación solar directa y global, la temperatura, la humedad, la velocidad y dirección del viento y el mapa de sombras.

El microcontrolador y los sensores serán el núcleo más importante ha desarrollar, pero para la obtención de un sistema completo y eficiente también se tienen que tener en cuenta otras consideraciones; disponer de memoria suficiente para poder almacenar los datos obtenidos, acceso a los datos sin ir físicamente al emplazamiento del sistema, para ello se ha decidido por implantar un modulo GPRS, también se precisa un calendario para saber fecha y hora a la que se realizan las medidas, por ultimo saber el consumo del sistema y como se alimenta. Se puede observar el esquema en la figura 3.1.1.

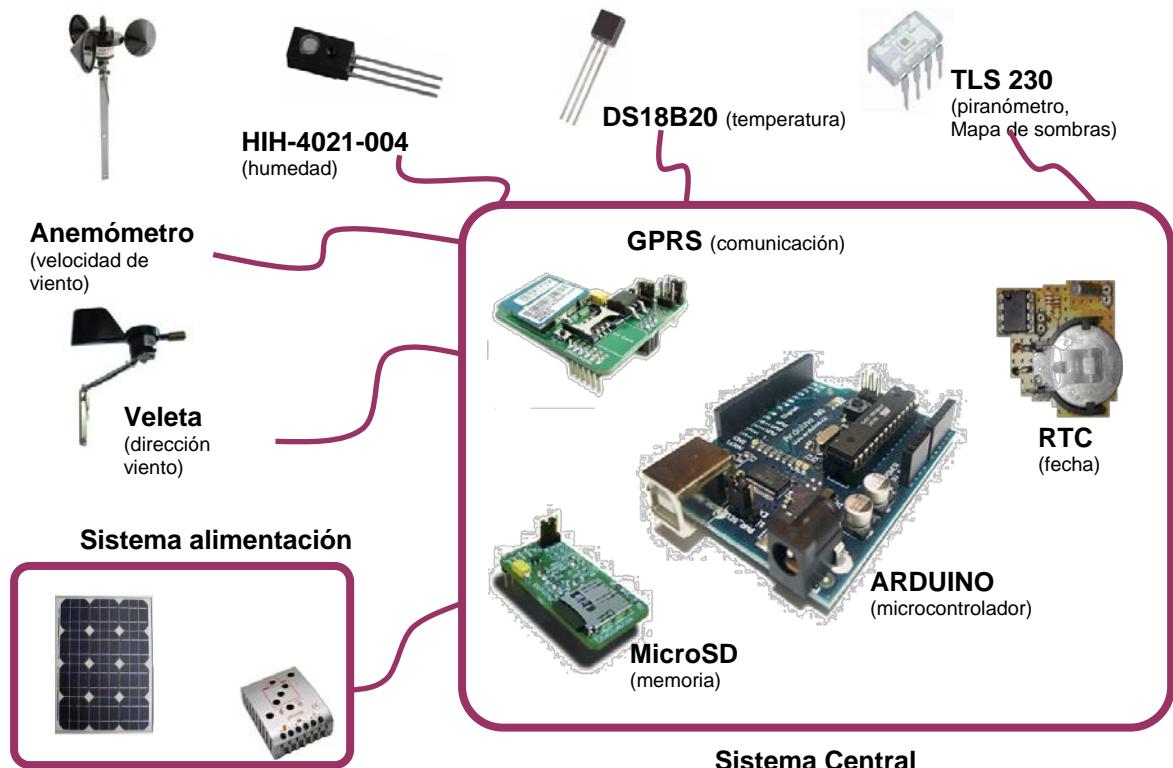


Figura 3.1.1 Esquema herramienta hardware.

3.1 Arduino

Arduino es una plataforma de hardware libre basada en una sencilla placa de entradas y salidas y un entorno de desarrollo que implementa el lenguaje de programación Processing/Wiring. Arduino se puede utilizar para desarrollar objetos interactivos autónomos o puede ser conectado al ordenador.

Las plataformas Arduino están basadas en los microcontroladores Atmega168, Atmega328 y otros similares, chips sencillos y de bajo coste que permiten el desarrollo de múltiples diseños. Al ser open-hardware, tanto su diseño como su distribución son libres. Es decir, puede utilizarse libremente para el desarrollo de cualquier tipo de proyecto sin haber adquirido ninguna licencia. La mayor parte de la información que a continuación se detalla ha sido extraída de [7].

Arduino Duemilanove

La placa Arduino que se va a utilizar va a ser Arduino duemilanove. La placa Duemilanove (“2009”). Es una placa con un microcontrolador basada en el ATmega168 o ATmega328. En este caso se ha elegido el ATmega328 por tener unas prestaciones superiores. El Duemilanove es el más popular en dentro de las series de placas con USB.

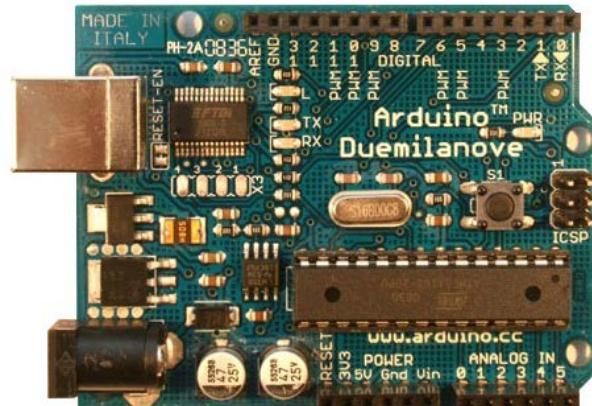


Figura 3.1.2 Arduino Duemilanove.

Tiene 14 pines con entradas/salidas digitales (6 de las cuales pueden ser usadas como salidas PWM), 6 entradas analógicas, un cristal oscilador a 16Mhz, conexión USB, entrada de alimentación, una cabecera ISCP, y un botón de reset. Contiene todo lo necesario para utilizar el microcontrolador; simplemente conectándolo al ordenador a través del USB o alimentándolo con un transformador o una batería.

Resumen

Microcontrolador	ATmega328 (ATmega168 en versiones anteriores)
Voltaje de funcionamiento	5V
Voltaje de entrada (recomendado)	7-12V
Voltaje de entrada (límite)	6-20V
Pines E/S digitales	14 (6 proporcionan salida PWM)
Pines de entrada analógica	6
Intensidad por pin	40 mA
Intensidad en pin 3.3V	50 mA
Memoria Flash	32 kB (ATmega328) de las cuales 2 kB las usa el gestor de arranque (bootloader)
SRAM	2 kB (ATmega328)
EEPROM	1 kB (ATmega328)
Velocidad de reloj	16 MHz

Tabla 3.1.1 Resumen de las características del Arduino duemilanove.

3.1.1 Esquema y pines

Elementos con los que podemos interactuar: (tomando como ejemplo la placa USB). Empezando en el sentido de las agujas del reloj desde el centro de la parte superior.

Esquema de pines:

- Pin de referencia analógica (naranja).
- Señal de tierra digital (verde claro).
- Pines digitales 3-13 (verde).
- Pines digitales 1 y 2. Entrada y salida del puerto serie. Transmisión y recepción (azul). (estándar de comunicación serie IC2).
- Botón de reset (negro).
- Entrada del circuito del programador serie (marrón).
- Pines de entrada analógica 0-5 (azul oscuro).
- Pines de alimentación y tierra (naranja y naranja claro).
- Entrada de la fuente de alimentación externa (9-12V DC) – X1 (gris).
- Comutación entre fuente de alimentación externa o alimentación a través del puerto USB – SV1.
- Puerto USB (rojo).

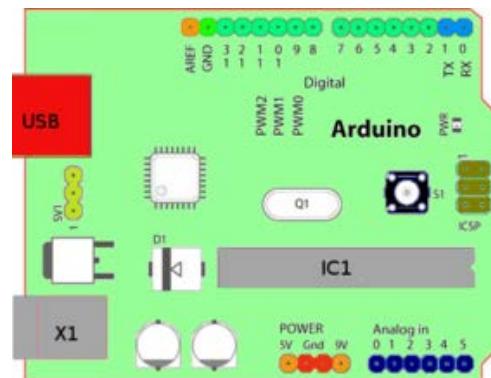


Figura 3.1.3 Esquema de componentes del Arduino Duemilanove.

3.1.1.1 Alimentación

El Arduino Duemilanove puede ser alimentado vía conexión USB o con una fuente de alimentación externa. El origen de la alimentación se selecciona automáticamente.

Las fuentes de alimentación externas (no-USB) pueden ser tanto un transformador o una batería. El transformador se puede conectar usando un conector macho de 2.1mm con centro positivo en el conector hembra de la placa. Los cables de la batería pueden conectarse a los pines Gnd y 639.

5Vin en los conectores de alimentación (POWER).

La placa puede trabajar con una alimentación externa de entre 6 a 20 voltios. Si el voltaje suministrado es inferior a 7V el pin de 5V puede proporcionar menos de 5 Voltios y la placa puede volverse inestable; si se usan más de 12V los reguladores de voltaje se pueden sobrecalentar y dañar la placa. El rango recomendado es de 7 a 12 voltios.

Los pines de alimentación son los siguientes:

VIN: La entrada de voltaje a la placa Arduino cuando se está usando una fuente externa de alimentación (en opuesto a los 5 voltios de la conexión USB). Se puede proporcionar voltaje a través de este pin, o, si se está alimentado a través de la conexión de 2.1mm, acceder a ella a través de este pin.

5V: La fuente de voltaje estabilizado usado para alimentar el microcontrolador y otros componentes de la placa. Esta puede provenir de VIN a través de un regulador integrado en la placa, o proporcionada directamente por el USB u otra fuente estabilizada de 5V.

3V3: Una fuente de voltaje a 3.3 voltios generada en el chip FTDI integrado en la placa. La corriente máxima soportada 50mA.

GND: Pines de toma de tierra.

3.1.1.2 Pines Digitales

Los pines del Arduino pueden configurarse como entradas o salidas. Este apartado explica el funcionamiento de los pines en esos modos. Si bien el título de este apartado se refiere a los pines digitales, es importante señalar que la gran mayoría de los pines analógicos del Arduino, pueden configurarse y utilizarse, exactamente de la misma manera que los pines digitales.

Propiedades de los Pines Configurados como Entrada (*INPUT*)

Los pines del Arduino por defecto son de entrada, por lo que no es necesario configurarlos explícitamente como entradas con *pinMode()*. Los pines configurados como entradas están en estado de alta impedancia. Esto significa que se necesita muy poca corriente para pasar el pin de entrada de un estado a otro, y puede hacer posible el uso de los pines para tareas como la utilización de un sensor capacitivo al tacto, la lectura de un LED como un fotodiodo, o la lectura de un sensor analógico con un esquema como el RCTime.

Esto también significa sin embargo, que los terminales de entrada sin conectar nada a ellos, o con los cables conectados a ellos sin estar conectados a otros circuitos, reflejarán cambios aparentemente aleatorios en el estado de pin, recogiendo el ruido eléctrico del entorno, o el acoplamiento capacitivo del estado de un pin próximo.

Resistencias Pullup

A menudo es útil para colocar un pin de entrada en un estado conocido si no hay un estado de entrada. Puede hacerse añadiendo una resistencia pull-up (a +5 V), o una

resistencia pull-down (resistencia a tierra) en la entrada; $10k\Omega$ suele ser un valor muy común.

Además el chip ATmega dispone de resistencias pull-up de $20k\Omega$ integradas en el chip a las que se puede acceder desde el software. Estas resistencias pull-up incorporadas son accedidas de la siguiente manera.

```
pinMode(pin, INPUT);      // pone el pin como entrada
digitalWrite(pin, HIGH); // activa la resistencia pull-up
```

Hay que tener en cuenta que las resistencias pull-up proporcionan suficiente corriente para dar luz tenue con un LED conectado a un pin que se ha configurado como entrada. Por lo que si vemos que un pin hace que un LED funcione pero con una luz muy tenue, es posible, que no se haya ajustado los pines como salida con el comando *pinMode()*.

Las resistencias pull-up son controladas por los mismos registros (posiciones de memoria interna del chip) que controlan si un pin está alto (HIGH) o bajo (LOW). Por consiguiente, un pin que se configura para tener las resistencias pull-up activadas cuando está configurado como entrada, debe tener el pin a alto (HIGH) si el pin es cambiado como salida (OUTPUT) con *pinMode()*. Esto funciona en la otra dirección también, y un pin de salida que queda en un estado alto tendrá las resistencias pull-up activas, si cambia a entrada (INPUT) con *pinMode()*.

NOTA. *El pin Digital 13 es más difícil de usar que otros pines digitales porque tiene un LED y una resistencia asociada soldados a la placa en la mayoría de las placas. Si activa la resistencia pull-up 20k del interior, se pondrá en alrededor de 1,7 V en lugar de los 5V que se esperan debido a que el LED integrado y la resistencia en serie bajan el nivel del voltaje, lo que se traduce en que siempre retornará bajo (LOW). Si estás obligado a usar el pin 13 como entrada digital, utiliza una resistencia pull-down externa.*

Propiedades de los Pines Configurados como salida (OUTPUT)

Los pines configurados como salida (OUTPUT) con *pinMode()* se dice que están en un estado de baja impedancia. Esto significa que puede proporcionar una cantidad sustancial de corriente a otros circuitos, de hasta 40mA. Esto es corriente suficiente para utilizar leds o para la mayoría de sensores, pero no es suficiente para alimentar la mayoría de relés o motores. Los cortocircuitos o intentar extraer mucha corriente puede dañar los transistores de salida o incluso dañar el chip ATMega. Por esta razón es buena idea conectar los pines de salida a otros dispositivos con resistencias de 1k, limitando la corriente máxima que los pines del Arduino pueden dar.

3.1.1.3 Pins de Entrada Analógica

Descripción de las entradas analógicas del Arduino.

Conversor A/D

El controlador ATmega que usa Arduino lleva incluido un conversor analógico-digital (A/D) de 6 canales con una resolución de 10 bits. El uso principal de estos pines es para la lectura de sensores analógicos, pero tienen también toda la funcionalidad de los pines de entrada-salida de propósito general (GPIO) (al igual que los pines 0 - 13). Por lo que, si un usuario necesita más pines de propósito general de entrada-salida, y no se está usando ningún pin analógico, estos pines pueden usarse como GPIO.

Mapeo de Pins

Los pines analógicos corresponden con los pines del 14 al 19 del Arduino. Los pines analógicos, pueden usarse de manera idéntica que los digitales, así que por ejemplo, podrías ver un código como el que se muestra a continuación para configurar un pin analógico, y establecerlo a HIGH:

```
pinMode(14, OUTPUT);
digitalWrite(14, HIGH);
```

Resistencias Pull-up

Los pines analógicos también tienen resistencias pull-up, las cuales funcionan igual que en los pines digitales. Se activan cuando usamos instrucciones como la siguiente:

```
digitalWrite(14, HIGH); // activa la resistencia pull-up en el pin analógico 0 Mientras el pin es una entrada (input).
```

Hay que tener en cuenta que si activamos una resistencia pull-up, mientras usamos algunos sensores, los valores obtenidos por *analogRead()* se verán afectados.

Detalles y Observaciones

El comando *analogRead* no funcionará adecuadamente si el pin analógico se estableció como salida. Habrá que establecerlo como entrada antes de usar *analogRead*. De igual manera si el pin, siendo de salida, se estableció a HIGH, la resistencia pull-up permanecerá activa, aunque volvamos a establecerla como entrada.

La hoja de datos de ATmega nos previene acerca de la conmutación (de analógico a digital) de los pines analógicos en las proximidades de una lectura (*analogRead*) en otros pines analógicos. Esto puede producir ruidos y fluctuaciones en el sistema analógico. Puede ser útil, después de manipular pines analógicos (en modo digital), añadir una pequeña parada (*delay*) antes de usar *analogRead()* en otros pines analógicos.

3.1.1.4 PWM

La modulación por ancho de pulsos (también conocida como PWM, siglas en inglés de pulse-width modulation) de una señal o fuente de energía es una técnica en la que se modifica el ciclo de trabajo de una señal periódica (una senoidal o una cuadrada, por ejemplo), ya sea para transmitir información a través de un canal de comunicaciones o para controlar la cantidad de energía que se envía a una carga.

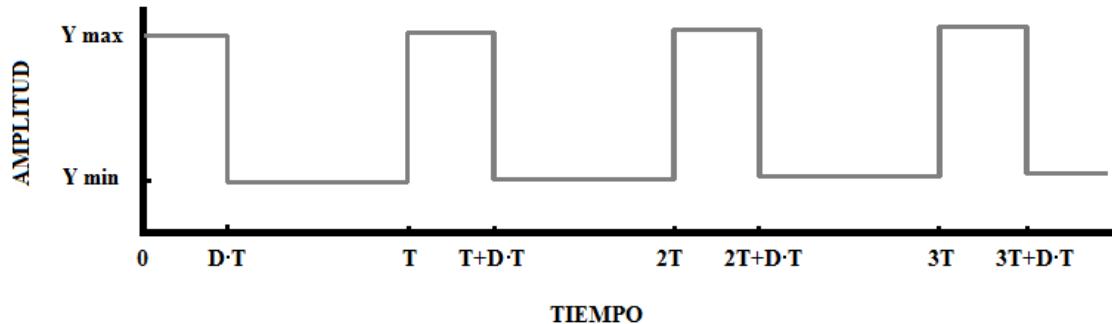


Figura 3.1.4 Señal de onda cuadrada de amplitud acotada, mostrando el ciclo de trabajo D.

El ciclo de trabajo de una señal periódica es el ancho relativo de su parte positiva en relación con el período. Expresado matemáticamente:

$$D = \frac{\tau}{T}$$

D es el ciclo de trabajo

τ es el tiempo en que la función es positiva (ancho del pulso)

T es el período de la función.

En este caso la técnica PWM se utiliza para simular una salida analógica con una salida digital. El control digital se usa para crear una onda cuadrada, una señal que conmuta constantemente entre encendido y apagado. Este patrón de encendido-apagado puede simular voltajes entre 0 (siempre apagado) y 5 voltios (siempre encendido) simplemente variando la proporción de tiempo entre encendido y apagado. Al tiempo de encendido (ON) se le llama Ancho de Pulso (pulse width). Para variar el valor analógico cambiamos, o modulamos, ese ancho de pulso. Si se repite el patrón de encendido-apagado el resultado es como si se tuviera una señal que puede variar entre 0 y 5 voltios.

En la figura 3.1.4 se puede observar como el período de la señal representado por la distancia entre las marcas verdes. El período es un período regular, esta duración o período es la inversa de la frecuencia del PWM. En el Arduino la frecuencia PWM es bastante próxima a 500Hz lo que equivale a períodos de 2 milisegundos cada uno.

La llamada a la función `analogWrite()` debe ser en la escala desde 0 a 255, siendo 255 el 100% de ciclo (siempre encendido), el valor 127 será el 50% del ciclo (la mitad del tiempo encendido), etc.

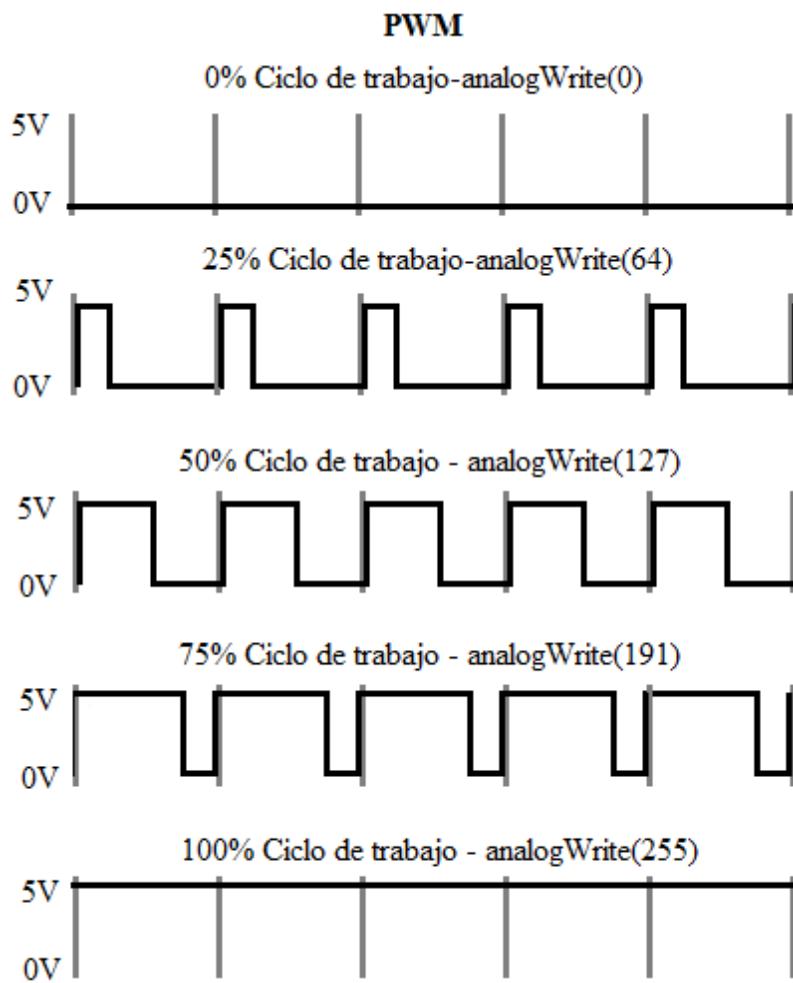


Figura 3.1.5 Relaciones entre el ancho del pulso y el valor analógico con la técnica PWM.

3.1.2 Memoria

Existen 3 fuentes de memoria en el microcontrolador utilizado por la placa Arduino (ATmega328):

- Memoria Flash (espacio del programa), donde Arduino almacena el sketch.
- SRAM (static random access memory, memoria estática de acceso aleatorio) donde los sketches almacena y manipulan variables al ejecutarse.
- EEPROM es un espacio de memoria que puede ser utilizada por los programadores para almacenar información de largo plazo.

La memoria Flash y EEPROM son no-volátiles (la información se mantiene en ellas luego de cortar la alimentación). SRAM es volátil y se perderá al reiniciar la unidad.

El chip ATmega328 cuenta con las siguientes cantidades de memoria:

- Flash 32kB (de los que 2k son utilizados por el bootloader).
- SRAM 2048 bytes.
- EEPROM 1024 bytes.

Habrá que tenerse en cuenta que la memoria SRAM disponible no es demasiado grande. Es relativamente fácil utilizarla toda si el programa cargado tiene muchas cadenas (strings) o muchas variables declaradas.

Si se utiliza toda la SRAM, el programa fallará de manera imprevista; el programa se cargará en la placa de manera correcta, pero no se ejecutara, o se ejecutara de manera extraña. Existen formas de solucionar este problema:

- Trasladar datos al ordenador, si el Arduino se comunica con una aplicación corriendo en uno.
- Si tienes tablas de referencia o gran tamaño, utiliza el tipo de datos más pequeño que puedas para almacenar estos datos.
- Si no necesitas modificar las cadenas o datos mientras el sketch se ejecuta, puedes almacenarlos en la memoria Flash (de programa) en vez de la SRAM.

3.1.3 Programación

Los programas hechos con Arduino se dividen en tres partes principales: *estructura*, *valores* (variables y constantes), y *funciones*. El Lenguaje de programación Arduino se basa en C/C++.

El ATmega328 en las placas Arduino Duemilanove viene precargado con un gestor de arranque (bootloader) que permite cargar un nuevo código sin necesidad de un programador por hardware externo. Se comunica utilizando el Protocolo SKT500 original (referencia, archivo de cabecera C). También te puedes saltar el gestor de arranque y programar directamente el microcontrolador a través del puerto ISCP (In Circuit Serial Programming);

Variables

Una variable es una ubicación para almacenar una porción de información. Tiene un nombre, un valor y un tipo.

Por ejemplo, esta sentencia o declaración: `int pin = 13;` crea una variable cuyo nombre es `pin`, cuyo valor es `13`, y su tipo es `int`. Más tarde en el programa, puedes hacer referencia a esta variable por su nombre, momento en el que puedes acceder a su valor y utilizarlo.

Funciones

Segmentar el código en funciones permite crear piezas modulares de código que realizan una tarea definida y vuelven a la zona del programa en la que fueron llamadas. La razón principal para crear una función es cuando se necesita realizar la misma acción múltiples veces dentro de un mismo programa.

La estandarización de fragmentos de código en funciones tiene diversas ventajas:

- Las funciones ayudan al programador a ser organizado. Además ayudan a conceptualizar el programa.
- Las funciones codifican una acción en un lugar, así que sólo deben ser depuradas de errores una vez.
- Reducen las posibilidades de error en modificaciones, si el código debe ser cambiado.
- Las funciones hacen el sketch más pequeño y más compacto porque las secciones de código se reutilizan varias veces.
- Hacén más fácil la reutilización de código en otros programas por hacerlo más modular y, como efecto paralelo, usando funciones se obtiene un código más legible.

Hay dos funciones necesarias en un sketch de Arduino: *setup()* y *loop()*. El resto de funciones debe ser definido fuera de las llaves de estas dos funciones.

Referencia de lenguaje

Estructura	Variables	Funciones
<ul style="list-style-type: none"> • <u>setup()</u> (<i>inicialización</i>) • <u>loop()</u> (<i>bucle</i>) 	<p>Constantes</p> <ul style="list-style-type: none"> • <u>HIGH LOW</u> • <u>INPUT OUTPUT</u> • <u>true false</u> • <u>Constantes Numéricas</u> <p>Tipos de Datos</p> <ul style="list-style-type: none"> • <u>boolean (booleano)</u> • <u>char (carácter)</u> • <u>byte</u> • <u>int (entero)</u> • <u>unsigned int (entero sin signo)</u> • <u>long (entero 32b)</u> • <u>unsigned long (entero 32b sin signo)</u> • <u>float (en coma flotante)</u> • <u>double (en coma flotante de 32b)</u> 	<p>E/S Digitales</p> <ul style="list-style-type: none"> • <u>pinMode()</u> • <u>digitalWrite()</u> • <u>digitalRead()</u> <p>E/S Analógicas</p> <ul style="list-style-type: none"> • <u>analogRead()</u> • <u>analogWrite() PWM (modulación por ancho de pulso)</u> <p>E/S Avanzadas</p> <ul style="list-style-type: none"> • <u>tone()</u> • <u>noTone()</u> • <u>shiftOut()</u> • <u>pulseIn()</u>
<p>Estructuras de control</p> <ul style="list-style-type: none"> • <u>if (comparador si-entonces)</u> • <u>if...else (comparador si...sino)</u> • <u>for (bucle con contador)</u> • <u>switch case (comparador múltiple)</u> • <u>while (bucle por comparación booleana)</u> • <u>do... while (bucle por comparación booleana)</u> • <u>break (salida de bloque de código)</u> • <u>continue (continuación en bloque de código)</u> • <u>return (devuelve valor a programa)</u> 		
<p>Sintaxis</p> <ul style="list-style-type: none"> • <u>;</u> (<i>punto y coma</i>) 		

- { } (llaves)
- // (comentarios en una línea)
- /* */ (comentarios en múltiples líneas)

Operadores Aritméticos

- = (asignación)
- + (suma)
- - (resta)
- * (multiplicación)
- / (división)
- % (resto)

Operadores Comparativos

- == (igual a)
- != (distinto de)
- < (menor que)
- > (mayor que)
- <= (menor o igual que)
- >= (mayor o igual que)
- Operadores Booleanos
- && (y)
- || (o)
- ! (negación)

Operadores de Composición

- ++ (incrementa)
- -- (decrementa)
- += (composición suma)
- -= (composición resta)
- *= (composición multiplicación)
- /= (composición división)

- string (cadena de caracteres)
- array (cadena)
- void (vacío)

Conversión

- char()
- byte()
- int()
- long()
- float()

Tiempo

- millis()
- micros()
- delay()
- delayMicroseconds()

Matemáticas

- min() (mínimo)
- max() (máximo)
- abs() (valor absoluto)
- constrain() (limita)
- map() (cambia valor de rango)
- pow() (eleva a un número)
- sq() (eleva al cuadrado)
- sqrt() (raíz cuadrada)

Trigonometría

- sin() (seno)
- cos() (coseno)
- tan() (tangente)

Números Aleatorios

- randomSeed()
- random()

Comunicación

- Serial

3.1.4 Comunicaciones

EL Arduino Duemilanove facilita en varios aspectos la comunicación con el ordenador, otro Arduino u otros microcontroladores. El ATmega328 proporciona comunicación vía serie UART TTL (5V), disponible a través de los pines digitales 0(RX) y 1(TX). Un chip FTDI FT232RL integrado en la placa canaliza esta comunicación serie a través del USB y los drivers FTDI (incluidos en el software de Arduino) proporcionan un puerto serie virtual en el ordenador. El software incluye un monitor de puerto serie que permite enviar y recibir información textual de la placa Arduino. Los LEDS RX y TX de la placa parpadearan cuando se detecte comunicación transmitida través del chip FTDI y la conexión USB (no parpadearan si se usa la comunicación serie a través de los pines 0 y 1).

La Librería SoftwareSerial permite comunicación serie por cualquier par de pines digitales del Duemilanove. El ATMega 328 también soporta la comunicación I²C (TWI) y SPI.

3.1.5 Reinicio Automático (Software)

El Arduino Duemilanove está diseñado de manera que es posible reiniciar por software desde el ordenador donde esté conectado, en vez de tener que reiniciar físicamente mediante el botón de reset antes de cargar. Una de las líneas de control de flujo (DTR) del FT232RL está conectada a la línea de reinicio del ATmega328 a través de un condensador de 100nF. Cuando se pone a LOW (0V), la línea de reinicio también se pone a LOW el tiempo suficiente para reiniciar el chip. El software de Arduino utiliza esta característica para permitir cargar los sketches con sólo apretar un botón del entorno. Dado que el gestor de arranque tiene un lapso de tiempo para ello, la activación del DTR y la carga del sketch se coordinan perfectamente.

Esta configuración tiene otras implicaciones. Cuando el Duemilanove se conecta a un ordenador con Mac OS X o Linux, esto reinicia la placa cada vez que se realiza una conexión desde el software (vía USB). Medio segundo después, se ejecuta el gestor de arranque.

El Duemilanove contiene una pista que puede ser cortada para deshabilitar el auto-reset. Las terminaciones a cada lado pueden ser soldadas entre ellas para rehabilitarlo. Están etiquetadas con “RESET-EN”. También podéis deshabilitar el auto-reset conectando una resistencia de 110Ω desde el pin 5V al pin de reset.

3.1.6 Protección contra sobretensiones en USB

El Arduino Duemilanove tiene un multifusible reinicializable que protege la conexión USB de tu ordenador de cortocircuitos y sobretensiones. La mayoría de ordenadores proporcionan su propia protección interna, el fusible proporciona una protección extra. Si se detectan más de 500mA en el puerto USB, el fusible corta la conexión hasta que el cortocircuito o la sobretensión desaparezcan.

3.1.7 Características Físicas

La longitud y amplitud máxima de la placa Duemilanove es de 6,9 y 5,3 cm respectivamente, con el conector USB y la conexión de alimentación sobresaliendo de estas dimensiones. Tres agujeros para fijación con tornillos permiten colocar la placa en superficies y cajas.

3.1.8 ATmega328

Información extraída de [8].

3.1.8.1 Características

- Microcontrolador de 8 bit AVR de alto rendimiento y bajo consumo
- Arquitectura RISC Avanzada:
 - 131 Instrucciones.
 - Reloj Individual de ejecución del ciclo.
 - Registros de Propósito General 32 x 8.
 - Funcionamiento totalmente estático.
 - Rendimiento Hasta 20 MIPS a 20 MHz.
 - Multiplicador de 2-ciclos en chip.
- Segmentos de memoria no-volátiles de larga duración.
 - 32 kBytes de memoria Flash Auto-programable en el Sistema.
 - 1 kByte de EEPROM.
 - 2 kBytes de SRAM Interna.
 - Ciclo de vida de Escritura /borrado: 10.000 Flash/100, 000 EEPROM.
 - Retención de datos: 20 años a 85 °C/100 años a 25 °C.
 - Código Sección arranque opcional con cerradura independiente Bits.
 - Bloqueo de programación para seguridad software.
- Características periféricas
 - Dos temporizador / Contadores de 8-bit con modo contador y comparador independiente.
 - Un temporizador / contador de 16-bit de con modo contador, comparador y capturador independientes.
 - Reloj de tiempo con oscilador separado.
 - Seis canales de PWM.
 - 8 canales de 10-bit ADC en TQFP y QFN / MLF.
 - Medición de temperatura
 - 6-canal de 10-bit ADC en PDIP.
 - Medición de temperatura
 - Serial USART programable.
 - Interfaz serial SPI Maestro / Esclavo.
 - Interfaz serial 2-wire (compatible I2C Philips).
 - Programable Watchdog Timer con oscilador en chip separado.
 - Comparador analógico en chip.
 - Interrumpiciones y activación por pin.
- Características especiales del microcontrolador.
 - Detección de Reset de encendido y detección programable brown-out.
 - Oscilador calibrado interno.
 - Fuentes de interrupción externa e interna.
 - Seis modos de sleep: libre, ADC reducción de ruido, ahorro de energía, apagado, en espera y espera extendido.
- I/O.
 - 23 Líneas I/O Programables.
 - 28-pines PDIP, 32 pistas TQFP, 28-pad QFN / FML y 32 de pad-QFN / FML.

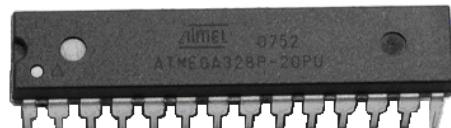


Figura 3.1.6 Chip ATmega 328

- Voltaje de funcionamiento:
 - 1,8 - 5,5 V.
- Rango de temperatura:
 - 40° C a 85° C.
- Velocidad:
 - 0 a 20 MHz @ 1.8 - 5.5V.
- Consumo a baja consumo a 1 MHz, 1,8 V, 25 ° C.
 - Modo activo: 0,2 mA.
 - Modo consumo apagado: 0,1 µA.
 - Modo de ahorro de energía: 0,75 µA (Por ejemplo 32 kHz RTC).

3.1.8.2 Configuración de pines

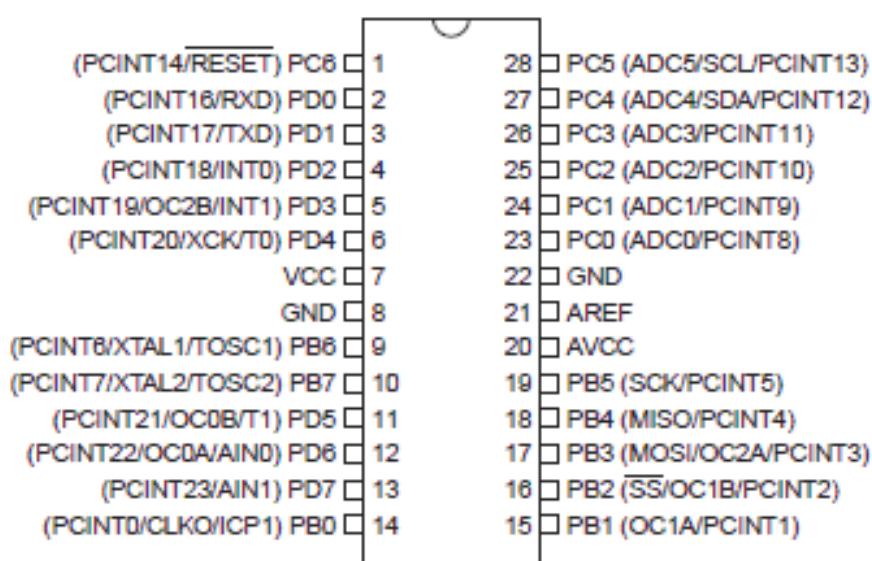


Figura 3.1.7 Configuración de pines.

Descripción

VCC: Tensión de alimentación.

GND: Tierra.

Puerto B (PB 0-7): Es un puerto de I/O de 8-bit con resistencias internas de pull-up (seleccionados por cada bit). Los Buffer de salida del puerto B tienen características simétricas con una gran capacidad y callback. Como entradas, los pines Puerto B que se pongan a tierra externamente, serán una fuente de corriente si las resistencias de pull-up están activas. Los pines del puerto B son tri-estado cuando una condición de reset se activa, incluso si el reloj no está funcionando.

Dependiendo de la configuración de fusibles del reloj, PB6 se puede utilizar como entrada amplificada de oscilador invertido y como entrada al circuito de

funcionamiento interno del reloj. Y PB7 se puede utilizar como salida amplificada del oscilador invertido.

Si el oscilador interno calibrado RC se utiliza como fuente de reloj de chip, PB 6-7 pueden usarse como TOSC 1-2 de entrada asíncrona para el reloj/contador2 si el bit en AS2 ASSR está seleccionado.

Puerto C (PC 0-5): Es un puerto de I/O de 7-bit con resistencias internas de pull-up (seleccionados por cada bit). Los Buffer de salida de los puertos C (PC 0-5) tienen características similares a los buffers de salida de los puertos B.

PC6/RESET: Si el fusible RSTDISBL está programado, PC6 se utiliza como un pin I/O. Tenga en cuenta que las características eléctricas de PC6 difieren de las de los demás pines de Puerto C.

Si el fusible RSTDISBL está programado, PC6 se utiliza como una entrada de reset. Un nivel bajo en este pin de mayor duración que el pulso mínimo generará un Reset, incluso si el reloj está parado.

Puerto D (PD 0-7): Es un puerto de I/O de 8-bit con resistencias internas de pull-up (seleccionados por cada bit). Los Buffer de salida del puerto B tienen características similares a los buffers de los puertos B y C.

AVCC: Es una fuente de alimentación para los convertidores A/D, PC 0-3 y ADC6-7. Debería estar conectado externamente a Vcc incluso si ADC no es usado. Si ADC es usado debería conectarse a Vcc través de un filtro de paso bajo. PC 4-6 se alimentan a través de la fuente de alimentación digital, Vcc.

AREF: Es el pin de referencia analógica para el convertidor A / D.

ADC 6-7 (TQFP y QFN / sólo paquete MLF): En el TQFP y QFN / paquete MLF, ADC 6-7 es usado como entradas analógicas para el convertidor A / D. Estos pines se alimentan desde la fuente analógica y sirven como canales de ADC de 10-bit.

3.1.8.3 Diagrama de bloques

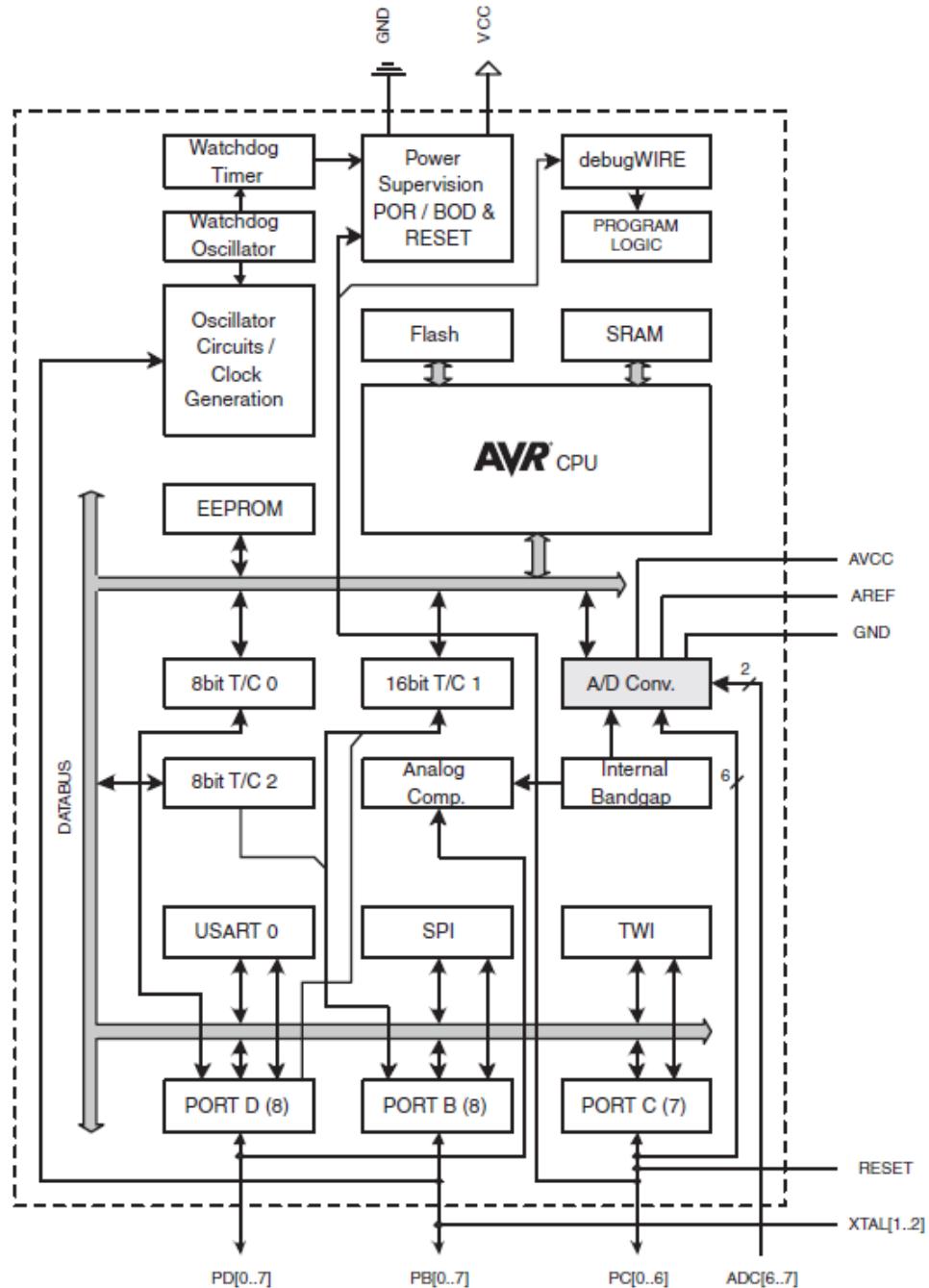


Figura 3.1.8 Diagrama de bloques ATmega328.

