

Arduino + módulo GSM/GPRS: monitorización, automatización y gestión remota en un viñedo.

MEMORIA TFG.

Grado de Tecnologías de Telecomunicación.

Alumno: Rubén Adrián de la Cámara.

Consultor: Carlos Gonzalo Moreno Soriano.

15/06/2017

© 2017. Rubén Adrián de la Cámara.

Reservados todos los derechos. Está prohibido la reproducción total o parcial de esta obra por cualquier medio o procedimiento, comprendidos la impresión, la reprografía, el microfilme, el tratamiento informático o cualquier otro sistema, así como la distribución de ejemplares mediante alquiler y préstamo, sin la autorización escrita del autor o de los límites que autorice la Ley de Propiedad Intelectual

FICHA DEL TRABAJO FINAL

| | |
|---|--|
| Título del trabajo: | Arduino + módulo GSM/GPRS: monitorización de parámetros ambientales y automatización en un viñedo. |
| Nombre del autor: | Rubén Adrián de la Cámara. |
| Nombre del consultor/es: | Carlos Gonzalo Moreno Soriano. |
| Fecha de entrega (mm/aaaa): | 06/2017 |
| Área del Trabajo Final: | Diseño de sistemas electrónicos |
| Titulación: | <i>Grado en Tecnologías de Telecomunicación.</i> |
| Resumen del Trabajo (máximo 250 palabras): | |
| <p>El presente proyecto consiste en el diseño de un sistema de monitorización de parámetros ambientales, automatización de procesos y gestión remota mediante la red GSM/GPRS. Se ha contextualizado para ser empleado en un viñedo por lo que se ha presentado la solución al problema con los argumentos pertinentes. Aparte, se ha recurrido al universo Arduino implementar la solución, debido a su potencia y escalabilidad.</p> <p>Por un lado, se ha diseñado una serie de bloques funcionales que al ser integrados conformarán un sistema totalmente operativo. Por ello, a lo largo de los capítulos de la memoria se pueden ver los diseños y demostraciones necesarias para comprobar la funcionalidad de estos bloques. Se parte de la descripción del universo Arduino y de la placa que se empleará como prototipo.</p> <p>Por otro lado, se comienza con la monitorización y automatización mediante sensores, actuadores y una placa Arduino, que con el código adecuado permite automatizar procesos como el riego automático o un sistema antiheladas. Se propone una fuente de alimentación, teniendo en cuenta los requisitos energéticos del sistema y se implementa un sistema de comunicaciones que permita la gestión remota mediante SMS.</p> <p>Finalmente, se propone la optimización del sistema con la finalidad de abaratar el circuito y poder mostrar información mediante un display LCD. Por último, se proponen una serie de mejoras y líneas de trabajo futuras que aportarían una serie de ventajas interesantes, entre ellas, destacar las relacionadas con el Internet de las cosas.</p> | |

Abstract (in English, 250 words or less):

The present project consists on the design of a system for monitoring environmental parameters, process automation and remote management through the GSM / GPRS network. It has been contextualized to be employed in a vineyard, so the problem has been contextualized with the necessary arguments. Arduino universe has been used to achieve its implementation, due to its power and scalability.

On the one hand, a series of functional blocks have been designed, that when integrated, will form a fully operational system. For this reason, throughout the chapters of the memory you can see the designs and demonstrations to check the functionality of these blocks. It starts from the description of the Arduino universe and the plate that will be used for doing simulations

On the other hand, it starts with the monitoring and automation by means of sensors, actuators and Arduino plate that with suitable code allows to automate process as the automatic irrigation or an antifrost system. It proposes a power supply taking into account the energy requirements of the system and a communications system is implemented that allows remote management by SMS.

Finally, it proposes the optimization of the system in order to reduce the circuit and to be able to display information through an LCD display. Last, it proposes a series of improvements and future lines of work that would provide a series of interesting advantages including those related to the Internet of Things. All this, based on the IP protocol, which would imply another system of communications like 3G.

Palabras clave (entre 4 y 8):

Viñedo, monitorización, automatización, comunicaciones, solución, Arduino.

TABLA DE CONTENIDO

| | |
|---|-----|
| Índice de figuras..... | vi |
| índice de tablas..... | vii |
| 1. Introducción..... | 1 |
| 1.1 Contexto y justificación del Trabajo..... | 1 |
| 1.2 Objetivos del Trabajo | 2 |
| 1.3 Enfoque y método seguido | 3 |
| 1.4 Planificación del Trabajo | 4 |
| 1.4.1 Alcance. | 4 |
| 1.4.2 Hitos..... | 4 |
| 1.4.3 Calendario de trabajo. | 5 |
| 1.4.4 Tareas y Diagrama de Gantt. | 6 |
| 1.4.5 Riesgos e incidencias..... | 8 |
| 1.5 Breve resumen de productos obtenidos | 9 |
| 1.6 Breve descripción de los otros capítulos de la memoria..... | 9 |
| Capítulo 1. Entradas y salidas del sistema. | 11 |
| 1.1 Sensores. | 11 |
| 1.1.1 Sensor de temperatura y humedad ambiental DHT22. | 11 |
| 1.1.2 Sensor humedad del suelo YL-69..... | 13 |
| 1.2 Actuadores..... | 14 |
| 1.2.1 Sistema de riego automático. | 14 |
| 1.2.2 Sistema de generación de viento..... | 15 |
| 1.3 Resumen capítulo. | 16 |
| Capítulo 2. Arduino..... | 17 |
| 2.1 Entorno Arduino. | 17 |
| 2.1.1 ¿Qué es Arduino? | 17 |
| 2.1.2 Hardware Arduino..... | 18 |
| 2.1.3 IDE Arduino. | 18 |
| 2.1.4 Comunidad Arduino..... | 19 |
| 2.2 Arduino en el presente proyecto. | 20 |
| 2.2.1 Arduino UNO R3. | 20 |
| 2.3 Resumen del capítulo. | 22 |
| Capítulo 3. Sistema de monitoreo y automatización. | 23 |
| 3.1 Monitorización de parámetros y salidas del sistema. | 23 |
| 3.1.1 Entradas..... | 23 |

| | |
|---|----|
| 3.1.2 Salidas..... | 24 |
| 3.2 Diseño del circuito..... | 26 |
| 3.3 Software de monitorización y automatización. | 26 |
| 3.4 Prototipo y prueba real..... | 28 |
| 3.4.1. Prototipo del sistema de monitorización y automatización. | 28 |
| 3.4.2 Pruebas..... | 29 |
| 3.5 Resumen del capítulo. | 31 |
| Capítulo 4. Fuente de alimentación. | 32 |
| 4.1 Tensiones y corrientes en el sistema. | 32 |
| 4.2 Circuito de fuente de alimentación. | 33 |
| 4.3 Esquemático y simulación..... | 36 |
| 4.4. Layout y modelo 3D. | 37 |
| 4.5 Resumen del capítulo. | 38 |
| Capítulo 5. Comunicaciones en el sistema..... | 39 |
| 5.1 Introducción a la tecnología GSM/GPRS. | 39 |
| 5.1.1 GSM (Groupe Special Mobile)..... | 39 |
| 5.1.2 GPRS (Servicio general de paquetes de radio). | 42 |
| 5.1.3 Servicios sobre GSM/GPRS..... | 42 |
| 5.2 Tarjeta Sim900GSM/GPRS. | 43 |
| 5.2.1 Hardware..... | 44 |
| 5.2.2 Software. | 46 |
| 5.3 Prototipo y PRUEBAS..... | 48 |
| 5.3.1 Verificación de comunicaciones..... | 48 |
| 5.3.2 Software de comunicaciones. | 50 |
| 5.3.3 Prototipo y Sketch. | 52 |
| 5.4 Resumen del capítulo. | 54 |
| Capítulo 6. Optimización del circuito..... | 56 |
| 6.1 Microcontrolador ATmega 328P..... | 56 |
| 6.2 Display LCD..... | 57 |
| 6.3 Adaptaciones necesarias. | 58 |
| 6.4 Implementación del circuito..... | 59 |
| 6.5 Resumen del capítulo. | 62 |
| Conclusiones y líneas futuras de trabajo. | 63 |
| Bibliografía. | 65 |

ÍNDICE DE FIGURAS.

| | |
|---|----|
| FIGURA 1. DIAGRAMA DE BLOQUES DEL SISTEMA..... | 3 |
| FIGURA 2. DIAGRAMA DE GANTT. | 8 |
| FIGURA 3. DTH22. | 11 |
| FIGURA 4. CONEXIÓN DHT 22 Y ARDUINO. | 12 |
| FIGURA 5. SENSOR HUMEDAD DEL SUELO YL-69. | 13 |
| FIGURA 6. CONEXIÓN SENSOR YL-69 Y ARDUINO. | 14 |
| FIGURA 7. RIEGO POR GOTEO. | 14 |
| FIGURA 8. ESQUEMA SISTEMA DE RIEGO. | 15 |
| FIGURA 9. TORRES GENERADORAS DE VIENTO. | 16 |
| FIGURA 10. EL UNIVERSO ARDUINO. | 17 |
| FIGURA 11. PLACAS ARDUINO. | 18 |
| FIGURA 12. INTERFAZ DE PROGRAMACIÓN DE ARDUINO. | 19 |
| FIGURA 13. ARDUINO UNO R3. | 20 |
| FIGURA 14. PROTOTIPO CON MICROCONTROLADOR ATMEGA 328. | 21 |
| FIGURA 15. MÓDULO DE DOS RELÉS. | 24 |
| FIGURA 16. CONTACTOR DE 380 V. | 25 |
| FIGURA 17. CIRCUITO DE AUTOMATIZACIÓN: ENTRADAS Y SALIDAS. | 26 |
| FIGURA 18. CAPTURA DE SUBLIME TEXT. | 27 |
| FIGURA 19. PROTOTIPO MONITORIZACIÓN DE PARÁMETROS Y AUTOMATIZACIÓN. | 29 |
| FIGURA 20. SIMULACIONES DE PARÁMETROS AMBIENTALES. | 29 |
| FIGURA 21. MONITOR SERIE: PARÁMETROS RECIBIDOS. | 30 |
| FIGURA 22. BLOQUES FUENTE DE ALIMENTACIÓN. | 34 |
| FIGURA 23. EJEMPLOS DE COMPONENTES EMPLEADOS EN LA FUENTE DE ALIMENTACIÓN. | 35 |
| FIGURA 24. SIMULACIÓN CIRCUITO FUENTE DE ALIMENTACIÓN. | 36 |
| FIGURA 25. ESQUEMA FUENTE DE ALIMENTACIÓN. | 37 |
| FIGURA 26. CIRCUITO BLOQUE FUENTE DE ALIMENTACIÓN. | 37 |
| FIGURA 27. TARJETA SIM900 GSM/GRPS. | 43 |
| FIGURA 28. DESCRIPCIÓN FÍSICA DE LA TARJETA SIM900 GSM/GPRS. | 44 |
| FIGURA 29. PIN OUT DEL CHIP SIM900 Y VISTA INFERIOR DE LA TARJETA. | 45 |
| FIGURA 30. CONEXIÓN MEDIANTE CONVERTOR FTDI E HYPERTERMINAL. | 47 |
| FIGURA 31. MONTAJE PARA PRUEBAS COMANDOS AT. | 49 |
| FIGURA 32. HYPERTERMINAL Y COMANDOS AT. | 49 |
| FIGURA 33. SIMULACIONES SIM Y SMS. | 50 |
| FIGURA 34. SMS ENVIADO A UN MÓVIL. | 51 |
| FIGURA 35. RECEPCIÓN Y ENVÍO DE SMS (MONITOR SERIE). | 51 |
| FIGURA 36. PROTOTIPO PARA SIMULACIÓN SISTEMA Y COMUNICACIONES. | 52 |
| FIGURA 37. PINES ATMEGA 328 P Y EQUIVALENCIA EN ARDUINO UNOR3. | 56 |
| FIGURA 38. DISPLAY LCD 16X2 CARACTERES. | 57 |
| FIGURA 39. ESQUEMA LED INTERMITENTE CON ATMEGA 328. | 59 |
| FIGURA 40. LAYOUT DEL CIRCUITO CON ATMEGA 328 Y LCD. | 61 |
| FIGURA 41. VERIFICACIÓN DEL PROTOTIPO DEFINITIVO. | 61 |

ÍNDICE DE TABLAS.

| | |
|--|----|
| TABLA 1. HITOS. | 5 |
| TABLA 2. DISTRIBUCIÓN DE HORAS MENSUAL. | 6 |
| TABLA 3. DTH11 VS DTH22. | 12 |
| TABLA 4. CARACTERÍSTICAS DE LOS RELÉS SDR. | 25 |
| TABLA 5. TENSIONES Y CORRIENTES DE LOS DISPOSITIVOS. | 32 |
| TABLA 6. DESCRIPCIÓN DE LOS LEDS DE LA TARJETA SIM900 GSM/GPRS. | 45 |
| TABLA 7. PARÁMETROS ATMEGA 328. | 57 |

1. INTRODUCCIÓN

Una vez finalizadas las asignaturas correspondientes al Grado en Tecnologías de Telecomunicación, el último paso para obtener la titulación consiste en desarrollar un proyecto que ponga de manifiesto las competencias adquiridas durante el Grado. De este modo, se persigue que el alumno demuestre sus capacidades de investigación y análisis de información, así como los conocimientos adquiridos durante los distintos semestres. Es por tanto, el momento de demostrar la capacidad de resolver un problema dentro de un ámbito de conocimiento o aportar alguna mejora, análisis o innovación, sobre algún objeto de dicho ámbito.

En este caso, el presente documento detalla el proceso de creación de un sistema de automatización y monitorización remota de una serie de parámetros que influyen en la producción de un viñedo. Así, se pretende exponer el conjunto de elementos hardware y software que serán necesarios, tanto para automatizar determinados procesos en el viñedo, como acceder a ciertos datos de forma remota empleando la red GSM/GPRS. Incluso, el sistema debe permitir la puesta en marcha y parada de determinados procesos de forma remota. Para realizar el proyecto se empleará Arduino, dado que es un sistema de desarrollo muy interesante y que aporta muchas facilidades técnicas.

Concluyendo, se verá que, empleando toda la potencia del sistema Arduino junto con los conocimientos adquiridos en las distintas asignaturas cursadas durante el Grado, será posible completar el desarrollo del proyecto. De este modo, quedará demostrado que, la versatilidad de Arduino junto con las competencias adquiridas durante el Grado, pueden aportar infinidad de soluciones técnicas a problemas de complejidad moderada. Con todo, este no es un proyecto cerrado, ya que con las modificaciones pertinentes podrá ser adaptado a cualquier otro entorno donde se necesite automatizar y monitorizar a distancia.

1.1 CONTEXTO Y JUSTIFICACIÓN DEL TRABAJO

Desde la antigüedad, uno de los mayores inconvenientes de la agricultura está en la meteorología, ya que los cultivos son muy sensibles a los parámetros ambientales. Ciertos parámetros como: temperatura, humedad, viento... afectan positiva o negativamente al cultivo, por lo que poder conocerlos de forma inmediata y sin tener que desplazarse al lugar de cultivo permite tomar decisiones sobre los tratamientos a realizar, en caso necesario.

Por un lado, una aproximación a los parámetros que más influyen en el viñedo consiste en monitorizar temperatura y humedad, ya que en las distintas etapas del ciclo productivo (floración, envero, cosecha) pueden afectar muy seriamente a la producción en cantidad y calidad. Como ejemplo se puede exponer el caso de la floración, siendo muy perjudicial el hecho de que se produzcan heladas durante esta etapa. Por ello, es muy importante tener un conocimiento actualizado de la situación ambiental del viñedo, incluso de la humedad del terreno, para poder realizar un riego en caso necesario.

Por otro lado, en el presente documento se pretende poder aportar una solución tecnológica para evitar esta incertidumbre y poder actuar en consecuencia. Así, el sistema permitirá conocer tanto la temperatura ambiental como la humedad de ambiente y del suelo, simplemente empleando un SMS. Esto servirá para poder realizar las labores de cuidado del viñedo de forma óptima. Además, el sistema permitirá activar el riego automático en función de los anteriores parámetros y activar unos generadores de viento para evitar el riesgo de heladas (temperaturas inferiores a 0 °C) en determinados momentos como pueden ser los meses de Abril y Mayo.

En resumen, el viñedo es un entorno muy sensible a los factores ambientales en ciertas épocas, de modo que si al menos se pueden conocer de forma inmediata dichos factores se podrá actuar más ágilmente. También, es muy interesante el hecho de poder independizar ciertas actuaciones de la presencia humana, de tal modo que el mismo sistema realice las operaciones para las que se programe, en función de los datos recibidos por los sensores instalados.

1.2 OBJETIVOS DEL TRABAJO

El objetivo fundamental del presente trabajo es el diseño de un sistema que permita monitorizar parámetros ambientales, automatizar determinados procesos y poder interactuar de forma remota con el sistema mediante la transmisión de SMS empleando la red GSM/GPRS. Para ello, será necesario alcanzar una serie de objetivos más concretos:

- Familiarizarse con el entorno Arduino, que será la tecnología a emplear.
- Determinar los sensores necesarios.
- Comprender y poder diseñar una fuente de alimentación.
- Realizar una aproximación a la tecnología GSM/GPRS.
- Entender el funcionamiento de un módulo de comunicaciones GSM/GPRS
- Introducirse en el software de diseño y simulación electrónico.
- Practicar el montaje de prototipos y su simulación.

Con todo, a la conclusión del trabajo se habrán alcanzado una serie de objetivos de forma indirecta:

- Identificar objetivos generales y concretos para la resolución de un problema.
- Demostrar los conocimientos competencias adquiridas a lo largo del Grado.
- Aplicar la capacidad de redacción de textos científico-técnicos.
- Aprender a realizar presentaciones formales de trabajos o proyectos técnicos.
- Demostrar proactividad y capacidad de análisis y síntesis.

1.3 ENFOQUE Y MÉTODO SEGUIDO

En primer lugar, es necesario aclarar que se trata de un trabajo académico, de modo que el proceso consistirá en el aprendizaje y la aplicación al proyecto de lo aprendido. Es decir, partiendo de que el objetivo fundamental del proyecto está definido, se requiere una investigación continua y su aplicación al sistema de todas las partes que conformarán el todo. Así, dado que se ha propuesto como tecnología a emplear el entorno Arduino, será necesario un esfuerzo continuo en conocer su hardware y software, con la finalidad de ser capaz de aplicarlo al proyecto. Esto es aplicable a todos los bloques que conformarán el sistema (control, monitorización, comunicaciones, alimentación) lo que puede verse en el diagrama mostrado en la figura 1.

Por otra parte, una vez planteado el diagrama de bloques, se procederá con la resolución de cada bloque siguiendo lo explicado anteriormente. De este modo, en cada bloque se buscará la información necesaria, se realizará el diseño y simulaciones pertinentes, bien sea mediante software o mediante prototipo, con la finalidad de poder comprobar la funcionalidad del bloque diseñado. Finalmente, se integrarán los bloques para lograr un prototipo totalmente funcional, que satisfaga los requisitos previstos.

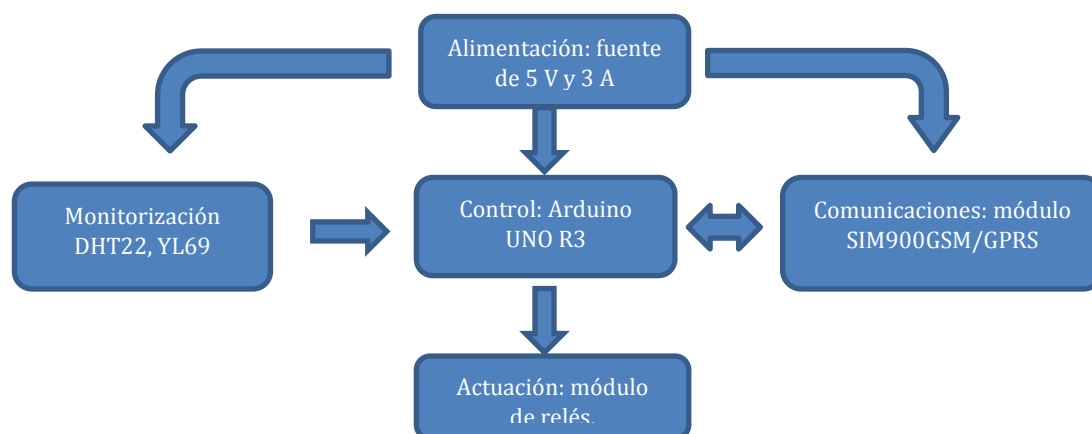


FIGURA 1. DIAGRAMA DE BLOQUES DEL SISTEMA.

