



**Universidad de Valladolid**



**ESCUELA DE INGENIERÍAS  
INDUSTRIALES**

**UNIVERSIDAD DE VALLADOLID**

**ESCUELA DE INGENIERIAS INDUSTRIALES**

**Grado en Ingeniería Electrónica Industrial y Automática**

**Gestión domótica de una casa unifamiliar  
basada en Arduino**

**Autor:**

**Rodríguez Diego, Miguel**

**Tutor:**

**Quintano Pastor, Carmen  
Departamento de Tecnología  
Electrónica**

**Valladolid, Septiembre 2017.**



## Resumen

La realización de este proyecto busca la gestión domótica de una casa unifamiliar basada en Arduino, para ello se estudiarán las distintas características de los sistemas domóticos, su manejo mediante controladores, sensores y actuadores y la comunicación entre estos. Además se desarrollarán los sistemas domóticos para una casa y se profundizará en el control e integración de todos los sistemas mediante Arduino, buscando una gestión eficaz y que se pueda implementar físicamente.

Palabras clave: arduino, domótica, sensores, automatización y gestión.



## Índice General

Introducción y objetivos .....	11
Objetivos .....	12
Introducción teórica .....	13
Capítulo 1. Introducción a la domótica.....	13
1. Características de las instalaciones domóticas.....	13
Gestión energética .....	13
Comunicaciones.....	14
Seguridad.....	14
Confort .....	15
2. Tipos de sistemas domóticos.....	16
Clasificación según la tipología.....	16
Clasificación según la topología.....	18
Clasificación según el método de transmisión .....	20
Capítulo 2. Sensores, actuadores y visualizador de datos en una vivienda domótica .....	21
1. Sensores .....	21
2. Actuadores.....	25
3. Visualizador de datos.....	27
Capítulo 3. Controlador de la vivienda .....	30
1. Microcontrolador Arduino .....	31
2. Sistemas de comunicación de Arduino .....	33
2.1. Comunicación puerto serie.....	33
2.2. I <sup>2</sup> C .....	33
2.3. Ethernet.....	35
Capítulo 4. Android.....	37
1. Características del sistema operativo Android .....	37
2. Arquitectura Android .....	38
Desarrollo del trabajo de fin de grado .....	41
Capítulo 5. Diseño de una vivienda domótica .....	41
1. Definición y planos de la vivienda.....	41
2. Sistemas domóticos que implantar en la vivienda.....	42
2.1. Sistema de calefacción.....	42

2.2.	Sistema de persianas .....	45
2.3.	Sistema de seguridad .....	46
2.4.	Puerta del garaje.....	48
2.5.	Pantalla táctil .....	49
Capítulo 6. Gestión domótica de una casa unifamiliar basada en Arduino		52
1.	Elección del microcontrolador .....	52
2.	Distribución de los sistemas domóticos en los microcontroladores	56
2.1.	Sistemas domóticos controlados por el Arduino Yún.....	56
2.2.	Sistemas domóticos controlados por el Arduino Mega.....	57
3.	Gestión de la casa domótica .....	59
4.	Programación de los microcontroladores y la pantalla táctil .....	61
4.1.	Comunicación entre Arduinos.....	61
4.1.1.	Trasmisión de información maestro-esclavo .....	62
4.1.2.	Demanda de información del maestro al esclavo.....	62
4.1.3.	Respuesta del esclavo a la demanda del maestro .....	62
4.1.4.	Recepción de la información en el dispositivo esclavo.....	63
4.2.	Comunicación entre el Arduino Yún y la pantalla Nextion .....	63
4.2.1.	Configuración e inicialización de la comunicación.....	63
4.2.2.	Instrucciones de comunicación con la pantalla .....	64
4.3.	Comunicación con el lector RFID.....	65
4.4.	Programación de la pantalla .....	65
4.5.	Programación general del Arduino Yún.....	66
4.6.	Programación general del Arduino Mega.....	67
4.7.	Programación de los motores, interrupciones y sensores de temperatura y luminosidad.....	67
4.7.1.	Motores.....	67
4.7.2.	Sensores de temperatura y luminosidad .....	68
4.7.3.	Interrupciones .....	69
4.8.	Librerías usadas para los programas.....	70
5.	Conexión hardware con los Arduinos .....	71
5.1.	Conexión entre los dos Arduinos .....	71
5.2.	Conexiones del Arduino Yún.....	71
5.2.1.	Conexión de la puerta del garaje.....	71

5.2.2. Conexión de la pantalla táctil .....	73
5.2.3. Conexión con los zumbadores activos.....	74
5.2.4. Conexión de las LDRs.....	75
5.3. Conexiones del Arduino Mega .....	75
5.3.1. Conexión de los sensores de movimiento .....	76
5.3.2. Conexión de los sensores de temperatura .....	76
5.3.3. Conexión de las electroválvulas. ....	77
Conclusiones y extensiones .....	79
Bibliografía.....	83
Anexos.....	85
Anexo I. Código Arduino Yún.....	85
Anexo II. Código Arduino Mega.....	90
Anexo III. Planos de la vivienda .....	94
Plano 1. Garaje.....	95
Plano 2. Planta principal.....	96
Plano 3. Planta alta.....	97
Plano 4. Sensores garaje.....	98
Plano 5. Sensores y persianas planta principal.....	99
Plano 6. Sensores y persianas planta alta .....	100

## Índice de figuras

Figura 1. Características de la domótica. ....	11
Figura 2. Sistema domótico centralizado .....	16
Figura 3. Sistema domótico descentralizado. ....	17
Figura 4. Sistema domótico distribuido o mixto.....	17
Figura 5. Topología en estrella. ....	18
Figura 6. Topología en anillo.....	18
Figura 7. Topología bus.....	18
Figura 8. Topología en malla. ....	19
Figura 9. Topología en árbol. ....	19
Figura 10. Topología mixta.....	19
Figura 11. Comparación señal analógica (Azul) y digital (Verde).....	22
Figura 12. Relé .....	25
Figura 13. Display de siete segmentos.....	27
Figura 14. Tres Displays de siete segmentos unidos. ....	27
Figura 15. Display LCD.....	28
Figura 16. Pantalla LCD Graficadora. ....	28
Figura 17. Pantalla LCD tft.....	28
Figura 18. Pantalla táctil.....	29
Figura 19. Transmisión UART. ....	33
Figura 20. Trama de datos UART.....	33
Figura 21. Esquema comunicación I <sup>2</sup> C.....	34
Figura 22. Trama de datos I <sup>2</sup> C.....	34
Figura 23. Trama de datos Ethernet. ....	35
Figura 24. Arquitectura Android. ....	38
Figura 25. Suelo radiante por agua caliente. ....	43
Figura 26. Encapsulado TMP36. ....	43
Figura 27. Voltaje de salida vs. Temperatura.....	44
Figura 28. Electroválvula desmontable Orkli.....	44
Figura 29. Motor tubular en el eje de la persiana.....	45
Figura 30. LDR.....	46
Figura 31. Sensor HC-SR501 PIR.....	47
Figura 32. Esquema de funcionamiento de un sensor PIR.....	47
Figura 33. Buzzer activo YL-44.....	48
Figura 34. Motor puerta garaje. ....	49
Figura 35. Receptor RC522.....	49
Figura 36. Nextion Editor .....	50
Figura 37. Pantalla Nextion NX4024T032_11.....	51
Figura 38. Inicialización de la comunicación en el Arduino Yún. ....	61
Figura 39. Inicialización de la comunicación en el Arduino Mega.....	61
Figura 40. Transmisión de información Arduino Yún – Mega.....	62
Figura 41. Demanda y captura de información del dispositivo maestro.....	62



Figura 42. Función receiveRequest del Arduino Mega. ....	62
Figura 43. Recepción de información en el Arduino Mega. ....	63
Figura 44. Configuración para la comunicación del Arduino Yún con la pantalla. .....	64
Figura 45. Inicialización de la comunicación con la pantalla. ....	64
Figura 46. Función para recibir información de la pantalla. ....	64
Figura 47. Función para cambiar valores de la pantalla.....	64
Figura 48. Función para enviar comandos a la pantalla.....	64
Figura 49 . Configuración para la comunicación con el lector RFID. ....	65
Figura 50. Función de inicio de comunicación con el lector RFID.....	65
Figura 51. Funciones de detección y lectura de una etiqueta.....	65
Figura 52. Programa Nextion Editor con zonas numeradas. ....	66
Figura 53. Loop del Arduino Yún.....	66
Figura 54. Loop del Arduino Mega.....	67
Figura 55. Programación del movimiento de la puerta del garaje. ....	68
Figura 56. Lectura de un sensor de temperatura TMP36. ....	68
Figura 57. Programa para utilizar la luminosidad. ....	69
Figura 58. Interrupción del Arduino Yún.....	69
Figura 59. Funciones para los Timers. ....	69
Figura 60. Librerías utilizadas.....	70
Figura 61. Esquema conexión entre Arduinos. ....	71
Figura 62. Conexión puerta del garaje. ....	72
Figura 63. Esquema de la conexión de los relés .....	72
Figura 64. Conexión a la pantalla táctil.....	73
Figura 65. Conexión con los buzzers activos. ....	74
Figura 66. Conexión de las fotorresistencias.....	75
Figura 67. Conexión a un sensor de HC-SR501. ....	76
Figura 68. Conexión de un sensor del sensor TMP36. ....	76
Figura 69. Conexión de la electroválvula. ....	77
Figura 70. Pantalla principal aplicación Android ControlDomotico. ....	80
Figura 71. Pantalla de navegación aplicación Android ControlDomotico. ....	81
Figura 72. Pantallas de alarma, persianas, calefacción y garaje (de izquierda a derecha) de la aplicación Android ControlDomotico. ....	81

## Índice de tablas

Tabla 1. Variantes de Ethernet.....	36
Tabla 2. Necesidades de entradas/salidas del microcontrolador.....	53
Tabla 3. Comparación características microcontroladores.....	54
Tabla 4. E/S microcontrolador Arduino Yún. ....	57
Tabla 5. E/S microcontrolador Arduino Mega. ....	58

## Introducción y objetivos

La sociedad vive hoy una era de despliegue tecnológico que inunda todos los aspectos de nuestra vida cotidiana, tareas como la limpieza, hacer la comida, comprar, vender, trabajar, comunicarnos... dependen en una medida más o menos grande de la tecnología de la que disponemos. La tecnología ha sido desarrollada con el fin de mejorar la calidad de vida del ser humano y facilitarnos lo que a nuestros antepasados les costó mucho más esfuerzo, de esta manera, por ejemplo, la aparición de tractores redujo mucho el esfuerzo que hacían los agricultores en las labores de la tierra o los sistemas de mensajería y redes sociales permiten tener una gran variedad de recursos para comunicarse e interrelacionarse.

La tecnología aporta también al ser humano información y un cierto grado de control, que infunden seguridad, por ejemplo, saber si mañana va a llover o no, o que carreteras están atascadas. Dentro de este progreso tecnológico en la unión de los campos seguridad, información, control, confort y comunicación, podemos situar la tecnología dentro del hogar para lograr controlar nuestra casa, con una gestión eficiente y de manera robusta, en la que la seguridad es un factor principal. Así nace el término domótica.

Según la asociación española de domótica e inmótica, “La domótica es el conjunto de tecnologías aplicadas al control y la automatización inteligente de la vivienda, que permite una gestión eficiente del uso de la energía, que aporta seguridad y confort, además de comunicación entre el usuario y el sistema” (CEDOM, 2017). La domótica es por lo tanto un sistema que permite controlar y gestionar de manera eficiente nuestro hogar, de manera que aumente nuestro confort y seguridad facilitando nuestra vida. También se puede ver la domótica como la confluencia de varias tecnologías aplicadas a una vivienda como representa la Figura 1.

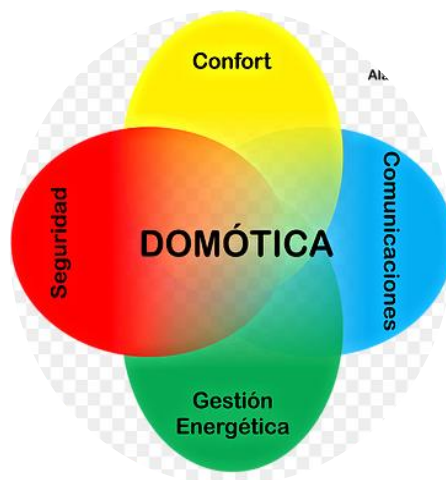


Figura 1. Características de la domótica.

El avance de la tecnología ha propiciado la introducción de la domótica en las casas, hoy en día los sistemas domóticos son relativamente baratos, por lo menos comparado con los de hace algunos años. La domótica ofrece gran versatilidad y gradualidad habiendo viviendas poco automatizadas o con pocos sensores y actuadores, y viviendas muy automatizadas. Otra ventaja de la domótica es la modularidad, es decir, se puede dividir el sistema domótico en varios subsistemas como la climatización, la seguridad, etc.

Cuando se habla domótica es importante definir términos como inmótica o edificio inteligente. La inmótica alude a la automatización de industrias para lograr confort, seguridad, una gestión eficiente de la energía y garantizar la comunicación entre el usuario y el sistema. El término edificio inteligente es parecido al de domótica, pero es utilizado para nombrar grandes edificios automatizados, como oficinas, universidades, etc. (Rodríguez, 2014).

### Objetivos

El objetivo general de este trabajo de fin de grado es la gestión domótica de una casa unifamiliar que esté controlada por un microcontrolador Arduino. Esto se podría desglosar en los siguientes objetivos específicos:

- Control de la calefacción.
- Control de las persianas de la casa.
- Alarma anti-intrusiones.
- Apertura de la puerta del garaje.
- Control del sistema mediante una pantalla táctil con una interfaz de usuario sencilla.
- Posibilidad, en la interfaz de usuario, de programar la automatización de las persianas, la calefacción, diferentes modos de funcionamiento de la alarma...
- Preparación de la evolución del sistema para la futura integración de otros controles domóticos.

## Introducción teórica

### Capítulo 1. Introducción a la domótica

#### 1. Características de las instalaciones domóticas

##### *Gestión energética*

Una de las principales características que ofrece un sistema domótico es la gestión energética, es decir, un uso eficiente de los recursos para un rendimiento y coste óptimos. Para una gestión energética eficiente lo primero que hay que hacer es estudiar y analizar la energía que se tiene y el uso que se va a hacer de ella, para ver donde se pueden reducir costes. Un estudio principal en la gestión energética es el estudio de las tarifas para aprovechar las más baratas.

Podemos diferenciar varios aspectos en la gestión de la energía:

- Control de la temperatura o climatización. Conseguir la temperatura deseada dentro de la vivienda es uno de los mayores gastos tanto en invierno con la calefacción, como en verano con el aire acondicionado, por eso un uso eficiente de la calefacción y el aire acondicionado ayudan a ahorrar. Para ello se disponen de diferentes herramientas:
  - Zonificación: consiste en la separación en zonas independientes de la casa, de manera que cada una pueda tener una temperatura diferente, o pueda estar encendida en unas zonas la calefacción mientras que en otras no. De esta manera se puede regular la temperatura de cada zona a través de un sensor de temperatura y el control sobre la climatización de esa zona permite por ejemplo en invierno poner mayor temperatura en sitios como el salón y menos en las habitaciones, también proporciona ahorro en la energía si hay zonas de la casa que no se utilizan mucho.
  - Programación: la opción de programación permite un ahorro energético porque se puede programar el uso de la climatización solo cuando sea necesario, por ejemplo, que se encienda la calefacción una hora antes de llegar a casa después de trabajar. También se pueden programar las diferentes zonas de la vivienda cada una con su temperatura. Además, se puede programar un valor por debajo de la cual no tiene que bajar la temperatura de una zona de la vivienda.
- Gestión de la electricidad: para la gestión de la electricidad se puede establecer el precio de cada tarifa según las horas del día que da la compañía eléctrica y utilizar los electrodomésticos que más consumen a las horas más económicas. Monitorizar el consumo de energía según las horas y la potencia puede ayudar a ver en qué se gasta más

electricidad y poder reducir el consumo y conseguir así un mayor ahorro (CEDOM, 2008).

- Control sobre el gasto de agua: el control sobre agua en una vivienda es muy importante para ello se puede introducir control sobre el riego, sensores de fugas de agua, reciclaje de aguas grises o grifería inteligente que regula el caudal y la temperatura (CEDOM, 2008).

### *Comunicaciones*

Otro aspecto esencial de las viviendas domotizadas es la comunicación. La comunicación en domótica va más allá de conocer el estado de la casa y de la automatización en tiempo real, supone el control de la casa y de los sistemas automatizados.

La comunicación por lo tanto debe realizarse a través de un canal seguro, para que la información en un sentido y en otro sea fiable, la comunicación habitualmente se suele realizar a través de internet, si se está fuera de la casa, o bluetooth si se está en el interior.

Habitualmente la comunicación se usa para dar mensajes de alarma o incidencias en el hogar, poder controlar algún sistema o ver los informes de actividad y consumo.

### *Seguridad*

La seguridad es un factor principal para todo ser humano por ello en la integración de la tecnología en el hogar, la seguridad es muy importante. Por este motivo se han desarrollado numerosas tecnologías que ayudan a que el hombre se sienta seguro tanto si está dentro como fuera de la vivienda. Entre los sistemas desarrollados para la seguridad destacan:

- Alarma anti-intrusiones. Este tipo de alarma sirve para defender y avisar en caso de que una persona ajena a la propiedad entre dentro de ella. Este tipo de seguridad está basado en la detección de personas o animales mediante sensores, ya sea de movimiento o de apertura no deseada de puertas y ventanas. Además, como prevención para posibles ladrones que merodeen se puede instalar un sistema que simule que hay alguien en la vivienda, es decir que se suban las persianas solas o se enciendan diferentes luces, por ejemplo.
- Sistemas de videovigilancia. Los sistemas de videovigilancia sirven para grabar y guardar las imágenes de lo que está sucediendo en el interior de la casa. Se pueden dividir en dos grandes subsistemas:
  - En el primero se puede ver lo que se está vigilando en una pantalla, ya sea porque la cámara o cámaras de videovigilancia están conectadas directamente a una pantalla o porque puedes realizar una conexión remota a través de internet.

- En este segundo subsistema las imágenes grabadas son guardadas, en un disco duro o una unidad de almacenamiento, y después, si es necesario, son revisadas.
- Sistemas de detección de problemas técnicos. Están diseñados para detectar una gran variedad de problemas niveles altos de gases tóxicos, incendios, fugas de agua, etc. Además de detectarlos también pueden actuar, por ejemplo, si se detecta una inundación puede cerrar la entrada de agua en la vivienda, o activar un sistema antincendios.

### *Confort*

Es la parte principal de la domótica, facilitar la vida y proporcionar mayor comodidad a los que residen en la vivienda, para ello se implementan diferentes sistemas:

- Control sobre la iluminación. En una casa domótica se puede instalar un sistema que permita controlar la iluminación de la casa, desde el control a distancia hasta la intensidad luminosa de cada una de ellas.
- Automatización de elementos básicos: se pueden automatizar las persianas de manera que cuando haya luz se suban para no tener que utilizar luz artificial o bajarse en verano para que no entre mucho calor; automatización de ventanas que estén programadas para ventilar la casa o que se cierren cuando llueven; toldos para que no le dé el sol a ciertas partes de la casa; también se pueden automatizar las puertas tanto exteriores, como la del garaje o la puerta principal, como todas las puertas de la casa; y automatización de las válvulas del gas y del agua (Rodríguez, 2014).
- Control de la vivienda a distancia mediante internet o cualquier otra forma de comunicación.
- Información sobre el estado de la casa y de los sistemas.

## 2. Tipos de sistemas domóticos

Se pueden clasificar los sistemas domóticos de distintas maneras atendiendo a la tipología, topología y medios de transmisión (Novel, 2013).

### *Clasificación según la tipología*

Según el tipo de tipología se pueden definir tres tipos de sistemas:

- Sistema centralizado. Los sistemas centralizados son aquellos en los que hay un único controlador principal que envía las señales a los actuadores y recibe la información de los sensores. Algunas de las ventajas son que es de fácil uso e instalación y el coste es menor respecto a otros tipos de sistemas. La desventaja es que necesita una interfaz de usuario, y que el correcto funcionamiento del sistema depende del controlador principal.

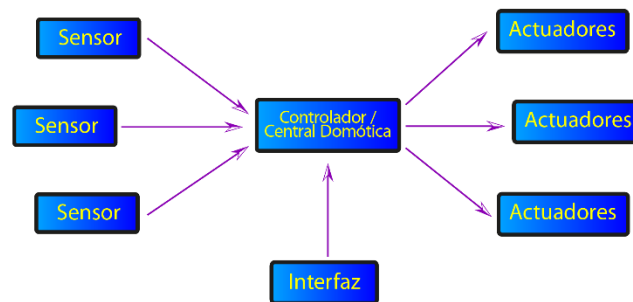


Figura 2. Sistema domótico centralizado

- Sistema descentralizado. Un sistema descentralizado se caracteriza porque los controladores del sistema están distribuidos en los actuadores y sensores, de manera que en vez de haber un único controlador central hay multitud de ellos (Rodríguez, 2014). Algunas de las ventajas de este sistema frente a los otros es que el sistema no depende de un único controlador por lo que es más fiable, además se puede ampliar fácilmente y es mucho más seguro en el funcionamiento. Como desventaja se podría tener que el coste es más elevado que el sistema anterior, además es más difícil de programar.



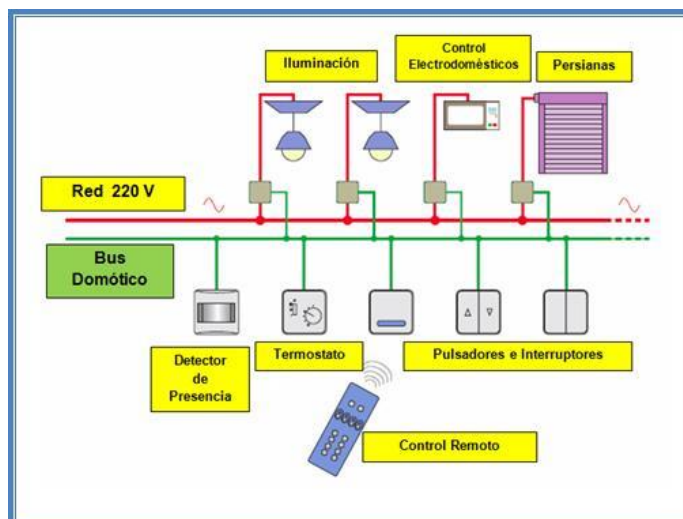


Figura 3. Sistema domótico descentralizado.

- Sistema domótico distribuido o mixto. Es una mezcla de los dos anteriores, donde puede haber uno o varios controladores principales y controladores en los sensores y actuadores como se puede ver en la Figura 4. Algunas de las ventajas de este sistema frente a otros son: moderado precio y cableado, fácilmente ampliable y se puede rediseñar la red. Como desventaja tenemos que requiere programación (Novel, 2013).

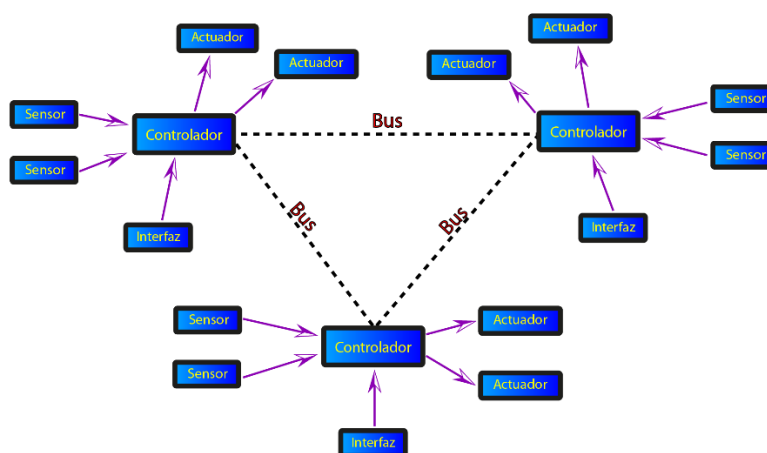


Figura 4. Sistema domótico distribuido o mixto

### Clasificación según la topología

La clasificación según la topología está referida al modo de conectar entre si los sensores, actuadores y controladores. Para sistemas domóticos podemos definir tres tipos de topologías principales.

- Topología en estrella. En esta topología toda la información pasaría por el controlador central y el la distribuiría según este programado (Figura 5). Este tipo de topología se suele utilizar para sistemas centralizados con un único controlador.

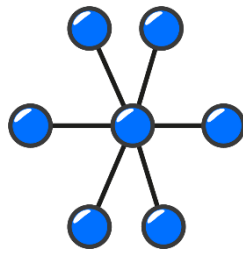


Figura 5. Topología en estrella.