El núcleo AVR combina un amplio conjunto de instrucciones con 32 registros de propósito general. Todos los registros están conectados directamente a la Unidad aritmético lógica (ALU), permitiendo que dos registros independientes puedan acceder en una sola instrucción ejecutada en un ciclo de reloj. La arquitectura resultante es más eficiente, pudiendo alcanzar rendimientos de hasta diez veces más rápido que los Microcontroladores convencionales CISC.

Las características del ATmega328 son: 32kB de memoria Flash programable con capacidad de lectura-escritura, 1kB EEPROM, 2kB de SRAM, 23 líneas I/O y 32 registros

de propósito general, tres temporizador/Contadores flexibles con modo comparador, interrupciones internas y externas, un USART serial programable, un byte orientada al interfaz serial 2-wire, un puerto serie SPI, 6 canales 10-bit ADC, un temporizador de vigilancia programable con alimentación interna del oscilador, y cinco modos de ahorro seleccionable por software.

El modo de espera detiene la CPU a la vez que la SRAM, el USART, el contador/Temporizador, el interfaz serial 2-wire, puerto SPI, y el sistema de interrupción continúan funcionando. El modo Power-down guarda el contenido del registro, pero congela el oscilador, desactivando todas las funciones del chip hasta la siguiente interrupción o reinicio de hardware. En el modo de ahorro de energía, el temporizador asíncrono continúa funcionando, permitiendo al usuario mantener una temporización base mientras el resto del dispositivo está durmiendo. El modo de reducción de ruido ADC detiene el CPU y todos los módulos I/O, excepto temporizador asíncrono y ADC, para minimizar el ruido de conmutación en conversiones ADC. En modo de espera, el oscilador está funcionando mientras que el resto del dispositivo está durmiendo. Esto permite una recuperación muy rápida con un bajo consumo.

El dispositivo está fabricado con tecnología de alta densidad Atmel de memoria no volátil. El chip Flash ISP permite ser reprogramado a través de un interfaz serial SPI o por un programa de arranque en el chip ejecutado en el núcleo AVR.

El ATmega328 es un microcontrolador de gran alcance que proporciona una solución altamente flexible y rentable para muchas aplicaciones de control embebido.

El AVR 328 es compatible con una gran gama de programas y herramientas de desarrollo incluyendo: Compilador C, Programas depurador/Simuladores, Macro Ensambladores, en Emuladores de circuitos y kits de evaluación.

3.2 Memoria Micro SD

El sistema necesita un método de almacenamiento masivo, para poder guardar todas las mediciones hechas durante un periodo de tiempo. Se precisará por diferentes motivos, el principal es para poder almacenar los datos adquiridos por el microcontrolador para un futuro procesamiento o envío por parte del sistema, para ello se ha empleado un modulo de memoria MicroSD.

Las tarjetas **MicroSD** o Transflash corresponden a un formato de tarjeta de memoria flash más pequeña que una tarjeta MiniSD convencional. Ha sido desarrollada por SanDisk y adoptada por la Asociación de Tarjetas SD (SD Card Association) bajo el nombre de MicroSD en julio de 2005. Mide tan sólo $15 \times 11 \times 0,7$ milímetros, lo cual le da un área de 165 mm^2 . Esto es tres veces y media más pequeña que la miniSD, que era hasta ahora el formato más pequeño de tarjetas SD, y es alrededor de un décimo del volumen de una SD card.

Este hecho es muy importante para el sistema divido a que el tamaño puede ser crítico, puesto que habrá que insertarla en el Arduino.

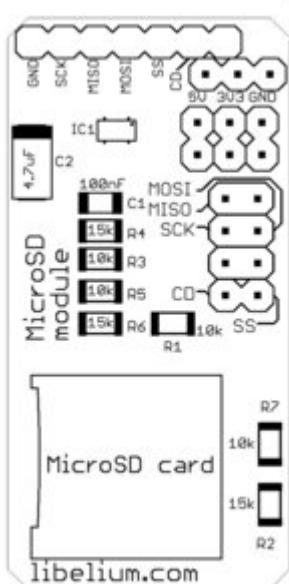


Figura 3.2.2 Componentes del módulo MicroSD.



Figura 3.2.1 Módulo de memoria MicroSD.

Sus tasas de Transferencia no son muy altas, sin embargo, empresas como SanDisk han trabajado en ello, llegando a versiones que soportan velocidades de lectura de hasta 10Mbit/s.

Debido a que su coste como poco duplica el de una Secure Digital equivalente, su uso se ciñe a aplicaciones donde el tamaño es crítico, como los teléfonos móviles, sistemas GPS o tarjetas Flash para consolas de mano.

El módulo se ha adquirido de la empresa *Libelium* la cual suministra componentes para el microcontrolador Arduino.

Además la empresa suministra una serie de librería para poder utilizar dicho módulo estas librerías son de gestión de memoria. Habrá que realizar las diferentes funciones en C para poder obtener un uso correcto del modulo MicroSD.

La estructura del módulo MicroSD se puede ver en la figura 3.2.2. Consta de diferentes componentes. No habrá que realizar ninguna operación sobre el módulo excepto el de insertar la tarjeta de memoria en la ranura a tal efecto y colocar el Módulo en la posición correcta en el Arduino. Tiene diferentes pines para poder acoplarlos al microcontrolador. Se puede hacer de dos formas al Arduino, la primera forma es a través de los pines 8 al 13 del Arduino o a través de los conectores ICSP. Dependiendo de que opción de acoplarlos se elija habrá que utilizar unos pines u otros del módulo. Por motivos de funcionalidad y sencillez se ha elegido utilizar los pines 8 al 13.

Al Haber elegido el método de los pines 8 al 13 el Jumper del módulo habrá que ponerlo en los pines 5V y 3V3 para un correcto funcionamiento.

Al haber elegido el método de conexión de los pines digitales del 8 al 13 el modulo MicroSD se engancha de una manera fácil al Arduino. Solamente hay que enganchar el Módulo MicroSD al Arduino de la forma correcta como se muestra en la figura 3.2.3.

Habrá que implementar las diferentes funciones para su uso, tales como Escritura, lectura, borrado.



Figura 3.2.3 Módulo MicroSD acoplado al Arduino.

3.2.1 Consideraciones previas

- Hay que tener en cuenta que no podremos crear ficheros por lo que los ficheros tendrán que crearse previamente en un ordenador e importarlos a la tarjeta MicroSD. En nuestro caso como lo que queremos serán los diferentes días del año para guardar las diferentes variables, no tendremos problemas puesto que sabemos los ficheros que queremos tener, uno por cada día de captación de variables, lo ideal sería un año completo por lo que necesitamos 365 archivos diferentes. Crearemos 365 ficheros con extensión txt y pondremos a cada uno el nombre de un dia. Así un archivo quedará como 09-09-10.txt.
- Al crear los diferentes archivos cada uno de ellos graba en la tarjeta MicroSD un carácter especial de fin de Archivo, el cual se utiliza para localizar los diferentes archivos dentro de la tarjeta. Habrá que tener en cuenta que el microcontrolador

está limitado en sus funciones y escribe de manera secuencial dentro de la tarjeta. Por lo que para poder incrementar un archivo perderemos información del archivo contiguo al que estamos incrementando, por lo que no podremos incrementar archivos. Para solucionar este problema tenemos que crear el archivo con la dimensión que nosotros necesitamos. Para ello llenamos los archivos con un carácter especial para poder insertar los diferentes datos adquiridos. El carácter elegido es EOF.

- Cuando queramos hacer diferentes pruebas y diferentes ensayos con el modulo MicroSD tendremos que borrar los archivos creados, para ello no podemos borrarlos con el microcontrolador, por operatividad; además si borramos un archivo o suprimimos parte de este con el ordenador el carácter fin de archivo se queda grabado en la tarjeta y no podremos hacer búsquedas de archivos y tendremos problemas con la lectura y la escritura en dicha tarjeta. Para borrar un archivo o parte de este tendremos que formatear la tarjeta MicroSD para luego poder utilizarla con toda normalidad.
- Debemos saber el tipo de caracteres que queremos insertar en nuestro archivo. Puesto que según el tipo de caracteres el proceso de lectura y escritura serán diferentes. En este caso se ha elegido caracteres tipo ASCILL para insertar en nuestros archivos.

3.2.2 Lectura Tarjeta MicroSD

A continuación se muestra el código desarrollado para hacer la lectura de la tarjeta MicroSD. La lectura de la tarjeta es la acción por la cual se transmite los datos de un archivo de la tarjeta MicroSD en el puerto serial del Arduino. Puede ser una pantalla, otro Arduino, u otro dispositivo. En este caso será el módulo GPRS. El código sería:

```
int SDPantalla(const char* filename){
    mmc::initialize();
    if (microfat::initialize(buffer)){
        if (microfat::locateFileStart(filename, sector, length)){
            if (length > BYTESPERSECTOR) {
                sectors = length / BYTESPERSECTOR;
                length = BYTESPERSECTOR;
            }
            for( long j = 0; j <= sectors; j++)
                if (RES_OK == mmc::readSectors(buffer, sector + j, 1)){
                    if (length > BYTESPERSECTOR)
                        length = BYTESPERSECTOR;
                    for(int i = 0; i < length; ++i){
                        if (buffer[i] == EOF) {
                            Serial.println();
                            return 0;
                        }
                        Serial.print(buffer[i],BYTE);
                    }
                }
            else return 1;
        }
    }
}
```

```

        }
        else return 1;
    }
    else return 1;
    Serial.println();
    return 0;
}

```

3.2.3 Escritura tarjeta MicroSD

A continuación se muestra el código usado para hacer la escritura en la tarjeta MicroSD. La escritura de la tarjeta es la acción por la cual se copia los datos captados por los diferentes sensores en la tarjeta MicroSD. Habrá que tener en cuenta que los datos se graban como caracteres en código ACILL. El código sería:

```

int InsertarSD(const char* filename, char* data){
    int offset1 = 0;
    long file1Length = usedBytes(filename);
    offset1 = (int) (file1Length % BYTESPERSECTOR);
    sectors = file1Length / BYTESPERSECTOR;
    if (RES_OK == mmc::readSectors(buffer, sector + sectors, 1)){
        if (length > BYTESPERSECTOR)
            length = BYTESPERSECTOR;
        int count = 0;
        for(count = offset1; count < length; ++count){
            if (data[count-offset1] != 0) {
                buffer[count] = data[count-offset1];
            }
            else break;
        }
        for(int i = count; i < length; ++i){
            buffer[i] = EOF;
        }
        for (int i = 0; i < 10; ++i){
            if (RES_OK == mmc::writeSectors(buffer, sector + sectors, 1)){ }
            else return 1;
        }
        if (data[count-offset1] != 0) {
            int count2 = 0;
            for(count2 = 0; count2 < length; ++count2){
                if (data[count2 + count - offset1] != 0) {
                    buffer[count2] = data[count2 + count - offset1];
                }
                else break;
            }
            if (count2 < length){
                buffer[count2] = EOF;
                for(int i = count2 + 1; i < length; ++i){
                    buffer[i] = ' ';
                }
                for (int i = 0; i < 10; ++i){
                    if (RES_OK==mmc::writeSectors(buffer,sector+sectors+1,1)){
                    }
                    else return 1;
                }
            }
        }
    }
    else return 1;
    return 0;
}

```

3.2.4 Resultado escritura Tarjeta MicroSD

El resultado de la escritura en la tarjeta MicroSD será un archivo en txt, el nombre del archivo será el día de la captación, el contenido del archivo se puede ver en la figura 3.2.4.

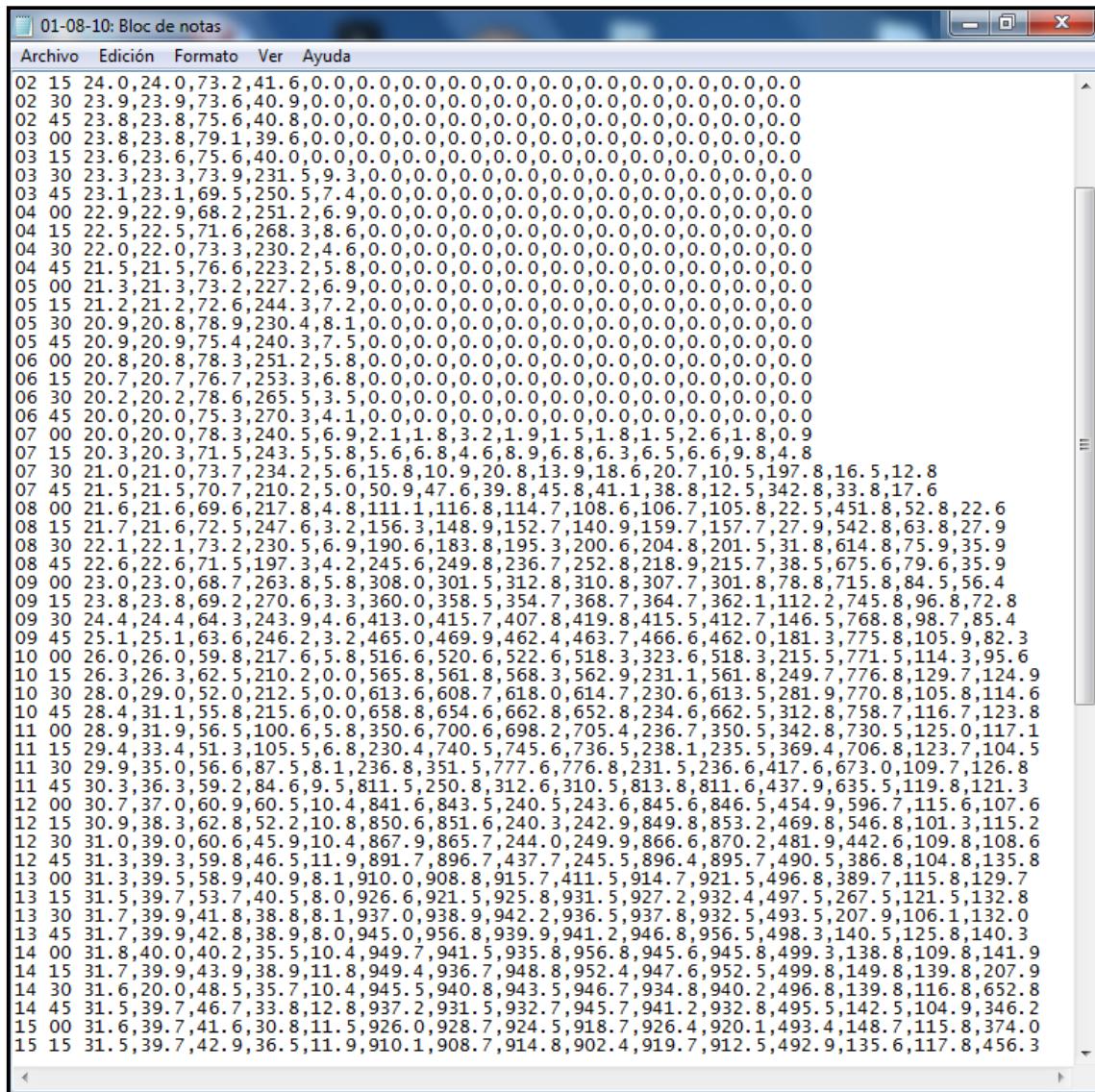


Figura 3.2.4 Archivo con los datos.

Como se puede observar en la figura 3.2.4 la lectura del archivo es de difícil compresión. Para una mejor comprensión por parte de cualquier usuario del sistema, se ha desarrollado un programa software que sea capaz de interpretar estos datos y representarlos de manera que sea comprensible para la mayoría de usuarios del sistema desarrollado en este proyecto.

3.3 GPRS

EL sistema necesita conexión vía radio para poder acceder a los datos sin necesidad de ir físicamente donde está situado. En este caso se ha elegido comunicación GPRS para poder comunicar cualquier zona sin necesidad de tener ningún sistema de comunicaciones cercano, como sería el caso de comunicación vía bluetooth ó wifi.

Para la comunicación GPRS, en este proyecto, se ha elegido el módulo de libelium GSM/GPRS, ver figura 3.3.1, es un módulo de comunicación que se acopla en el Arduino.

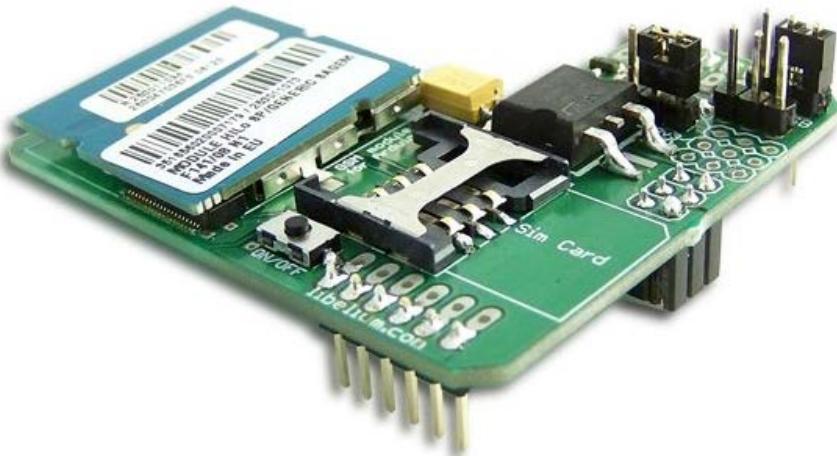


Figura 3.3.1 Módulo GPRS.

GPRS (General Packet Radio Service) o servicio general de paquetes vía radio es una extensión del Sistema Global para Comunicaciones Móviles (Global System for Mobile Communications o GSM) para la transmisión de datos no conmutada (o por paquetes). Permite velocidades de transferencia de 56 a 144 kbps.

Para establecer una conexión GPRS con un modem, un usuario debe especificar un APN, un usuario y contraseña, opcionalmente una dirección IP y la DNS correspondiente. Con GPRS pueden utilizar servicios tales como Wireless Application Protocol (WAP), servicio de mensajes cortos (SMS), servicio de mensajería multimedia (MMS), Internet y para los servicios de comunicación, como el correo electrónico y la World Wide Web (WWW).

Este módulo además de comunicaciones de datos GPRS también dispone de comunicaciones GSM para hacer llamadas y mandar y recibir mensajes.

Este módulo tiene diferentes comunicaciones dependiendo de la colocación de sus jumpers. Como se observa en la figura 3.2.2, existen 2 Jumper a configurar, el primero, decide como se alimenta el módulo; puede hacerlo a través del Arduino a 5 voltios o con una fuente de alimentación externa. El segundo, un par de jumpers, deciden cómo se comunica el Módulo. Se puede comunicar con el Arduino, bien para que envíe las

instrucciones para que el módulo realice las operaciones deseadas, o utilizarlo como puerto de comunicación para comunicar el módulo con un ordenador u otro dispositivo a través del USB del Arduino. Para poder establecer este tipo de comunicación hace falta un programa de comunicaciones como “cutecom” o “hypertermianl”. El dispositivo también nos da la opción de conectar un micrófono y un altavoz para usar el módulo para realizar llamadas de voz.

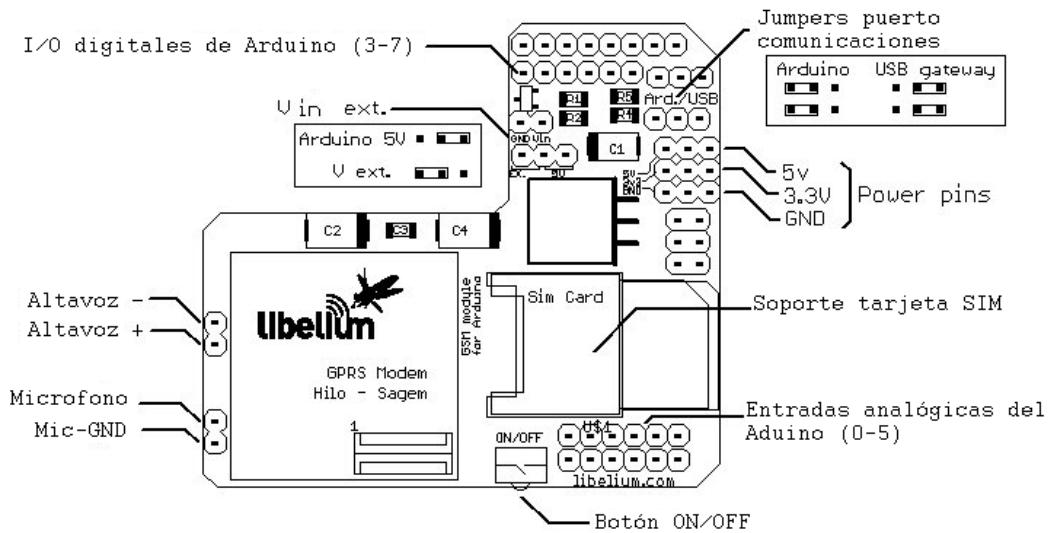


Figura 3.3.2 Esquema de los elementos módulos GPRS.

La colocación de los jumpers en la realización de este proyecto son de Arduino 5V en el jumper de alimentación, que alimentará el módulo a través del Arduino y en los Jumperes de comunicaciones se coloca en la opción Arduino puesto que el módulo debe comunicarse con el Arduino ya que es este dispositivo el que tiene los datos a enviar.

La alimentación del módulo se realiza a través del Arduino, pero hay que tener en cuenta que el módulo GPRS precisa de una fuente de alimentación que sea capaz de proporcionar una corriente de alimentación superior a 2'2 Amperios. Por lo que aunque se alimente a través del Arduino, éste debe de estar alimentado con una fuente que proporcione una con un amperaje superior a los 2'2A mencionados anteriormente. Si la fuente de alimentación es inferior se debe colocar un condensador entre los pines 5V y GND del módulo GPRS para que pueda proporcionar la potencia necesaria cuando el módulo lo requiera. Para este caso no será necesario puesto que el Arduino se alimentará a través de una batería, la cual puede proporcionar la intensidad necesaria requerida por el módulo GPRS. Aún así se ha colocado el condensador para reducir los picos de consumo producido por este módulo.



Figura 3.3.3 Módulo GPRS acoplado al Arduino.

En esta figura también se puede observar como se coloca la tarjeta SIM en el módulo GPRS.

Para establecer la comunicación GPRS se ha optado por la compañía Vodafone. Los parámetros de configuración de la conexión deben ser suministrados por la compañía de teléfono que se contrate.

En este caso los datos son:

- APN: airtelnel.es.
- Login: vodafone.
- PassWord: vodafone.
- IP: 0.0.0.0 (ip dinámica).
- DNS1: 212.073.032.003.
- DNS2: 212.073.032.067.

Una vez establecida la conexión GPRS se precisa una forma de comunicación entre el sistema y el usuario que pretenda acceder a los datos. En este caso el método elegido es una comunicación a un servidor FTP.

Como se ve en la figura 3.3.3, no se puede acceder a las entradas analógicas y digitales que quedan libres después de conectar el módulo GPRS, se debe acceder a través de este módulo, puesto que cuando se conecta el módulo GPRS se anula todas las entradas del Arduino para que el módulo esté bien sujeto a este.

3.3.1 Servidor FTP

Una vez establecidos los parámetros de conexión GPRS, habrá que establecer un tipo de comunicación para enviar los datos recogidos. Se pueden emplear diferentes métodos y establecer por diferentes medios, mediante Conexión TCP, conexión UDP, por correo electrónico, MMS, por conexión directa a otro dispositivo GPRS, etc. Por diferentes motivos se han ido descartando cada opción.

En este proyecto se ha optado por el envío de datos a un servidor FTP. Este servidor puede ser un servidor FTP que el usuario disponga en su ordenador y con su conexión a Internet o un servidor externo al que se pueda acceder para poder descargar los ficheros con los datos del sistema. Esta opción ha sido la elegida puesto que se puede enviar los datos cuando se disponga sin necesidad de tener que habilitar ninguna conexión, además se puede estar el tiempo que se desee sin consultar los datos y no se perderá la información, puesto que estarán los archivos en el servidor seleccionado. Otra ventaja es que se puede disponer de esos archivos en cualquier sitio con conexión a Internet.

Se ha configurado un servidor web en un ordenador particular, que permanece encendido la mayoría del tiempo.

Para ello se han realizado una serie de acciones:

- 1) Identificar la dirección IP de la conexión a Internet desde la que se conecta el servidor FTP. Se puede conocer a través de www.cualesmiip.com. La dirección IP en este caso es 80.36.10.112.
- 2) Crear el servidor FTP. Para el proyecto se ha elegido el servidor *FTP Quick'n Easy FTP Server 3.1 Lite*. Se ha elegido dicho servidor por su facilidad de uso y por ser de uso libre.
- 3) Se configura la conexión del servidor. En la opción *Show configuration*. Las opciones que hay que configurar son el puerto del servidor que habrá que definirlo como el 21. Después hay que abrir el puerto 21 del router de la conexión Internet y redirigir ese puerto al ordenador donde esté el servidor FTP.
- 4) Crear una cuenta para acceder al servidor FTP. La cuenta creada tendrá una serie de permisos, hay que darle todos los permisos para la cuenta seleccionada. Esa cuenta tiene un usuario y una contraseña que luego hay que emplear para la programación del módulo GPRS para que pueda acceder al servidor FTP. No dejar la contraseña en blanco, el módulo FTP da error.

3.3.2 Problemas

En esta sección se expondrán los problemas acaecidos con el módulo GPRS. Estos problemas son el resultado de incidencias ajenas a las propias de la configuración y programación. Debido al tiempo y al esfuerzo empleado en la resolución de estos problemas se ha optado por crear este apartado. Para definirlos se utilizará un diagrama de tiempo con las diferentes incidencias y comunicaciones a la empresa fabricante del módulo para entenderlos.

- **Abril 09**

Pedimos el módulo GPRS a la empresa Libelium. Al retrasarse el pedido me pongo en contacto con la empresa para ver la incidencia ocurrida.

- **21 mayo 09**

Tras ponernos en contacto con la compañía por diferentes medios recibimos el email que se muestra en la figura 3.3.4.

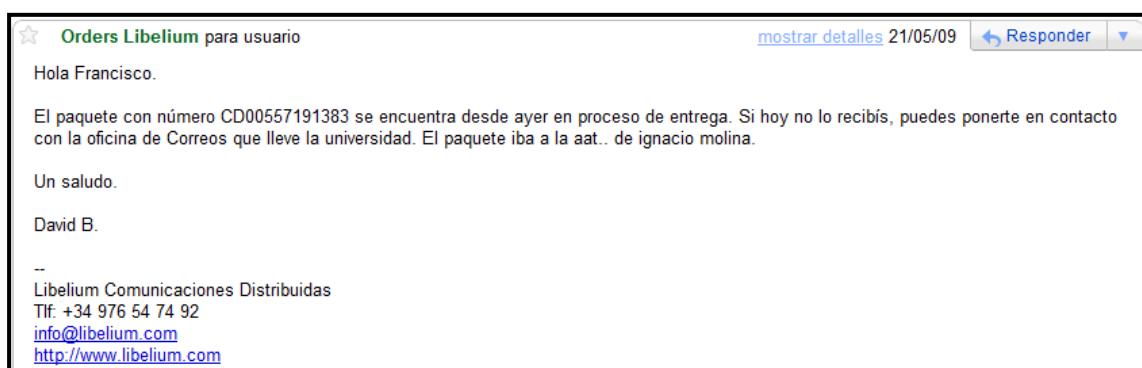


Figura 3.3.4 Email recibido de Libelium 1.

Al día siguiente se recibe el módulo en cuestión y se empieza la configuración y programación del módulo GPRS.

- **21 mayo- 29 Junio 09**

Después de más de un mes intentando que el módulo funcione, éste no realiza ninguna acción. Utilizamos el *hyperterminal* para ver cuales son las respuestas del módulo ante unas entradas determinadas. El resultado siempre es negativo, puesto que no realiza acción ninguna. Después de comprobar que no hay solución e intentar múltiples actuaciones sobre el módulo decidimos ponernos en contacto con el fabricante.

A continuación se puede ver la cabecera de este email, que consta de varios párrafos más:

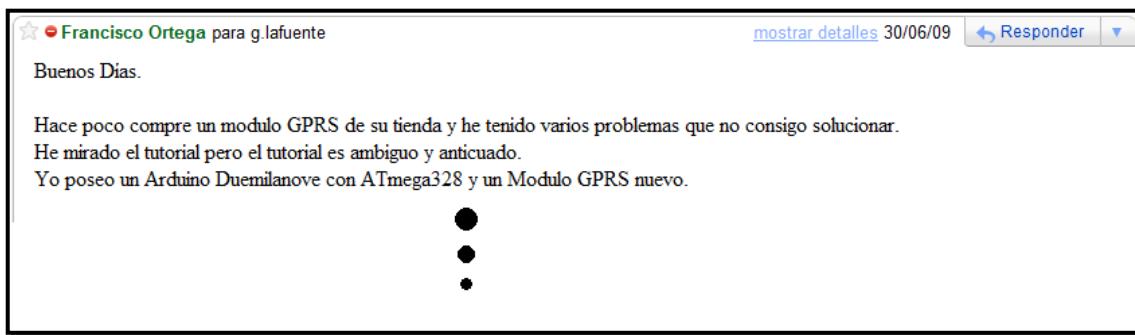


Figura 3.3.5 Email enviado a Libelium 1.

Después de una serie de emails intercambiados con el fabricante, como se ve en la figura 3.3.5, para buscar una solución, contándole previamente los múltiples problemas a la hora de conseguir su correcta puesta en marcha, decidimos que el módulo tiene algún tipo de problema físico o eléctrico y decidimos devolver el módulo al fabricante.

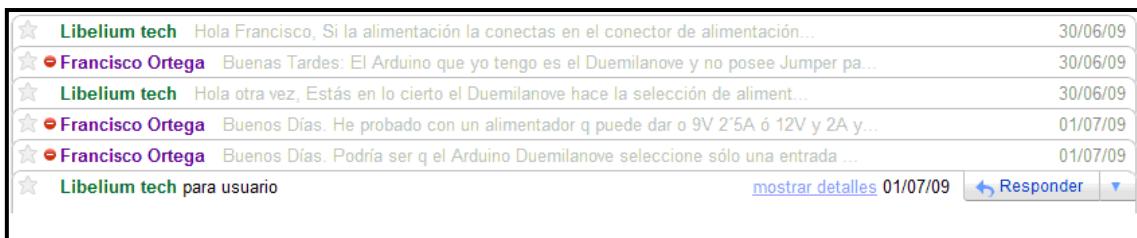


Figura 3.3.6 Lista de emails intercambiados con Libelium 1.

El fabricante, una vez comprobado que efectivamente el problema que tenemos con el módulo GPRS es que el módulo tiene algún tipo de problema físico, decide cambiarlo

- **2 julio- 8 noviembre 09**

Durante varios meses, realizando todo tipo de pruebas con el módulo, tanto con el *Hyperterminal* como a través del *Arduino*. Conseguimos buenos resultados tanto en recibir como en hacer llamadas y mensajes, pero no en realizar una conexión GPRS estable. Existen problemas cuando se intenta realizar una conexión TCP ó UDP, el módulo se apaga. Tras muchas pruebas con el tutor del proyecto y preguntar a personas más especializadas en el tema, realizamos muchas pruebas con diferentes propuestas dadas, pero no conseguimos un resultado óptimo.

Decidimos volver a ponerme en contacto con el fabricante para saber el posible problema con el módulo. Para ello le escribimos un email explicando la situación, se puede ver en la figura 3.3.6.

Francisco Ortega para Libelium

Buenos Días.

Hace ya algún tiempo compro un módulo GPRS después de mucho pelearme con él he conseguido hacer llamadas y mandar sms pero me gustaría saber si se puede hacer trasferencia GPRS y como.
He intentado mandar archivos por FTP y no me deja por TCP y no me deja. Me gustaría saber si alguien lo ha conseguido y como. Me hace falta mandar datos por GPRS y si no mandas datos por GPRS no se para que lo quiero.
Espero su respuesta cuanto antes.

Muchas gracias

mostrar detalles 08/11/09 [Responder](#)

[Responder](#) [Reenviar](#)

Figura 3.3.7 Email enviado a Libelium 2.

Después de varios meses pensando que la configuración era errónea e intentando diversas pruebas. La respuesta dada por el fabricante se puede ver en la figura 3.3.7.

Libelium tech para usuario

Hola,

Para enviar datos usando GPRS hemos detectado algunos problemas con el firmware del GPRS, por lo tanto te recomendaría que actualizaras el firmware, puedes descargar los ficheros necesarios aquí:
http://www.libelium.com/squidbee/index.php?title=Uploading_the_firmware

Saludos

- Mostrar texto citado -

--

Libelium Comunicaciones Distribuidas
Tlf: +34 976547492
<http://www.libelium.com>

mostrar detalles 09/11/09 [Responder](#)

Figura 3.3.8 Email recibido de Libelium 2.

El problema que se ha tenido a la hora de poder hacer conexiones GPRS ha sido de firmware por parte del propio módulo.

- **8 noviembre – 13 noviembre 09**

Intentamos cargar el firmware como se especifica en la página, pero no describe bien el proceso y no conseguimos cargarlo. El tutor de este proyecto, está presente en todo el proceso y entre los dos analizamos la forma correcta de cargarlo según la escasa explicación por parte del fabricante. No lo conseguimos cargar correctamente, por lo que decidimos volver a preguntar al fabricante.

- **13 noviembre 09**

Decidimos volver a escribir un email al fabricante para explicar la situación de cargar el firmware. En la figura 3.3.8 se puede ver el email enviado al fabricante.

Francisco Ortega para g.lafuente [mostrar detalles](#) 13/11/09 [Responder](#)

Buenos Días.

Compre el modulo GPRS sobre el mes de abril. Puse a funcionar el módulo y el módulo no hacia nada. Después de mucho luchar con el modulo, muchos correos intercambiados con Libelium y ninguna solución descubrí que el módulo estaba roto. libelium me cambió el modulo. El segundo módulo funcionaba pero se me apagaba por falta de alimentación. Después de muchos correos con Libelium, puse un alimentación de 3A y un condensador como dice en la página, pero seguía apagándose. Descubrí que el problema es que cuando estaba la fuente externa y el USB conectado se alimentaba del USB por lo que la fuente externa no funciona, con lo que hay que desconectar el usb para tener corriente de la fuente externa.

Ahora, después de mucho, mucho pelearme con el módulo y unos cuantos correos enviados me dicen que el problema es del Firmware, me dan una pagina donde las explicaciones brillan por su ausencia.

No puedo entender como venden un módulo con el firmware desactualizado para mandar datos en un módulo GPRS que se utiliza para eso.

Intento cargar el firmware con las pocas explicaciones que me da la página y me una serie de mensajes que no entiendo muy bien. Se supone que hace el "Downloading" bien pero en el "Restoring Parameter" me da un error y pone "nok".

Sin mayores explicaciones, no se lo que ha pasado. Intento probar el módulo y ahora ni hace llamadas ni manda sms que antes sí hacia. Entiendo que no se han cargado bien los parámetros.

Estoy un poco cansado de tantos email's y de tener una lucha continua con el módulo GPRS me gustaría que me dieran una solución. Han pasado muchos meses y mucho esfuerzo por mi parte con un módulo que supuestamente lo venden listo para funcionar.

Me podrían mandar un módulo que esté listo para funcionar o darme algún tipo de solución.

Espero su respuesta lo antes posible ya que me urge una solución. Gracias de antemano.

Un saludo.

Figura 3.3.9 Email enviado a Libelium 3.

En la figura 3.3.8 se puede observar la respuesta por parte del fabricante.

Gerardo Lafuente para usuario [mostrar detalles](#) 13/11/09 [Responder](#)

Hola Francisco.

Mándanos el módulo GPRS y tan pronto lo tengamos aquí te mandaremos uno nuevo inmediatamente.

También es cierto que el tutorial para actualizar el firmware no es muy explícito, así que el que te envíemos llevará el firmware correcto.

No obstante en éstos días, quizás hoy, colgaremos en la web unas nuevas capturas de pantalla para posteriores ocasiones.

Disculpa las molestias

Un saludo

Figura 3.3.10 Email recibido de Libelium 3.

En este punto el módulo es devuelto al fabricante para que le instalen el firmware adecuado. El fabricante nos envía otro módulo diferente con el firmware adecuado.

- **15 noviembre – 22 noviembre**

Se prueba de nuevo el siguiente módulo y no funciona correctamente. Realizamos diferentes acciones pero no comunica con el servidor FTP por lo que el problema no se soluciona. En este momento decidimos volver a ponerme en contacto con el fabricante y devolverle el módulo para que me manden otro correcto.

- **22 noviembre 1 diciembre**

Tras otra serie de emails intercambiados con el fabricante, ver figura 3.3.9, el fabricante nos informa que el módulo funciona correctamente y le informamos que lo han probado más de 1 personas con mucha experiencia en la programación de diferentes tipos de dispositivos electrónicos.

⭐	Francisco Ortega	Buenos días. El módulo que me enviaron no funciona correctamente. Se apaga en...	22/11/09
⭐	Francisco Ortega	Mensaje reenviado ----- De: Francisco Ortega <fozamorano@gmai...	23/11/09
⭐	Libelium tech	Hola Francisco, Según entiendo, el módulo funciona hasta que lo conectas con ...	23/11/09
⭐	Francisco Ortega	Buenos días. No, no tiene el PIN conectado la SIM. Yo el otro módulo también ...	23/11/09
⭐	Libelium tech	Hola, La versión de firmware que lleva es la 7.0, y está probado con Arduino ...	23/11/09
⭐	Francisco Ortega	Buenos días. Yo uso el arduino Duemilanove con el software Versión 16 de la p...	23/11/09
⭐	Libelium tech	Buenos días, Si tienes contacto entre GND y 3.3 V, no lo vuelvas a encender, ...	24/11/09
⭐	Gerardo Lafuente	Marcos ha enviado éste correo a tech o solo a mí? por favor respóndele desde ...	24/11/09
⭐	Francisco Ortega	Buenos Días. Detecté ese contacto antes de poner el condensador. Ademas yo po...	24/11/09
⭐	Libelium tech	Buenos días Francisco, Lo mejor será que nos envíes el módulo para que lo pod...	24/11/09
⭐	Francisco Ortega	Buenos días. El módulo lo he enviado. Después a compañeros y al director del ...	26/11/09
⭐	Libelium tech	Buenos días Francisco, El módulo ha llegado esta mañana, lo he conectado a un...	27/11/09
⭐	Francisco Ortega	Buenos días. El problema es que yo lo pruebo en modo standalone. De hecho me ...	27/11/09
⭐	Libelium tech para usuario	mostrar detalles 01/12/09	Responder ▾

Figura 3.3.11 Lista de emails intercambiados con Libelium 2.

- **Diciembre 09**

El fabricante devuelve el módulo diciendo que funciona correctamente y que han cambiado el tutorial colgado en la web para la configuración del módulo GPRS. En ese punto me devuelven el módulo. Se puede observar en la figura 3.3.10.

⭐	Libelium tech para usuario	mostrar detalles 01/12/09	Responder ▾
Buenos días Francisco,			
Te repito que tu módulo funciona perfectamente, lo acabo de probar en standalone con Arduino Duemilanove, de esas pruebas ha salido el tutorial que acabo de colgar en la web.			
http://www.libelium.com/squidbee/index.php?title=New_GPRS_module_for_Arduino_%28Hilo_-_Sagem%29#Connecting_Arduino_to_Internet			
Así que procedemos a enviártelo (a portes debidos como ya te comenté) para que puedas probarlo.			

Figura 3.3.12 Email recibido de Libelium 4.

Después de recibir el módulo se realizan las mismas pruebas que la última vez que se le envió al fabricante y efectivamente empieza a dar los resultados esperados.

En este momento se puede empezar a programar adecuadamente el módulo GPRS.

3.3.3 Comandos AT

Los comandos AT es un lenguaje desarrollado por la compañía Hayes communications que prácticamente se convirtió en estándar abierto de comandos para configurar y parametrizar módems. Los caracteres «AT», que preceden a todos los comandos, significan «Atención», e hicieron que se conocieran como comandos AT.

Para programar el módulo GPRS hay que utilizar un subconjunto de los comandos AT, este tipo de comandos hay que programarlos para que se escriba en mayúscula.

A continuación se muestra una serie de comandos AT utilizados en este proyecto.

3.3.3.1 Comandos específicos de GPRS

Son empleados para definir y configurar una conexión de datos vía GPRS por medio de la compañía elegida para este proyecto. Se pondrá una tarjeta SIM de esa misma compañía. Los comandos son los siguientes:

- **AT+KCNXCFG=<cnx cnf>,"GPRS",<apn>,[<login>],[<password>],
[<ip>],[<dns1>],[<dns2>]**

Parámetros

<cnx cnf>: Índice de un grupo de parámetros para configurar una conexión (Valor max:200).
<apn>: (Access Point Name) Cadena String (Tamaño max 63 bytes) ,nombre lógico utilizado para seleccionar el GGSN o la red de paquetes de datos externos.
<login>: Una cadena string (Tamaño max 24 bytes), indica el nombre de usuario del cnx
<password>: Una cadena string (Tamaño max 24 bytes), indica el password del cnx
<ip>: Una cadena string. Ip de la conexión. Una Ip dinámica se escribe con el valor "0.0.0.0"
<dns1>, <dns2>: Una cadena string. DNS de la conexión

- **AT+KCNXTIMER=<cnx cnf>,[<tim1>],[<nbrtrial>],[<tim2>]**

Parámetros

<tim1>: Tiempo espera de conexión en segundos. Valor entre de 15s a 120s (30 por defecto)
<tim2>: Tiempo de desconexión. Debe ser de 60 a 300 (60 por defecto)
0: desactivado (conexión no se cierra por sí mismo)
<nbrtrial>: Número de intentos de conexión a la red. Valor entre 1 y 4 (2 por defecto)

- **AT+KCNXPROFILE=<cnx cnf>**

3.3.3.2 Comandos específicos de FTP

Son los utilizados para configurar y definir la conexión con el servidor FTP. Habrá que tener en cuenta que para tener una conexión FTP primero habrá que disponer de una conexión GPRS previamente establecida por el usuario. Los comandos utilizados son los siguientes:

- **AT+KFTPCFG=[<cnx cnf>],<server_name>,[<login>],<password>,[<remote port number>],[<mode>]**

Parámetros

<Server name>: String. Nombre del servidor FTP al que se desea conectar.