1.4 PLANIFICACIÓN DEL TRABAJO

A continuación se describen de los recursos necesarios para realizar el trabajo, las tareas a realizar y una planificación temporal de cada tarea utilizando un diagrama de Gantt. De este modo es posible marcar cuáles son los hitos parciales de cada una de las PEC.

1.4.1 ALCANCE.

Se enumeran a continuación los objetos que forman parte del alcance del proyecto:

- Contexto del proyecto.
- Justificación de la elección del sistema Arduino.
- Planificación del proyecto.
- Selección de elementos hardware necesarios.
- Codificación del software necesario.
- Diseño del circuito.
- Pruebas y simulaciones mediante software y prototipos.
- Redacción de la memoria del proyecto.
- Realización de la presentación en video del proyecto.

No está previsto que formen parte del alcance los siguientes:

- Implementación física del sistema (creación del circuito impreso con componentes).
- Validación del sistema en el lugar de destino (viñedo).

1.4.2 HITOS.

Los hitos del proyecto corresponden a las entregas parciales y a la entrega de la memoria y de la presentación. Así mismo, se considerarán también como hitos los diferentes borradores correspondientes a cada entregable. También, se añaden como hitos la fecha de inicio del proyecto y del debate virtual, tabla 1.

| Grupo | Hito | Fecha |
|------------------------|---------------------------------------|-------------------|
| PEC 1 | Borrador PEC 1 | 10/03/2017 |
| | Entrega | 13/03/2017 |
| | Planificación | |
| PEC 2 | Borrador PEC 2 | 12/04/2017 |
| | Entrega PEC 2 | 15/04/2017 |
| | | |
| PEC 3 | Borrador PEC 3 | 14/05/2017 |
| | Entrega PEC 3 | 19/05/2017 |
| | | |
| PEC 4 | Borrador PEC 4 | 14/06/2017 |
| | Entrega Memoria y presentación | 19/06/2017 |
| | | |
| Debate virtual. | Respuestas. | 30/06/2017 |

TABLA 1. HITOS.

1.4.3 CALENDARIO DE TRABAJO.

El TFG tiene una carga de 12 créditos, de modo que se trata de repartir 300 horas (12 créditos x 25 horas/crédito) entre 15 semanas que hay desde el comienzo del proyecto hasta su entrega definitiva. De este modo, se puede considerar una carga semanal de 20 horas, por lo que si se establece una dedicación de dos horas diarias de lunes viernes y diez horas a lo largo del fin de semana, se estaría cumpliendo con la distribución temporal necesaria. Así mismo, se considera la oportunidad de aprovechar los festivos existentes para incrementar dicho número de horas y tratar de adelantar todo el trabajo posible, para evitar los imprevistos que puedan acontecer a lo largo del proyecto. Como puede verse, se trata de una fuerte dedicación al proyecto dentro del tiempo libre que existe fuera del horario laboral. En la tabla 2 puede verse la distribución de horas mensuales en función del número de días laborables y de fines de semana. .

| Mes | Días de L a V | Horas | S y D | Horas | Total de horas mensuales |
|-------|---------------|-------|-------|-------|--------------------------|
| Marzo | 22 | 44 | 8 | 40 | 84 |
| Abril | 20 | 40 | 10 | 50 | 90 |
| Mayo | 23 | 46 | 8 | 40 | 86 |
| Junio | 13 | 26 | 6 | 30 | 76 |

TABLA 2. DISTRIBUCIÓN DE HORAS MENSUAL.

El total de horas que se obtienen es de 336, de modo que, en caso de cualquier contratiempo, hay 36 horas extra sobre las teóricas 300 que requiere el TFG.

1.4.4 TAREAS Y DIAGRAMA DE GANTT.

En este punto se procede a definir las tareas que formaran parte del trabajo necesario para alcanzar cada uno de los hitos. De este modo, se puede desglosar el proyecto en las siguientes tareas:

1. PEC 1: Planificación (23/2/2017-13/3/2017).

- Concreción del TFG.
- Busquedas de información y documentación.
- Definición de hitos.
- Definición de tareas y diagrama de Gannt.
- Elaboración del documento PEC1.
- Revision y modificaciones.
- Entrega del plan de proyecto.

2. PEC 2. (14/3/2017-18/4/2017)

- Sistema Arduino en este proyecto: justificación.
- Software de diseño.
- Selección de sensores.
- Circuitos de adaptación de entrada y salida.
- Módulo de comunicaciones GSM/GPRS.
- Fuente de alimentación del sistema.
- Documentación PEC 2.
- Revisión y modificaciones.
- Entrega PEC 2.

3. PEC 3. (19/04/2017- 22/05/2017)

- Revisión feedback y modificaciones.
- Simulaciones del sistema y cambios.
- Comunicaciones del sistema.
- Simulación del sistema completo.
- Layout del circuito.
- Conclusiones y líneas futuras.
- Documento PEC3.
- Revisión e imprevistos.
- Entrega PEC3.

4. Finalización del proyecto. (23/05/2017- 30/06/2017)

- Revisión feedback y modificaciones.
- Finalizar la memoria.
- Elaborar la presentación.
- Revisión de memoria y presentación.
- Colchon temporal: imprevistos, mejoras y cambios.
- Entrega del proyecto.
- Debate virtual (no se contempla dentro del diagrama de Gantt).

Como puede verse, se ha tratado de ser escueto en la definición de las tareas para no alargar excesivamente el documento. Es obvio que todas estas tareas se subdividen en subtareas que no se han incluido en el diagrama de Gantt que puede verse en la figura 2, para mayor claridad. Por otro lado, hay que tener en cuenta que la planificación da lugar una presunción temporal donde se contemplan aproximadamente 100 horas para la PEC 2 y la PEC 3, ya que suponen la mayor parte de la carga de trabajo. Para la PEC4 se prevee una dedicación de 60-70 horas, que junto con las aproximadamente 30 horas de la planificación, da lugar a las 300 horas que suponen los 12 créditos del TFG .

Concluyendo, se trata de una planificación bastante ajustada a la carga lectiva, por lo que, en principio, si se deberían poder cumplir los objetivos básicos del proyecto.

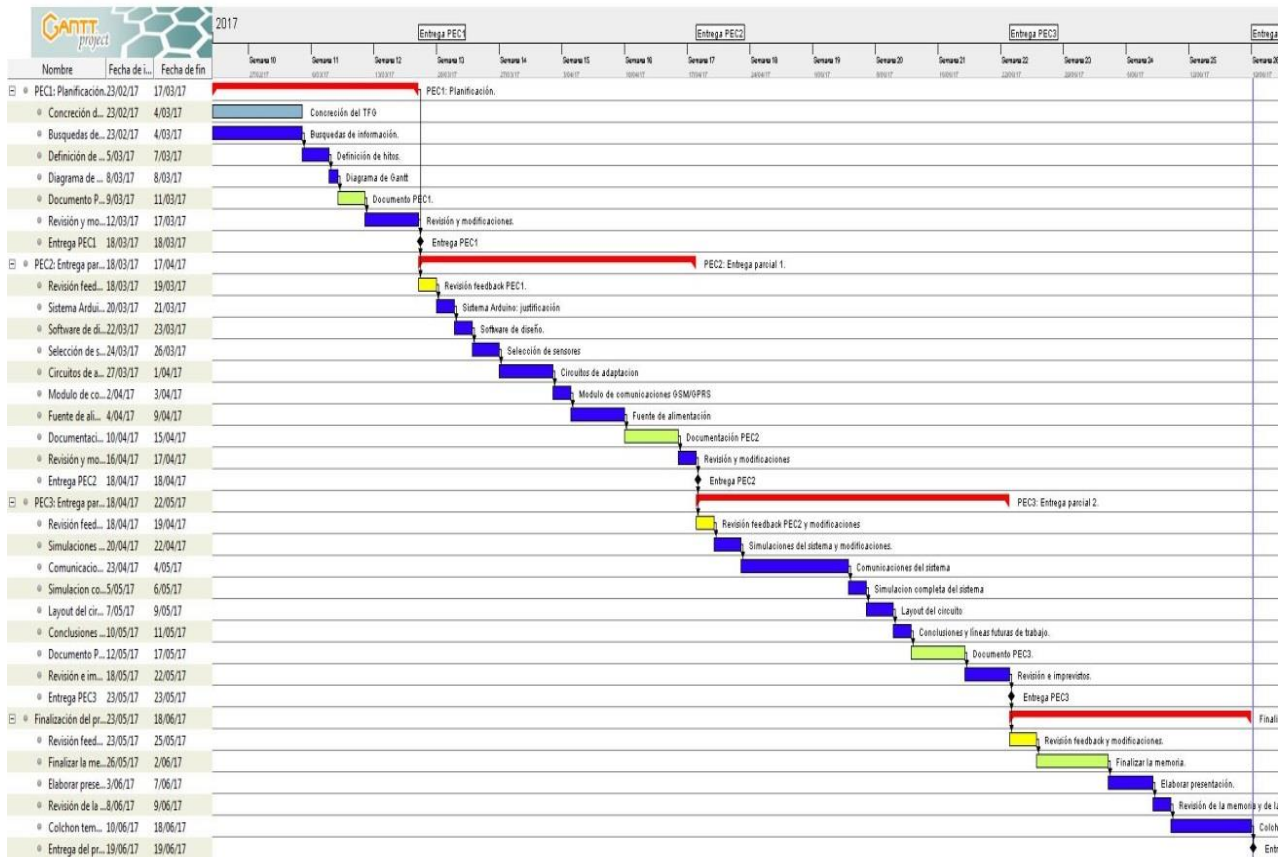


FIGURA 2. DIAGRAMA DE GANTT.

1.4.5 RIESGOS E INCIDENCIAS.

En este punto es importante tratar de advertir una serie de riesgos que pueden acontecer durante la elaboración del proyecto, con la finalidad de poder mitigar sus efectos en caso de que se materialicen. De este modo se exponen los que a priori pueden presentarse:

1. Desfases temporales respecto a lo planificado: probabilidad media, impacto medio. En este caso, se plantea como contramedida para evitarlo tratar de aprovechar al máximo los días festivos que existen a lo largo del proyecto. Es decir, tratar de ir siempre por delante de lo planificado en cuanto a la finalización de las diferentes tareas.
2. Problemas con el software y el hardware: probabilidad media, impacto alto. Como contingencia se empleará un segundo equipo informático para realizar las tareas que no puedan lograrse con el equipo principal. El primer equipo emplea Windows y el segundo Linux.

3. Pérdida de datos o documentación del proyecto: probabilidad baja, impacto alto. Para evitar pérdidas de información se realizarán copias diarias del trabajo realizado en una unidad de memoria externa y en la nube.

Una vez finalizado el proyecto se puede decir que no han surgido incidentes ni problemas que hayan supuesto un obstáculo en el normal desarrollo del proyecto. Solo comentar ciertas dificultades debidas a no estar familiarizado con todo lo empleado (hardware y software), lo que ha supuesto bastante esfuerzo de aprendizaje.

1.5 BREVE SUMARIO DE PRODUCTOS OBTENIDOS

Dado que el objetivo fundamental del presente trabajo es la obtención de un sistema que permita la captación de parámetros ambientales, la automatización de ciertos procesos y la gestión remota del mismo, se trata pues de obtener los siguientes productos:

- Prototipo hardware del sistema perfectamente operativo.
- Software que implemente las funciones elementales que demuestren el correcto funcionamiento del sistema.
- Memoria del trabajo, donde se expone detalladamente todo el proceso de diseño y su argumentación.
- Presentación del proyecto en Power Point o similar, para darlo a conocer en una primera aproximación.
- Breve guía de montaje del prototipo y pruebas.

Respecto al software, el archivo se adjuntará no dentro de los anexos sino como un archivo de Arduino, de forma que pueda ser compilado y ejecutado en el IDE de Arduino. De este modo, en caso de probarlo en un prototipo, se podrá ver su funcionamiento y los mensajes que el monitor serie muestra por pantalla.

1.6 BREVE DESCRIPCIÓN DE LOS OTROS CAPÍTULOS DE LA MEMORIA

Para la realización de este trabajo y, teniendo en cuenta que es un proceso de aprendizaje unido a un proceso de diseño, se divide en el conjunto en las siguientes partes:

- Capítulo 1: se describen los sensores que requiere el sistema para captar los datos que el proyecto necesita. Así, se ven sensores de temperatura y humedad ambiental y de humedad del suelo, que deben ser compatibles con Arduino.
- Capítulo 2: se pone en contexto que es Arduino y como emplearlo en este proyecto. De igual modo, se indica la placa elegida y sus características, dejando claro que no

es más que un microcontrolador con una serie de elementos auxiliares que permiten su empleo inmediato.

- Capítulo 3: trata del sistema de monitoreo de parámetros ambientales explicando las vertientes hardware y software., de forma que se llega a la realización de simulaciones en Proteus y al prototipo y pruebas reales de este subsistema.
- Capítulo 4: se explica el proceso de diseño de la fuente de alimentación que el sistema requiere, donde se identifican las tensiones y corrientes necesarias, de forma que se pueden determinar los elementos que conformarán dicho subsistema. Todo ello partiendo del supuesto de que existe una tensión de alimentación de $220 V_{AC}$ en el lugar de operación del sistema.
- Capítulo 5: se desarrolla la parte que permite al prototipo enviar y recibir información y órdenes de forma remota, empleando la red GSM/GPRS, lo que implica un proceso largo de aprendizaje y adaptación al sistema. Para ello, se requiere un módulo de comunicaciones que soporte esta tecnología, y sea compatible con Arduino.
- Capítulo 6: tratando de optimizar el sistema se plantea la posibilidad y el proceso de evolucionar el prototipo hacia algo más específico. Se trata de obviar todo lo innecesario de la tarjeta Arduino y aprovechar solo el microcontrolador y los elementos necesarios para su correcto funcionamiento. De igual modo, se incluye una pantalla LCD en el prototipo, con la finalidad de poder visualizar la información in situ.

Como puede verse, se trata de desglosar el proceso de diseño en una serie de capítulos que se corresponden con el orden lógico de creación del prototipo. Si bien, es cierto que se podría escoger cualquier otro orden, dado que se trata de desarrollar los diferentes bloques y después integrarlos; pero, como ya se ha mencionado, es el orden más lógico.

CAPÍTULO 1. ENTRADAS Y SALIDAS DEL SISTEMA.

En este capítulo se presentan las entradas y salidas que el proyecto incluye, realizando una descripción de los elementos que lo forman y sus características, así como la justificación. Como todo sistema que interactúe con el entorno, se requiere un conjunto de sensores que permitan al bloque de control tener un conocimiento preciso de los parámetros que requiere para realizar el proceso que se ha codificado. De igual modo, para realizar dichos procesos se requiere un conjunto de actuadores que lleven a cabo estas tareas. Con todo, no es un objetivo de este proyecto entrar en detalle sobre los actuadores empleados o los sistemas que se activan, sino simplemente realizar una pequeña puesta en escena de estos y dar a conocer que serán activados cuando se cumplan una serie de requisitos.

1.1 SENSORES.

En este punto se describen los sensores que requiere el sistema para captar los datos que el proyecto necesita. Así, se verán sensores de temperatura y humedad ambiental y de humedad del suelo. Los sensores a emplear serán, siempre que sea posible, sensores que sean compatibles con Arduino, de forma que las adaptaciones a realizar sean inexistentes o mínimas.

1.1.1 SENSOR DE TEMPERATURA Y HUMEDAD AMBIENTAL DHT22.

Como se ha comentado, se trata de escoger un dispositivo que requiera las mínimas adaptaciones electrónicas, de modo que este sensor es idóneo para emplearlo en un sistema basado en Arduino. En la figura 3 se puede ver una imagen de este modelo de sensor de temperatura y humedad ambiental. Si bien, es cierto que se podía haber seleccionado el DHT 11, ya que su precio es inferior, pero presenta peores características técnicas, las cuales pueden verse en la tabla 3. Aun así, no permite realizar lecturas con un periodo inferior a dos segundos, aunque esto no es un inconveniente en este proyecto.

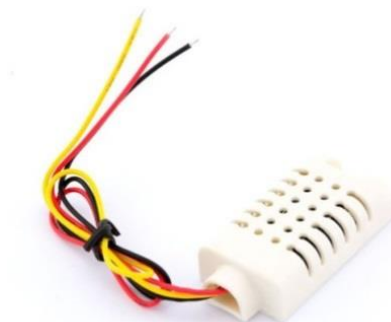


FIGURA 3. DTH22.

| <i>Modelo</i> | <i>DHT11</i> | <i>DHT22</i> |
|----------------------------------|--------------|--------------|
| Rango de medición de humedad | 20-90 % HR | 0-100 % HR |
| Rango de medición de temperatura | 0-50 °C | -40-80 °C |
| Precisión de temperatura | ±2 °C | ±0.5 °C |
| Precisión de humedad | ±5 % HR | ±2 % HR |
| Tiempo mínimo de lectura | 1 s. | 2 s. |
| Alimentación | 3-5 Vdc | 3.3-6 Vdc |
| Precio orientativo | 0.70 € | 2.40 € |

TABLA 3 DTH11 VS DTH22.

En la figura 4 puede verse un esquema de conexión y un ejemplo, donde se ve cómo se alimenta desde Arduino al sensor a través de los pines GND y Vcc del mismo. Por otro lado, la salida Output se conecta a una entrada digital de Arduino, siendo necesario poner una resistencia de 10K (pull-up) entre Vcc y el Pin Output. En este proyecto se emplea un módulo que integra el sensor y la resistencia de pull-up ya viene incluida, de modo que se conectará directamente a la tarjeta Arduino. Aparte, mencionar que DHT22 usa su propio sistema de comunicación bidireccional mediante un único conductor, empleando señales temporizadas. Así, en cada envío de medición, el sensor envía un total de 40 bits, en 4 ms. Estos 40 bits corresponden con 2 Bytes para la medición de humedad, 2 Bytes para la medición de temperatura, más un Byte final para la comprobación de errores (8 bit integral RH data + 8 bit decimal RH data + 8 bit integral T data + 8 bit decimal T data + 8 bit check sum) [1]. Finalmente, hay que aclarar que se requiere la librería DHT.h, para poder trabajar en el IDE de Arduino con más facilidad y así poder realizar las mediciones de temperatura y humedad ambientales, sin el inconveniente de complicarse con la codificación.

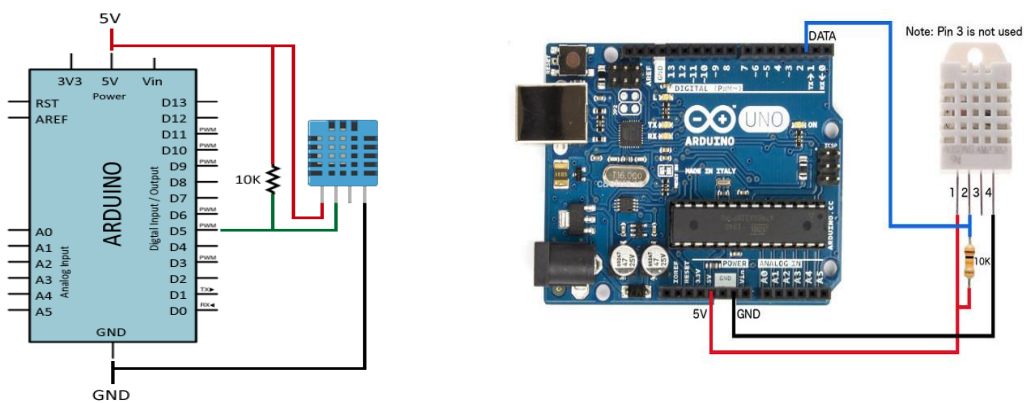


FIGURA 4. CONEXIÓN DHT 22 Y ARDUINO.

1.1.2 SENSOR HUMEDAD DEL SUELO YL-69.

Tenido en cuenta los módulos existentes en el universo Arduino, el sensor de humedad del suelo escogido es el que se muestra en la figura 5. Se trata de un sensor bastante empleado por los usuarios de este universo, dado que es totalmente compatible con Arduino cuya salida analógica ofrece un valor de voltaje en función del nivel de humedad. Además, presenta una salida digital que pasa a estado alto cuando se ha alcanzado la cota ajustada mediante el potenciómetro que regula el nivel de humedad. Consiste en una sonda YL-69 con dos terminales separados adecuadamente y un módulo YL-38 que contiene un circuito comparador LM393 SMD, un led de encendido y otro de activación de salida digital.

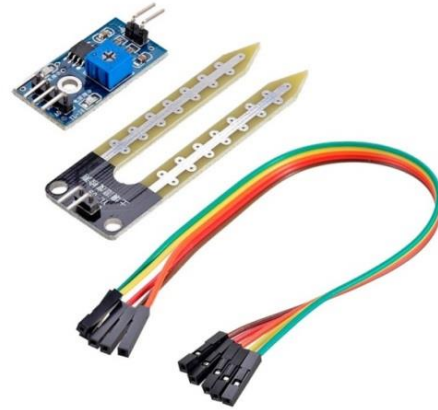


FIGURA 5. SENSOR HUMEDAD DEL SUELO YL-69.