- Topología en anillo. En esta configuración la información pasa de un dispositivo al siguiente hasta que se cierra el círculo de manera que cada dispositivo solo tiene comunicación directa con otros dos como se puede ver en Figura 6.

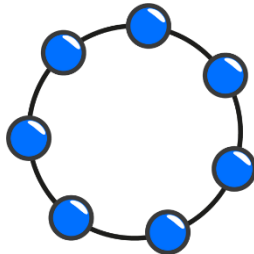


Figura 6. Topología en anillo

- Topología en bus. Para esta tecnología hay un cable central o bus al que se conectan todos los sensores, actuadores y controladores (Figura 7). Esta topología es la utilizada para sistemas domóticos descentralizados.

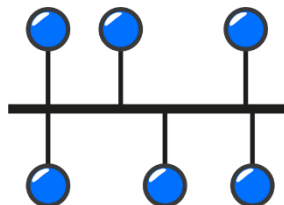


Figura 7. Topología bus.

- Topología en malla. En esta topología la información se intercambia entre dos puntos interconectados y si no hay conexión directa tiene que

ir desde un punto a otro hasta que llegue a su destino, como se puede ver en la Figura 8. Esta topología se puede utilizar para sistemas distribuidos.

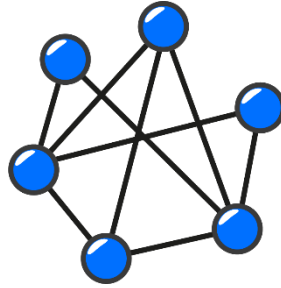


Figura 8. Topología en malla.

- Topología en árbol. Es una variante de la topología estrella, cada sensor o actuador se conecta a un controlador, pero no todos se conectan al controlador principal. Figura 9.

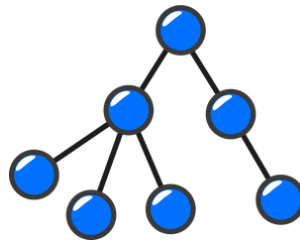


Figura 9. Topología en árbol.

- Topología mixta. La topología mixta es la combinación de una o más de las anteriores topologías. Figura 10.

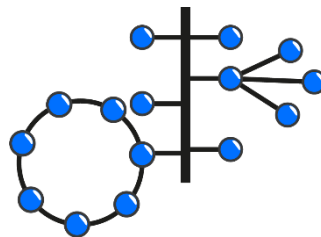


Figura 10. Topología mixta.

### *Clasificación según el método de transmisión*

Los sistemas domóticos también se pueden clasificar según el método con el que se transmite la información entre sensores, actuadores y controlador o controladores. Según el método de transmisión se puede clasificar en (Novel, 2013):

- Corrientes portadoras. La transmisión por corriente portadoras se realiza a través del cableado ya instalado de baja tensión.
- Transmisión cableada. Es mucho más fiable que la anterior, se utiliza corriente continua para transmitir la información.
- Transmisión por fibra óptica. La información de los sensores, actuadores y controladores es transmitida en de cables de fibra óptica a través de impulsos luminosos. La ventaja de este tipo de comunicación es la rapidez de la transmisión.
- Transmisión inalámbrica. Se caracteriza porque no hace falta cableado para la transmisión de la información, el controlador se comunica con los sensores y actuadores a través de wifi, bluetooth, radiofrecuencia...

## Capítulo 2. Sensores, actuadores y visualizador de datos en una vivienda domótica

Dos elementos vitales para el correcto funcionamiento de una vivienda domótica son los sensores y actuadores, que serán abordados en este capítulo. En domótica los sensores y actuadores son los que proporcionan la información del medio y la capacidad para poder actuar sobre algunas variables del entorno, todo ello es controlado y gestionado por el controlador, del que se hablará en el siguiente [capítulo](#).

### 1. Sensores

Un sensor es un dispositivo capaz de producir una señal traducible en función de la energía o magnitud física que quiere medir (Ruiz González, 2016). Es decir, un sensor es un objeto capaz de detectar y medir una magnitud física y ofrecer una salida que este relaciona con la medición. En los sensores podemos diferenciar dos partes importantes, la primera es la entrada que es la que está en contacto con la señal física que se quiere medir y la segunda es la salida que da la información sobre la variable medida, la señal de salida puede tener la misma magnitud física u otra distinta, por ejemplo, un sensor que mide la luminosidad puede tener una señal de salida que sea eléctrica o luminosa. En el caso de los sensores toman información de un proceso o del entorno físico y convierten la magnitud en una señal eléctrica (Escuela, ISA, (Ingeniería de Sistemas y Automática), 2007), para que la información llegue al controlador o pueda ser procesada en otro sitio. Cuando se habla de señales eléctricas en sensores podemos distinguir dos tipos:

- Señal analógica: la salida del sensor puede tomar cualquier valor dentro de un rango en función de la entrada.
- Señal digital: la salida solo puede tomar dos valores dentro de un rango en función de la entrada, estos dos valores se corresponden con el cero y el uno lógico.

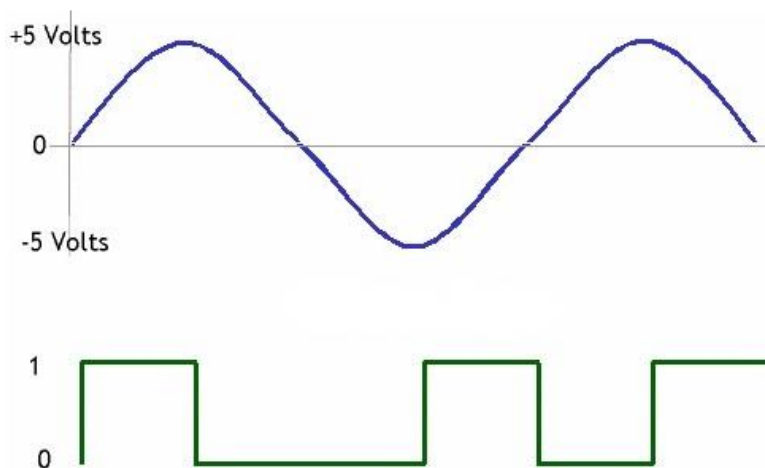


Figura 11. Comparación señal analógica (Azul) y digital (Verde)

Como se puede ver en la Figura 11 la señal analógica puede tomar todos los valores dentro del rango de menos cinco a más cinco voltios, por otro lado, la señal digital solo puede tomar o el valor cero o el uno y el cambio de un valor a otro es brusco.

Para el correcto funcionamiento de los sensores es necesario conocer sus características, de manera que se sepa si la medición va a ser correcta o no en las condiciones de funcionamiento. Las características más importantes de los sensores son:

- Exactitud: muestra cuanto se acerca la medida del sensor al valor real.
- Fidelidad: es otra característica que muestra la capacidad del sensor de que repetir la medida cuando la entrada es la misma.
- Reproducibilidad: es la coincidencia entre los resultados obtenidos cuando la entrada es la misma y las mediciones se realizan en un plazo largo de tiempo y por distintas personas.
- Sensibilidad: la sensibilidad en un sensor es la correspondencia entre el cambio en la salida y la variación en la entrada que lo ha provocado.
- Linealidad: es la característica que define como se ajustan las diferentes medidas de un sensor a una recta dada.

Los sensores se pueden clasificar en función de la energía que necesitan:

- Sensores activos: la energía de la señal de salida enteramente o en gran medida de una fuente de energía auxiliar.
- Sensores pasivos: la energía de la señal de salida depende de la propia medición.

Los sensores también se pueden clasificar en función del parámetro que varía con la medición de manera que los sensores pueden ser resistivos, capacitivos, reluctivos... (Ruiz González, 2016).

La última clasificación y más importante es en función de la medida que realizan o el objeto final de sensor. Tipos de sensores (Escuela, ISA, (Ingeniería de Sistemas y Automática), 2007):

- Sensores de fuego-humo, gas y CO. El sensor humo-fuego da alerta cuando hay una concentración alta de humo o una fuente de calor originada por un fuego. Los detectores de gas como su propio nombre indican miden el nivel de los gases en una habitación para detectar fugas o concentraciones que sean peligrosas para el ser humano. Los detectores de CO (monóxido de carbono) son un tipo de los detectores de gas, el CO es incoloro e inodoro muy tóxico para el ser humano, de ahí su importancia.
- Sensores de presencia. El sensor de presencia detecta cuerpos en movimiento. Este tipo de sensor es utilizado en múltiples tareas (Escuela, ISA, (Ingeniería de Sistemas y Automática), 2007):
  - Iluminación. Se puede utilizar para controlar la iluminación de pasillos, garajes, portales, pasillos de una casa...
  - Apertura y cierre de puertas.
  - Seguridad: como aviso de una intrusión.
  - Accionamiento de cualquier tipo de mecanismo.

Hay dos tipos de sensores de presencia:

- Volumétricos: sirven para el interior de la casa pueden colocarse en la pared, en el techo, empotrados...
- Perimétricos: son para el exterior de la vivienda.
- Sensores magnéticos y de rotura de cristal: Los sensores magnéticos se utilizan en las ventanas, persianas y puertas para detectar la apertura y cierre, este tipo de sensores se pueden utilizar en alarmas y control de la climatización de la casa; y los sensores de rotura de cristal se utilizan en zonas acristaladas para protección, hay diferentes sensores según la necesidad de estar adheridos al cristal o no y el mecanismo interno del sensor.
- Barreras de seguridad. Son sensores constituidos por un emisor y receptor de rayos infrarrojos (no visibles al ojo humano) que se colocan de forma vertical. Cuando un sólido opaco corta los infrarrojos el sensor lo detecta.
- Sensores de luminosidad. Detectan la luminosidad que hay en el entorno, es importante en este sensor una buena colocación para obtener una buena medición.

- Detectores de pánico y emergencias. Este tipo de sensores son muy importantes porque dan seguridad a las personas, podemos distinguir varios tipos de sensores: sensores de alarma médica y emergencias que se utilizan en caso de emergencia, como pueden ser las pulseras para ancianos que llevan un botón de alarma y detectan si se ha caído la persona que los lleva y manda una señal de emergencia a un móvil.
- Detectores de inundación. Detectan fugas de agua o inundaciones, pueden estar conectados a la llave general del agua y así poder cortar el suministro de agua en caso de que se esté inundando la vivienda.
- Sensores de temperatura. Mide la temperatura de la habitación donde está situado el sensor, son muy importantes para un uso eficiente de la energía tanto en climatización como en calefacción.
- Sensores de viento, lluvia, humedad. Este tipo de sensores son utilizados en el exterior de la vivienda.

El sensor de viento se basa en principio de funcionamiento del anemómetro, es decir unas paletas que giran con una velocidad proporcional al viento. Este tipo de sensores son utilizados sobre todo en toldos para que se recojan cuando hay mucho viento.

El sensor de lluvia detecta cuando empieza a llover o cuando han caído las suficientes gotas para que la lluvia sea importante, de esta manera se pueden cerrar puertas, ventanas y persianas para que no entre agua en el interior de la vivienda.

Los sensores de humedad pueden ser utilizados para los riegos, para medir cuando un jardín o un terreno necesitan ser regados, también pueden medir la humedad ambiente tanto en interior como en el exterior de la vivienda.

- Otros sensores. Hay infinidad de sensores más que pueden ser utilizados en la domótica de una vivienda, algunos de los más importantes son:
  - Vídeo cámaras que son utilizadas generalmente en la seguridad de la casa.
  - Lectores de los sistemas de identificación. Se utilizan para el acceso a la vivienda o a zonas de la vivienda, pueden ser claves por teclados, lectores de huella, escáner ocular, reconocimiento de voz, tarjetas magnéticas...
  - Receptores de infrarrojos, radiofrecuencia o bluetooth utilizados como mandos a distancia de diferentes accionamientos.

Además de todos los sensores ya mencionados hay una infinidad más de ellos con muchas aplicaciones cada uno, de manera que se pueden realizar gran variedad de montajes respondiendo a las necesidades de una vivienda domótica.



## 2. Actuadores

Un actuador es un dispositivo capaz de transformar un tipo de energía en otra, con el fin de modificar el entorno o una variable. Los actuadores son normalmente eléctricos, neumáticos o hidráulicos, en el caso de la domótica los más utilizados son los eléctricos porque no hacen mucho ruido y solo se necesita electricidad (Rodríguez, 2014).

En domótica existen gran cantidad de actuadores, más adelante se hablará de los más importantes, pero antes es necesario explicar que son los accionadores, pues lo llevan muchos actuadores. Un accionador aporta la energía necesaria para modificar la variable que se quiere controlar (Escuela, ISA, (Ingeniería de Sistemas y Automática), 2007), los accionadores más importantes son los relés y los contactores.

Un relé es un dispositivo eléctrico que consta de dos circuitos separados y aislados, con el que se pueden utilizar grandes potencias, con señales de control de poca potencia (Escuela, ISA, (Ingeniería de Sistemas y Automática), 2007). La primera parte del relé consta de una bobina que cuando pasa por ella una pequeña corriente se excita y atrae el inducido de hierro dulce que hace que se junten los contactos, que son la segunda parte del relé. Todo esto se puede ver en la Figura 12

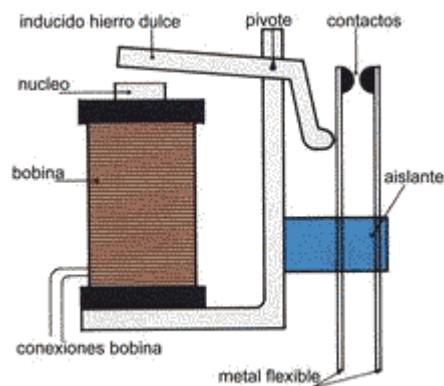


Figura 12. Relé

Los contactores son interruptores automáticos pensados para para corrientes y tensiones más elevadas que las de los relés (Escuela, ISA, (Ingeniería de Sistemas y Automática), 2007).

Tipos de actuadores (Escuela, ISA, (Ingeniería de Sistemas y Automática), 2007):

- Motores eléctricos: transforma la energía eléctrica en movimiento, ya sea rotatorio o lineal. El motor eléctrico es uno de los actuadores más

importantes dentro de la domótica, pues interviene en casi todos los sistemas domotizados de la vivienda y en otros actuadores.

- Persianas y toldos. Las persianas y toldos en una vivienda domótica están automatizadas y pueden ser controladas por el controlador de la vivienda, normalmente están construidas a partir de un motor tubular, también pueden contar con finales de carrera, control por radiofrecuencia o mando a distancia, además de las botoneras al lado de la persiana o toldo.
- Lámparas y luminarias. La iluminación es una parte esencial de la vivienda, tanto para el correcto desarrollo de la vida de las personas que en ella viven, como para un uso eficiente y cómodo. La iluminación es uno de los sistemas que más energía consume una vivienda, por ello la instalación de las luces debe ser adecuada a las necesidades y con la tecnología que consuma la menor potencia posible, además también se puede hacer un uso eficiente de ellas, colocando sensores en los pasillos para que solo se enciendan cuando pasan las personas o con reguladores de luminosidad para habitaciones o salones. Con los reguladores de luminosidad además se consigue una mayor comodidad al poder elegir la luz que se necesita.
- Calefacción. El sistema de calefacción puede variar dependiendo de si se utiliza aire, agua o electricidad. El control de cada uno de ellos es importante para el confort y que no se desperdicie la energía, lo más útil en una vivienda a la hora de calefacción (y en la climatización en general), es la zonificación, es decir, dividir la vivienda en zonas cada una con la temperatura necesaria. Si la calefacción es eléctrica se utilizan calefactores, que son dispositivos utilizados para calentar el ambiente basados en el efecto Joule, es decir, la corriente pasa por una resistencia, de manera que la potencia se disipa en forma de calor.
- Aire acondicionado: podemos dividirlo en dos tipos, un sistema central que se encarga de toda la vivienda normalmente a través de un flujo de aire o un sistema individual que acondiciona espacios individuales.
- Electroválvulas. Son dispositivos que permiten o cortan el suministro de fluidos (agua, gas, gasoil...), son muy importantes ya que con ellas por ejemplo podemos controlar que zonas queremos calentar de la casa o cortar el suministro general de la vivienda si hay una fuga o se está inundando.
- Cerraduras eléctricas. Son dispositivos que facilitan el acceso autorizado a la vivienda de una forma segura.
- Señalización acústica y luminosa: este tipo de actuadores se utiliza para dar alarmas o informas de algún evento que está sucediendo en la vivienda.

- Otros actuadores. Electrodomésticos. Cada vez la automatización es mayor de manera que algunos electrodomésticos pueden ser manejados por el controlador de la casa.

### 3. Visualizador de datos

El visualizador de datos es un dispositivo físico con un interfaz hombre-máquina sirve para ver los datos de los sistemas de la vivienda, es decir que se puede conocer desde el estado de los sistemas hasta informes sobre energía y gastos, pasando por las medidas de los sensores.

Los visualizadores han ido variando y evolucionando a lo largo de los años, además se encuentran gran variedad de ellos. Los primeros visualizadores de datos que salieron al mercado fueron los displays de siete segmentos (como se ve en la Figura 13) en los que se pueden ver caracteres alfanuméricos. Este tipo de display tiene una gran limitación a la hora de mostrar información, aunque se puedan poner varios seguidos como en la imagen de la Figura 14.

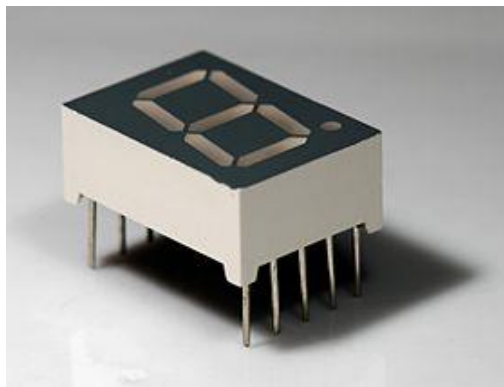


Figura 13. Display de siete segmentos.

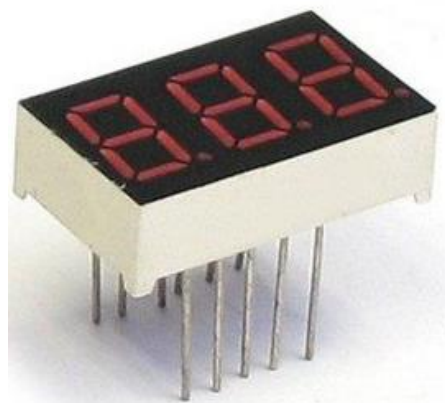


Figura 14. Tres Displays de siete segmentos unidos.

Las pantallas LCD (Liquid Cristal Display), como la de la Figura 15, se desarrollaron más tarde, son pantallas que muestran caracteres alfanuméricos, que constan de dos líneas y dieciséis caracteres en cada línea.

En este tipo de pantalla la información que se puede mostrar es mayor que en los displays.



Figura 15. Display LCD.

Después de las pantallas LCD, las pantallas han ido evolucionando y mejorando el entorno gráfico para una mejor comunicación y visualización de los datos, desde una pantalla como la que se ve en la Figura 16, que es en dos colores, hasta una pantalla como la de la Figura 17, que suelen ser más grandes y permite un entorno gráfico más desarrollado.

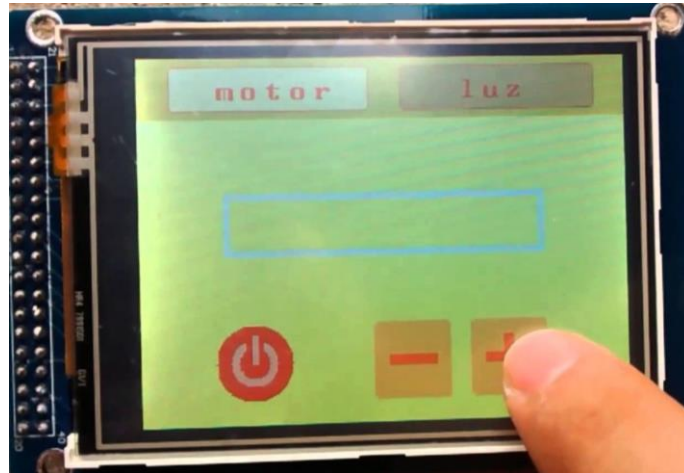


Figura 16. Pantalla LCD Graficadora.



Figura 17. Pantalla LCD tft.

La última variante que interesa para el desarrollo de este trabajo son las pantallas táctiles (como la de la Figura 18), que suponen una revolución porque se puede ver los datos e interactuar con el sistema a través de un mismo elemento.



*Figura 18. Pantalla táctil.*

Las pantallas se pueden clasificar por la tecnología con la que son construidas. La mayoría de las pantallas utilizan la tecnología LCD en la variante TFT-LC (Thin Film Transistor) que utilizan transistores de película delgada en los píxeles del cristal líquido. También se debe distinguir en la tecnología LCD entre los normales, iluminados por lámparas fluorescentes de cátodo frío, y los LCD-Led que están cuya iluminación se produce a través de tecnología led. En un desarrollo tecnológico parecido al de las pantallas LCD se encuentran las pantallas de plasma que utilizan gases nobles para generar puntos de color, el problema de esta tecnología es que se degrada y dura menos tiempo que las pantallas LCD. Por último, se encuentran las pantallas OLED que utilizan diodos de materiales orgánicos, con esto se crean pantallas ultra finas y flexibles.

### Capítulo 3. Controlador de la vivienda

El controlador es el que se encarga del gobierno y correcto funcionamiento de la vivienda, es por llamarlo así, el cerebro del hogar. El controlador recoge la información de todos los sensores instalados y da las ordenes pertinentes a los actuadores según la información recibida.

Para el correcto desarrollo del este trabajo se debe diferenciar entre controlador y microcontrolador.

Un controlador es un dispositivo que ejerce su control sobre uno o varios procesos, un microcontrolador también manda sobre uno o varios procesos, la diferencia es que el microcontrolador es un circuito integrado con alto nivel de integración que contiene la mayoría o todos los elementos de un controlador (Universidad Miguel Hernández, 2007), es decir ya integrados las memorias, los puertos de entrada/salida... Por lo tanto, a partir de aquí se hablará de microcontrolador.

Los elementos principales de un microcontrolador ( $\mu C$ ) son (Universidad Miguel Hernández, 2007):

- Procesador o CPU. Es el hardware donde se procesan los datos, está compuesto generalmente: de la unidad aritmético-lógica (ALU), donde se realizan las operaciones aritméticas y lógicas; y la unidad de control (CP) que recoge tomas las instrucciones de la memoria, y las ejecuta, después de decodificarlas.
- Memoria RAM. Es una memoria de acceso aleatorio que utiliza donde el microcontrolador guarda los datos que utiliza en el momento, este tipo de memoria es utilizada porque es mucho más rápida que las memorias normales, aunque su capacidad es mucho menor y la relación coste capacidad mayor. La memoria RAM es una memoria volátil que en cuanto se apaga el dispositivo ser borra.
- Memoria para programa (ROM/PROM/EPR0M). Es una memoria no volátil, es decir que no se borra al apagarse, que tiene las instrucciones para la correcta ejecución del programa, también puede tener información para el arranque e información sobre el fabricante...
- Puertos de E/S (Entrada/Salida). Sirven para comunicarse con el exterior, por ejemplo, a través de los puertos de entrada al microcontrolador le llega información de los sensores y por medio de las salidas puede dar órdenes a los actuadores o mostrar datos en pantallas.
- Módulos de control de periféricos. Son módulos creados para que el programa y el control de los periféricos sea correcto. Algunos de los más importantes son los temporizadores, puertos de comunicación, conversor analógico/digital y viceversa, interrupciones, watchdog...

- Reloj del sistema. El reloj del sistema genera impulsos que sincroniza todas las acciones para que el  $\mu$ C funcione correctamente.

Los microcontroladores son construidos por numerosos fabricantes, dentro de cada fabricante hay diversas familias con diversos  $\mu$ C y cada uno de ellos tiene unas características concretas. Para el desarrollo de este trabajo se utilizará un microcontrolador Arduino.

### 1. Microcontrolador Arduino

Arduino es una idea que va un paso más allá de un microcontrolador, Arduino es “una plataforma de código abierto basada en software y hardware fáciles de usar” (Aprendiendo Arduino, 2017). Está pensado y diseñado para que el uso, el diseño y la programación de microcontroladores sea más sencillo. Arduino nació en el *Ivrea Interaction Design Institute* con la idea de que fuera una herramienta de prototipado rápido para personas con escasa formación (Aprendiendo Arduino, 2017). La potencia de la idea Arduino radica en el código abierto y en una comunidad que comparte miles de proyectos, por lo que es muy fácil aprender, mejorar y crear proyectos con la experiencia de los anteriores.

Las características positivas de Arduino son:

- Dispositivos baratos en comparación con otros microcontroladores.
- Multiplataforma. El entorno de programación de Arduino es ejecutable en Windows, Macintosh OSX y Linux.
- El entorno de programación es sencillo y fácil de usar.
- Es de código abierto, una herramienta muy útil para los programadores expertos.
- Los planos de microcontroladores están disponibles de forma que se pueden modificar y mejorar.