<login>: Cadena string. Indica el usuario de la conexión

<password>: Cadena stringo. Contraseña de la conexión FTP

<remote port number >: Numérico (0 a 65535). Indica el puerto de la conexión (21 por defecto).

<modo>: Indica el modo de la conexión FTP.

0 - Activa. El servidor es el que inicia la conexión FTP.

1 - pasiva. El cliente es el que inicia la conexión FTP, para evitar el filtrado de Proxy.

- **AT+KFTPSND=[<server_path>],<file_name>,[<type of file>]**

Parámetros

<server_path>: String. Indica la ruta del archivo para ser cargado. Una cadena vacía indica que el archivo está en el raíz del servidor.

<file_name>: String. Indica el nombre del archivo a enviar.

<ftp_cause>: Indica el tipo de archivo para transferir (ASCII o binario)

0 - binarios, (valor predeterminado)

1 - ASCII.

- **AT+KFTPCLOSE**

3.3.4 Programación GPRS

El programa realizado para la comunicación FTP a través de una conexión GPRS con el módulo de comunicaciones de Libelium es el que se muestra a continuación. Se muestra en dos columnas para que ocupe menos espacio en la redacción de este proyecto:

```

delay(1500);                                Serial.flush();
Serial.println("AT&K3");                      Serial.println("AT+KCNXTIMER=0,60,2,0");
delay(5000);                                delay(5000);
Serial.print("AT+KCNXCFG=0,");                Serial.println("AT+KCNXPROFILE=0");
Serial.print(34,BYTE);                        delay(2000);
Serial.print("GPRS");                         Serial.println("AT+CGATT=1");
Serial.print(34,BYTE);                        delay(2000);
Serial.print("airtelnet.es");                  Serial.flush();
Serial.print(34,BYTE);                        Serial.print("AT+KFTPCFG=0,");
Serial.print("airtelnet.es");                  Serial.print(34,BYTE);
Serial.print(34,BYTE);                        Serial.print("80.36.10.112");
Serial.print("airtelnet.es");                  Serial.print(34,BYTE);
Serial.print(34,BYTE);                        Serial.print("redow");
Serial.print("airtelnet.es");                  Serial.print(34,BYTE);
Serial.print(34,BYTE);                        Serial.print("aviador");
Serial.print("airtelnet.es");                  Serial.print(34,BYTE);
Serial.print(34,BYTE);                        Serial.println(",21,1");
Serial.print("0.0.0.0");                      delay(2000);
Serial.print(34,BYTE);                        Serial.flush();
Serial.print(",");                           Serial.print("AT+KFTPSND=0,,");
Serial.print(34,BYTE);                        Serial.print(34,BYTE);
Serial.print("212.073.032.003");  Serial.print(fecha);
Serial.print(34,BYTE);                        Serial.print(34,BYTE);
Serial.print(",");                           Serial.println(",0");
Serial.print(34,BYTE);                        delay(20000);
Serial.print("212.073.032.067");  SDPantalla(fecha);
Serial.println(34,BYTE);                      delay(5000);
delay(2000);                                Serial.println("AT+KFTPCLOSE");

```

La parte de código anteriormente implementado hará las siguientes funciones:

Enviado: AT&K3

Respuesta: OK

Activa el control del flujo del hardware.

Env: AT+KCNXCFG=0,"GPRS","airtelnet.es","vodafone","vodafone","0.0.0.0",
"212.073.032.003","212.073.032.067" **Res:** OK

Se Selecciona los parámetros de la conexión GPRS.

Env: AT+KCNXTIMER=0,60,2,0

Res: OK

Se Selecciona los parámetros de tiempo de la conexión.

Env: AT+KCNXPROFILE=0

Res: OK

Este comando AT especifica el perfil actual para configurar el GPRS o GSM de conexión para todas las posteriores conexiones TCP o FTP.

Env: AT+CGATT=1

Res: OK

Indica el estado de la conexión 1: Activa, 0: Desactiva.

Env: AT+KFTPCFG=0,"80.36.10.112","redow","aviador",21,1 **Res:** OK

Indica los parámetros de configuración de la conexión FTP.

Env: AT+KFTPSND=0,"archivo",0

Res: OK

Se realiza la conexión al servidor FTP. En este momento se envían todos los datos mediante el comando SDpantalla. Hay que tener en cuenta que el archivo tiene que tener un carácter especial de fin de archivo. Si fuera otro tipo de conexión habría que establecer cual es el fin de archivo y enviarlo.

Env: AT+KFTPCLOSE=0

Res: OK

Cierra la conexión FTP.

3.4 Tiempo real (RTC)

En el diseño del sistema debemos tener en cuenta la fecha y hora en cada momento. Esto se podría hacer con un reloj secuencial en el microcontrolador, pero para evitar que el reloj pierda la cuenta con una desconexión fortuita o con un reset involuntario, se ha decidido poner un reloj de tiempo real para que no pierda ni la hora ni la fecha.

Se ha elegido el circuito integrado DS1307, es un **RTC** (reloj en tiempo real) que se puede comunicar utilizando protocolo I²C contra el Arduino y que tiene funciones de backup de hora y fecha con su propia batería.

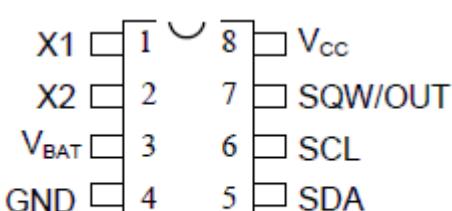
3.4.1 DS1307

Información extraída de [9]. El reloj/calendario del DS1307, provee información de segundos, minutos, horas, día, fecha, mes y año. La fecha al final del mes durante los meses con menos de 31 días, se ajusta automáticamente e incluye correcciones para el año bisiesto. El reloj funciona en cualquiera de los formatos de hora 24 o 12 horas con indicador AM/PM. El DS1307 tiene un circuito integrado en el sensor de energía que detecta los fallos de alimentación y cambia automáticamente a la batería.

Características:

- Reloj en tiempo real (RTC) cuenta los segundos, minutos, horas, día del mes y de la semana mes, y año. Con compensación por año bisiesto válida hasta 2100.
- 56-byte de memoria RAM para almacenamiento de datos, no volátil (NV) con el respaldo de la batería.
- Interfaz serial I²C.
- La señal de salida es programable de pulsos cuadrados.
- Detecta batería baja y cambia de alimentación automáticamente.
- Consumo menor de 500nA en modo de funcionamiento con el oscilador.
- Rango de temperatura: -40° C a +85° C.
- Disponible en 8-pin DIP o SOIC.

Asignación de pines:



DS1307 8-Pin DIP (300-mil)

Figura 3.4.1 Pines del Chip DS1307.

Descripción de pines:

- VCC: Fuente de alimentación primaria.
- X1, X2: Conexión oscilador a Cristal, 32.768kHz.
- VBAT: Entrada de la batería de 3V.
- GND: Tierra.
- SDA: Datos serie.
- SCL: Reloj serie.
- SQW/OUT: ondas cuadradas/señal de salida.

Estructura de un circuito típico

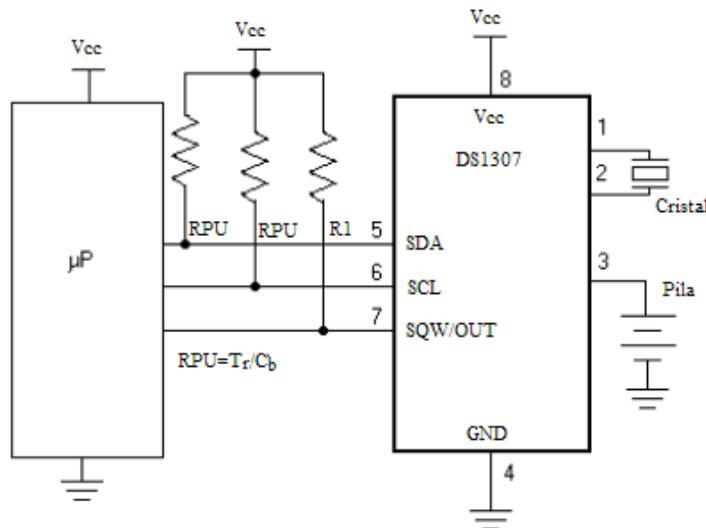


Figura 3.4.2 Estructura de un circuito típico para montar un DS1307.

3.4.2 BUS I²C

I²C es un bus de comunicaciones en serie, es la abreviatura de *Inter-Integrated Circuit* (Circuitos Inter-Integrados). Para simplificar la interconexión de dispositivos a un microprocesador, Philips desarrolló un sencillo bus bidireccional basado en dos hilos por el que se transmiten los datos vía serie.

La velocidad es de 100kbits por segundo en el modo estándar, aunque también permite velocidades de 3.4 Mbit/s. Es un bus muy usado en la industria, principalmente para comunicar micrcontroladores y sus periféricos en sistemas integrados y generalizando más para comunicar circuitos integrados entre sí que normalmente residen en un mismo circuito impreso.

La principal característica de I²C es que utiliza dos líneas para transmitir la información: una para los datos y otra para la señal de reloj. También es necesaria una tercera línea, pero esta sólo es la referencia (masa). Como suelen comunicarse circuitos en una misma placa que comparten una misma masa esta tercera línea no suele ser necesaria.

3.4.2.1 Descripción de las señales

- **SCL:** línea de los pulsos de reloj que sincronizan el sistema.
- **SDA:** línea por la que se mueven los datos entre los dispositivos.
- **GND:** común todos los dispositivos presentes en el bus.

Las líneas **SDA** y **SCL** son del tipo drenaje abierto, es decir, un estado similar al de colector abierto, pero asociadas a un transistor de efecto de campo (o FET). Se deben polarizar en estado alto (conectando a la alimentación por medio de resistencias "pull-up") lo que define una estructura de bus que permite conectar en paralelo múltiples entradas y

salidas. La figura 3.4.3 muestra la conexión de este tipo de bus. Las dos líneas del bus están en un nivel lógico alto cuando están inactivas.

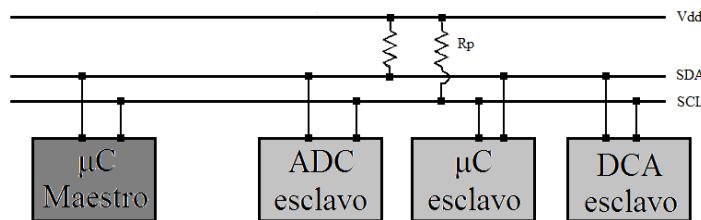


Figura 3.4.3 Diagrama de la conexión de un bus I²C.

En principio, el número de dispositivos que se puede conectar al bus no tiene límites, aunque hay que observar que la capacidad máxima sumada de todos los dispositivos no supere los 400pF. El valor de las resistencias de polarización no es muy crítico, y puede ir desde 1.8kΩ a 47kΩ. Un valor menor de resistencia incrementa el consumo de los integrados pero disminuye la sensibilidad al ruido y mejora el tiempo de los flancos de subida y bajada de las señales. Los valores más comunes en uso son entre 1.8kΩ y 10kΩ.

3.4.2.2 Definición de términos

- **Maestro:** Dispositivo que determina los tiempos y la dirección del tráfico en el bus. Es el único que aplica los pulsos de reloj en la línea SCL. Cuando se conectan varios dispositivos maestros a un mismo bus la configuración obtenida se denomina "multi-maestro".
- **Esclavo:** Todo dispositivo conectado al bus que no tiene la capacidad de generar pulsos de reloj. Los dispositivos esclavos reciben señales de comando y de reloj generados desde el maestro.
- **Bus libre:** Estado en el que ambas líneas (SDA y SCL) están inactivas, presentando un estado lógico alto. Es el único momento en que un dispositivo maestro puede comenzar a hacer uso del bus.
- **Comienzo:** Se produce cuando un dispositivo maestro ocupa el bus, generando la condición. La línea de datos (SDA) toma un estado bajo mientras que la línea de reloj (SCL) permanece alta.
- **Parada:** Un dispositivo maestro puede generar esta condición, dejando libre el bus. La línea de datos y la de reloj toman un estado lógico alto.
- **Dato válido:** Situación presente cuando un dato presente en la línea SDA es estable al tiempo que la línea SCL está a nivel lógico alto.
- **Formato de Datos:** La transmisión de un dato a través de este bus consiste de 8 bits de dato (1 byte). A cada byte transmitido al bus le sigue un noveno pulso de

reloj durante el cual el dispositivo receptor del byte debe generar un pulso de reconocimiento.

- **Reconocimiento:** El pulso de reconocimiento, conocido como ACK (del inglés Acknowledge), se logra colocando la línea de datos a un nivel lógico bajo durante el transcurso del noveno pulso de reloj.
- **Dirección:** Todo dispositivo diseñado para funcionar en este bus posee su propia y única dirección de acceso, preestablecida por el fabricante. Hay dispositivos que permiten definir externamente parte de la dirección de acceso, lo que habilita que se pueda conectar en un mismo bus un conjunto de dispositivos del mismo tipo, sin problemas de identificación. La dirección 00 es la denominada "de acceso general"; a ésta responden todos los dispositivos conectados al bus.
- **Lectura/Escritura (Bit R/W):** Cada dispositivo tiene una dirección de 7 bits. El octavo bit (el menos significativo) que se envía durante la operación de direccionamiento, completando el byte, indica el tipo de operación a realizar. Si este bit es alto el dispositivo maestro lee información proveniente de un dispositivo esclavo. Si este bit es bajo, el dispositivo maestro escribe información en un dispositivo esclavo.

3.4.2.3 Protocolo de comunicación del bus I²C

Habiendo varios dispositivos conectados sobre el bus, es lógico que para establecer una comunicación a través de él se deba respetar un protocolo. Digamos, en primer lugar, lo más importante: existen dispositivos maestros y dispositivos esclavos. Sólo los dispositivos maestros pueden iniciar una comunicación.

La condición inicial, de bus libre, es cuando ambas señales están en estado lógico alto. En este estado cualquier dispositivo maestro puede ocuparlo, estableciendo la condición de *inicio*. Esta condición se presenta cuando un dispositivo maestro pone en estado bajo la línea de datos (SDA), pero dejando en alto la línea de reloj (SCL).

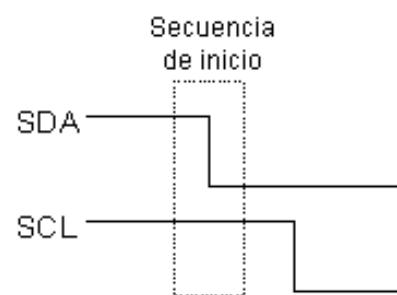


Figura 3.4.4 Condición de inicio Bus I²C.

El primer byte que se transmite luego de la condición de inicio contiene siete bits que componen la dirección del dispositivo que se desea seleccionar, y un octavo bit que corresponde a la operación que se quiere realizar con él (lectura o escritura).

Si el dispositivo cuya dirección corresponde a la que se indica en los siete bits (A0-A6) está presente en el bus, éste contesta con un bit en bajo, ubicado inmediatamente luego del octavo bit que ha enviado el dispositivo maestro. Este bit de reconocimiento (ACK) en

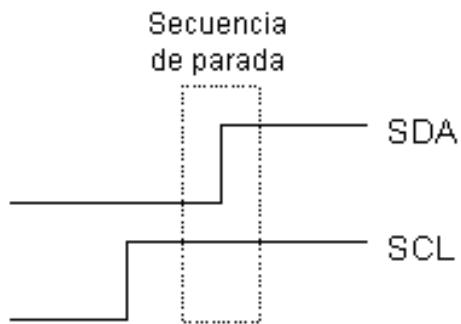
abajo le indica al dispositivo maestro que el esclavo reconoce la solicitud y está en condiciones de comunicarse. Aquí la comunicación se establece en firme y comienza el intercambio de información entre los dispositivos.



Figura 3.4.5 Secuencia de inicio de bus I²C.

Si el bit de lectura/escritura (R/W) fue puesto en esta comunicación a nivel lógico bajo (escritura), el dispositivo maestro envía datos al dispositivo esclavo. Esto se mantiene mientras continúe recibiendo señales de reconocimiento, y el contacto concluye cuando se hayan transmitido todos los datos.

En el caso contrario, cuando el bit de lectura/escritura estaba a nivel lógico alto (lectura), el dispositivo maestro genera pulsos de reloj para que el dispositivo esclavo pueda enviar los datos. Luego de cada byte recibido el dispositivo maestro (quien está recibiendo los datos) genera un pulso de reconocimiento. El dispositivo maestro puede dejar libre el bus generando una condición de parada.



Si se desea seguir transmitiendo, el dispositivo maestro puede generar otra condición de inicio en lugar de una condición de parada. Esta nueva condición de inicio se denomina "inicio reiterado" y se puede emplear para direccionar un dispositivo esclavo diferente o para alterar el estado del bit de lectura/escritura.

Figura 3.4.6 Secuencia deparada del bus I²C.

3.4.2.4 La comunicación en más detalle

Cuando el dispositivo maestro quiere comunicarse con un esclavo, produce una secuencia de inicio en el bus. La secuencia de inicio es una de las dos secuencias especiales que se han definido en el bus I²C; la otra es la secuencia de parada. Las secuencias de inicio y la de parada son especiales porque son los dos únicos casos en que se permite que la línea de datos (SDA) cambie cuando la línea de reloj (SCL) está alta. Cuando se están transmitiendo datos, la línea SDA debe permanecer estable, y jamás cambiar, mientras la línea SCL está alta. Las secuencias de inicio y de parada señalan el comienzo y el final de una transacción con los dispositivos esclavos.

Los datos se transfieren en secuencias de 8 bits. Estos bits se colocan en la línea SDA comenzando por el bit de más peso (o más significativo). Una vez puesto un bit en

SDA, se lleva la línea SCL a alto. Debemos recordar que el chip no puede llevar la línea a un estado alto, en realidad, lo que hace es "soltarla", y el que la pone en nivel lógico alto es la resistencia de polarización. Por cada 8 bits que se transfieren, el dispositivo que recibe el dato envía de regreso un bit de reconocimiento, de modo que en realidad por cada byte de dato se producen 9 pulsos sobre la línea SCL (es decir, 9 pulsos de reloj por cada 8 bits de dato). Si el dispositivo que recibe envía un bit de reconocimiento bajo, indica que ha recibido el dato y que está listo para aceptar otro byte. Si retorna un alto, lo que indica es que no puede recibir más datos y el dispositivo maestro debería terminar la transferencia enviando una secuencia de parada.

3.4.2.5 Direccionamiento de dispositivos en el bus I²C

Lo más común en los dispositivos para el bus I²C es que utilicen direcciones de 7 bits, aunque existen dispositivos de 10 bits. Este último caso es raro. Una dirección de 7 bits implica que se pueden poner hasta 128 dispositivos sobre un bus I²C, ya que un número de 7 bits puede ir desde 0 a 127. Cuando se envían las direcciones de 7 bit, de cualquier modo la transmisión es de 8 bits. El bit extra se utiliza para informarle al dispositivo esclavo si el dispositivo maestro va a escribir o va a leer datos desde él. Si el bit de lectura/escritura (R/W) es cero, el dispositivo maestro está escribiendo en el esclavo. Si el bit es 1 el maestro está leyendo desde el esclavo. La dirección de 7 bit se coloca en los 7 bits más significativos del byte y el bit de lectura/escritura es el bit menos significativo.

3.4.3 Realización física:

La realización del RTC es sencilla:

- Se pone la alimentación en los pines correspondientes, Vcc en el pin 8 y GND en el pin 4.
- Una pila de botón de 3V, tipo *BR3032* en la pata 3 (para conservar la fecha y hora).
- Un cristal de cuarzo de 32,768kHz en los pines 1 y 2.
- Comunicaciones en la 6 con el **SCL** y en la 5 con el **SDA** con unas resistencias de pull-up de valor 1kΩ ya que es mejor para el ruido y los flancos de subida y bajada aunque aumente el consumo.

En la figura 3.4.7 podemos observar la implementación del reloj de tiempo real con el circuito integrado DS1307, la pila de botón para una autonomía propia y poder conservar fecha y hora, el cristal de cuarzo y las resistencias de pull-up.

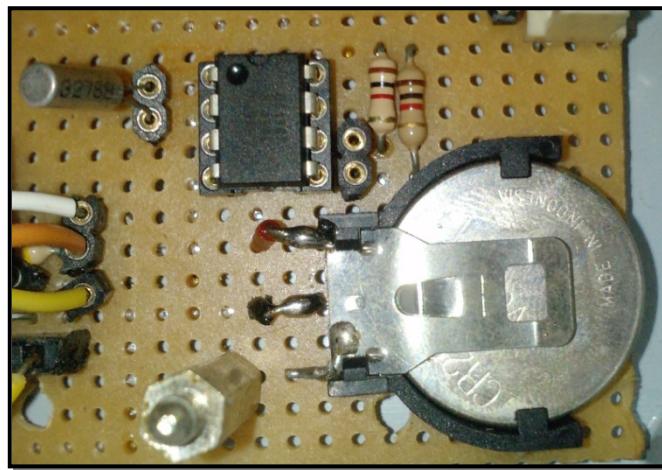


Figura 3.4.7 Realización física de un reloj de tiempo real (RTC).

3.4.4 Programación para el RTC

En el siguiente código podemos observar la función empleado para la obtención de la fecha y hora utilizando el circuito anteriormente descrito. Hay que tener en cuenta que para cargar la fecha y la hora, hay que realizarlo al menos una vez en nuestro circuito. Para ello habrá que implementar un programa fuera de nuestro programa principal a tal efecto.

```
void LeerTime(void){
    Wire.beginTransmission(DirDS1307);
    Wire.send(0x00);
    Wire.endTransmission();
    Wire.requestFrom(DirDS1307, 7);
    for(int i=0; i<7; i++){
        BufTime[i]=Wire.receive();
    }
    ss=(10*((BufTime[0] & DS1307_HI_seg_min)>>4 ))+(BufTime[0] & DS1307_LO);
    mm=(10*((BufTime[1] & DS1307_HI_seg_min)>>4 ))+(BufTime[1] & DS1307_LO);
    hh=(10*((BufTime[2] & DS1307_HI_hor_dia)>>4 ))+(BufTime[2] & DS1307_LO);
    DD=(10*((BufTime[4] & DS1307_HI_hor_dia)>>4 ))+(BufTime[4] & DS1307_LO);
    MM=(10*((BufTime[5] & DS1307_HI_mes)>>4 ))+(BufTime[5] & DS1307_LO);
    AA=(10*((BufTime[6] & DS1307_HI)>>4 ))+(BufTime[6] & DS1307_LO);
}
```

Las acciones que esta función realiza serán las siguientes:

- 1) El maestro, el microcontrolador, genera una secuencia de inicio de la transmisión con la dirección del RTC. Para después generar una condición de parada para que responda el Circuito esclavo al que se ha dirigido, en este caso el RTC.
- 2) El RTC responde poniendo en la línea un byte con los datos del RTC.
- 3) El RTC o dispositivo esclavo le responde para coger el bus y le dice cuantos Byte quiere leer.
- 4) El RTC escribe los 7 Bytes que corresponde con la fecha y hora.
- 5) Sacando la correspondencia a través del datasheet del RTC se adquiere la fecha y la hora en el formato deseado.

3.5 Anemómetro y Veleta

Objetivo: Se desea realizar la captación de las variables relacionadas con el viento, tanto velocidad como dirección.

El viento como se ha visto anteriormente en el capítulo 2 puede ser considerado como la forma principal de climatización pasiva, por lo que en la arquitectura, el viento, es estudiado de diversas maneras. La forma principal es a través de maquetas expuestas a túneles de viento donde se observa mediante simuladores, los obstáculos a los que se enfrenta el movimiento del aire y se determinan las posibles formaciones de turbulencias para mejorar el diseño de las edificaciones. Puesto que la simulación a través de maquetas y túneles del viento no es viable para la realización de este proyecto, el estudio se realiza a través de la adquisición de las variables más importantes, velocidad y dirección.

Así mismo, el viento es considerado un parámetro fundamental en el análisis ambiental de las edificaciones, puesto que puede ser forma de climatización, y también generar sensaciones de malestar en los ocupantes de un espacio e incluso ocasionar problemas de estabilidad en las edificaciones.

Para este objetivo se ha decidido por la adquisición de un anemómetro y una veleta de *inspeed*.

3.5.1 Anemómetro

El anemómetro de *inspeed* consta de 3 copas las cuales giran por la acción del viento sobre un rotor que hace contacto cada vuelta completa del anemómetro. Es un mecanismo sencillo para el cual necesitamos una serie de elementos para poder calcular a la velocidad que gira el anemómetro; y así saber la velocidad del viento.



Figura 3.5.1 Copas del anemómetro de *inspeed*.

Para ello habrá que colocar un resistencia de pull-up entre Vcc y el cable de salida; haciendo esto, cada vuelta de anemómetro generará un pulso cuadrado, una serie de vueltas del anemómetro genera un tren de pulsos cuadrado con una frecuencia determinada. Determinando la relación entre la velocidad del viento y la frecuencia de dicho tren de pulsos, es como se calcula la velocidad del viento.

3.5.1.1 Especificaciones

SENSOR TYPE	3-Cup rotor Reed switch/magnet provide 1 pulse per rotation.
OUTPUT for D2 Rotor (Shown in photo)	1 pulse per rotation 2.5 mph per Hz
ROTOR DIAMETER	approx. 5 in (~125 mm)
SPEED RANGE	approx. 3 mph to 125+ mph (~5 kph to over 200 kph)
MOUNTING	Supplied with an aluminum mounting bracket with 2 holes for screws.
BRACKET	Designed to be mounted on top of a pole or bracket. Custom brackets available up request (offset, for example)
WIRE	Standard length is 25 feet (8m) custom lengths available upon request - tested OK to over 1,500 feet The wire is provided stripped and unterminated 2 small wire nuts provided to connect to the display once installed
DISPLAY	None provided with the sensor only Formula for converting pulses to speed: 2.5 mph per Hz (2.5 mph per pulse/second)
POWER	No power required

Tabla 3.5.1 Especificaciones entregadas por el fabricante.

En la tabla 3.5.1 se muestra las especificaciones tal y como las entrega el fabricante. En estas especificaciones se puede obtener la relación frecuencia-velocidad: del viento, se observa que por cada hercio del tren de pulsos cuadrado generado por el anemómetro equivale a 2.5mph (millas por hora). Pasando dicha equivalencia al sistema internacional se obtiene que la velocidad del viento equivalga a 1.12 m/s por cada hertzio que obtengamos en la salida de nuestro anemómetro. $1.12\text{m/s} = 4.02\text{km/h}$.

Esta salida se introduce en el microcontrolador por una entrada digital para poder captar la frecuencia de la salida del anemómetro y así saber la velocidad del viento.

3.5.1.2 Esquema de funcionamiento del anemómetro

El viento incide en las copas del anemómetro, haciendo girar su mecanismo. Cuando la estructura metálica no hace contacto sobre los cables, como se muestra en la figura 3.5.2, los dos cables están en circuito abierto por lo que el cable de entrada (rojo) se encuentra con la tensión Vcc o un 1 lógico

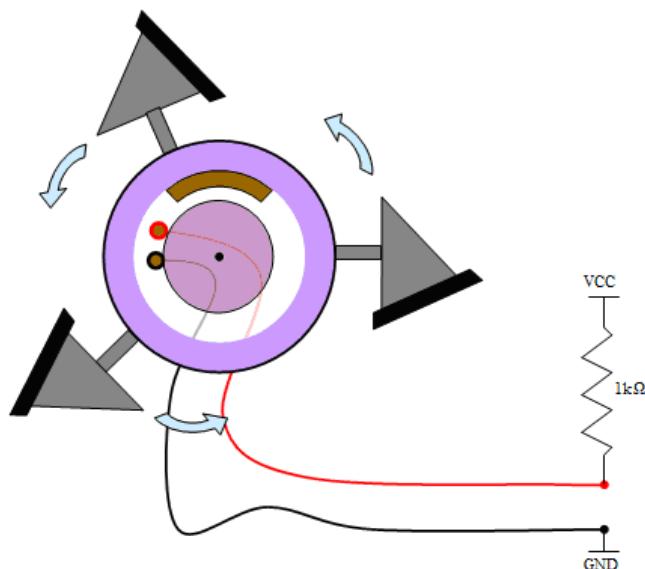


Figura 3.5.2 Esquema de funcionamiento del anemómetro cuando su salida es un 1 lógico.

Cuando el giro de las copas del anemómetro hace que el mecanismo entre en contacto gracias a la estructura, como se muestra en la figura 3.5.3 los dos cables entran en contacto, estos se hallarán en cortocircuito por lo que en la entrada, (cable rojo) tendremos una tensión de 0V ó un 0 lógico.

Este mecanismo hace que si el anemómetro gira un número indeterminado de vueltas en la entrada obtengamos una serie de pulsos cuadrados con una frecuencia determinada.

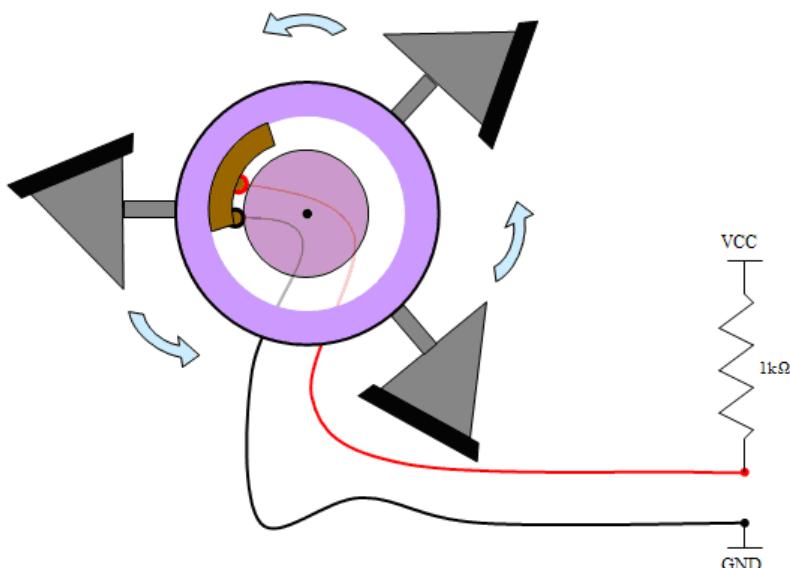


Figura 3.5.3 Esquema de funcionamiento del anemómetro cuando su salida es un 0 lógico.

3.5.1.3 Programación anemómetro

En el siguiente código se observa una serie de funciones que se utilizan para la adquisición de la variable velocidad del viento a partir de la frecuencia de entrada al Arduino:

```
void VelocViento(){
    attachInterrupt(0, IncrementarContador, RISING);
    ContadorPulsos = 0;
    delay(10000);
    detachInterrupt(0);
    Velocidad=ConvertirF_Vel(ContadorPulsos);
    delay(1);
    digitalWrite(clk,HIGH);
    delay(1);
    digitalWrite(clk,LOW);
}

float ConvertirF_W(unsigned long freq){
    float x=0;
    x= float (freq)*0.402;
    return x;
}

void IncrementarContador() {
    ContadorPulsos++;
    return;
}
```

Las acciones que realizan estas funciones son:

- 1) Activa las interrupciones del pin 3 de tal forma que cada vez que en el pin 3 se detecte un flanco de subida se ejecute la función `IncrementarContador`.
- 2) Se espera 10 segundos para saber cuantos flancos de subida hay en esos 10 segundos y así saber la frecuencia del tren de pulsos cuadrado.
- 3) Se pasa la frecuencia obtenida a velocidad del viento, km/h, con la relación obtenida previamente.
- 4) Se incrementa el contador del reloj de la lógica para seleccionar otro sensor que genere tren de pulsos cuadrados.

3.5.2 Veleta

La veleta que se utiliza para medir la dirección del viento es de la marca “inspeed”. Dicha veleta gira en el sentido en el que sopla el viento, internamente consta de un potenciómetro capaz de girar 360° por lo que la señal de salida será una variable comprendida entre los parámetros de entradas; dicho potenciómetro no consta de topes por lo que puede hacer vueltas completas sin problemas.

3.5.2.1 Especificaciones



Figura 3.5.4 Veleta de Inspeed.

SENSOR TYPE	Balanced wind vane connected to an active, non-contact, zero friction Hall Effect sensor Sealed magnetic Hall Effect sensor. Magnet hovers over the sensor to provide ~0-5VDC output
SENSOR RANGE	Full 360 degrees , zero deadband
ACCURACY/LINEARITY	+/-0.3 to 0.5% of signal range
RESOLUTION	12 bit or 0.025 degrees
ELECTRICAL	3 wire flying leads. Supply voltage 4.5 to 5.5 VDC Current 15 mA typical Output 5% to 95% of input voltage (0.25 to 4.75VDC) Length of wire: optional (standard lengths provided - click to order to view options)
MOUNTING	The E-Vane is provided with an offset aluminum bracket with 2 mounting holes. The mounting is compatible with Inspeed Vortex Wind Sensors (the 2 holes match).
DIMENSIONS	Directional Vane approximately 8 inches
COMPATIBILITY	The Inspeed e-Vane is compatible with Inspeed WindWorks It is not compatible with Inspeed Windware software (which provides wind speed only)

Tabla 3.5.2 Especificaciones de la veleta dadas por el fabricante.

3.5.2.2 Esquema eléctrico

La veleta consta de 2 cables de alimentación y un cable que es la señal de salida. Dependiendo de la posición de la veleta hace girar el potenciómetro interno, por lo que la señal de salida variará entre GND y Vcc. Habrá que tener en cuenta donde se encuentra el sur y saber cual es su señal y tenerla como referencia.

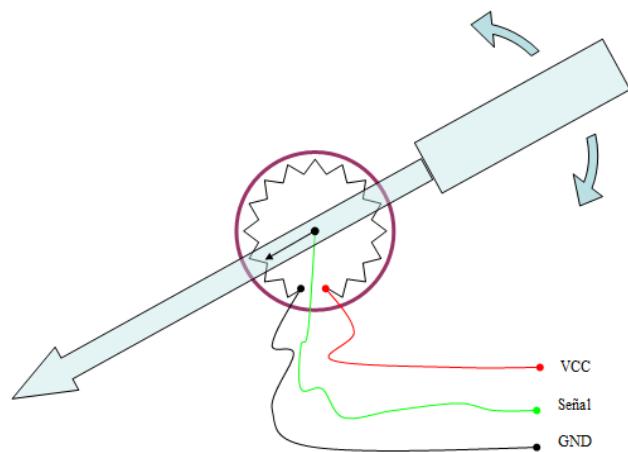


Figura 3.5.5 Esquema eléctrico de la veleta de Inspeed

3.5.2.3 Programación Veleta

A continuación se muestra la función generada para poder obtener los grados de variación de la veleta con respecto al sur a partir de la señal analógica de entrada:

```
void TomarDire(){
    float voltios;
    voltios=(float(analogRead(pinDire)))*5/1024;
    Direccion=360*(voltios-1.53)/4.74;
    if (Direccion > 180){Direccion=Direccion-360 ; }
}
```

Esta función realizará las siguientes acciones:

- 1) Toma el valor del pin de dirección del viento a través del convertidor analógico-digital.
- 2) Con ese valor obtenemos los voltios de salida del sensor.
- 3) Con los datos de las especificaciones, se pasa de voltios a la dirección de la veleta.

3.5.3 Montaje

En este apartado se muestran diferentes fotografías de cómo quedan tanto la veleta como el anemómetro físicamente. Ver figura 3.5.6 y 3.5.7.



Figura 3.5.6 Fotografía de la veleta.



Figura 3.5.7 Fotografía anemómetro.

3.6 Piranómetro

Objetivo: Se desea realizar un Piranómetro para medir la potencia lumínica del sol (Radiación Solar), en cualquier momento y además saber la posición solar, Además con una serie de piranómetros se precisa realizar un mapa de sombras para conocer los elementos físicos que generan sombras a lo largo de un día de una determinada zona., siempre bajo la premisa de un coste reducido.

La radiación solar es un parámetro ambiental que varía constantemente, a lo largo del día, de diferentes meses, dependiendo de las estaciones y del año. El análisis de un piranómetro permite determinar la cantidad de energía que llega a una superficie. Puede ser de manera directa, difusa o reflejada según la inclinación de la radiación solar con el piranómetro. Su estudio es necesario, pues la radiación produce un incremento de la temperatura en las superficies envolventes, que luego desprenden ese calor al interior de las edificaciones y genera movimientos de masas de aire por diferencias de temperatura entre las zonas expuestas al sol y las que están a la sombra.

El mapa de sombras es una representación gráfica de las diferentes sombras que se producen en una determinada zona por elementos físicos, ya sean vegetación, edificaciones cercanas o cualquier obstáculo que pueda producir sombras. Es muy importante determinar a qué hora se producen y en qué día, puesto que las sombras dependerán de la posición solar.

Este análisis debe tenerse muy presente tanto en el diseño de una obra nueva como en la evaluación del comportamiento de una edificación y su posterior acondicionamiento. Como se puede notar, la incidencia de la radiación solar determina la ubicación y posición de las aberturas y de las partes ciegas, de los elementos de protección, de los captadores solares, etc.

3.6.1 TAOS TSL230R

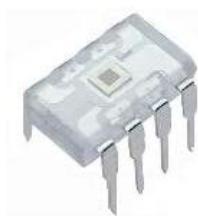


Figura 3.6.1 Fotodiodo TSL230.

Información extraída de [10]. Los Convertidores Programables Luz-Frecuencia, TSL230, TSL230A y TSL230B combinan un fotodiodo de silicio configurable y un convertidor de corriente a frecuencia en un solo circuito integrado monolítico CMOS. El de salida puede ser un tren de pulsos o una onda cuadrada (50% ciclo de trabajo) con una frecuencia directamente proporcional a la intensidad de la luz.

La sensibilidad del dispositivo se puede seleccionar en tres rangos, proporcionando dos dígitos de ajuste. La frecuencia de salida se puede dividir por uno de los cuatro valores predefinidos para poder tener frecuencias de trabajo más reducidas si se requiere. Todas las entradas y las salidas son TTL compatible, que permite la comunicación directa de dos vías con un microcontrolador para la programación y la interfaz de salida.

Posee una entrada para habilitar el dispositivo, la entrada OE, en caso de que se active, coloca la salida en un estado de alta impedancia para el intercambio de múltiples unidades con un microcontrolador con la misma línea de entrada.

Los dispositivos están disponibles con diferentes tolerancias: $\pm 5\%$ (TSL230B), $\pm 10\%$ (TSL230A), o de un $\pm 20\%$ (TSL230). Se ha compensado la temperatura para el rango ultravioleta a la luz visible de 300nm a 700nm y se caracterizan por tener un rango de operación de temperatura de -25°C a 70°C .

3.6.1.1 Configuración de pines TAOS TSL230

TERMINAL NOMBRE	I/O	DESCRIPCIÓN
GND 4		Tierra
OE 3	I	Habilitar f_O (activo nivel bajo)
OUT 6	O	Frecuencia de salida (f_O)
S0, S1 1, 2	I	Entradas selección sensibilidad
S2, S3 7, 8	I	Entradas selección escalado f_O
VDD 5		Alimentación

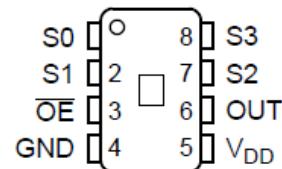


Figura 3.6.2 Esquema de pines TAOS TSL230.

Tabla 3.6.1 Esquema de pines TAOS TSL230.

En la tabla 3.6.2 se puede observar los diferentes posibles valores de la sensibilidad y del escalado de la frecuencia. Para el propósito del proyecto se ha elegido una sensibilidad de 1x y un escalado de frecuencia de 1x. La elección se explicará más adelante.

S1	S0	Sensibilidad
L	L	Power Down
L	H	1x
H	L	10x
H	H	100x

S3	S3	Escalado de f_O (dividido por)
L	L	1
L	H	2
H	L	10
H	H	100

Tabla 3.6.2 Sensibilidad y escalado de frecuencia TAOS TSL230.

3.6.1.2 Características de funcionamiento (a VDD = 5 V, TA = 25° C)

En la tabla 3.6.3 se observan las características de funcionamiento, estas características son importantes para entender el funcionamiento del fotodiodo.