Las características principales que presenta son:

- Alimentación: 3.3 a 5 V_{DC}
- Voltaje de Salida: 0 a 4.2 V_{DC}
- Corriente: 35 mA
- A0: Salida analógica que entrega una tensión proporcional a la humedad.
- D0: Salida digital; este módulo permite ajustar cuándo el nivel lógico en esta salida pasa de bajo a alto mediante el potenciómetro.

Aplicando una pequeña tensión entre los terminales del módulo YL-69 se hace pasar una corriente que depende básicamente de la resistencia que se genera en el suelo y ésta depende en gran medida de la humedad. Por tanto, al aumentar la humedad la corriente crece y al bajar la humedad la corriente disminuye [2]. Los valores obtenidos van desde 0 sumergido en agua, a 1023 en el aire (o en un suelo muy seco). Un suelo ligeramente húmedo daría valores típicos de 600-700. Un suelo seco tendrá valores de 800-1023. Por otra parte, la salida digital dispara cuando el valor de humedad supera un cierto umbral, que se ajusta mediante el potenciómetro. Por tanto, se obtiene una señal LOW cuando el suelo no está húmedo, y HIGH cuando la humedad supera el valor de consigna. En la figura 6 se muestra un ejemplo de conexión a la placa Arduino, incluyendo las salidas analógica y digital.

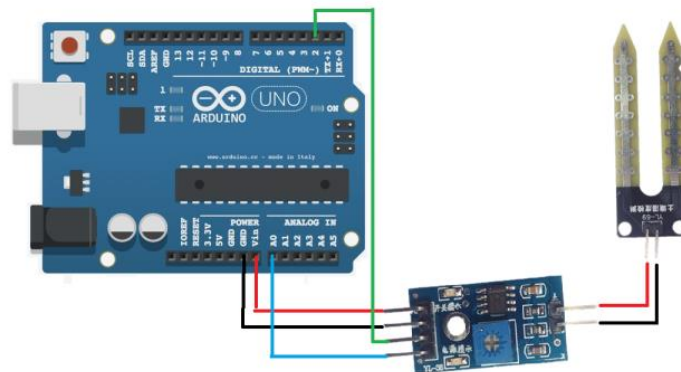


FIGURA 6. CONEXIÓN SENSOR YL-69 Y ARDUINO.

1.2 ACTUADORES.

En esta sección se enumeran los actuadores empleados en el proyecto. No se profundizará en sus características y aspectos técnicos, ya que queda fuera del alcance del presente proyecto. Por tanto, se planteará su existencia en el proyecto y una breve explicación.

1.2.1 SISTEMA DE RIEGO AUTOMÁTICO.

Se trata de un típico sistema de riego de una explotación agrícola basado en el goteo. En este caso al tratarse de un viñedo, los tubos de riego discurren por debajo de las cepas como puede verse en la figura 7. De este modo, los goteros se colocan justo en la base de la planta, produciéndose un riego exclusivo del terreno más próximo a la raíz de la planta.



FIGURA 7. RIEGO POR GOTEO.

En la figura 8 se puede ver un esquema de un típico sistema de riego, en el cual pueden apreciarse todos sus componentes. Dado que no es objeto de este trabajo profundizar más en este tema, simplemente comentar que la parte que interesa es la de activación del riego. En consecuencia, el presente proyecto debe permitir actuar sobre la activación y parada del sistema de riego, en base a unos parámetros que automatizarán el proceso. Así, la

activación del sistema de riego, al igual que puede realizarse manualmente, debe ser posible de forma automática (código Arduino) o de forma remota manual (GSM/GPRS).

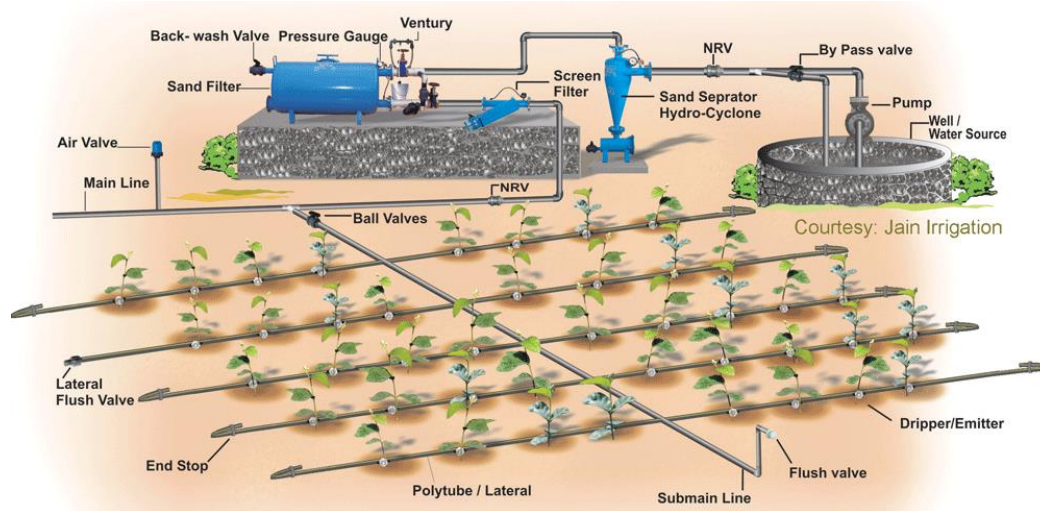


FIGURA 8. ESQUEMA SISTEMA DE RIEGO.

Concluyendo, se trata de automatizar y/o facilitar este proceso, que incluso podría servir para realizar tareas de abonado y tratamientos fitosanitarios. Estos tratamientos podrían ir disueltos en el agua de riego, con la finalidad de optimizar y facilitar todas estas tareas.

1.2.2 SISTEMA DE GENERACIÓN DE VIENTO.

Se trata de un conjunto de torres eólicas con la particularidad de que su cometido es producir viento a partir de energía eléctrica o de motores de explosión. Su finalidad es producir corrientes de aire que eviten las heladas al amanecer, en periodos sensibles del ciclo vegetativo del viñedo. Como puede verse en la figura 9, no se trata de una torre de grandes dimensiones y es un elemento muy recomendado para todo tipo de plantaciones, sobre todo frutales y viñedos. Según [3], siempre ha sido una de las mayores preocupaciones el problema de las heladas, de modo que emplear sistemas que puedan combatir esta adversidad aportará un retorno económico año tras año en forma de evitar las pérdidas o disminución de producción por este fenómeno.



FIGURA 9. TORRES GENERADORAS DE VIENTO.

Por otro lado, lo que atañe a este proyecto es su activación, de modo que simplemente se requiere tener control sobre el mecanismo de activación y parada. Al igual que con el sistema de riego, interesará poder activarlo manualmente in situ, de forma manual remota (GSM/GPRS) o que Arduino, teniendo en cuenta los parámetros necesarios, determine su activación.

1.3 RESUMEN CAPÍTULO.

En este capítulo se ha presentado el conjunto de sensores y actuadores que forman parte del sistema que se desarrolla en el proyecto. Sin duda, no se trata de profundizar el tema del cultivo del viñedo y de los mecanismos que existen para su optimización, pero es interesante poder conocer algunos de los elementos existentes como parte introductoria de este proyecto. En resumen, si se trata de buscar una mejora en algún sistema, primero hay que conocer cómo opera el sistema y las partes que lo forman. Después, se tratará de buscar las mejores alternativas e introducir la innovación pertinente, dotando al conjunto de un valor añadido que permita su optimización. Por todo esto, solo se ha comentado la existencia de estos dispositivos y sistemas, siendo la parte importante determinar cómo interactuarán con el sistema de automatización y control remoto que se plantea en el presente trabajo.

CAPÍTULO 2. ARDUINO.

Para comenzar, se considera fundamental realizar una descripción del sistema Arduino con la finalidad de aclarar el motivo de emplearlo como base para la realización de este proyecto. Sin duda, el proyecto se puede realizar empleando la metodología tradicional consistente en diseñar todo el hardware desde cero y codificar el software de forma específica para el circuito diseñado. En cambio, se ha optado por Arduino ya que facilita los trabajos de crear los circuitos y la posterior codificación y carga de programa.

2.1 ENTORNO ARDUINO.

En los siguientes apartados se describe en que consiste Arduino y las partes que lo forman, ya que ni es hardware (placas de circuito impreso), ni es software (entorno de programación), sino más bien un conjunto muy completo y flexible que abarca ambos mundos.

2.1.1 ¿QUÉ ES ARDUINO?

Según [4], Arduino es una compañía de hardware libre y una comunidad tecnológica que diseña y manufactura placas de desarrollo de hardware y software. Estando compuestas, respectivamente, por circuitos impresos que integran un microcontrolador y un entorno de desarrollo (IDE) para programar cada placa. Por tanto, se trata de una herramienta para facilitar el uso de la electrónica en proyectos multidisciplinarios.

Por otra parte, como se puede ver en la figura 10, es el corazón de cualquier sistema formado por sensores y actuadores, de tal forma que, una vez definido un conjunto de estos dispositivos requeridos en cualquier aplicación, se puede programar en la placa un código que permita automatizar cualquier proceso.

Así, como puede verse en [5], gracias a su gran versatilidad y flexibilidad, se pueden aportar soluciones tecnológicas en cualquier ámbito, como puede ser: domótica, industria, robótica, agricultura, IoT (Internet de las cosas), etc.

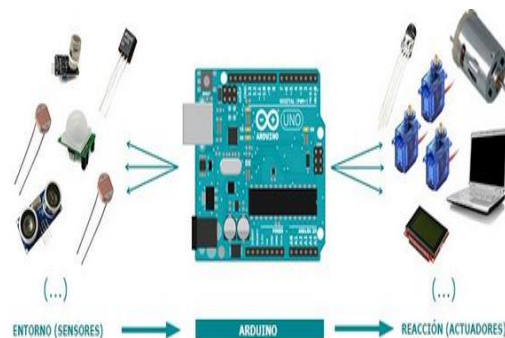


FIGURA 10. EL UNIVERSO ARDUINO.

2.1.2 HARDWARE ARDUINO.

En este punto, se pretende poner de manifiesto la variedad de placas existentes en la actualidad, sin entrar en detalles técnicos sobre las mismas. Simplemente, mencionar que en función de los requisitos de la aplicación, será más interesante el empleo de una placa u otra. Si bien es cierto que tienen un comportamiento muy similar, existen diferencias relativas a su número de entradas y salidas, tamaño, precio, potencia del microcontrolador, etc. Por tanto, cada aplicación puede emplear bastantes placas sin ningún problema; pero seguro que hay alguna que es óptima frente al resto, ya que no hay que olvidar el equilibrio entre solución y factor precio. Por nombrar algunas y sin entrar en sus detalles técnicos, se pueden mencionar las siguientes: UNO, LEONARDO, MEGA, EXPLORA, LILYPAD... siendo estas placas oficiales [6]. Es decir, también existen placas no oficiales que son muy similares, aunque no están fabricadas bajo la marca Arduino, por lo que llevan otros nombres. En la figura 11, se muestran algunas de las placas existentes.



FIGURA 11. PLACAS ARDUINO.

2.1.3 IDE ARDUINO.

El entorno de programación de Arduino, del cual se muestra una imagen de su interfaz en la figura 12, emplea un lenguaje similar al lenguaje de programación C, donde se introducen las líneas de comandos que componen lo que se denomina sketch (programa de Arduino). Como puede observarse hay dos bloques principales:

- **void setup:** en este bloque se realiza la configuración del dispositivo y la asignación de pines (entradas y salidas analógicas y digitales).
- **void loop:** se corresponde con el bucle que contiene el programa principal, donde se incluyen todas las sentencias necesarias para obtener la funcionalidad deseada.

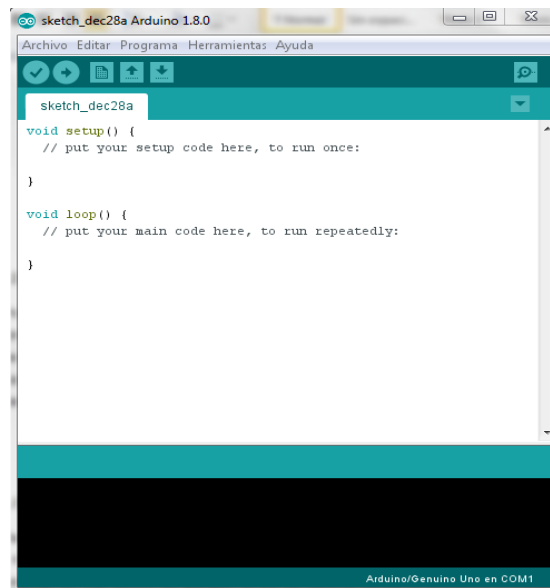


FIGURA 12. INTERFAZ DE PROGRAMACIÓN DE ARDUINO.

Por otra parte, en [4] se puede ver que existen otras interfaces de programación que pueden ser empleadas, ya que el único requisito es que permitan la transmisión por puerto serie. De este modo, se pueden emplear otros lenguajes de programación para programar una placa Arduino. Incluso, existen librerías diseñadas para permitir todo tipo de compatibilidades. Una alternativa muy interesante es emplear Sublime Text que, con las adaptaciones necesarias, permite trabajar con Arduino al igual que el IDE propio de Arduino, pero con ciertas ventajas que facilitan todo el proceso de codificación.

Aparte, es importante destacar que existen gran cantidad de librerías creadas por usuarios y que son de código libre. Así, se pueden emplear en cualquier proyecto con la finalidad de reutilizar código y poder simplificar el trabajo de programación. De esta forma, se consigue reducir el número de líneas de código, ya que simplemente se requiere importar una librería y definir variables de este tipo (similar a la programación orientada a objetos), facilitando en gran medida las tareas de programación.

2.1.4 COMUNIDAD ARDUINO.

Un factor del éxito de Arduino ha sido la comunidad que está apoyando este proyecto y que día a día publica nuevo contenido, divulga y responde a las dudas. En Internet hay disponible todo tipo de cursos, tutoriales, herramientas de consulta, proyectos, etc... que ayudan a que se pueda usar Arduino con facilidad [7]. Además, en [7] se ve que Arduino playground es un wiki donde todos los usuarios de Arduino pueden contribuir, siendo el lugar

donde publicar y compartir código, diagramas de circuitos, tutoriales, trucos, cursos y, sobre todo, el lugar donde buscar cuando existen dudas sobre un problema, una librería adecuada para un proyecto, siendo la base de datos de conocimiento por excelencia de Arduino.

2.2 ARDUINO EN EL PRESENTE PROYECTO.

En el anterior punto se ha podido ver superficialmente en qué consiste Arduino. A continuación, se tratará de exponer con mayor grado de detalle su implicación en el presente proyecto, poniendo de manifiesto las principales características de la placa empleada.

2.2.1 ARDUINO UNO R3.

La placa escogida para realizar este proyecto es la versión UNO R3, la cual puede verse en la figura 13 junto con su esquema eléctrico. Esto es así, simplemente, por el hecho de que es la placa que se recomienda para principiantes en el mundo de Arduino, siendo uno de los objetivos del proyecto familiarizarse con este universo de soluciones electrónicas.

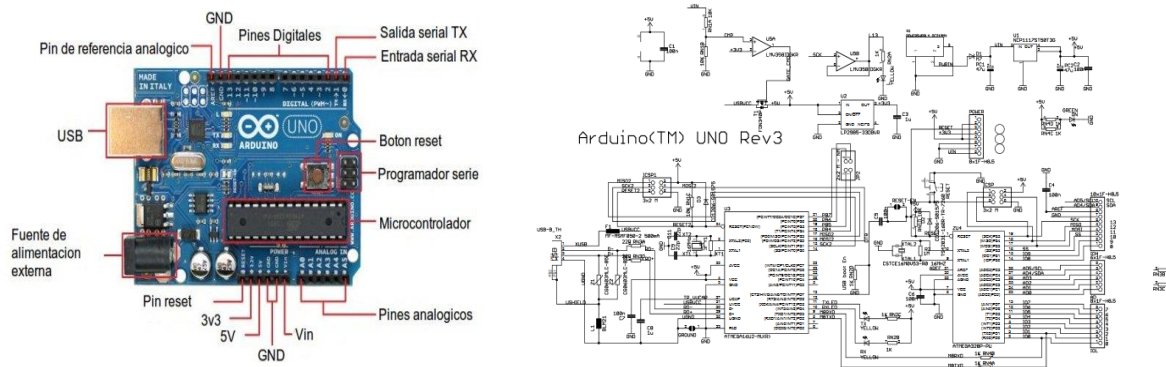


FIGURA 13. ARDUINO UNO R3.

En primer lugar, como puede verse en [8], las principales características que presenta este modelo de placa son:

- Microcontrolador ATmega328.
- Voltaje de entrada 7-12V.
- 14 pines digitales de I/O (6 salidas PWM).
- 6 entradas analógicas.
- 32k de memoria Flash.
- Reloj de 16MHz de velocidad.

En segundo lugar, como puede verse en la figura 13, la placa cuenta con un puerto USB que permite la conexión con un PC, para realizar la carga del programa y el intercambio de información. De igual modo, se ve que existen pines que permiten la transmisión serial con cualquier dispositivo que admita este sistema de comunicación. También, es importante destacar que presenta una entrada de alimentación auxiliar que puede variar entre siete y doce voltios, gracias al regulador de tensión que incorpora la placa. Aparte, para programar o para aplicaciones de muy bajo consumo puede emplearse la alimentación que proporciona el puerto USB.

En tercer lugar, dado que la placa presenta bastantes entradas y salidas, tanto analógicas como digitales, se puede afirmar que la capacidad de escalar el proyecto actual aumentando el número de sensores y actuadores sería muy factible. Además, el hecho de que proporcione salidas PWM permite emular salidas analógicas, lo que puede resultar muy conveniente cuando se pretenda controlar un actuador mediante una señal variable.

Finalmente, hay que mencionar la posibilidad de emplear un circuito basado en el microcontrolador de Arduino. Como puede verse en la protoboard de la figura 14, el circuito solo consistiría en los elementos necesarios para que el microcontrolador pueda realizar sus operaciones, una vez cargado el firmware correspondiente y conectadas las entradas y salidas necesarias. Una vez realizado el prototipo y verificado su funcionamiento, sería viable el diseño del layout y la fabricación de una PCB con las conexiones de entrada/salida necesarias, ocupando un espacio muy inferior al que requiere la placa Arduino.

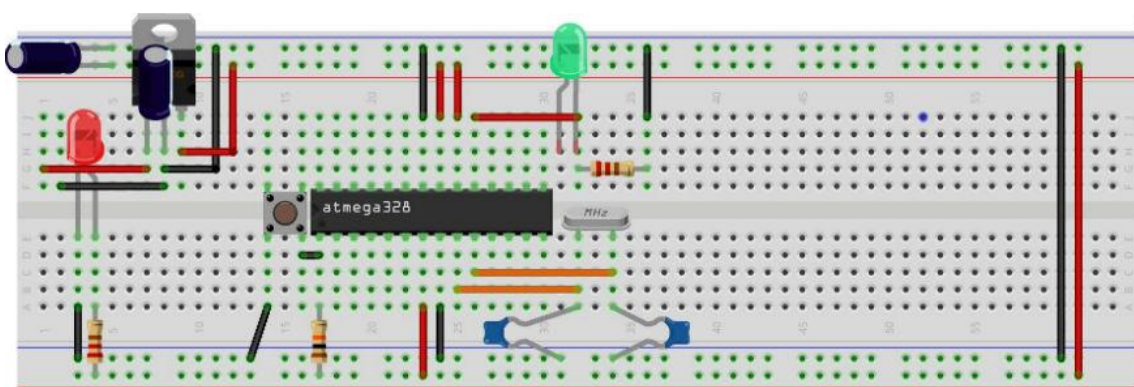


FIGURA 14. PROTOTIPO CON MICROCONTROLADOR ATMEGA 328.

2.3 RESUMEN DEL CAPÍTULO.

A lo largo de este capítulo se ha presentado Arduino, donde se ha puesto de manifiesto la gran versatilidad y potencia del sistema. Así, ha quedado de manifiesto que se trata de un entorno hardware y software que permite diseñar infinidad de soluciones electrónicas que pueden ser empleadas en todo tipo de ámbitos.

Por un lado, se ha expuesto la variedad de placas existentes, que permiten seleccionar la más adecuada en función del tamaño y complejidad del proyecto. También, se ha descrito el entorno de programación, donde se ha podido ver muy básicamente como es la interfaz que permite codificar el sketch y subirlos a la placa.