Las características negativas son:

- Para que la programación sea sencilla se utiliza librerías que ralentizan la ejecución del programa, para programas normales no tiene mucha importancia, pero para programas en los que cada microsegundo cuenta es importante (ej.: sistema ABS de un coche).
- El microcontrolador está montado ya en una placa de circuito impreso con unas determinadas características lo cual quita flexibilidad a los proyectos que se pueden realizar.

La idea de Arduino se ha desarrollado mucho en los últimos años, gracias a ello tenemos una gran variedad de microcontroladores para multitud de aplicaciones. Tipos de microcontroladores:

- Nivel inicial: son Arduinos sencillos destinados para personas que estén empezando o para proyectos de poca envergadura.
- Características mejoradas: tienen una o más características mejoradas para aplicaciones concretas o para simplemente para darle mayor potencia.
- Internet of things: estos Arduinos están diseñados para la conexión a internet a través de Wifi o cable ethernet.
- Educación: Arduino tiene un microcontrolador destinado para la educación.
- Usables: son microcontroladores pequeños y de pequeño tamaño destinados a proyectos como la ropa inteligente.
- Impresión 3D: De esta clase solo hay un microcontrolador que está diseñado para su uso en el control y manejo de impresoras 3D.

De cada tipo suele haber varios microcontroladores dependiendo de la finalidad, el número de entradas y salidas, puertos de comunicación...



## 2. Sistemas de comunicación de Arduino

En los sistemas domóticos son importantes las formas de comunicación con alguno de los sensores, los actuadores y sobre todo con otros microcontroladores y con el exterior. Poder controlar el sistema domótico a distancia, ya sea desde un smartphone o desde un ordenador se ha convertido en una prioridad, por este motivo hay numerosos protocolos de transmisión de información. A continuación, se explicarán los que se han considerado más importantes para el desarrollo de este trabajo.

### 2.1. Comunicación puerto serie

La comunicación se realiza a través del puerto UART (recepción-transmisión asíncrona universal), como su propio nombre indica es una comunicación asíncrona por lo que no utiliza una señal de reloj para sincronismo. Este tipo de comunicación necesita dos cables, como se ve en la Figura 19, uno de ellos es para transmitir la información (tx) y el otro para recibirla (rx).

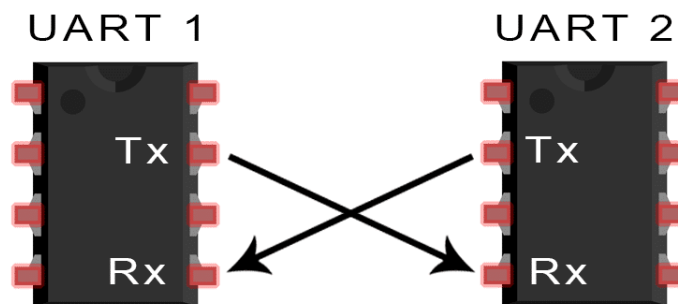


Figura 19. Transmisión UART.

La transmisión de información se realiza como se ve en la Figura 20, donde el primer bit es de inicio de comunicación, a nivel bajo (cero lógico), después les siguen 8 bits de datos y finaliza con un bit de finalización, a nivel alto (uno lógico).

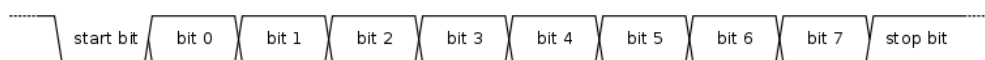


Figura 20. Trama de datos UART

La velocidad de transmisión de la comunicación UART está limitada a 2Mbps. En este tipo de comunicación es necesario unir las masas para que los valores de referencia de tensión sea la misma.

### 2.2. I<sup>2</sup>C

La comunicación I<sup>2</sup>C (Inter-Integrated Circuit) es un protocolo de comunicación síncrono. Físicamente está compuesto de tres cables, el primero de ellos es la masa que debe ser común a los dos dispositivos, cosa sencilla porque normalmente los dispositivos que se quieren comunicar están en la misma

placa de circuito impreso o comparten la misma fuente de alimentación, por lo que en realidad se podría decir que son necesarios solo dos cables. La utilidad de estos dos cables es la de reloj (SCL) y la de datos (SDA). El cable SCL envía una señal de reloj que sincroniza los dos dispositivos, de manera que el dispositivo crea esta señal de reloj es el maestro y el otro el esclavo, a través del cable SDA se mandan datos en los dos sentidos, aunque las comunicaciones están controladas por el dispositivo maestro.

I<sup>2</sup>C es una comunicación serie cuya primera versión fue creada en el año 1992 por Philips, la segunda versión (versión 2.1) es del año 2000. La comunicación estándar tiene una velocidad de 100 Kbit/s, también existe el “fast” (rápido) que alcanza velocidades de 3.4Mbit/s. Este protocolo es utilizado en industria en la comunicación entre microcontroladores y periféricos en sistemas integrados (Aprendiendo Arduino, 2017).

Una de las ventajas de este protocolo de comunicación es que puede haber múltiples esclavos por lo que puede haber comunicación entre gran variedad de dispositivos, como se puede ver en la Figura 21. Una de las ventajas de esta comunicación es que el maestro no tiene que ser siempre el mismo y puede cambiar de un dispositivo a otro según las necesidades.

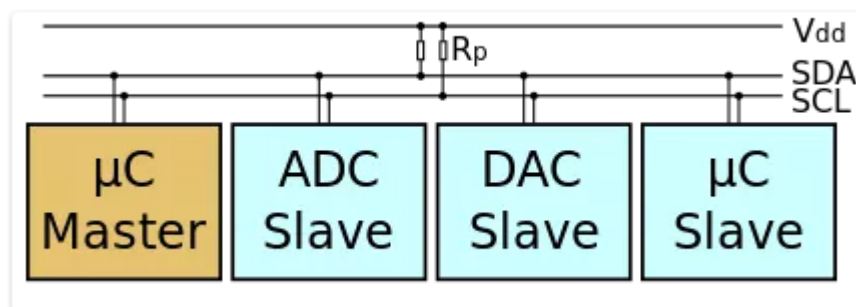


Figura 21. Esquema comunicación I<sup>2</sup>C

La comunicación sigue se produce como en la Figura 22, cada una de las partes tiene la siguiente función (Aprendiendo Arduino, 2017):

| start | A7 A6 A5 A4 A3 A2 A1 R/W | ACK | ... DATA ... | ACK | stop | idle |

Figura 22. Trama de datos I<sup>2</sup>C

- Start. El maestro empieza la comunicación enviando una secuencia llamada “start condition” que pone en espera los esclavos para que esperen una comunicación.
- En el siguiente byte A7-A1 está la dirección del dispositivo con el que quiere comunicarse, y el bit de menor peso (A0) como de lectura/escritura si A0=1 entonces la operación es de lectura, el

maestro recibe información del esclavo, y si A0=0 la operación es de escritura, el maestro manda información al esclavo. Cuando el maestro manda la dirección cada esclavo la compara con la suya y si coincide ya está direccionado como esclavo-emisor o esclavo receptor.

- ACK (acknowledge). Es un bit de reconocimiento que se envía después de cada byte de información.
- Data. Envía la información que se quiere comunicar, después de cada byte de información hay un bit ACK.
- Stop. Para terminar la comunicación y dejar libre el bus el maestro manda el patrón “stop condition”.

### 2.3. Ethernet

Ethernet es un estándar para redes de área local, que define las características de cableado y los formatos de las tramas datos, también es conocido con IEE 802.3.

Ethernet sigue las reglas del protocolo CSMA/CD (acceso al medio por detección de portadora y detección de colisiones) que son (Turiel, 2016):

- Escanear el bus en busca de actividad, si no la hay transmitir después de un retardo llamado “*interframe delay*”. Si hay actividad ir al paso 2.
- Cuando hay actividad en el medio, se escucha hasta que se queda libre, después de esperar el *interframe* se empieza a transmitir.
- Si durante la transmisión de la información se detecta una colisión, ocurrirá sobre todo al principio, se aborta la transmisión y se envía una señal de atasco “*Jam signal*”.
- Una vez enviado el “*Jam signal*” hay que esperar un tiempo aleatorio y volver al paso uno.

La trama de datos se envía como se ve en la Figura 23: primero se envía un preámbulo (PRE) que tiene un tamaño de 7 octetos; después se envía el delimitador del envío de trama (SFD) cuyo tamaño es de un octeto; al SFD le sigue la dirección adonde se envía la trama (DA); la siguiente dirección es desde donde se envía la trama (SA), el tamaño de los dos últimos datos es de dos octetos; los siguientes dos octetos definen un campo de control; después le siguen los bytes de datos (Data) cuyo tamaño puede variar desde 46 a 1500 octetos; la penúltima información es el PAD, que es la parte transmitida que realmente se quiere enviar; y por último se envía el FCS, que tiene un tamaño de octetos, y es la secuencia de comprobación de frames.

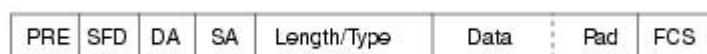


Figura 23. Trama de datos Ethernet.

La tecnología ethernet ha ido variando y evolucionando con el paso del tiempo, sobre todo en cuanto a velocidad se refiere, por ello, hay diferentes variantes como se puede ver en Tabla 1.

Tabla 1. Variantes de Ethernet.

Tecnología	Velocidad de transmisión	Tipo de cable	Distancia máxima de segmento
10Base2	10 Mbit/s	Coaxial	185 m
10Base5	10 Mbit/s	Coaxial	500 m
10BaseT	10 Mbit/s	Par Trenzado	100 m
10BaseF	10 Mbit/s	Fibra óptica	2000 m
100BaseT4	100 Mbit/s	Par Trenzado (categoría 3UTP)	100 m
100BaseTX	100 Mbit/s	Par Trenzado (categoría 5UTP)	100 m
100BaseFX	100 Mbit/s	Fibra óptica	2000 m
1000BaseT	1000 Mbit/s	(categoría 5e ó 6UTP)	100 m
1000BaseSX	1000 Mbit/s	Fibra óptica	550 m
1000BaseLX	1000 Mbit/s	Fibra óptica	5000 m

## Capítulo 4. Android

Android es un sistema operativo basado en Linux, que se utiliza para móviles inteligentes, tablets e incluso relojes, televisores y automóviles. Este sistema operativo fue diseñado para dispositivos con pantalla táctil por la empresa Android Inc. en que más tarde fue comprada por Google. La ventaja principal de Android es que es un sistema operativo de código abierto al que cualquier persona tiene acceso, desde los fabricantes de smartphones hasta los usuarios, esto permite una gran versatilidad y desarrollo en la fabricación de teléfonos móviles y además mejora la competencia y competitividad en el mercado, favoreciendo una disminución de los precios. Además, el código abierto y la gran difusión del sistema operativo Android facilitan la creación de aplicaciones que llegan a gran parte del público, lo que favorece la creación y el asentamiento de empresas y oportunidades de desarrollo para personas emprendedoras.

### 1. Características del sistema operativo Android

Las características principales de Android son (Caraterísticas, Android OS, 2017):

- Sistema operativo de código abierto para dispositivos con pantalla táctil.
- Núcleo del sistema operativo basado en el núcleo de Linux.
- Android se puede adaptar fácilmente a muchos tipos de pantallas y resoluciones.
- Para el almacenamiento de datos utiliza SQLite, que es un motor de bases de datos relacionales potente y de poco peso.
- Tiene diversas formas de mensajería entre dispositivos.
- El navegar web está basado en WebKit, que es la base de navegadores como Google Chrome o Safari.
- Soporta aplicaciones java y la mayoría de los formatos multimedia.
- Soporta HTML, HTML5...
- Herramientas de depuración de la memoria, análisis de rendimiento software y un emulador de dispositivos.
- Capacidad de realizar multitarea de aplicaciones.
- Aplicación Google Play donde ver y descargar cualquier tipo de aplicación.

## 2. Arquitectura Android

Para comprender Android es necesario conocer su arquitectura (Figura 24), lo primero es el kernel de Linux, que es la base, sirve para la generación de subprocesos y la administración de memoria de bajo nivel, además controla mediante los drivers (controladores) el hardware. Después tenemos la capa de abstracción de hardware que aporta interfaces estándares para la utilización del hardware desde el *JAVA API Framework*, como el módulo de la cámara o del bluetooth.

Al mismo nivel tenemos el tiempo de ejecución de Android y las bibliotecas C/C++ nativas. El tiempo de ejecución de Android (ART) sirve para ejecutar varias máquinas virtuales en dispositivos de memoria baja que están ejecutando archivos DEX.

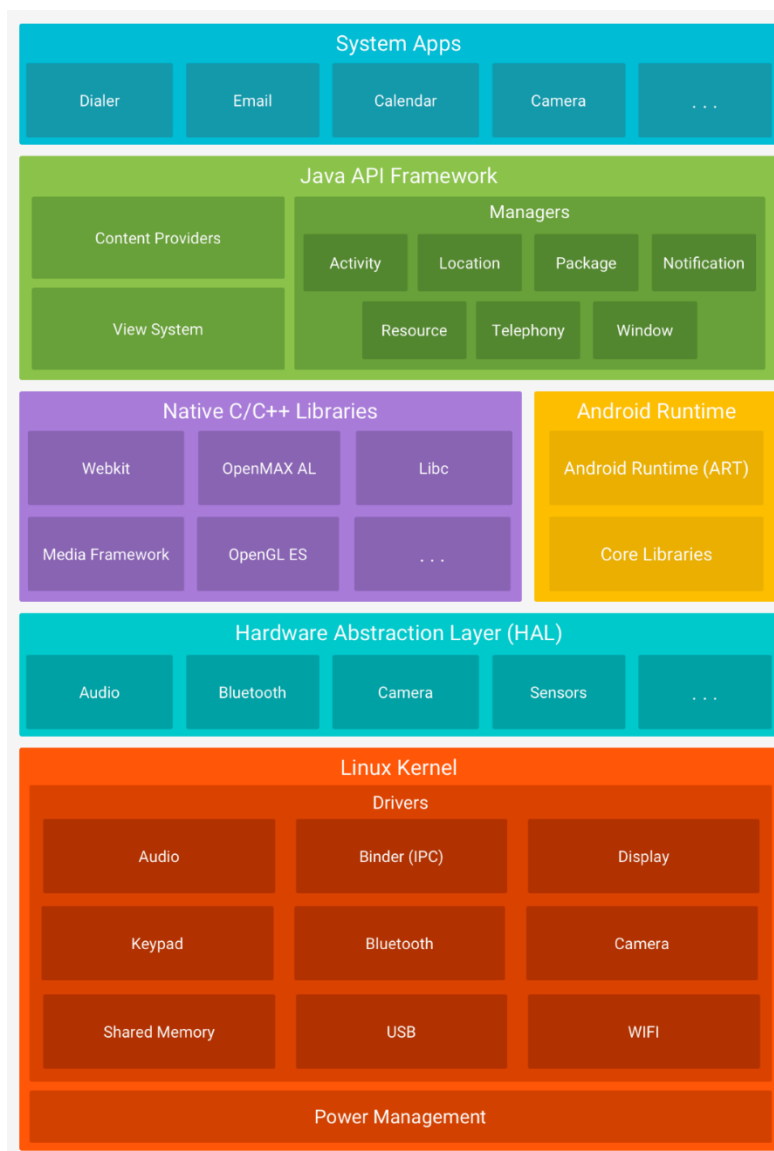


Figura 24. Arquitectura Android.

Las bibliotecas C/C++ nativas son necesarias porque muchos de los servicios centrales y componentes de Android están basadas en códigos que necesitan bibliotecas escritas en C y C++.

Entorno de trabajo de la Java Api (interfaz de programación de aplicaciones). Todas las funciones de Android están escritas en Java y disponibles a través de la APIs.

Android tiene también Apps del sistema como el la de correo electrónico, mensajería, calendarios, contactos, navegador web... estas aplicaciones son como las que un usuario puede instalar (API Guides, Android Developer, 2017).





## Desarrollo del trabajo de fin de grado

### Capítulo 5. Diseño de una vivienda domótica

En este capítulo se va a tratar sobre el diseño de la vivienda y de los sistemas domóticos que se quieren implantar. La primera parte del capítulo se dedica a la explicación de la vivienda con las características fundamentales con el fin de obtener una visión de la casa. La segunda parte del capítulo se hablará de los sistemas domóticos que se van a integrar en la vivienda, así como de las razones por las que se han introducido estos sistemas.

#### 1. Definición y planos de la vivienda

La vivienda que es el objeto de aplicación de este TFG es una casa de tres pisos de altura, dos de ellos están dedicados a la vivienda y el otro es un garaje/sótano, que puede ser usado para guardar dos vehículos y como despensa.

El garaje/sótano es un único espacio que no está dividido, donde hay sitio para almacenaje de diversos objetos, se guardan los vehículos y está ubicada la caldera para la calefacción, este piso tiene una entrada desde el exterior a través de la puerta del garaje y de unas escaleras desde la planta principal. El garaje cuenta con dos sensores de movimiento. Todo se puede ver en los planos 1 y 4 que están en el anexo III.

La planta principal contiene un salón, una cocina, el cuarto de lavadoras y un baño, además de el recibidor o hall de la casa, en esta planta es donde se desarrolla la vida cotidiana de los ocupantes de la vivienda. Esta planta cuenta con dos entradas/salidas al exterior, una es la puerta principal de la vivienda y la otra es una puerta que da a un porche además también se puede acceder a través del garaje o del piso superior, como se puede ver en el anexo III, plano 2. La planta se divide en dos zonas para la calefacción, una de ellas la cocina y la otra el salón y además tiene 4 sensores para la alarma (anexo III – plano 5).

La planta superior de la vivienda está destinada a los cuartos personales de los ocupantes de la vivienda, cuenta con cuatro habitaciones y dos baños, uno de ellos está dentro de la habitación principal, a esta planta solo se puede acceder a través de interior de la vivienda por una escalera, como se puede ver en el plano 3 (anexo III). Esta planta cuenta con cuatro zonas independientes en cuanto a calefacción, que son las cuatro habitaciones y cinco sensores para la alarma (Plano 6 – anexo III).

## 2. Sistemas domóticos que implantar en la vivienda

En este apartado se explicarán los sistemas domóticos que se van a implantar en la vivienda unifamiliar. Como se observa del apartado anterior la vivienda es muy amplia lo que dificulta que haya muchos sistemas automatizados debido a que se necesitarían múltiples sensores y actuadores.

Como se ve en la Figura 1, los aspectos fundamentales de la domótica son la gestión energética, la seguridad, las comunicaciones y el confort, teniendo en cuenta estas características se ha decidido integrar los sistemas de calefacción, control de las persianas, control automático de la puerta del garaje y seguridad en la vivienda. Además de estos sistemas es importante el visualizador de datos, que será una pantalla táctil donde se verán datos, pero sobre todo donde se elegirán los sistemas que se quieren activar y como se quieren activar.

### 2.1. Sistema de calefacción

El sistema de calefacción es uno de los sistemas más importantes de la vivienda pues aporta confort al hogar y es imprescindible en zonas geográficas con un frío intenso.

Un sistema de calefacción optimizado, además de aportar confort a la vivienda es imprescindible para un uso y control eficiente de la energía. La energía empleada en el calentamiento de la casa normalmente es alta y fácil de malgastar.

Uno de los subsistemas más importantes en la climatización es la zonificación, es decir, dividir el espacio de la vivienda en zonas independientes, visto desde el punto de vista climático, que normalmente están separadas también físicamente. Al separar la vivienda en zonas se puede trabajar con cada zona como si fuera un sistema independiente, de manera que cada zona puede tener una temperatura diferente en función de las necesidades de los ocupantes de la vivienda, por ejemplo: normalmente las zonas como el salón necesitan una temperatura mayor que las habitaciones donde se va a dormir. Otra de las ventajas de la zonificación es que permite calentar unas zonas de la vivienda y otras no, de manera que si no se utiliza una habitación o una zona puede estar desconectada del resto y así se ahorra energía.

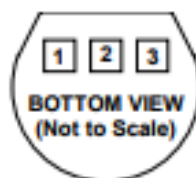
El sistema de calefacción elegido para esta vivienda es calefacción de suelo radiante mediante agua caliente (Figura 25), para calentar el agua se utiliza una caldera de gas estándar. Para controlar las zonas con este sistema de calefacción se debe controlar el flujo a través de la manguera que llevan el agua caliente, siendo cada una de las tuberías la que corresponde a una de las zonas del domicilio.



Figura 25. Suelo radiante por agua caliente.

Para el control y buen uso de este sistema se tiene la ayuda de dos elementos: sensores y actuadores. Los sensores para este sistema son sensores de temperatura, que ofrecen una medida de la temperatura real de la sala, los sensores de temperatura que se van a utilizar son los TMP36.

El sensor TMP36 es un sensor de temperatura lineal, en que al subir la temperatura aumenta el voltaje de salida, además tiene un encapsulado similar al de un transistor, como se puede ver en la Figura 26.



PIN 1, +V<sub>S</sub>; PIN 2, V<sub>OUT</sub>; PIN 3, GND

Figura 26. Encapsulado TMP36.

El sensor TMP36 tiene un rango de medida que va desde -40°C hasta 125°C, y tiene una exactitud de  $\pm 1^\circ\text{C}$ , a 25°C la salida del sensor es 750mV y el aumento de la temperatura es 10mV/°C. Como se observa en el Figura 27 es un sensor lineal del cual es posible sacar fácilmente la ecuación de la recta que rige el sensor.

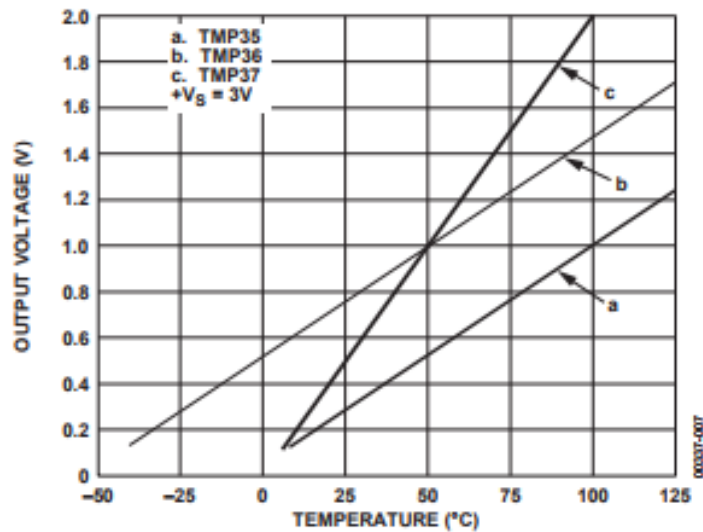


Figura 27. Voltaje de salida vs. Temperatura.

Los actuadores del sistema de calefacción son dos. El primero de ellos es la caldera de gas que calienta el agua que luego circulará por las tuberías. El segundo actuador de este sistema son las electroválvulas con las que se controlaran el paso de agua caliente a las zonas, y así se controla la temperatura. Las electroválvulas que se instalarán en la vivienda son válvulas desmontables Orkli, como la que se ve en la Figura 28.



Figura 28. Electroválvula desmontable Orkli.

La tensión de alimentación de la electroválvula puede ser de 230V o 24V. La temperatura del fluido que circula a través de ella no debe ser inferior a 5°C, ni superior a 90°C, además la presión máxima estática es de 10 bares. La electroválvula tarda un tiempo de 12 segundos en abrirse completamente y 5 segundos en cerrarse.

## 2.2. Sistema de persianas

El control automatizado de las persianas es un elemento que ayuda a la hora de volver más cómoda la vivienda, que se puedan subir y bajar las persianas simplemente con pulsar un botón o una pantalla táctil aporta confort y facilita la vida de los inquilinos, además hay persianas que por su gran tamaño son difíciles de mover. También que este sistema esté automatizado es de gran ayuda para los ocupantes de la casa que tengan las capacidades físicas mermadas, como personas mayores o discapacitados físicos.

Para que la subida y bajada de persianas sea automático se instalarán motores tubulares que se instalaran en el eje de persianas, como se ve en la Figura 29.



Figura 29. Motor tubular en el eje de la persiana.

Los motores tubulares que se usarán van a ser motores estándar de 230V como el que se ve en la Figura 29, el consumo y la potencia de cada motor vendrá determinado por las características de la persiana (tamaño, peso...). Además, en cada persiana se utilizarán finales de carrera para controlar donde si la persiana está abajo o arriba del todo.

El sistema de persianas es importante y está relacionado también con el sistema de seguridad, que en el siguiente apartado se explicará, cuando en la vivienda se simule que hay ocupantes se subirán las persianas por el día y se bajaran por la noche para ello se medirá la luz exterior a través de una resistencia *ldr* (light-dependent resistor). Una *ldr* es una resistencia que varía en función de la luz que recibe, tiene un aspecto como el de la Figura 30. En esta instalación domótica se van a utilizar dos *ldr* una de cada lado de la casa para tener dos medidas y actuar en función de las medidas.