Parámetros	Condiciones test	TSL230			TSL230A			TSL230B			Unidad								
		MIN	TIP	MAX	MIN	TIP	MAX	MIN	TIP	MAX									
f_o Frecuencia de salida	$S_0 = H$, $S_1 = S_2 = S_3 = L$, $E_e = 130 \text{ mW/cm}^2$, $\lambda_p = 670 \text{ nm}$	0,8	1	1,2	0,9	1	1,1	0,95	1	1,05	MHz								
	$E_e = 0$, $S_0 = H$, $S_1 = S_2 = S_3 = L$	0,1	10		0,1	10		0,1	10		Hz								
	$S_1 = H$, $S_0 = S_2 = S_3 = L$, $E_e = 13 \text{ mW/cm}^2$, $\lambda_p = 670 \text{ nm}$	0,8	1	1,2	0,9	1	1,1	0,95	1	1,05	MHz								
	$E_e = 0$ $S_1 = H$, $S_0 = S_2 = S_3 = L$	0,13	10		0,13	10		0,13	10		Hz								
	$S_0 = S_1 = H$, $S_2 = S_3 = L$, $E_e = 1,3 \text{ mW/cm}^2$, $\lambda_p = 670 \text{ nm}$	0,8	1	1,2	0,9	1	1,1	0,95	1	1,05	MHz								
	$E_e = 0$, $S_0 = S_1 = H$, $S_2 = S_3 = L$	0,5	10		0,5	10		0,5	10		Hz								
t_w Duración pulso salida	$S_2 = S_3 = L$	125	550		125	550		125	550		ns								
	$S_2 = S_3 = H$	1/2 f_o			1/2 f_o			1/2 f_o			s								
No linealidad	$f_o = 0 \text{ MHz to } 10 \text{ kHz}$	$\pm 0,1\%$		$\pm 0,1\%$		$\pm 0,1\%$		$\pm 0,1\%$		%F.S.									
	$f_o = 0 \text{ MHz to } 10 \text{ kHz}$	$\pm 0,2\%$		$\pm 0,2\%$		$\pm 0,2\%$		$\pm 0,2\%$		%F.S.									
	$f_o = 0 \text{ MHz to } 10 \text{ kHz}$	$\pm 0,5\%$		$\pm 0,5\%$		$\pm 0,5\%$		$\pm 0,5\%$		%F.S.									
Recuperación de apagado		100		100		100		100		μs									
Respuesta paso de full-scale a entrada		1 impulso de la nueva frecuencia, más 1 μs																	
Tiempo de respuesta a Cambio programado		2 períodos de nueva frecuencia principal más 1 μs																	
Tiempo respuesta a habilitar la salida (OE)		150	50		50	150		50	150		ns								

Tabla 3.6.3 Diferentes características de funcionamiento del fotodiodo TSL230.

3.6.1.3 Características eléctricas (a 25°C y V_{DD} = 5V)

PARAMETROS	CONDICIÓN TEST	MIN	TIP	MAX	UNIDAD
V _{OH} Voltaje de salida nivel alto	I _{OH} = -4mA	4	4,3		V
V _{OL} Voltaje de salida nivel bajo	I _{OL} =4mA		0,17	0,26	V
I _{IH} Corriente máxima de entrada			1		μA
I _{IL} Corriente mínima de entrada			1		μA
I _{DD} Corriente suministrada	Modo Power-on	2	3		mA
	Modo Power-Down		10		μA
Frecuencia a escala máxima		1.1			MHz
Coeficiente de temperatura de la frecuencia de salida	λ≤ 700nm, -25°C≤ TA ≤70°C	± 100			ppm/°C
kSVS Sensibilidad fuente alimentación	VDD = 5 V ±10%	0,5			%/V

Tabla 3.6.4 Diferentes características eléctricas del fotodiodo TLS230.

3.6.1.4 Las curvas características TSL230

- **Respuesta espectral de fotodiodo**

En la figura 3.6.3 se puede observar la respuesta espectral del fotodiodo. Dicha respuesta viene determinada por la longitud de onda, λ , que viene expresada en nm.

Habrá que tener en cuenta a la hora de diseñar nuestro sensor que la respuesta espectral de la luz solar tiene una anchura superior a la de la respuesta del fotodiodo.

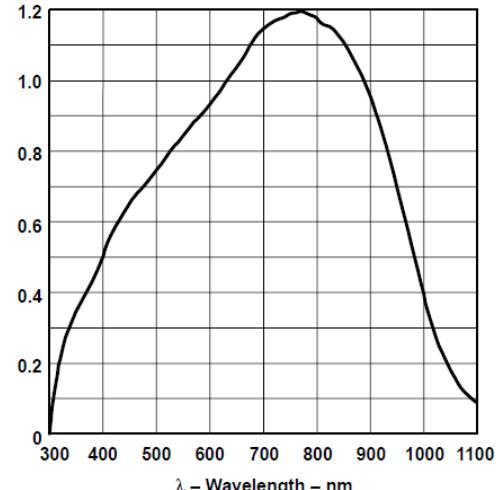


Figura 3.6.3 Respuesta espectral del fotodiodo.

- **Coeficiente de temperatura de la frecuencia de salida**

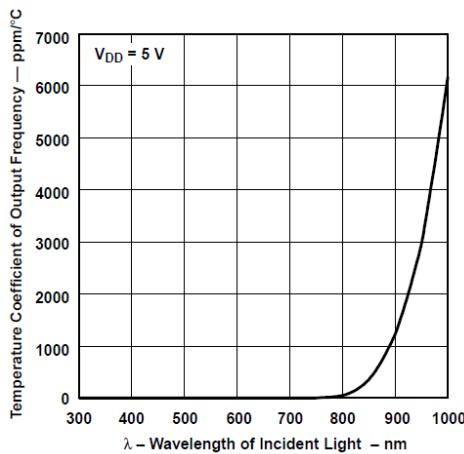


Figura 3.6.4 Coeficiente de temperatura de la frecuencia de salida.

En la gráfica 3.6.4 se puede observar el coeficiente de temperatura de la frecuencia de salida. Dicho coeficiente viene expresado en ppm (partes por millón), esto hay que tenerlo en cuenta debido al exceso de calor que puede recibir el fotodiodo.

$$TC(V_0) = 10^6 \cdot \frac{\Delta V_0/V_0}{\Delta T} (ppm/^\circ C)$$

- **Frecuencia de salida en función de la intensidad irradiada al fotodiodo**

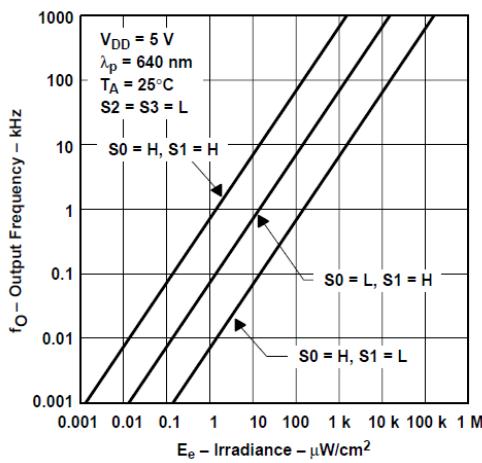


Figura 3.6.5 Frecuencia de salida en función de la intensidad irradiada.

La curva que se ha escogido es la de máxima captación de la radiación, $S0=H$ y $S1=L$, debido a que es la única que puede medir toda la potencia solar, ya que esta puede llegar a una intensidad de 1000W/m^2 aproximadamente. Además se ha elegido el escalado de frecuencia 1x, es decir $S2=L$ y $S3=L$ para obtener una mejor medición de la frecuencia, ya que un escalado mayor puede hacer una medición errónea.

3.6.2 Espectro Solar

El espectro solar es necesario para hacer el estudio teórico de la radiación solar que va a recibir el fotodiodo. Puesto que la radiación solar tiene un espectro solar determinado y el fotodiodo tiene una respuestapectral en función de la longitud de onda, se necesita saber el espectro solar teórico.

El espectro solar ha sido adquirido de ASTM G-173, como se puede observar en la figura 3.6.6, muestra la radiación solar extraterrestre, en la capa más alta de la atmósfera (Etr) y la radiación Global del sol a nivel del mar.

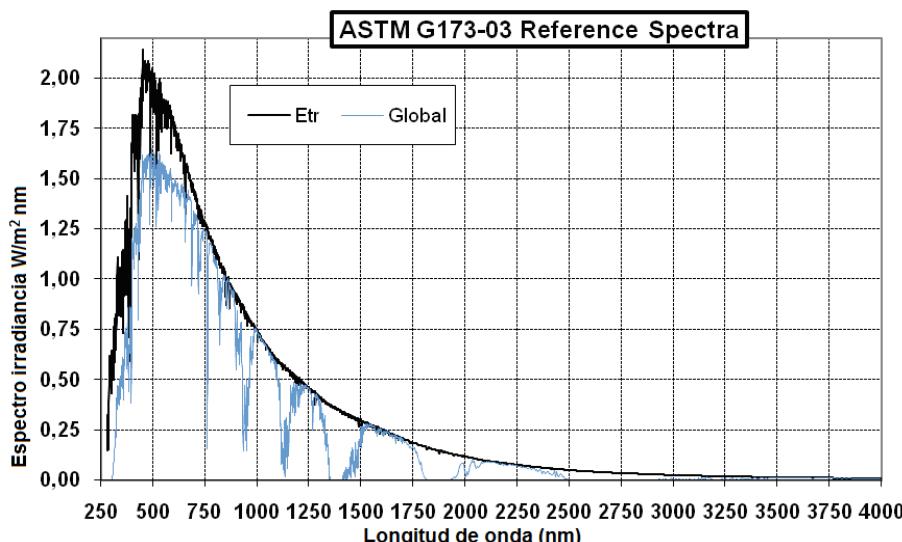


Figura 3.6.6 Respuesta espectral solar a nivel del mar y en la atmósfera.

En la figura 3.6.5 se puede observar la frecuencia de salida del fotodiodo en función de la intensidad irradiada.

La intensidad irradiada viene reflejada en $\mu\text{W/cm}^2$. Habrá que tener en cuenta que la intensidad solar, que es la medida que se va a realizar, se expresa en W/m^2 , por lo que $1\text{W/m}^2 = 100\mu\text{W/cm}^2$.

La radiación solar Etr es el conjunto de radiaciones electromagnéticas emitidas por el sol. Puede considerarse como la radiación emitida por un cuerpo negro a una temperatura de 6000° K.

Ley de planck:

La intensidad de la radiación emitida por un cuerpo negro con una temperatura T viene dada por:

$$I(f, T) = \frac{2hf^3}{c^2} \frac{1}{e^{\frac{hf}{kT}} - 1}$$

Donde $I(f)$ δf es la cantidad de energía por unidad de área, unidad de tiempo y unidad de ángulo sólido emitida en el rango de frecuencias entre f y $f+\delta f$.

La Radiación solar Global o Total es la radiación solar captada en la superficie terrestre. Hay que diferenciar la radiación solar total en la superficie terrestre de la radiación global a una superficie perpendicular a la radiación, puesto que la radiación perpendicular será superior a la radiación global en la superficie terrestre.

Dicha Radiación consta de dos componentes, la radiación directa y la difusa:

- La directa es la radiación solar que incide perpendicularmente a la superficie de la tierra.
- La difusa es la radiación que ha sido reflejada y difundida por la atmósfera.

La diferencia entre la radiación Etr y global se debe a los procesos atmosféricos que sufre la luz al pasar la atmósfera. El scattering es la razón principal de la pérdida de potencia, siendo el O₂, el O₃ y el H₂O los responsables fundamentales de esta pérdida de potencia.

La gráfica de la figura 3.6.6 consta de un total de 2000 puntos de adquisición de datos. Para hallar la potencia total de ambas señales se calcula con Matlab, siendo el resultado de 1358 W/m² para Etr y de 1002 W/m² para Global.

Habrá que tenerse en cuenta que el cálculo de la radiación solar global es una estimación media de la radiación solar global a lo largo del año para días despejados. La radiación instantánea podría ser superior a ese valor aunque nunca podrá superar los 1358 W/m² de la radiación solar global en la atmósfera.

3.6.3 Radiación solar partiendo de la frecuencia del fotodiodo

En primer lugar habrá que tener en cuenta que el dato de salida del fotodiodo dependerá de la función de transferencia del fotodiodo y de la temperatura ambiente, por lo que habrá que hacer una estimación de dicho parámetro en función de estas variables.

Calculamos la atenuación que sufre la señal producida por dichas variables. Para ello se ha realizado un programa en Matlab para realizar dicho proceso y se ha calculado la potencia que interpreta el fotodiodo en función de la potencia recibida.

Paso 1: Se inserta la radiación solar global en Matlab y se calcula la potencia de la radiación, para ello se cogen los puntos suministrados por de ASTM G-173 y se calcula el valor de la potencia total. Ver figura 3.6.7.

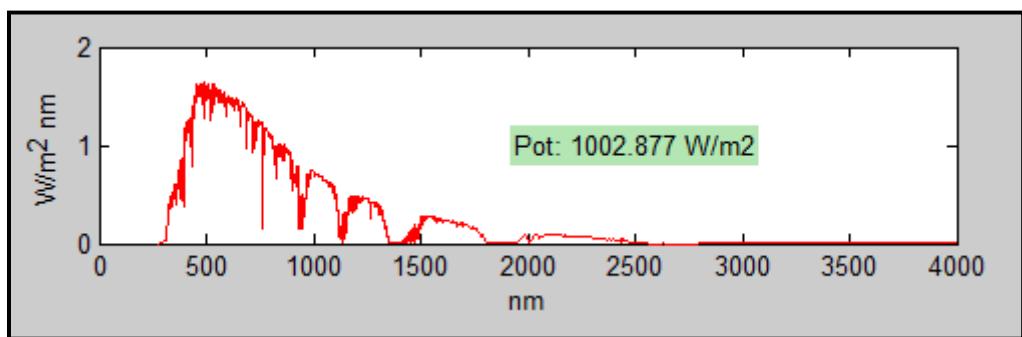


Figura 3.6.7 Función de transferencia de la radiación solar.

Paso 2: Como el fabricante no da los puntos de la función de transferencia del fotodiodo, se intenta captar con la mayor precisión posible una evaluación de los posibles valores de la grafica en el mayor numero de longitudes de ondas posibles. En este caso se ha realizado la obtención de 100 puntos para así poder reconstruir una gráfica en Matlab. Al ser este número de puntos un número bastante inferior al de la función de transferencia de la radiación global hay que interpolar dichos puntos para conseguir el mismo número de puntos en ambas gráficas para así poder operar entre ellas. Para dicha interpolación se ha elegido la interpolación “*interp1*”. Ver figura 3.6.8.

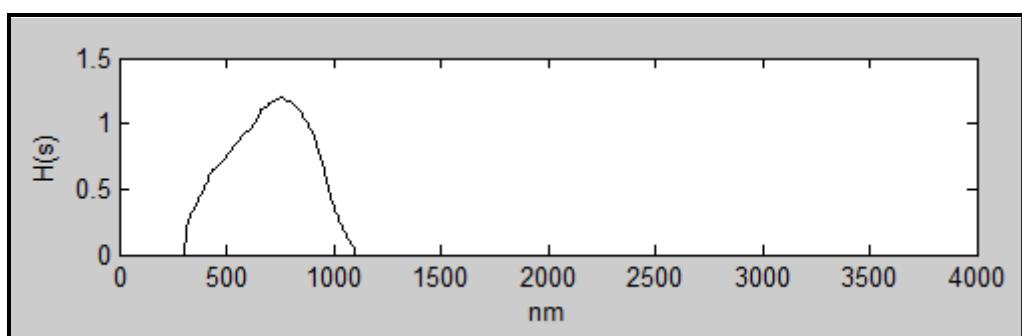


Figura 3.6.8 Función de transferencia del fotodiodo.

Paso 3: Se realiza el mismo proceso que en el paso 2 con el coeficiente de temperatura de la frecuencia de salida. Aunque en este caso se deja una variable a determinar que es la temperatura. Además, también se interpola con “interp1” igual que la función de transferencia. En este caso hemos elegido 30°C.

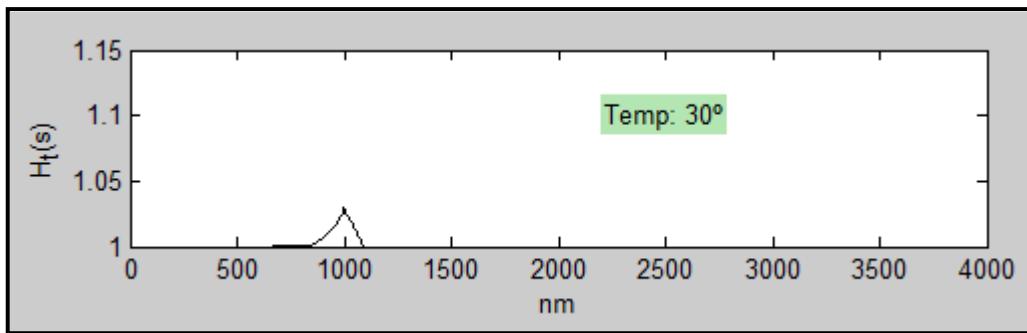


Figura 3.6.9 Función de transferencia del coeficiente de temperatura.

Paso 4: Se calcula la grafica resultante de pasar la radiación solar global por cada filtro y se calcula la potencia resultante.

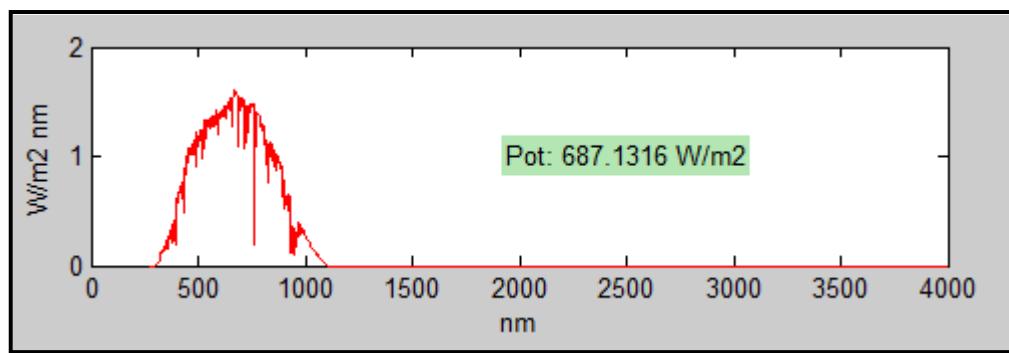


Figura 3.6.10 Función de transferencia del coeficiente de temperatura.

Como resultado vemos que la potencia se reduce un 68'5%. Para completar el cálculo de la reducción de potencia que sufre la radiación solar hacemos el mismo proceso para valores extremos de temperatura a -10°C y a 80°C y vemos el margen de la reducción. En dichos márgenes la potencia se reduce a un 67'8% y un 69'6% respectivamente, por lo que el margen de error producido por la temperatura será menor de un 1'5%.

Después, para calcular la potencia de salida del fotodiodo en función de la frecuencia, habrá que tener en cuenta que siempre que pongamos un fotodiodo tendremos que poner una estructura que lo aíslle de las inclemencias del tiempo, por lo que para poder poner un fotodiodo habrá que cubrirlo con algún tipo de material para exponerlo al aire libre. El material utilizado es polimetacrilato, también llamado como Polimetilmetacrilato ó por sus siglas PMMA.

Se ha elegido este material por sus diferentes propiedades, las más relevantes para su uso en cuestión son:

- Transparencia de alrededor del 93%. El más transparente de los plásticos.
- Alta resistencia al impacto, de unas 10 a 20 veces la del vidrio.
- Resistente a la intemperie y a los rayos ultravioleta. No hay un envejecimiento apreciable en 10 años de exposición exterior.
- Excelente aislante térmico y acústico.
- Gran facilidad de mecanización y moldeo.

Tras diferentes pruebas realizadas sobre este material y tras numerosas mediciones se llega a la conclusión que el polimetacrilato atenúa la potencia solar en un 7% aproximadamente con un margen de error de un 2%.

Por último tenemos que ver la relación existente entre la frecuencia y la radiación solar. En la grafica 3.6.4 se observa ésta relación.

Para una lectura más fina se puede recurrir al datasheet donde se puede observar que dicha relación es de $1\text{Hz}=0.0013\text{W/m}^2$.

Por lo que la relación final sería de $1\text{Hz}=0.0013*1.46*1.11=0.0021\text{W/m}^2$

3.6.4 Programación TAOS TLS 230

A continuación se puede observar el código que se ha desarrollado para obtener la potencia solar a partir del tren de pulsos cuadrados entregados por el fotodiodo TAOS TSL230.

```

float TomarRadiacion(){
    float watioLocal;
    attachInterrupt(1, IncrementarContador, RISING);
    ContadorPulsos = 0;
    delay(1000);
    detachInterrupt(0);
    watioLocal=ConvertirF_W(ContadorPulsos);
    delay(1);
    digitalWrite(clk,HIGH);
    delay(1);
    digitalWrite(clk,LOW);
    return watioLocal;
}

void IncrementarContador() {
    ContadorPulsos++;
    return;
}
float ConvertirF_W(unsigned long frec){
    float x=0;
    x= float(frec)* *0.0021;
    return x;
}

```

Estas funciones realizan las siguientes acciones:

- 1) Activa las interrupciones del pin 3 de tal forma que cada vez que en el pin 3 se detecte un flanco de subida se ejecute la función `IncrementarContador`.
- 2) Se espera un segundo para saber cuantos flancos de subida hay en ese segundo y así saber la frecuencia del tren de pulsos cuadrado.
- 3) Se pasa la frecuencia obtenida a radiación solar, W/m^2 , con la relación obtenida previamente.
- 4) Se incrementa la señal de reloj utilizada en la lógica para seleccionar los diferentes sensores del mismo tipo, que genere tren de pulsos cuadrados.

3.6.5 Realización física piranómetro

Un piranómetro consiste en un sensor que mide la radiación solar global, tanto la componente directa como la componente difusa. Para hacer un sensor con un fotodiodo habría que tener dicho fotodiodo perpendicular a los rayos de sol en todo momento, para poder captar esa componente, como esta solución no es viable se ha optado por otro tipo de estrategia.

Para poder saber la radiación solar global se realiza una medición en las 3 componentes vectoriales (x , y , z) para luego poder sacar el módulo de ese vector y saber la radiación solar global perpendicular. Como se observa en la figura 3.6.11 sabiendo la componente a_x , a_y y a_z se puede calcular el módulo de ese vector (a).

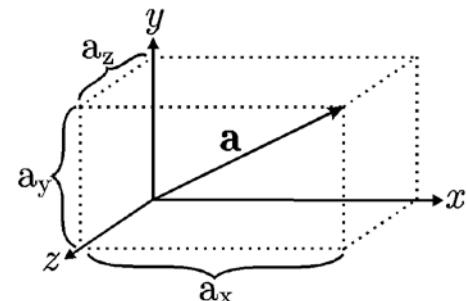


Figura 3.6.11 Vector tridimensional.

Para poder hacer una representación total del espacio vectorial habrá que colocar los fotodiodos en posición para que nos de las tres componentes vectoriales, por lo que hay que poner un fotodiodo perpendicular al sur, ya que el sol siempre está en el sur en el hemisferio norte, otro fotodiodo perpendicular al cenit, y nos hará falta otros 2 fotodiodos, uno perpendicular al este y otro perpendicular al oeste, dependiendo el momento horario. Además se puede considerar poner otro fotodiodo perpendicular al norte, para captar la radiación solar difusa, ya que a ese fotodiodo jamás le darán los rayos solares de forma directa, sólo captará la radiación difusa.

Considerando todas las posibles soluciones se ha optado por realizar un cubo en el que todas sus paredes menos una (la base), tendrán un fotodiodo para poder calcular la radiación solar global, la radiación directa y la difusa. En la figura 3.6.12 se puede observar la realización de dicho cubo y como en cada lado de este se ha colocado un fotodiodo para poder medir la radiación solar directa perpendicular a esa posición.

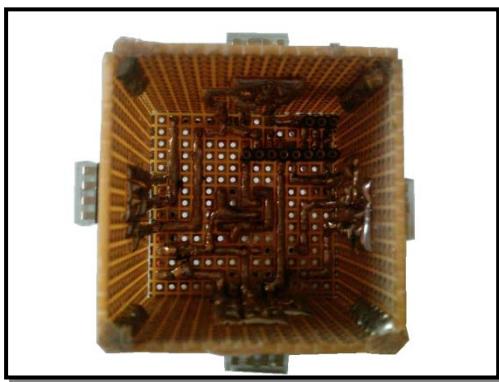


Figura 3.6.13 Cubo de fotodiodos, vista desde abajo.

Además para poder captar a la temperatura que se realizan las mediciones se ha colocado un sensor de temperatura MAXIM DS18B20, se puede ver el funcionamiento en el próximo apartado, para poder tener en cuenta la temperatura para el coeficiente de temperatura de la frecuencia de salida del fotodiodo.

En la figura 3.6.145 se muestra el cubo de fotodiodos en la cubierta de metacrilato, cubierta que debe estar herméticamente cerrada y sólo disponer de una pequeña abertura por la parte inferior para introducir los cables que llegan al piranómetro.

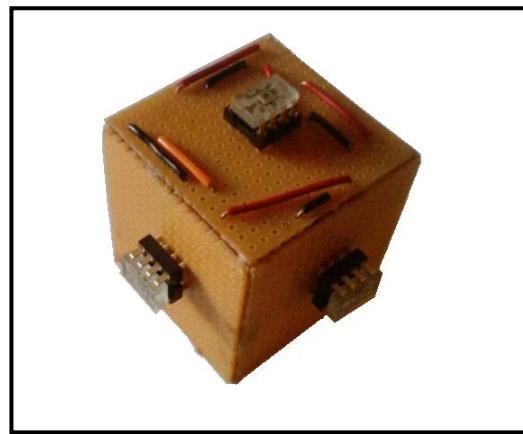


Figura 3.6.12 Cubo de fotodiodos, vista desde arriba.

En la figura 3.6.13 se observa como se ha soldado todas las pistas del cubo por dentro de este para intentar que hubiera el menor número de componentes externos al cubo que pudiera reflejar o atenuar la radiación solar y como consecuencia alterar los datos adquiridos.

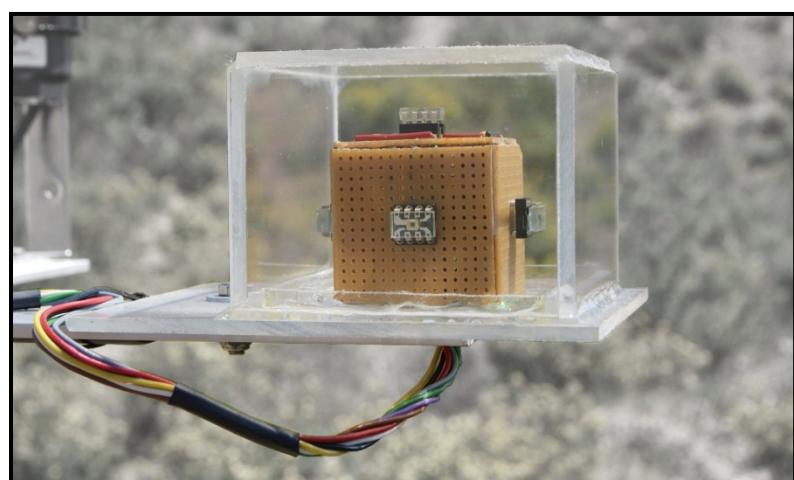


Figura 3.6.14 Piranómetro con la cubierta de metacrilato.

En la figura 3.6.15 Se muestra como queda el piranómetro en la estructura para soportar todo el sistema



Figura 3.6.15 Piranómetro.

3.6.6 Mapa de sombras

El mapa de sombras es un mapa o gráfica donde se muestra la irradiancia solar recibida por los diferentes puntos de una zona concreta, se marcarán los puntos donde la potencia solar es menor al resto, eso indica que se produce punto de sombra. Como no se puede captar la potencia solar recibida por todos los puntos, se escogen unos puntos concretos del área a cubrir y se extrapolan los resultados al resto, dando un valor aproximado de lo que ocurre globalmente en la zona a estudiar en un periodo de tiempo.

Los 6 puntos de adquisición de datos son 5 sensores de captación de potencia solar, realizados con fotodiodos TAOS TLS230 y el fotodiodo de la cara superior del piranómetro. Se desea cubrir un área de 20x30 metros aproximadamente, por lo que primero habrá que determinar la distancia máxima del sensor al microcontrolador. La distancia vendrá determinada por la atenuación del cable. Para calcular la atenuación del cable se necesita una serie de parámetros.

Primero hay que ver cual es la potencia mínima de salida del fotodiodo, en la tabla 3.6.4 se observa que el voltaje mínimo para nivel alto es de 4V. En el datasheet del ATmega se observa que el voltaje mínimo para un pin del microcontrolador es de 2,4V y la corriente proporcionada por un pin del microcontrolador es 40mA como se observa en la tabla 3.1.1. Con estos datos se observa que la atenuación máxima soportada por el cable que comunica el fotodiodo con el microcontrolador es 1,6V. Este es el dato más restrictivo puesto que tanto la alimentación del fotodiodo como el voltaje en nivel bajo son menos restrictivos que el nivel alto.

Si el voltaje máximo de atenuación es de 1,6V y la intensidad de comunicaciones es de 40mA entonces la resistencia máxima del cable ha de ser 40Ω .

La formula de la resistencia del cable es:

$$R = \rho \frac{L}{s}$$

ρ es el coeficiente de resistividad.
L la longitud del cable.
s la superficie de la sección del cable.

El cable tiene un diámetro 0,2mm y es de cobre, por lo que la superficie es 0,031416mm² y ρ es igual a 0,0172 Ω mm² /m. Entonces la resistencia será de 0,5475 L. Entonces la longitud máxima del cable ha de ser menor a 73,06m, para compensar los redondeos la longitud máxima será de 70m.

En el caso específico de este proyecto ha de cubrirse una superficie de 600m², puesto que la superficie es de 20x30 metros. Como se disponen de 6 puntos de captación, cada uno representará un área de 100m², es decir un área de 10x10 m. La longitud máxima que se precisa es de 14,14m, distancia muy inferior a la máxima permitida. En la figura 3.6.16 la disposición de los diferentes sensores de radiación solar para poder cubrir la zona en cuestión

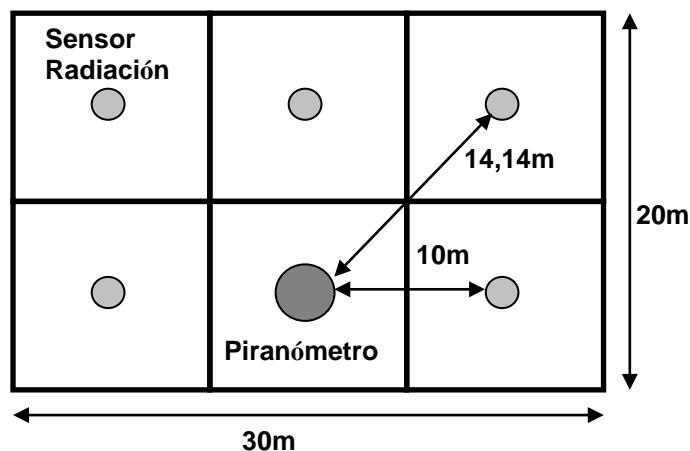


Figura 3.6.16 Mapa de zona.

Los cinco fotodiodos, van encapsulados en pequeñas cajas como se observa en la figura 3.6.78, la cara superior debe ser de metacrilato para que nos sirva el estudio previo realizado sobre los fotodiodos TAOS TLS230. En la figura 3.6.17 se puede observar los 5 fotodiodos necesarios para hacer el mapa de sombras.



Figura 3.6.17 Fotodiodo encapsulado.

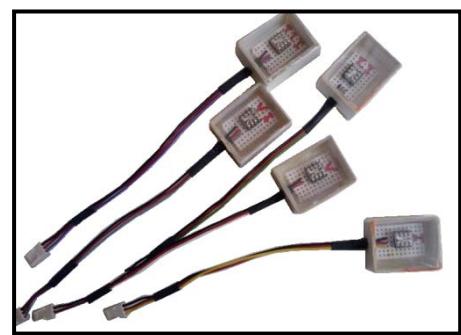


Figura 3.6.18 Fotodiodos necesarios para realizar mapa de sombras.

3.7 Sensor de temperatura y Sensor de Humedad

Objetivo: Realizar la captación de la temperatura y de la humedad ambiente.

La Temperatura es probablemente uno de los parámetros fundamentales del clima. Aunque básicamente consiste en el estado relativo de calor o frío, la temperatura está condicionada por otros parámetros y factores como la radiación solar, el viento, la composición y naturaleza de las superficies, la situación geográfica, la altura sobre el nivel del mar, el factor de continentalidad, el relieve, la vegetación, la topografía y algunas condiciones meteorológicas. Asimismo, hay que señalar que su valor influye, a su vez, en la evaporación, radiación y movimiento del aire a través de sus variaciones diarias y estacionales.

Desde el punto de vista arquitectónico resulta fundamental en el análisis del comportamiento de las edificaciones, ya que junto con los resultados obtenidos de otros parámetros podríamos determinar si se ofrecen o no unas condiciones climáticas de confort, al mismo tiempo que determina, en gran medida, el sistema constructivo a utilizar y las medidas correctoras en el reacondicionamiento.

La humedad es entendida como la cantidad de vapor de agua que contiene el aire, como resultado de la evaporación de las masas de agua producto del calentamiento generado por la radiación solar y la evapotranspiración animal y vegetal. Este valor varía de acuerdo al tiempo y lugar y, junto con la temperatura, es fundamental para determinar el clima de un sitio. No obstante, no se utilizan frecuentemente como valor absoluto en el diseño ni en las edificaciones. Generalmente, lo que se toma en consideración es el valor de la humedad relativa que, aunque es de tipo macroclimático puede modificarse debido a las variaciones microclimáticas.

Por otra parte hay que señalar que este elemento de gran importancia tanto en el diseño como en el planteamiento de medidas correctoras, ya que junto con la temperatura del aire y el movimiento del aire puede incidir directamente en las condiciones de confort y, especialmente, en la sensación de temperatura.

3.7.1 Sensor de Temperatura. MAXIM DS18B20

Para realizar el sensor de temperatura se ha optado por el sensor de Maxim DS18B20. La información y los datos de este apartado han sido extraídos de [11].

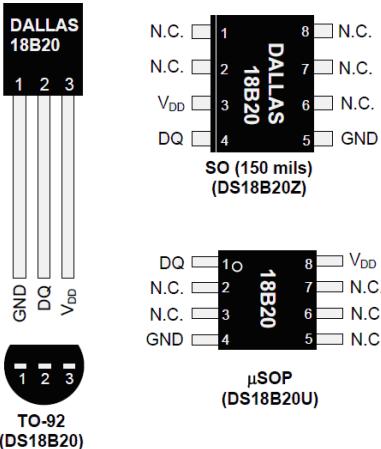


Figura 3.7.1 Sensor DS18B20.

3.7.1.1 Características

- Interfaz 1-Wire, requiere sólo un pin para las comunicaciones.
- Cada dispositivo tiene un código de serie único de 64 bits Almacenados en una ROM.
- Capacidad de uso multipunto, simplifica aplicaciones de detecciones de temperaturas distribuidas.
- No requiere componentes externos.
- Puede ser alimentado desde la línea de datos. El rango de alimentación será entre 3.0V y 5.5V.
- Temperaturas Medidas de -55°C a $+125^{\circ}\text{C}$ (-67°F a 257°F).
- $\pm 0,5^{\circ}\text{C}$ de precisión en el rango de -10°C a $+85^{\circ}\text{C}$.
- La resolución es seleccionable por el usuario de 9 a 12 bits.
- Convierte la temperatura a un valor digital de 12 bits 750 ms (máx.).
- Alarma configurable, no volátil configurable por el usuario.
- Identifica en el registro el comando alarma y código de serie del dispositivo de los sensores cuya temperatura esta fuera de los límites programados (condición de Alarma).
- Disponible en formato 8 pines SO (150 milésimas), 8-Pin μ SOP, y 3-Pin A-92.
- Software compatible con el DS1822.
- Aplicaciones incluidas en controles termostáticos, Sistemas Industriales, Productos de Consumo, Termómetros, o cualquier Sistema térmicamente sensible.

3.7.1.2 Configuración de pines



Pin			Nombre	Descripción
SO	μSOP	TO-92		
1, 2, 6, 7, 8	2, 3, 5, 6, 7	—	N.C.	Sin conexión
3	8	3	VDD	VDD opcional. VDD debe estar conectado a tierra en modo de parásito.
4	1	2	DQ	Datos entrada / salida. Pin con interfaz 1-wire. Proporciona alimentación modo parásito.
5	4	1	GND	Tierra

Figura 3.7.2 Diagrama de pines DS18B20.

Tabla 3.7.1 Descripción de pines DS18B20

3.7.1.3 Resumen

En la figura 3.7.3 se muestra un diagrama de bloques del DS18B20, y en la tabla 3.7.1 se da una descripción de los pines. La ROM de 64 bits del dispositivo contiene un número de serie único para cada dispositivo que lo identifica. La memoria scratchpad contiene un registro de 2 bytes de temperatura que almacena la salida digital del sensor. Además, el scratchpad proporciona acceso a los registros de 1 byte de activación superior e inferior de la alarma (TH y TL) y al registro de 1 byte de configuración. El registro de configuración permite al usuario establecer la resolución de la conversión digital de la temperatura a 9, 10, 11 ó 12 bits. Los TH, TL, y el registro de configuración son memoria no volátil (EEPROM), por lo que se conservan los datos cuando el dispositivo está apagado.

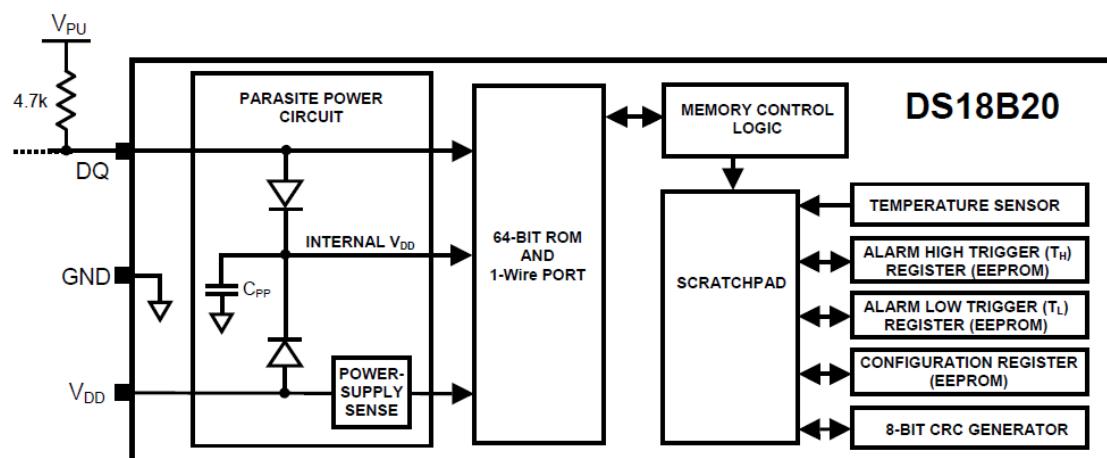


Figura 3.7.3 Diagrama de bloques del sensor DS18B20.

El DS18B20 usa el protocolo de bus 1-Wire de Maxim que implementa la comunicación de bus con una señal de control. La línea de control requiere una resistencia de pull-up a la cual todos los dispositivos están conectados por el bus a través de un puerto triestado o colector abierto (el pin DQ en el caso de la DS18B20). En este sistema de bus, el microprocesador (El dispositivo maestro) identifica y opera a cada dispositivo del bus con el código de serie de 64 bits, único para cada uno. Debido a que cada dispositivo tiene un código único, el número de éstos que puede haber en un bus es virtualmente ilimitado.

Otra característica de la DS18B20 es la capacidad de funcionar sin una fuente de alimentación externa. La alimentación puede ser suministrada a través de una resistencia de pull-up a través del pin DQ cuando el bus está a nivel alto. El bus a nivel alto también cargará el condensador interno (CPP), que a su vez alimentará al dispositivo cuando el bus esté a nivel bajo. Este método de obtención de energía del bus 1-Wire se conoce como "parasite power." Como alternativa, el DS18B20 también puede ser alimentado por una fuente externa de V_{DD} .

3.7.1.4 Especificaciones Eléctricas. (-55°C a 125°C, $V_{DD}=3.0V$ a 5.5V)

Parámetros	Símbolo	Condición	Min	Tip	Max	Unidades
Tensión Alimentación	V_{DD}	Local Power	3,0		5,5	V
Tensión Alimentación de pullup	V_{PU}	Parasite Power	3,0		5,5	V
		Local power	3,0		V_{DD}	
Error Sensor de Temperatura	t_{ERR}	-10°C a 85°C			$\pm 0,5$	°C
		-55°C a 125°C			$\pm 0,5$	
Entrada 0 lógico	V_{IL}		3		5,5	V
Entrada 1 lógico	V_{IH}	Local power	2,2	El menor de		V
		Parasite Power	3,0	5,5 o $VDD+0,3$		
Corriente inversa	I_L	$VI/O = 0,4V$	4,0			mA
Corriente Standby	I_{DDS}		750		1000	nA
Corriente en activo	I_{DD}	$VDD = 5V$	1		1,5	mA
Corriente entrada DQ	I_{DQ}		5			μA
Deriva				$\pm 0,2$		°C

Tabla 3.7.2 Especificaciones eléctricas DS18B20.

3.7.1.5 Comunicaciones 1-Wire

1-Wire es un protocolo de comunicaciones serie diseñado por *Dallas Semiconductor*. Está basado en un bus, un maestro y varios esclavos de una sola línea de datos de la que se alimentan. Por supuesto, necesita una referencia a tierra común a todos los dispositivos.

En un sistema 1-Wire, el dispositivo maestro controla uno o varios dispositivos esclavos. El DS18B20 es siempre esclavo. Cuando hay un solo esclavo en el bus, el sistema se conoce como un sistema "single-drop" si por el contrario hay varios esclavos en el bus se conoce como sistema "multidrop".

Todos los datos y los comandos se transmiten por el bus 1-wire con el bit menos significativo primero.

A. Configuración Hardware

El bus 1-Wire tiene, por definición, sólo una línea de datos única. Cada dispositivo (maestro o esclavo) se conecta a la línea de datos a través de un colector abierto o de un puerto triestado. Esto permite que cada dispositivo "libere" la línea de datos cuando no esté transmitiendo para que el bus pueda estar libre para otro dispositivo.