Por otro lado, se ha definido y detallado la placa Arduino UNO R3, que es la placa que se emplea en este proyecto como unidad de control del sistema. Se han especificado sus características principales (alimentación, reloj, memoria, comunicaciones, etc.) y se ha comentado la versatilidad de sus pines de entrada y salida.

Finalmente, se ha mencionado la posibilidad de emplear el microcontrolador de Arduino UNO (ATMega 328) como núcleo de un circuito diseñado específicamente para una solución, de modo que no sea necesaria la placa Arduino. Esto conlleva una reducción del tamaño bastante considerable, por lo que puede ser una buena opción de cara a optimizar el tamaño de un sistema electrónico. Para ello, se graba el sketch en el microcontrolador empleando la placa Arduino y se coloca este microcontrolador en el zócalo de la PCB creada.

CAPÍTULO 3. SISTEMA DE MONITOREO Y AUTOMATIZACIÓN.

A lo largo de este capítulo se detallarán todas las fases del diseño del sistema de automatización tanto a nivel hardware como software. Así, se pondrá de manifiesto cómo se realiza la conexión de entradas y salidas en la placa Arduino y la codificación necesaria del software, para que el sistema realice los procesos para los que está diseñado.

3.1 MONITORIZACIÓN DE PARÁMETROS Y SALIDAS DEL SISTEMA.

En el capítulo 1 se comentó que se trataba de emplear sensores que fuera posible conectar directamente a la placa Arduino sin necesidad de realizar circuitos de adaptación. Por otro lado, las salidas de Arduino requieren ser amplificadas para poder actuar sobre los sistemas que se plantean en este proyecto (sistema de riego y sistema antiheladas). Consecuentemente, a continuación, se procede a detallar como serán realizadas las entradas y salidas del sistema, teniendo en cuenta lo anterior.

3.1.1 ENTRADAS.

En primer lugar, debe quedar claro que para simplificar y ser práctico se ha decidido emplear sensores que se pueden conectar directamente a Arduino. Esto permite que sea más sencillo el proceso de realización del circuito y que presente mayor fiabilidad, debido a que se trata de dispositivos suficientemente probados.

Por otra parte, ya quedó explicado en el capítulo 1 el tipo de sensores y sus características, de modo que solo queda exponer como quedan conectados al sistema y como se tratará la información que faciliten. Así, los datos a tener en cuenta serán los siguientes:

Sensor de temperatura y humedad DHT22: se tendrá en cuenta la información suministrada (temperatura y humedad actual) con un periodo entre mediciones de diez minutos. En caso de que el usuario quiera conocer la temperatura y humedad del ambiente en tiempo real de forma remota, el sistema debe permitirlo. Por tanto, este sensor estará activo constantemente y será el código Arduino el que determine qué hacer con los datos recibidos.

Sensor de humedad del suelo YL-69: En este caso se puede decir que sucede lo mismo que antes, con la salvedad de que la unidad de control debe tener en cuenta un límite de humedad a partir del cual debe activarse el riego.

Para finalizar, hay que decir que dado que son sensores que están perfectamente adecuados para conectarlos directamente a la placa Arduino, no se requiere alargar más este punto. Así, más adelante se verá cómo se realizan las conexiones y cómo se emplean los datos.

3.1.2 SALIDAS.

En este caso, dado que hay que operar sobre un sistema de riego y un sistema antiheladas y teniendo en cuenta que los pines de salida de Arduino suministran una corriente máxima de 40 mA, es necesario amplificar dichas señales. Suponiendo que los sistemas comentados operan con tensiones de 220 V, la forma de poder actuar sobre estos sistemas será mediante el empleo de relés conectados a los pines de salida correspondientes. Así, se puede emplear un módulo de dos relés como el que se muestra en la figura 15, junto con su esquema eléctrico, donde en una placa preparada a tal efecto con los optoacopladores, transistores y resistencias precisas, se han montado dos relés SDR-05VDC-SL-C.

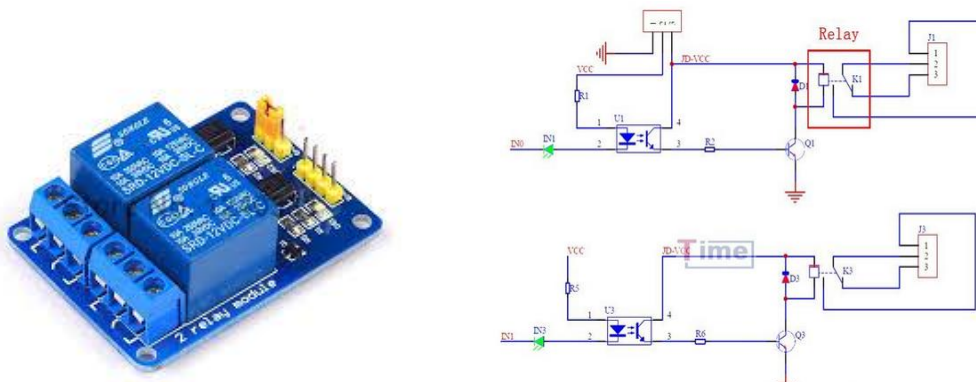


FIGURA 15. MÓDULO DE DOS RELÉS.

Como puede verse en [9], donde se describe un módulo con un único relé del tipo seleccionado, sus características permiten conectarlo directamente a la placa Arduino. Si bien, en la tabla 4 se muestran las características de consumos de la familia de relés SDR, se ha podido comprobar de forma experimental que la corriente de alimentación del relé requiere aproximadamente 70 mA, por lo que al tratarse de dos relés, la placa Arduino no podrá alimentarlo. Aun así, la señal para activar el cada relé requiere aproximadamente 12 mA, por lo que los pines de salida de Arduino sí podrán pilotar los relés. En consecuencia, será necesario tener en cuenta este factor al diseñar la fuente de alimentación del sistema, que deberá ser capaz de suministrar la intensidad que requiere la alimentación del módulo de relés.

| Coil Sensitivity | Coil Voltage Code | Nominal Voltage (VDC) | Nominal Current (mA) | Coil Resistance (Ω) $\pm 10\%$ | Power Consumption (W) | Pull-In Voltage (VDC) | Drop-Out Voltage (VDC) | Max-Allowable Voltage (VDC) |
|------------------------|-------------------|-----------------------|----------------------|---|-----------------------|-----------------------|------------------------|-----------------------------|
| SRD (High Sensitivity) | 03 | 03 | 120 | 25 | abt. 0.36W | 75%Max. | 10% Min. | 120% |
| | 05 | 05 | 71.4 | 70 | | | | |
| | 06 | 06 | 60 | 100 | | | | |
| | 09 | 09 | 40 | 225 | | | | |
| | 12 | 12 | 30 | 400 | | | | |
| | 24 | 24 | 15 | 1600 | | | | |
| | 48 | 48 | 7.5 | 6400 | | | | |
| SRD (Standard) | 03 | 03 | 150 | 20 | abt. 0.45W | 75% Max. | 10% Min. | 110% |
| | 05 | 05 | 89.3 | 55 | | | | |
| | 06 | 06 | 75 | 80 | | | | |
| | 09 | 09 | 50 | 180 | | | | |
| | 12 | 12 | 37.5 | 320 | | | | |
| | 24 | 24 | 18.7 | 1280 | | | | |
| | 48 | 48 | 10 | 4500 | abt. 0.51W | | | |

TABLA 4. CARACTERÍSTICAS DE LOS RELÉS SDR.

Finalmente, hay que decir que este tipo de relé puede soportar una corriente máxima de salida de diez amperios a 220V, por lo que en caso de que sea necesaria más corriente o tensión para activar los sistemas de riego y antiheladas, sería necesario recurrir a contactores de mayor potencia, como el de la figura 16, que serían pilotados por el sistema mediante los relés de conectados a los pines de salida. Con todo, queda fuera de los objetivos del proyecto profundizar en estos aspectos, por lo que se comenta a modo de anotación.



FIGURA 16. CONTACTOR DE 380 V.

3.2 Diseño del circuito.

Esta parte del sistema consiste en un circuito muy simple, como puede verse en la figura 16. De este modo, basta con conectar los sensores que se comentaron en el capítulo correspondiente junto con los relés que se encargarán de accionar los sistemas de riego y antiheladas.

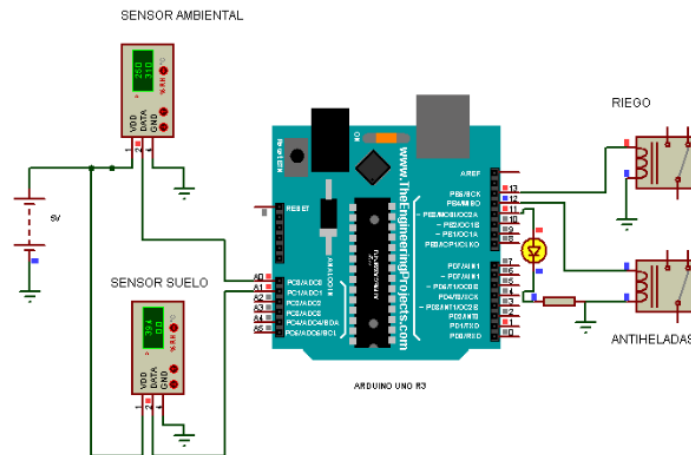


FIGURA 17. CIRCUITO DE AUTOMATIZACIÓN: ENTRADAS Y SALIDAS.

En el esquema –realizado con Proteus 8- puede verse que se ha empleado el mismo tipo de sensor para parametrizar los valores de temperatura y humedad ambiental así como la humedad del suelo. Esto es debido a que no se ha encontrado un sensor específico para este segundo fin en el programa. No obstante, de cara a realizar simulaciones, no supone ningún inconveniente, porque bastará con descartar el dato de temperatura del suelo.

Por otra parte, hay que decir que las conexiones son muy elementales, donde aparte de la alimentación de ambos sensores ($5 V_{DD}$), simplemente se ha conectado el terminal DATA a los pines analógicos de entrada de Arduino, debido a que la información que envían es información analógica. De igual modo, se aprecia que las salidas de la placa Arduino empleadas son los pines digitales 13 y 12, que son los encargados de activar los relés de riego y antiheladas al poner un valor alto (5 V). Además, en el pin 11 se ha conectado un led que sirve de testigo, para verificar que el programa ha comenzado a correr.

3.3 SOFTWARE DE MONITORIZACIÓN Y AUTOMATIZACIÓN.

El código creado para esta parte del sistema ha sido realizado con la herramienta Sublime Text, de la que se muestra una captura de pantalla en la figura 18. Como puede verse, el

archivo .ino creado es compatible con el IDE de Arduino, de forma que puede abrirse y realizar sobre el todo tipo de modificaciones (incluso compilación y cargarlo en la placa) con ambos entornos indistintamente. El programa completo se encuentra en la parte de Anexos al final de este documento.

```

TFG_DTH22.ino
9  #include <DHT.h>
10
11  int salidaRiego=13;
12  int salidaAntihehadas=12;
13  int testigo =11;
14  int tiempoRiego=5000;
15  int tiempoViento= 5000;
16
17
18  DHT sensorAmbiente(A0, DHT22);
19  DHT sensorSuelo(A1, DHT22);
20
21  void setup() {
22
23      pinMode(salidaRiego, OUTPUT);
24      pinMode(salidaAntihehadas, OUTPUT);
25      pinMode(testigo, OUTPUT);
26
27      Serial.begin(9600);
28      sensorAmbiente.begin();
29      sensorSuelo.begin();
30  }
31
32  void loop() {
33      digitalWrite(testigo, LOW);
34      delay(2000);
35      digitalWrite(testigo, HIGH);
36
37      float temperaturaAmbienta= sensorAmbiente.readTemperature();
38      float humedadAmbienta= sensorAmbiente.readHumidity();
39      float humedadSuelo = sensorSuelo.readHumidity();
40
41      Serial.println(temperaturaAmbienta);
42      Serial.println(humedadAmbienta);
43      Serial.println(humedadSuelo);
44
45      if(humedadSuelo < 40)
46      {
47          if(temperaturaAmbienta>30 && humedadAmbienta< 50)
48          {
49              Serial.println("Riego activado!!!");
50              digitalWrite(salidaRiego, HIGH);

```

FIGURA 18. CAPTURA DE SUBLIME TEXT.

En la codificación de ha tenido en cuenta lo siguiente:

1. Entradas:

- a. Sensor de temperatura y humedad ambiental: pin 8 (digital).
- b. Sensor de humedad del suelo: pin A1 (analógico).

2. Salidas:

- a. Sistema de riego: pin 13 (digital).
- b. Sistema antiheladas: pin 12 (digital).
- c. Indicador programa activo (led): pin 11 (digital).

3. Variables auxiliares:

- a. int tiempoRiego.: tiempo variable que estará activo el sistema de riego.
- b. int tiempoViento: ídem anterior para el sistema antiheladas.

Para las simulaciones se han propuesto las siguientes condiciones:

- a. **Sistema de riego:** se activará en caso de que el sensor de humedad del suelo envíe un valor inferior al establecido, además de que la temperatura ambiental sea superior a una determinada y la humedad del ambiente sea inferior a la preestablecida. Esto es para que el riego solo se active en verano, de forma que aunque la humedad del suelo sea muy baja en invierno no se active.
- b. **Sistema antiheladas:** en este caso solo dependerá de la temperatura ambiental, de forma que cuando se detecte una temperatura inferior a 2 °C, se activarán los molinos. La forma de evitar que este sistema se active durante épocas que no es necesario, es simplemente desconectar los molinos de su fuente de alimentación. Es decir, dado que el sistema antiheladas solo es útil durante los meses de abril y mayo, no es necesario tener conectados los molinos a su fuente de alimentación el resto del año.

Finalmente, hay que decir que el código empleado solo pretende mostrar un ejemplo de aplicación. Además, este código sirve para la simulación en Proteus; pero para una simulación real en Arduino hay que realizar algunas adaptaciones debidas a las diferencias de tratamiento de la información de los sensores en la simulación y en la realidad.

3.4 PROTOTIPO Y PRUEBA REAL.

A continuación, se propone un ejemplo de prototipo de esta parte del sistema junto con capturas del monitor serie. La finalidad de todo esto es ver desde un punto de vista más práctico y real como sería la implementación del sistema y los resultados. Así mismo, se pueden ver y corregir posibles errores que no se hayan percibido con anterioridad.

3.4.1. PROTOTIPO DEL SISTEMA DE MONITORIZACIÓN Y AUTOMATIZACIÓN.

En la figura 19 se ve el prototipo que se ha creado para poder simular mediciones de temperatura y humedad ambientales, así como de la humedad del suelo. Básicamente, en este caso, se ha empleado un sensor DHT11 para medir los parámetros del aire y un sensor YL-69 para la humedad del suelo. El YL-69 se encuentra conectado directamente a la placa (clon de Arduino), mientras que en la pequeña protoboard se ha conectado el DHT11. Del mismo modo, en la protoboard se han conectado tres leds cuya función es simular las salidas del sistema de automatización del riego (azul), antiheladas (rojo) y el testigo de

ejecución de programa (verde). Para ello, han sido necesarias tres resistencias de 330 Ω , cuya finalidad es limitar la corriente de forma que no se produzcan daños en los leds.

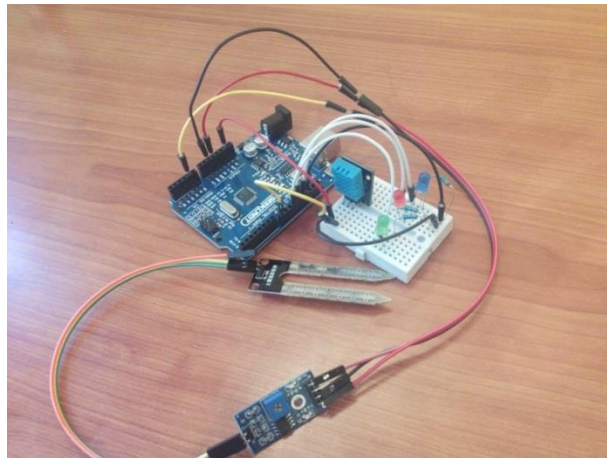


FIGURA 19. PROTOTIPO MONITORIZACIÓN DE PARÁMETROS Y AUTOMATIZACIÓN.

3.4.2 PRUEBAS.

En primer lugar, dado que es complicado llevar a extremos la temperatura y humedad ambiental sin emplear algún elemento de calor o frío, no se ha variado la temperatura entre extremos; pero se puede ver que el sensor detecta perfectamente la temperatura y humedad ambiental actuales en cualquier entorno. Por otro lado, para dotar de mayor realismo a la simulación se ha empleado una maceta, en la cual se ha insertado el sensor YL-69 para medir la humedad del sustrato. Así, aplicando un pequeño riego se puede comprobar cómo evoluciona la humedad del terreno. Todo esto puede verse en la figura 20.



Figura 20. Simulaciones de parámetros ambientales.

En segundo lugar, para que la placa se alimente y se pueda ver la información de los parámetros medidos, se requiere conectar la placa al ordenador mediante un puerto USB. De esta forma, el monitor serie que permite emplear el IDE de Arduino mostrará en pantalla los valores medidos, dado que se emplea una comunicación serie entre la placa y el ordenador. En consecuencia, una vez cargado el firmware en la placa, el sistema comienza a medir los datos y enviarlos al pc. Como puede verse en la figura 3.7, donde la imagen izquierda es una ampliación de lo que muestra la pantalla del PC de la figura 3.6, la humedad del suelo es diferente antes y después de aplicar un pequeño riego (imagen derecha de la misma figura).

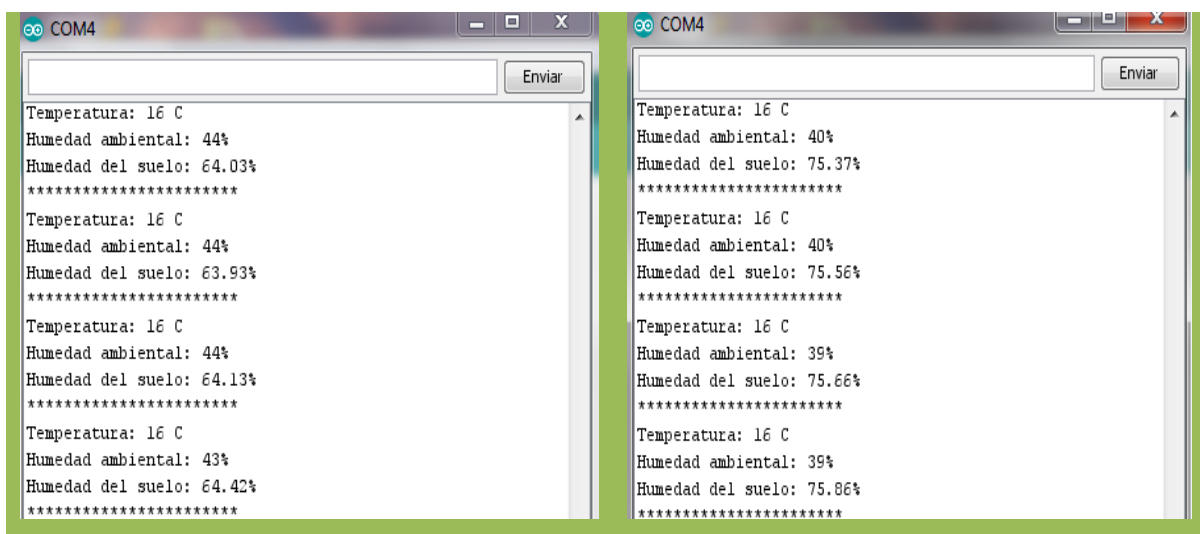


FIGURA 21. MONITOR SERIE: PARÁMETROS RECIBIDOS.

Finalmente, se ha comprobado que tanto el led azul como el verde (simulan los relés que pilotan estos sistemas) se activan en función de las condiciones que se impongan en la programación del microcontrolador. Por otro lado, el software que se ha empleado en la simulación real se encuentra en la parte de Anexos. En este código los parámetros que imponen las condiciones de activación de las salidas son valores para simulación, no los que se deberían emplear en un viñedo. Aun así, modificando el código se pueden emplear constantes que almacenen estos valores, de forma que bastaría con variar la definición de estas constantes, como por ejemplo el valor temperatura mínima que al ser sobrepasado activaría los molinos de viento.

3.5 RESUMEN DEL CAPÍTULO.

A lo largo de este capítulo se ha tratado de explicar las vertientes hardware y software del sistema de monitorización y automatización. Así, se han detallado las entradas y salidas del sistema empleando los sensores y actuadores que fueron comentados en el capítulo 1. De igual modo, se ha creado un esquemático con el software Proteus y se ha simulado variando los valores de entrada que ofrecen los sensores. En este punto, hay que decir que el código empleado es válido para la simulación en este entorno, ya que los sensores que ofrece Proteus no son idénticos a los que se emplearán en la realidad y ha sido necesario adaptarse a esta circunstancia.