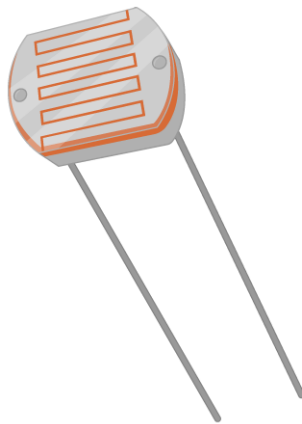


Figura 30. LDR.

La *ldr* que se va a utilizar en este proyecto es una GL5528 cuyas características son: que aguanta un voltaje máximo de 150V; funciona con temperaturas entre -30°C y 70°C; la resistencia que ofrece en oscuridad es de 10 a 20 K $\Omega$  y con luz brillante unos 1000K $\Omega$ ; y tiene un tiempo de respuesta ante cambios de luminosidad de 25 milisegundos ( (pi.gate.ac.uk, 2017)).

### 2.3. Sistema de seguridad

La seguridad es fundamental en una vivienda, tener la certeza de que la casa y posesiones están a salvo ha sido siempre una preocupación para la humanidad, por lo tanto, el sistema de seguridad de una vivienda es crítico.

Además de la confianza que infunde tener un sistema de seguridad, en la vivienda actúa como elemento disuasorio frente a intrusiones de delincuentes.

El sistema de seguridad de esta vivienda se va a basar en la colocación de sensores en los posibles puntos de acceso de la vivienda, zumbadores cuando salte la alarma y el control de las persianas para simular que la casa está habitada y asustar a los que hayan entrado en la vivienda de forma ilegal.

Unas persianas automatizadas permiten poder simular que hay personas viviendo en la casa, subiéndolas por el día y bajándolas por la noche, cuando por ejemplo los ocupantes están de vacaciones durante un largo periodo de tiempo, disuadiendo a ladrones que busquen casas deshabitadas o en la que los inquilinos no estén. Además, cuando salte la alarma se pueden bajar las persianas automáticamente, que en conjunto con el sonido estridente de los zumbadores sirven para desorientar y sobresaltar a los ladrones para que abandonen la casa. En conclusión, las persianas y los zumbadores son los actuadores de nuestro sistema.

La presencia de intrusos en la vivienda se detectará mediante el sensor HC-SR501 PIR Motion Sensor, como el de la Figura 31, que detecta el movimiento

a través del calor que emiten los cuerpos. Este sensor detecta cuando hay una variación de calor. El calor se emite en forma de radiación infrarroja que es capturada por el sensor y activa una salida de detección de movimiento.



Figura 31. Sensor HC-SR501 PIR.

Normalmente el PIR está dividido en dos mitades, como se ve en Figura 32, para detectar el cambio de radiación en uno y otro.

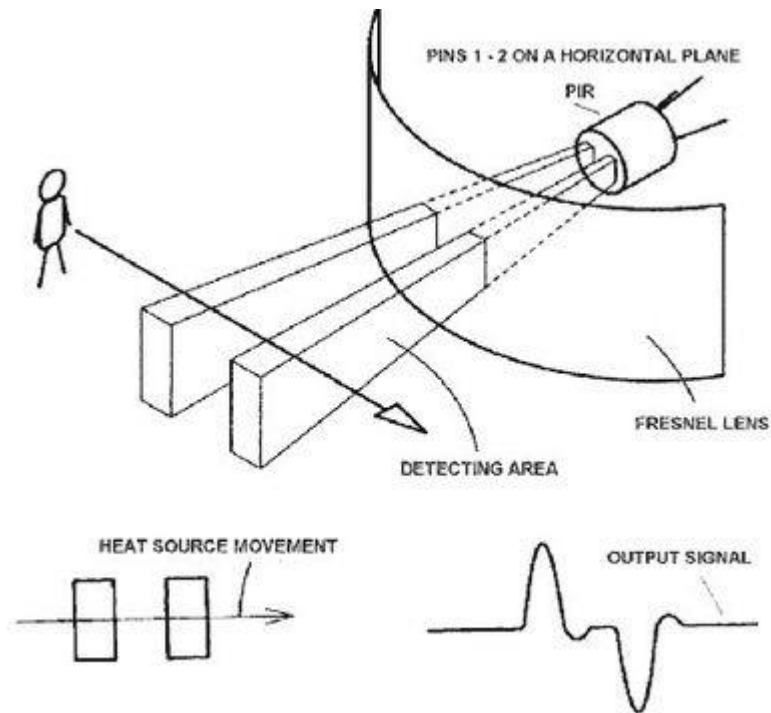


Figura 32. Esquema de funcionamiento de un sensor PIR.

Las características principales del sensor con el que se va a trabajar son:

- Rango de voltaje de 4.5V a 20V.
- Consumo cuando no está activo 50uA.
- El nivel alto son 3.3V y el nivel bajo son 0V.



- Las dimensiones de la placa de circuito impreso son 32x24mm.
- El ángulo de detección es de 110°.
- La máxima distancia a la cual detecta un objeto es 7.7m.

Los actuadores del sistema de seguridad son las persianas y los zumbadores, de las primeras ya se ha hablado en el bloque anterior. Los zumbadores entran dentro de la categoría de señalización acústica, sirven para avisar a los ocupantes de la vivienda de una entrada no autorizada. Un zumbador es un transductor electroacústico que produce sonido, hay dos tipos de zumbadores o buzzers:

- Buzzer pasivos: son aquellos que necesitan recibir una onda de frecuencia para generar un sonido.
- Buzzer activo: son aquellos que solo necesitan alimentación para funcionar, pues tienen un oscilador interno que genera una frecuencia.

En la vivienda domótica se va a utilizar el buzzer activo YL-44 (Figura 33), que admite un voltaje de entrada de 3.3V a 5V, tiene el pin I/O, como se ve en la Figura 33, que es un “enable” externo que sirve para activar o desactivar el zumbador.



Figura 33. Buzzer activo YL-44

#### 2.4. Puerta del garaje

El control de la puerta del garaje a distancia y de forma automática está cada vez más extendido y es casi impensable poner una puerta nueva a un garaje y que no abra y cierre sola, por ello en la vivienda domótica la puerta del garaje será automática. Para ello, debemos tener un actuador que controlado permita abrir y cerrar la puerta.

El motor que vamos a utilizar es un motor para una puerta de persiana, como el de la Figura 34, que funciona a 230V y con final de carrera para conocer cuando está completamente cerrada y cuando completamente abierta la puerta.





Figura 34. Motor puerta garaje.

La forma que tendrán los ocupantes del hogar de abrir la puerta desde el exterior es a través de un mando que hace las veces de “sensor” y desde el interior con un botón en la pantalla táctil.

El mando se hará con una tarjeta o llavero que el RFID RC522 leerá, como el de la Figura 35. El RFID es un identificador por radiofrecuencia inalámbrico que lee las etiquetas. Las etiquetas pueden ser llaveros, tarjetas, etc. La distancia a la que se puede realizar la medida varía en función de la frecuencia que se utilice. El mecanismo funciona de la siguiente manera, el lector envía una señal para iniciar la comunicación con las etiquetas, si hay alguna dentro del alcance recibe esta señal, la etiqueta a su vez envía otra señal al lector con la información que tiene guardada en el interior.

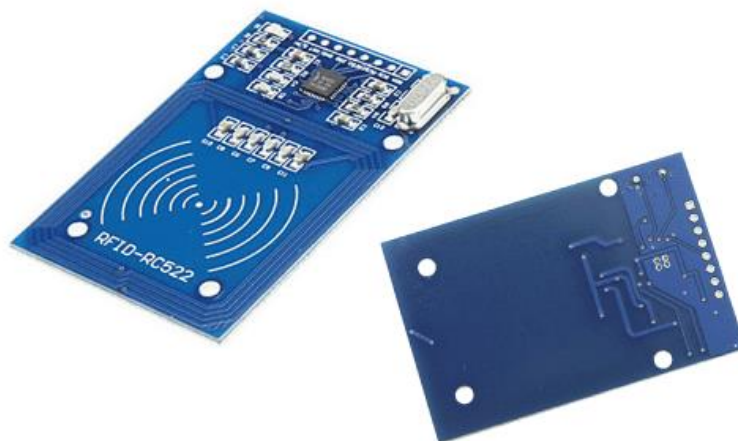


Figura 35. Receptor RC522.

### 2.5. Pantalla táctil

La pantalla táctil dentro del sistema domótico es muy importante porque es utilizada para dar información a los usuarios y también para que los usuarios introduzcan sus preferencias.

Un entorno gráfico claro y sencillo ayuda en la comprensión y utilización de los sistemas domóticos, por estos motivos se ha escogido para la vivienda Nextion. Nextion es interfaz hombre máquina para el control y visualización de una máquina o proceso. La ventaja de utilizar una pantalla Nextion frente a otro tipo de pantalla es que Nextion proporcionar un editor gráfico llamado *Nextion Editor*, como el que se muestra en la Figura 36, que permite desarrollar de manera sencilla un entorno gráfico con navegación entre pantallas y multitud de opciones para botones, barras de progreso, textos, inserción de imágenes... Otra de las ventajas de esta marca es que se carga el programa en la pantalla y en el microcontrolador solo hay que preocuparse de la comunicación con la pantalla, no hace falta programar la pantalla, que sería mucho más tedioso.

Además de un editor bueno, otra de las ventajas de usar pantallas de la marca Nextion es la gran variedad que ofrece tanto en tamaño como en recursos.

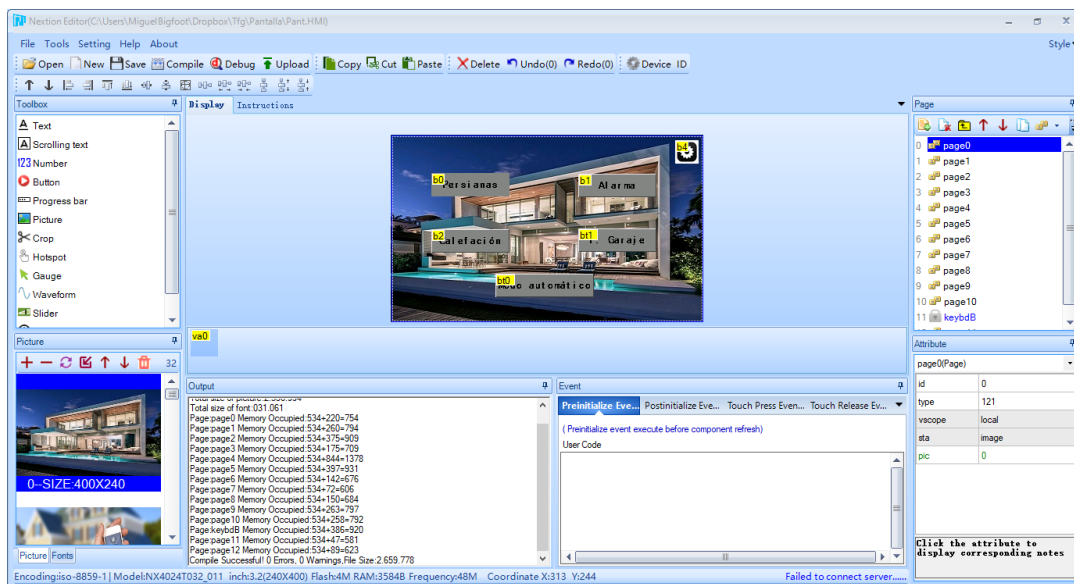
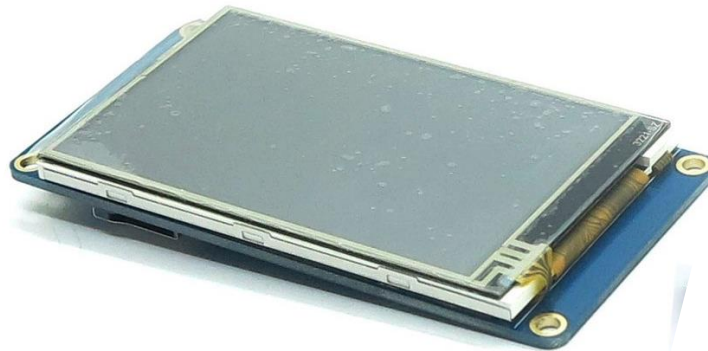


Figura 36. Nextion Editor

La pantalla elegida para este proyecto es la Nextion NX02T032\_011 (Figura 37), que ofrece una solución aceptable en cuanto a tamaño y rendimiento.



*Figura 37. Pantalla Nextion NX4024T032\_11*

Las características de la pantalla Nextion NX4024T032\_011 son:

- ✓ Tamaño de la pantalla 3.2 pulgadas.
- ✓ Resolución de la pantalla 240x400.
- ✓ Memoria flash 4Mb.
- ✓ Panel resistivo táctil.
- ✓ Memoria RAM 2048 bytes.
- ✓ Brillo 180 nit (candela por metro cuadrado).
- ✓ Peso de 42.5 gramos.
- ✓ Ranura para tarjeta micro SD.

## Capítulo 6. Gestión domótica de una casa unifamiliar basada en Arduino

En este capítulo se desarrollará la gestión e implementación de los sistemas domóticos mediante Arduino, basado en las características descritas en capítulos anteriores. Como se ha dicho, la gestión y el control de todos los sistemas domóticos se realizarán a través de Arduino, en concreto a través de dos Arduinos enlazados y comunicados en el que uno de ellos será el microcontrolador maestro y otro el esclavo.

### 1. Elección del microcontrolador

La elección de los microcontroladores debe realizarse teniendo en cuenta las necesidades del sistema domótico. Uno de los primeros parámetros que marcan o limitan el uso de los controladores es el número de entradas y salidas que tienen, por lo que se deben conocer es el número de entradas y salidas que se necesitan para el sistema domótico.

El sistema de calefacción tiene seis zonas independientes, como con los sensores de temperatura en los planos 5 y 6 del anexo III, para cada una de esas zonas es necesario un sensor de temperatura, que necesita una entrada analógica, y una salida digital para el control de la electroválvula, además es también necesaria una salida digital a mayores para el control el encendido de la caldera. En resumen, para este sistema se necesitan 6 entradas analógicas y 7 salidas digitales.

En el sistema de seguridad marcan las necesidades los sensores de movimiento y los dos actuadores, como se ve en los planos 4,5y 6 del anexo III, hay un total de once sensores que necesitan cada uno una entrada digital y dos zumbadores que necesitan únicamente una salida digital.

El control de las persianas requiere el uso de motores de corriente alterna monofásicos, su control se va realizar mediante relés, que se introducirán más adelante. Cada motor necesitará para su control tres relés que serán controlados con dos salidas digitales y además habrá dos finales de carrera en cada persiana que necesitan una entrada digital. Teniendo en cuenta que hay once ventanas como se ve en los planos 5 y 6 del anexo III, se necesitaran 22 salidas digitales y 11 entradas digitales, aparte de esto, se necesitan también dos entradas analógicas para los sensores de luminosidad.

El control de la puerta del garaje se realiza como el de una persiana más por lo que se necesitan dos salidas digitales y una entrada digital, además el lector de tarjetas se comunica con Arduino por I<sup>2</sup>C por lo que se necesita conectar dos cables, SDA y SCL, que están en unos pines concretos.

Además de los sistemas mencionados anteriormente también hay que tener en cuenta la pantalla táctil. La pantalla táctil se conecta al Arduino a través del

puerto serie, es decir que necesita dos conexiones (rx y tx) que están situados en unos pines también concretos, que varían en función del microcontrolador.

Como resumen se muestra la Tabla 2, donde se ve que se necesitan 8 entradas analógicas, 23 entradas digitales, 32 salidas digitales y 4 pines para la comunicación.

Tabla 2. Necesidades de entradas/salidas del microcontrolador.

	<b>Entradas analógicas</b>	<b>Entradas digitales</b>	<b>Salidas digitales</b>	<b>Pines de comunicación</b>
Sistema de calefacción	6	0	7	0
Sistema de seguridad	0	11	1	0
Sistema de persianas	2	11	22	0
Puerta del garaje	0	1	2	2
Pantalla táctil	0	0	0	2
<b>Suma</b>	<b>8</b>	<b>23</b>	<b>32</b>	<b>4</b>
<b>Total</b>	<b>8</b>	<b>59</b>		

En los microcontroladores Arduino los pines digitales se pueden usar indistintamente como entrada o como salida, además algunos de ellos tienen funciones a mayores como puertos de comunicación. Con la información que aporta la Tabla 2 ya se puede iniciar la búsqueda de uno o varios microcontroladores que cumplan con las necesidades del sistema.

Las necesidades básicas de nuestro sistema a la hora de elegir un microcontrolador son el número de entradas y salidas necesarias, y los puertos de comunicación que necesitamos. También hay que tener en cuenta la velocidad de la CPU, la memoria que tiene... aunque en este caso el programa no es muy complejo y no requiere muchos recursos la ejecución por que cualquiera de los microcontroladores Arduino puede servir, en este sentido.

En la Tabla 3, se han resumido las características de cada microcontrolador Arduino, teniendo en cuenta lo dicho anteriormente, para realizar una adecuada elección del microcontrolador.

Tabla 3. Comparación características microcontroladores

Nombre del $\mu$ C	Voltaje de entrada/operación	Velocidad de la CPU	E/S analógicas	E/S digitales	UART
101	3.3 V/ 7-12V	32 MHz	6/0	14	-
Uno	5V/ 7-12V	16 MHz	6/0	14	1
Pro	5V/5-12V	16 MHz	6/0	14	1
Mega 2560	5V/ 7-12V	16 MHz	16/0	54	4
Due	3.3 V/ 7-12V	84 MHz	12/2	54	4
Micro	5V/ 7-12V	16 MHz	12/0	20	1
Yun	5V	16 MHz	12/0	20	1
Ethernet	5V/ 7-12V	16 MHz	6/0	14	-

Después de observar todas las características (Tabla 3), se ve que ningún microcontrolador satisface las necesidades de entrada y salida que requiere el sistema domótico, por lo tanto, habrá que utilizar dos Arduinos y comunicarlos entre si para poder controlar el sistema. La utilización de dos Arduinos implica que se tienen que comunicar entre ellos, y que uno debe mandar sobre el otro, es decir, en la transmisión de información uno debe ser el maestro y el otro el esclavo. El primer  $\mu$ C que se ha elegido es el Arduino Mega por su gran número de entradas y salidas, aunque el Arduino Mega y Due tienen el mismo número de entradas y salidas digitales, el Arduino Due es más caro y el Mega cumple las necesidades del sistema, es decir el número de E/S es apropiado, tiene comunicación por puerto serie y también soporta comunicación I<sup>2</sup>C.

El segundo dispositivo que se necesita podría ser cualquiera que cumpliera con las necesidades de comunicación, es decir que tuviera puerto UART y permita comunicación I<sup>2</sup>C. Las necesidades de E/S las cumplen todos los dispositivos, por lo tanto, la mayoría de los de la Tabla 3 podría valer, sin embargo, además de estas características se busca un microcontrolador que se pueda conectar a internet, para una posible ampliación del sistema en el que se pudiera tener acceso a los sistemas de la casa vía Web o por una aplicación para smartphone. Teniendo esto en cuenta podríamos elegir entre el Arduino Ethernet o el Arduino Yún, de los dos se ha elegido el Arduino Yún, pues además de poder conectarse a través de cable ethernet, tiene la posibilidad de conectarse a través de Wifi. El Arduino Yun es una placa con un microcontrolador basado en el ATmega 32u4 y el Atheros AR9331, este último es un procesador que soporta una distribución de Linux, llamada Linino. La comunicación entre ambos sistemas ofrece un ordenador en red en que se pueden escribir scripts de Shell y Python para interacciones robustas (Arduino Yun, Arduino, 2017).

En conclusión, se opta por el Arduino Mega 2560 y el Arduino Yún para controlar la casa domótica, en las comunicaciones se utilizará el Arduino Yún como dispositivo maestro y el Mega como esclavo. A nivel de ejecución de los sistemas también hay diferencias entre los microcontroladores, porque uno de ellos tendrá la mayoría de los sistemas y será mandado por otro para activarlos o no. El sistema domótico de esta vivienda se puede entender, por lo tanto, como sistema domótico distribuido con dos microcontroladores, en el que uno de ellos es el maestro (Yún) y el otro esclavo (Mega), con una topología en estrella y la transmisión de la información es cableada.

## 2. Distribución de los sistemas domóticos en los microcontroladores

En el apartado anterior se explicaba que por las necesidades del sistema domótico era necesario el uso de dos microcontroladores, esto implica que es necesario comunicar los microcontroladores y los sistemas que conforman la vivienda domótica, por eso es de vital importancia hacer un buen reparto que satisfaga las necesidades del sistema, teniendo en cuenta que el Arduino Yún es el maestro y el Mega el esclavo.

### 2.1. *Sistemas domóticos controlados por el Arduino Yún*

El Arduino Yún será el maestro en el sistema domótico, por lo tanto, controlará todos los sistemas, a través del otro  $\mu\text{C}$  y el mismo, y también gobernará la comunicación con el otro Arduino. Aunque el Arduino Yún sea el que controla todos los sistemas, directamente solo controlará unos cuantos.

Como se ha dicho, este microcontrolador es el maestro, por lo tanto, controla los sistemas, pero el que decide que sistema debe encenderse y cual no, es la pantalla táctil, que es el interfaz hombre-máquina, donde el usuario decide que quiere hacer con la vivienda y como debe funcionar cada cosa. La pantalla táctil se conecta al microcontrolador a través del puerto serie, concretamente de los cables rx y tx, lo que quiere decir que necesita dos pines digitales.

Además de la pantalla táctil se ha decidido incluir el sistema de la puerta del garaje, que es controlada por la pantalla táctil y por el lector RFID, en conclusión, este sistema necesita dos salidas para controlar el motor y una salida con I<sup>2</sup>C para la comunicación con el lector RFID.

A parte de sistemas enteros, como la puerta del garaje, controla parte de otros sistemas: del sistema de alarma controla los dos zumbadores a través de una única salida, y además necesita una entrada digital con interrupción<sup>1</sup> para cuando la alarma este activada y algún sensor que detecte movimiento haga saltar en este microcontrolador la alarma, para que se activen los zumbadores; y también en este micro están las fotorresistencias para cuando esté activado el modo automático hacer la medida de la luminosidad que hay en el ambiente, estas dos LDRs necesitan una entrada analógica cada una.

Hay que tener en cuenta de igual manera que la comunicación se realiza a través de I<sup>2</sup>C, para lo que son necesarios dos pines. Como se vio cuando se habló de la [comunicación I<sup>2</sup>C](#) el mismo bus para todos los dispositivos, es decir que para la comunicación con el lector RFID y con el otro microcontrolador los cables son los mismos.

---

<sup>1</sup> Entrada con interrupción: es un pin digital del microcontrolador, configurado como entrada que cuando cambia que tiene, para la ejecución del programa y ejecuta una función especial.



Tabla 4. E/S microcontrolador Arduino Yún.

	<i>Entradas analógicas</i>	<i>Entradas digitales</i>	<i>Salidas digitales</i>	<i>Pines de comunicación</i>
<i>Sistema de seguridad</i>	0	1	1	0
<i>Sistema de persianas</i>	2	0	0	0
<i>Puerta del garaje</i>	0	1	2	2(I <sup>2</sup> C)
<i>Pantalla táctil</i>	0	0	0	2
<i>Comunicación con Arduino Mega</i>	0	0	0	2 (I <sup>2</sup> C)
<i>Total</i>	<b>2</b>		<b>9</b>	

Comparando la Tabla 4 y la Tabla 3 se ve que no se superan las capacidades del microcontrolador y que tenemos más entradas y salidas para futuras ampliaciones.

### *2.2. Sistemas domóticos controlados por el Arduino Mega*

El microcontrolador Arduino Mega actúa como esclavo y tiene que obedecer las órdenes que el Arduino Yún le comunica (I<sup>2</sup>C para comunicación), y asume las necesidades de gestión de los sistemas domóticos a su cargo. El Arduino Mega va a tener la mayoría de la información de los sensores y el control de los actuadores por lo que su buen funcionamiento es crítico.

El primer sistema que maneja este micro y al que se le da más importancia es el de seguridad, que como se ha dicho anteriormente es de vital importancia para los inquilinos de la casa. Este sistema costa de once sensores que necesitan once entradas digitales y una salida digital que haga saltar la interrupción en el Arduino Yún.

El siguiente sistema que controla es de la calefacción, que ayuda en el confort de la casa, este sistema tiene la peculiaridad que es el que más entradas analógicas utiliza, seis, una por cada sensor que está en una zona, además también controla las seis electroválvulas y el encendido de la caldera.

Por último, también controla el movimiento de las persianas a través de 22 salidas digitales para los motores y once entradas digitales para los finales de carrera.

Tabla 5. E/S microcontrolador Arduino Mega.