El bus de un sistema 1-wire requiere de una resistencia de pull-up externa de aproximadamente $5k\Omega$, por lo que el estado de reposo para el bus 1-Wire es nivel alto. Si la transmisión es suspendida, el bus debe ser dejado en modo inactivo. Si el bus se mantiene a nivel bajo por más de $480\mu s$, todos los componentes del sistema se restablecerán.

B. Secuencia de Operación

La secuencia para acceder a la DS18B20 es el siguiente:

Paso 1. Inicialización.

Paso 2. Comandos ROM (seguido de cualquier intercambio de datos requeridos).

Paso 3. Comando de función del DS18B20 (seguido por un intercambio de datos requeridos).

Habrá que seguir esta secuencia para poder acceder al DS18B20, Sino se siguiera esta secuencia el dispositivo podría hacer una acción incorrecta o no hacer nada. Las excepciones a esta regla son *SEARCH ROM [F0h]* y *ALARM SEARCH [ECH]*.

Inicialización

Todas las operaciones en un bus 1-wire comienzan con una secuencia de inicialización que consiste en un pulso de reset transmitido por el dispositivo maestro bus seguido por el pulso de presencia, transmitido por los dispositivos esclavos.

Comandos ROM

Después de que el maestro haya detectado un pulso presencia, puede emitir un comando de ROM. Estos comandos actúan en la ROM de 64 bits de cada dispositivo de forma independiente y permiten al maestro determinar cuantos y que tipo de dispositivos están en el bus 1-Wire. Hay cinco comandos de ROM, y cada comando es de 8 bits de longitud. El dispositivo maestro debe emitir un comando ROM apropiado antes de emitir un comando de la función DS18B20. La figura 3.7.4 muestra un diagrama de flujo para la operación con los comandos ROM. A continuación se nombran estos comandos y una breve definición.

- **SEARCH ROM [F0h]:** Cuando un sistema se inicializa, el maestro debe conocer los códigos ROM de todos los dispositivos esclavos. Así se determina el número de esclavos y el tipo que hay en el bus. Despues de un proceso el maestro sabe el código ROM de cada dispositivo esclavo. Despues de cada ciclo de búsqueda de la ROM, el maestro debe volver al paso 1 (inicialización).
- **READ ROM [33h]:** Este comando sólo se puede utilizar si hay un solo esclavo en el bus. Permite leer la ROM del dispositivo esclavo sin utilizar el procedimiento de búsqueda. Si se utiliza y hay más de uno se provoca una colisión de datos.
- **MATCH ROM [55h]:** Este comando seguido por una secuencia de 64 bits permite al maestro seleccionar un dispositivo esclavo. Sólo el dispositivo que su dirección coincida con los 64 bits responderá al comando de función emitida por el maestro.
- **SKIP ROM [CCh]:** El dispositivo maestro puede utilizar este comando para seleccionar a todos los dispositivos en el bus al mismo tiempo, sin que haya que enviar otra información.
- **ALARM SEARCH [ECH]:** El funcionamiento de este comando es idéntico al comando *SEARCH ROM*, salvo que sólo responderán los esclavos con una alarma activada.

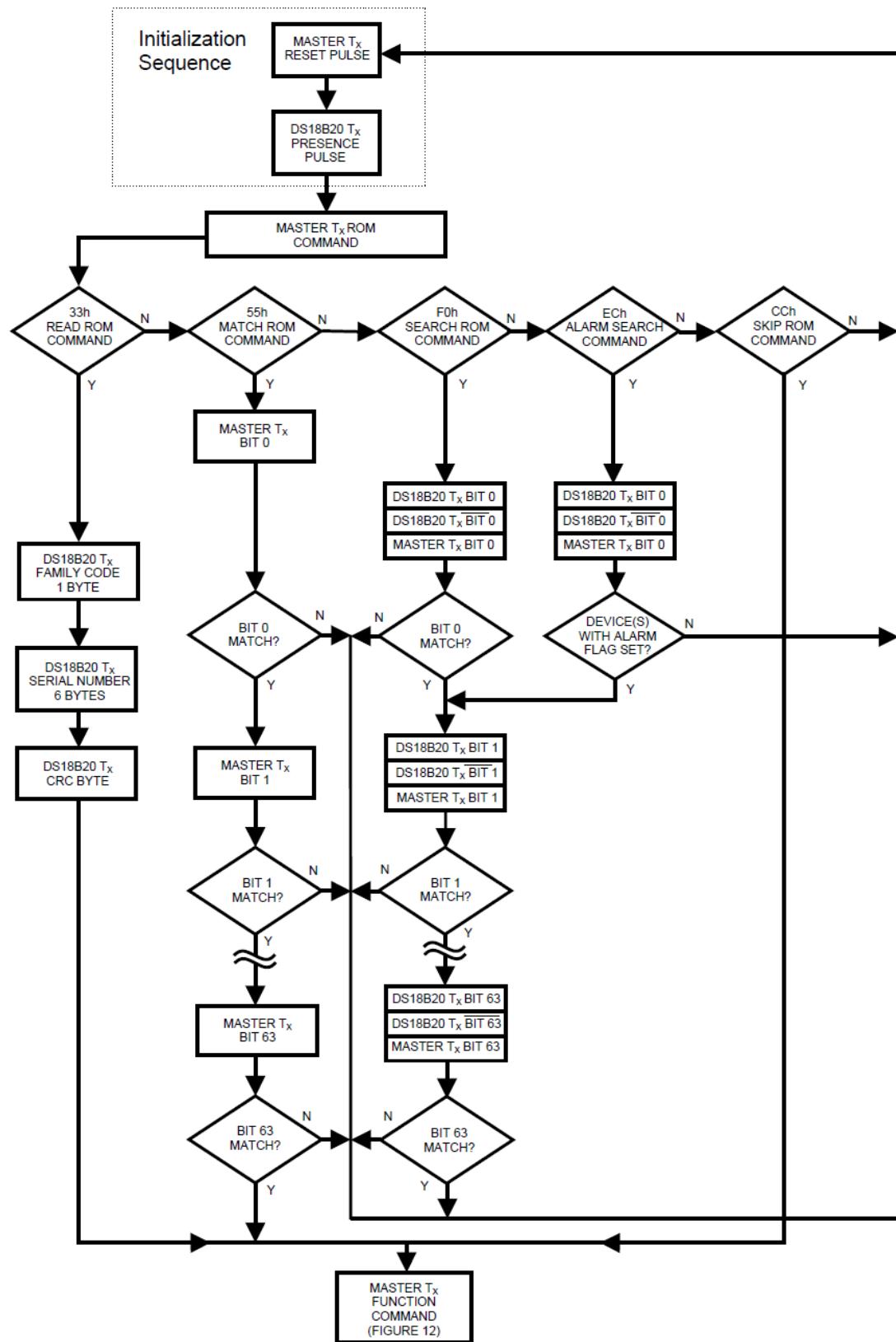


Figura 3.7.4 Diagrama de flujo para la operación con los comandos ROM.

Comandos de función DS18B20

Después de que el dispositivo maestro del bus ha utilizado el comando ROM para seleccionar el dispositivo DS18B20 con el que desea comunicación, el maestro puede emitir uno de los comandos de función DS18B20. Los comandos de la función DS18B20, que se describen a continuación, son ilustrados por el diagrama de flujo en la Figura 3.7.5:

- **CONVERT T [44h]:** Este comando inicia una conversión de temperatura individual. Tras la conversión, los datos se almacenan en el registro de temperatura de 2 bytes en la memoria y pasa el DS18B20 a su estado de reposo.
- **WRITE SCRATCHPAD [4Eh]:** Este comando permite al dispositivo maestro escribir 3 bytes de datos en el scratchpad del DS18B20. El primer byte es escrito en el registro de TH (byte 2 del scratchpad bloc de notas), el segundo byte se escribe en el registro TL (Byte 3), y el tercer byte se escribe en el registro de configuración (byte 4). Los datos deben de transmitirse con el bit menos significativo primero.
- **READ SCRATCHPAD [BEh]:** Este comando permite al dispositivo maestro leer el contenido del scratchpad. La transferencia de datos se inicia con el bit menos significativo del byte 0. El dispositivo maestro puede abortar la lectura si ya ha leído la información que necesita.
- **COPY SCRATCHPAD [48h]:** Este comando copia el contenido TH, TL y registro de configuración (bytes 2, 3 y 4) del scratchpad a la EEPROM.
- **RECALL E² [B8h]:** Este comando recupera los valores de disparo de la alarma (TH y TL) y los datos de configuración de la EEPROM y coloca los datos en bytes, 2, 3 y 4, respectivamente, en la memoria del scratchpad.
- **READ POWER SUPPLY [B4h]:** Este comando determina si algún DS18B20 en el bus está en configuración modo parasito.

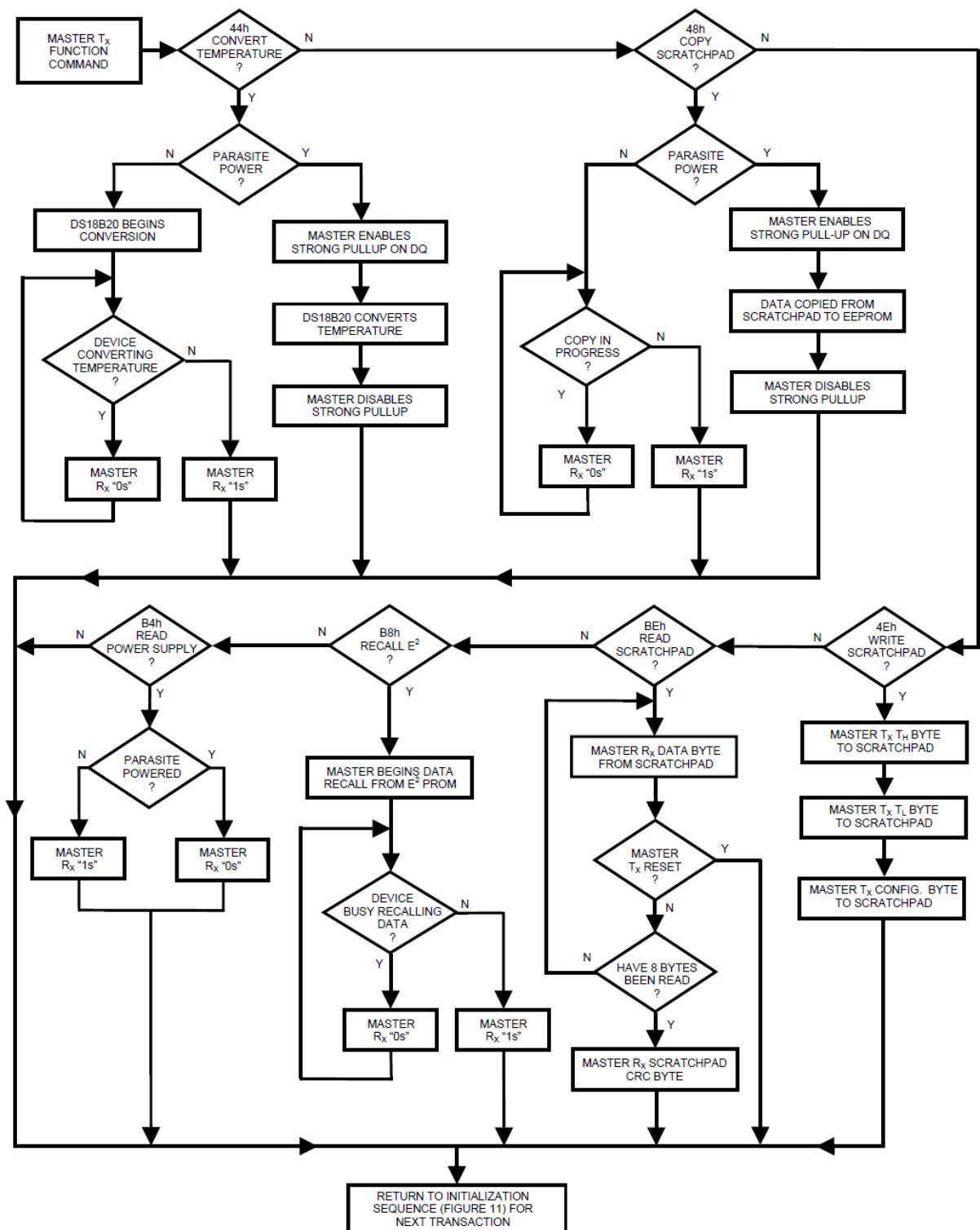


Figura 3.7.5 Flujo para la operación con los comandos de la función DS18B20.

C. Señalización 1-Wire

El DS18B20 utiliza un estricto protocolo de comunicación 1-Wire para garantizar la integridad de los datos. Diferentes tipos de señales son definidas en este protocolo: pulso de reset, pulso presencia, escribir 0, escribir 1, leer 0, y leer 1. El dispositivo maestro inicia todas las señales excepto el pulso presencia.

Procedimiento de inicialización, reset y pulso presencia

Cualquier comunicación con un dispositivo DS18B20 comienza con una secuencia de inicialización, que consiste en un reset enviado por el dispositivo maestro, seguido por un pulso de presencia del DS18B20. Esto se ilustra en la Figura 3.7.6. Cuando la DS18B20 envía el pulso de presencia en respuesta al reset, se indica al maestro que éste está en el bus y listo para funcionar.

Durante la secuencia de inicialización del bus, el dispositivo maestro transmite (T_x) el pulso de reset poniendo al bus a nivel bajo durante un mínimo de $480\mu s$. Despues el maestro libera el bus y entra en modo recepción (R_x). Cuando el bus se libera, la resistencia de pull-up $5k\Omega$ hace que el bus pase a nivel alto. El DS18B20 detecta este flanco de subida, espera entre $15\mu s$ y $60\mu s$ y después transmite un pulso de presencia que consiste en poner el bus a nivel bajo de $60\mu s$ a $240\mu s$.

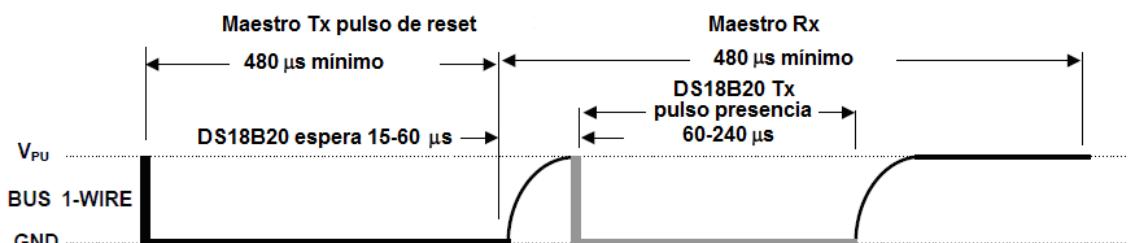


Figura 3.7.6 Secuencia de inicio de un bus 1-wire.

Intervalos de tiempo Lectura/Escritura

El dispositivo maestro debe escribir datos en el DS18B20 durante el intervalo de escritura y leerlos durante los intervalos de lectura. Un bit se transmite por el bus por intervalo de tiempo.

Tiempo de escritura

Hay dos tipos de intervalo de tiempo de escritura: "Escribir 1" y "Escribir 0". Ambos deben tener un mínimo de $60\mu s$ de duración con un tiempo de restablecimiento de $1\mu s$ entre intervalos de escritura. Ambos son iniciados por el dispositivo maestro poniendo el bus a nivel bajo. Ver figura 3.7.7.

Para generar un intervalo de tiempo, Escribir 1, después de poner el bus a nivel bajo, el maestro debe liberar el bus durante $15\mu s$. Cuando el bus se libera, la resistencia de

pull-up $5k\Omega$ pone el bus a nivel alto. Para generar un intervalo de tiempo, Escribir 0, después de poner el bus a nivel bajo el dispositivo maestro debe seguir manteniéndolo durante al menos $60\mu\text{s}$.

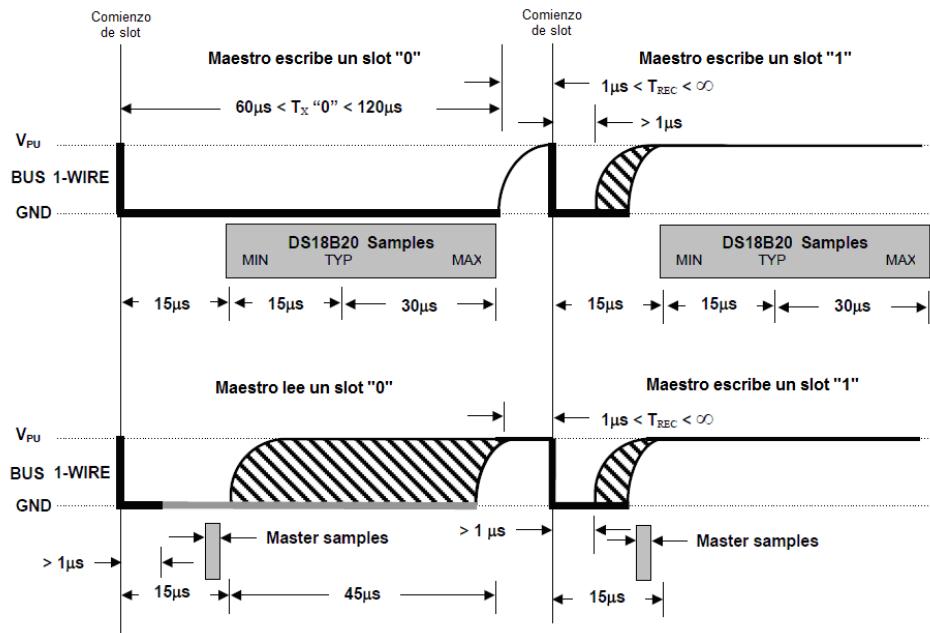


Figura 3.7.7 Escritura de un bus 1-wire.

Tiempo de lectura

El DS18B20 sólo puede transmitir datos al dispositivo maestro en el intervalo de tiempo de lectura. Por lo tanto, el maestro debe generar un intervalo de tiempo Leer, inmediatamente después de la emisión de un comando *Read Scratchpad [BEh]* o *Read Power Supply [B4h]*, para que el DS18B20 pueda proporcionar los datos solicitados. Además, se pueden generar intervalos de tiempo de lectura después de un comando *Convert T [44h]* o *Recall E² [B8h]* para saber el estado de la operación tal y como se indica en la sección Comandos de función.

Todos los intervalos de tiempo de lectura deben ser de al menos $60\mu\text{s}$ de duración, con un tiempo de restablecimiento de $1\mu\text{s}$ entre las ranuras. Un intervalo de tiempo de lectura se inicia con el dispositivo maestro poniendo el bus a nivel bajo durante al menos $1\mu\text{s}$ y luego liberándolo (ver Figura X). Después de que el maestro inicie el intervalo de tiempo de lectura, el DS18B20 comenzará a transmitir un 1 o un 0 en el bus. Los datos de salida del DS18B20 serán válidos $15\mu\text{s}$ después del flanco de bajada que inició el intervalo de tiempo de lectura. Por lo tanto, el maestro debe liberar el bus y mostrar su estado dentro de $15\mu\text{s}$ desde el inicio del intervalo.

3.7.1.6 Programación DS18B20

Para realizar la programación del sensor DS18B20 se han realizado diferentes funciones, puesto que el diagrama de flujo del sensor es complejo y es necesario una serie de funciones.

A continuación se muestra la función implementada para realizar la secuencia de inicio de transmisión de una comunicación en un bus 1-wire.

```
byte reset(){
    int espera=250;
    byte x=0;
    pinMode(pinTemp, INPUT);
    do {
        if (espera -- == 0) return 0;
        delayMicroseconds(1);
    }
    while (digitalRead(pinTemp)==0);
    digitalWrite(pinTemp, LOW);
    pinMode(pinTemp, OUTPUT);
    delayMicroseconds(500);
    pinMode(pinTemp, INPUT);
    delayMicroseconds(65);
    x = !digitalRead(pinTemp);
    delayMicroseconds(490);
    return x;
}
```

Las acciones que realiza esta función son las explicadas para la secuencia de inicio de un bus 1-wire:

- 1) Se espera el tiempo mínimo de presencia de un sensor DS18B20 si la línea está en un 0 lógico quiere decir que hay un sensor en la línea por lo que la función de reset finaliza en ese momento.
- 2) Si se espera los 240 μ s sin leer en la línea 1 lógico significa que no hay ningún sensor en la línea en ese momento el microcontrolador (maestro) pone a nivel bajo, 0 lógico, durante al menos 480 μ s.
- 3) En ese momento se espera el tiempo suficiente para que un sensor ponga la línea a nivel alto (1 lógico). Al menos 60 μ s y se observa la línea para observar si existe una respuesta por parte de algún sensor DS18B20.
- 4) Finalmente se espera al menos 480 μ s que es el tiempo que debe estar el microcontrolador recibiendo. Puesto que hay que esperar los 240 μ s del pulso de presencia del sensor DS18B20 y 240 μ s del tiempo para poner la línea a 1 lógico.

A continuación se muestra el código implementado para capturar la temperatura de un sensor DS18B20; se puede observar como sigue el patrón de operación de un bus 1-wire, con una inicialización un comando ROM y un comando función para poder operar con un DS18B20.

```

void ElegirSensor(byte A){
    if (reset()==0){Err=0;}
    delay(1);
    EscribirComando(85);
    for (byte i=0; i <= 63; i++){
        Escribirbit(sensor[A][i]);
    }
}
float TomarTemp(byte A){
    int EsperaConversion=0;
    byte Temp1,Temp0;
    float resultado;
    ElegirSensor(A);
    EscribirComando(68);
    do{
        EsperaConversion++;
        delay(1);
    }
    while ((Leerbit()==0)&&(EsperaConversion<2000));
    if (EsperaConversion>1500){Err=1;}
    ElegirSensor(A);
    EscribirComando(190);
    Temp0=LeerByte();
    Temp1=LeerByte();
    if (reset()==0){Err=0;}
    delay(1);
    resultado=TempxPant(Temp0,Temp1);
    return resultado;
}

```

Las acciones derivadas de este código son las siguientes:

- 1) Se elige el sensor a seleccionar. Para ello se realiza una secuencia de inicio. Seguido del comando ROM MATCH ROM que se emplea para seleccionar un dispositivo esclavo (DS18B20).
- 2) Se utiliza un comando función CONVERT T y se espera la conversión de la temperatura.
- 3) Se vuelve a hacer el paso 1 para volver a seleccionar el mismo sensor.
- 4) Se vuelve a emplear un comando función pero ahora se emplea el comando READ SCRATCHPAD.
- 5) Se leen los primeros Bytes y se emplea una función para convertir los valores a una variable flota.

La siguiente función muestra como se puede capturar las ROM de todos los sensores que haya en un bus 1-wire. Esta función no acaba con un comando de función.

```
void IniciarSensorT();
byte bit, bitnegado;
int distinto=-1;
int local=-1;
byte j=0;
byte otro=0;
int distintoTEMP=-1
do {
if (reset()==0){Err=0;}
delay(1);
EscribirComando(240);
for (int i=0; i <= 63; i++){
    bitnegado=Leerbit();
    local=-1;
    if (bit==bitnegado)&&(distinto<i)){bit=1;distintoTEMP=i;otro=1;}
    if ((bit==bitnegado)&&(distinto==i)){otro=0;}
    distinto=distintoTEMP;
    sensor[j][i]=(bit);
    Escribirbit(bit);
}
j=j+1;
}
while (otro==1);
}
```

Las acciones que realiza esta función son las explicadas son:

- 1) Se ejecuta la secuencia de inicio de comunicación para luego emplear un comando de operación ROM en este caso SEARCH ROM.
- 2) Se leen los 64 bits que se transmiten por la línea, proveyendo si existe o no colisión de un sensor para luego seguir buscando más sensores.
- 3) Se completa la variable global sensor con los bits leídos.
- 4) Si ha habido colisión se repite el proceso hasta que no haya más colisiones y se complete las lecturas de todos los sensores.

3.7.2 Sensor de humedad

Para el sensor de humedad se ha elegido el sensor de Honeywell, modelo HIH-4021-004 debido a sus altas prestaciones y a su uso sencillo.



Figura 3.7.6 Sensor de humedad de Honeywell.

3.7.2.1 Descripción

Los sensores de humedad de la serie HIH-4010/4020/4021 son diseñados específicamente para el alto número de usuarios OEM (Original Equipment Fabricante).

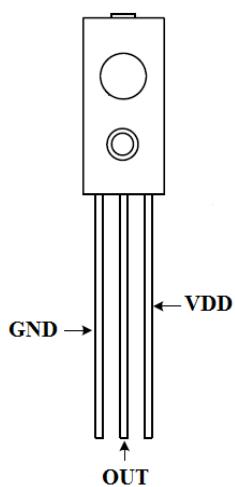
El HIH-4010 es un sensor de humedad integrado no cubierto, el HIH-4020 es un sensor de humedad integrado cubierto y el HIH-4021 es un sensor de humedad integrado cubierto anticondensación. Los tres modelos están disponibles con dos configuraciones de patillas.

El sensor se puede conectar directamente a una entrada de un microcontrolador u otro dispositivo gracias a que este dispositivo tiene una salida de tensión lineal proporcional a la humedad, con un consumo de corriente típico de tan sólo 200mA.

Este sensor es hermético y reduce o elimina el coste de calibración ya que se proporcionan los datos de calibración individuales de cada dispositivo.

La construcción del sensor en múltiples capas ofrece una excelente resistencia a posibles acciones que lo deterioren, como una soldadura, polvo, suciedad, aceites y productos químicos ambientales.

3.7.2.2 Configuración de pines



Nombre	Descripción
VDD	Alimentación del dispositivo
DQ	Datos de salida. Salida analógica proporcional a la humedad.
GND	Tierra

Tabla 3.7.3 Descripción pines Sensor humedad.

Figura 3.7.7 Configuraciones pines del sensor de humedad de honeywell.

3.7.2.3 Especificaciones

MODEL:	HIH-4021-004
Channel:	158
Serial Number	27082300151
Wafer Number	81546B08
MRP	6840512
	Calculated Values at 5V
Vout @0%RH:	0.805303V
Vout @75.3%RH:	3.129566V
Accuracy @25 C:	3.5 %RH
Zero Offset:	0.805303
Slope:	30.866707 mV/%RH
Sensor RH:	(Vout-0.805)/30.867
	Ratiometric Response
	For 0 to 100%RH
Vout=	Vsupply (0.166 to 0.773)

Tabla 3.7.4 Especificaciones sensor HIH-4021-004.

En la figura 3.7.8 podemos observar la zona operativa del sensor HIH-4021-004. Se aprecia que el sensor tiene un rango de funcionamiento a respetar. A una temperatura ambiente excesiva podría dañar el sensor. Este sensor irá acompañado de otro de temperatura que la controla para que el sensor de humedad no sufra desperfectos.

Las especificaciones entregadas por el fabricante son individuales en el modelo elegido así para el sensor de humedad HIH-4021-004 el fabricante nos entrega una hoja con las características individual del sensor adquirido. En la tabla 3.7.4 se muestran las especificaciones del sensor. En ellas se muestran la relación lineal que existe entre la tensión de salida del sensor y la humedad relativa.

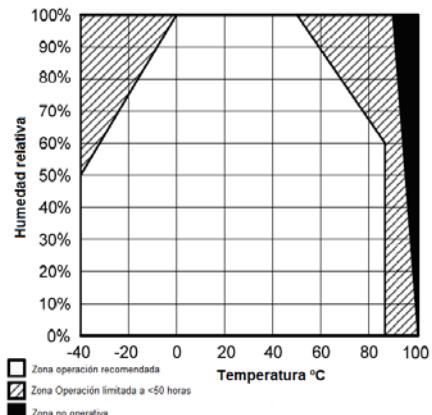


Figura 3.7.8 Entorno operativo sensor HIH-4021-004.

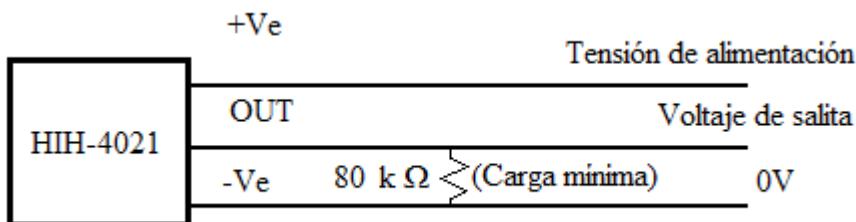


Figura 3.7.9 Circuito típico del sensor HIH-4021-004.

En la figura 3.7.9 se puede apreciar el circuito típico de ejecución del sensor HIH-021-004, el único elemento necesario es una resistencia de valor mínimo 80kΩ entre la señal de salida y GND. En este caso la resistencia utilizada para tal efecto ha sido de 100kΩ.

3.7.2.4 Programación sensor humedad

A continuación se observa la función diseñada para captar la señal de salida del sensor de humedad y convertirlo a humedad relativa.

```
void TomarHumedad(){
    float voltios;
    voltios=(float(analogRead(pinHumedad)))*5/1024;
    Humedad=(voltios-0.805303)/0.03099;
}
```

Esta función realiza las siguientes acciones:

- 1) Toma el valor del pin de humedad a través del convertidor analógico-digital.
- 2) Con ese valor obtenemos los voltios de salida del sensor.
- 3) Con los datos de las especificaciones, se pasa de voltios a la humedad relativa del aire.

3.7.3 Realización física del sensor de temperatura y humedad

En la figura 3.7.10 aparecen los dos sensores, el sensor DS18B20 y el HIH-4021. Para tener unas mediciones reales de temperatura y humedad, necesitamos que los sensores estén al aire libre pero sin las posibles inclemencias meteorológicas.

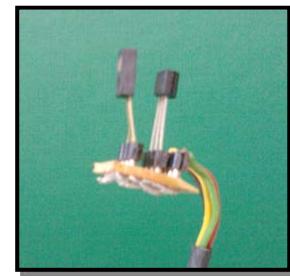


Figura 3.7.10 Sensores DS18B20 y HIH-4021.

Para ello se ha realizado una estructura hexagonal en la que puede entrar el aire pero no la lluvia ni el rocío. En la figura 3.7.11 se puede apreciar el esquema de dicha estructura.

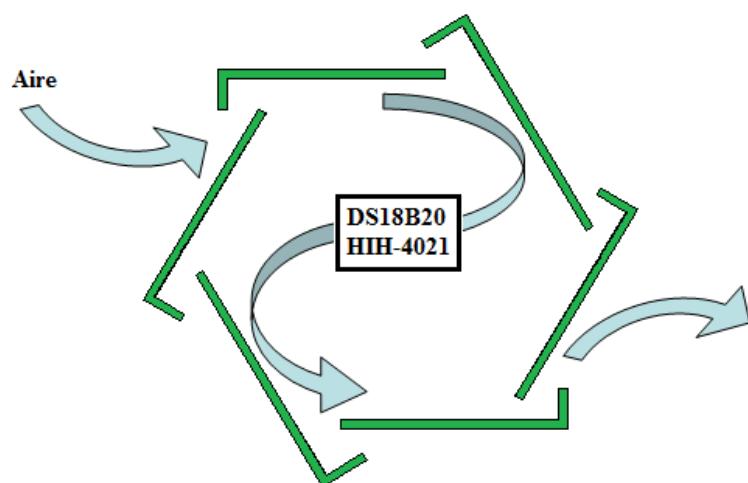


Figura 3.7.11 Representación de la estructura de cobertura de los sensores.

Ésta será de plástico resistente, de color blanco, para repeler en la medida de lo posible el efecto de la radiación solar, pudiendo obtener un valor real de la temperatura.

En la figura 3.7.12 se puede observar la estructura desde la parte de abajo y su forma hexagonal y en la figura 3.7.13 se observa su abertura en cada lado de la estructura. En ambas figuras se observa el material de la estructura y el color blanco mencionado anteriormente.

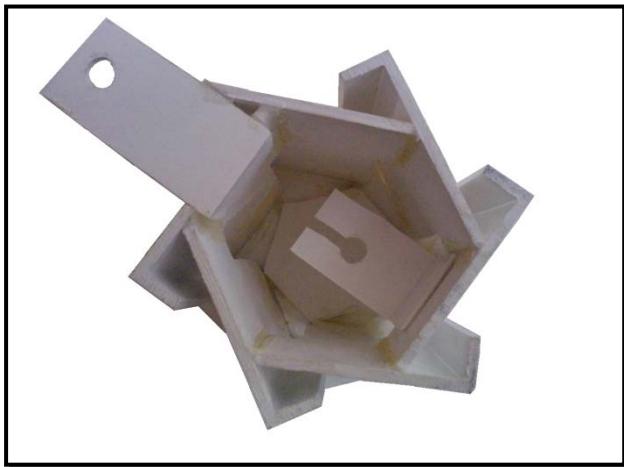


Figura 3.7.12 Vista, desde debajo, de la estructura protectora de sensores.



Figura 3.7.13 Abertura lateral de la estructura protectora de sensores.

La cobertura con los sensores dentro, colocada en la estructura utilizada para soportar el sistema se puede ver en la figura 3.7.14.



Figura 3.7.14 Sensor de Temperatura y Humedad.

3.8 Alimentación

Con el fin de que el proyecto tenga autonomía, se ha diseñado un sistema de alimentación para poder cubrir el consumo a lo largo de un tiempo determinado. Lo ideal es que se pueda tener alimentado a toda la herramienta hardware durante el tiempo que debería estar funcional, es decir durante un año seguido.

En este caso la única solución viable es la de una instalación solar fotovoltaica aislada de baja potencia. Este tipo de instalaciones constan de una placa solar fotovoltaica, una batería, un regulador y un inversor. El inversor, convierte corriente continua en alterna, en este caso no es necesario puesto que nuestro sistema se alimenta con corriente continua. En la figura 3.8.1 se puede observar el esquema de este tipo de instalaciones.

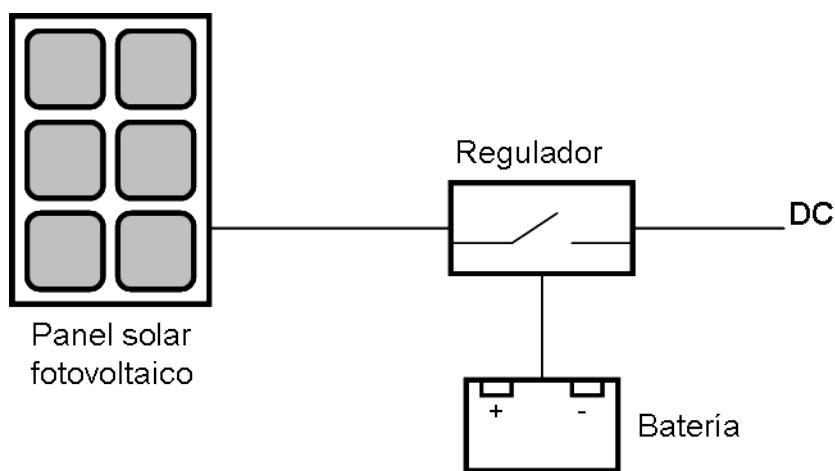


Figura 3.8.1 Instalación fotovoltaica aislada.

Para realizar este tipo de instalaciones, habrá que seguir unos pasos determinados; primero habrá que determinar el consumo del sistema que se quiere alimentar, en este caso la herramienta hardware; segundo habrá que dimensionar la instalación y por ultimo habrá que hacer la realización física.

3.8.1 Consumo diario

Habrá que determinar el consumo, tanto de forma teórica como práctica. El problema de hacerlo de forma teórica es que algunos dispositivos no se conocen las especificaciones eléctricas, como son del Módulo SD y del GPRS, aunque de este último si se conoce el consumo máximo, 2,2A, por lo que se ha optado por valores medidos.

Se mide el consumo instantáneo del sistema a lo largo de una hora, ésta debe coincidir con una transmisión del fichero de datos al servidor FTP. Con un programa realizado en Matlab se genera una gráfica que representa el consumo instantáneo. El

sistema dispone de 3 tipos diferentes de consumos promedio. El de medida, el de stand-by y el de transmisión, como se observa en la 3.8.2

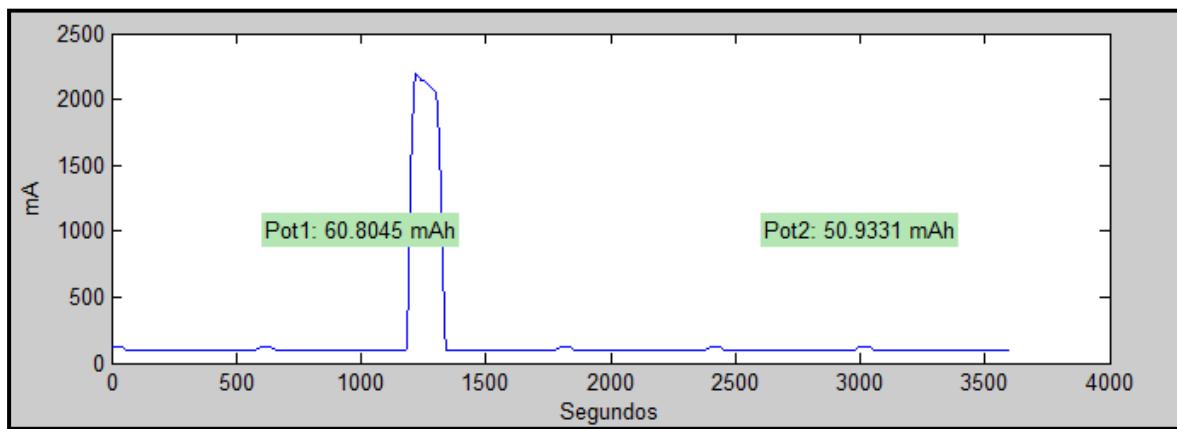


Figura 3.8.2 Consumo del sistema a lo largo de una hora.

En la figura se muestran dos potencias consumidas cada media hora, en una se transmite el archivo al servidor FTP y en la otra no. Con estas dos potencias se puede extraer los resultados para calcular el consumo a lo largo de un día, para ello debemos multiplicar Pot2, consumo en la media hora que no se transmite mediante GPRS por 47 que son las medias horas al día que se produce este evento y se le suma Pot1 que es la media hora en la que se transmite al servidor FTP. El resultado:

$$\text{Consumo en un día} = 47 * \text{Pot2} + \text{Pot1} = 2453,1 \text{ mAh.}$$

3.8.2 Dimensionado sistema de alimentación

El dimensionado del sistema de alimentación tendrá que partir de una serie de premisas. La primera será determinar la capacidad de la batería, ésta deberá ser suficiente para poder alimentar al sistema durante un tiempo determinado, pero sin que sea excesiva para que el tamaño y el precio de la batería no sea demasiado grande.

Se ha partido de la premisa de que la batería aguante más de 2 días sin ningún otro tipo de apoyo. Se ha elegido este tiempo puesto que la zona climática a estudiar dispone de más de un 80% de días soleados.

Para poder cumplir dicha premisa, la batería debe ser mayor a $2 \cdot 2,45 \text{ Ah} (> 5 \text{ Ah})$. Puesto que debe superar dicha capacidad se ha optado por una de 7,2AH/20H, como se observa en la figura 3.8.2, ésta capacidad suministra suficiente potencia para que se disponga de casi 3 días de ausencia de alimentación adicional.



Figura 3.8.3 Batería 7,2Ah/20H.

Después se dimensiona la placa solar fotovoltaica. Para hacerlo hay que determinar la inclinación y la potencia. Primero se determina la inclinación del panel solar, la óptima en Andalucía es de 32° con respecto al plano horizontal, esa inclinación dará la máxima potencia anual, pero en instalaciones aisladas sin apoyo de otro tipo de generadores eléctricos, para que la potencia media mensual sea lo más homogénea posible a lo largo del año se opta por inclinar los módulos solares. En este caso se inclina la placa a 40° para aumentar la potencia media mensual en los meses de invierno con respecto a los de verano, como se puede ver en la tabla 3.8.1.

Inclinación	Enero	Febr.	Marz.	Abril	Mayo	Junio	Julio	Agos.	Sept.	Octu.	Novi.	Dici.	Total
Óptima	3986	4314	5332	5429	6193	6493	6435	6155	5566	5055	3584	3338	5162
40°	4215	4457	5368	5301	5902	6111	6088	5951	5542	5184	3745	3535	5121

Tabla 3.8.1 Potencia media mensual según inclinación.

Para poder tener dicha inclinación se optado por realizar una estructura que soporte la placa. Una escuadra a partir de 3 perfiles de aluminio. Se han diseñado con 3 medidas diferentes 28,5cm (la anchura de la placa solar), 21,83cm y 18,32cm estas medidas nos dan una escuadra con un ángulo de 40° . Como se muestra en la figura 3.8.4.

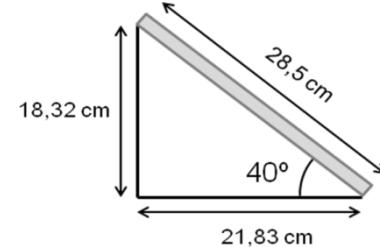


Figura 3.8.4 Estructura de soporte.

A continuación se dimensiona la potencia del panel solar. Teniendo en cuenta que la potencia suministrada por la placa a de ser suficiente para suprir el consumo diario de la herramienta hardware.

El consumo diario de la herramienta hardware es de 2,5Ah, por lo que si se supone el caso peor en 5 las horas del día en la que se recibe irradiación solar. Entonces la potencia que debe suministrar la placa es de al menos 500mAh.

La placa solar seccionada para el sistema de alimentación es una de Isofotón con las siguientes características eléctricas:

- | | |
|--|-------------------|
| • Potencia nominal | Pmax (Wp) 10 |
| • Tensión en circuito abierto | Voc (V) 21,6 |
| • Tensión en el punto de máxima potencia | Vmax (V) 17,4 |
| • Corriente de cortocircuit | Isc (A) 0,82 |
| • Corriente en el punto de máxima potencia | Imax (A) 0,58 |
| • Tolerancia de potencia | (%/Pmax). ± 1 |



Figura 3.8.5 Panel solar

Por último habrá que dimensionar el regulador, para ello habrá que determinar la corriente instantánea máxima que circula por el regulador el regulador y la tensión que de salida de dicho regulador.