Por otro lado, se ha realizado un prototipo empleando hardware similar al que se emplearía en el proyecto, con la finalidad de poder realizar una serie de simulaciones más aproximadas a la realidad. Por ello, ha sido necesario desarrollar un código que permitiera simular la automatización en función de los parámetros de entrada; pero teniendo en cuenta las características de los sensores empleados y que no se pueden variar las condiciones ambientales fácilmente. Aun así, se ha podido demostrar que los leds que emulan los sistemas de salida se activan en función de cómo se codifique el programa de forma correcta. Es más, se ha demostrado mediante el monitor serie que las lecturas de los sensores y la activación de salidas es totalmente coherente con lo programados en cada caso.

En resumen, la parte de implementación práctica y simulación del sistema expuesto en este capítulo confirma el correcto funcionamiento de lo que se planteaba de forma teórica. En consecuencia, se puede afirmar que el sistema cumple con las perspectivas esperadas y logra sus objetivos. Además, con pequeñas variaciones del código desarrollado, se puede adaptar el sistema a las condiciones de cada punto de explotación.

CAPÍTULO 4. FUENTE DE ALIMENTACIÓN.

A lo largo de este capítulo se describen todos los pasos necesarios para el diseño de la fuente de alimentación del sistema. Para ello, será necesario conocer las tensiones y corrientes que cada componente necesita para realizar su cometido. Se parte de la base de que hay disponible un acceso a una tensión 220 VAC. Aparte, hay que mencionar que aún no se ha comentado nada acerca del módulo de comunicaciones del sistema. Aun así, el módulo elegido es el SIM900 GPRS/GSM Arduino shield, del cual en su datasheet se pueden ver sus características eléctricas. Se profundizará en este módulo en el capítulo 6.

4.1 TENSIONES Y CORRIENTES EN EL SISTEMA.

En primer lugar, es necesario determinar las tensiones y corrientes que requiere cada elemento del sistema. De este modo, teniendo en cuenta las características técnicas que ofrecen los datasheets de los dispositivos empleados, se muestran en la tabla 3 los datos correspondientes.

| Dispositivo | Tensión | Corriente |
|--------------------------------|---------|-----------|
| Arduino UNO R3 | 7-12 V | 0.8-1 A |
| DHT22 | 3.3-6 V | 1-1.5 mA |
| YL-69 | 3.3-5 V | 35 mA |
| SRD-5VDC-SL-C x2 | 5 V | 71.4 mA |
| SIM900 GPRS/GSM Arduino shield | 5 V | 2 A |

TABLA 5. TENSIONES Y CORRIENTES DE LOS DISPOSITIVOS.

En segundo lugar, se debe tener en cuenta el conexionado de elementos a la tarjeta Arduino, de forma que hay que verificar qué tensiones y corrientes puede ofrecer esta tarjeta. Por tanto, se describen a continuación una serie de características relativas a tensiones y corrientes de la placa Arduino UNO R3:

- a. La placa se puede alimentar con en un rango de tensiones entre 7 y 12 voltios VDC con una intensidad máxima de 1 A. Aun así, no se recomienda superar 800 mA, debido a que se puede producir inestabilidad en su funcionamiento.

- b. Los pines de entrada y salida pueden ofrecer una tensión máxima de 5 V con una intensidad máxima de 40 mA.
- c. Los pines que pueden alimentar otros dispositivos presentan estos parámetros:
 - Pin 5 VDC: proporciona una corriente máxima de 300 mA.
 - Pin 3.3 VDC: proporciona una corriente máxima de 50 mA.
 - Pin VIN: proporciona la máxima tensión con la que se está alimentando a la placa.

En tercer lugar, en base a estos datos y a los presentados en la tabla 3, se puede ver que el único elemento del circuito que no puede alimentar Arduino directamente es el módulo de comunicaciones. Por tanto, será necesario que la fuente de alimentación que se proyecte tenga en cuenta este hecho. Si bien, en el capítulo 2 se planteó que Arduino se puede alimentar mediante el puerto USB, esto es válido para programación y simulaciones y no tiene sentido en el presente proyecto, debido a que el sistema debe operar alimentado mediante el conector Jack de la placa (7-12 V DC).

En conclusión, se requiere que la fuente de alimentación disponga de dos tensiones y corrientes diferentes en su salida:

- a. **alimentación de la placa Arduino (7-12 V_{DC} y 1 A)**, que alimentará los dispositivos comentados.
- b. **alimentación del módulo de comunicaciones (5 V_{DC} y 2A)**.

Hay que matizar que el módulo de comunicaciones no estará consumiendo 2 A constantemente, ya que su consumo típico es inferior a 450 mA; pero en ciertas ráfagas si puede llegarse a dicho nivel.

4.2 CIRCUITO DE FUENTE DE ALIMENTACIÓN.

Se parte de la base ya comentada de que en el punto de operación del sistema hay disponible un acceso a una tensión monofásica de 220 V. Por tanto, será necesaria una fuente de alimentación completa: transformación, rectificación, filtrado, regulación y protección. En la figura 22 puede verse un esquema de bloques de la fuente de alimentación, donde a la entrada hay una tensión alterna de 220 V.

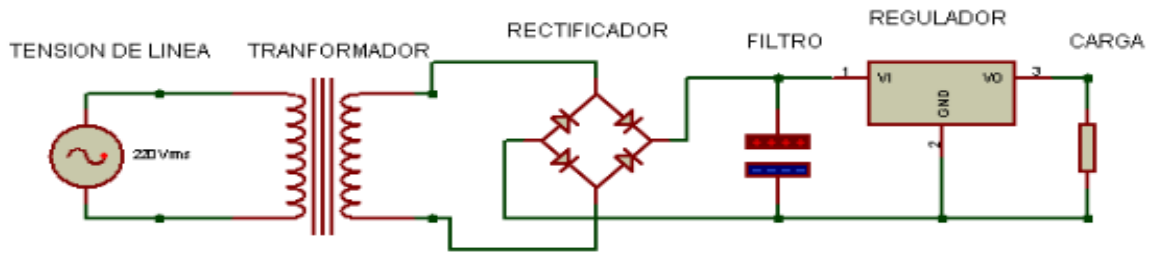


FIGURA 22. BLOQUES FUENTE DE ALIMENTACIÓN.

Entonces, a continuación se describen los componentes necesarios que incluirá cada bloque, teniendo en cuenta que hay que obtener la dos salidas comentadas anteriormente. Para ello, los dispositivos fundamentales son los reguladores de tensión, que proporcionan una tensión de salida fija en función del integrado escogido. Por ello, hay que tener en cuenta las características de estos dispositivos para escoger el resto de elementos de la fuente de alimentación.

- a. **Transformación:** se requiere un transformador que reduzca la tensión presente en el primario de $220 V_{RMS}$ a $15 V_{RMS}$ en el secundario, de forma que esta tensión equivale a $V_P = 1.41 \cdot 15 = 21 V_P$. Esto es así debido a que la tensión de pico obtenida a la salida del transformador, aproximadamente, es la tensión de continua que existirá en la carga, debido a las pequeñas pérdidas que introducen el resto de componentes.
- b. **Rectificación:** para obtener una señal que no presente semiciclos negativos, se emplea un puente de diodos o puente rectificador. Así, se obtiene una tensión que solo varía entre cero y el valor de tensión pico. En el esquema se emplea el modelo KBU4A.
- c. **Filtrado:** mediante el filtrado se consigue que la señal se aproxime lo máximo posible a una señal de tensión continua. Para ello, se emplean condensadores que se cargan según aumenta la tensión y se descargan cuando esta disminuye, dando lugar a una aproximación a tensión continua. Aun así, aparece una pequeña tensión de rizado que se debe minimizar. En este caso se emplean dos condensadores de 25 V y $10.000\mu F$.
- d. **Regulación:** la función de los reguladores de tensión es proporcionar una salida de tensión continua constante a un valor prefijado e independiente de la tensión que alimenta el regulador. Hay que tener en cuenta que los reguladores pueden ser fijos

o variables y de tensión positiva o negativa (respecto a tierra). Otro factor importante es que pueden alimentarse con una amplio margen de tensiones, pero siempre deben ser superiores a la tensión de salida que se pretende obtener (siempre consumen algo de tensión). Para tener una señal más pura se incluyen condensadores en la entrada y salida de cada regulador, siguiendo las especificaciones de los datasheets. En este proyecto se emplean:

- 7809: regulador de tensión con salida de 9 V y 1 A, para alimentar la placa Arduino.
- 78T05: regulador de 5 V y 3 A para alimentar el módulo de comunicaciones. En el esquema se emplean dos reguladores 7805 en paralelo para obtener una tensión de salida de 5 V y 2 A, ya que no se encuentra en el software el regulador requerido.

e. **Protección del circuito:** para proteger el circuito se deben emplear fusibles o diodos, incluso varistores, con la finalidad de que en caso de cortocircuito o sobretensión no se dañen ni los elementos del circuito creado ni se produzcan situaciones peligrosas en la red de alimentación. Se incluye un fusible rápido de 3 A, para proteger contra cortocircuitos, un varistor de 30 V para proteger contra sobretensión de alimentación y diodos rectificadores 1N4007 para proteger los reguladores.

Hasta aquí se han planteado los bloques funcionales de la fuente de alimentación, de modo que en el siguiente punto se detalla cómo será dicha fuente. Hay que tener en cuenta que el circuito no incluirá el transformador, para disminuir el tamaño de la placa, de forma que se simulará la señal de entrada al circuito como la de salida del secundario. En la figura 23 se muestran ejemplos de estos dispositivos, donde se ven de izquierda a derecha un puente rectificador, un condensador de 25 V 10.000 μ F, un regulador de tensión 7805 y un varistor de 25 V.

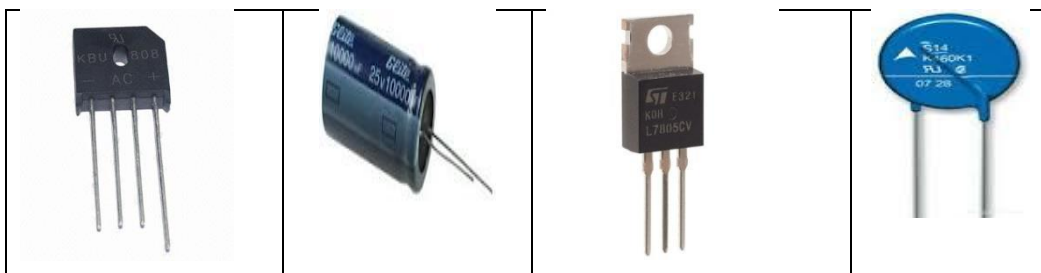


FIGURA 23. EJEMPLOS DE COMPONENTES EMPLEADOS EN LA FUENTE DE ALIMENTACIÓN.

4.4. LAYOUT Y MODELO 3D.

Para crear la placa de circuito impreso de la fuente de alimentación se ha empleado el esquema diseñado en Proteus, donde se han eliminado los dispositivos de medida y si se ha incluido un fusible de 3 A, un varistor de 20 V_{rms} y un conector de cinco pines. Este conector permite introducir la tensión proveniente del secundario transformador y alimentar tanto la placa Arduino como el módulo de comunicaciones, siendo común el pin de tierra en ambos casos. En la Figura 25 se representa el esquema listo para crear el layout del circuito definitivo.

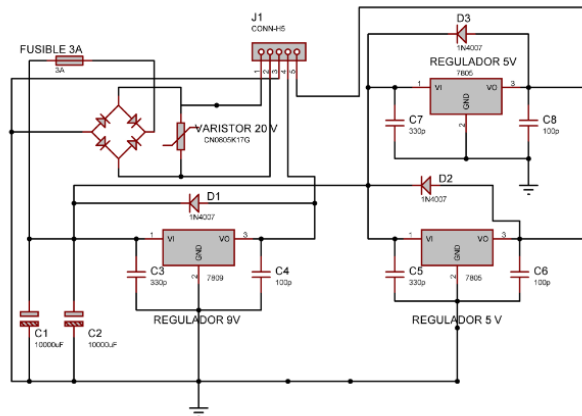


FIGURA 25. ESQUEMA FUENTE DE ALIMENTACIÓN.

Por otro lado, en la figura 26 se muestra el layout del circuito y una imagen en 3D de la placa de circuito impreso resultante del desarrollo de todo el proceso.

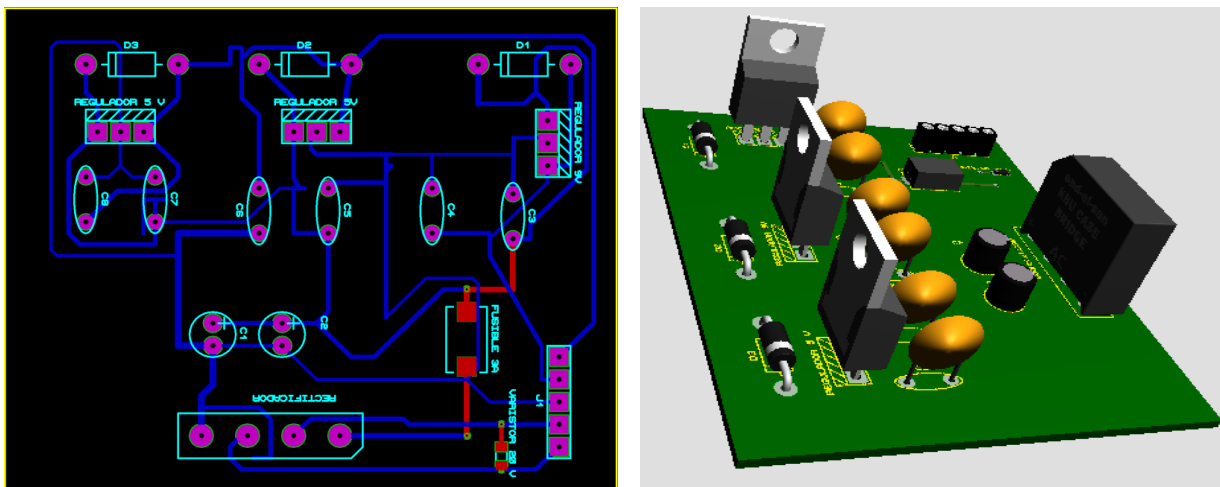


FIGURA 26. CIRCUITO BLOQUE FUENTE DE ALIMENTACIÓN.

4.5 RESUMEN DEL CAPÍTULO.

A lo largo de este capítulo se ha detallado el diseño de la fuente de alimentación que requiere el sistema. Por un lado, ha sido necesario determinar las corrientes y tensiones que requieren los distintos elementos para su funcionamiento. Por otro lado, ha sido necesario un análisis de los distintos elementos empleados para convertir una tensión alterna de 220 V_{RMS} en sendas tensiones continuas de 5 V y 9 V con sus corrientes máximas requeridas.

Aparte, mediante el proceso de diseño del esquema y su correspondiente simulación, se ha demostrado que los dispositivos empleados en la fuente de alimentación diseñada cumplen los requisitos de tensiones y corrientes que se plantearon al principio del capítulo. Así mismo, se ha desarrollado el layout del circuito para su posterior implementación en una PCB y se ha mostrado una imagen en 3D con la finalidad de previsualizar como sería el prototipo de dicha placa.

Concluyendo, se puede afirmar que la fuente diseñada será apta para alimentar los dispositivos del presente proyecto, presentando las funcionalidades requeridas, con un tamaño bastante optimizado. Además, como se ha demostrado, las tensiones y corrientes que se han obtenido son las adecuadas para el correcto funcionamiento del sistema.

CAPÍTULO 5. COMUNICACIONES EN EL SISTEMA.

En este capítulo se desarrolla el sistema de comunicaciones que empleará el sistema. Una vez finalizado el desarrollo del sistema de monitorización y automatización, el siguiente paso es dotar al sistema de las comunicaciones que permitan conocer los valores medidos y poder interactuar de forma remota. Para ello, se empleará la tecnología GSM/GPRS, de forma que el usuario pueda interactuar con el sistema mediante su terminal de telefonía móvil. Por tanto, en los siguientes puntos se presenta todo el desarrollo de esta parte del proyecto, incluyendo los argumentos necesarios y suficientes, sin entrar en una profundidad excesiva que no aporte utilidad al trabajo.

5.1 INTRODUCCIÓN A LA TECNOLOGÍA GSM/GPRS.

En primer lugar, en este punto se trata de realizar una aproximación a la tecnología que se empleará en las comunicaciones del sistema. Para ello, se recurre a la segunda generación de telefonía móvil y a una serie de evoluciones que permiten la transferencia de datos a través de Internet. Del mismo modo, se presentará GPRS considerada como la 2.5G y que permite transferencia de datos a través de Internet.

5.1.1 GSM (GROUPE SPECIAL MOBILE).

Se trata de un estándar de comunicaciones, también denominado a partir de 1991 como Sistema Global de comunicaciones Móviles, que a lo largo de diferentes fases ha ido evolucionando dentro de lo que se ha denominado como telefonía móvil de segunda generación. En Europa, el estándar GSM usa las bandas de frecuencia de 900MHz y 1800 MHz. Sin embargo, en los Estados Unidos se usa la banda de frecuencia de 1900 MHz. Por ello, los teléfonos portátiles que funcionan tanto en Europa como en los Estados Unidos se llaman tribanda y aquellos que funcionan sólo en Europa se denominan bibanda.

Por una lado, el estándar GSM soporta un tráfico máximo de 9,6 kbps, que permite transmisiones de voz y de datos digitales de volumen bajo: mensajes de texto (SMS, Servicio de mensajes cortos) o mensajes multimedia (MMS, Servicio de mensajes multimedia) [12]. Además, hay que decir que es un sistema que es totalmente digital, no como su antecesor que era analógico, donde se siguen empleando celdas para cubrir el territorio, donde cada celda contiene una estación base que da cobertura.

Por otro lado, como puede verse en [13] se pueden especificar una serie servicios, la arquitectura y evolución de GSM:

1. Servicios:

a. Básicos:

- Teleservicios: dan un servicio completo dentro de la red GSM. Proporcionan capacidad para comunicarse. Es el caso de la voz, fax grupo 3 (el grupo 4 va sobre RDSI), mensajes cortos (SMS, EMS, MMS), llamadas de emergencia...
 - Portadores (bearer): GSM proporciona sólo las capas bajas de la estructura OSI. Proporcionan capacidad para transmitir señales. Es el caso de las transmisiones de datos entre PDA (personal digital assistant) o PC mediante GSM.
- b. Suplementarios: son un valor añadido a los servicios portadores o a los teleservicios: limitación de llamadas salientes, multiconferencia, redirección de llamadas (si el abonado llamado está ocupado, no responde...), identificación de quién llama, restricción de identificación...

2. Arquitectura:

a. MS (mobile station, estación móvil)

- TE (terminal equipment): genera los datos.
- MT (mobile termination): es el conversor de señal eléctrica a radio (y al revés).
- TA (terminal adaptor): es la interfaz entre el TE y el MT.
- SIM (subscriber identity module): es la tarjeta que ponemos dentro del terminal.

b. BSS (base station subsystem, subsistema de estaciones base)

- BTS (base transceiver station): son las estaciones base. Entre otros elementos, aquí tenemos las antenas y los amplificadores de potencia.
- BSC (base station controller): controlador de las estaciones base.

c. NSS (network and switching subsystem, subsistema de red)

d. OSS (*operation support subsystem*, subsistema de gestión)

3. Evolución:

a. GSM fase 1 (1990)

- Uso del terminal y la tarjeta en cualquier lugar de Europa (itinerancia o roaming).

- Capacidad elevada, para que no se sature rápido.
 - Servicios de voz (telefonía) y datos (hasta 9.600 bps).
 - Frecuencias: 890-915 MHz y 935-960 MHz.
 - El coste está en la red, y no en los terminales.
 - Cifrado de la información, para ofrecer seguridad.
 - Uso del sistema de señalización internacional CCITT núm. 7. Lo usan los operadores para comunicarse entre sí (o entre elementos de la misma red).
 - Servicio de mensajes cortos (SMS).
 - Número único para llamada de emergencia.
 - Fax.
 - Restricción de llamadas.
- b. GSM fase 2 (1995)
- Ampliación del espectro: 880-915 y 925-960 MHz.
 - Permite utilizar un codificador más rápido (half-rate speech coder). Reduce el número de bits necesario para codificar la voz, manteniendo una calidad aceptable.
 - Identificación de quién llama, llamada en espera, llamada a 3...
 - Opciones específicas para grupos de usuarios: llamada de grupo, llamada para todos...
- c. GSM fase 2+ (1996 en adelante)
- Terminales duales GSM/DCS.
 - Interconexión con DECT.
 - CAMEL (customized application for mobile network enhanced logic).
 - Ampliación de la funcionalidad del SIM.
 - Permite llamadas de emergencia combinando un terminal GSM y un receptor GPS.
 - GPRS (1997).
 - UMTS (1999).
 - HSDPA (3GPP release 5, 2001).
 - HSUPA (3GPP release 6)
 - LTE (3GPP release 8)

Finalmente, mencionar que es una red que sigue desplegada conviviendo con las redes 3G y 4G, de forma que en entornos rurales se sigue empleando este sistema debido a su mayor alcance y a la baja densidad de población. Pese a que ya se está desarrollando la

tecnología 5G (probablemente se implementará a partir de 2020), no está prevista la reconversión del 2G a otras tecnologías más avanzadas en zonas de poca población, debido al poco interés comercial que supondría realizar estas inversiones.