	<i>Entradas analógicas</i>	<i>Entradas digitales</i>	<i>Salidas digitales</i>	<i>Pines de comunicación</i>
<i>Sistema de calefacción</i>	<b>6</b>	<b>0</b>	<b>7</b>	<b>0</b>
<i>Sistema de seguridad</i>	<b>0</b>	<b>11</b>	<b>1</b>	<b>0</b>
<i>Sistema de persianas</i>	<b>0</b>	<b>11</b>	<b>22</b>	<b>0</b>
<i>Comunicación Arduino Yún</i>	<b>0</b>	<b>0</b>	<b>0</b>	<b>2</b>
<i>Total</i>	<b>6</b>		<b>54</b>	

Comparando la Tabla 5 y la Tabla 3 se ve que se van utilizar todos los pines de E/S digitales del microcontrolador, con las entradas analógicas no se tienen problemas porque aún quedan libres.

### 3. Gestión de la casa domótica

En este apartado se hablará sobre la gestión y los diferentes usos de los sistemas de la casa domótica. En capítulos anteriores se habló de los diferentes sistemas que forman parte de la vivienda, la integración eficaz de todos ellos es importante pues están relacionados unos con otros.

La gestión de los diferentes sistemas se llevará a cabo por parte de los Arduinos que son los que tienen realmente el control sobre las diferentes partes, aunque los microcontroladores deberán obedecer, siempre que sea posible, las necesidades de los ocupantes de la vivienda. Por lo tanto, dentro de la gestión de la vivienda, es un elemento de vital importancia la pantalla táctil, porque es la forma que tiene el usuario de comunicarse con los microcontroladores.

La pantalla táctil es la interfaz hombre-máquina que comunica y permite hacer uso de los distintos sistemas a los usuarios, además posibilita usar los sistemas de dos modos: el manual y el automático.

El modo manual deja modificar o activar los sistemas de calefacción, seguridad, persianas y puerta del garaje de forma independiente, en menús o pantallas distintas. En la pantalla del sistema de calefacción cuando está desactivado se muestran las temperaturas reales de las zonas y cuando está activado se muestran las temperaturas elegidas y mientras se pulsa un botón las reales. En las pantallas de control de las persianas se pueden subir o bajar una a una, o todas de golpe. El control de la puerta del garaje desde la pantalla táctil se realiza a través de un botón en la pantalla, que se ilumina mientras se abre y se cierra la puerta.

El sistema de seguridad se puede activar tanto en modo manual como en modo automático. Este sistema tiene a su vez dos modos de funcionamiento: el primero es el **modo noche** en el que se enciende el sistema de seguridad parcialmente, es decir, están activos los sensores de la planta baja y del garaje, los del piso superior, los sensores de las habitaciones no están conectados, este sistema es útil para sentirse protegido durante la noche; y el segundo modo de funcionamiento es el **modo alarma** en el que todos los sensores están conectados que será el utilizado cuando la casa esté vacía completamente. Además, el sistema de seguridad tiene características propias que no se pueden modificar, una de ellas es que cuando un sensor detecta movimiento por primera vez da diez segundos de margen para desconectar la alarma, por si fuera el dueño de la vivienda el que ha entrado, después de los diez segundos se activan los zumbadores para alertar de una intrusión a los ocupantes de la vivienda, además los ruidos estridentes confunden y ponen nerviosos a los posibles intrusos. Si el sistema de seguridad está conectado en modo noche seguiría sonando hasta que se desactive el sistema de seguridad en la pantalla táctil, pero si está en modo alarma, cuando empiezan a sonar los zumbadores,

bajarían también todas las persianas de la casa para tratar de asustar a los intrusos.

En la pantalla principal hay también un botón que lleva a una pantalla de ajustes donde se podrá cambiar la contraseña de la alarma y modificar los ajustes del modo automático.

Como ya se ha dicho cuando se activa el modo automático solo se puede conectar la alarma, pero en la configuración del modo automático se puede configurar para que tenga caliente la casa, es decir activado el sistema de calefacción, y también se puede configurar para que se levanten las persianas cuando amanezca y se bajen cuando oscurezca, y de este modo simular que en la vivienda hay personas, para disuadir a posibles ladrones y que no entren.

#### 4. Programación de los microcontroladores y la pantalla táctil

En el apartado anterior se han visto las necesidades y la configuración de los sistemas en los distintos microcontroladores, en este apartado se verán las soluciones de programación de ambos microcontroladores y de la pantalla táctil para la gestión de la casa unifamiliar. En este punto es importante tener claro cuáles son las características del sistema, además de los modos de funcionamiento.

El sistema domótico de la vivienda, como se ha dicho anteriormente, tiene que controlar los sistemas de seguridad, calefacción, persianas y puerta del garaje. En este apartado no se mostrará la programación entera, que estará en el anexo I y el anexo II, sino que se mostrarán las partes más importantes de código, por lo que dividiremos este apartado en subapartados.

##### 4.1. Comunicación entre Arduinos

La comunicación entre los distintos microcontroladores se realiza mediante el protocolo de comunicación I<sup>2</sup>C, que permite la comunicación a través de dos cables de datos. Como se ha explicado anteriormente, esta comunicación es maestro-esclavo, en la que puede haber un solo maestro y varios esclavos, lo que se utilizará porque el bus I<sup>2</sup>C se utiliza para la comunicación entre Arduinos y con la tarjeta RFID, el maestro controla la transmisión de información. En este apartado se hablará de dispositivo maestro o Arduino Yún, y dispositivo esclavo o Arduino Mega indistintamente.

El primer paso para la comunicación I<sup>2</sup>C es iniciar la comunicación, como se puede ver en la Figura 38 y Figura 39. Cuando se inicia la comunicación en el maestro (Figura 38) solo hay que introducir la instrucción que se ve, para el esclavo además hay que darle una dirección que sea única para que solo recoja la información que se dirige al esclavo.

```
Wire.begin(); //Empezamos la comunicacion I2C como maestro
```

Figura 38. Inicialización de la comunicación en el Arduino Yún.

```
//Configuración de la comunicación  
Wire.begin(1); //Empezamos la comunicacion y se pone la dirección 1  
Wire.onReceive(receiveEvent); //Función que se ejecuta cuando recibe información  
Wire.onRequest(receiveRequest); //Función que se ejecuta cuando el maestro pide información
```

Figura 39. Inicialización de la comunicación en el Arduino Mega.

En la Figura 39 se ven otras dos instrucciones aparte de la inicialización, la primera sirve para ejecutar una función cuando el esclavo detecta que estás mandando información para él. Normalmente es una función para guardar la información que está siendo recibida. La segunda instrucción sirve para ejecutar una función cuando el maestro demanda información.

#### 4.1.1. Trasmisión de información maestro-esclavo

El maestro transmite información con las tres funciones que se ven en la Figura 40, la primera de ellas sirve para comenzar la comunicación entre el maestro y el dispositivo que tiene la dirección entre paréntesis, después se utiliza la función **Wire.write()** que manda la información que quiere enviar el dispositivo y cuando se ha enviado toda la información se cierra la comunicación con ese dispositivo.

```
Wire.beginTransmission(1);  
Wire.write(AlarState);  
Wire.endTransmission(1);
```

Figura 40. Trasmisión de información Arduino Yún – Mega.

#### 4.1.2. Demanda de información del maestro al esclavo

Cuando el maestro necesita información de uno de los esclavos utiliza la función **Wire.requestFrom(a,b)** donde **a** es la dirección de dispositivo al que se pide información y **b** el número bytes que puede enviar el dispositivo.

```
Wire.requestFrom(1,8);  
while(Wire.available()) idSen=Wire.read();
```

Figura 41. Demanda y captura de información del dispositivo maestro.

En la Figura 41 se ve que pide 8 bytes de información al dispositivo con la dirección 1. La siguiente línea de código sirve para capturar la información que manda el esclavo.

#### 4.1.3. Respuesta del esclavo a la demanda del maestro

Como se ve en la Figura 39 cuando se llega una petición del maestro se ejecuta la función **receiveRequest** en la que se puede programar el envío de una información u otra en función otros parámetros del programa, como se ve en la Figura 42.

```
void receiveRequest() {  
  if(intru==1) {  
    for(int i=0;i<n;i++) {  
      if (idsen[i]==1) {  
        Wire.write(i);  
        intru==0;  
        break;  
      }  
    }  
  }  
  else for(int i=0;i<6;i++) Wire.write(TRzon[i]);  
}
```

Figura 42. Función receiveRequest del Arduino Mega.

#### 4.1.4. Recepción de la información en el dispositivo esclavo

En el momento en que el dispositivo esclavo detecta que el maestro le está enviando información ejecuta la función `receiveEvent`, programada en la Figura 39. Esta función es la encargada de recibir y almacenar toda la información que manda el dispositivo maestro.

Como se puede ver en la Figura 43 se utiliza la función `Wire.available()` con un `switch` para conocer el número de mensajes que ha enviado el maestro y en función de eso guardarlo en una variable o en otra, la instrucción que se utiliza para leer y guardar la información es `Wire.read()`.

```
void receiveEvent (int howmany){
  int i=0;
  switch(Wire.available()){
    case 1: //Información de la caldera
      CalState=Wire.read();
      break;
    case 2: //Información de la alarma
      AlarState=Wire.read();
      n=Wire.read();
      if(n==0) n=6;
      else n=11;
      break;
    case 7: //Información de la temperatura
      i=0;
      CalState=Wire.read();
      while (Wire.available(>0){
        Tzon[i]=Wire.read();
        i++;
      }
  }
```

Figura 43. Recepción de información en el Arduino Mega.

#### 4.2. Comunicación entre el Arduino Yún y la pantalla Nextion

La comunicación entre el Arduino Yún y la pantalla táctil de Nextion se realiza a través del puerto serie, como ya se ha mencionado, para ello se utilizan unas cuantas instrucciones que permiten la comunicación.

La comunicación entre estos dispositivos es especial porque la pantalla no se programa desde el microcontrolador como otras pantallas táctiles, sino que la pantalla de Nextion tiene su propia programación independiente. Por ello el micro pregunta cómo están determinados valores de los sistemas de la pantalla o los cambia, de manera que es siempre el microcontrolador que inicia la comunicación y espera la respuesta de la pantalla.

##### 4.2.1. Configuración e inicialización de la comunicación

Lo primero que debemos hacer para el control de la pantalla es crear un objeto de la clase Nextion que permita utilizar las funciones esa clase. Para ello debemos antes crear un puerto virtual para la comunicación serie en las patillas que elijamos, porque el Arduino Yún solo tiene un puerto serie que es utilizado para la comunicación con el ordenador, después ya se puede crear el objeto de la clase, como se ve en la Figura 44.

```
SoftwareSerial nextion(0,1);
Nextion nexPant(nextion,9600);
```

Figura 44. Configuración para la comunicación del Arduino Yún con la pantalla.

Después en la función `setup()` se inicia la comunicación por el puerto serie con la instrucción `nexPant.init()`, como se ve en la Figura 45.

```
nexPant.init();
```

Figura 45. Inicialización de la comunicación con la pantalla.

#### 4.2.2. Instrucciones de comunicación con la pantalla

La comunicación con la pantalla, como se ha dicho anteriormente, depende del microcontrolador que da instrucciones o pide valores a la pantalla, y esta contesta o efectúa las ordenes. Para la comunicación con la pantalla se utilizan tres funciones de las seis que ofrece la librería `Nextion.h`, de la que se hablará más adelante. Las funciones utilizadas podemos dividir las en dos tipos:

- Función para obtener información. Para obtener información de las variables los pulsadores y demás elementos de la pantalla utilizamos la función `GetComponentValue()`, dentro de los se pone el valor que se quiere obtener como se ve en la Figura 46.

```
CalState=nexPant.GetComponentValue("page4.bt0");
```

Figura 46. Función para recibir información de la pantalla.

- Funciones para mandar información y comandos. Estas funciones sirven para modificar los valores de las variables, pulsadores u objetos que estén en la pantalla y además también para mandar comandos. La función que permite modificar los valores es `setComponentValue("a",b)`, como se ve en la Figura 47, donde `"a"` es nombre del objeto o variable que se quiere cambiar y el `b` es el nuevo valor que se le da.

```
nexPant.setComponentValue("page4.va6",TRzon[0]);
```

Figura 47. Función para cambiar valores de la pantalla.

Para mandar comandos se utiliza la función `sendComand("")`, y entre las comillas se pone el comando que se quiere mandar, como se ve en la Figura 48. Hay que decir que los comandos que se pueden mandar son los propios de la pantalla, no comando en C o C++.

```
nexPant.sendCommand("page 13");
```

Figura 48. Función para enviar comandos a la pantalla



#### 4.3. Comunicación con el lector RFID

La comunicación con el lector de radiofrecuencia se realiza mediante I<sup>2</sup>C utilizando el mismo bus que para la comunicación que entre Arduinos, las funciones que se utilizan pertenecen a la librería **MFRC522.h**, de la que más adelante se hablará. Lo primero que debe hacerse es crear un objeto de la clase **MFRC522** para utilizar las funciones de esta clase y después se le pasa un clave de encriptación a la librería que más tarde se rellenará, como se ve en la Figura 49.

```
MFRC522 rfid(SS_Pin,RST_Pin);  
MFRC522::MIFARE_Key key;
```

Figura 49 . Configuración para la comunicación con el lector RFID.

El siguiente paso es iniciar la comunicación con el lector de tarjetas, como se ve en la Figura 50.

```
rfid.PCD_Init();
```

Figura 50. Función de inicio de comunicación con el lector RFID

En este momento ya es posible comunicarse con el lector RFID y se hará mediante dos funciones. La primera es una función para detectar si hay una tarjeta de radiofrecuencia cerca del lector, función **PICC\_IsNewCardPresent()** que devuelve un 1 si hay una tarjeta y un cero si no la hay. La segunda función le los datos de dicha tarjeta y los guarda en variables para la posterior utilización. Todo ello se ve en la Figura 51

```
rfid.PICC_IsNewCardPresent();  
rfid.PICC_ReadCardSerial();
```

Figura 51. Funciones de detección y lectura de una etiqueta.

#### 4.4. Programación de la pantalla

En capítulo cinco se habló de la pantalla táctil y de la forma de programación a través del programa *Nextion Editor*. Este programa permite desarrollar un entorno gráfico de forma sencilla, como se ve en la Figura 52. Como se puede ver el editor de Nextion está dividido en siete zonas: la primera es la barra de herramientas donde se elige poner elementos como botones, barras de texto...; la segunda es la zona para cargar las imágenes y las fuentes de texto que se van a utilizar en la pantalla; la tercera es donde se muestran los resultados de compilación de la pantalla; la cuarta es la zona de programación de los objetos que se han puesto en la pantalla; la quinta es la zona de características de cada objeto; la sexta es la zona de navegación entre las distintas pantallas creadas; y la séptima es donde se configura todo el entorno gráfico.

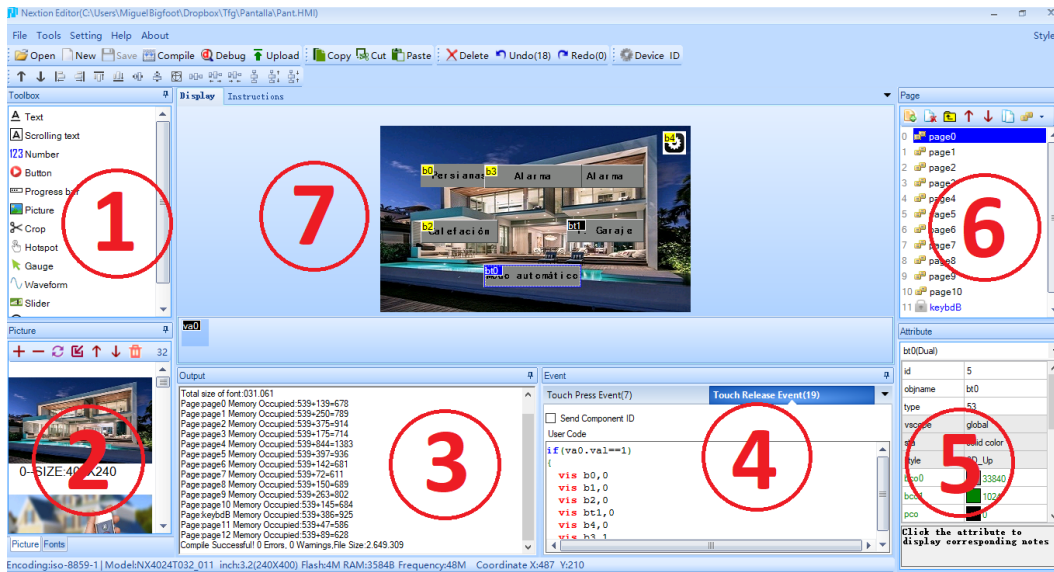


Figura 52. Programa Nextion Editor con zonas numeradas.

#### 4.5. Programación general del Arduino Yún

En este apartado se hablará de la programación del `loop()`, es decir, de la ejecución cíclica del programa. El `loop` de este micro se muestra en la Figura 53, donde se llaman a todas las funciones del programa.

```

void loop() {
  mandarAlarma(); //Función de activación de la alarma
  infor(); //Funcion para pedir información al esclavo
  actualizar(); //Función para actualizamos la información de la pantalla
  mandarCalefac(); //Funcion para mandar la información de la calefaccion al
  //arduino esclavo
  lector_RFID(); //Función para leer el lector de tarjetas
  boolean pgaraje=nexPant.getComponentValue("page0.bt1"); //Función para abrir
  if (pgaraje==1) { //la puerta del garaje
    digitalWrite(PinG[0],HIGH); //desde la pantalla
    digitalWrite(PinG[1],LOW);
    int finalcarrera=0;
    while (finalcarrera!=1) finalcarrera=digitalRead(fc);
    finalcarrera=0;
    digitalWrite(PinG[0],LOW);
    digitalWrite(PinG[1],LOW);
    delay(10000);
    digitalWrite(PinG[0],HIGH);
    digitalWrite(PinG[1],HIGH);
    while (finalcarrera!=1) finalcarrera=digitalRead(fc);
    finalcarrera=0;
    digitalWrite(PinG[0],LOW);
    digitalWrite(PinG[1],LOW);
    nexPant.setComponentValue("page0.bt1",0);
  }
  modAutoma(); //Funcion para el modo automatico
  mandarPers(); //Funcion para mandar información de las
  //de las persianas
}

```

Figura 53. Loop del Arduino Yún.

Las funciones están colocadas por orden de importancia, por ello la primera es la de activación de la alarma, después van las de obtener la información de los sensores de temperatura, actualización de la pantalla, control de la calefacción, modo automático y persianas. El control de la puerta del garaje a diferencia de las otras no se realiza en una función sino en el loop. Cada función realiza las tareas oportunas para cada sistema.

#### 4.6. Programación general del Arduino Mega

Al igual que en el apartado anterior la programación general es la del `loop()`, en el caso de este microcontrolador sigue las ordenes que le llegan desde el Arduino Yún. Como se puede ver en la Figura 54 en el loop se comprueba continuamente si el maestro ha activado algún sistema para entrar en las funciones que controlan ese sistema. La única función que se ejecuta siempre es la de lectura de los sensores para que el Arduino Yún tenga la información siempre que la pida.

```
void loop() {
  if (AlarState==1){
    alarm(n);
  }
  Leer_Temp();
  if (CalState==1){
    digitalWrite(PCal,HIGH);
    Calef();
  }else digitalWrite(PCal,LOW);
  if (PerState==1){
    persact();
    PerState=0;
  }
}
```

Figura 54. Loop del Arduino Mega

#### 4.7. Programación de los motores, interrupciones y sensores de temperatura y luminosidad

Este apartado trata de la programación de algunos sensores y motores, así como peculiaridades del microcontrolador como son las interrupciones.

##### 4.7.1. Motores

Los motores que se van a utilizar tanto en las persianas como en la puerta del garaje son motores de alterna, es decir, que para cambiar el sentido de giro hay que cambiar la fase y el neutro de posición, además no siempre pueden estar conectados a la red porque si no girarían continuamente, por ello se han instalado finales de carrera que avisen cuando debe terminar el movimiento

del motor. La conexión de los finales de carrera al Arduino es directa y se conectan los dos de cada persiana o puerta de garaje en una misma entrada digital.

El acoplamiento de los finales de carrera con el Arduino se realiza a través de relés, cuyo circuito de control lo activamos con una salida digital. Cada motor necesita tres relés: uno que actúa de interruptor, deja pasar o no la corriente, y otros dos para cambiar la fase y el neutro del motor.

Por ello la programación de cualquier motor es como la que se ve en la Figura 55, que en concreto, es de la puerta del garaje -que primero activa el movimiento en un sentido, cuando se activa el final de carrera para durante diez segundos, se mueve en sentido contrario y cuando llega al final se para.

```
digitalWrite(PinG[0],HIGH);
digitalWrite(PinG[1],LOW);
int finalcarrera=0;
while (finalcarrera!=1) finalcarrera=digitalRead(fc);
finalcarrera=0;
digitalWrite(PinG[0],LOW);
digitalWrite(PinG[1],LOW);
delay(10000);
digitalWrite(PinG[0],HIGH);
digitalWrite(PinG[1],HIGH);
while (finalcarrera!=1) finalcarrera=digitalRead(fc);
finalcarrera=0;
digitalWrite(PinG[0],LOW);
digitalWrite(PinG[1],LOW);
```

Figura 55. Programación del movimiento de la puerta del garaje.

#### 4.7.2. Sensores de temperatura y luminosidad

El sensor de temperatura está conectado a una entrada analógica que da al programa un valor entre 0 y 1023 dependiendo del voltaje que llega a la entrada, donde cero son 0 V y 1023 son 5 V. Utilizando la Figura 27 y los datos anteriores se puede hallar la ecuación que nos da como resultado la temperatura en grados centígrados.

```
TRzon[i]=analogRead(i);
TRzon[i]={(TRzon[i]*5/1023)-0.52}/0.0096;
```

Figura 56. Lectura de un sensor de temperatura TMP36.

Como se ve en la Figura 56, primero se lee la entrada analógica y luego ya se transforma a grados centígrados y así se obtiene el dato necesario.

El sensor de luminosidad habría que probarlo en cada caso, porque todos son distintos, y ver que valores de tensión ofrece, después ya se haría un ajuste fino. La utilización de sensor de luminosidad para las persianas quedaría como

en la Figura 57, donde *lux* es la medida de la luminosidad y si los dos están por debajo de un cierto valor se bajan las persianas y si están por encima se suben.

```
if (lux[0]<300&&lux[1]<300) {
  nexPant.setComponentValue("page1.b3",1);
  nexPant.setComponentValue("page2.b3",1);
}else if (lux[0]>900&&lux[1]>900) {
  nexPant.setComponentValue("page1.b2",1);
  nexPant.setComponentValue("page2.b2",1);
}
```

Figura 57. Programa para utilizar la luminosidad.

#### 4.7.3. Interrupciones

En el programa se utilizan dos tipos de interrupciones. El primer tipo de interrupción es externa, es decir, cuando cambia el valor de una entrada se ejecuta una función, mediante ese método se pueden programar instrucciones para que se ejecuten solo ante un evento especial, en el caso de la vivienda domótica de este trabajo se utiliza para la alarma.

```
attachInterrupt(digitalPinToInterrupt(interruptPin), alarma, RISING);
```

Figura 58. Interrupción del Arduino Yún.

En la Figura 58 se observa la única instrucción que hace falta para el uso de una interrupción, el primer valor de la función es el pin en que se va a producir la interrupción, no todos los pines del microcontrolador pueden ser utilizados. El segundo elemento es la función a la que se llama cuando se produce una interrupción y el tercero es la forma en la que se va a producir la interrupción, hay cuatro maneras: **low** se activa la interrupción cuando tiene a la entrada un nivel bajo; **change** se dispara la interrupción cuando es cambia de valor la entrada; **rising** se ejecuta la función cuando pasa de un valor bajo a un valor alto, es el que se utiliza en esta programación; y **falling** se dispara la interrupción cuando pasa de un nivel alto a bajo la entrada.

El segundo tipo de interrupción es el **timer** o interrupción interna, que se dispara periódicamente. El periodo de tiempo entre cada disparo se puede elegir y se calcula utilizando el reloj del microcontrolador, cuanto mayor sea el tiempo de disparo menos preciso es. Para el control del timer se han utilizado las tres funciones de la Figura 59.

```
Timer1.initialize(10000000);
Timer1.attachInterrupt(margenOff);
Timer1.detachInterrupt();
```

Figura 59. Funciones para los Timers.

La primera función se utiliza para inicializar el *timer* e introducir el periodo entre interrupciones, en este caso 10 segundos; la segunda función sirve para activar el *timer* y asignar la función que debe ejecutar; y la tercera función desactiva el *timer*.