La corriente máxima ocurre cuando transmite el módulo GPRS, en este caso será de algo menos de 2,5 A la potencia consumida por el sistema hardware, y el voltaje de salida ha de ser 12 voltios. En este caso el regulador seleccionado es de pocos con las siguientes características:

- Máxima corriente de módulos 5A
- Máxima corriente de consumo 5A
- Voltaje nominal del sistema 12/24
- Autoconsumo <4mA
- Dimensiones 80 x 100 x 32 mm
- Tipo de protección IP22



Figura 3.8.6 Regulador phocos

3.8.3 Realización física sistema de alimentación

Para hacer la realización física del sistema de alimentación se ha precisado de una cobertura para la batería y el regulador. Se ha empleado una pequeña caja de plástico que se ha intentando sellar lo máximo posible para poder estar a la intemperie sin que los componentes se mojen por las inclemencias meteorológicas. A la caja se le ha puesto un ventilador para que la temperatura de los componentes no alcance un valor por encima de lo permitido en las especificaciones, para que funcione sólo en los momentos necesarios se ha conectado a la placa solar para que cuando haya suficiente potencia solar que provoca el aumento de la temperatura el ventilador disipe esta potencia de la caja protectora. En la figura 3.8.6 podemos observar los dispositivos dentro de la caja, mientras la figura 3.8.7 muestra el ventilador en la parte lateral para dar ventilación.



Figura 3.8.7 Caja Protectora.



Figura 3.8.8 Visión frontal del ventilador.

En la figura 3.8.8 se muestra el sistema de alimentación completo. En esta figura se puede observar la inclinación seleccionada de la placa solar.



Figura 3.8.9 Sistema alimentación.

3.9 Herramienta hardware completo

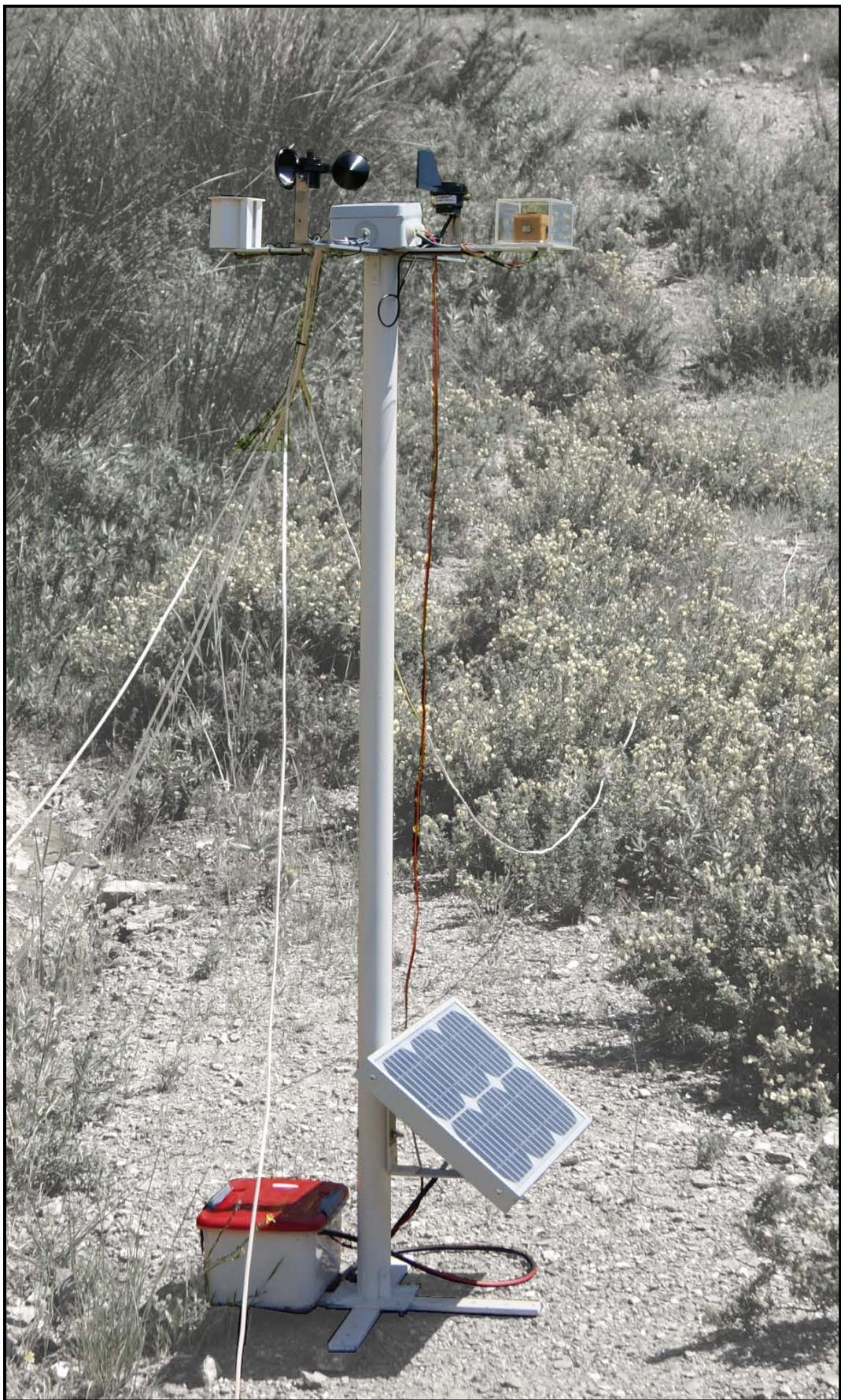


Figura 3.9.1 Herramienta hardware completo

Capítulo 4

Verificación del sistema hardware

En este capítulo se realizan una serie de pruebas para comprobar el correcto funcionamiento del sistema hardware y de sus componentes.

Los dispositivos empleados han sido calibrados previamente por los respectivos fabricantes, por lo que este capítulo no trata éste tema, sino la verificación propiamente dicha.

Para realizar la verificación de los componentes del sistema, cada sensor y cada elemento del mismo, es necesaria una serie de instrumentación externa al propio sistema para poder probar si los diferentes sensores empleados funcionan correctamente.

4.1 Instrumentación externa

4.1.1 Equipo multifunción 4 en 1 PCE-EM882

A. Descripción general

El medidor medioambiental multifunción 4 en 1 consta de un sonómetro, un luxómetro, un medidor de humedad y un medidor de temperatura.

El sonómetro posibilita la realización de mediciones sonoras en diferentes ambientes. El luxómetro determina las condiciones de luz al aire libre y en el puesto de trabajo. La medición se lleva a cabo con un diodo de silicio sensible a la luz. El medidor de temperatura y humedad trabaja con un semiconductor de aire y humedad y un sensor tipo K para mediciones de humedad de aire y temperatura.



Figura 4.1.1 Equipo PCE-EM882.

B. Especificaciones técnicas

Sonómetro:

Rango de medición	A/C Lo (bajo) 35-100 dB, A/C Hi (alto) 65-130 dB
Resolución	0,1 dB
Rango de frecuencia	30 Hz 10 kHz
Valoraciones de frecuencia	A y C
Valoraciones de tiempo	Rápida
Precisión	± 3,5 dB con 94 dB nivel sonoro Señal senoidal de 1 kHz
Micrófono	Micrófono eléctrico con condensador

Luxómetro:

Rangos de medición	20/200/2000/20000 Lux (20000 rango lux = indicador x 10)
Salto de rango	“1” en la posición de más valor
Precisión	± 5% del valor de medición + 10 posiciones (Calibrado bombilla normal Au con una temperatura cromática de 2856° k) repetición de medición ± 2 %
Característica de temperatura	± 0,1 % / 0,1° C
Sensor	Fotodiodo de silicio con filtro

Medidor de humedad de aire:

Rango de medición	25 - 95 % humedad de aire
Resolución	0,1 %
Precisión	± 5% (con 25%, 35% y 95% humedad de aire)
Protección de sobrecarga	60 V DC o 24 V AC

Medidor de temperatura:

Rango de medición	De -20°C a +200°C / de -20°C a +750°C
Resolución	0,1°C
Precisión	± 3 % del valor de medición ± 2°C (-20...+200°C)
Protección de sobrecarga	60 V DC o 24 V AC

4.1.2 Anemómetro PCE-AM81**A. Descripción general.**

Anemómetro manejable con sensor de rueda alada para determinar la velocidad del viento, se puede realizar un chequeo in situ de manera rápida. Su ligera rueda alada percibe las más pequeñas velocidades del aire.

**Figura 4.1.2 Equipo PCE-AM81.****B. Especificaciones técnicas**

Rangos de medición	0,4-30,0 m/s → 1,4-108,0 km/h
Resolución	0,1 m/s → 0,1 km/h
Precisión (para todos los rangos)	± 3 % (< 20 m/s) y ± 4 % (> 20 m/s)
Pantalla	LCD de 8 mm
Superación de rango máximo	En la pantalla aparece “- - -”
Carcasa	Plástico ABS
Alimentación	Batería de bloque de 9 V
Dimensiones del aparato	156 x 60 x 33 mm
Peso	160 g
Rango de temperatura ambiental	0 - 50 °C
Rango de humedad ambiental	0 - 80 % H. r.

4.2 Verificación de los sensores

4.2.1 Verificación sensor de temperatura

Para realizar la verificación se usa el PCE-EM882. Para las pruebas se han realizado distintas medidas a lo largo del tiempo, en diferentes escenarios. Se han intentado buscar escenarios extremos, pero debido a la imposibilidad de alcanzar valores extremadamente altos y bajos se ha optado por diferentes escenarios.

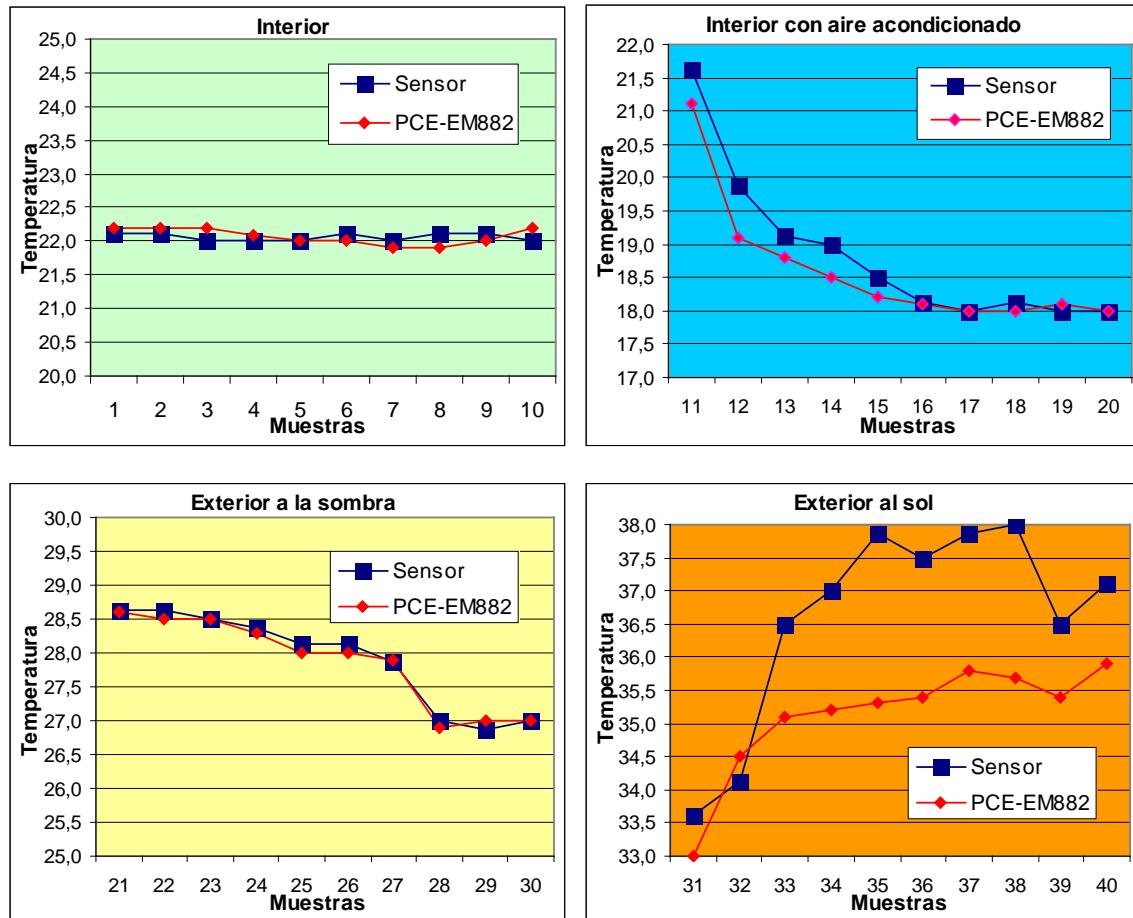


Figura 4.2.1 Medidas de distintos escenarios del sensor de temperatura y del PCE-EM882.

En las diferentes gráficas de la figura 4.2.1 se observa que el sensor de temperatura da un valor bastante exacto y homogéneo. Cabe destacar que las modificaciones bruscas de temperatura tarda más en captarlas, ya que debe cambiar todo el aire circundante al sensor puesto que se encuentra en la envoltura. Por este motivo al estar al sol percibe mucho más calor que el termómetro del PCE-EM882 ya que el sensor capta la temperatura del aire dentro de la envoltura, que si no hay corrientes de aire, no se renovará y se recalentará, mientras que el PCE-EM882 sólo se encuentra al sol en el momento de la medición. Por este motivo las mediciones al sol son diferentes al PCE-EM882.

4.2.2 Verificación del sensor de Humedad

Para realizar la verificación se ha empleado el PCE-EM882 y realizado pruebas, distintas medidas a lo largo del tiempo. Puesto que no se ha podido variar la humedad relativa del ambiente, se ha optado por diferentes escenarios en diferentes días.

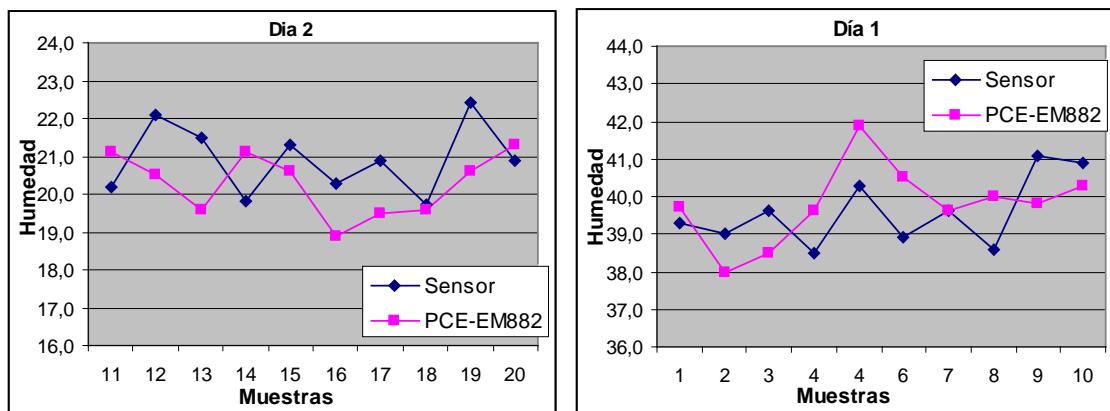


Figura 4.2.2 Medidas en diferentes días del sensor de Humedad respecto al PCE-EM882.

Se puede observar que ambos aparatos de medidas oscilan bastante en sus mediciones. Puede ser debido a que el sensor de humedad tiene una precisión del 3'5% y el PCE-EM882 tiene una precisión del 3% por lo que los valores son bastante razonables.

Vemos que ambos aparatos oscilan alrededor de un valor medio que es bastante similar. En el día 1 el valor medio para el sensor es 39'6% y para el PCE-EM882 es 39'8%. En el día 2 para el sensor es 20'9% y para el PCE-EM882 es 20'3%. Vemos que son valores bastante parecidos y que cada muestra no tiene una desviación demasiado elevada.

La comprobación de los valores extremos de humedad tampoco se podrían hacer puesto que no se puede suprimir la humedad ambiente de un lugar, ni tampoco se podría incrementar la humedad a valores cercanos al 99% puesto que debería haber agua en suspensión en el ambiente y en ese caso el aparato PCE-EM882 podría sufrir averías.

4.2.3 Verificación Anemómetro

Para la verificación del Anemómetro se ha empleado el PCE-AM81 y un ventilador, para generar una corriente constante de aire y así poder hacer las medidas necesarias. Se han tomado diferentes muestras con diferentes velocidades de aire.

En la figura 4.2.3 se observa las distintas medidas de los diferentes aparatos. Las variaciones del anemómetro con respecto al PCE-AM81, aunque apreciables, no son excesivamente significativas.

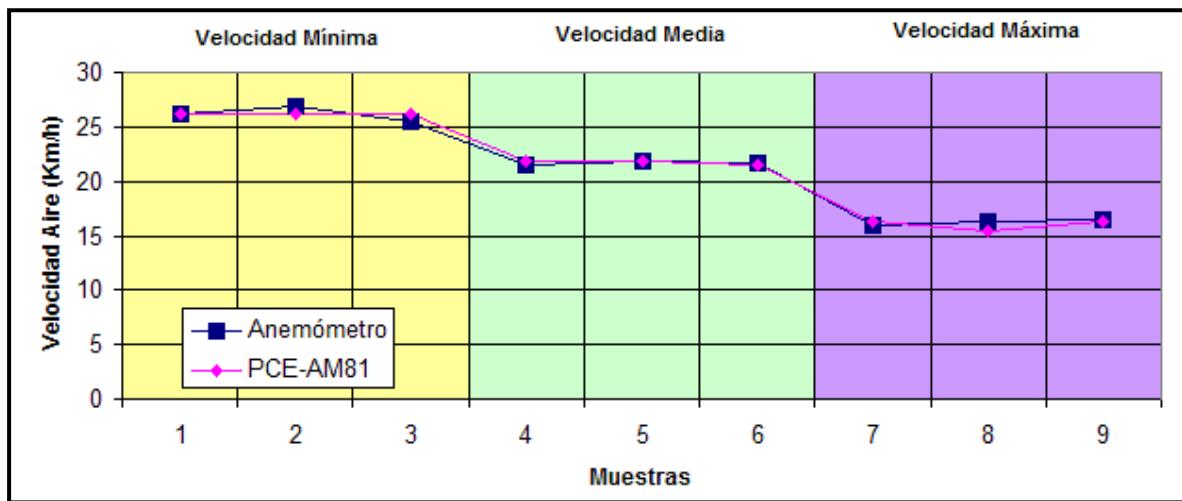


Figura 4.2.3 Medidas en diferentes condiciones del anemómetro respecto al PCE-AM81.

Los valores medidos deberían ser constantes para cada velocidad programada, pero ocurren variaciones debido al margen de error de medida de los aparatos, estas variaciones son muy pequeñas y se puede constatar que el anemómetro tiene una buena calibración.

4.2.4 Verificación Veleta

Para verificar el correcto funcionamiento de la veleta, se ha puesto la veleta apuntando a diferentes posiciones y se mide el valor que devuelve. Se compara con el valor teórico que debería salir. Hay que tener en cuenta que al poner la veleta en una posición ya cometemos un error puesto que no es exactamente la posición que supuestamente se está midiendo, el error humano es muy influyente en este experimento. Por lo que esta verificación siempre tendremos que tomarla con un error bastante elevado. De todas formas al tomar la variable dirección del viento siempre se toma un rango de valores, es decir entre un grado y otro, por lo que una pequeña variación no es significativa.

Valor Teórico	0°	45°	90°	135°	180°	225°	270°	315°
Valor Medido	358'6°	48'8°	87'2°	142'8°	185'7°	220'8°	271'8°	324'4°

Tabla 4.2.1 Diferentes medidas del anemómetro respecto al PCE-AM81.

Como se observa en la Tabla 4.2.1 Los valores obtenidos distan un poco de los valores ideales, pero es que poner la veleta en la posición ideal es prácticamente imposible, por lo que se unen dos errores el humano, poner la veleta, y el error propio de la veleta. Aún así son valores bastante aproximados.

4.2.5 Verificación Piranómetro

Para verificarlo no se puede utilizar ningún otro aparato de medida de los disponibles, ya que se precisa otro piranómetro, los cuales no se han podido adquirir por su elevado precio. El luxómetro, aparato de medida del que se dispone, no se puede utilizar puesto que no hay una relación directa entre lux y Watios. Esta relación existe cuando es conocida la longitud de onda de la luz a medir. En este caso el espectro luminoso es complejo y cambiante, por lo que no se podrá optar por este tipo de solución.

Para la verificación de este sensor se ha optado por los valores teóricos de radiación solar. Estos valores teóricos se han sacado de [12].

Para obtenerlos sólo se necesita saber los valores GPS de la zona en cuestión. Los valores GPS se obtienen in situ en la zona con un GPS o a través de Internet, con diferentes páginas webs, ésta última es la utilizada, se han obtenido de [13].

Una vez conocidos los valores GPS se harán los siguientes pasos:

- 1) Escribir las coordenadas GPS, latitud y longitud, en las pestañas a tal efecto.
- 2) Se pulsará sobre visualizar y se abrirá otra página.
- 3) En la última página se pulsará sobre la opción:

Visualizar [variación diaria de irradiancia](#).

Se abrirá otra página donde aparecerán los valores de radiación diaria de la zona en un determinado día genérico de un mes a elegir en un plano de inclinación a seleccionar.

- 4) Se elige el Mes, el plano de inclinación que se desea y se pulsara sobre “enviar”. En ese momento se podrán visualizar los datos teóricos de radiación de un día del mes elegido con un determinado plano de inclinación.

Una vez realizado, se tiene una serie de columnas con los datos de radiación diarios. Para poder compararlos con los datos medidos, se ha realizado un programa en Matlab que dibuja dos series de datos; una será la radiación medida a lo largo de un día y otra serán los datos teóricos.

A la hora de la verificación se hará de los 5 sensores de radiación solar que conforman el mapa de sombras y del piranómetro. Para obtener los valores teóricos medidos habrá que insertar los valores de inclinación de:

Inclinación de los módulos [0;90] **0** grado

Orientación de los módulos [-180;180](E:-90 s:0) **0** grado

El primer valor indica que es con inclinación horizontal y el segundo valor indica que es orientado al sur, en este caso no tendrá relevancia la orientación por ser un plano horizontal. En la figura 4.2.4 se puede observar el resultado obtenido.

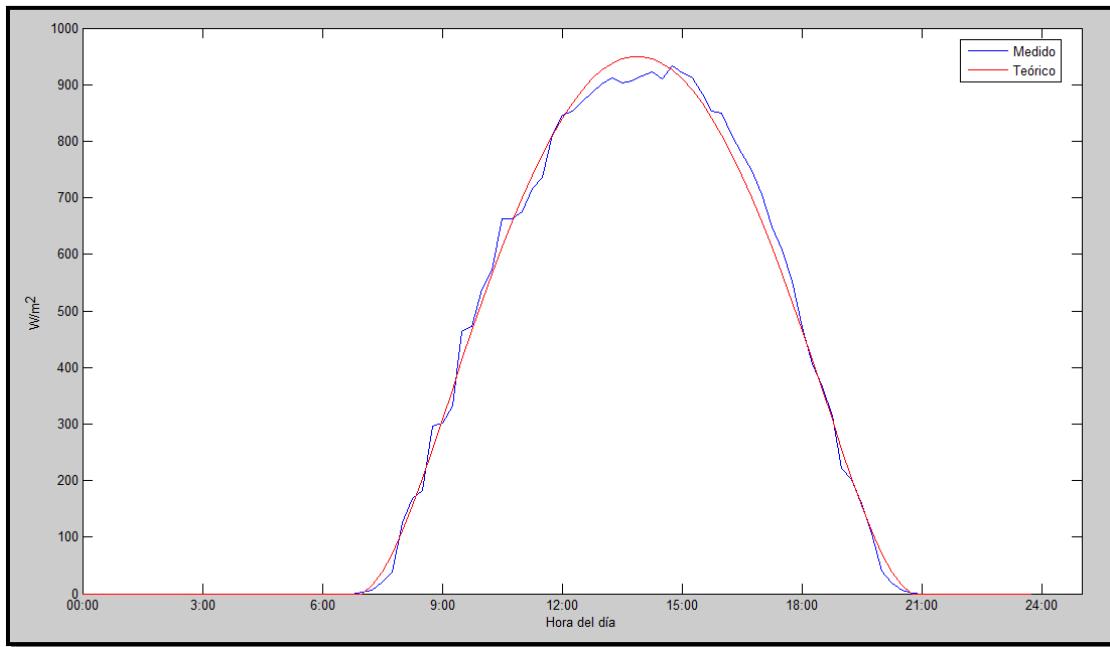


Figura 4.2.4 Radiación diaria, teórica y Medida, de un día de agosto.

Podemos observar que existe una pequeña variación entre datos teóricos y datos medidos. Existen unas pequeñas desviaciones con una radiación baja que se van incrementando conforme aumenta la radiación. Esto ocurre por el incremento de temperatura producido sobre los fotodioides, que produce una desviación que se incrementada a temperaturas mayores.

No obstante los resultados obtenidos son bastante aceptables para poder realizar un estudio aproximado de la radiación solar.

También habrá que realizar la verificación del piranómetro. Para ello hay que realizar los mismos pasos que para un sensor de radiación en el plano horizontal. La diferencia es que para obtener los valores teóricos de la radiación solar directa habrá que seleccionar la opción:

“Mostrar la irradiancia para un seguidor solar de 2 ejes”

Con esta opción se tiene la radiación solar directa perpendicular al sol en todo el momento del día. Cuando se seleccione esta opción aparecerá una columna nueva, 2 ejes, que habrá que usar a la hora de insertar los valores en el programa de Matlab.

Los resultados obtenidos se puede observar el la figura 4.2.5.

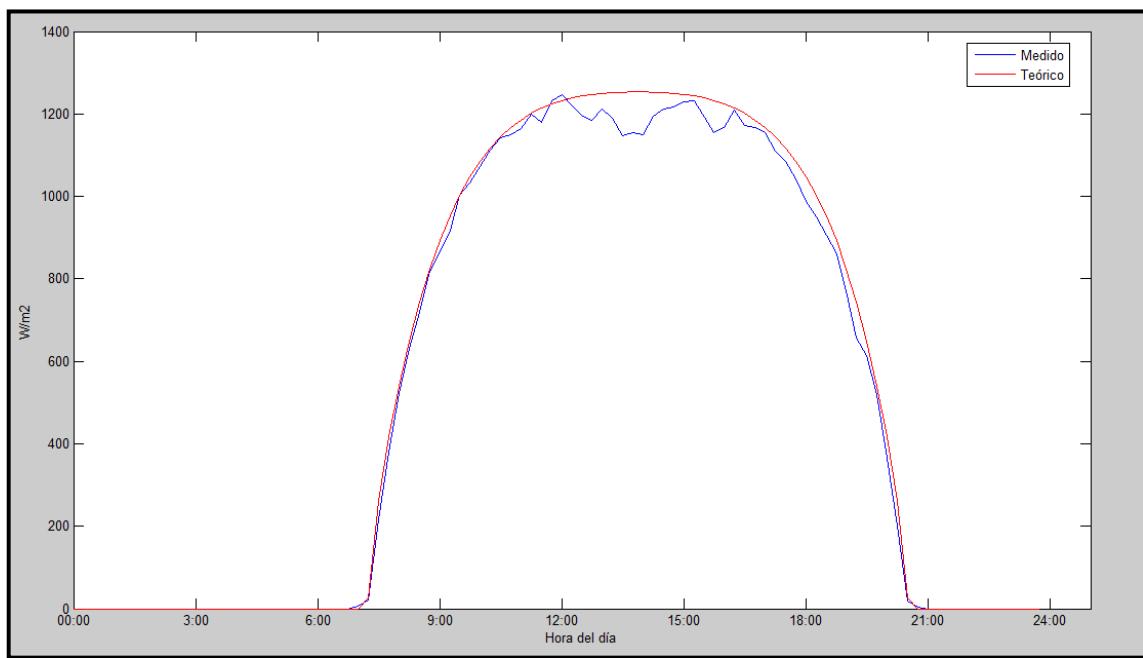


Figura 4.2.5 Radiación perpendicular diaria, teórica y Medida, de un día de agosto.

Se puede observar la diferencia entre los valores medidos y teóricos de la radiación solar, la mayor diferencia se observan cuando hay mayor radiación, este efecto ya ha sido comentado anteriormente, y es debido a la temperatura. Pero también se puede observar que la diferencia entre radiaciones elevadas y las más bajas son más claras que para el caso de un sensor en el plano horizontal.

Esto es debido a que las bajas radiaciones son a la hora del día donde el sol está en una posición en que incide con la misma inclinación sobre los fotodiodos, que hacen la aportación para calcular la potencia medida, entre las 12:00 y las 15:00 el sol se encuentra en un punto alto sobre el horizonte por lo que la aportación del sensor con orientación sur es muy baja comparados con otras horas del día en la que el sol se encuentra en una posición más baja con respecto al horizonte y además está en el sur, por lo que la aportación al valor medido de los sensores con orientación este y oeste será muy baja o nula. Sólo se mide la aportación de un solo sensor en la hora del día en la que hay mayor temperatura por lo que la desviación de la radiación medida será mayor. En contraposición sobre las 9:00 y las 18:00 el sol se encuentra en puntos más bajos sobre el horizonte y en una posición al este o al oeste dependiendo de la hora. En este caso la aportación de la radiación medida vendrá dada por la aportación de 3 fotodiodos, por lo que es más fácil que la desviación de un fotodiodo se compense con la desviación contraria de otro fotodiodo a la hora de calcular la radiación medida. Además son horas del día con menos temperatura por lo que la desviación producida por la temperatura será aún menor.

Por este motivo los datos resultantes tendrán este aspecto, en el que habrá mayor diferencia entre los valores teóricos y medidos a las horas donde el sol se encuentre en su posición más alto con respecto al horizonte y será menor o muy baja cuando el sol se encuentre más bajo en su posición respecto al horizonte.

Podemos observar que los datos obtenidos difirieren de los datos teóricos, esta diferencia es relevante a la hora de realizar un estudio de la zona, puesto que se puede observar que describe el comportamiento del sol con bastante fidelidad y además los valores que necesitamos no son los datos exactos de radiación sino la tendencia de estos datos. En este sentido los datos obtenidos sirven con bastante precisión a la hora de hacer el estudio de la zona.

4.3 Verificación GPRS

Se desea comprobar el correcto funcionamiento del módulo de memoria GPRS en el sistema, para ello se han realizado una serie de pruebas.

Primero se comprueba que el GPRS manda datos; sin los sensores, es decir sólo el Arduino el módulo de memoria y el de GPRS. El Arduino programado para que mande un archivo, éste se ha grabado previamente en el módulo de memoria y dentro del archivo se escribe unos caracteres aleatorios. Para ver cuales eran los valores del módulo GPRS se conecta el módulo al ordenador y con el hyperterminal se comprueba el valor de la señal del GPRS, para ello se escribe el comando “AT+CSQ”, la respuesta del módulo es 21, este valor representa que el módulo recibe la señal a una potencia de -109 a -53 dBm. Valores normales para un dispositivo GPRS. Se comprueba de forma afirmativa que ha enviado el archivo creado de forma correcta.

Después guardamos en la memoria un archivo con los datos recogidos de un día cualquiera, y se configura el Arduino para que lo transmita, para ver si el formato especial que tiene el archivo se transmite con normalidad, se comprueba si efectivamente transmite el archivo con el formato adecuado. Por último se comprueba la transmisión consecutiva de archivo de días de captación para comprobar la transmisión en una concatenación de archivos. Después de todo el proceso se observa que no existen problemas de transmisión.

4.4 Verificación módulo MicroSD

La verificación de la memoria MicroSD se realiza comprobando que se guardan los archivos necesarios. Primero se comprueba que se guardan caracteres en la tarjeta MicroSD, después de comprobar que se guardan los caracteres se conecta el Arduino con el módulo de memoria MicroSD para guardar en la tarjeta los datos diarios.

Una vez comprobado que se guardan los datos en el archivo correspondiente, se realizan todas las acciones que se pueden realizar sobre el módulo de memoria, es decir se guardan, se leen y se borran datos. El borrado de memoria no es necesario incluirlo en el programa del proyecto, pero es conveniente comprobar que funciona la función implementada para tal efecto.

Por último se genera un código en el Arduino para guardar diferentes datos en diferentes archivos, a fin de comprobar que se puede guardar en diferentes archivos de la memoria. Se comprueba que las funciones implementadas para actuar sobre el módulo de memoria funcionan correctamente.

4.5 Verificación sistema de alimentación

Con el fin de comprobar el sistema de alimentación se hacen las pruebas pertinentes. Primero se conecta el sistema de alimentación y se comprueba que el sistema está alimentado, cuando se certifica que el sistema de alimentación funciona correctamente, se realizan diferentes pruebas.

La primera prueba determina cuantos días de oscuridad, sin alimentación del panel solar, se pueden tener en el sistema. En este caso se han obtenido 2 días y medio de días de oscuridad total. Esta situación puede ocurrir, por lo tanto se comprueba que sucede cuando se produce. Lo sucedido es que el archivo del día que se pierde la alimentación habrá que descártalo al igual que el día que se recupera la alimentación y todos los días intermedios, pero a partir de ese día el sistema hardware se recupera sin ningún problema.

La segunda prueba es mandar datos hasta que se agote la batería, sin la alimentación de la placa solar, se genera un archivo con una extensión de más de medio Gigabyte y se comprueba cual es el tiempo máximo de transmisión y cual es la capacidad del archivo enviado. El sistema tarda en apagarse un poco menos de 3 horas (2 horas 53 minutos) y se envía un archivo de un poco más de 50 Megabyte (53774kB).Lo primero que se comprueba es que el archivo enviado no se guarda de forma correcta ya que no se puede abrir. Por lo que cualquier archivo mandado debe tener como máximo 50MB. Para que se asegure el envío de forma correcta. Aún siendo esta situación una que rara vez se va a dar, puesto que se ha escogido mandar datos a medio día, que es la situación que la placa solar más potencia va a dar, así no habrá ningún problema con la alimentación. Aún así se restringe en tamaño máximo del archivo a mandar a 50MB.

4.6 Verificación sistema hardware completo

La verificación del sistema hardware completo se realiza en diferentes pasos. Primero integramos todo nuestro sistema, una vez realizado se deja una hora, en la cual se manden datos de un día al servidor FTP. Simultáneamente se cogen, con la instrumentación externa, los datos climáticos de la situación donde se emplace el sistema. Pasada la hora se accede al archivo en el servidor FTP y se abre el archivo; en este paso se comprueba que el módulo de memoria y el GPRS funcionan correctamente. Con los datos recogidos con la instrumentación externa se comprueban si los registrados son los valores adecuados; se comprueban que los datos son los correctos.

Una vez comprobado que el sistema funciona correctamente se realizan pruebas marginales, es decir, en días donde las condiciones climatológicas sean adversas. En días muy cálidos y en días fríos. Las pruebas marginales deberían ser comprobadas en situaciones límites del microcontrolador, los módulos y los sensores, el problema es que no se disponen de situaciones climatológicas tan extremas. Aún así se comprueban en días muy cálidos y fríos comprobando que el sistema funciona correctamente.

Por último el sistema se deja una serie de días para comprobar que todo funciona con normalidad, a la vez que se realizan las pruebas de verificación se recogen los datos necesarios para realizar un ejemplo de la aplicación, especificada en el capítulo 6.

Capítulo 5

Herramienta software

Para poder visualizar e interpretar los datos recogidos se ha realizado una aplicación software en Matlab debido a que los datos recogidos por parte del microcontrolador son difícilmente legibles, como se ha visto anteriormente.

En esté capítulo se explica el entorno empleado para el desarrollo de la aplicación software, se ha generado en Matlab por ser una herramienta potente de cálculo y representación gráfica, además se explica el modo de utilización y se describe el proceso de desarrollo empleado para el correcto funcionamiento de la herramienta.

5.1 Matlab

Según [14], **MATLAB** (abreviatura de *MAT*rix *LAB*oratory, "laboratorio de matrices") es un software matemático que ofrece un entorno de desarrollo integrado (IDE) con un lenguaje de programación propio (lenguaje M). Está disponible para las plataformas Unix, Windows y Apple Mac OS X.

Entre sus prestaciones básicas se hallan: la manipulación de matrices, la representación de datos y funciones, la implementación de algoritmos, la creación de interfaces de usuario (GUI) y la comunicación con programas en otros lenguajes y con otros dispositivos hardware. El paquete MATLAB dispone de dos herramientas adicionales que expanden sus prestaciones, a saber, Simulink (plataforma de simulación multidominio) y GUIDE (editor de interfaces de usuario - GUI). Además, se pueden ampliar las capacidades de MATLAB con las *cajas de herramientas* (*toolboxes*); y las de Simulink con los *paquetes de bloques* (*blocksets*).

Es un software muy usado en universidades y centros de investigación y desarrollo. En los últimos años ha aumentado el número de prestaciones, como la de programar directamente procesadores digitales de señal o crear código VHDL.

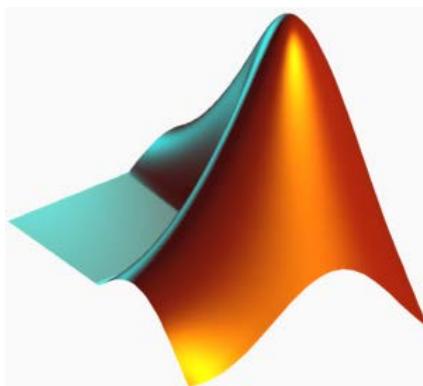


Figura 5.1.1 Logotipo del Matlab.

Sus características principales son:

- Lenguaje de alto nivel para cálculo técnico.
- Entorno de desarrollo para la gestión de código, archivos y datos.
- Herramientas interactivas para exploración, diseño y resolución de problemas iterativos
- Funciones matemáticas para álgebra lineal, estadística, análisis de Fourier, filtraje, optimización e integración numérica
- Funciones gráficas bidimensionales y tridimensionales para visualización de datos.
- Herramientas para crear interfaces gráficas de usuario personalizadas.
- Funciones para integrar los algoritmos basados en MATLAB con aplicaciones y lenguajes externos, tales como C/C++, FORTRAN, Java, COM y Microsoft Excel.

5.2 Programación software

Desarrollar una aplicación software es una acción indispensable para poder analizar los datos recogidos por la herramienta hardware. Se ha realizado mediante el interfaz gráfico de Matlab, GUI, la cual es una prestación básica, para la creación de interfaces. En la figura 5.2.1 se observa la pantalla de inicio de la GUI.

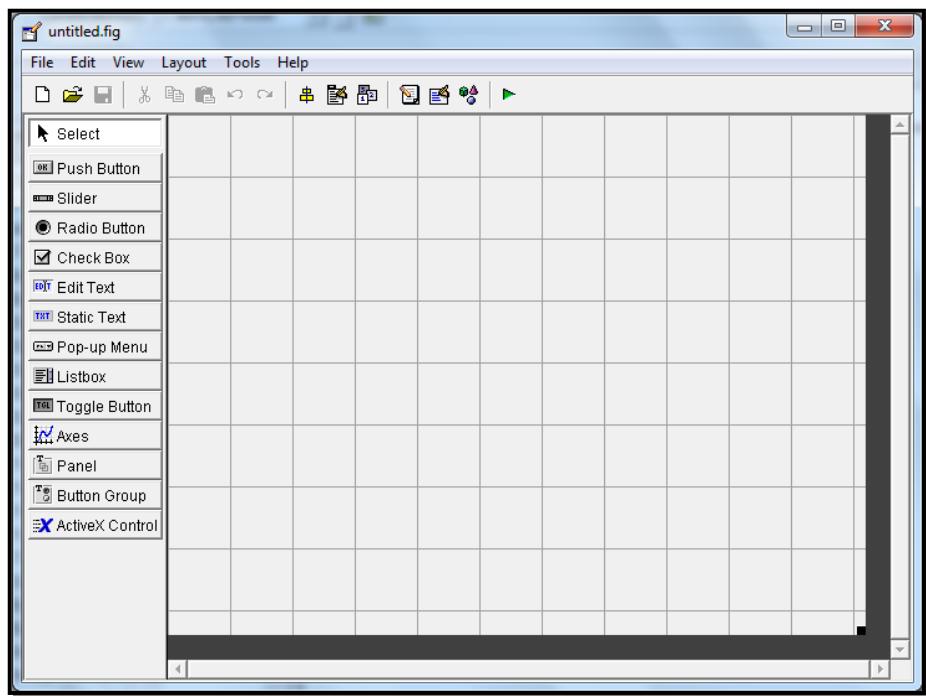


Figura 5.2.1 Pantalla de inicio de la aplicación GUI del Matlab.

Esta aplicación es una herramienta de Matlab para crear un interfaz gráfico con el que poder representar y realizar acciones dependiendo de una serie de variables a introducir.

Permite poner una serie de iconos de diferente tipología, iconos de pulsación, iconos de selección, desplegables, etc. Estos iconos generan unas variables que serán utilizadas para realizar diferentes acciones. Además dispone de la opción de introducir texto para que el interfaz se comprensible y autoexplicable. Todo ello para realizar una serie de acciones, desde interactuar con puertos del ordenador a realizar operaciones aritméticas complejas, también se puede, y es el caso que ocupa el proyecto, realizar diferentes tipos de gráficas.

Para realizar la herramienta software se ha optado por:

- Representar dos graficas simultáneamente en unos casos y una en otros.
- Poder seleccionar el día a representar insertando el día, mes y el año.
- Tener cuatro paciones donde represente diferentes gráficas.
- Escribir una serie de textos para entender el interfaz.

La GUI que se ha creado para poder hacer la herramienta software se puede observar en la figura 5.2.2.