5.1.2 GPRS (SERVICIO GENERAL DE PAQUETES DE RADIO).

Es una evolución del estándar GSM por lo que en algunos casos se denomina GSM++ (o GSM 2+). Dado que es un estándar de telefonía de segunda generación que permite una transición hacia la tercera generación (3G), el estándar GPRS por lo general se clasifica como 2.5G [14]. Permite la transferencia de datos del paquete con una tasa de datos teóricos de alrededor de 171,2 Kbps (hasta 114 Kbps en la práctica). Gracias a su modo de transferencia en paquetes, las transmisiones de datos sólo usan la red cuando es necesario. Por lo tanto, el estándar GPRS permite que el usuario reciba facturas por volumen de datos en lugar de la duración de la conexión, lo que significa especialmente que el usuario puede permanecer conectado sin costo adicional. Para el transporte de voz, el estándar GPRS emplea la arquitectura de red GSM y provee acceso a la red de datos (especialmente Internet) por medio del protocolo IP o del protocolo X.25. Además, admite características nuevas que no están disponibles en el estándar GSM y que se pueden clasificar en los siguientes tipos de servicios:

- Servicio de punto a punto (PTP): es la capacidad de conectarse en modo cliente-servidor a un equipo en una red IP.
- Servicio de punto a multipunto (PTMP): constituye la capacidad de enviar paquetes a un grupo de destinatarios (Multidifusión).
- Servicio de mensajes cortos (SMS).

Por otro lado, según [13], GSM fue pensado para transmitir voz, aunque ofrece la posibilidad de transmitir datos de hasta 9.600 bps. Se desarrollaron tres sistemas que permitían obtener más velocidad aplicando algunas modificaciones a la red GSM. Estos sistemas son el HSCSD, GPRS y EDGE, y se consideran sistemas 2.5G, donde queda patente que GPRS no es la única tecnología de intermedia entre la segunda y tercera generación de telefonía móvil. Además, este sistema es útil para aplicaciones que requieren muchas transmisiones de pocos datos o pocas transmisiones de un volumen medio de datos, dado que GPRS permite más velocidad que GSM.

5.1.3 SERVICIOS SOBRE GSM/GPRS.

Como puede verse en [13], sobre GSM y GPRS podemos tener varios servicios, además del servicio de voz:

- Servicio de datos: velocidades de hasta 9.600 bps (en el caso de GSM), aunque puede variar si entre los dos extremos establecemos protocolos de corrección de errores y/o de compresión.
- Servicio de mensajes cortos (SMS, short message service): puede ser punto a punto (hasta 160 caracteres, con la posibilidad de enviar mensajes más largos encadenando varios SMS), pero también puede ser punto a multipunto (por ejemplo, una autopista que envíe mensajes con el estado del tráfico a los móviles que están circulando por una determinada zona conflictiva).
- Datáfono: servicio dirigido a empresarios que no tienen a su alcance una línea telefónica fija y que necesitan disponer de la posibilidad de cobrar con tarjeta de crédito.
- Servicios de posicionamiento: como la red GSM sabe en qué estación base estamos y las operadoras saben las coordenadas de las estaciones base, hay servicios que aprovechan esta información para poder localizar el terminal.
- Pago por móvil: son sistemas en los cuales escribimos el número de móvil en el terminal punto de venta –TPV– en lugar del número de la tarjeta. El número de móvil recibe un mensaje en el que se pide autorización.
- Otros: correo electrónico, servicio de fax, acceso a Internet, telemedicina, etc.

5.2 TARJETA SIM900GSM/GPRS.

En este punto se procede a presentar la tarjeta o shield SIM 900GSM/GPRS para Arduino, que puede verse en la figura 27 y que será la encargada de permitir al sistema las comunicaciones necesarias con el usuario de forma remota.



FIGURA 27. TARJETA SIM900 GSM/GRPS.

Se trata de una tarjeta o módulo de expansión perfectamente compatible con cualquier tarjeta del universo Arduino, incluso es posible encastrarla sobre la tarjeta Arduino UNO R3, debido a que los conectores que posee están alineados con esta finalidad. En la figura 28 se describen los elementos principales.

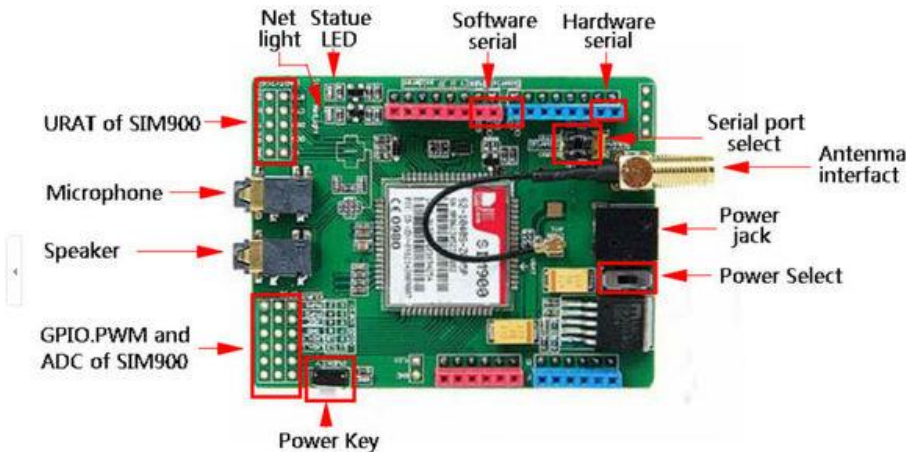


FIGURA 28. DESCRIPCIÓN FÍSICA DE LA TARJETA SIM900 GSM/GPRS.

Se adjunta en los Anexos el esquema eléctrico de esta tarjeta, de la cual a continuación se describen las características hardware y software.

5.2.1 HARDWARE.

Los elementos principales que conforman esta tarjeta son:

- Chip SIM900: en su datasheet [15] se pueden encontrar todas las características y funcionalidades. En la figura 29 se muestra el pin out del chip.
- Antena: tiene un zócalo para conectar una antena mediante un terminal para cable coaxial, directamente al chip.
- Zócalo tarjeta SIM: en la parte inferior de la tarjeta se encuentra el zócalo donde se debe insertar la tarjeta SIM desbloqueada. También en ese mismo lado se encuentra el alojamiento para una pila de botón de 3 V, que permite mantener la información cuando la tarjeta no tiene alimentación. Figura 29.
- Conector de alimentación tipo Jack: para alimentar la tarjeta con 5 V_{DC}. Normalmente, la intensidad operativa es de 450 mA, pero pueden haber ráfagas en las que se requiera hasta 2 A, por lo que la fuente de alimentación debe ser consecuente con este hecho.

- e) UART: comunicaciones serie asíncronas por hardware mediante los pines D0 y D8 o mediante software por los pines D7 y D8. Esta característica se configura mediante unos jumpers a tal efecto.
- f) Pines de entrada y salida: permiten la conexión con Arduino y presentan características similares aunque no coinciden todos. Hay pines analógicos, digitales y PWM.
- g) Conectores para micrófono y auricular: conectores de tipo minijack para emplearlos en llamadas telefónicas.
- h) Leds indicadores: indican el estado de los elementos de la tarjeta. En la tabla 5 se muestra la información que ofrecen.

Obviamente la tarjeta consta de toda una serie de componentes electrónicos (condensadores, resistencias, reguladores...) para permitir la correcta interacción entre los distintos elementos, pero no se entrará más en detalle en este aspecto.

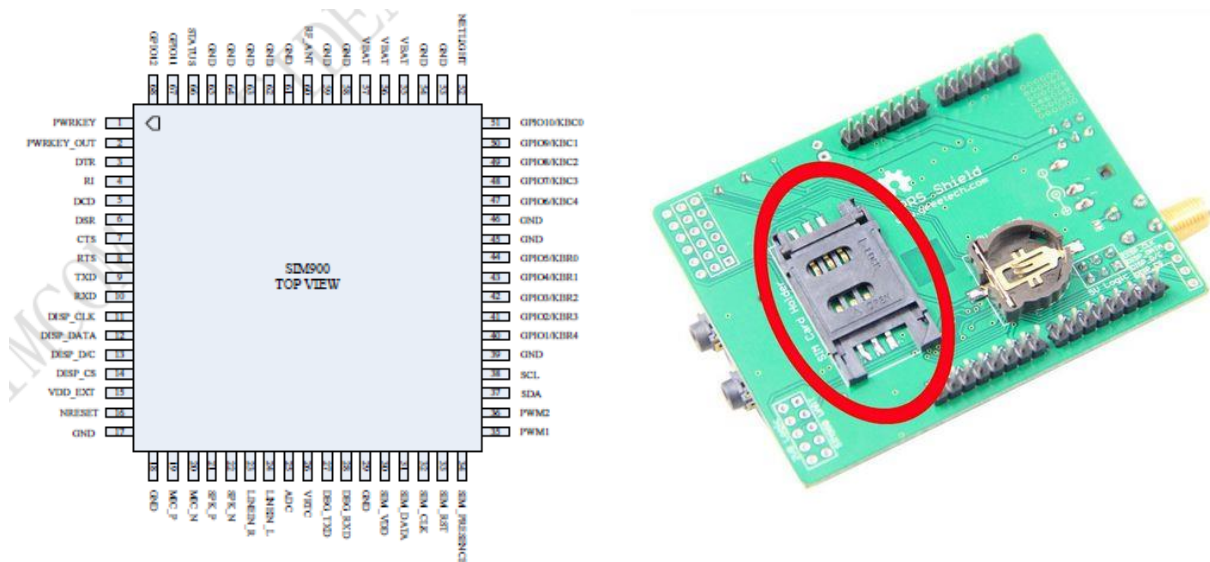


FIGURA 29. PIN OUT DEL CHIP SIM900 Y VISTA INFERIOR DE LA TARJETA.

| LEDs(color) | Status | Description |
|---------------|--------------------|--|
| PWR(Green) | ON | Power of the GPRS Shield is on |
| | OFF | Power of the GPRS Shield is off |
| Staus(Blue) | ON | SIM900 is on |
| | OFF | SIM900 is off |
| Netlight(Red) | 64ms On/800ms Off | SIM900 has not registered to a network |
| | 64ms On/3000ms Off | SIM900 has registered to a network |
| | 64ms On/300ms Off | GPRS communication |
| | OFF | SIM900 is not running |

TABLA 6. DESCRIPCIÓN DE LOS LEDS DE LA TARJETA SIM900 GSM/GPRS.

5.2.2 SOFTWARE.

Para programar las funcionalidades de la tarjeta se requiere emplear comandos AT y pese a que la lista de estos comandos es bastante extensa, en este caso se emplearán los que se detallan en [16]. Hay que aclarar que son los comandos AT y para ello, se puede decir que son lo que se conoce como el conjunto de comandos Hayes: 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 conociera también a este conjunto de comandos como comandos AT [17].

Por un lado, aunque la lista de comandos AT es muy extensa, los comandos principales que se emplearán son los siguientes:

AT: para verificar si el módulo SIM900 está funcionando adecuadamente para entrar en modo comando. Al enviar AT el SIM deberá contestarnos con un OK.

AT+CGMI: nombre del fabricante.

ATI: Ver la información del producto.

AT+IPR=?: el Baud Rate en el que puede operar el SIM.

AT+IPR?: para preguntar el Baud Rate actual.

AT+IPR=XXXX: configura la frecuencia deseada.

AT+COPS?: nombre de la compañía telefónica.

AT+CGSN: visualizar el IMEI del chip utilizado.

AT+CSCS?: tipo de texto.

AT+CSCS="XXX": Configurar a tipo de texto.

AT+CMGF?: Ver el formato de un mensaje, ya sea PDU(0) o SMS(1) .

AT+CMGF=1; // modo texto para SMS

AT+CMGS=04455XXXXXXXXX: Enviar un SMS Se despliega el símbolo mayor que

'>' Escribir mensaje y al finalizar presiona Ctrl+Z retornará OK si el SMS se envió correctamente.

AT+CMGL=ALL: Sirve para ver todos los mensajes que nos han llegado al SIM.

ATD04455XXXXXXX: Sirve para hacer una llamada a cualquier teléfono móvil.

ATA: Sirve para contestar una llamada.

ATH: Sirve para colgar una llamada.

Por otra parte, se pueden diferenciar dos operaciones software sobre la tarjeta:

- a) Programación y verificación de la tarjeta, donde conectando la tarjeta SIM900 GSM/GPRS a un PC, ya sea mediante un conversor FTDI que convierta la señal del puerto USB o bien empleando el conversor de la tarjeta Arduino, es posible verificar el funcionamiento y características de la tarjeta. Para ello, se recurre a un emulador de puerto serie como puede ser Hyperterminal o Teraterm o cualquier otro similar. Figura 30.

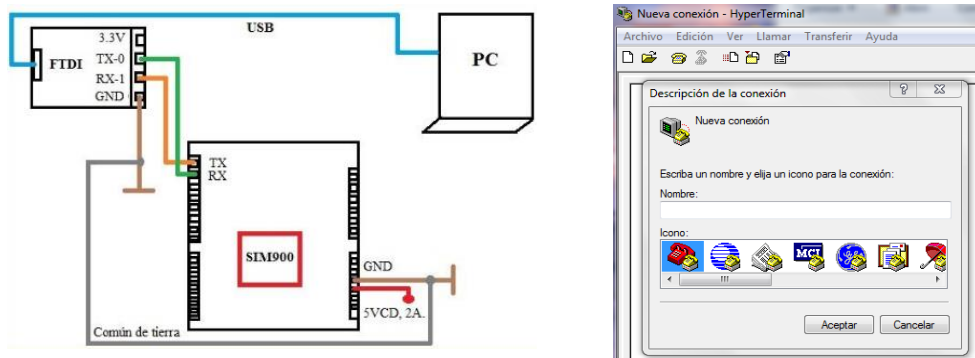


FIGURA 30. CONEXIÓN MEDIANTE CONVERTOR FTDI E HYPERTERMINAL.

- b) Programación del sketch en Arduino, donde los comandos AT son interpretados por el ATmega 328 de la tarjeta Arduino y, mediante la comunicación serie, se transmiten a la tarjeta de comunicaciones para que realice las operaciones correspondientes. Esto es, el propio sketch cargado en el microcontrolador incluye los comandos AT, de forma que dependiendo de qué programa este ejecutando, puede ordenar a la tarjeta de comunicaciones enviar o recibir datos. Por el contrario, también puede ser que sea la tarjeta de comunicaciones la que contenga alguna instrucción o datos que necesite transmitir al microcontrolador, por lo que esto debe estar incluido en el sketch. En todo caso la carga de programa es como se planteó en el capítulo

correspondiente, bien empleando el IDE de Arduino o bien mediante Sublime Text con el plugin para Arduino.

Finalmente, mencionar que todo esto que se ha expuesto será necesario emplearlo en el desarrollo del prototipo del sistema y en las simulaciones. Así, en el siguiente punto se muestra con ejemplos el desarrollo de todo el proceso de diseño del prototipo y simulaciones.

5.3 PROTOTIPO Y PRUEBAS.

En este punto se desarrolla el proceso de creación del prototipo, teniendo en cuenta todo lo expuesto en el anterior punto. Por ello, es necesario ir paso a paso para mostrar con claridad y concisión los pasos dados.

5.3.1 VERIFICACIÓN DE COMUNICACIONES.

En primer lugar, se procede comprobando la operatividad de la tarjeta de comunicaciones. Para ello, se realizan los siguientes pasos:

1. Se alimenta la tarjeta con $5 V_{DC}$ por el conector jack y se coloca el selector de alimentación en la posición de alimentación externa. La alimentación proviene de un regulador conectado a una fuente de $12 V_{DC}$.
2. Se acopla la antena.
3. Se inserta la tarjeta SIM desbloqueada en su ranura.
4. Se seleccionan los pines D0 y D1 como puerto de comunicaciones serie, mediante el jumper a tal efecto.
5. Se ejecuta Hyperterminal, introduciendo los parámetros necesarios y se verifica la operatividad del módulo mediante los parámetros AT. Ver figuras 32 y 33.

La tarjeta SIM900 se alimenta mediante un regulador que está conectado a una fuente de alimentación de $12 V_{DC}$. Además, se ha empleado la tarjeta Arduino como conversor USB a serie, para lo que es necesario quitar el microcontrolador. De este modo, se comunica directamente la tarjeta de comunicaciones con el PC, conectando en paralelo los pines Rx y Tx de ambas tarjetas. La alimentación de la tarjeta Arduino se realiza mediante el puerto USB, dado que aporta suficiente corriente para la operatividad de la tarjeta y permite que en el monitor serie del IDE se pueda ver lo que sucede. Figura 31.

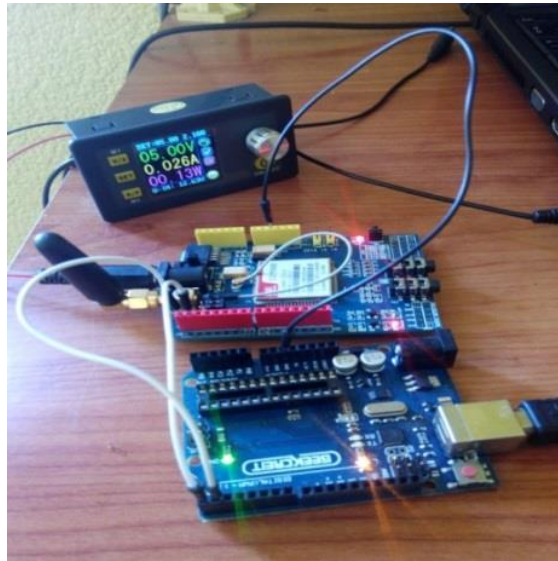


FIGURA 31. MONTAJE PARA PRUEBAS COMANDOS AT.

Una vez abierto Hyperterminal y configurados los parámetros de comunicación, se van introduciendo los comandos AT para ver si la tarjeta de comunicaciones está operativa, figura 5.6. Se puede ver que muestran tanto las velocidades de comunicación serie (AT+IPR=?), como la velocidad actual del módulo (AT+IPR?). De igual modo, informa del fabricante (AT+CGMI) y del IMEI del chip de comunicaciones (AT+CGSN). Con todo, hay que mencionar que este proceso se puede realizar con cualquier otro emulador de puerto serie, incluso con el monitor serie del IDE de Arduino, sin necesidad de cargar ningún sketch, previamente.

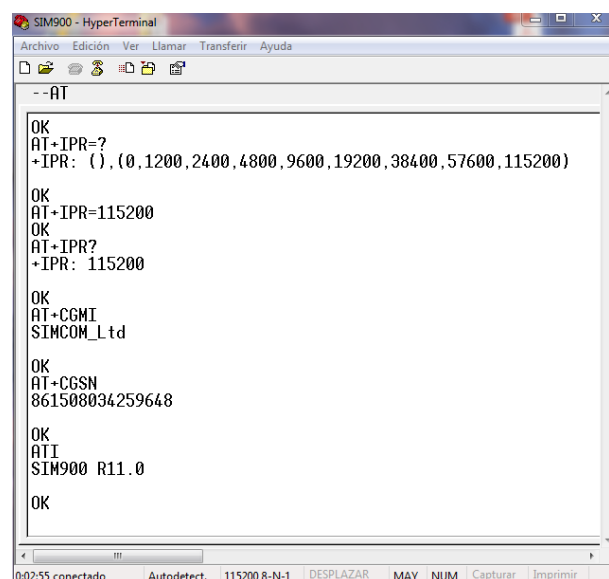


FIGURA 32. HYPERTERMINAL Y COMANDOS AT.

En segundo lugar, el siguiente paso es introducir una tarjeta SIM, para verificar que hay conectividad GSM, figura 33. Esto es necesario, debido a que puede ser que puede haber algún problema como: IMEI de la tarjeta bloqueado, no exista cobertura de servicio, fuente de alimentación sin la energía suficiente, etc. Así, pueden verse los comandos empleados

para verificar que hay comunicación mediante la red GSM: el comando AT+COPS? permite conocer a que operadora está conectada la tarjeta, con AT+CMGF? se informa del tipo de mensajes (PDU(0) o SMS(1)), con AT+CMGS="NUMERO DE TELÉFONO" se envía un SMS, para ello una vez introducido el comando aparece el signo '>' donde se escribe el mensaje y se pulsa CTRL+Z para enviarlo. Una enviado se informa sobre este hecho si no hay errores. Finalmente, se pueden realizar llamadas telefónicas empleando el comando ATDXXXXXXX(Nº de teléfono); y se puede aceptar una llamada entrante o finalizarla mediante ATA y ATH, respectivamente.

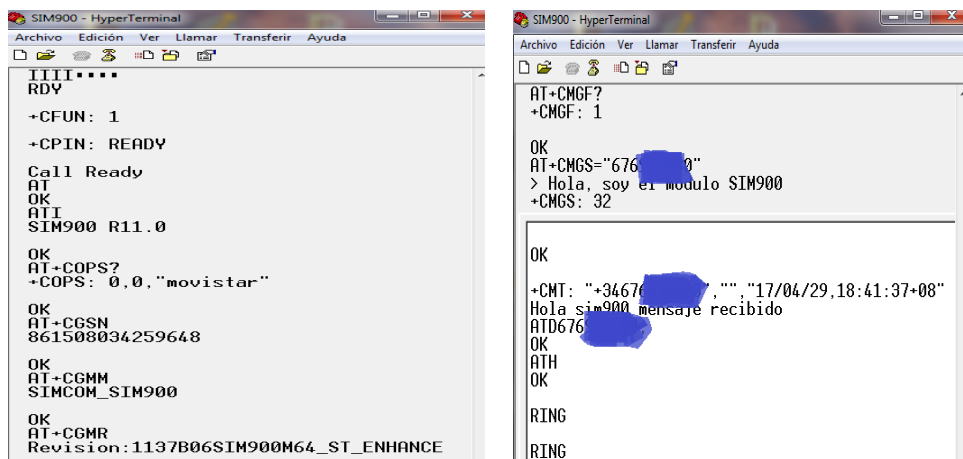


FIGURA 33. SIMULACIONES SIM Y SMS.

Existen muchos más comandos AT, pero con verificar algunos que se han probado es suficiente para tener la certeza de que el prototipo está preparado para comunicarse de forma remota.

5.3.2 SOFTWARE DE COMUNICACIONES.

Se procede a comprobar que el sketch que se diseñado para Arduino permite que el sistema pueda comunicarse con el exterior. Para ello, es necesaria una comprobación previa de que un pequeño sketch cargado en la tarjeta Arduino es capaz de enviar y recibir un SMS a través del módulo GSM/GPRS. Anteriormente, ya se ha comprobado que mediante Hyperterminal y empleando los comandos AT necesarios se puede realizar un a transmisión de datos y enviar y recibir llamadas; pero ahora hay que demostrar que el sketch de Arduino es capaz de realizar esto mismo. Así, se presentan las figuras 34 y 35, donde se demuestra que con el código que se incluye en los anexos, es posible enviar un SMS al sistema y que este realice las operaciones necesarias y responda con otro SMS.

En la imagen se ve que un móvil ha enviado un mensaje y ha recibido contestación del sistema, incluyendo los textos que presentarán la información de temperaturas y humedades. Para ello, se ha codificado en el sketch que el mensaje enviado por el móvil debe ser "@Info", de forma que este es el mensaje que permitirá que el sistema retorne las lecturas de temperatura y humedades. De este modo, para cualquier otro texto de SMS, el sistema no devolverá nada. Se puede decir que es la clave que permite al usuario realizar la petición de información al sistema.

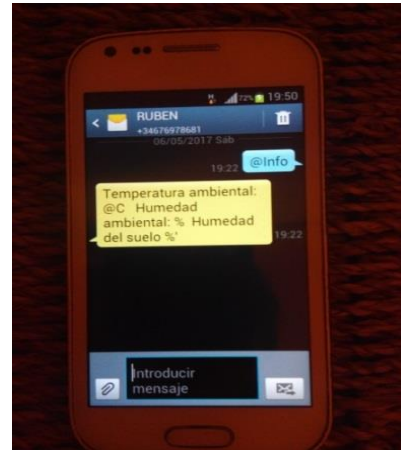


FIGURA 34. SMS ENVIADO A UN MÓVIL.

Por otro lado, en la figura 35 se muestra una captura de pantalla del monitor serie del IDE de Arduino, donde se puede ver como evoluciona el sistema en espera de recibir algo a través del módulo de comunicaciones. Así, cuando se recibe un mensaje con el texto adecuado, el sketch esta codificado para que realice las operaciones oportunas, que en este caso sería mostrar cierta información por el monitor serie, encender un testigo de solicitud de información y enviar un SMS al móvil configurado como destinatario de dicha información.

```

COM3 (Arduino/Genuino Uno)
Esperando instrucciones...
Sistema listo...
Esperando instrucciones...
Sistema listo...
Esperando instrucciones...
Sistema listo...
Esperando instrucciones...
Sistema listo...
Esperando instrucciones...
Sistema listo...
Esperando instrucciones...
Sistema listo...
Esperando instrucciones...
Sistema listo...
Esperando instrucciones...
Sistema listo...
Esperando instrucciones...
Sistema listo...
Hay datos procedentes del modulo de comunicaciones!!!!!!!!!!!!!!
Se ha recibido una peticion de informacion!!!
Info
Procesando orden.....!!!
Mensaje recibido!!!
La temperatura y humedades son:.....
AT+CMGF=1
OK
AT+CMGR=?
OK
AT+CHMI=2,2,0,0
OK
AT+CMGS=SMS enviado al número 699 [redacted]
Fin de la funcion leerSMS().....
Sistema listo...
Esperando instrucciones...
Hay datos procedentes del modulo de comunicaciones!!!!!!!!!!!!!!
Fin de la funcion leerSMS().....
Autoscroll
    
```

FIGURA 35. RECEPCIÓN Y ENVÍO DE SMS (MONITOR SERIE).

Finalmente, hay que decir, que lo presentado hasta ahora es la demostración del proceso que permite verificar las comunicaciones entre el sistema y el exterior, mediante el módulo de comunicaciones SIM900 GSM/GPRS. De lo expuesto hasta este punto, se puede afirmar

que el sistema está perfectamente operativo y es capaz de realizar una comunicación bidireccional, que permita la interacción remota con el usuario final.

5.3.3 PROTOTIPO Y SKETCH.

El prototipo para comprobar que el sistema funciona se puede ver en la figura 36. En este montaje se ha recurrido al mismo montaje que el de la figura 31, donde se han añadido los elementos necesarios para la monitorización de los parámetros y se ha insertado el microcontrolador que contiene el sketch en su zócalo de la tarjeta Arduino. Así mismo, se han incluido una serie de leds para simular las salidas del sistema de riego y antiheladas y unos testigos para visualizar sucesos durante la ejecución.

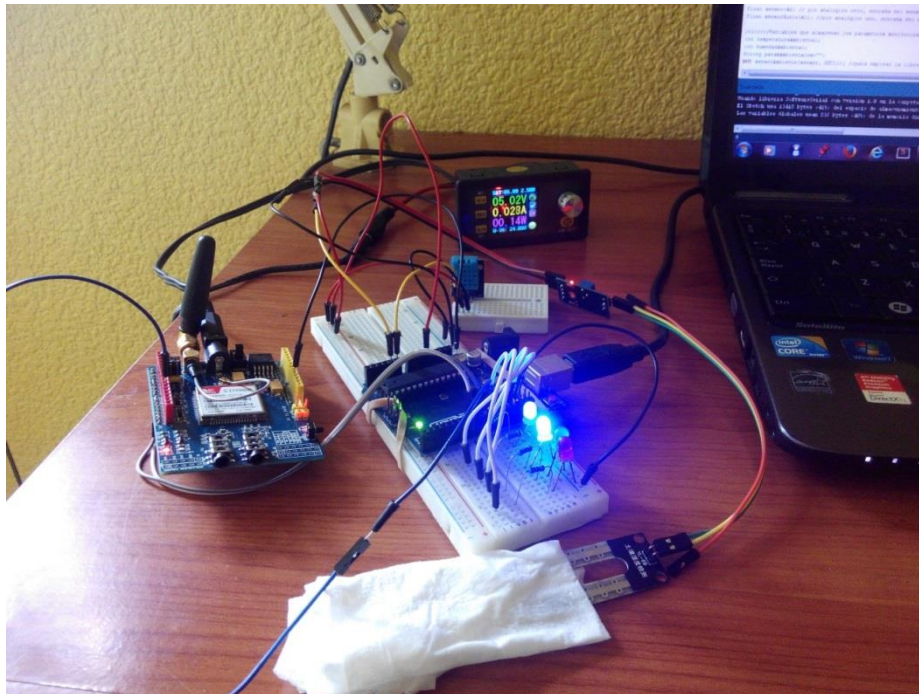


FIGURA 36. PROTOTIPO PARA SIMULACIÓN SISTEMA Y COMUNICACIONES.

Para mayor claridad, se comentan algunas conexiones del montaje realizado:

- pines 7 y 8 de la SIM 900 conectados a los mismos pines de Arduino, ya que se emplean este puerto para las comunicaciones serie (por software).
- pin 9 al pin 9 de Arduino, de forma que se consigue iniciar la tarjeta de comunicaciones por software (enviando un pulso de 2 segundos).
- pin GND de la SIM900 conectado a GND de Arduino
- pines 13, 12, 11 y 10 de Arduino son los leds para visualizar las simulaciones.
- pines A0 y A1 de Arduino son las entradas de los sensores.

- los demás cables son alimentaciones y tierras.
- alimentación de la SIM900 mediante el regulador que puede suministrar la intensidad requerida a los 5 V_{DC} necesarios.
- alimentación de Arduino y comunicación serie con el PC mediante el conector USB, que corresponde al puerto serie por hardware (pines 0 y 1).

En todo caso, en los Anexos se incluye el sketch definitivo que permite que el sistema opere cumpliendo los objetivos requeridos: monitorizar los parámetros ambientales, automatizar los procesos de riego y de evitar el daño por heladas y poder monitorizar y actuar sobre el sistema de forma remota mediante la red GSM/GPRS. Aun así, se procede a comentar la estructura del sketch, para aportar mayor claridad:

1. En primer lugar hay una serie de definiciones de constantes, variable, pines y librerías que se requieren para las diversas operaciones.
2. En la configuración inicial (`void setup()`) se inicializan todo lo necesario, como declarar los pines como entradas o salidas, inicializar los puertos serie, inicializar la tarjeta de comunicaciones...
3. Se ha recurrido a crear diversas funciones que contienen los bloques de código necesarios para las diversa operaciones que se realizan: encender el módulo de comunicaciones por software, enviar un SMS, leer un SMS que llega por el puerto serie desde el módulo de comunicaciones, activar y desactivar los diversos sistemas en función de órdenes recibidas o de condiciones preestablecidas...
4. La función principal (`void loop()`) se encarga de estar constantemente monitorizando los parámetros recibidos por los sensores y compararlos con las condiciones que activarían las salidas correspondientes (riego y antiheladas). Además, también monitoriza su puerto conectado a la tarjeta de comunicaciones para ver si está llegando alguna orden mediante SMS, en cuyo caso la función encargada de leer el mensaje toma el control y llama a las funciones correspondientes.
5. Finalmente, hay que decir, que se han incluido gran cantidad de sentencias que permiten la presentación en pantalla del monitor serie del IDE de Arduino, para poder comprobar que sucede en cada momento. De este modo, modificando en el Sketch las temperaturas límite y tratando de actuar artificialmente sobre los sensores se puede ver qué información hay presente y cómo actúa el sistema en cada caso.

Con todo, dado que no es un objetivo del proyecto demostrar unas habilidades de programación elevadas, se considera que es sketch es más que suficiente para demostrar la

viabilidad y operatividad del sistema. Así, en los anexos, se presentan una serie de figuras (5.11, 5.12, 5.13, 5.14 y 5.15) donde pueden verse los resultados de diversas simulaciones de la parte de comunicaciones. De este modo, se puede ver como evoluciona el sistema dependiendo de las órdenes enviadas mediante SMS.

Finalmente, comentar que el carácter '@' se introduce para que resulte más sencillo identificar el texto del SMS del dentro de los datos recibidos desde la SIM900. Ya se comentó, que cuando llega un SMS el módulo de comunicaciones no solo envía el texto, sino otra diversa información. Por ello, se ha considerado oportuno identificar el comienzo del texto mediante un carácter que actúe de chivato. Es decir, cuando Arduino lee la información recibida desde el módulo de comunicaciones, solo almacena en el String que contendrá el mensaje de llegada los caracteres que suceden al carácter '@'. De esta forma, simplemente comparando este String recibido con los que se tienen almacenados, se determina lo que el sistema debe realizar.

5.4 RESUMEN DEL CAPÍTULO.

A lo largo de este capítulo se ha desarrollado y explicado todo el proceso que permitirá al sistema comunicarse con el exterior, permitiendo al usuario interactuar de forma remota mediante su terminal de telefonía móvil y la red GSM/GPRS.

En primer lugar, se ha realizado una pequeña introducción a la tecnología GSM/GPRS, detallando algunos aspectos técnicos, con la finalidad de contextualizar el entorno en el que se realizarán las comunicaciones. Dado que no se requiere una tasa de transferencia de información elevada, se ha considerado que es más que suficiente esta tecnología frente a la posibilidad de haber empleado otras como las de tercera generación y sucesivas.

En segundo lugar, se ha realizado una descripción básica del módulo de comunicaciones que se empleará en el proyecto, tanto a nivel de hardware como del software que se requiere para su control. Así, se ha explicado cómo es físicamente la tarjeta y su pin out para conocer cómo se puede conectar a Arduino. También, se ha explicado que son los comandos AT y se han enumerado algunos de los que se requieren para poder interactuar con el módulo de comunicaciones.

En tercer lugar, se ha realizado el montaje de un prototipo completo del sistema y el sketch que permitirá que sea operativo y realice las funciones básicas que se persiguen en el proyecto. Para ello, ha sido necesario entender y después demostrar cómo se puede

interactuar con el módulo de comunicaciones mediante los comandos AT. Por un lado, se ha demostrado como se pueden enviar y recibir SMS y llamadas conectando el módulo de comunicaciones directamente al PC. Esto ha permitido conocer que sucede cuando se envía un comando (que información devuelve la tarjeta de comunicaciones). Por otro lado, se ha realizado un primer sketch para verificar que una vez cargado en Arduino y conectado al módulo de comunicaciones, el sistema por si solo es capaz de enviar y recibir SMS. Así, se ha conseguido integrar este sketch con el de la parte de monitorización y automatización, para lograr el sketch definitivo que permita que el sistema sea plenamente operativo.

Concluyendo, pese a que se ha requerido bastante esfuerzo de investigación y muchas pruebas, se ha logrado obtener un sistema de comunicación que funciona perfectamente y que permite que este proyecto sea completo en cuanto a funcionalidades deseadas. Sin embargo, hay que comentar que se han incluido demasiadas líneas de código, para poder mostrar y demostrar mediante el monitor serie del IDE de Arduino cómo evoluciona el sketch en función de parámetros, condiciones y órdenes recibidas. Por ello, en un proyecto no académico, todas estas líneas de código no serían necesarias en el sketch definitivo, aunque podrían emplearse durante el proceso de pruebas.

CAPÍTULO 6. OPTIMIZACIÓN DEL CIRCUITO.

En este capítulo se presentan las fases y características que permitirán optimizar el hardware del circuito empleando el microcontrolador ATmega 328P y una pantalla LCD de 2x16 caracteres. La finalidad de este proceso es tratar de demostrar que, pese a que Arduino es una buena solución tecnológica para facilitar el aprendizaje y el desarrollo de aplicaciones electrónicas, no deja de ser un microcontrolador con una serie de periféricos. Además, el LCD permitirá visualizar datos in situ. De este modo, a lo largo de los diversos apartados, se tratará de exponer de manera clara y concisa todo el proceso y los argumentos pertinentes que permitan llegar a la solución optimizada.

6.1 MICROCONTROLADOR ATMEGA 328P.

El microcontrolador ATmega 328 p es el núcleo de la placa Arduino UNO R3. Por ello, es factible sustituir en el presente proyecto la tarjeta Arduino por un circuito que contenga este microcontrolador y los elementos necesarios para que pueda operar. En la figura 37 se muestra el pin out del microcontrolador y su equivalencia con los pines de entrada/salida de la tarjeta Arduino UNO R3. Entre las características principales de este microcontrolador [10] se pueden detallar las que figuran en la tabla 7. Con todo, toda la información necesaria puede verse en [11].

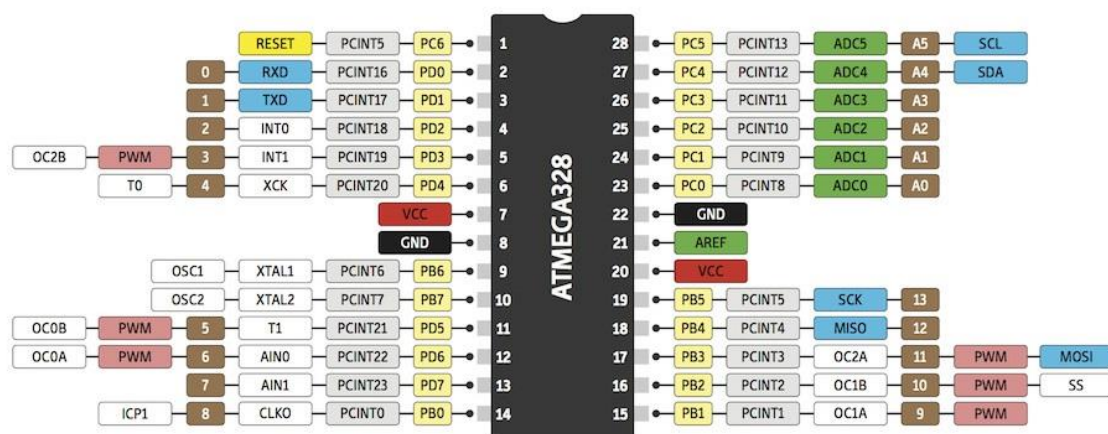


FIGURA 37. PINES ATMEGA 328 P Y EQUIVALENCIA EN ARDUINO UNOR3.

| PARÁMETROS | VALORES |
|--------------------------------|------------|
| Flash | 32 Kbytes |
| SRAM | 2 Kbytes |
| Cantidad Pines | 28 |
| Frecuencia máxima de operación | 20 MHz |
| CPU | 8-bit AVIR |
| Pines máximos de E/S | 23 |
| Interrupciones internas | 24 |
| SPI | 1 |
| UART | 1 |
| Canales ADC | 8 |
| Resolucion de ADC | 10 |
| Eeprom | 1K |
| Canales PWM | 6 |
| Voltaje de operacion | 1.8-5.5 v |
| Timers | 3 |

TABLA 7. PARÁMETROS ATMEGA 328.

Como puede verse, con pequeñas adaptaciones hardware puede incluirse este microcontrolador en el proyecto y así obtener un ahorro al no ser necesario comprar más placas en caso de que el prototipo sea funcional. Es decir, una vez que ha quedado demostrado en los capítulos anteriores que el sistema realiza su cometido, ya no sea necesario emplear una placa Arduino en cada nuevo sistema implementado, ya que será suficiente con la circuitería que se mostrará en los siguientes apartados.

6.2 DISPLAY LCD.

Se ha considerado interesante el hecho de incluir en el prototipo un display que permita visualizar la información in situ. Para ello, se recurre a un dispositivo como el que se muestra en la figura 38, donde se ve una imagen real junto a su esquema de pin out.

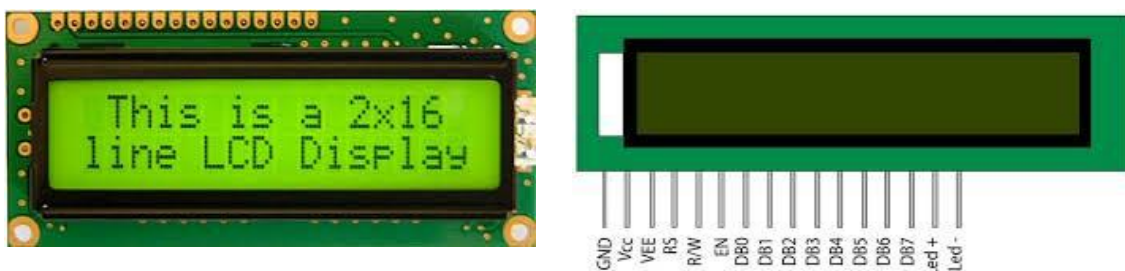


FIGURA 38. DISPLAY LCD 16X2 CARACTERES.

Mediante este elemento será posible conocer los parámetros que se están monitorizando y poder realizar pruebas de software sin tener que recurrir al monitor serie del IDE de Arduino, dado que mostrará mensajes de control.

6.3 ADAPTACIONES NECESARIAS.

Como se ha comentado, se requieren algunos componentes para que el microcontrolador pueda sustituir a la tarjeta Arduino en este proyecto, siendo estos elementos los indispensables para el correcto funcionamiento. Por otra parte, en cuanto al software, se puede decir que es tan sencillo como emplear Arduino para todas las operaciones de programación. Por tanto, se pueden enumerar en los siguientes elementos hardware:

- Fuente de alimentación a 5 V_{DC}.
- Cristal de 16 MHz.
- Condensadores de 22 pF.

Con estos pocos componentes se puede comprobar que el microcontrolador queda completamente operativo.

Por otro lado, respecto al LCD, simplemente será necesario tener en cuenta cómo se conectan sus pines y, en su caso, la inclusión de un potenciómetro de 10 K Ω para poder regular el contraste de iluminación de la pantalla. Es posible que la no inclusión de este dispositivo dificulte la visibilidad de los caracteres en la pantalla. Además, para poder controlar el LCD se debe incluir la librería "LiquidCrystal.h", lo que facilitará en gran medida el trabajo de programación.

Aparte, no se requiere ningún lenguaje de programación especial, dado que el microcontrolador se puede comprar con un bootloader precargado que permite programarlo con código Arduino como si fuera una tarjeta Arduino. En caso de que no venga con bootloader, también es posible programarlo mediante el IDE de Arduino. De este modo, en ambos casos, simplemente se emplea la tarjeta Arduino como interface de programación, de forma que una vez que el software esté cargado se puede integrar el micro en el nuevo circuito. Es decir, se emplea la tarjeta Arduino como tarjeta programadora, donde el chip a programar se inserta en el zócalo y una vez grabado el programa se pasa el chip al zócalo del circuito destino.

Finalmente, en la figura 39 se muestra un esquema realizado en Proteus de los elementos necesarios para la simulación de un led intermitente (simula el sketch básico de Arduino, el

famoso Blink). Así, puede verse como se conectan los componentes que se han mencionado, teniendo en cuenta que el pin out del microcontrolador no tiene nada que ver con los pines de la tarjeta Arduino, porque la numeración es completamente distinta. En el siguiente apartado, se plantea el circuito que comprende todo lo realizado hasta este punto; pero con el ATmega 328 en lugar de la tarjeta Arduino y con el display LCD. Es decir, se verá el diseño definitivo que será la alternativa al mostrado en el capítulo anterior.

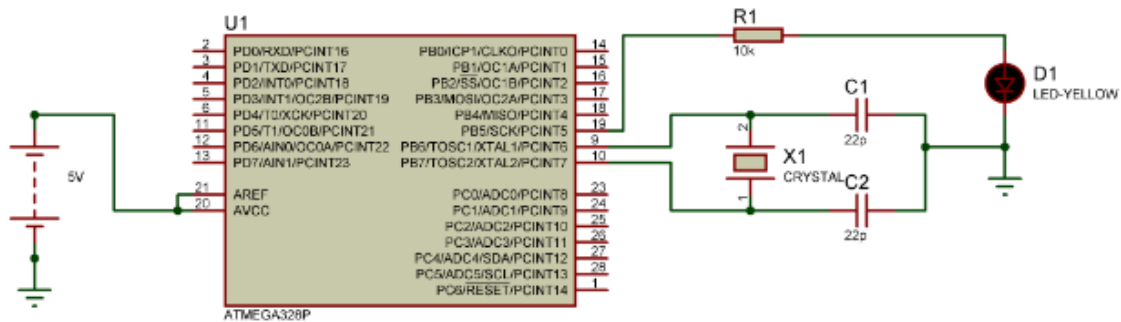


FIGURA 39. ESQUEMA LED INTERMITENTE CON ATMEGA 328.

6.4 IMPLEMENTACIÓN DEL CIRCUITO.

En primer lugar, se requiere tener presente lo expuesto anteriormente acerca del microcontrolador y su pin out, de modo que hay que ser conscientes de la relación existente entre los pines del micro y los pines de Arduino (ver figura 37). Una vez que este tema está claro, se deben considerar los siguientes factores:

- Se puede alimentar el micro mediante la fuente de alimentación que se diseñó en el capítulo 4; pero se debe sustituir el regulador de nueve voltios que alimentaba la placa por uno de cinco voltios. Entonces, se puede emplear un único regulador 78T05 que sea capaz de suministrar hasta tres amperios, de manera que se simplifica bastante el circuito.
- Se incluye en el circuito un cristal de 16 MHz junto con sendos condensadores de 22 pF, para proporcionar la señal de reloj al microcontrolador. La finalidad de los condensadores es dar mayor estabilidad a la señal de reloj.
- Las salidas del microcontrolador pueden alimentar directamente el módulo de relés, así no será necesario ningún cambio en este sentido.
- Se incluye un potenciómetro de 10 KΩ, para regular la retroiluminación del display LCD.
- La programación del micro se debe realizar empleando la tarjeta Arduino, de tal forma que se inserta el micro en el zócalo de la tarjeta y una vez programado se

retira y se coloca en el zócalo del circuito definitivo. De esta forma, se pueden programar todos los microcontroladores que se requieran en prototipos futuros.

En segundo lugar, una vez finalizado el proceso de rediseño del circuito, cuyos resultados se presentan en la figuras 47, 48 y 49 en los anexos, el layout definitivo es como se ve en la figura 40. Considerando todas estas figuras y para su mejor comprensión, se pueden describir los siguientes elementos:

a) Conector 1:

- pines 1 y 2 para entrada de tensión procedente del secundario del transformador.
- pines 3, 4 y 5 correspondientes a la entrada de datos, VDC y GND del sensor ambiental
- pines 6, 7 y 8 correspondientes a entrada de datos, VDC y GND del sensor de humedad del suelo
- pines 9 y 10 correspondientes a las salidas de activación de los relés de riego y del sistema antiheladas (módulo de relés).

b) Conector GND, para proporcionar tierra a cualquier elemento exterior, independientemente de los anteriores, en caso necesario.

c) Conector VDC-5V, para proporcionar alimentación al módulo de relés y a otros dispositivos, en caso necesario.

d) Conector módulo de comunicaciones: sirve de nexo entre el microcontrolador y la tarjeta SIM900, conectando los pines correspondientes entre ambos dispositivos.

e) Microcontrolador:

- pines 21 y 22 alimentados con VDC.
- pines 9 y 10 conectados al cristal de 16 MHz, con sus respectivos condensadores.
- pines 23 y 24 entradas de datos de los sensores ambiental y de suelo.
- pines 17, 18 y 19 salida de testigo de funcionamiento, al relé del sistema antiheladas y al relé del sistema de riego a través del conector.

f) LCD: pines D4-D7 a los pines 11, 6, 5 y 4 del micro, para los datos. E (Enable) al pin 12, habilita el display, RS al pin 16 y RW a GND. Los pines de alimentación a 5 V y GND respectivamente.

g) Fuente de alimentación: ha sido modificada de modo que solo incluye un regulador 78T05 con capacidad de suministrar 3 A y 5 V_{DC}. También se observan el varistor para proteger contra sobretensiones y el fusible para proteger contra cortocircuitos.

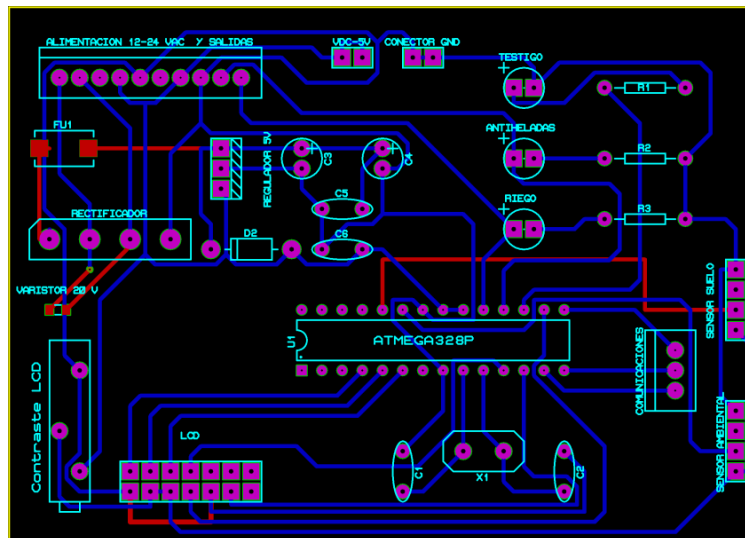


FIGURA 40. LAYOUT DEL CIRCUITO CON ATMEGA 328 Y LCD.

Finalmente, en la figura 41 se muestra un prototipo para pruebas en el cual se ha incluido el display LCD para verificar el correcto funcionamiento del sistema completo. En las figuras 50, 51 y 52 de los anexos se pueden ver algunas simulaciones, donde se han enviado ordenes desde un terminal móvil y el display muestra información sobre el evento.

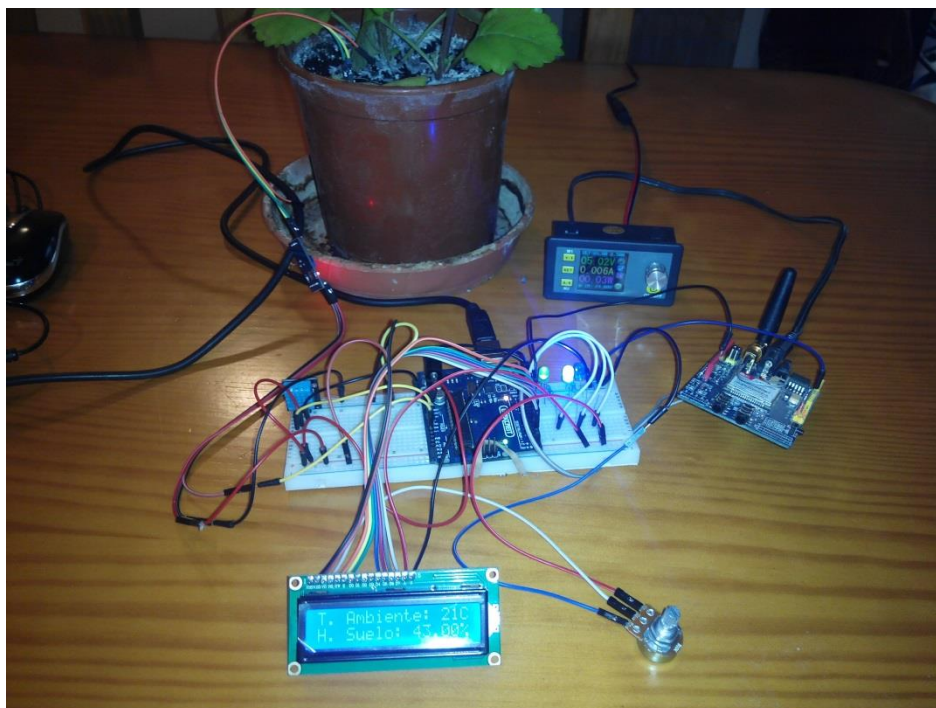


FIGURA 41. VERIFICACIÓN DEL PROTOTIPO DEFINITIVO.

6.5 RESUMEN DEL CAPÍTULO.

A lo largo de este capítulo se ha tratado de plantear de forma concisa el desarrollo requerido para optimizar la fabricación en serie del prototipo de automatización. Esto es, sustituir la tarjeta Arduino por un microcontrolador ATmega 328 y los elementos necesarios para su funcionamiento. Así, ha quedado patente que Arduino no es más que un buen entorno de aprendizaje y desarrollo de soluciones electrónicas, basado en este tipo de microcontroladores y con un conjunto de sistemas que aportan más funcionalidades a estas placas (regulador de alimentación, comunicaciones, programación, etc.).

En primer lugar, se ha realizado una descripción del microcontrolador, donde se ha puesto de manifiesto que, dado que es el núcleo de la tarjeta Arduino, bien puede sustituirse dicha tarjeta por este microcontrolador. Así mismo, se han enumerado las características del micro quedando suficientemente justificada la solución planteada.

En segundo lugar, se han descrito los elementos que requiere el microcontrolador para poder operar. Además, se ha incluido un display LCD en el prototipo con la finalidad de visualizar información in situ, comentándose los requisitos necesarios para su integración en el sistema.

Finalmente, se ha presentado el diseño del circuito de automatización, donde las diferentes imágenes que muestran los circuitos vistos en su esquemático, layout y 3D, dan una idea clara de cómo será el prototipo definitivo. Además, se han mencionado las características de la placa, mediante la descripción de algunos componentes y las conexiones con el exterior (sensores, alimentación, módulo de relés y comunicaciones). Incluso, se ha presentado una imagen de un montaje experimental del sistema completo que ha permitido verificar el correcto funcionamiento del mismo.

CONCLUSIONES Y LINEAS FUTURAS DE TRABAJO.

Una vez finalizado el presente trabajo, es momento de realizar una reflexión relativa a todo el proceso y poder obtener una serie de conclusiones que permitan asentar conocimientos y procedimientos. De este modo, se pueden enunciar las siguientes:

1. Se ha logrado alcanzar el objetivo fundamental del proyecto: diseño de un sistema de monitorización, automatización y gestión remota. Queda demostrado que el sistema diseñado cumple con los requerimientos planteados en los objetivos.
2. Se ha demostrado con todo tipo de montaje de prototipos, simulaciones, capturas de pantalla y fotografías que el proceso de diseño es correcto y funcional. Esto conlleva una extensión de la memoria superior a lo necesario, dado que se ha tratado de demostrar la operatividad de cada bloque que se diseñaba.
3. Se han alcanzado los hitos según lo planteado en la planificación, sin surgir imprevistos con gran impacto. En este punto hay que comentar que la parte de comunicaciones ha requerido un esfuerzo mayor al esperado, debido a su dificultad.
4. Se ha alcanzado un conocimiento inicial de software y hardware bastante variado, que de no ser por la realización del presente proyecto no habría sido posible. Por tanto, esta es una de las conclusiones más importantes que se pueden transmitir, ya que implica conocer nuevas herramientas de diseño y prototipado, para obtener cualquier tipo de solución electrónica.
5. Se ha conseguido poner en práctica competencias comunicativas y la redacción de textos técnicos, sin duda componentes muy importantes en el ámbito de la Ingeniería.
6. Se ha manifestado la exigencia de una buena planificación inicial, de forma que esta sea consecuente con la dificultad del mismo, con la finalidad de evitar desvíos temporales importantes por falta de previsión.

Aparte de lo anterior, se considera necesario comentar que ha sido un proceso muy enriquecedor y muy entretenido. Esto es así, dado que me ha permitido aprender y practicar con el entorno Arduino, algo inicialmente desconocido para mí y que presenta una potencia inimaginable. Quizás la posibilidad de poder montar los circuitos, desarrollar el código y realizar las simulaciones es lo que hace que todo este proceso haya sido ameno; pero también hay que decir que es muy exigente todo el proceso de búsqueda de información y aprendizaje. Con todo, considero que realizar este proyecto en concreto ha sido una muy grata experiencia.

Por otro lado, considerando ampliaciones o futuras líneas de trabajo se pueden enumerar las siguientes:

1. A nivel hardware, posiblemente lo más interesante sería realizar una nueva optimización consistente en incluir el chip de comunicaciones y los componentes necesarios dentro del circuito definitivo. Es decir, no tener que emplear el módulo de comunicaciones, sino incluir sus elementos imprescindibles en el circuito definitivo.
2. Una opción alternativa a la anterior podría ser explorar la posibilidad de emplear la tarjeta SIM900 como núcleo del sistema. Esto consistiría en obviar la tarjeta Arduino y todas las entradas y salidas conectarlas a dicho módulo. De este modo, solo sería necesario de la fuente de alimentación dado que el sketch de control podría ser cargado en el chip del módulo. Sin duda, esto requiere un análisis más a fondo del módulo de comunicaciones, para poder conocer si es viable esta alternativa.
3. A nivel de comunicaciones, se podría evolucionar hacia comunicaciones de banda ancha (3G y posteriores, empleando otro tipo de módulos) y mediante IP enviar los parámetros medidos a un servidor. Así, sería posible la creación de bases de datos para su posterior análisis, creación de gráficos, consultas... En definitiva, integrar el sistema en el Internet de las cosas (IoT).
4. En cuanto a software, en el prototipo se podrían incluir una serie de números de terminales móviles de forma que no sea un único usuario el que pueda interactuar con el sistema. Incluso, teniendo en cuenta lo expuesto en el anterior punto, se podría considerar la opción de acceder al sistema desde Internet mediante usuarios registrados y claves de acceso.
5. Monitorizar otros parámetros con alto grado de impacto (presencia de enfermedades), dependiendo de la posibilidad de contar con sensores adecuados.

En resumen, se pueden considerar diversas mejoras que requieren más tiempo y dedicación del disponible para el presente trabajo. Es más, se considera suficiente todo lo desarrollado en el presente proyecto, porque se han puesto en de manifiesto técnicas de programación, diseño y simulaciones con software y montaje de prototipos y simulaciones reales. Aun así, no es descartable implementar alguna de estas mejoras como reto personal. Con todo y como comentario personal, considero que ha sido un proyecto muy ameno y enriquecedor del que estoy bastante satisfecho (es una pena no haber podido ir más allá).

BIBLIOGRAFÍA.

- [1]. Medir temperatura y humedad con Arduino y sensor DHT11-DHT22. [Sede web]. Luis Llamas. [Acceso: 16/03/17]. Disponible en: <https://www.luisllamas.es/arduino-dht11-dht22/>
- [2]. Sensor de humedad del suelo YL38 y YL69. [Sede web]. [Acceso: 16/03/2017]. Disponible en: <https://www.taloselectronics.com/producto/sensor-de-humedad-del-suelo/>
- [3]. La Ribera del Duero utiliza tecnología de vanguardia para lograr mejores vinos. [Sede web]. [Acceso: 17/03/2017]. Disponible en: <http://www.20minutos.es/noticia/419178/0/torres/antihelada/burgos/>
- [4] Arduino. [Sede web]. [Acceso: 15/03/17]. Disponible en: <https://es.wikipedia.org/wiki/Arduino>
- [5] Entornos de Aplicación Arduino. [Sede web]. [Acceso: 15/03/17]. Disponible en: <https://aprendiendoarduino.wordpress.com/2016/06/26/entornos-de-aplicacion-arduino/>
- [6]. Análisis comparativo de las placas Arduino (oficiales y compatibles). Isaac P E. [Sede web]. [Acceso: 15/03/2017]. Disponible en: <http://comohacer.eu/analisis-comparativo-placas-arduino-oficiales-compatibles/>
- [7]. Comunidad Arduino. [Sede web]. [Acceso: 15/03/17]. Disponible en: <https://aprendiendoarduino.wordpress.com/2016/03/31/comunidad-arduino/>
- [8]. ARDUINO UNO R3. [Sede web]. [Acceso: 16/03/2017]. Disponible en: <http://arduino.cl/arduino-uno/>
- [9] Arduino y solo Arduino. Módulo Relé 5V DC - Keyes KY019 - Parte 1. [Sede web]. [Acceso: 19/03/2017]. Disponible en: <https://soloarduino.blogspot.com.es/2014/01/modulo-keyes-rele-sr1y.html>
- [10] Atmega328 [Sede web]. [Acceso 3/4/2017]. Disponible en: <https://es.wikipedia.org/wiki/Atmega328>
- [11]. Atmel Corporation. (2009). ATmega 328P [archivo PDF]. [Acceso 04/04/2017] Recuperado de: http://www.atmel.com/Images/Atmel-42735-8-bit-AVR-Microcontroller-ATmega328-328P_Datasheet.pdf

- [12] Estándar GSM (Sistema global de comunicaciones móviles). [Sede web]. [Acceso 21/04/17]. Disponible en: <http://es.ccm.net/contents/681-estandar-gsm-sistema-global-de-comunicaciones-moviles>
- [13] Redes de gran alcance sin hilos. Antonio Satué Villar. Barcelona: FUOC; 2012.
- [14] Estándar GPRS (Servicio general de paquetes de radio). [Sede web]. [Acceso: 22/04/2017]. Disponible en: <http://es.ccm.net/contents/680-estandar-gprs-servicio-general-de-paquetes-de-radio>
- [15] SIM900_Hardware Design_V2.00. [PDF]. Disponible en: ftp://imall.iteadstudio.com/IM120417009_IComSat/DOC_SIM900_Hardware_Design_V2.00.pdf
- [16] SIM900_AT CommandManual_V1.03 [PDF]. Disponible en: <https://es.scribd.com/document/147490424/SIM900-at-Command-Manual-V1-03-Desbloqueado>
- [17] Conjunto de comandos Hayes. [Sede web]. [Acceso: 23/04/17]. Disponible en: https://es.wikipedia.org/wiki/Conjunto_de_comandos_Hayes