#### 4.8. Librerías usadas para los programas

Las librerías son colecciones de funciones y código específicas para un componente que facilitan su manejo. En este trabajo se han utilizado las librerías de la Figura 60 y, que como se muestra, se incluyen en el programa.

```
#include <MFRC522.h>
#include <TimerOne.h>
#include <SoftwareSerial.h>
#include <Nextion.h>
#include <Wire.h>
```

Figura 60. Librerías utilizadas

En este apartado no se va a entrar en un análisis detallado, solo se va a nombrar y explicar para que componente se utilizan. La librería **MFRC522.h** se utiliza para el control y la comunicación del Arduino con el lector RFID.

La librería **TimerOne.h** controla los *Timers* de los que antes se ha hablado. **SoftwareSerial.h** es una librería desarrollada por Arduino para el control de la comunicación por el puerto serie.

La siguiente librería es **Nextion.h** que deja controlar al Arduino la pantalla. Por último, está la librería **Wire.h** que también es de Arduino y que es utilizada para la comunicación mediante I<sup>2</sup>C.

## 5. Conexión hardware con los Arduinos

La conexión de los distintos sistemas con los Arduino es muy importante para el correcto funcionamiento de la casa domótica, por ello algunas partes deben especificarse y dar una visión general de cómo se realizarán las conexiones.

### 5.1. Conexión entre los dos Arduinos

La conexión que debe realizarse entre los Arduinos es la necesaria para soportar la comunicación I<sup>2</sup>C, es decir, un cable para la señal de reloj (SCL), otro para los datos (SDA) y una masa común. Como se ve la Figura 61 el cable amarillo SCL, el azul SDA y el negro es la masa.

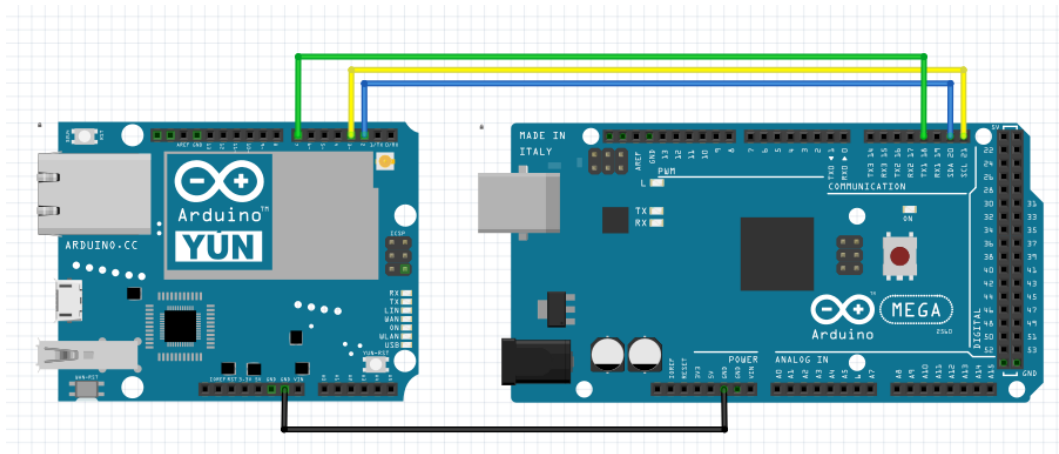


Figura 61. Esquema conexión entre Arduinos.

Además de la comunicación entre los Arduinos se conecta también cable verde para que salte una interrupción en el Arduino Yún cuando alguno de los sensores de movimiento del Arduino Mega detecta una intrusión. Tanto la comunicación I<sup>2</sup>C como el pin de la interrupción son pines digitales.

### 5.2. Conexiones del Arduino Yún

En el apartado se veía la conexión para la interrupción de la alarma y la comunicación con el Arduino Mega, al mismo tiempo controla la puerta del garaje, los zumbadores de la alarma, los sensores de luminosidad y la pantalla táctil.

#### 5.2.1. Conexión de la puerta del garaje

La puerta del garaje es controlada por el lector RFID, a través de I<sup>2</sup>C, los relés que controlan el motor y el final de carrera. Como se ve en la Figura 62, el lector RFID se conecta a través del bus I<sup>2</sup>C que está en los mismos pines que en la Figura 61. El final de carrera está conectado a una resistencia de pull down y a una entrada digital del Arduino para su control.

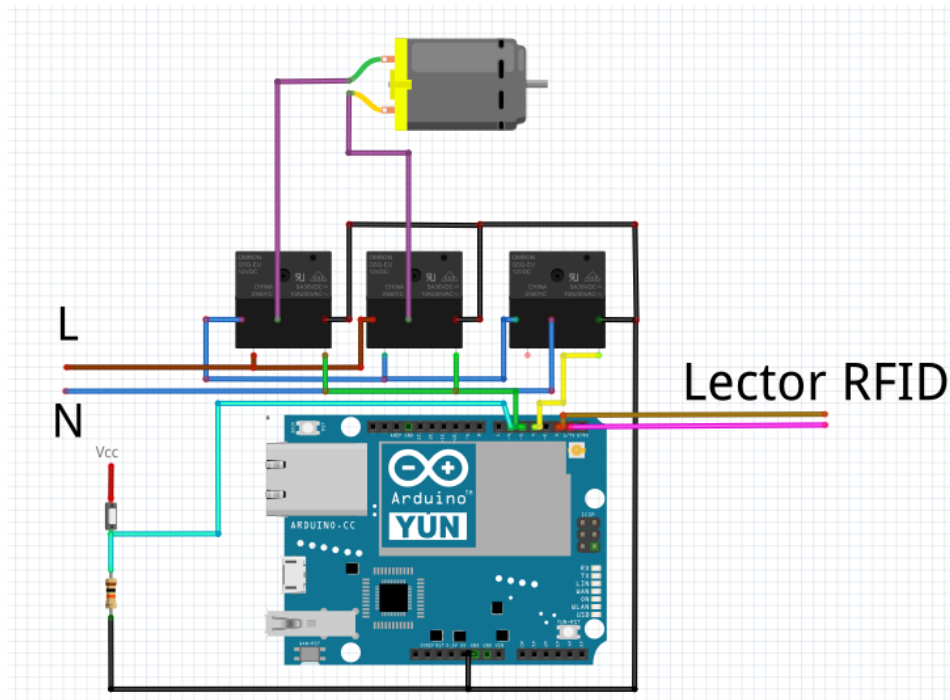


Figura 62. Conexión puerta del garaje.

El control de los motores se hace mediante tres relés conectados a dos salidas digitales, el primero controla el suministro de electricidad y los otros dos el sentido de giro del motor, como se muestra más claramente en el esquema de la Figura 63.

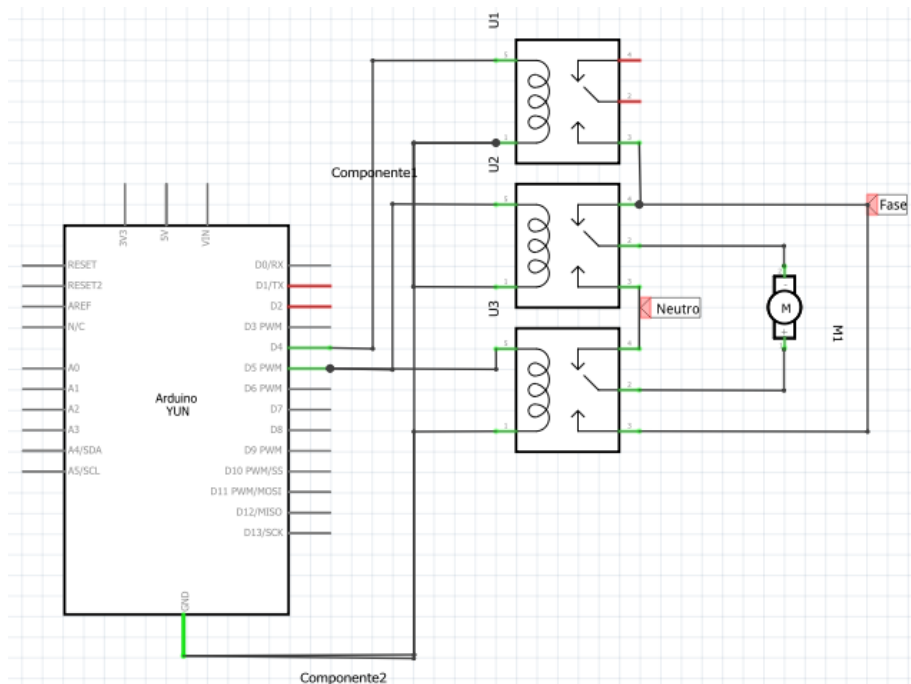


Figura 63. Esquema de la conexión de los relés



### 5.2.2. Conexión de la pantalla táctil

La pantalla táctil solamente necesita la conexión al microcontrolador a través del puerto serie, es decir, con los cables rx y tx como se ve en la Figura 64, aunque la pantalla de la figura no es la misma que la que se utiliza para el sistema domótico del trabajo, pero sirve para ilustrar las conexiones.

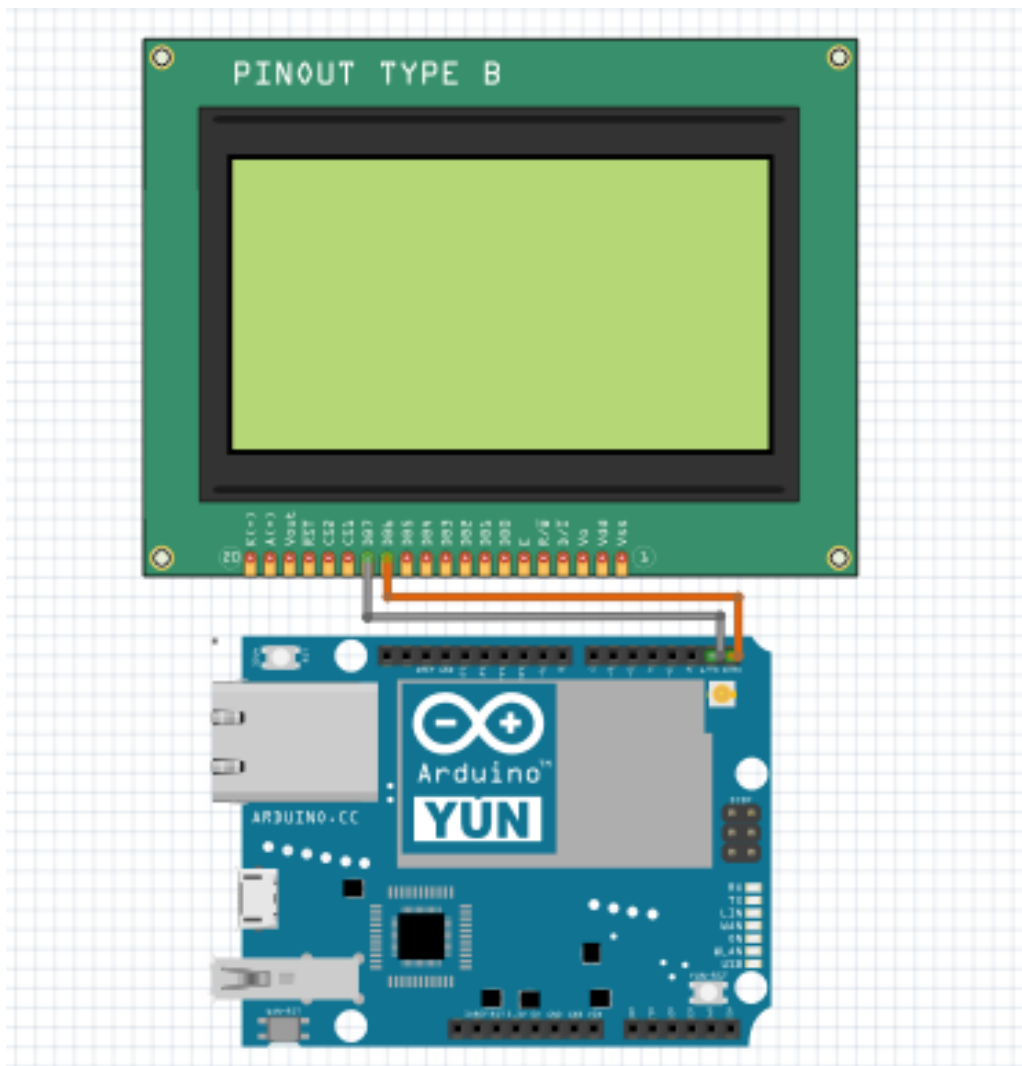


Figura 64. Conexión a la pantalla táctil.

### 5.2.3. Conexión con los zumbadores activos

Los zumbadores que se utilizan en este trabajo son activos, por tanto, el único control que tiene que haber sobre ellos es una salida digital, que los active y desactive cuando sea necesario, además tienen que ir conectados a masa y a cinco voltios, como se ve en la Figura 65.

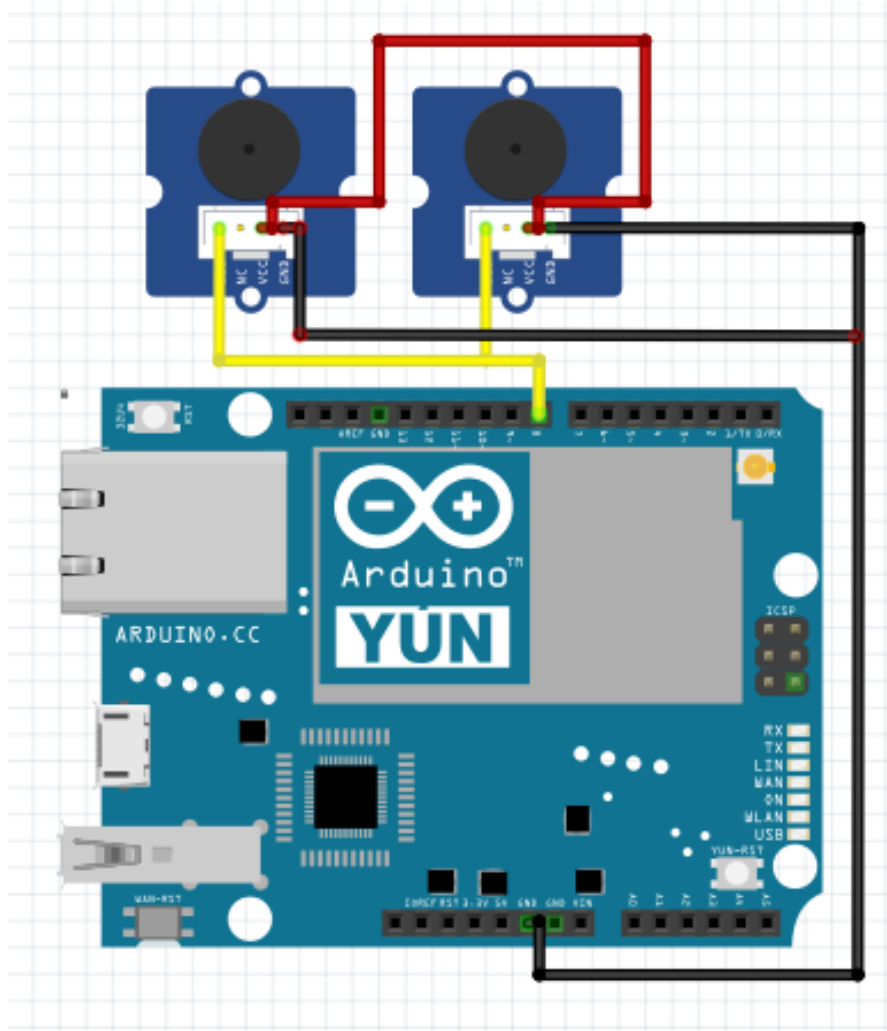


Figura 65. Conexión con los buzzers activos.

#### 5.2.4. Conexión de las LDRs

Las fotorresistencias o LDR actúan como sensor de luminosidad para el modo automático, las LDRs se instalarán en dos fachadas opuestas de la casa, para comparar las mediciones y tener un valor más real de la luminosidad ambiente. Las fotorresistencias deben conectarse junto con una resistencia a una entrada analógica, como se ve en la Figura 66.

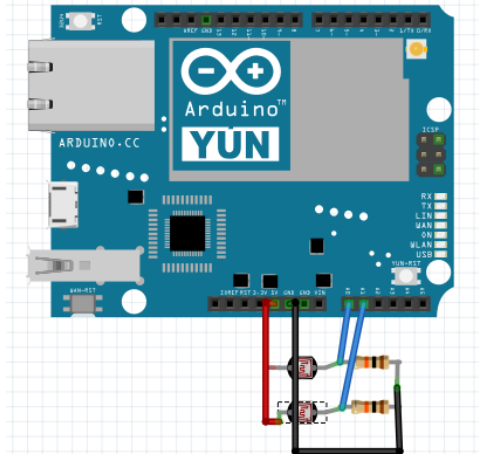


Figura 66. Conexión de las fotorresistencias.

La resistencia que se instala con la fotorresistencia sirve para tener un divisor de tensión y así el valor del voltaje de entrada varía con la LDR pues la otra resistencia es fija.

#### 5.3. Conexiones del Arduino Mega

El Arduino Mega controla los sensores de la alarma, el sistema de calefacción y las persianas de la casa, asimismo se comunica con el Arduino Yún mediante I<sup>2</sup>C y el pin de la interrupción, como ya se vio en la Figura 61.

Las conexiones para el control de las persianas son iguales que las del motor de la puerta del garaje que se ve en la Figura 62, por lo que queda de explicar las conexiones de los sensores de movimiento, los sensores temperatura y las electroválvulas.

### 5.3.1. Conexión de los sensores de movimiento

Los sensores de movimiento HC-SR501 requieren la conexión a una fuente de alimentación y a una entrada digital cada uno, como se muestra en la Figura 67.

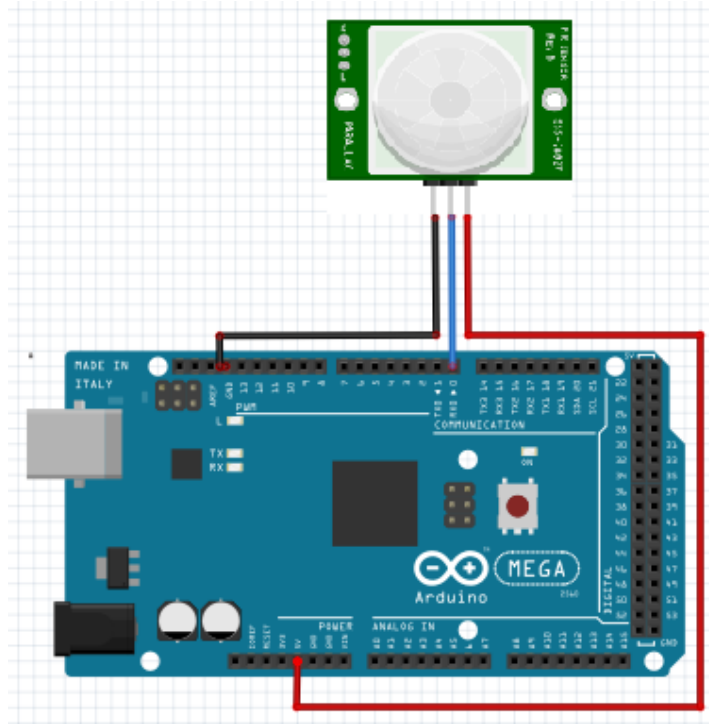


Figura 67. Conexión a un sensor de HC-SR501.

### 5.3.2. Conexión de los sensores de temperatura

Los sensores de temperatura TMP36 necesitan una entrada analógica a la que son conectados directamente (Figura 68), además también necesitan conectarse a una fuente de alimentación.

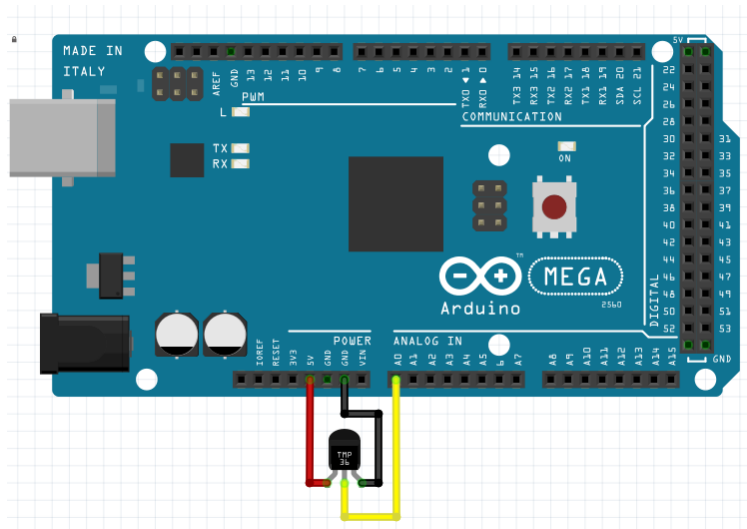


Figura 68. Conexión de un sensor del sensor TMP36.

### 5.3.3. Conexión de las electroválvulas.

El sistema de calefacción, como ya se ha dicho, está dividido en zonas, que se calientan cuando una electroválvula deja pasar agua caliente para el suelo radiante. El control de la electroválvula se realiza a través de un relé conectado a una salida digital, como se puede observar en la Figura 69.

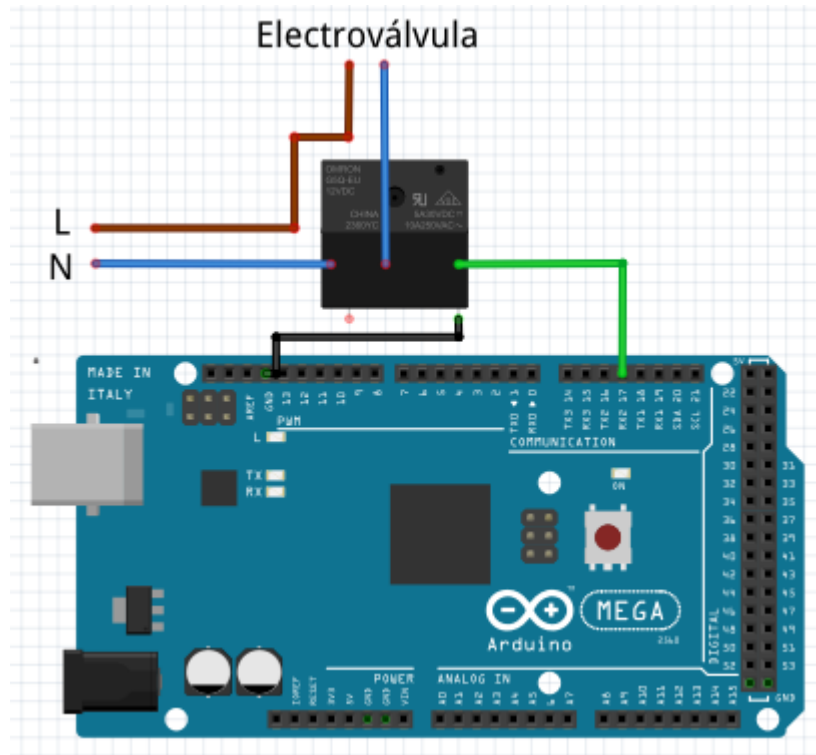


Figura 69. Conexión de la electroválvula.



## Conclusiones y extensiones

La primera parte de este trabajo habló sobre los objetivos que se querían cumplir, una vez desarrollado el trabajo se puede hablar de las conclusiones, que se han sacado de este trabajo y de su realización.

Una de las primeras conclusiones a las que se llega es que la automatización de una vivienda es complicada y se pueden implementar multitud de sistemas que faciliten y vuelvan más cómoda la vida de los ocupantes de la casa. Cuantos más sistemas se instalen más difícil es control óptimo de todos los sistemas y la interrelación entre ellos.

La comunicación en la domótica es un aspecto fundamental desde la comunicación de los señores y actuadores con el controlador de la vivienda, hasta la comunicación del usuario con todo el sistema.

Los microcontroladores de Arduino, como se ha visto en el desarrollo del trabajo, sirven perfectamente para poder controlar una casa domótica, aunque este trabajo solo se centra en la gestión y diseño, habría que realizar un prototipo para poder afirmar esto con seguridad. Las ventajas de la utilización de Arduino son el fácil entorno de programación, multitud de librerías que facilitan el control de sensores, pantallas..., precio bajo en comparación con otros microcontroladores y gran oferta de microcontroladores en función de las necesidades del proyecto, es decir, necesidades de E/S, comunicación ethernet o wifi... La principal desventaja de utilizar Arduino para este trabajo es que el número de entradas y salidas es limitado y por eso se han tenido que utilizar dos microcontroladores.

En el principio del trabajo se marcó como objetivo general fue la gestión domótica de una casa unifamiliar basada en Arduino que como se puede ver en anteriores capítulos se ha cumplido pues todos los sistemas están controlados por los microcontroladores Arduino. Los objetivos específicos eran el control sobre los sistemas de calefacción, alarma anti-intrusiones, persianas y apertura de la puerta del garaje, todos ellos se han llevado a cabo para su implementación y buen funcionamiento dentro de la casa domótica.