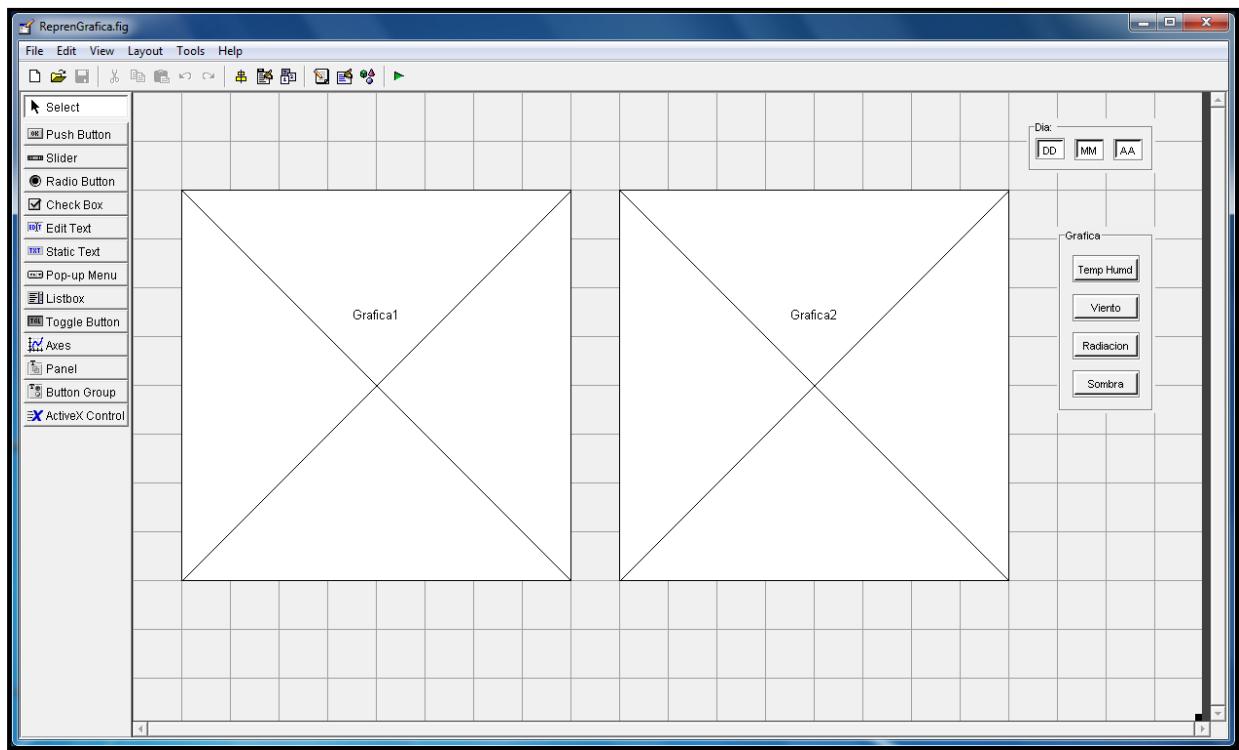


Figura 5.2.2 Pantalla de la aplicación GUI.

Esta pantalla genera un archivo .m en Matlab en el cual está las diferentes opciones y variables generadas en la GUI. Este archivo se llama igual que el nombre asignado a la GUI. En este punto habrá que definir que hacer para las diferentes variables y opciones definidas y programar las gráficas a implementar.

El archivo consta de cuatro zonas a programar que representan las cuatro opciones implementadas en la GUI. En cada opción la primera acción que se realiza es cargar el archivo del día seleccionado. El código implementado para dicha acción es:

```
A=get(handles.edit1,'string');
B=get(handles.edit2,'string');
C=get(handles.edit3,'string');
S=load([A '-' B '-' C '.txt']);
```

Donde A será el valor del día introducido en DD, B el valor del mes y C el valor del año. Despues habrá que programar cada opción que será la representación gráfica de los datos obtenidos.

5.2.1 Código Matlab

En la primera opción se representa la temperatura y la humedad para ello el código implementado para dicha función será:

```
[a,b]=size(S);
for i=1:1:a
    Hora(i)=S(i,1);
    Min(i)=S(i,2);
    Temp1(i)=S(i,3);
    Temp2(i)=S(i,4);
    Humd(i)=S(i,5);
    tiempoChar=[int2str(Hora(i)),':',int2str(Min(i))];
    [s,t]=size(tiempoChar);
    Tiempo2(i)=Hora(i)+Min(i)/60;
end
axes(handles.Grafical)
set(handles.Grafical,'position',[15 29 160 15 ]);
plot(Tiempo2,Temp1)
title('Temperatura a lo largo del día');
set(gca,'XTick',0:3:24);
set(gca,'XTickLabel',{'00:00','3:00','6:00','9:00','12:00','15:00',...
    '18:00','21:00','24:00'});
Ylim([min(Temp1)-10 max(Temp1)+10]);
Ylabel('Temp °C');
Xlabel('Hora del día');
zoom ON;
axes(handles.Grafica2);
set(handles.Grafica2,'position',[15 8 160 15 ]);
plot(Tiempo2,Humd,'r')
title('Humedad relativa a lo largo del día');
set(gca,'XTick',0:3:24);
set(gca,'XTickLabel',{'00:00','3:00','6:00','9:00','12:00','15:00',...
    '18:00','21:00','24:00'});
Ylim([0 100]);
Ylabel('Humedad %');
Xlabel('Hora del día');
zoom ON;
```

El código representa dos graficas, la primera representa la temperatura en la parte superior y la segunda la humedad en la parte inferior. La función de representación de la gráfica es la típica función Plot de Matlab, simplemente habrá que añadirle diferentes textos a la gráfica, incluido el nombre de las coordenadas.

En la segunda opción se representa la velocidad y dirección del viento para ello el código implementado para dicha función será:

```
[a,b]=size(S);
for i=1:1:a
    Dire(i)=S(i,6);
    Vel(i)=S(i,7);
end
DireRad=Dire*pi/180;
[XVel,YVel] = pol2cart(DireRad,Vel);
axes(handles.Grafical)
zoom ON;
set(handles.Grafical,'position',[10 10 80 30 ]);
Numpart=12;
[R T]=rose(DireRad,Numpart)
```

```

H=rose(DireRad,Numpart);
hline = findobj(gca,'Type','line');
set(hline,'LineWidth',2);
title('Dirección del viento (%)');
rotate(H,[0 0 1],270);
axes(handles.Grafica2);
zoom ON;
set(handles.Grafica2,'position',[100 10 80 30 ]);
H=compass(XVel,YVel);
hline = findobj(gca,'Type','line');
set(hline,'LineWidth',1.5);
title('Velocidad del viento (km/h)');
rotate(H,[0 0 1],270);

```

El código representa la dirección y la velocidad del viento en dos gráficas diferenciadas. Se ha utilizado dos funciones que representan los valores en polares, rose y compass, ambas funciones representan los valores de forma diferentes. La primera representa el porcentaje de valores dentro de un rango de ángulos y la segunda representa el módulo y la fase de los diferentes valores. Como estas funciones representan los 0º a la derecha de la gráfica y para el presente proyecto se ha decidido que el sur son los 0º, se ha rotado la representación de la gráfica para que se visualice los 0º en la parte inferior de la gráfica y así sea una representación real de la dirección del viento.

En la tercera opción se representa la radiación solar global en el día determinado, se ha realizado con el trozo de código que se ve a continuación:

```

[a,b]=size(S);
for i=1:1:a
    Hora(i)=S(i,1);
    Min(i)=S(i,2);
    Hor(i)=S(i,13);
    Sur(i)=S(i,14);
    Este(i)=S(i,15);
    Norte(i)=S(i,16);
    Oeste(i)=S(i,17);
    Tiempo2(i)=Hora(i)+Min(i)/60;
end
for i=1:1:length(Norte)
    if (Este(i)>=Oeste(i))
        Terc(i)=Este(i);
    else
        Terc(i)=Oeste(i);
    end
    RadG(i)=sqrt(Hor(i)*Hor(i)+Sur(i)*Sur(i)+Terc(i)*Terc(i));
    RadG(i)=RadG(i)+Norte(i);
end
axes(handles.Grafical);
set(handles.Grafical,'position',[1100 1000 80 30 ]);
axes(handles.Grafica2);
set(handles.Grafica2,'position',[20 7 150 35 ]);
plot(Tiempo2,RadG);
ylabel('Potencia solar: W/m2');
xlabel('Hora del día');
set(gca,'XTick',0:3:24);
set(gca,'XTickLabel',{'00:00','3:00','6:00','9:00','12:00','15:00',...
    '18:00','21:00','24:00'});
title('Radiacion solar global y perpendicular');
zoom ON;

```

En este código se ha programado una sola gráfica, representa la radiación, se utiliza la función Plot aunque para ello habrá que calcular la potencia solar directa, ya que los datos cargados al código representa la radiación solar en los cuatro puntos cardinales y el horizonte, para que la segunda no se visualice se ha representado fuera de las coordenadas de la pantalla.

La cuarta opción representa la evolución temporal del mapa de sombras de la zona de estudio, el código que se ha implementado para este fin ha sido:

```
[a,b]=size(S);
for i=1:1:a
    Hora(i)=S(i,1);
    Min(i)=S(i,2);
    Luz1(i)=S(i,8);
    Luz2(i)=S(i,9);
    Luz3(i)=S(i,10);
    Luz4(i)=S(i,11);
    Luz5(i)=S(i,12);
    Luz6(i)=S(i,13);
    if (Hora(i)<10)&(Min(i)==0)
        tiempoChar=['0',int2str(Hora(i)),':',int2str(Min(i)),'0'];
    elseif (Hora(i)<10)&(Min(i)>0)
        tiempoChar=['0',int2str(Hora(i)),':',int2str(Min(i))];
    elseif (Hora(i)>=10)&(Min(i)>0)
        tiempoChar=[int2str(Hora(i)),':',int2str(Min(i))];
    elseif (Hora(i)>=10)&(Min(i)==0)
        tiempoChar=[int2str(Hora(i)),':',int2str(Min(i)),'0'];
    end
    for j=1:1:length(tiempoChar)
        Tiempo(i,j)=tiempoChar(j);
    end
    Tiempo2(i)=Hora(i)+Min(i)/60;
end
axes(handles.Grafica2);
set(handles.Grafica2,'position',[1000 1000 70 30 ]);
for n=29:1:length(Tiempo2)-11
    Bluz(1,1)=Luz5(n);Bluz(1,2)=Luz5(n);Bluz(1,3)=(Luz5(n)+Luz6(n))/2;
    Bluz(1,4)=Luz6(n);Bluz(1,5)=(Luz6(n)+Luz4(n))/2;Bluz(1,6)=Luz4(n);Bluz(1,7)=Luz4(n);

    Bluz(2,1)=Luz5(n);Bluz(2,2)=Luz5(n);Bluz(2,3)=(Luz5(n)+Luz6(n))/2;
    Bluz(2,4)=Luz6(n);Bluz(2,5)=(Luz6(n)+Luz4(n))/2;Bluz(2,6)=Luz4(n);Bluz(2,7)=Luz4(n);

    Bluz(4,1)=Luz1(n);Bluz(4,2)=Luz1(n);Bluz(4,3)=(Luz1(n)+Luz2(n))/2;
    Bluz(4,4)=Luz2(n);Bluz(4,5)=(Luz2(n)+Luz3(n))/2;Bluz(4,6)=Luz3(n);Bluz(4,7)=Luz3(n);

    Bluz(5,1)=Luz1(n);Bluz(5,2)=Luz1(n);Bluz(5,3)=(Luz1(n)+Luz2(n))/2;
    Bluz(5,4)=Luz2(n);Bluz(5,5)=(Luz2(n)+Luz3(n))/2;Bluz(5,6)=Luz3(n);Bluz(5,7)=Luz3(n);

    Bluz(3,1)=(Bluz(2,1)+Bluz(4,1))/2;Bluz(3,2)=(Bluz(2,2)+Bluz(4,2))/2;
    Bluz(3,3)=(Bluz(2,3)+Bluz(4,3))/2;
    Bluz(3,4)=(Bluz(2,4)+Bluz(4,4))/2;Bluz(3,5)=(Bluz(2,5)+Bluz(4,5))/2;
    Bluz(3,6)=(Bluz(2,6)+Bluz(4,6))/2;Bluz(3,7)=(Bluz(2,7)+Bluz(4,7))/2;
    Res=interp2(Bluz,1);
    %Res2=interp2(Bluz,5,'spline');
    Res3=interp2(Res,2,'spline');
```

```

[xT,yT]=size(Tiempo);
for nH=1:1:yT
    PonerHora(nH)=Tiempo(n,nH);
end
axes(handles.Grafical);
set(handles.Grafical,'position',[15 10 150 30]);
[Q,W]=size(Res3);mesh(Res3)
axis([0 W 0 Q 0 1000 0 1000])
set(gca,'XTick',0:W/6:W);
set(gca,'XTickLabel',0:30/6:30);
set(gca,'YTick',0:Q/4:Q);
set(gca,'YTickLabel',0:20/4:20);
xlabel('Longitud (m)');
ylabel('Longitud (m)');
zlabel('Potencia solar: W/m2');
text(20,50,900,['Hora: ',PonerHora,]);
title('Potencia solar en una extensión de terreno');
%k = waitforbuttonpress
pause(1);
end

```

Este código se ha vuelto a programar una sola gráfica, la segunda se ha programado para que no aparezca en pantalla. La gráfica representará la evolución temporal de las sombras de la zona. Como sólo se disponen de 6 puntos de representación se ha realizado una serie de operaciones para que represente una superficie.

Cada punto de representación se convierte en 9 puntos que conforman un cuadrado, donde los puntos centrales son los datos adquiridos y los puntos que conforman las aristas son la media con los puntos adyacentes. Se puede observar en la figura 5.2.3.

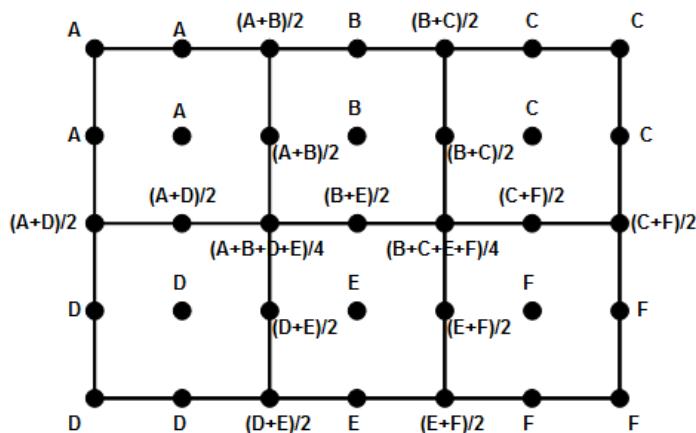


Figura 5.2.3 Puntos de representación de la zona

Para completar la representación de la superficie se realiza una interpolación de los datos de una superficie con la función Matlab *interp2*. Se ha interpolado 2 veces, con dos variables diferentes. Una primera vez añadiendo otro punto de forma lineal y otra segunda vez añadiendo 2 puntos más de forma cuadrática. El número de puntos de la interpolación no se podrá incrementar a más de 5, puesto que el programa tardará demasiado tiempo en la representación

5.2.2 Creación de ejecutables

Para concluir la creación de la aplicación software se pretende que se autoejecutable, es decir que no se precise del programa Matlab para poder disponer de la herramienta creada, puesto que no todos los usuarios de la aplicación dispondrán de este programa. A tal fin se ejecuta un comando en el Matlab para convertir el archivo .m, el archivo donde se programa, y el .fig que es el que genera la GUI en un archivo ejecutable.

El comando es: `mcc .-m RepresenGrafica.m RepresenGrafica.fig`

El número de archivos que aparecen son 4:

RepresenGrafica.ctf
 RepresenGrafica
 represengrafica_main.c
 represengrafica_mcc_component_data.c

Para poder ejecutar bien el programa, éste debe estar en la misma carpeta en la que se encuentren todos los archivos de los datos con los diferentes días guardados en el servidor FTP por el sistema hardware.

5.3 Interfaz de la aplicación

Una vez ejecutado el programa la pantalla de inicio se muestra en la figura 5.3.1.

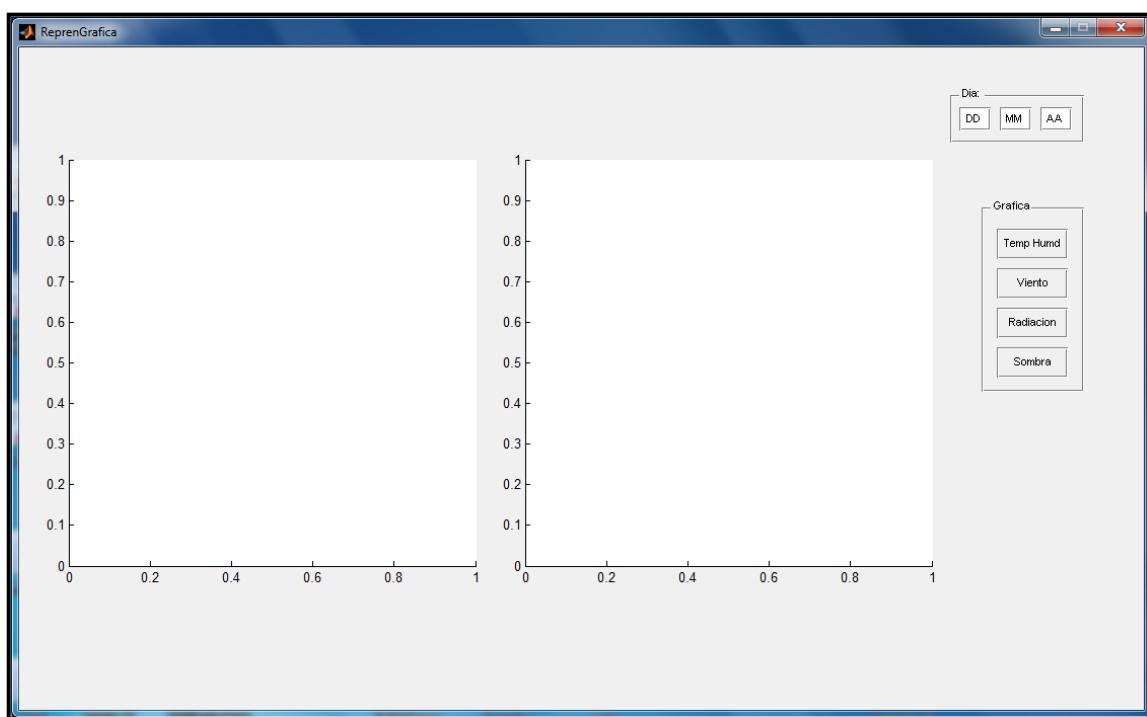


Figura 5.3.1 Pantalla inicial del programa en Matlab.

En esta primera pantalla se debe elegir el día que se quiere visualizar para ello hay que introducir el día en el cuadro “Día”:



En este cuadro debemos poner el día, el mes y el año. Todos deben de ser representados con 2 dígitos de esta forma si queremos visualizar el día uno de agosto del 2010 debemos poner 01 en el recuadro DD, 08 en el recuadro MM y 10 en el recuadro AA. Una vez introducido el día que vamos a visualizar existen 4 opciones que corresponden a gráficas diferentes.

La primera opción es: Temp Humd

En esta opción se mostrarán 2 gráficas diferentes, en la primera se podrá observar la evolución de la temperatura a lo largo del día seleccionado, y en la segunda se puede ver la humedad del mismo día, como se puede observar en la figura 5.3.2.

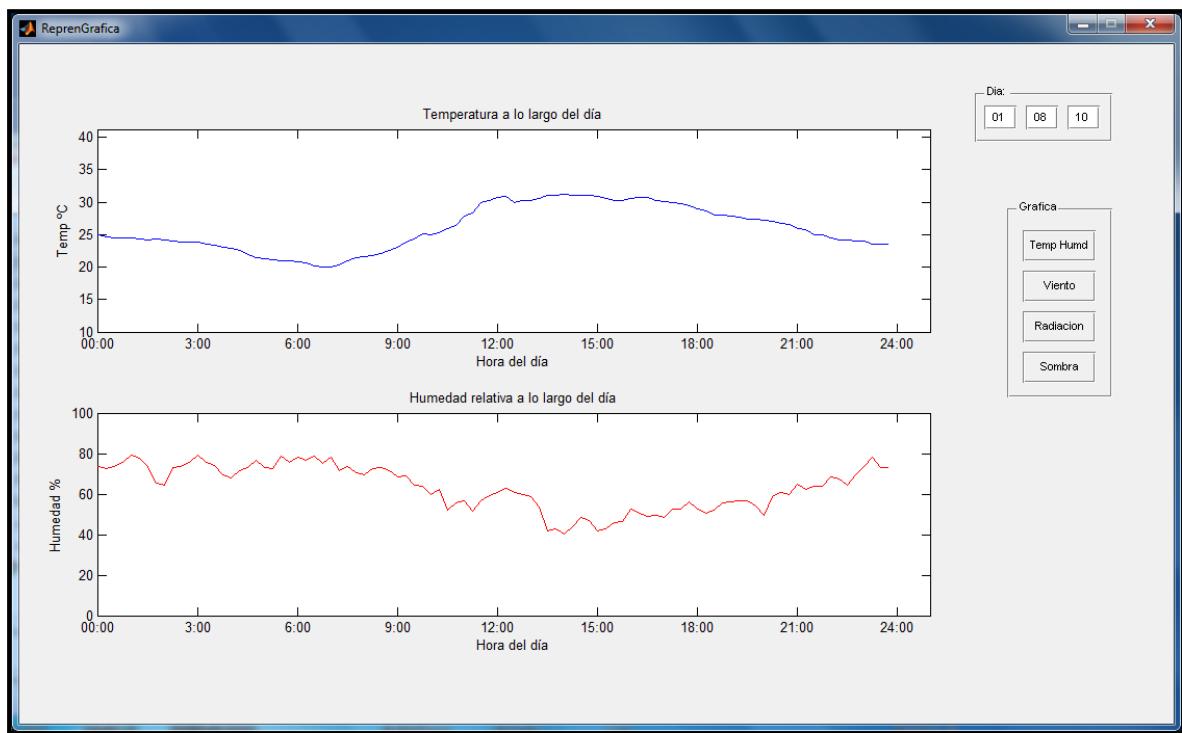


Figura 5.3.2 Representación de los valores de temperatura y humedad.

En esta primera pantalla se puede observar 2 gráficas, en la primera gráfica se puede ver la evolución de la temperatura, a lo largo del día. Hay que tener en cuenta que los valores son adquiridos cada 15 minutos y el rango de valores de representación de la temperatura variará dependiendo de la temperatura máxima y mínima del día.

En la segunda gráfica se observa la humedad relativa del día entero, en esta segunda se ha optado por representarla con todos los valores posibles de humedad, es decir el eje de ordenadas, que representa el valor de la humedad, incluye desde el mínimo valor posible 0% hasta el máximo valor posible 100%.

La siguiente opción de representación, corresponde con gráficas diferentes. Se seleccionan pulsando sobre:

En esta opción se muestran otras 2 gráficas diferentes, que son las variables obtenidas del viento, velocidad y dirección.

Las gráficas se pueden observar en la figura 5.3.3. La primera (Izquierda) representa la dirección del viento, la segunda representa la velocidad.

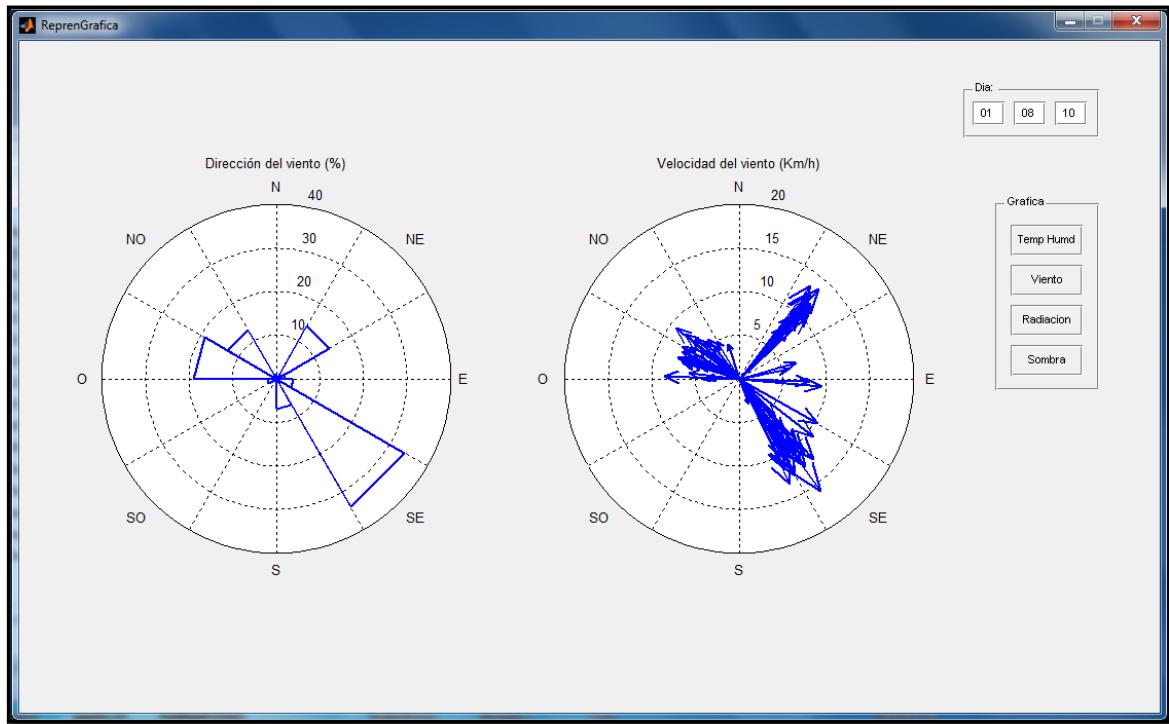


Figura 5.3.3 Representación de los valores de temperatura y humedad.

La primera gráfica, la dirección del viento, representa el porcentaje diario de la dirección del viento. Es decir hace una estimación del tiempo que el viento “sopla” en una determinada dirección. Con ella se estima la dirección predominante del viento en un determinado día. No se representa la evolución diaria de la dirección del viento puesto que según los estudios climáticos es importante saber que tipo de viento hay en un determinado día, no saber la evolución de este a lo largo del día.

En la segunda gráfica, velocidad del viento, representa la velocidad en km/h de las diferentes muestras obtenidas, según su dirección. Representa todas las muestras obtenidas superponiéndose dos muestras iguales, pero la gráfica representa una estimación de la velocidad según una dirección, por lo que visualmente es así como podemos ver la velocidad del viento cuando “sopla” en una determinada dirección.

Se podría haber realizado una representación de la velocidad a lo largo del día, pero los estudios climáticos estiman que es más relevante este tipo de gráficas.

La siguiente opción se muestra pulsando:

En esta opción se muestra una sola grafica, que será la representación de la radiación solar en watos por metros cuadrados sobre una superficie perpendicular a dicha radiación a lo largo de un día.

Dicha representación de esta variable se puede ver en la figura 5.3.4.

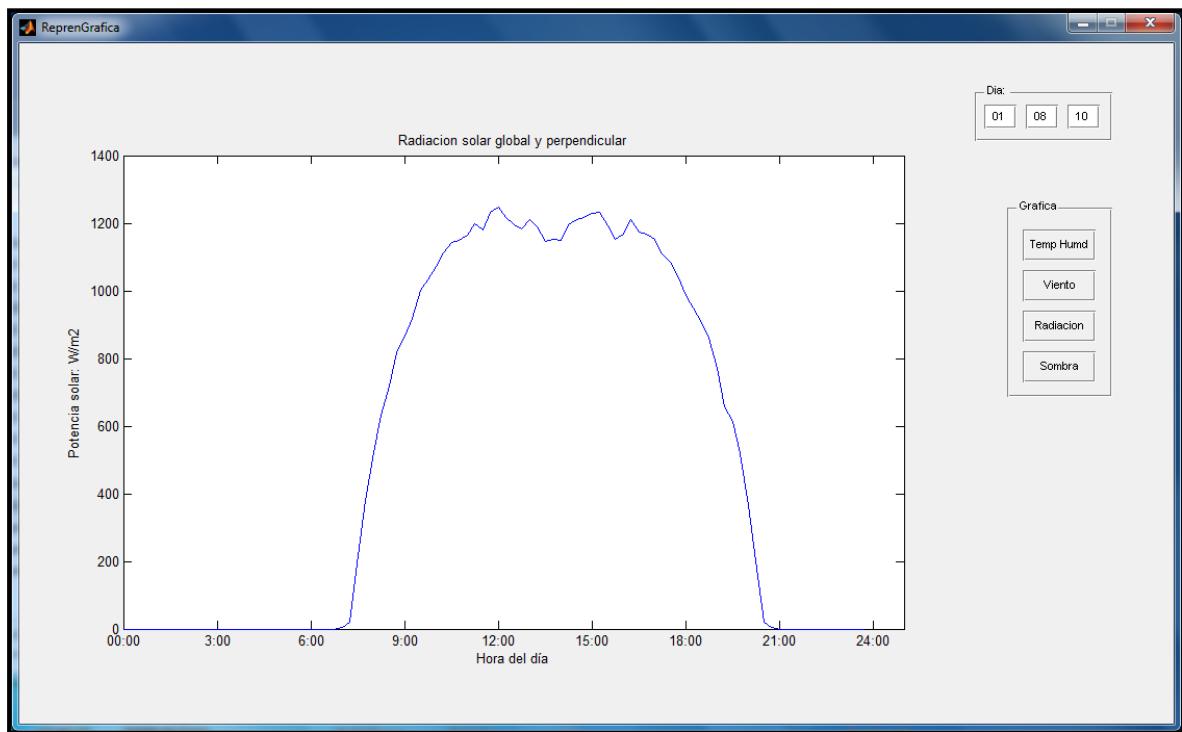


Figura 5.3.4 Representación de la radiación solar perpendicular.

La última alternativa que hay en el cuadro de opciones es pulsando sobre:

En dicha opción se representa la evolución temporal de las sombras proyectadas sobre la zona de estudio. Es decir se hace un análisis temporal de las sombras acaecidas sobre dicha zona. Cada Zona de estudio será diferente así que habrá que modificar el programa para adaptarlo a cada una.

La representación de este mapa de sombras se hace mediante una gráfica como se muestra en la figura 5.3.5. Esta representa la distribución de la radiación en la zona de estudio. Hay 6 puntos de recogida de datos, por lo que para hacer la representación gráfica se han interpolado los demás valores de la gráfica.

La grafica es una evolución temporal por lo que cambiará cada cierto tiempo para representar la evolución temporal. Se modificará cada pocos segundos para hacer una representación de la evolución temporal. Hay que tener en cuenta que hasta que no termina dicha evolución temporal no se podrá parar la representación del mapa de sombras.

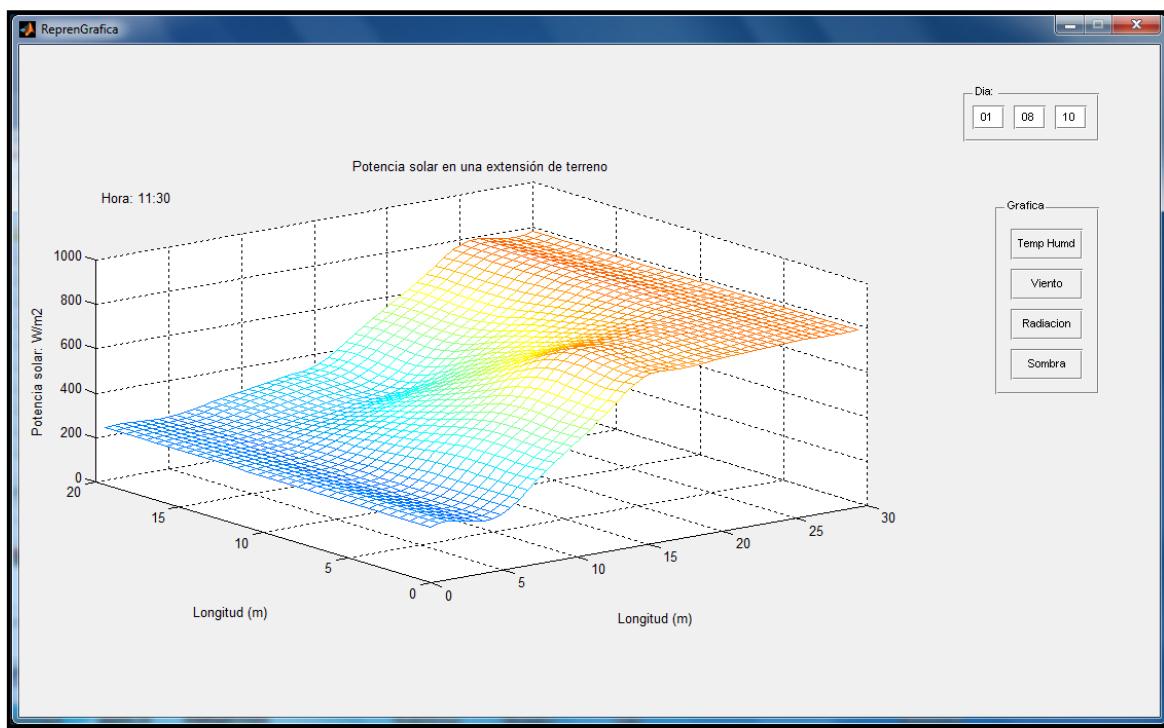


Figura 5.3.5 Representación del mapa de sombras.

La evolución temporal se ha representado desde las 7:00 hasta las 21:00, entendiendo que el resto del tiempo la radiación solar sobre la zona será mínima, prácticamente cero, por lo que no es necesario representar el mapa de sombras dicha franja horaria.

Se puede observar parte de esa evolución temporal en la gráfica 5.1.5.

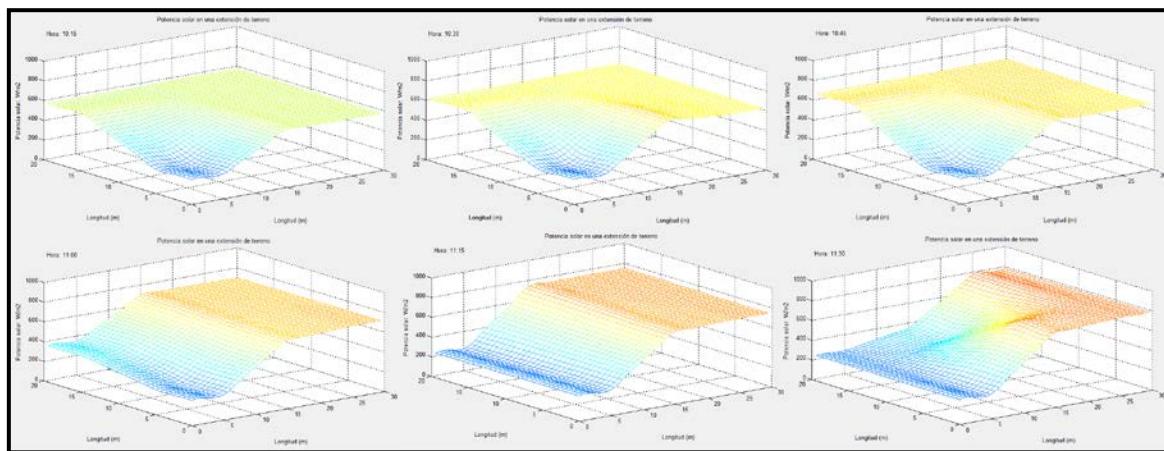


Figura 5.3.6 Evolución temporal del mapa de sombras.

5.4 Verificación de la aplicación software

En la verificación de la aplicación software se han realizado diferentes pruebas. Las primeras pruebas se han realizado mientras se realizaba la aplicación, para ver cómo funcionaba, por lo que al hacer la verificación final ya se habían realizado numerosas pruebas de funcionamiento y se esperaba una verificación sin problemas.

La prueba principal de funcionamiento es representar las diferentes gráficas de un día seleccionado y analizar si la representación hecho de ese día coincide con los valores teóricos. Se comprueba que la aplicación realiza el proceso correcto de representación.

Otras pruebas realizadas son las que se han realizado para comprobar el funcionamiento de la aplicación si el documento que contiene los datos no está completo, es decir se ha reiniciado el sistema en algún momento o se ha perdido la alimentación o algún incidente que ha provocado que los datos no se hayan grabado en la tarjeta MicroSD o que en el envío de datos haya habido algún error. En este caso hay dos situaciones la primera, en la que los datos están “cortados” en la misma línea de datos, es decir no se han grabado alguna variable en una misma hora, o la segunda, en la que los datos se han “cortado” de una hora del día a la siguiente, es decir se ha producido un error de una línea a la siguiente. Se puede observar en las figuras 5.4.1 y 5.4.2.

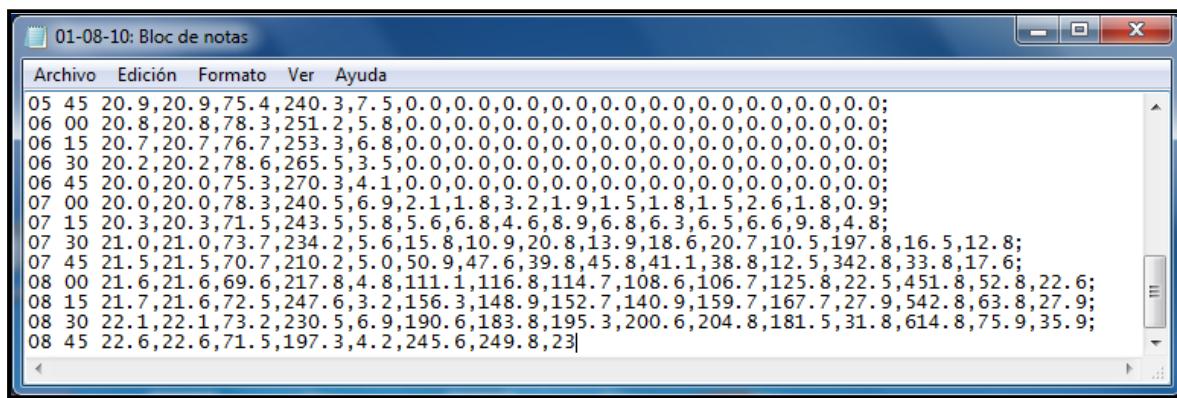


Figura 5.4.1 Situación 1 en archivo incorrecto.

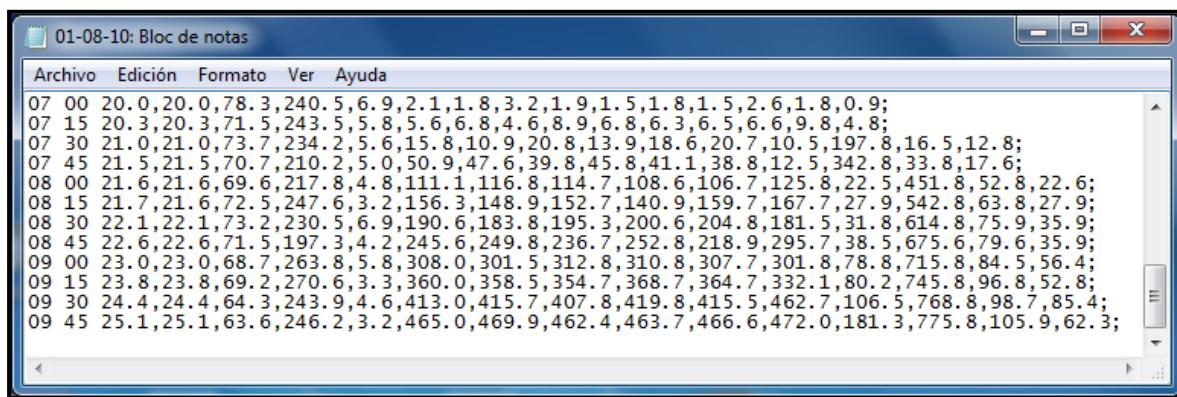


Figura 5.4.2 Situación 2 en archivo incorrecto.

En la situación número 1 no se ha podido realizar ningún programa o aplicación en Matlab para poder leer ese archivo, puesto que el Matlab dará un error.

“Error while evaluating uicontrol Callback”

Dicho error es producido porque la matriz creada debe tener las mismas columnas por fila. No se ha conseguido ninguna solución para ésta situación.

En la situación numero 2 se ha podido realizar una aplicación para completar el archivo con el número adecuado de filas para que el programa creado pueda interpretarlo adecuadamente. El problema es que algunas gráficas estarán incompletas y otras no tendrán sentido. Las graficas de temperatura, humedad y radiación estarán incompletas en las horas donde no se han grabados datos, puesto que en esas horas se ha completado con valores de 0. Las gráficas de viento, tanto velocidad como dirección serán erróneas puesto que estas gráficas representan un estadísticos y completando con 0, los estadísticos son erróneos. La gráfica de sombras no dará valores erróneos pero carecerán de sentido, puesto que no da una idea global de la zona, que es su cometido.

Por lo tanto cuando se ha producido algún error en el archivo, lo mejor será descartarlo, puesto que será mejor no tener datos de ese día, que tenerlos de manera incorrecta.

Capítulo 6

Ejemplo de aplicación

En este capítulo se pretende desarrollar un ejemplo del posible estudio de una zona. Se ha emplazado el sistema hardware en una zona a estudiar, la cual es una parcela sin edificar en una urbanización en la que existen edificaciones. Se ha colocado 3 días distintos, las 24 horas del día, que corresponden a diferentes estados climáticos. Un día de agosto, otro de octubre y uno de diciembre.

Se analizan los resultados obtenidos de la herramienta hardware e interpretados gracias a la aplicación software, con ello se pueden extraer los datos y sacar conclusiones sobre el microclima y las diferentes variables medidas de la zona, con el fin de poder utilizarlos en la edificación de una arquitectura bioclimática, una casa en este ejemplo.

6.1 Datos de la zona

Hay que determinar algunos aspectos del emplazamiento de la estructura bioclimática. Primero habrá que determinar las coordenadas GPS de la zona, para ello se ha empleado una herramienta de [google.maps](http://www.google.es/maps) que nos permite conocer las coordenadas GPS. Como se puede observar en la figura 6.1.1.



Figura 6.1.1 Coordenadas GPS de la zona a estudiar.

Con las coordenadas GPS se conoce el emplazamiento de la vivienda y su entorno. El emplazamiento será importante porque habrá que determinar cual es microclima determinante de la zona. Según [15] se puede afirmar que el clima predominante en Málaga es el mediterráneo subtropical, característico de la mayor parte de la costa mediterránea andaluza. El hecho más característico de este ámbito es la gran suavidad térmica invernal, en virtud de la intervención de tres mecanismos básicos: la influencia suavizadora del mar, la orientación sur de la costa, que la convierte en una solana privilegiadísima para la recepción de los rayos solares, y la protección frente a las coladas septentrionales que le otorgan las cadenas Béticas. Esta definición debería ser válida para la zona a tratar puesto que visto el emplazamiento es lo que le correspondería. El estudio debe determinar si esta consideración previa es cierta o no.

Además nos sirve para localizar nuestro emplazamiento en un mapa topográfico que será necesario a la hora de realizar el estudio y el análisis posterior de la zona, puesto que es necesario conocer el nivel de la zona y su orografía.

En la figura 6.1.2 se puede ver el mapa topográfico de la zona en cuestión.



Figura 6.1.2 Mapa topográfico de la zona a estudiar.

La zona en cuestión está emplazada en una depresión que cruza de sureste a noroeste, además de pasar una autovía por esa depresión, además se puede observar las cotas cercanas a la zona de estudio.

6.2 Días medidos

6.2.1 Día de agosto (6-8-10)

A continuación se muestra la representación gráfica del día 6 de agosto del 2010:

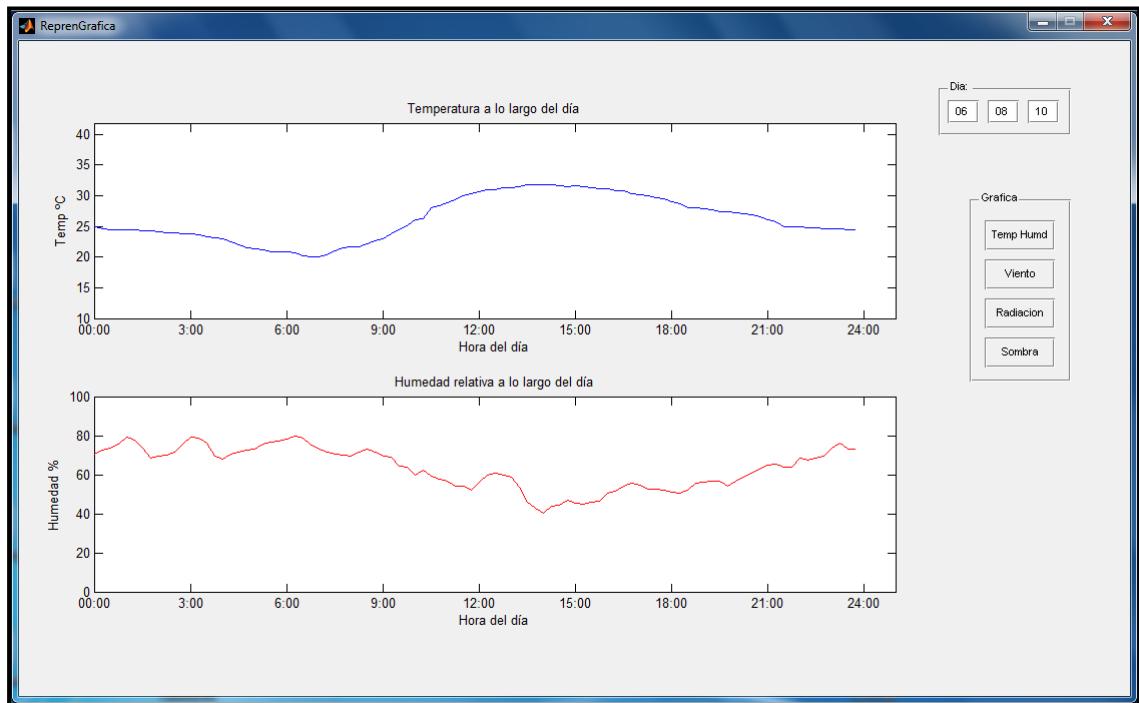


Figura 6.2.1 Representación de la temperatura y humedad del 6-8-10.

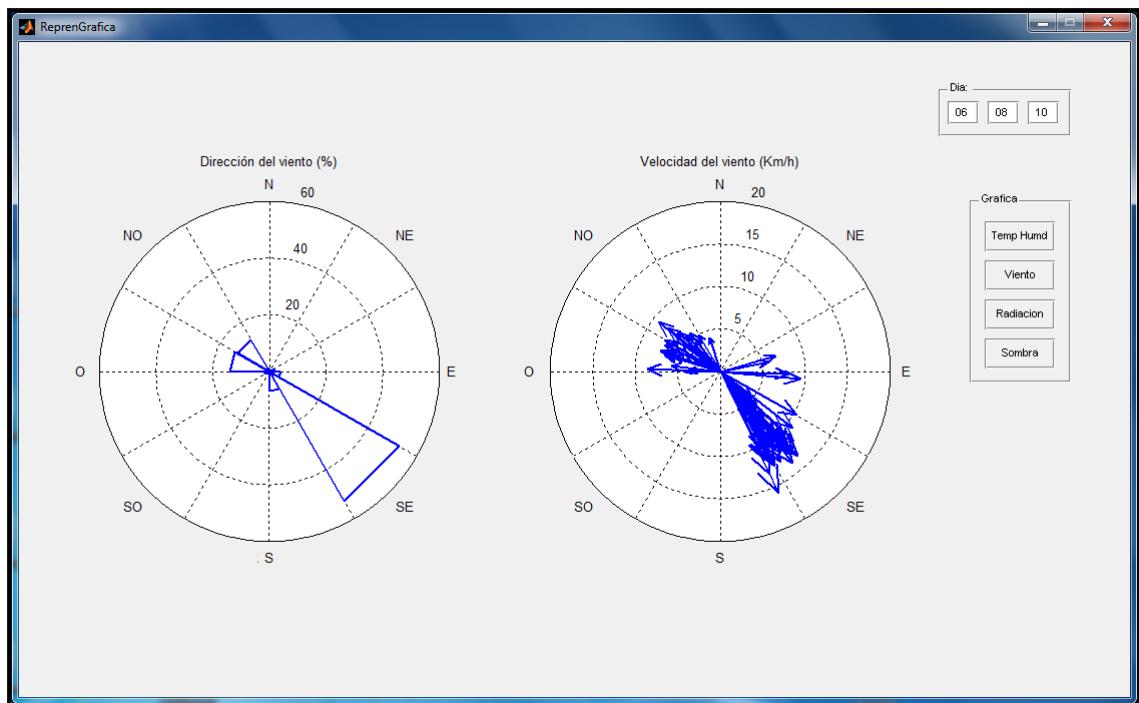


Figura 6.2.2 Representación de la velocidad y dirección del viento del 6-8-10.

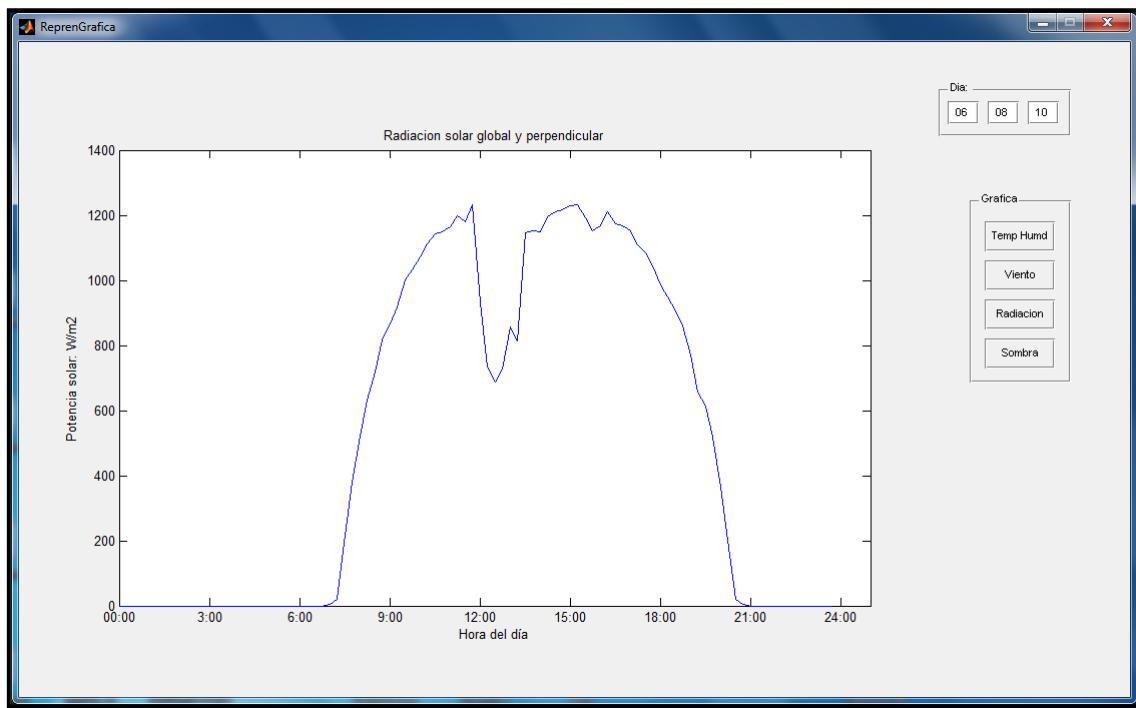


Figura 6.2.3 Representación de la radiación solar del 6-8-10.

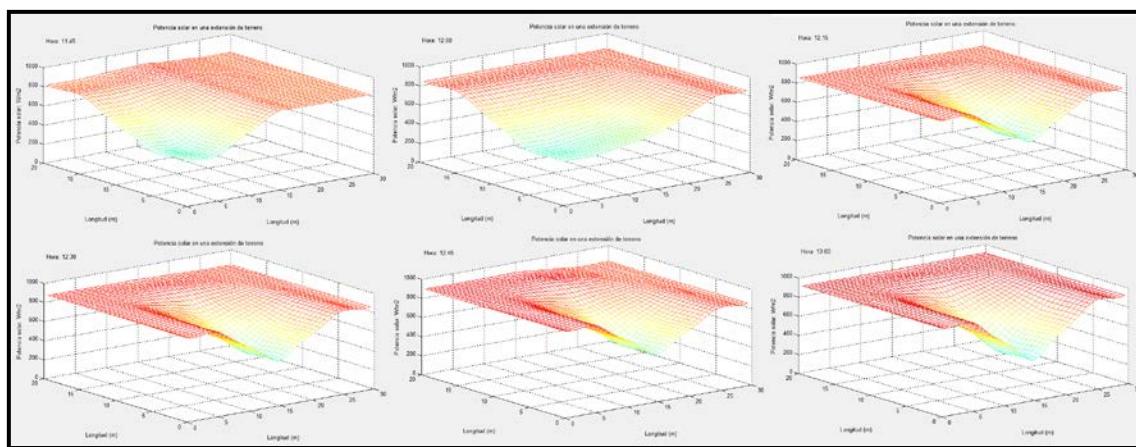


Figura 6.2.4 Parte de la evolución del mapa de sombras del 6-8-10.

La temperatura tiene un mínimo de poco más de 20°C y un máximo de menos de 32°C . La humedad oscila alrededor de los 70°C , teniendo un mínimo del 40% alrededor de las 14:00. El viento “sopla” en dirección sureste en la mayoría de las horas del día, siendo esta dirección la de máxima velocidad, con una velocidad de 12km/h de moda, otra dirección es de oeste-noroeste, pero el % es muy inferior. La radiación solar debería tener el máximo alrededor de las 14:00, pero no se puede observar porque se produce una sombra a esa hora, por el mapa de sombras se observa que se trata de obstáculo y no por nubes. Además amanece alrededor de las 7:30 y anochece alrededor de las 20:30.

6.2.2 Día de octubre (4-10-10)

A continuación se muestra la representación gráfica del día 4 de octubre del 2010:

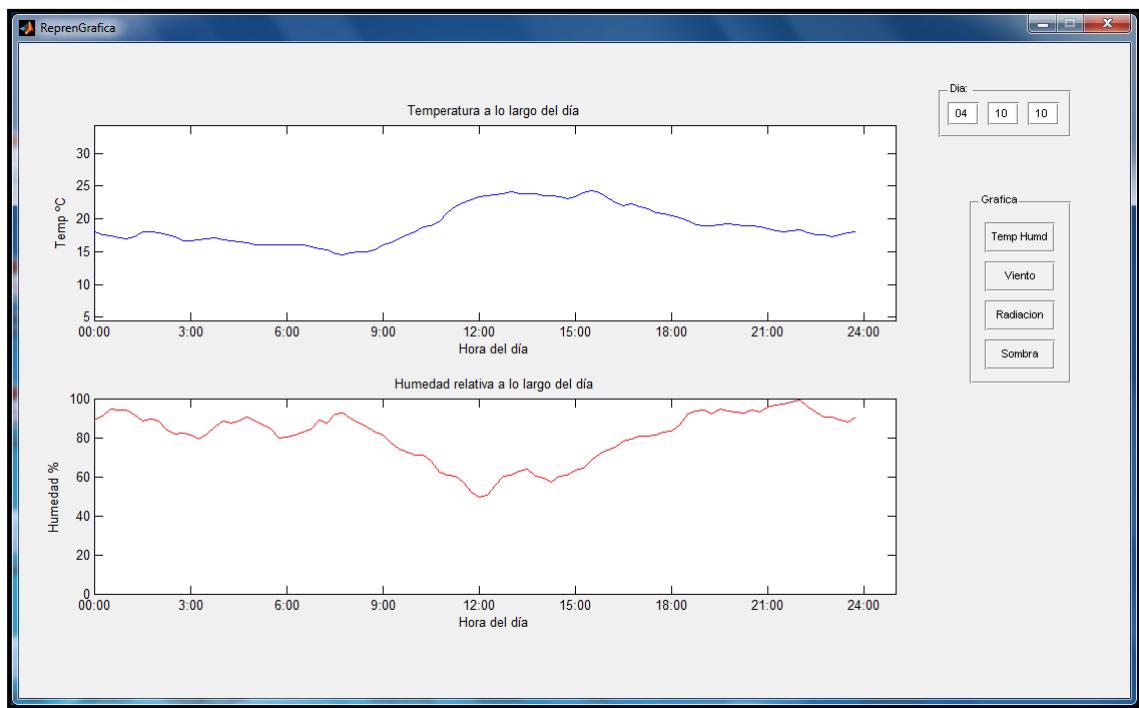


Figura 6.2.5 Representación de la temperatura y humedad del 4-10-10.

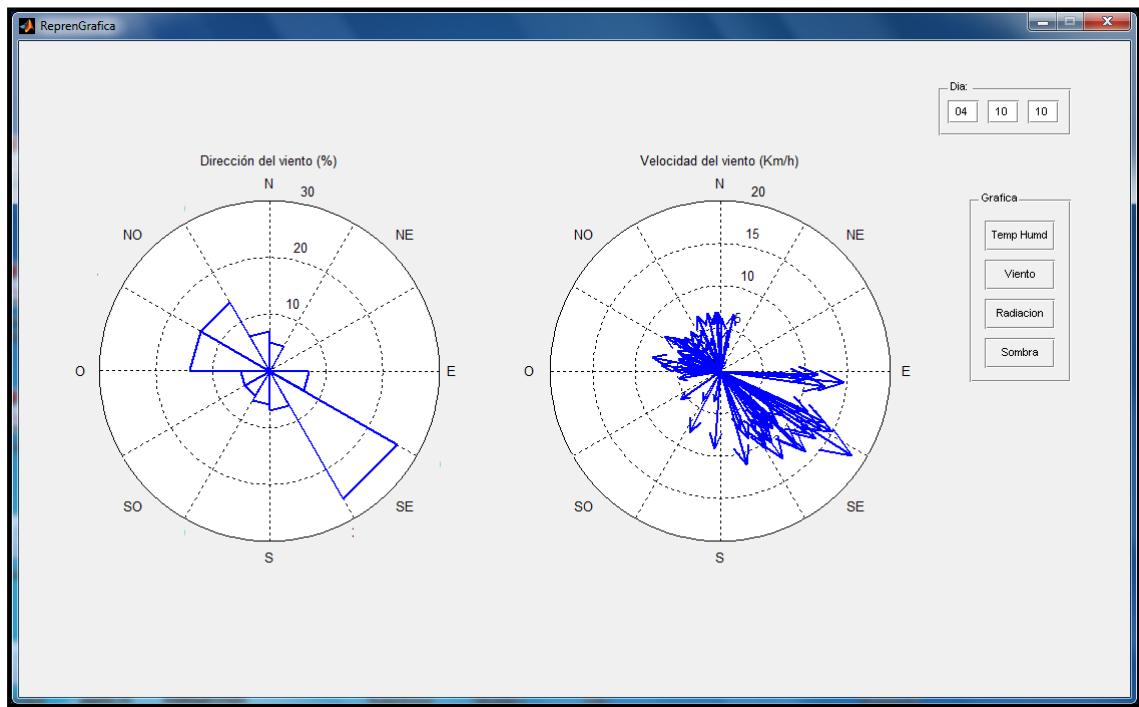


Figura 6.2.6 Representación de la velocidad y dirección del viento del 4-10-10.

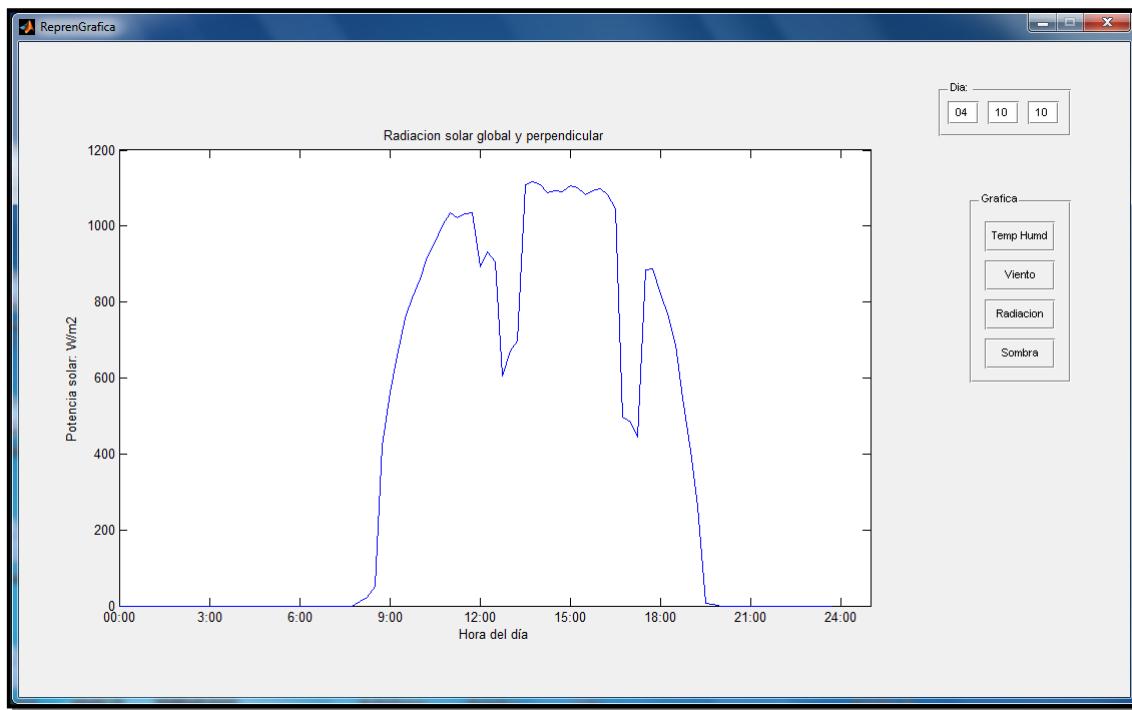


Figura 6.2.7 Representación de la radiación solar del 4-10-10.

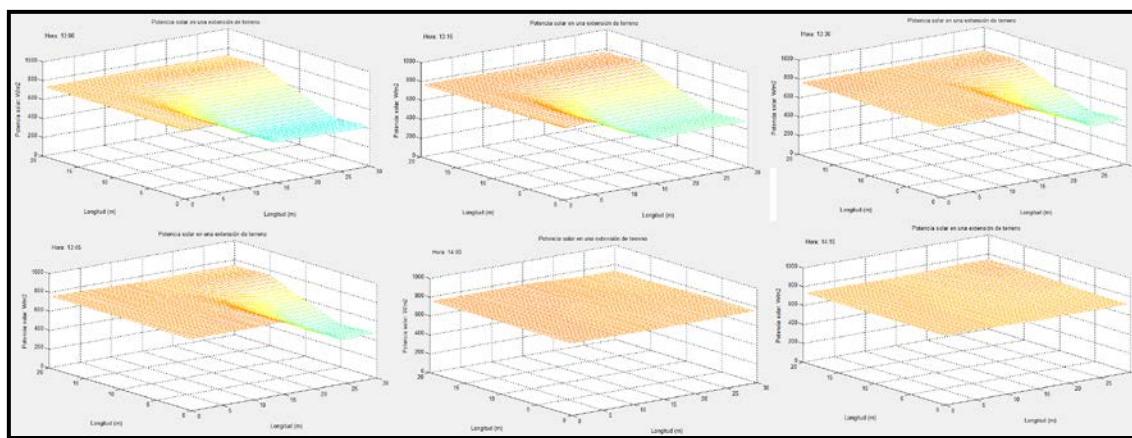


Figura 6.2.8 Parte de la evolución del mapa de sombras del 6-8-10.

La temperatura tiene un mínimo de poco más de $14,5^{\circ}\text{C}$ y un máximo de menos de 24°C , es bastante estable a lo largo del día. La humedad es muy elevada, con máximos superiores al 90%, en los momentos que no hay sol, eso significa que hay niebla espesa a esas horas, a otras, se reduce hasta el 60%, muy elevada para tener la presencia del sol. Hay viento variable, en casi todas direcciones, pero la dirección sureste es la predominante en la mayoría del tiempo, con velocidades superiores a 15km/h. Con la radiación solar observamos que tenemos el mismo obstáculo que el observado en el apartado 5.3.1, pero además se observa nubosidad alrededor de las 18:00.

6.2.3 Día de diciembre (3-12-10)

A continuación se muestra la representación gráfica del día 3 de diciembre del 2010:

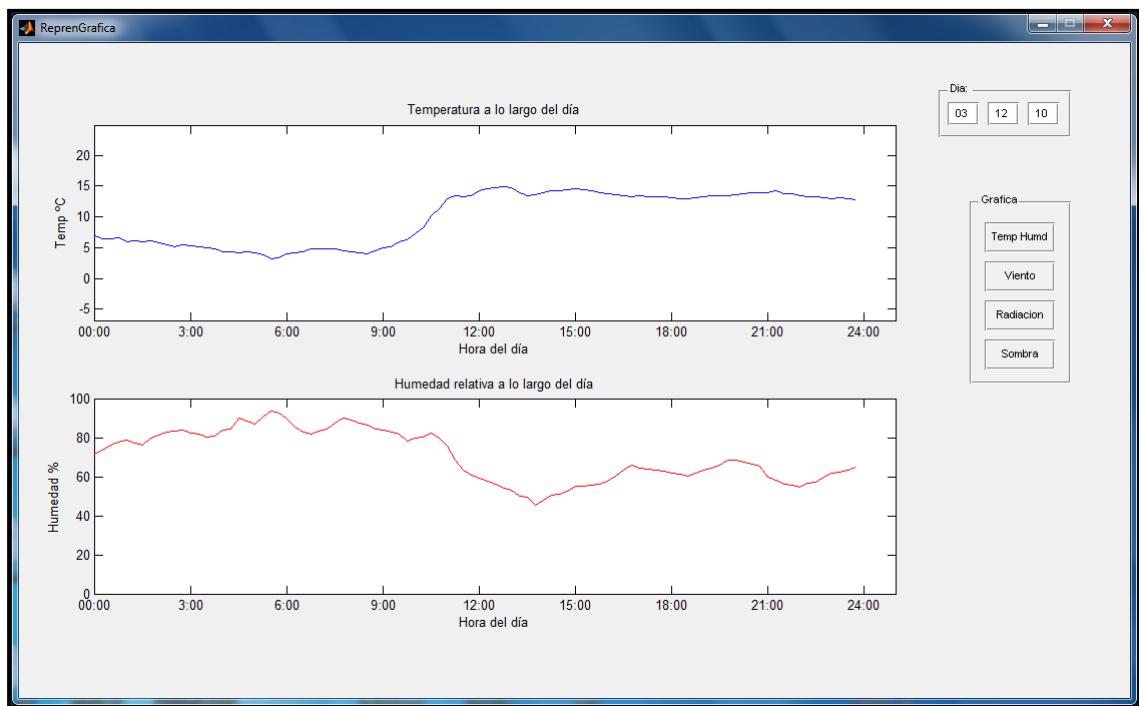


Figura 6.2.9 Representación de la temperatura y humedad del 3-12-10.

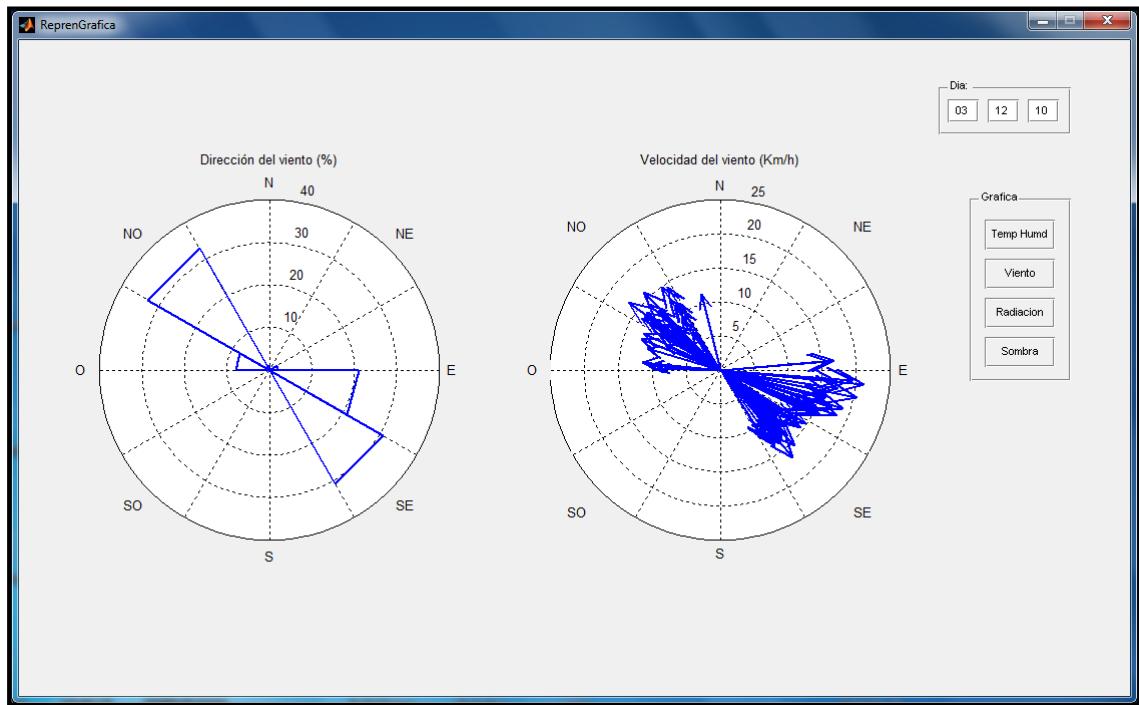


Figura 6.2.10 Representación de la velocidad y dirección del viento del 3-12-10.

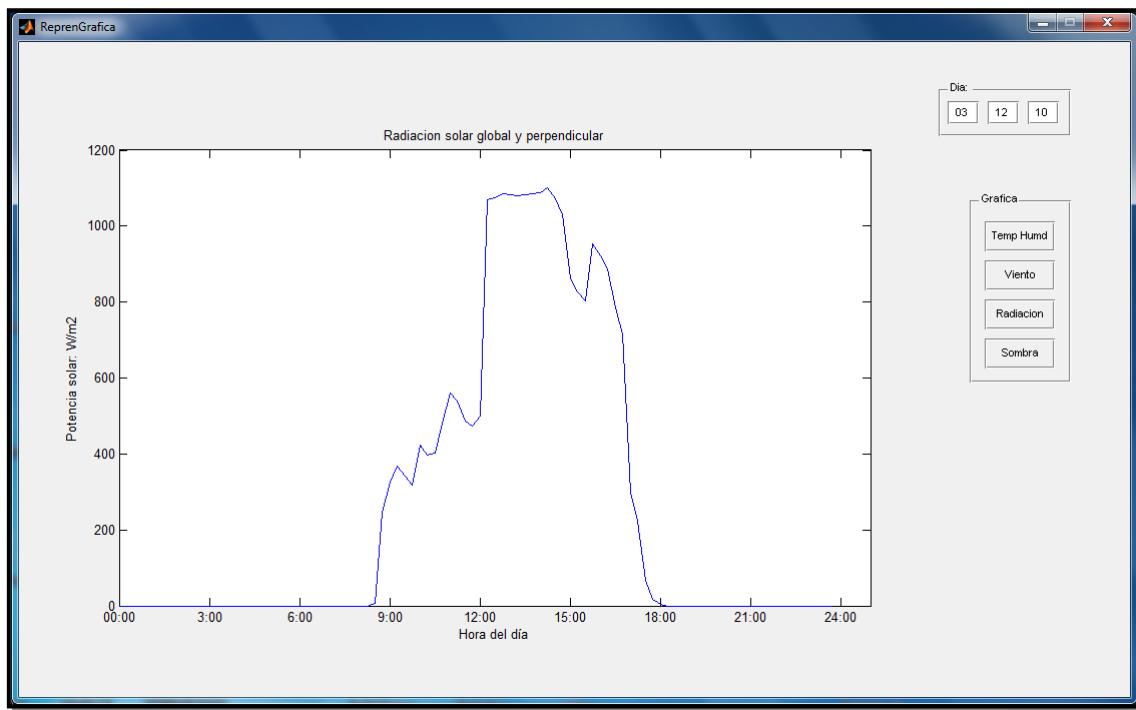


Figura 6.2.11 Representación de la radiación solar del 3-12-10.

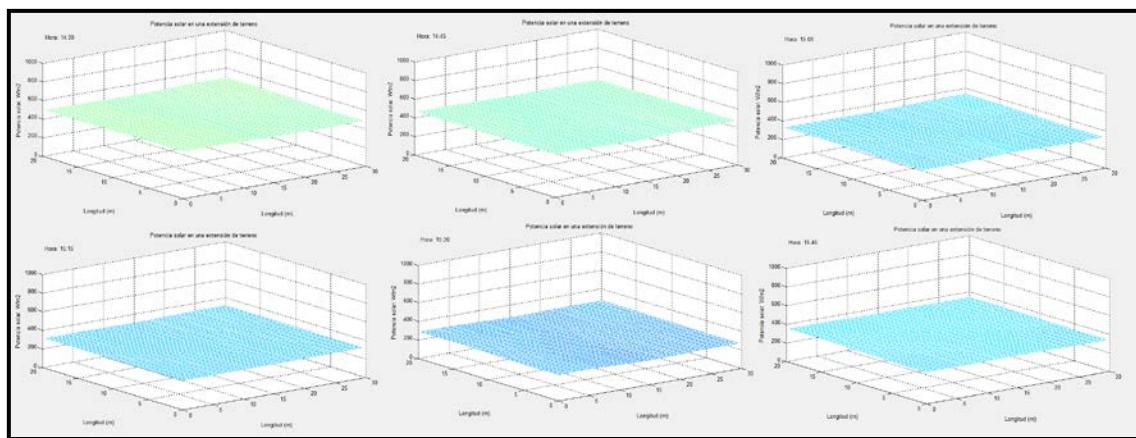


Figura 6.2.12 Parte de la evolución del mapa de sombras del 3-12-10.

La temperatura roza mínimos de poco más de 3º C y un máximo de menos de 13º C, con un salto térmico cuando amanece, temperaturas bajas para la zona climática que le correspondería por su ubicación. La humedad es elevada, alrededor del 80%, mínimos superiores al 50%. El viento tiene direcciones muy marcadas, sureste o noroeste, con velocidades superiores al 15% con máximas de 20km/h en dirección sureste, algo menos para la noroeste. En este caso no se puede contemplar el obstáculo observado en los anteriores apartados puesto que hubo nubosidad en la mayor parte del día, como se ve en el mapa de sombras. Si se puede observar que amanece a la misma hora que el día de octubre, y anocchece mucho antes, esto es dividido al cambio horario, que da aspecto de desplazamiento de la gráfica.

6.3 Conclusiones

Para sacar las conclusiones hay que tener en cuenta dos premisas:

- Los resultados obtenidos son de sólo tres días; por motivos de tiempo, no se puede dejar el sistema durante un año, que sería lo adecuado, y motivos de seguridad, como la pérdida, rotura o degradación de los componentes.
- Las conclusiones derivadas de los resultados, no son realizadas por expertos en la materia. Los datos deben analizarlos un arquitecto bioclimático, experto en condiciones microclimáticas que no es el perfil de los creadores de este proyecto. Se analizan de manera aproximada para poner en contexto al mismo.

Las variables obtenidas de la extrapolación de los datos son:

- Temperatura: Son muy estables a lo largo del día, con sólo un pequeño aumento cuando amanece; sin embargo no sucede lo mismo cuando anochece ya que el descenso es gradual. Las temperaturas máximas son las esperadas para un clima mediterráneo, pero las mínimas son inferiores a las que se suponían.
- Humedad: Elevada, previsibles para la climatología, con máximos muy cercanos al 100% que corresponde a una zona con gran concentración de niebla espesa.
- Viento: Las direcciones predominantes son sureste y noroeste, esta dirección se debe sobre todo a la topografía de la zona; en la figura 5.2.2 se comprueba que la zona se encuentra en una pequeña depresión que hace que el viento se encauce en estas direcciones. La velocidad del viento es muy estable, sin máximas demasiado elevadas, pero pocas veces con ausencia de viento.
- Radiación: Es bastante similar a la esperada, comprobando que se trata de una zona con muchos días de sol.
- Mapa de sombras: Existe un obstáculo en dirección sur. Cuando el sol se encuentra en su posición más alta genera una sombra en la parte sur de la zona.

Las actuaciones bioclimáticas a realizar en una vivienda a construir son para conseguir confort térmico y visual, no se ha tenido en cuenta el acústico porque no existen fuentes de ruido significativas.

Se empieza evaluando el clima de la región y el microclima de la zona. La región es de clima mediterráneo, pero el microclima de la zona, no lo es exactamente ya que tiene temperaturas más extremas que los mediterráneos y se generan nieblas densas. El viento es característico por la orografía y su ubicación.

Las principales actuaciones que se pueden realizar son:

- Orientar las habitaciones principales al sur, por el confort visual, ya que hay que tener en cuenta el nivel visual de la estructura resultante, como se indica en el apartado 2.1.2.
- La adaptación de la temperatura es, como ya se ha dicho, de los factores más importantes. Las temperaturas mínimas son más bajas de lo normal para la región, podría ser efectivo un invernadero o un muro trombe para captar el calor cuando se precise, ambas construcciones se deben localizar al este. Si se orientase al sur perderían eficiencia por las sombras generadas, igual que si se utilizasen sistemas de protección solar.
- Al precisarse calentamientos más efectivos, se puede poner una serie de paneles solares térmicos, para poder tener elementos radiantes, y así ahorrar en calefacción.
- Poner paneles solares fotovoltaicos para compensar el consumo eléctrico de la vivienda. Se situarán al norte de la zona orientados al sur ya que es donde no se producen sombras.
- Prescindir de dispositivos mecánicos de enfriamiento que consumen electricidad, ya que si se adecua una chimenea solar orientada en situación noroeste o sureste, direcciones principales del viento en esta zona, se puede tener una temperatura adecuada sin consumo eléctrico.
- Colocar aerogeneradores en los conductos de la chimenea solar y la zona exterior orientados en dirección del viento para terminar de suplir el consumo eléctrico de la vivienda.

Además de estas actuaciones, también hay que tener en cuenta otros factores de una estructura bioclimática, como son la utilización de materiales ecológicos y/o reciclados para la construcción de la vivienda, que ayuden a la sostenibilidad. Y la integración de la estructura en el entorno, sin que haya un impacto ecológico y visual son otras actuaciones.

Capítulo 7

Conclusiones y líneas futuras

En este capítulo se va a estudiar las conclusiones más destacadas, así como las posibles líneas de actuación que podría tener el presente proyecto para su mejora. Habrá que tener en cuenta que tanto las conclusiones como las líneas futuras son percepciones personales del autor, pudiéndose sacar otras conclusiones u otras líneas futuras según los diferentes lectores de este proyecto.

7.1 Conclusiones

Las conclusiones que se pueden sacar del proyecto son:

- Las herramientas implementadas, tanto hardware como software, cumplen adecuadamente los objetivos marcados previamente a la realización del proyecto, puesto que como se ha observado se puede realizar adecuadamente el análisis de una zona determinada.
- Las arquitecturas bioclimáticas son los sistemas de edificación del futuro, puesto que no sólo incrementa el confort de las personas que se hallan en el interior, además el consumo eléctrico de este tipo de arquitecturas se puede reducir fácilmente a la mitad.
- El dispositivo de hardware libre Arduino es un sistema que siendo limitado en cuanto a la potencia del microcontrolador dispone de una versatilidad y funcionalidad que no disponen otro tipo de sistemas basados en otros microcontroladores.
- Las comunicaciones GPRS son un tipo de comunicaciones difícilmente utilizables para la comunicación de un microcontrolador como el ATMega 328, debido a su escasa potencia y al tipo de programación que precisa un módulo GPRS con comandos AT.
- Las variables obtenidas con el sistema implementado son bastante próximas a la que obtendríamos con otros sistemas de medidas más caros y sin posibilidad de integración entre sí. Por lo que este sistema se podría utilizar como un sistema prototípico de bajo coste.
- La captación de la radiación solar es la variable climática que más cuesta medir puesto que el sistema consta de una serie de fotodiodos que toman un valor diferente dependiendo del espectro de la luz captada y de la temperatura que puede llegar a alcanzar.

7.2 Líneas futuras

Las posibles líneas futuras de actuación para la mejora del sistema pueden ser:

- Posible realización con otro sistema que disponga de un microcontrolador más potente, tanto en el número de entradas como en potencia de proceso. Ya que el sistema Arduino, aún siendo suficiente para la realización de nuestro sistema, puede ser insuficiente si queremos ampliar las funciones de nuestro sistema.
- En vez de utilizar comunicaciones GPRS o además de utilizar este tipo de comunicaciones, utilizar otro tipo que sean más económicas y más fácilmente implementables y configurables, como pueden ser comunicaciones wifi o bluetooth. Este tipo de comunicaciones tienen una serie de restricciones, pero como complemento o para sistema en zonas muy determinadas pueden ser una buena solución.
- Utilizar sensores inalámbricos para no necesitar cables para utilizar los sensores distribuidos, ni tener las restricciones en cuanto a distancia y coste del cable.
- Se puede hacer otro tipo de captadores solares, se podrían poner más de un fotodiodo por cada lado del sistema de captación solar y coger la media de dichos fotodiodos o utilizar piranómetros profesionales en vez de fotodiodos para captar la radiación. Este tipo de implementaciones el problema que tienen es el coste, además de necesitar otro microcontrolador. El precio puede no hacer rentable el grado de mejora en la captación de dicha medida.
- Crear una aplicación para poder visualizar las diferentes variables de un determinado sistema, vía Web sin necesidad de disponer la aplicación creada en Matlab. Así podríamos acceder de manera remota a todos los datos desde cualquier ordenador.

Referencias

- [1] John Theodore Houghton, fundador del Centro Hadley y copresidente del grupo de evaluación científica del IPCC.
- [2] Ministerio de medio ambiente, y medio rural y marino.
- [3] Ignacio Molina Conde, “Metodología de análisis del espacio público”.
- [4] http://es.wikipedia.org/wiki/Confort_higrotermico
- [5] Rafael Agüero León. Máster Energía y medio ambiente 08/09. Universidad politécnica de catalunya.
- [6] Katia Simancas Yovane, “Reacondicionamiento bioclimático de viviendas de segunda residencia en clima mediterráneo”.
- [7] Manual del Arduino.
- [8] Datasheet ATMega 328.
- [9] Datasheet DS1307.
- [10] Datasheet TAOS TSL230R.
- [11] Datasheet MAXIM DS18B20.
- [12] <http://re.jrc.ec.europa.eu/pvgis/apps/radmonth.php?lang=es&map=europe>
- [13] <http://maps.google.com/>
- [14] <http://es.wikipedia.org/wiki/MATLAB>
- [15] Ayuntamiento de Málaga. Apuntes medioambientales: climatología de Málaga.
http://www.ayto-malaga.es/apuntes_medioambientales/cap_1.html.
- [16] Real Academia Española (RAE)

Bibliografía

- Ignacio Molina Conde, “Metodología de análisis del espacio público”.
- Katia Simancas Yovane, “Reacondicionamiento bioclimático de viviendas de segunda residencia en clima mediterráneo”.
- <http://es.wikipedia.org/>
- Arquitectura Bioclimática y construcción sostenible: Diseño y cálculo.
- Manual del Arduino.
- Ayuda del Matlab.