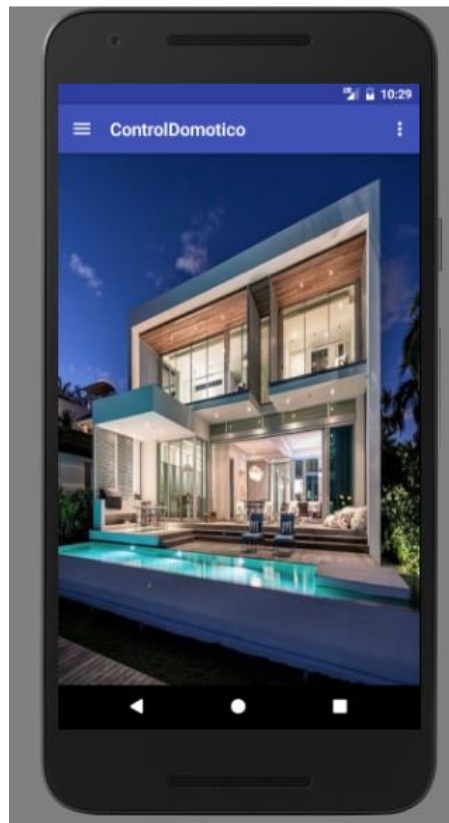
Además el usuario debía disponer de una interfaz donde controlar los sistemas y poder programar la automatización de las persianas, la alarma o utilizar diferentes modos de funcionamiento de la casa domótica, llevándose a cabo a través de la pantalla táctil Nextion.

El último objetivo específico marcado y cumplido fue la preparación para poder instalar nuevos sistemas domóticos, que se ha llevado a cabo a través de dos aspectos fundamentales: el primero de ellos es que los microcontroladores siguen teniendo entradas y salidas disponibles para nuevos sistemas; y el

segundo y más potente es que a través del bus I<sup>2</sup>C se pueden conectar nuevos microcontroladores que controlen nuevos sistemas domóticos y nuevos sensores o actuadores que utilicen este protocolo de información.

La variedad de sistemas que se pueden integrar dentro de una casa domótica es muy grande por lo que cualquiera de ellos sería una extensión de este trabajo. Sistemas como el control de las luces, cerraduras electrónicas, control de incendios, humos, sustancias nocivas en el ambiente, control de riego... son claros ejemplos de sistemas que se podrían instalar una casa domótica.

A parte de estos sistemas una interesante extensión de este proyecto sería realizar una aplicación para Android desde la cual se pudiera manejar la casa a distancia, esto aportaría manejabilidad y flexibilidad al sistema y aumentaría en control, comodidad y seguridad del usuario. La aplicación podría ser como la de la Figura 70, en la que se muestra la pantalla principal donde se podrían ver mensajes programados.



*Figura 70. Pantalla principal aplicación Android ControlDomotico.*



La aplicación tiene también un menú de navegación (☰) para que lleva a diferentes pantallas para controlar los diferentes sistemas de la casa.

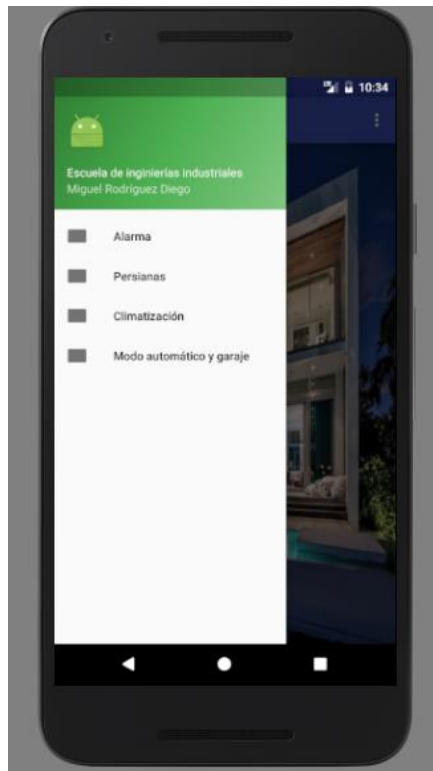


Figura 71. Pantalla de navegación aplicación Android ControlDomotico.

En la Figura 72 se pueden ver las diferentes pantallas para el control de todos los sistemas de la casa domótica de este proyecto.



Figura 72. Pantallas de alarma, persianas, calefacción y garaje (de izquierda a derecha) de la aplicación Android ControlDomotico.



## Bibliografía

- API Guides, Android Developer.* (04 de Septiembre de 2017). Obtenido de Android Developer Web site: <https://developer.android.com/guide/platform/index.html>
- Aprendiendo Arduino.* (01 de 09 de 2017). Obtenido de WordPress web Site: <https://aprendiendoarduino.wordpress.com/2014/11/18/tema-6-comunicaciones-con-arduino-4/>
- Aprendiendo Arduino.* (30 de Agosto de 2017). Obtenido de Arduino Web site: <https://www.arduino.cc/en/Guide/Introduction?setlang=en#>
- Arduino Yun, Arduino.* (07 de Septiembre de 2017). Obtenido de Arduino Web site: <https://store.arduino.cc/arduino-yun>
- Caraterísticas, Android OS.* (04 de Septiembre de 2017). Obtenido de Android OS Web site: <http://androidos.readthedocs.io/en/latest/data/caracteristicas/>
- CEDOM. (2008). *Inicio, Idae.* Obtenido de Idae: [http://www.idae.es/uploads/documentos/documentos\\_11187\\_domotica\\_en\\_su\\_vivienda\\_08\\_3d3614fe.pdf](http://www.idae.es/uploads/documentos/documentos_11187_domotica_en_su_vivienda_08_3d3614fe.pdf)
- CEDOM. (14 de Junio de 2017). *Sobre domótica, CEDOM.* Obtenido de CEDOM | Asociación Española de Domótica e Inmótica: <http://www.cedom.es/sobre-domotica/que-es-domotica>
- E. P. (1 de Marzo de 2007). *ISA, (Ingeniería de Sistemas y Automática).* Obtenido de Universidad de Oviedo: <http://isa.uniovi.es/docencia/AutomEdificios/transparencias/sensores.pdf>
- E. P. (1 de Marzo de 2007). *ISA, (Ingeniería de Sistemas y Automática).* Obtenido de Universidad de Oviedo: <http://isa.uniovi.es/docencia/AutomEdificios/transparencias/actuadores.pdf>
- Novel, B. (07 de Septiembre de 2013). *Sobre domótica, CEDOM.* Obtenido de CEDOM: <http://www.cedom.es/sobre-domotica/normativa-domotica-e-inmotica/tipos-de-sistemas-domoticos-y-organismos-de-normalizacion>
- pi.gate.ac.uk. (5 de Septiembre de 2017). Obtenido de <https://pi.gate.ac.uk/pages/airpi-files/PD0001.pdf>

Rodríguez, A. (Agosto de 2014). *Repositorio Documental, Universidad de Valladolid*. Obtenido de Universidad de Valladolid:  
<http://uvadoc.uva.es/handle/10324/12922>

Ruiz González, J. M. (2016). Apuntes Instrumentación Electrónica. Tema 2.- Transductores. Valladolid, España.

Turiel, J. (2016). Apuntes Control y Comunicaciones Industriales. Arquitecturas de Red - Redes de área local. Valladolid, España.

Universidad Miguel Hernández. (2007). *Ingeniería de Sistemas y Automática, Universidad Miguel Hernández*. Obtenido de Universidad Miguel Hernández:  
[http://isa.umh.es/asignaturas/sea/teoria\\_2C\\_curso0708/tema2\\_intro\\_micro.pdf](http://isa.umh.es/asignaturas/sea/teoria_2C_curso0708/tema2_intro_micro.pdf)

## Anexos

### Anexo I. Código Arduino Yún

```
/*Programa Master para el Arduino Yún que se comunica con el
Arduino Mega y le da las ordenes
necesarias para el correcto funcionamiento del sistema, ademas
tambien controlala pantalla
Nextion, la puerta del garaje y el lector RFID, y los
zumbadores*/

#include <MFRC522.h>           //Libreria para el lector RFID
#include <TimerOne.h>         //Libreria para el Timer
#include <SoftwareSerial.h>   //Libreria para la comunicacion por
el puerto serie
#include <Nextion.h>          //Libreria para la pantalla
#include <Wire.h>             //Libreria para la comunicación I2C

/*Declaración de las variables necesarias para el programa*/
//Variables para temperatura
int Tzon[]={20,20,20,20,20,20}; //Temperatura deseada en la
zona
short TRzon[]={19,19,19,19,19,19}; //Temperatura real medida por
los sensores
int CalState=0;              //Estado de la caldera

//Variables para la alarma
int AlarState=0;            //Estado de la alarma
int idSen;
int pinBuzzers;            //Un pin para los dos zumbadores

//variables para las persianas
int Nper[11];               //Estado deseado de la persiana
int PerState=0;             //Conectado o no el sistema de persianas.
int pLdr[2];

//Variables de la puerta del garaje
int PinG[2];               //Pines de los relés que controlan el motor de la
pta. del garaje
int fc;                     //Un un pin para los dos finales de carrera

//Pin de interrupcion
int interruptPin=7;

//definimos los pines de la comunicacion I2C
#define SS_Pin 2
#define RST_Pin 3

/*Configuracion para la comunicación con la pantalla y el lector*/
//Configuracion del lector RFID
MFRC522 rfid(SS_Pin,RST_Pin);
MFRC522::MIFARE_Key key;

//Definimos las tarjetas autorizadas para la apertura de la puerta
del garaje
#define Num_tarj 2
byte tarj_autoriza[Num_tarj][6]={
    {0x02,0x83,0xFF,0x54,0xF9,0x35,}
```

```

    ,{0x5D,0x7E,0x3F,0x72,0xA1,0x8B,}
    //...
};

//Se crea un objeto de la clase Nextion para el control de la
pantalla
SoftwareSerial nextion(0,1);
Nextion nexPant(nextion,9600);

/*Configuración de los pines, interrupciones e inicialización de
las comunicaciones*/
void setup() {
    pinMode(pinBuzzers,OUTPUT);
    digitalWrite(pinBuzzers,LOW);

    Timer1.initialize(10000000); //Timer de 10 segundos para que
salte la alarma

    nexPant.init();

attachInterrupt(digitalPinToInterrupt(interruptPin),alarma,RISING)
; //Configuración de la interrupción externa

    Wire.begin(); //Empezamos la comunicación I2C
como maestro
    rfid.PCD_Init(); //iniciamos la comunicación con
el lector de tarjetas
    for(byte i=0;i<6;i++) key.keyByte[i]=0xFF; //Establecemos la
clave de encriptación
}

void loop() {
    mandarAlarm(); //Función de activación de la alarma
    infor(); //Función para pedir información al esclavo
    actualizar(); //Función para actualizamos la información de la
pantalla
    mandarCalefac(); //Función para mandar la información de la
calefacción al
//arduino esclavo
    lector_RFID(); //Función para leer el lector de tarjetas
    boolean pgaraje=nexPant.getComponentValue("page0.bt1");
//Función para abrir
    if (pgaraje==1){ //la
puerta del garaje
        digitalWrite(PinG[0],HIGH); //desde
la pantalla
        digitalWrite(PinG[1],LOW);
        int finalcarrera=0;
        while (finalcarrera!=1) finalcarrera=digitalRead(fc);
        finalcarrera=0;
        digitalWrite(PinG[0],LOW);
        digitalWrite(PinG[1],LOW);
        delay(10000);
        digitalWrite(PinG[0],HIGH);
        digitalWrite(PinG[1],HIGH);
        while (finalcarrera!=1) finalcarrera=digitalRead(fc);

```

```

    finalcarrera=0;
    digitalWrite (PinG[0],LOW);
    digitalWrite (PinG[1],LOW);
    nexPant.setComponentValue ("page0.bt1",0);
}
modAutoma(); //Funcion para el modo automatico1
mandarPers(); //Funcion para mandar información de las
               //de las persianas
}

//Mandar información de alarma
void mandarAlarm(){
    boolean alartot=nexPant.getComponentValue ("page6.bt1");
    boolean alarpar=nexPant.getComponentValue ("page6.bt0");
    int n;
    if(alarpar==1){
        AlarState=1;
        n=0;
    }else if(alartot=1){
        AlarState=1;
        n=1;
    }else{
        AlarState=0;
        digitalWrite (pinBuzzers,LOW);
    }
    Wire.beginTransaction(1);
    Wire.write (AlarState);
    Wire.write (n);
    Wire.endTransmission(1);
}

//Coge la informacion de los sensores de temperatura
void infor(){
    Wire.requestFrom(1,8);
    while(Wire.available()){
        for (int i=0;i<6;i++) TRzon[i]=Wire.read();
    }
}

//Actualizar los valores de la panatalla
void actualizar(){
    nexPant.setComponentValue ("page4.va6",TRzon[0]);
    nexPant.setComponentValue ("page4.va7",TRzon[1]);
    nexPant.setComponentValue ("page4.va8",TRzon[2]);
    nexPant.setComponentValue ("page4.va9",TRzon[3]);
    nexPant.setComponentValue ("page4.va10",TRzon[4]);
    nexPant.setComponentValue ("page4.va11",TRzon[5]);
}

//Mandar información de la calefacción
void mandarCalefac(){
    CalState=nexPant.getComponentValue ("page4.bt0");
    Tzon[0]=nexPant.getComponentValue ("page4.va0");
    Tzon[1]=nexPant.getComponentValue ("page4.va1");
    Tzon[2]=nexPant.getComponentValue ("page4.va2");
    Tzon[3]=nexPant.getComponentValue ("page4.va3");
    Tzon[4]=nexPant.getComponentValue ("page4.va4");
    Tzon[5]=nexPant.getComponentValue ("page4.va5");
    Wire.beginTransaction(1);
}

```

```

Wire.write(CalState);
if(CalState==1){
    for(int i=0;i<6;i++) Wire.write(Tzon[i]);
}
Wire.endTransmission(1);
}

//Lector RFID
void lector_RFID(){
    if(rfid.PICC_IsNewCardPresent()){
        rfid.PICC_ReadCardSerial();
        byte id=rfid.uid.uidByte;
        for (int i=0;i<Num_tarj;i++){
            if(id==tarj_autoriza[i]){
                nexPant.setComponentValue("page0.bt0",1);
                break;
            }
        }
    }
}

//Modo automático
void modAutoma(){
    boolean autom=nexPant.getComponentValue("page0.bt0");
    if (autom==1){
        boolean persic=nexPant.getComponentValue("page10.bt0");
        if (persic==1){
            int lux[2];
            for(int i=0;i<2;i++) lux[i]=analogRead(pLdr[i]);
            if (lux[0]<300&&lux[1]<300){ //Baja o sube las persianas en
funcion de la luz que haya
                nexPant.setComponentValue("page1.b3",1);
                nexPant.setComponentValue("page2.b3",1);
            }else if (lux[0]>900&&lux[1]>900){
                nexPant.setComponentValue("page1.b2",1);
                nexPant.setComponentValue("page2.b2",1);
            }
        }
    }
}

//Mandar informaion de las persianas
void mandarPers(){
    Nper[0]=nexPant.getComponentValue("page1.bt0");
    Nper[1]=nexPant.getComponentValue("page1.bt1");
    Nper[2]=nexPant.getComponentValue("page1.bt2");
    Nper[3]=nexPant.getComponentValue("page1.bt3");
    Nper[4]=nexPant.getComponentValue("page1.bt4");
    Nper[5]=nexPant.getComponentValue("page2.bt0");
    Nper[6]=nexPant.getComponentValue("page2.bt1");
    Nper[7]=nexPant.getComponentValue("page2.bt2");
    Nper[8]=nexPant.getComponentValue("page2.bt3");
    Nper[9]=nexPant.getComponentValue("page2.bt4");
    Nper[10]=nexPant.getComponentValue("page2.bt5");
    Wire.beginTransmission(1);
    for(int i=0;i<11;i++) Wire.write(Nper[i]);
    Wire.endTransmission(1);
}

//Función de la interrupcion de la alarma

```



```

void alarma(){
    Wire.requestFrom(1,8);
    Timer1.attachInterrupt(margenOff);
    while(Wire.available()) idSen=Wire.read(); //Falta revisar que
pasa con el segundo sensor
}

/*Funcion de timer para la alarma*/
void margenOff(){
    boolean alartot=nexPant.getComponentValue("page6.bt1");
    boolean alarpar=nexPant.getComponentValue("page6.bt0");
    if (alartot==1||alarpar==1){
        digitalWrite(pinBuzzers,LOW); //Activa los
zumbadores
        if (alartot==1){
            Wire.beginTransaction(1);
            for(int i=0;i<11;i++){
                Nper[i]=1;
                Wire.write(Nper[i]);
            }
            Wire.endTransmission(1);
            nexPant.sendCommand("page 13");
            if (idSen==0||idSen==1)
nexPant.setComponentText("page13.t1","Intrusión por el garaje");
            if (idSen==2)
nexPant.setComponentText("page13.t1","Intrusión por la cocina");
            if (idSen==3)
nexPant.setComponentText("page13.t1","Intrusión por la puerta
principal");
            if (idSen==4)
nexPant.setComponentText("page13.t1","Intrusión por el salón");
            if (idSen==5)
nexPant.setComponentText("page13.t1","Intrusión por las escaleras
o el baño");
            if (idSen==6)
nexPant.setComponentText("page13.t1","Intrusión por la habitación
de los niños");
            if (idSen==7)
nexPant.setComponentText("page13.t1","Intrusión por el despacho");
            if (idSen==8)
nexPant.setComponentText("page13.t1","Intrusión por las escaleras
o el baño de arriba");
            if (idSen==9)
nexPant.setComponentText("page13.t1","Intrusión por la habitación
de invitados");
            if (idSen==10)
nexPant.setComponentText("page13.t1","Intrusión por el dormitorio
principal");
        }
    }
    else Timer1.detachInterrupt();
}

```

## Anexo II. Código Arduino Mega

```
/*Programa slave para el Arduino Mega, encargado de los sistemas
de seguridad, persianas
y calefacción, además se comunica con el Arduino Yún, que le da
instrucciones*/
#include <Wire.h> //Libreria para la comunicación I2C

//Variables para temperatura
int Tzon[]={0,0,0,0,0,0}; //Temperatura deseada en la zona
short TRzon[]={0,0,0,0,0,0}; //Temperatura real medida por los
sensores
int Pzon[]={1,2,3,4,5,6}; //Pines de las zonas de temperatura
int PCal=7; //Pin de encendido de la caldera
int CalState=0; //Estado de la caldera

//variables para la alarma
int Psen[11]; //Pines de los sensores
int AlarState=0; //Estado de la alarma
int n; //Modo de conectar la alarma (parcial o total)
int idsen[11]={0,0,0,0,0,0,0,0,0,0,0}; //Identificador de sensor
int intru=18; //Pin para activar la interrupcion
boolean intrusion=0;

//variables para las persianas
int Nper[11],Pfc[11]; //Estado deseado de la persiana
int PerState=0; //Conectado o no el sistema de persianas.
int estper[11]; //estado actual de las persianas
int Pinh[22]; //Pines de los Relés
//SoftwareSerial Temp(10,11); // Rx,Tx

//Configuración de los parametros iniciales
void setup() {
  //Configuración inicial de la temperatura
  for(int i=0;i<6;i++){ //Configuramos los pines como
salida
    pinMode(Pzon[i],OUTPUT); //y lo ponemos en estdo bajo
    digitalWrite(Pzon[i],LOW);
  }
  pinMode(PCal,OUTPUT);
  digitalWrite(PCal,LOW);

  //Configuración inicial de la alarma
  for(int i=0;i<11;i++) pinMode(Psen[i],INPUT); //Configuramos los
sensores como entrada
  pinMode(intru,OUTPUT);
  digitalWrite(intru,LOW);

  //Configuración inicial de las persianas
  for(int i=0;i<22;i++){
    pinMode(Pinh[i],OUTPUT); //Configuración de los pines que
controlan los
    digitalWrite(Pinh[i],LOW); //relés como salidas y puesta a
cero.
  }

  //Configuración de la comunicación
```

```

    Wire.begin(1); //Empezamos la comunicacion y se
pone la dirección 1
    Wire.onReceive(receiveEvent); //Función que se ejecuta cuando
recibe información
    Wire.onRequest(receiveRequest); //Función que se ejecuta cuando
el maestro pide información
}

void loop() {
    if (AlarState==1){
        alarm(n);
    }
    Leer_Temp();
    if (CalState==1){
        digitalWrite(PCal,HIGH);
        Calef();
    }else digitalWrite(PCal,LOW);
    if (PerState==1){
        persact();
        PerState=0;
    }
}

/*En esta función se leen y se guardan los valores de los sensores
de temperatura*/
void Leer_Temp(){
    for(int i=0;i<6;i++){
        TRzon[i]=analogRead(i);
        TRzon[i]=((TRzon[i]*5/1023)-0.52)/0.0096; //Convertimos el
valor medido
    } //en una
temperatura en °C
}

/*La función sirve para encender la calefacción por zonas, en
función del valor
deseado de temperatura*/
void Calef(){
    for(int i=0;i<6;i++){
        if ((TRzon[i]-0.6)<Tzon[i]){
            digitalWrite(Pzon[i],HIGH);
        }
        else if ((TRzon[i]+1)>Tzon[i]) digitalWrite(Pzon[i],LOW);
    }
}

/*Funcion para el control de las persianas*/
void persact(){
    for(int i=0;i<11;i++){
        int fc=0;
        if(Nper[i]!=estper[i]){
            if(Nper[i]==1){
                digitalWrite(Pinh[i],HIGH);
                digitalWrite(Pinh[i+11],LOW);
            }
            else{
                digitalWrite(Pinh[i],HIGH);
                digitalWrite(Pinh[i+11],HIGH);
            }
        }
    }
}

```

```

    }
    while(fc!=1){
        fc=digitalRead(i);
    }
    digitalWrite(Pinh[i],LOW);
    digitalWrite(Pinh[i+11],LOW);
    estper[i]=Nper[i];
}
}
}

/*Lectura de los sensores de la alarma*/
int alarm(int n){
    int Sen[n];
    for(int i=0;i<n;i++){
        Sen[i]=digitalRead(Psen[i]);
        if (Sen[i]==1){
            idsen[i]=1;
            intrusion=1;
            digitalWrite(intru,HIGH); //Interrupción para el
Arduino Yún
        }
        else idsen[i]=0;
    }
}

/*Comunicación para mandar información al maestro
cuando él la pide, la información enviada depende
de si la alarma ha sido activada o no*/
void receiveRequest(){
    if(intrusion==1){
        for(int i=0;i<n;i++){
            if (idsen[i]==1) {
                Wire.write(i);
                digitalWrite(intru,LOW);
                intrusion=0;
                break;
            }
        }
    }
    else for(int i=0;i<6;i++) Wire.write(TRzon[i]);
}

/*Comunicación por I2C, esta función se ejecuta cuando llega un
mensaje por I2C
dependiendo de la cantidad de mensajes guarda la información en
una variable o
en otra*/
void receiveEvent (int howmany){
    int i=0;
    switch(Wire.available()){
        case 1: //Información de la caldera
            CalState=Wire.read();
            break;
        case 2: //Información de la alarma
            AlarState=Wire.read();
            n=Wire.read();
            if(n==0) n=6;
            else n=11;
    }
}

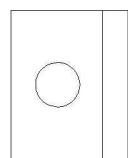
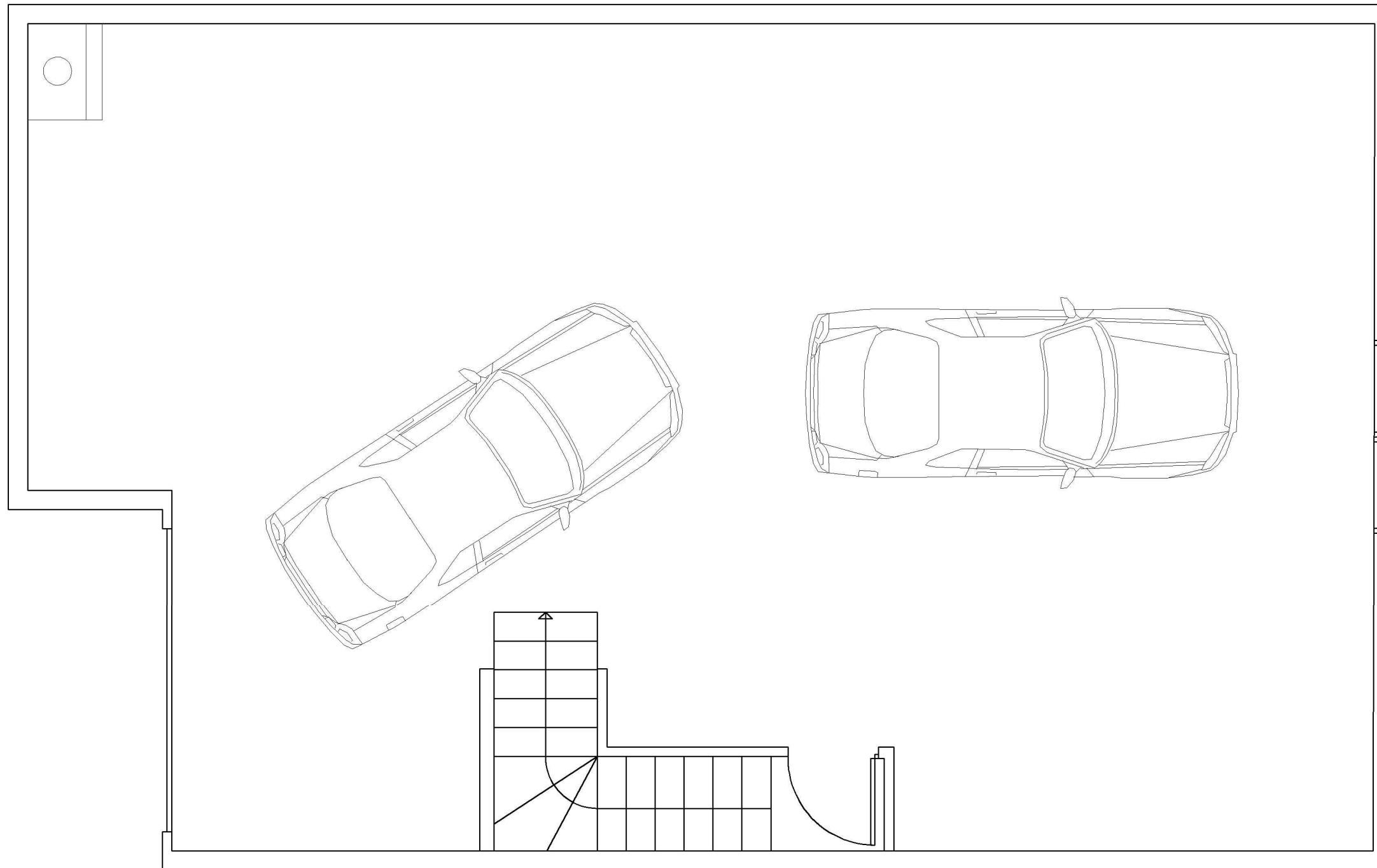
```

```

break;
case 7: //Información de la temperatura
    i=0;
    CalState=Wire.read();
    while (Wire.available(>0){
        Tzon[i]=Wire.read();
        i++;
    }
break;
case 11: ////Información de las persianas
    while(Wire.available(>0){
        Nper[i]=Wire.read();
        i++;
    }
    for(int j=0,i=0;i<11;i++){
        if(Nper[i]!=estper[i]) j++;
        if(j>0){
            PerState=1;
            break;
        }
    }
break;
}
}

```

## Anexo III. Planos de la vivienda



**Caldera**


**UNIVERSIDAD DE VALLADOLID**  
**ESCUELA DE INGENIERÍAS INDUSTRIALES**

**TITULO PROYECTO:**  
**Gestión domótica de una casa unifamiliar  
 basada en Arduino**

**PLANO:**  
**Garaje**

**Departamento:**  
**Tecnología Electrónica**

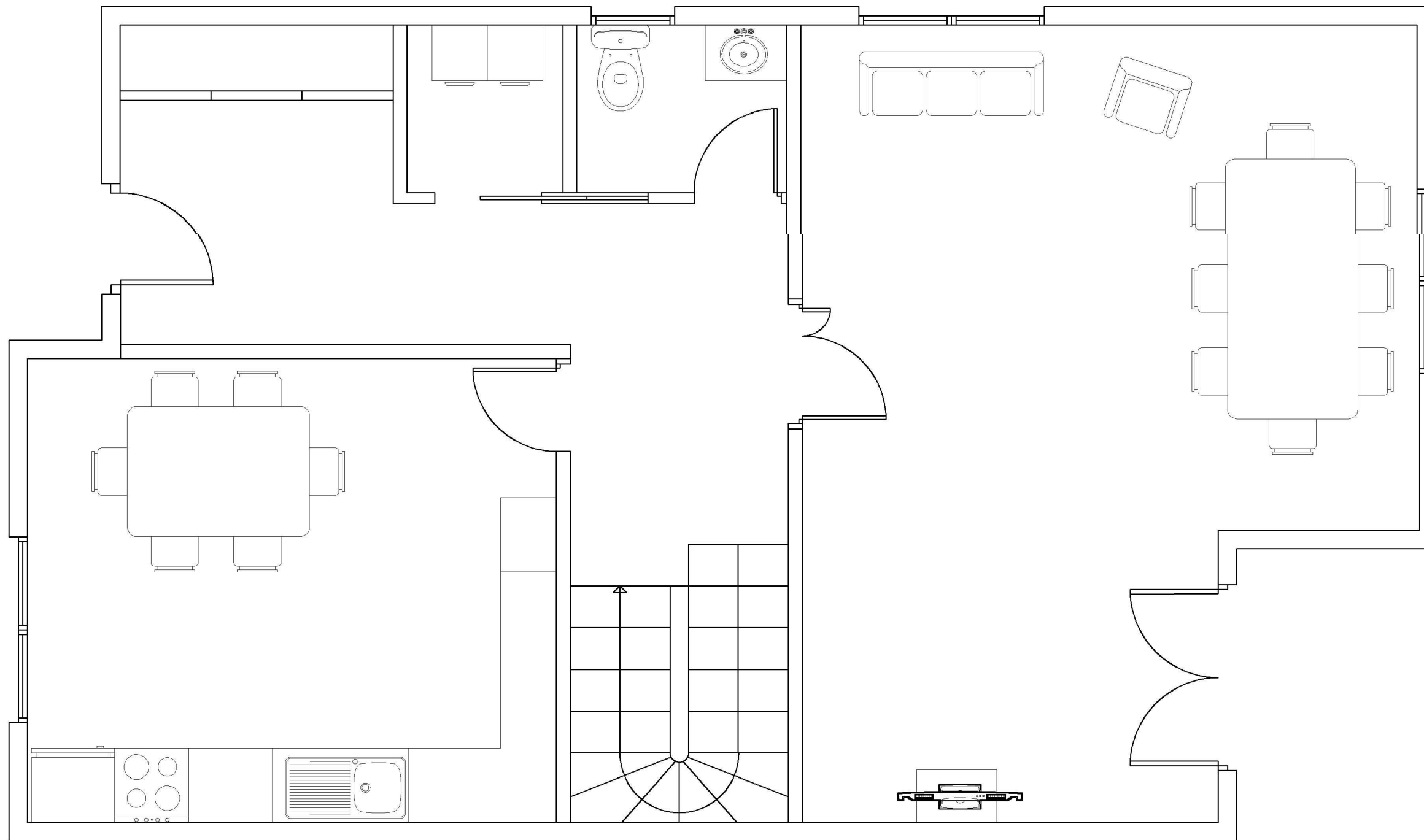
**FECHA:**  
**05-2017**

**Nº PLANO:**  
**1**

**Tutora:**  
**Carmen Quintano Pastor**

**ESCALA:**  
**1:50**

**FIRMA:**  
 EL ALUMNO:



**TITULO PROYECTO:**  
**Gestión domótica de una casa unifamiliar  
 basada en Arduino**

**PLANO:**  
**Planta Principal**

**Departamento:**  
**Tecnología Electrónica**

**FECHA:**  
**05-2017**

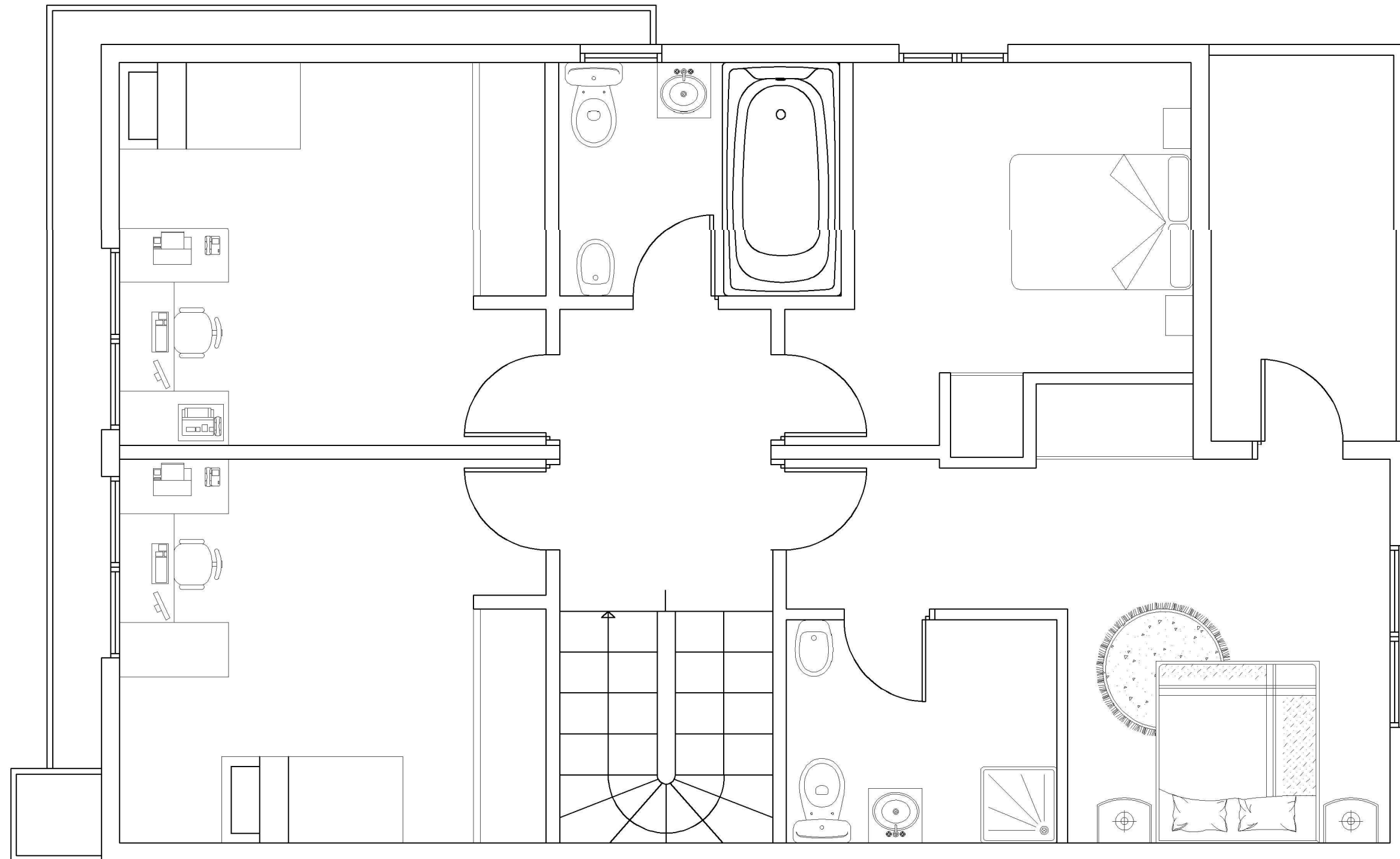
**Nº PLANO:**  
**2**

**Tutora:**  
**Carmen Quintano Pastor**

**ESCALA:**  
**1:50**

**FIRMA:**  
 EL ALUMNO:





 **UNIVERSIDAD DE VALLADOLID**  
**ESCUELA DE INGENIERÍAS INDUSTRIALES**

**TITULO PROYECTO:**  
**Gestión domótica de una casa unifamiliar  
 basada en Arduino**

**PLANO:**  
**Planta alta**

**Departamento:**  
**Tecnología Electrónica**

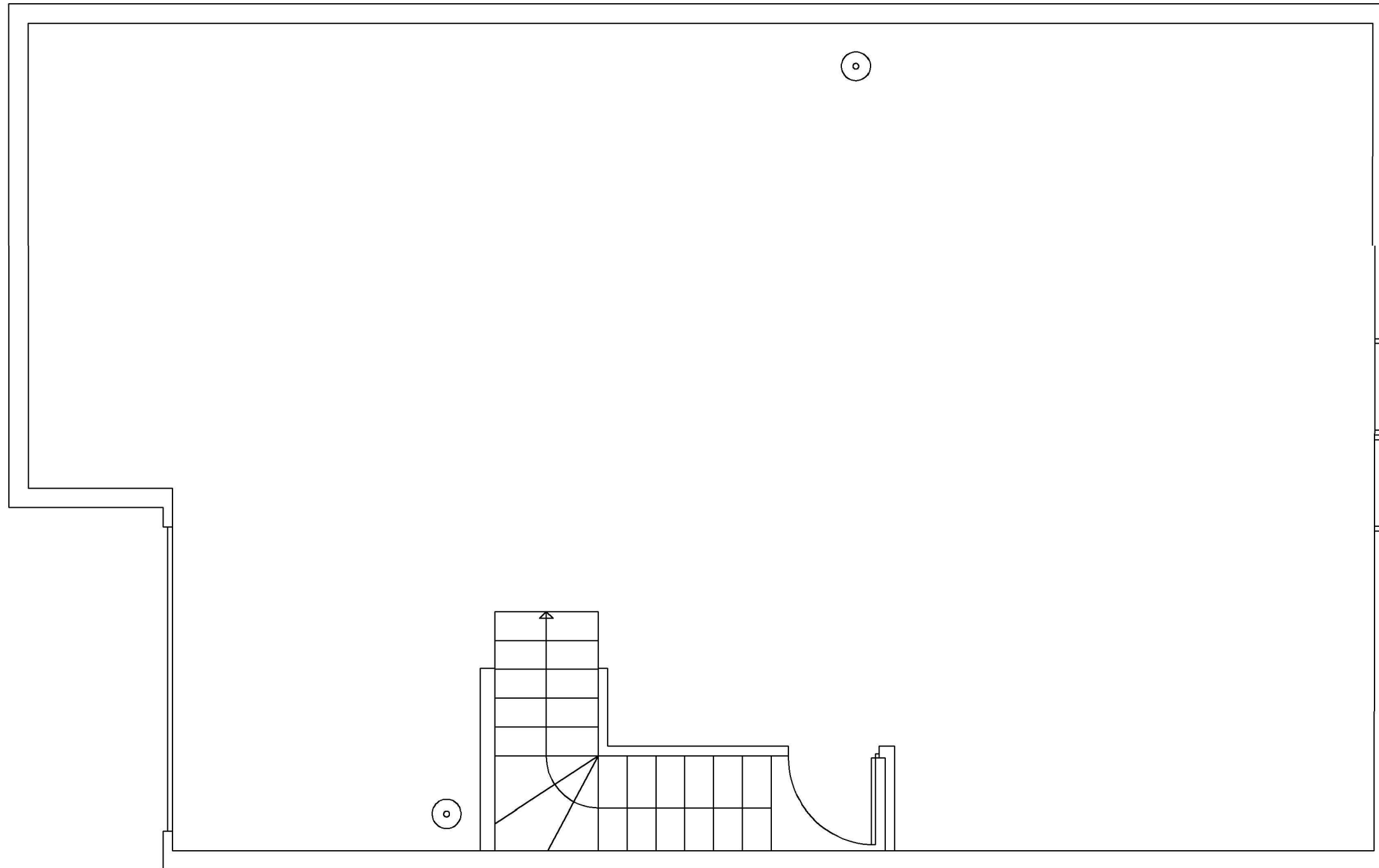
**FECHA:**  
**05-2017**

**Nº PLANO:**  
**3**

**Tutora:**  
**Carmen Quintano Pastor**

**ESCALA:**  
**1:50**

**FIRMA:**  
 EL ALUMNO:



○ **Sensor volumétrico**

 **UNIVERSIDAD DE VALLADOLID**  
 **ESCUELA DE INGENIERÍAS INDUSTRIALES**

**TITULO PROYECTO:**  
**Gestión domótica de una casa unifamiliar  
 basada en Arduino**

**PLANO:**  
**Sensores Garaje**

**Departamento:**  
**Tecnología Electrónica**

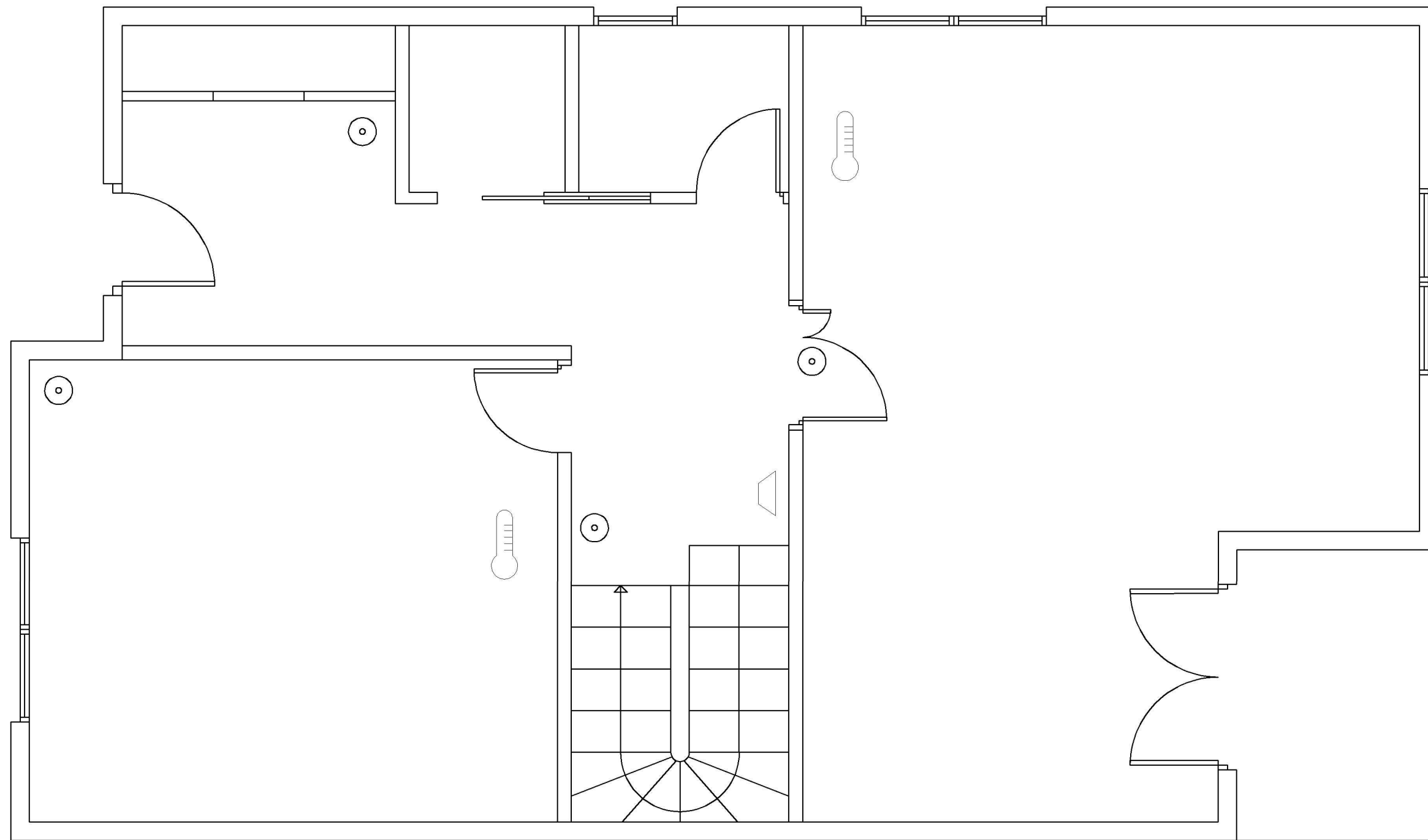
**FECHA:**  
**05-2017**

**Nº PLANO:**  
**4**

**Tutora:**  
**Carmen Quintano Pastor**

**ESCALA:**  
**1:50**

**FIRMA:**  
 EL ALUMNO:



 **Sensor volumétrico**

 **Persiana**

 **Zumbador**

 **Sensor de temperatura**

 **UNIVERSIDAD DE VALLADOLID**  
 **ESCUELA DE INGENIERÍAS INDUSTRIALES**

**TITULO PROYECTO:**  
**Gestión domótica de una casa unifamiliar  
 basada en Arduino**

**PLANO:**  
**Sensores y actuadores planta principal**

**Departamento:**  
**Tecnología Electrónica**

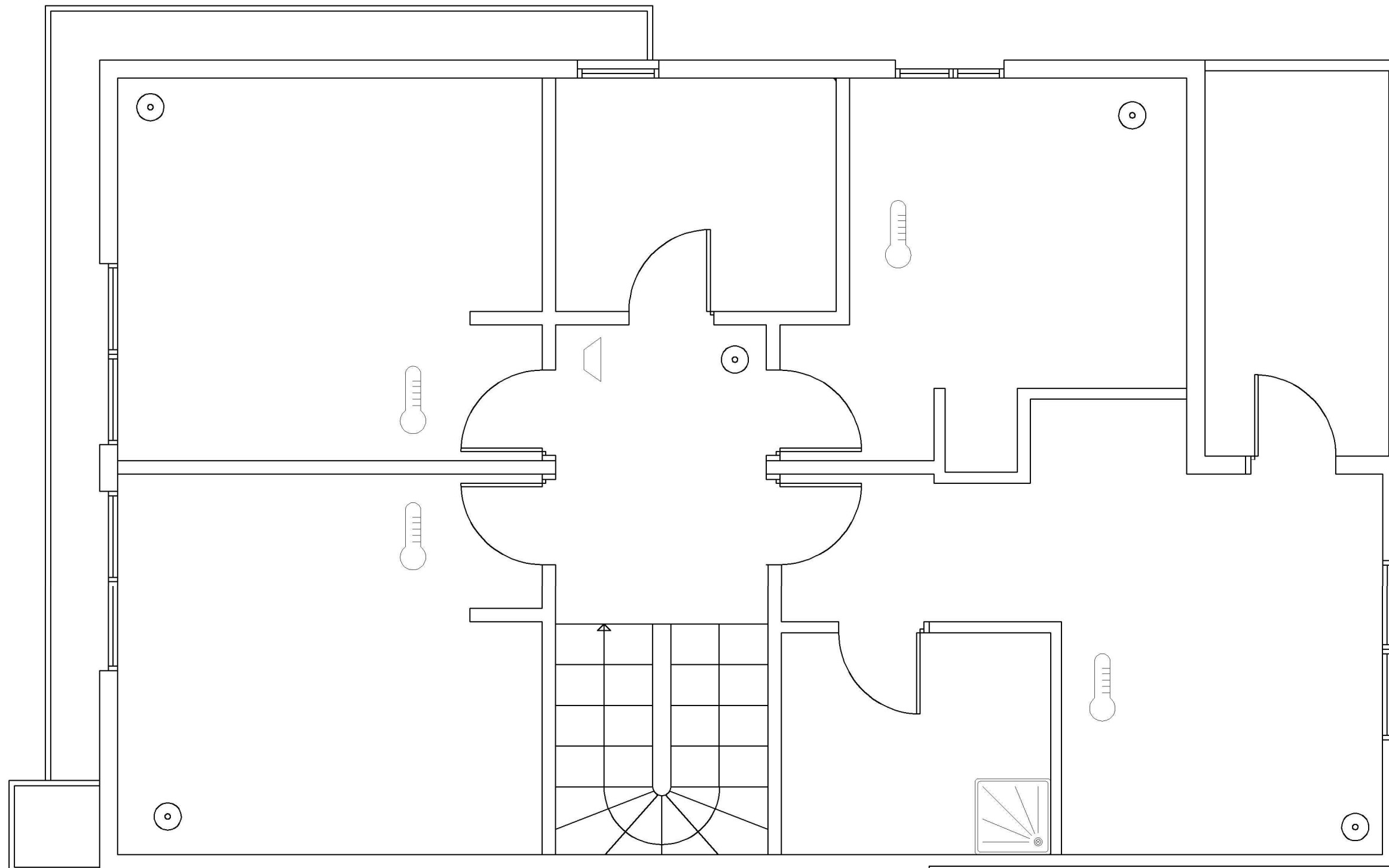
**FECHA:**  
**05-2017**

**Nº PLANO:**  
**5**

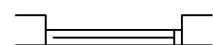
**Tutora:**  
**Carmen Quintano Pastor**

**ESCALA:**  
**1:50**

**FIRMA:**  
 EL ALUMNO:



**Sensor volumétrico**



**Persiana**



**Zumbador**



**Sensor de temperatura**


**UNIVERSIDAD DE VALLADOLID**  
**ESCUELA DE INGENIERÍAS INDUSTRIALES**

**TITULO PROYECTO:**

**Gestión domótica de una casa unifamiliar  
basada en Arduino**

**PLANO:**

**Sensores y actuadores planta alta**

**Departamento:**  
**Tecnología Electrónica**

**FECHA:**  
**05-2017**

**Nº PLANO:**  
**6**

**Tutora:**

**Carmen Quintano Pastor**

**ESCALA:**  
**1:50**

**FIRMA:**  
EL ALUMNO: