



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño

Master Universitario En Sensores Para Aplicaciones Industriales

TRABAJO FINAL DE MASTER

Sistema de sensorización para viviendas con acceso remoto

Autor: Jesús De Hoz Diego

Tutor: Rafael Masot Peris

Co-tutor: Miguel Alcañiz Fillol

INDICE

1	Introducción	4
2	Diagrama de bloques.....	5
2.1	Comunicación entre el Smartphone y las placas	5
2.1.1	Router (Wifi):.....	5
2.1.2	Ethernet:.....	5
2.1.3	Módulos de Radiofrecuencia 433 MHz:	6
2.1.3.1	Emisor:	6
2.1.3.2	Receptor:	6
	Sistema de comunicación General.....	¡Error! Marcador no definido.
2.1	Comunicación entre actuadores/sensores y las placas.....	8
3	DISEÑO DEL SISTEMA	9
3.1	Microcontrolador	11
3.2	Lenguajes.....	11
3.2.1	Lenguaje C	12
3.3	Introducción a Arduino (Software).....	13
3.4	Soluciones alternativas.....	16
3.5	Sistema servidor.....	17
3.5.1	Diagrama de bloques.....	17
3.5.2	Placa Arduino ATmega 2560 + Shield Ethernet.....	18
3.5.2.1	Microcontroladores	18
3.5.2.2	Arduino	19
3.5.3	Comunicaciones	20
3.5.3.1	Recepción de datos vía Wifi.....	20
3.5.3.2	Recepción de datos vía Ethernet	21
3.5.3.3	Recepción de datos entre Shield Y Placa servidora	21
3.5.3.4	Emisión de datos vía radiofrecuencia a otros sistemas	21
3.5.3.5	Recepción de datos de los sensores analógicos	21
3.5.4	Sensores	21
3.5.4.1	Sistema de medida	21
3.5.4.2	Bloque adquisición de señal	22
3.5.4.3	Variables	23
3.5.4.4	Sensores y material	23
3.5.4.4.1	Sensor de Temperatura/Humedad	24
3.5.4.4.2	Sensor de Luz (LDR)	24
3.5.4.4.3	Otros sensores.....	25
3.5.5	Actuadores	29
3.5.6	Firmware	29
3.5.6.1	Led	29
3.5.6.1.1	Activación de librerías	30
3.5.6.1.2	Conexión y seguridad	30
3.5.6.1.3	Programación de los diferentes sensores	30
3.5.6.1.4	Programación del sistema de comunicación por radiofrecuencias.....	30

3.5.6.1.5	Cambio horario.....	30
3.5.6.1.6	Actuadores	30
3.5.6.1.7	Debug_Mode.....	30
3.5.6.2	Common_Functions.....	31
3.5.6.3	Excontrol_def.....	31
3.5.6.4	InfraRedFunction	31
3.5.6.5	Mhz433.....	31
3.5.6.6	PrintLDC.....	31
3.5.6.7	Activación de librerías.	31
3.5.6.8	Una función de inicialización	32
3.5.6.9	Bucle del programa.....	32
3.6	Nodo Esclavo	32
3.6.1	Diagrama de bloques.....	32
3.6.2	Placa Uno.....	33
3.6.3	Comunicaciones	34
3.6.3.1	Recepción de la orden por radiofrecuencia.....	34
3.6.4	Actuadores	34
3.6.5	Firmware	34
3.6.5.1	Activación de librerías.	34
3.6.5.2	Una función de inicialización	34
3.6.5.3	Bucle del programa.....	35
3.7	Controlador inalámbrico de activación de carga (ciac)	35
3.7.1	Diagrama de bloques.....	35
3.7.2	Placa Arduino	36
3.7.3	Comunicaciones	36
3.7.3.1	Recepción de la orden por radiofrecuencia.....	37
3.7.4	Firmware	37
3.7.4.1	Activación de librerías.	37
3.7.4.2	Una función de inicialización	37
3.7.4.3	Bucle del programa.....	37
4	Validación	37
4.1	Dispositivo Smartphone	37
5	Conclusiones.....	40
6	Futuras Líneas De Trabajo.....	40
7	Bibliografía.....	41

1 Introducción

El objetivo del presente proyecto es el de diseñar, desarrollar e instalar un sistema de sensorización (domótico) para una vivienda con acceso remoto. Para ello se tendrá que implementar el hardware (placas, nodos, sensores, actuadores, etc...) y el software necesario.

La parte de hardware consiste en el acondicionamiento de las tensiones de entrada de los diferentes sensores y de las diferentes plataformas y acondicionar las tensiones de salida del mismo, para que pueda interactuar con los actuadores. Consta de un puerto serie para comunicarse con el ordenador, una App para comunicarse con el dispositivo y poder visualizar los diferentes sensores (incluyendo sus respectivas lecturas con respecto al tiempo) y para poder comandar los diferentes actuadores.

La parte de software se implementó mediante la plataforma Arduino utilizando el lenguaje de programación C, en el que se desarrolla el algoritmo para un óptimo funcionamiento de la placa.

El proyecto ha sido ideado y propuesto por mí, Jesús de Hoz Diego, alumno de la Universidad Politécnica de Valencia con el propósito de aplicar e incrementar los conocimientos adquiridos en el Master Universitario en Sensores para Aplicaciones Industriales el cual tiene el propósito de añadir comodidades en la vida privada de cada uno en su casa a su vez que aportar un ahorro energético y mantener informado al usuario en todo momento de la cantidad de información que este sistema puede ofrecer (Humedad, temperatura, luminosidad, avisos en caso de la aparición de gases nocivos, etc...).

Todo el sistema domótico podrá ser visualizado desde una App y a su vez el usuario podrá interactuar.

Para que esto sea posible tendremos que tener una serie de conceptos claros los cuales nos facilitaran el entendimiento del consiguiente proyecto.

2 Diagrama de bloques

Para una mejor explicación del sistema domotico a continuación veremos el sistema que sigue una orden desde que sale del dispositivo móvil hasta que llega a su actuador pertinente. Para ello dividiremos el diagrama de bloques en dos partes:

2.1 Comunicación entre el Smartphone y las placas

Este tipo de comunicación es el encargado de interrelacionar el dispositivo móvil con las diferentes placas a través de un router a través de una comunicación wifi, comunicación Ethernet y comunicación por radiofrecuencia. A continuación, se expondrá los diferentes tipos mecanismos de comunicación y podremos ver un esquema total de la misma:

2.1.1 Router (Wifi):

Es un dispositivo de unión de comunicaciones entre el dispositivo Android y el Sistema servidor.



Figura 1: Router

2.1.2 Ethernet:

Es una placa de unión entre el router y la placa central de procesamiento (Arduino ATMEGA2560).



Figura 2: Placa Ethernet

2.1.3 Módulos de Radiofrecuencia 433 MHz:

Son los encargados de mantener una comunicación entre los diferentes sistemas de nuestra domótica.

2.1.3.1 Emisor:

Es uno de los dispositivos de comunicación por radiofrecuencia el cual se encarga de emitir unos códigos determinados afiliados al tipo de acción que se quiera realizar.



Figura 3: Dispositivo emisor de radiofrecuencia

2.1.3.2 Receptor:

Es uno de los dispositivos de comunicación por radiofrecuencia el cual se encarga de recibir los códigos enviados por el emisor para un posterior procesamiento de los mismos.



Figura 4: Dispositivo receptor de radiofrecuencia

2.2 Sistema de comunicación General

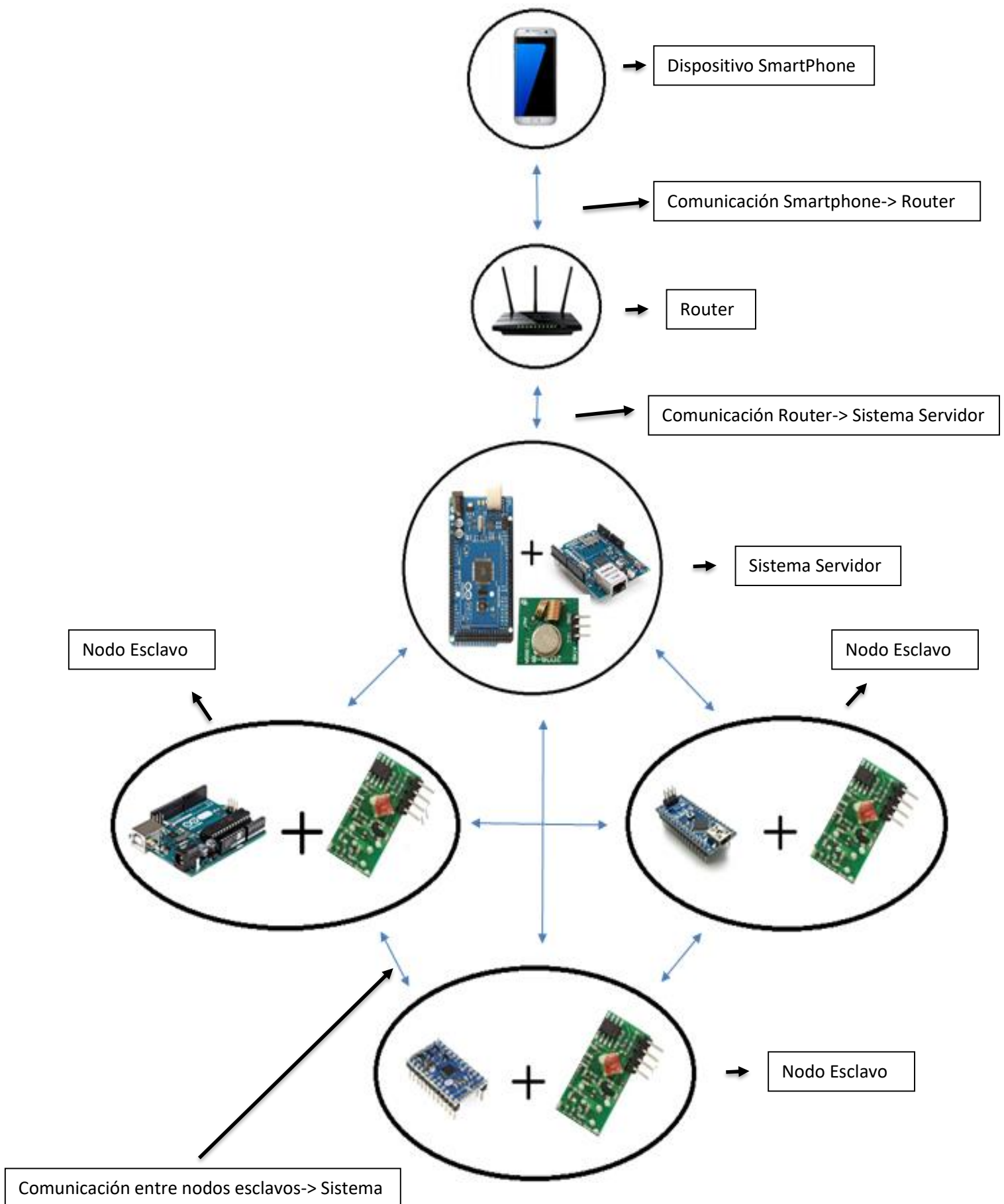


Figura 5: Diagrama de bloques de la comunicación general

2.1 Comunicación entre actuadores/sensores y las placas

Este tipo de comunicación es el encargado de interrelacionar cada uno de los diferentes sensores y actuadores del sistema con su placa correspondiente

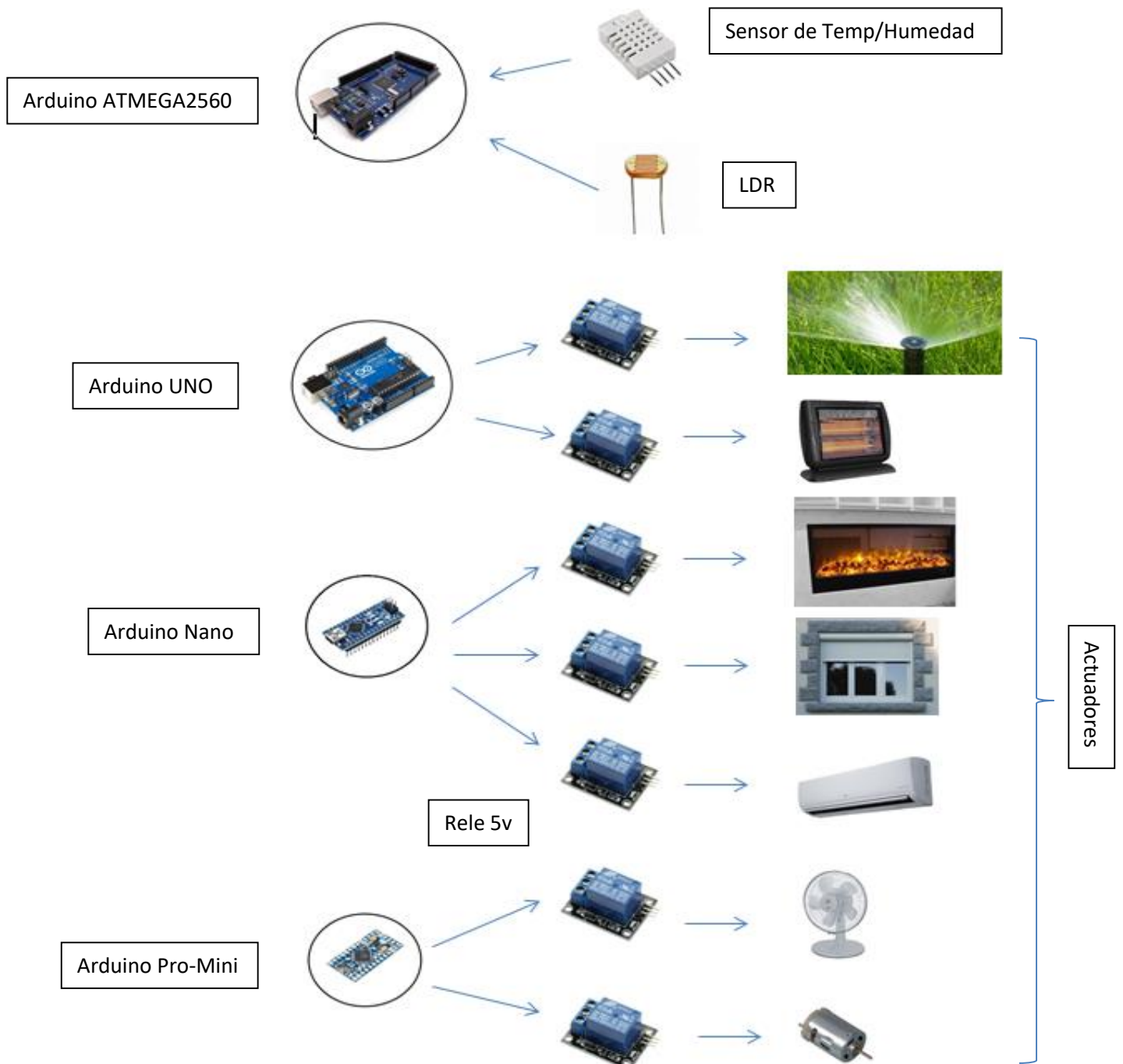


Figura 5: Diagrama de bloques de la comunicación general

3 DISEÑO DEL SISTEMA

Nuestro sistema de sensorización ha sido diseñado para una maqueta de una vivienda “x”, la cual constará de 1 habitación, 1 baño, 1 salón, 1 cocina y un jardín. Para poder simular la activación de los actuadores, se ha posicionado un led en la situación hipotética donde estarían, de forma que cuando comandemos sobre ellos se irán encendiendo y apagando según nosotros las vayamos activando.

- El circuito domótico ha sido diseñado de forma que el camino que sigue una orden realizada desde la App hasta que llega al actuador o sensor es el siguiente:
 1. La orden se realiza desde cualquier dispositivo móvil Android.
 2. Esta orden viaja por wifi hasta un router al que el dispositivo Android estará conectado.
 3. Dicho Router estará comunicado con la tarjeta Arduino ATMEGA 2560 por medio de un Shield Ethernet (Tarjeta de expansión con conexión Ethernet).
 4. Una vez ha llegado la orden a la Tarjeta Arduino ATMEGA2560, si requiriese comunicarse con la tarjeta Arduino UNO situada en otro punto de la vivienda la tarjeta Arduino ATMEGA2560 codificaría la orden y la enviaría por medio de un sistema de radiofrecuencias conectado a ambas placas.
 5. Una vez la orden ha llegado a la Tarjeta Arduino UNO, este se encarga de la salida determinada con aquel actuador que quisiésemos activar/desactivar o leer los datos de un sensor.

En este caso la Tarjeta Arduino ATMEGA2560 actuará de servidor. Una peculiaridad de este sistema de comunicación entre Tarjetas, es la transmisión de datos por red de radiofrecuencias.

- Nuestro sistema de sensorización estará provisto de una contraseña como protección.
- Contará con un sistema de apagado general como opción dentro de la programación.
- Además, podremos graficar los datos de los sensores que nosotros decidamos pudiendo observar dichos datos en el intervalo de días deseado.
- Todos los datos de los sensores del sistema se guardarán en una memoria SD cada 15 minutos.
- El reloj interno contara con la opción de calibrado por si se desconfigurase.

A continuación, expondremos en qué placa, pines, y código de radiofrecuencia están conectados los diferentes actuados y sensores del sistema diseñado en este proyecto.

	A. ATMEGA	A. UNO	A. nano	Código Radiofrecuencia	Pin	Actuador	Sensor
Puerta garaje	No	SI	No	5000	3	SI	No
Toldo	No	SI	No	5001	4	SI	No
Luz jardín	No	SI	No	5002	5	SI	No
Riego automático	No	SI	No	5003	6	SI	No
luz salón	No	SI	No	5004	7	SI	No
Persianas	No	SI	No	5005	8	SI	No
Chimenea	No	SI	No	5006	9	SI	No
Ventilador	No	SI	No	5007	10	SI	No
Luz baño	No	SI	No	5008	11	SI	No
Calefactor baño	No	SI	No	5009	12	SI	No
CIAC	No	No	Si	5010	5	SI	No
Luz Habitación	SI	No	No	-	22	SI	No
Persianas	SI	No	No	-	24	SI	No
Mesa De Trabajo	SI	No	No	-	28	SI	No
Aire Acondicionado Calor	SI	No	No	-	32	SI	No
Aire Acondicionado Frio	SI	No	No	-	30	SI	No
Luz Cocina	SI	No	No	-	40	SI	No
Tostadora	SI	No	No	-	34	SI	No
Campana	SI	No	No	-	36	SI	No
Lavavajillas	SI	No	No	-	38	SI	No
Radio	SI	No	No	-	39	SI	No
Temperatura	SI	No	No	-	15	SI	No
Humedad	SI	No	No	-	15	No	Si
LDR	SI	No	No	-	A3	No	Si
Comunicación	SI	No	No	-	14	No	No
Comunicación	No	SI	No	-	2	No	No
Comunicación	No	No	SI	-	2	No	No

En cuanto al sistema general de la domótica podremos diferenciar el sistema servidor, el nodo esclavo y el controlador inalámbrico de activación de carga (ciac).

3.1 Microcontrolador

En los últimos años se ha facilitado enormemente el trabajo con los microcontroladores al bajar los precios, aumentar las prestaciones y simplificar los montajes, de manera que en muchas ocasiones merece la pena utilizarlos en aplicaciones donde antes se utilizaba lógica discreta.

Hoy en día Arduino es uno de los microcontroladores más aceptados para diseños de aficionados.

Uno de los inconvenientes encontrados son los puertos de salida, solo constaría de dos salidas analógicas.

3.2 Lenguajes

Un lenguaje de programación es un idioma artificial diseñado para expresar computaciones que pueden ser llevadas a cabo por máquinas como las computadoras, PICs o microcontroladores.

Para que la computadora entienda nuestras instrucciones debe usarse un lenguaje específico conocido como código máquina, el cual la máquina comprende fácilmente, pero que lo hace excesivamente complicado para las personas. De hecho, solo consiste en extensas cadenas de 0 y 1.

Para facilitar este trabajo usamos programas que se utilizan como traductores para que podamos utilizar palabras en vez de estas secuencias de números. Esta secuencia de palabras que podemos utilizar en los programas se le llamó lenguaje ensamblador.

La estructura de los lenguajes es:

- Lenguaje máquina: es la representación booleana de los procesos aritmético lógicos o invocaciones de memoria. Es el único lenguaje que realmente entiende un sistema informático. Son lenguajes dependientes de la máquina, lo que significa que cada procesador emplea su código máquina particular. Se precisa un profundo conocimiento de la estructura interna del computador para poder operar sobre él.

- Lenguajes ensambladores: son instrucciones del lenguaje máquina escritas en forma simbólica mediante códigos nemotécnicos. No es directamente ejecutable por la máquina por lo que se debe traducir a un programa equivalente en código máquina por programas llamados ensambladores. El uso del lenguaje ensamblador es una solución idónea en el caso de sistemas basados en pequeños microcontroladores debido a que generan código compacto y rápido.
- Lenguajes de alto nivel: permiten una representación más parecida a la escritura lógico-matemática. No interesan los conceptos de la máquina, con lo cual los programas son más cortos y claros. Un mismo programa puede ser ejecutado en distintos ordenadores. Algunos ejemplos de este tipo de lenguajes serían Pascal, Basic, Java, C, etc. Esta serie de programas necesitan traducir el código o el lenguaje de alto nivel a código ejecutable, para eso utilizan compiladores o intérpretes.

Los compiladores traducen el programa original (fuente) al programa equivalente en código máquina (objeto) directamente ejecutable por la máquina o microcontrolador. Los programas hechos con lenguajes compilados suelen ser más lentos y ocupar más memoria que sus equivalentes en ensamblador. El C es un lenguaje compilado.

3.2.1 Lenguaje C

La opción de escoger el lenguaje de programación C fue porque el mismo que nos proporciona los microcontroladores también ofrece una suite de programación C para sus micros. Esta suite contiene el Arduino, que es el entorno de programación donde desarrollaremos el código del programa.

La opción de elegir un micro que se pueda programar en C es por la eficiencia del código que produce y es el lenguaje de programación más popular para crear software de sistemas, aunque también se utiliza para crear aplicaciones.

Uno de los objetivos de diseño del lenguaje C es que sólo sean necesarias unas pocas instrucciones en el lenguaje máquina para traducir cada elemento del lenguaje, sin que haga falta un soporte intenso en tiempo de ejecución.

3.3 Introducción a Arduino (Software).

Arduino es un entorno de desarrollo gratuito para microcontroladores con núcleo Arduino (Hardware). Integra un compilador GCC, un programador de memoria flash y un asistente que permite generar código. Uno de los principales atractivos de este entorno es la comunidad Arduino por la que los usuarios pueden compartir su código con el resto de la comunidad fácilmente a través del software de desarrollo y de diferentes foros, de esta manera podemos encontrar librerías en C que permiten trabajar con los diferentes componentes de las placas y que pueden ser incorporadas y modificadas en los trabajos de los desarrolladores facilitando así su tarea. La aplicación puede ser adquirida en la web del proveedor <https://www.arduino.cc/>, actualmente la última versión estable es V1.8.2. La instalación no presenta ninguna dificultad, tan sólo hay que seleccionar la ubicación del software y la aplicación se ocupará de todo. La primera vez que se inicie el programa se visualizará una pantalla como la que se muestra en la siguiente imagen.

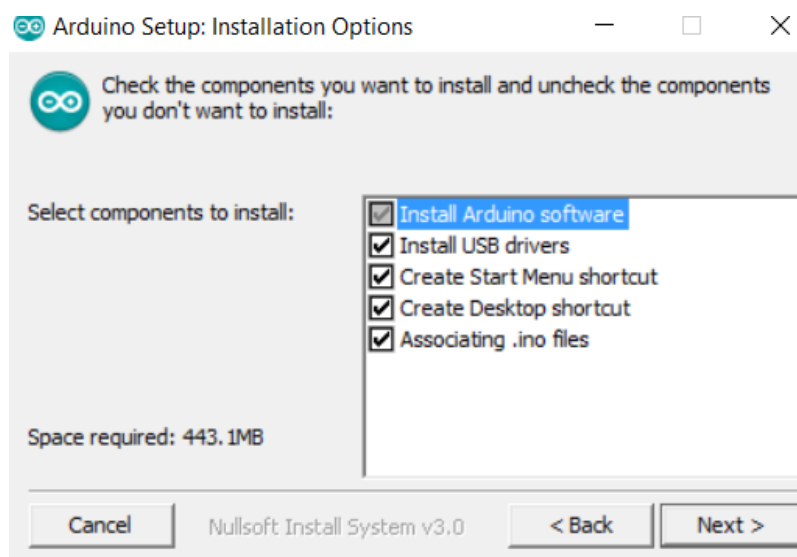


Figura 6: Selección de componentes a instalar

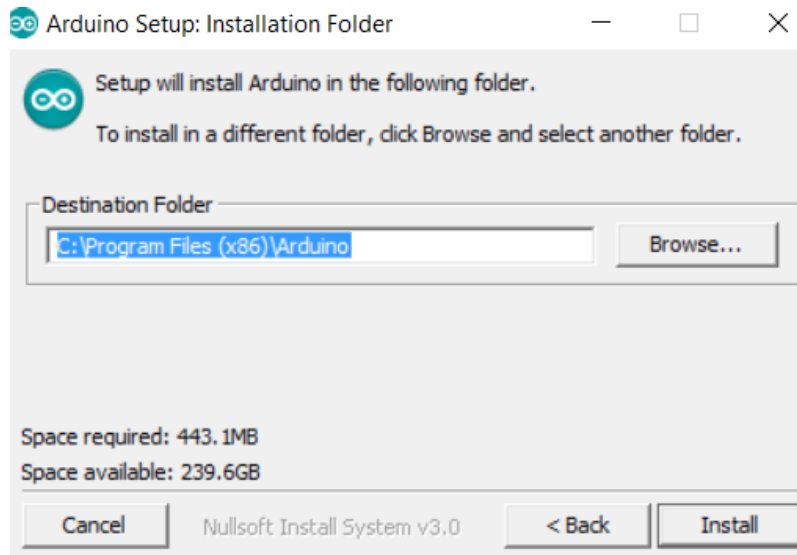


Figura 7: Selección del directorio de instalación

Una vez dentro del programa nos aparecerá La ventana de programación donde podremos comenzar a escribir nuestro código

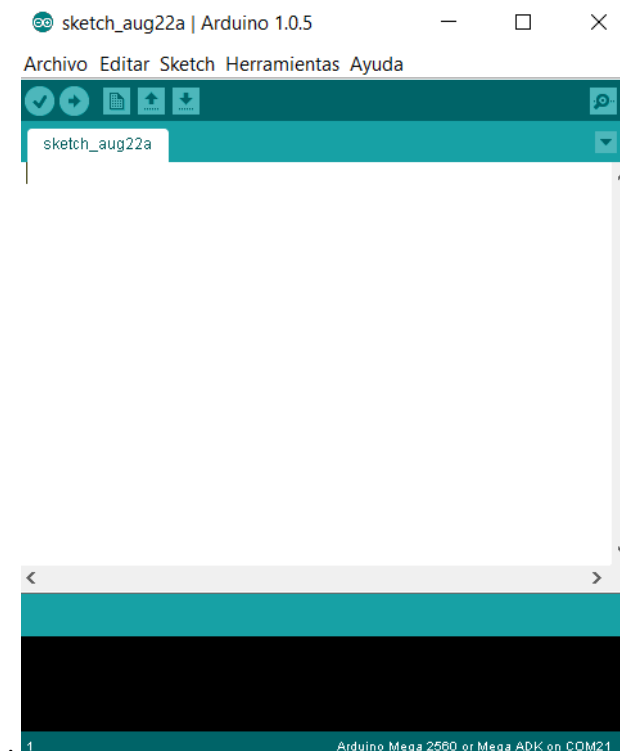


Figura 8: Pantalla de programación

Una vez tengamos escrito el código debemos seguir los siguientes pasos.

- Seleccionar la placa donde volveremos el código en Herramientas/Tarjeta.
- Seleccionar el puerto donde se encuentre conectada la placa en Herramientas/Puerto.
- Hacer Clic sobre la Flecha situada en la parte superior izquierda de la ventana del programa la cual se encargará de compilar el código y subirlo a la placa anteriormente configurada conectada en el puerto anteriormente configurado.

Para poder ejecutar algunas funciones de la programación tendremos que hacer uso de unas librerías determinadas las cuales podremos ver en:

- La carpeta del juego Arduino/Libraries

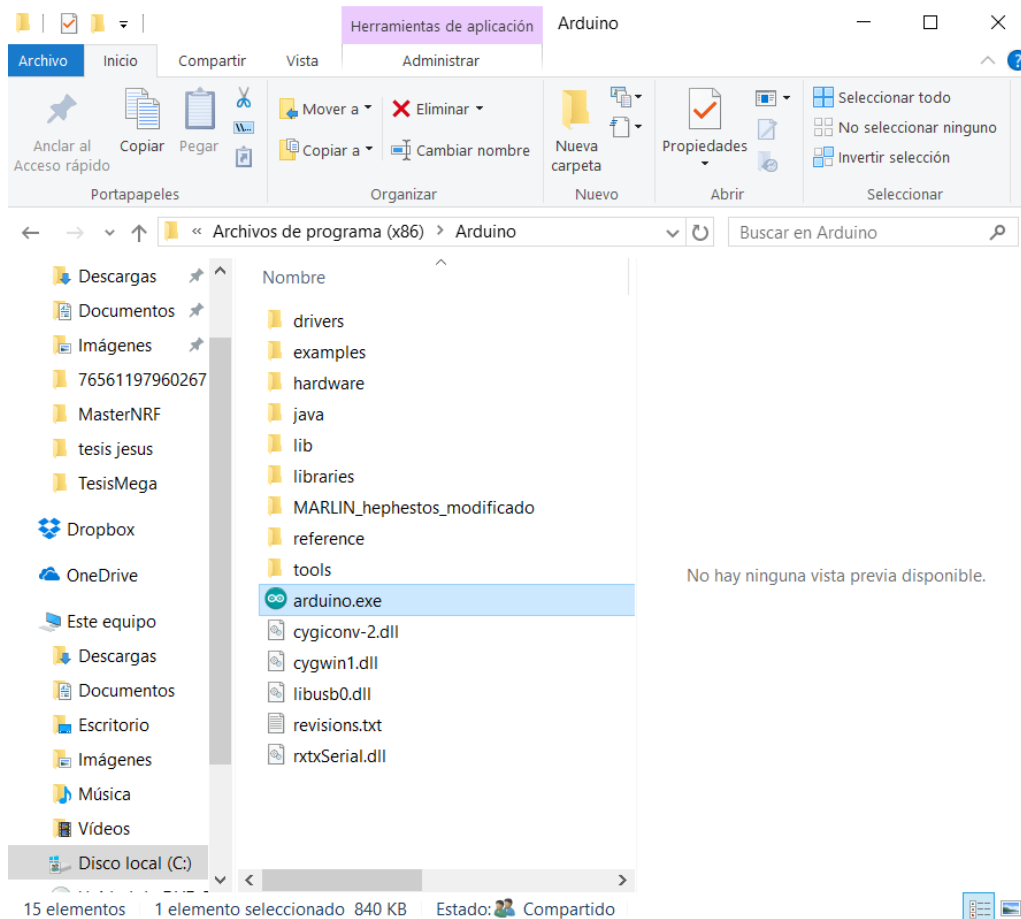


Figura 9: Directorio de instalación

- En la ventana del software, en la pestaña superior Sketch/Importar Librería.

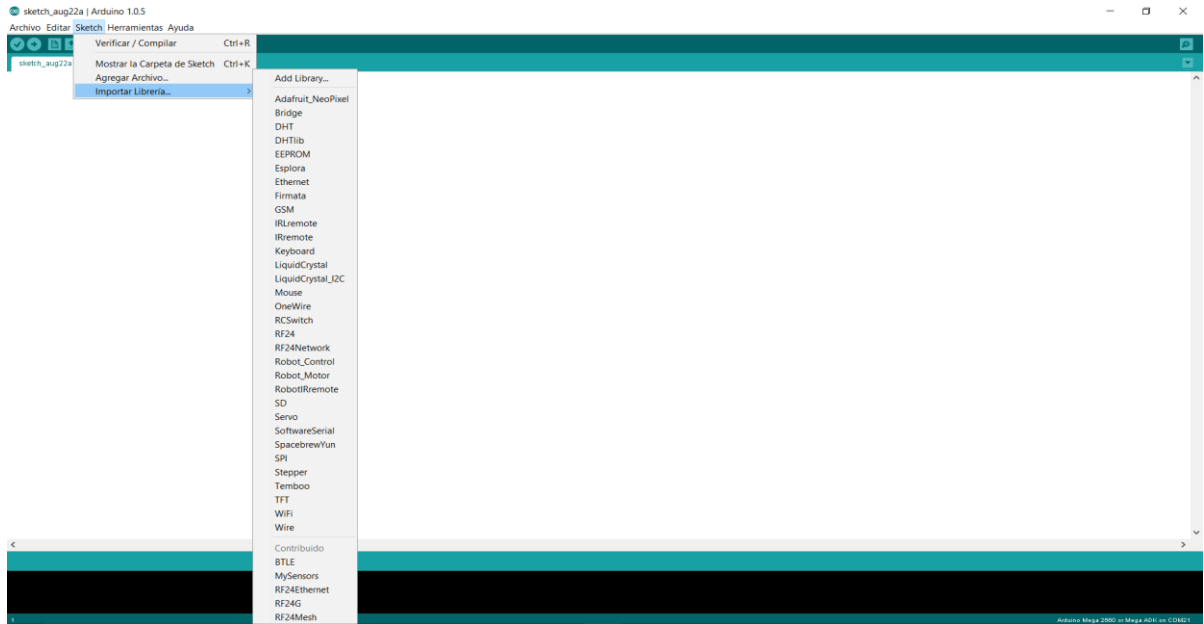


Figura 10: Importación de librerías

3.4 Soluciones alternativas

El sistema anteriormente descrito requiere de unos elementos tanto hardware como software para efectuar tanto la comunicación entre pc-> placa servidora -> placas esclavas -> sensores, como el control de cada placa, por lo tanto, se requiere de la implementación de diferentes tipos de comunicación con el fin de realizar el control del proceso.

La finalidad de este proyecto es realizar un sistema que sea capaz de:

- comunicar entre una placa servidora y el computador mediante un puerto USB.
- Comunicar un dispositivo móvil Android con la placa servidora mediante un router wifi el cual estará conectado vía ethernet a la placa servidora.
- Comunicar entre la placa servidora y el resto de placas mediante sistemas por radiofrecuencias
- Comunicar los diferentes sensores y actuadores con las diferentes placas esclavas.

3.5 Sistema servidor.

Esta parte del sistema domótico es la encargada de manejar toda la información, recibirla vía wifi y enviarla por radiofrecuencia a los diferentes sistemas del circuito. Además, también es el encargado de recibir la información de los diferentes sensores, procesarla codificarla y enviarla al dispositivo móvil vía Ethernet/wifi.

También posee un reloj el cual nos servirá no solo para mostrarnos la hora actual sino también para poder medir la escala de tiempos de las gráficas que podemos dibujar de los sensores.

3.5.1 Diagrama de bloques

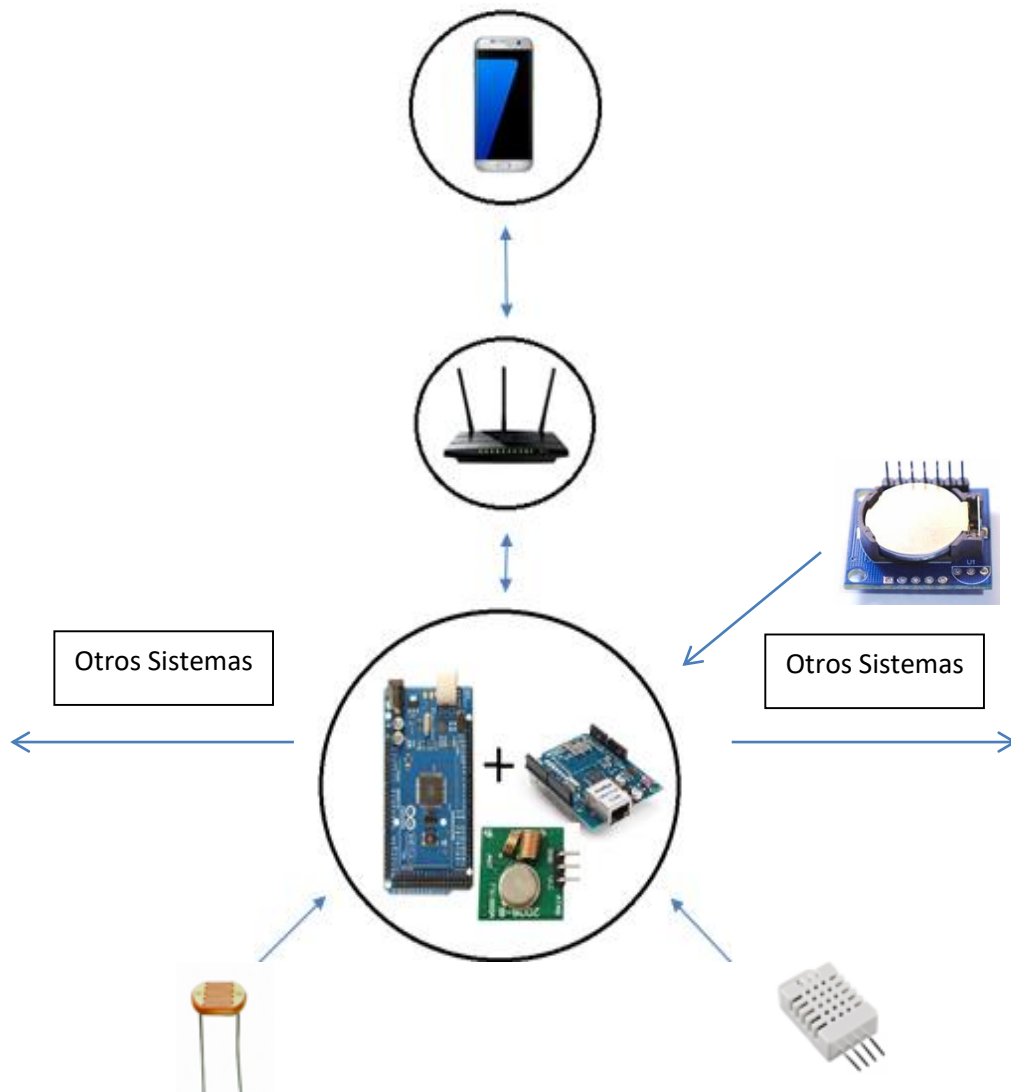


Figura 11: Diagrama de bloques Sistema servidor

3.5.2 Placa Arduino ATmega 2560 + Shield Ethernet

Para poder nombrar este campo, es necesario que antes conozcamos algunos conceptos básicos sobre el tema en cuestión.

3.5.2.1 Microcontroladores

Un microcontrolador es un circuito integrado que incluye en su interior las tres unidades funcionales de una computadora: unidad central de proceso, memoria y periféricos de entrada y salida.

Los microcontroladores se utilizan en circuitos electrónicos comerciales desde hace unos años de forma masiva, debido que permiten reducir el tamaño y el precio de los equipos.

En los últimos años se ha facilitado enormemente el trabajo con los ellos al bajar los precios, aumentar las prestaciones y simplificar los montajes, de manera que en muchas ocasiones merece la pena utilizarlos en aplicaciones donde antes se utilizaba lógica discreta.

Son diseñados para reducir el costo económico y el consumo de energía de un sistema en particular. La idea es que el circuito integrado se coloque en el dispositivo, enganchado a una fuente de energía y de la información necesaria.

Un microcontrolador típico tendrá un generador de reloj integrado y una pequeña cantidad de memoria, disponen de entradas y salidas, convertidores de analógico a digital y viceversa, temporizadores, UARTs y buses de serie de interfaz especializados como el CAN o I2C. Normalmente vienen con lenguajes de programación integrados como BASIC o C.



Figura 12: Imagen de Microcontroladores.

3.5.2.2 Arduino

Los microcontroladores analizados, son de Arduino ya que además de disponer del puerto de comunicación RS232, contiene el número de entradas y salidas tanto analógicas como digitales deseadas para la realización del TFM tal y como viene referenciado en los datasheet del anexo.

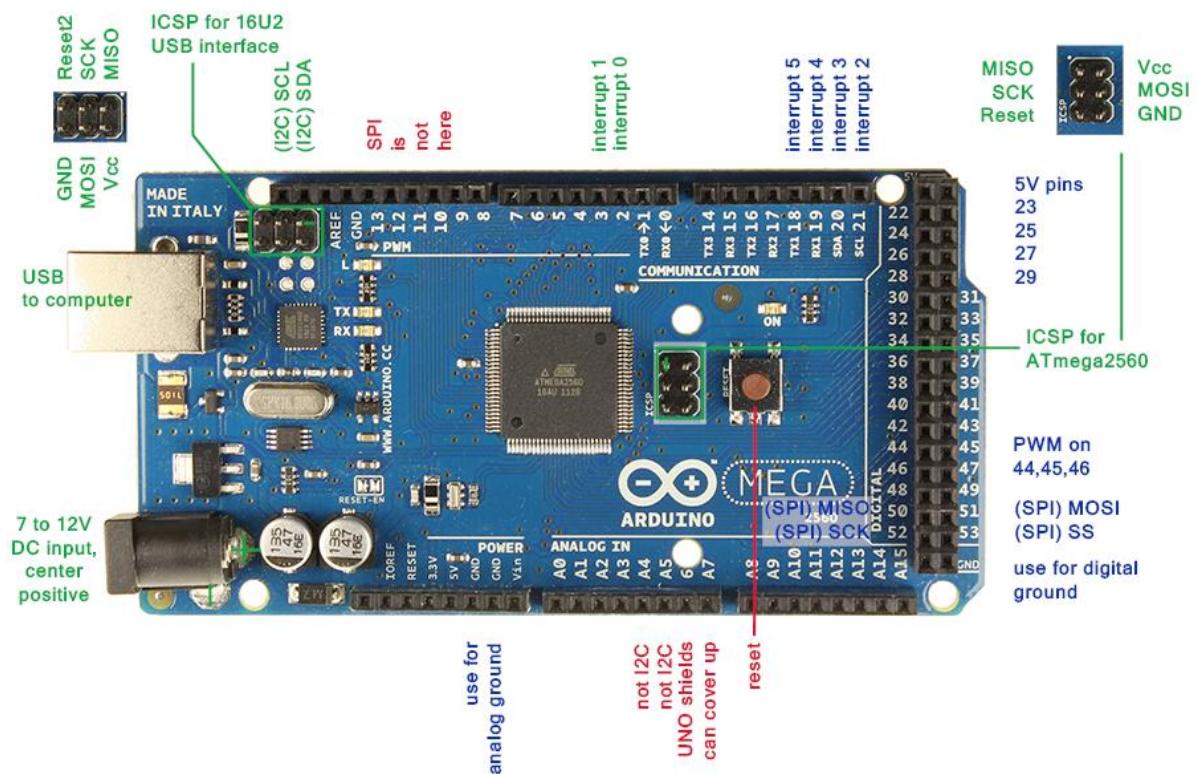


Figura 13: Placa Arduino MEGA AT2560

- Núcleo: ATMEGA2560
 - 16 MHz de frecuencia máxima,
 - Multiplicador de un solo ciclo y hardware división

- Memorias
 - 64 a 256 Kbytes de memoria flash
 - 8 Kbytes de SRAM
 - 4Kbytes EEPROM

- Periféricos
 - Periféricos soportados: temporizadores, ADC, SPI, I2Cs, USARTs y DAC

- 1 × 12 bits, 1.2 μ s convertidor A / D (hasta 16canales)
- x 4 Serial USARTs
- x 12 PWM channels
- 2 × 12 bits convertidores D / A
- Hasta 86 puertos de E / S rápidas
- **Microcontrolador:** ATmega2560
- **Voltaje Operativo:** 5V
- **Voltaje de Entrada:** 7-12V
- **Voltaje de Entrada(límites):** 6-20V
- **Pines digitales de Entrada/Salida:** 54 (de los cuales 15 proveen salida PWM)
- **Pines análogos de entrada:** 16
- **Corriente DC por cada Pin Entrada/Salida:** 40 mA
- **Corriente DC entregada en el Pin 3.3V:** 50 mA
- **Clock Speed:** 16 MHz

Temperatura ambiente de funcionamiento: -40 to $+85$ °C / -40 to $+105$ °C

Temperatura de unión: -40 to $+125$ °C

3.5.3 Comunicaciones

En este en este sistema tendremos multitud de comunicaciones las cuales están divididas en:

3.5.3.1 Recepción de datos vía Wifi

El sistema recibe los datos o la orden del dispositivo Android la cual viaja vía Wifi hasta el router, al que debe estar conectado tanto el Dispositivo Android como el Sistema servidor.

3.5.3.2 Recepción de datos vía Ethernet

El sistema servidor observa la orden recibida vía Wifi y la traduce a lenguaje Ethernet para un posterior análisis de la misma.

3.5.3.3 Recepción de datos entre Shield Y Placa servidora

Una vez la orden llega al Shield Ethernet de Arduino, esta, lo que hace es traducir dicha información para poder procesarla en la unidad central del sistema servidor (Placa ATMEGA 2560).

3.5.3.4 Emisión de datos vía radiofrecuencia a otros sistemas

Dependiendo de la información procesada, la unidad central se encargara de, o activar los actuadores pertinentes que se encuentren conectados a dicha placa, o enviar al módulo emisor de radio frecuencias la orden pertinente con el código necesario para comunicarse con el sistema esclavo especificado en la orden para que pueda ser emitida correctamente y posteriormente recibida por el sistema pertinente para una activación inmediata del actuador en cuestión.

3.5.3.5 Recepción de datos de los sensores analógicos

En la unidad central están conectados los diferentes sensores del sistema, los cuales están en continua emisión de datos, los cuales son procesados y enviados al dispositivo Android para una posible monitorización en el caso de que se requiriese.

Además, dichos datos serán guardados cada 15 minutos en una memoria SD situada en el Shield Ethernet de Arduino.

3.5.4 Sensores

Para poder nombrar este campo, es necesario que antes conozcamos algunos conceptos básicos sobre el tema en cuestión.

3.5.4.1 Sistema de medida

Un sistema electrónico de medida es aquel cuya finalidad es obtener información acerca de un proceso físico. Se encarga de obtener la señal que provee un sensor o transductor, acondicionarla, procesarla y en función de la aplicación transmitir, almacenar y/o visualizar los datos obtenidos.

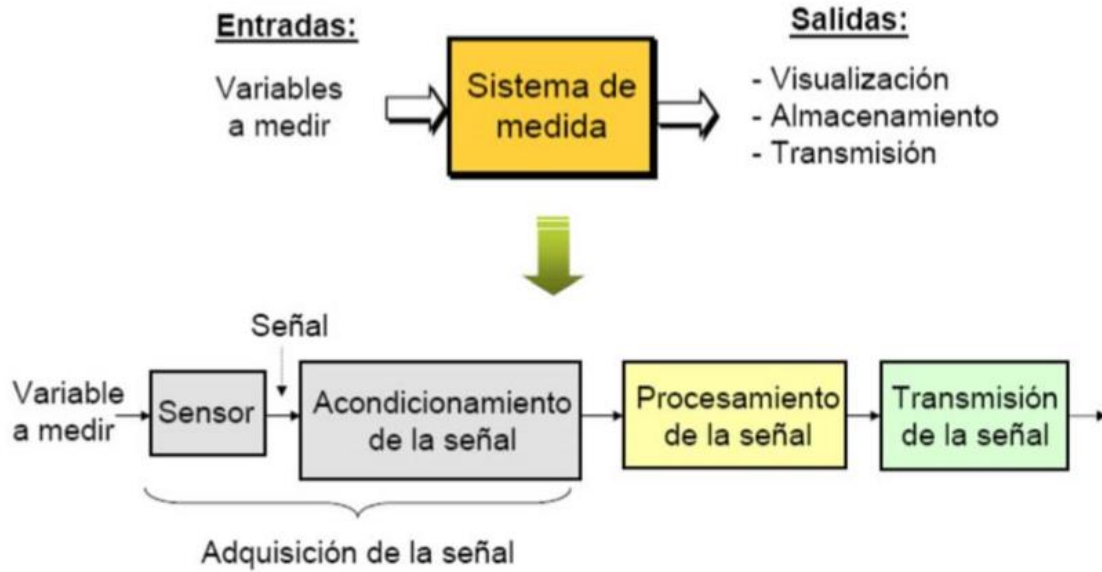


Figura 14: Funcionamiento teórico de los sensores.

3.5.4.2 Bloque adquisición de señal

La información de las magnitudes físicas es adquirida y convertida en una señal eléctrica. La variable del mundo físico es convertida en una señal eléctrica mediante un dispositivo sensor a fin de ser procesada. Con frecuencia la señal procedente del sensor tiene unas características (tensión y corriente muy pequeñas) que la hacen poco adecuada para ser procesada.

SENSOR (TRANSDUCTOR): Convierte una magnitud física (temperatura, presión, etc.) en una magnitud eléctrica (tensión, corriente) **ACONDICIONADOR DE SEÑAL:** Convierte la señal que proporciona el sensor en una magnitud eléctrica adecuada a las exigencias del dispositivo visualizador o del circuito que controla el proceso. Para ello los procesos pueden ser:

- Amplificación: incrementa el nivel de potencia de la señal
- Filtrado: elimina las componentes de la señal no deseada
- Linealización: obtiene una señal de salida que varíe linealmente con la variable que se desea medir
- Adaptación de niveles de señal: modifica los niveles de la señal de entrada a los niveles requeridos por el sistema de procesamiento.

3.5.4.3 Variables

La información (datos) de una determinada magnitud física se denomina variable. Cuando esta información es de naturaleza eléctrica, la variable se denomina señal. A continuación, se mostrarán los diferentes tipos de variables y la naturaleza a la que pertenecen:

Naturaleza de la variable	Tipo de variable
Mecánica	Fuerza, presión, aceleración, espesor, nivel, posición
Térmica	Temperatura, calor específico, entropía, etc.
Magnética	Campo magnético, flujo, permeabilidad magnética
Eléctrica	Tensión, corriente, carga, conductividad, resistencia
Óptica	Amplitud, fase, índice de refracción, emisividad
Biológica	Proteínas, hormonas, células, glucosa, colesterol
Química	Humedad, Composición, concentración, pH

Figura 15: Tipología de los sensores

3.5.4.4 Sensores y material

En el sistema central utilizaremos diferentes sensores de medida cuyas especificaciones vendrán adjuntadas en el anexo del documento.

- DHT22: Es un sensor de temperatura y humedad
- LDR: Es un sensor que mide la cantidad de luz en el ambiente

Sensor	Pin
Humedad	15
LDR	54
Temperatura	15

3.5.4.4.1 Sensor de Temperatura/Humedad

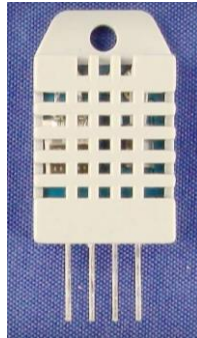


Figura 16: DHT22

Es un sensor de doble funcionalidad, tiene una salida de señal digital calibrada. Utiliza la técnica exclusiva de recolección de señal digital y humedad, garantizando su fiabilidad y estabilidad. Sus elementos sensores están conectados con un solo chip de 8 bits de computación.

Cada sensor de este modelo está calibrado de forma precisa en una cámara en temperatura y calibrado en cámara de calibración precisa y el coeficiente de calibración se guarda en el tipo de programa en la memoria OTP, cuando el sensor está detectando, se citará el coeficiente de la dicha memoria.

El tamaño pequeño y el consumo bajo y la distancia larga de la transmisión (20m) permiten DHT22 ser servido en todas las clases de duras ocasiones de aplicación.

3.5.4.4.2 Sensor de Luz (LDR)



Figura 17: LDR

Una LDR es una resistencia que varía su valor en función de la luz que incide sobre ella. Así de fácil. También se llama fotocélula o resistencia LDR. Cuando hay luz sobre la LDR su

resistencia disminuye, cuanto menos luz tenga la LDR mayor será su resistencia. Los valores de las resistencias con la luz dependen del tipo de la LDR. Aquí vemos la típica LDR que se usa en los talleres de tecnología. En esta LDR cuando hay luz prácticamente no tiene resistencia.

3.5.4.4.3 Otros sensores

Dada la versatilidad del sistema de Arduino para poder controlar o monitorizar una multitud de sensores, en este proyecto se podrían utilizar los siguientes sensores.

3.5.4.4.3.1 Acelerómetro de 3 ejes ADXL345



Figura 18: Acelerómetro de 3 ejes

Acelerómetro de 3 ejes de Adafruit basado en el ADXL345 de Analog Devices con interface digital I2C y SPI, monta un regulador de 3.3V para poder usarse a 3.3V y 5V, es decir, con cualquier placa Arduino.

El sensor cuenta con tres ejes de medida: X, Y y Z y los pines se pueden usar como I2C o SPI. Se puede ajustar la sensibilidad a 2g, 4g, 8g y 16g. El menor valor da más resolución en movimientos lentos, mientras que los valores altos se usan para movimientos rápidos.

Incluye una regleta de 9 pines para poder usarse en una placa de prototipado o en una placa perforada.

3.5.4.4.3.2 Ultrasonidos HC-SR04



Figura 19: Sensor Ultrasonidos

Sensor de distancia ultrasónico de alta precisión HC-SR04 para Arduino.

- Voltaje Operativo: 5V DC
- Intensidad: Menos de 2mA.
- Angulo de Detección: No más de 15°
- Distancia de Detección: 2cm~450cm.
- Precisión: Hasta 3mm

3.5.4.4.3.3 Sensor para flujo de líquidos Arduino



Figura 20: Sensor para flujo de líquidos

Este sensor de flujo líquido consiste en una válvula de cuerpo plástico, un rotor y un sensor de efecto hall, cuando el líquido fluye la velocidad del rotor varía en función del flujo del líquido, de esta forma el sensor de efecto hall emite el pulso correspondiente.

3.5.4.4.3.4 Sensor de humedad del suelo

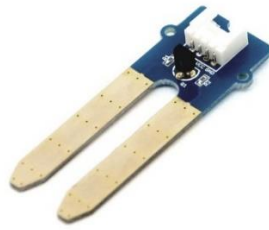


Figura 21: Sensor de humedad del suelo

Se trata de un sensor de humedad con salida analógica, da un valor de voltaje un función del nivel de humedad.

- Alimentación: 3.3 a 5V
- Voltaje de Salida: 0 a 4.2V DC
- Consumo: 35mA
- Dimensiones: 5.5x2.3x0,7cm

3.5.4.4.3.5 Sensor de luz ultravioleta UV GUVA-S12SD.



Figura 22: Sensor de luz ultravioleta

Este sensor analógico de luz ultravioleta usa un photodiodo ultravioleta que detecta el rango 240-370nm, es decir todo el espectro UVB y la mayor parte del UVA.

Al ser un sensor analógico su uso es muy sencillo metemos voltaje por el pin V+ (entre 2.2 y 5V DC), conectamos el pin V- al GND y realizamos la lectura del pin OUT, el voltaje de salida del mismo es 4.3 multiplicado por la intensidad del photodiodo en micro amperios (μA), así que si la intensidad del photodiodo es de $1\mu\text{A}$ ($9 \text{ mW}/\text{cm}^2$) el voltaje será de 4.3V. Para obtener el Índice UV dividimos el voltaje de salida entre 0.1V.

3.5.4.4.3.6 Sensor PIR de movimiento



Figura 23: Sensor PIR de movimiento

Detector de movimiento por IR compatible con Arduino con un rango de detección de 7 metros y un ángulo operativo de 120°.

- Distancia de Detección: 7 metros
- Angulo de Detección: 120°
- Alimentacion: 5V a 16V DC
- Salida Analógica: 3.3V (High) 0V (Low)
- Distancia de Detección: 7 metros
- Angulo de Detección: 120°
- Retardo: Ajustable
- Temperatura Operativa: -20°C a 70°C
- Dimensiones: 3.2x2.4x2.4cm

3.5.4.4.3.7 Sensor de peso FSR

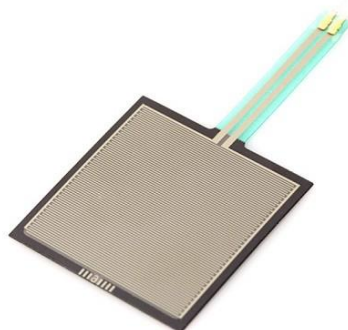


Figura 24: Sensor de Peso

Los sensores de fuerza resistivos son sensores que pueden detectar la presión, el deslizamiento y el peso. Son muy económicos y sencillos de usar con plataformas como Arduino. Un sensor FSR es básicamente un resistencia que cambia su valor resistivo, en Ohmios, en función de la presión que se ejerce sobre él. En este caso se trata de un sensor Interlink modelo 406.

Dimensiones:

- Largo: 44mm
- Ancho: 44mm
- Altura: 0.42mm
- Peso: 1.12gr

3.5.5 Actuadores

Los actuadores empleados en esta parte del sistema y los pines de conexión son los siguientes:

Actuador	Pin
Luz Habitación	22
Persianas	24
Mesa De Trabajo	28
Aire Acondicionado Calor	32
Aire Acondicionado Frio	30
Luz Cocina	40
Tostadora	34
Campana	36
Lavavajillas	38

3.5.6 Firmware

La programación del Sistema servidor está dividida en 5 grandes grupos:

3.5.6.1 Led

En esta parte de programación tendremos las siguientes secciones:

3.5.6.1.1 Activación de librerías

En esta parte del programa activaremos las diferentes librerías que nos harán falta para una posible y correcta ejecución del programa.

3.5.6.1.2 Conexión y seguridad

Aquí se programa la IP interna del servidor (Red Ethernet), la IP externa, y el password necesario para poder acceder al sistema de sonorización con el dispositivo Android.

3.5.6.1.3 Programación de los diferentes sensores

En esta parte se programa todo lo relacionado con el sensor de temperatura DHT22, el reloj, el sensor de luz o LDR, el sensor de humedad el cual también es el DHT22.

3.5.6.1.4 Programación del sistema de comunicación por radiofrecuencias

Esta es la parte del programa en donde configuramos el emisor de radiofrecuencias para emitir los códigos pertinentes dependiendo de la orden recibida por el dispositivo Android

3.5.6.1.5 Cambio horario

Se realizará un algoritmo para que dependiendo del día del mes en el que estemos, sume una hora o reste una hora para adaptar el sistema domotico a la hora social.

3.5.6.1.6 Actuadores

Se configura las diferentes salidas para los actuadores correspondientes dependiendo de la orden de llegada del sistema de comunicaciones realizada desde el dispositivo Android.

3.5.6.1.7 Debug_Mode

Se Activará el Debug_Mode. Este es un sistema donde podremos realizar pruebas de nuestra programación en modo "Run" para corroborar su correcto funcionamiento.

3.5.6.2 Common_Functions

En esta función están pre-programados las funciones esenciales de la programación como el control del wifi, el control del sistema de comunicación vía Ethernet, la parte analógica del movimiento de las persianas, configuración de los diferentes timers, interrupciones, configuración del reloj del sistema, la posibilidad de graficar los valores de las gráficas, el guardado en la SD de los valores de los sensores y el sistema de comunicación tipo red entre los dispositivos Arduino.

3.5.6.3 Excontrol_def

En esta función están programados los diferentes tipos de opciones que utilizaremos en el dispositivo Android para poder afiliarlos a las diferentes señales que utilizaremos para nuestro sistema de sensorización.

3.5.6.4 InfraRedFunction

En esta función esta pre-programado los infrarrojos por si en un futuro quisiésemos incorporarlos a nuestro sistema de sensorización.

3.5.6.5 Mhz433

En esta función están pre-programados todos los parámetros necesarios para el sistema de comunicación por radiofrecuencias para un óptimo funcionamiento del mismo, el cual utilizaremos en la función de programa principal (Led).

3.5.6.6 PrintLDC

En esta función esta pre-programado el LCD por si en un futuro quisiésemos incorporar una a nuestro sistema de sonorización.

3.5.6.7 Activación de librerías.

En esta parte del programa activaremos la librería RCwitch.h la cual nos permite el uso de las radiofrecuencias

3.5.6.8 Una función de inicialización

En esta parte del programa inicializaremos los pines a utilizar, y los baudrates de comunicación.

3.5.6.9 Bucle del programa

En esta parte del programa crearemos un bucle en el cual estaremos escuchando con el receptor de radiofrecuencia de forma que si llega el código adecuado, se activara el pin en cuestión que permitirá cerrar el circuito y activarse una carga determinada.

3.6 Nodo Esclavo

Esta parte del sistema se encarga de recibir las órdenes del sistema servidor y activar los actuadores pertinentes dependiendo de la orden recibida desde el dispositivo de Android. La placa que se encargará de ello será la Arduino UNO.

3.6.1 Diagrama de bloques

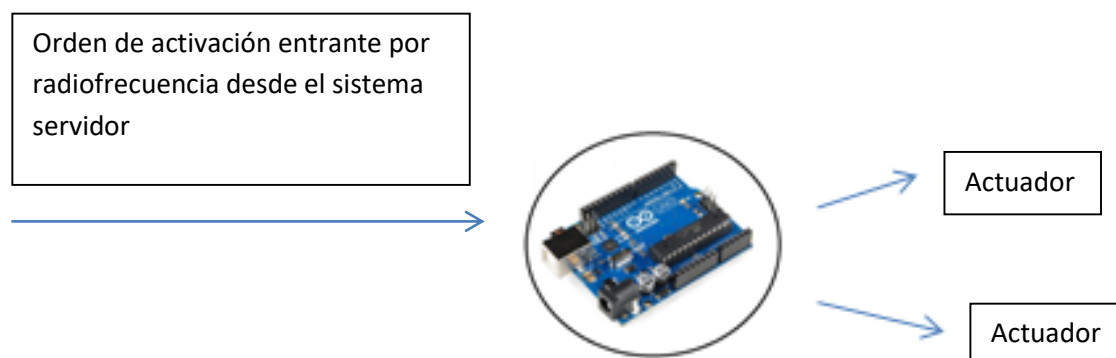


Figura 25: Diagrama de bloques Nodo esclavo

3.6.2 Placa Uno

Las tarjetas utilizadas en dicho apartado será un Arduino UNO, la cuales consta de las siguientes especificaciones:

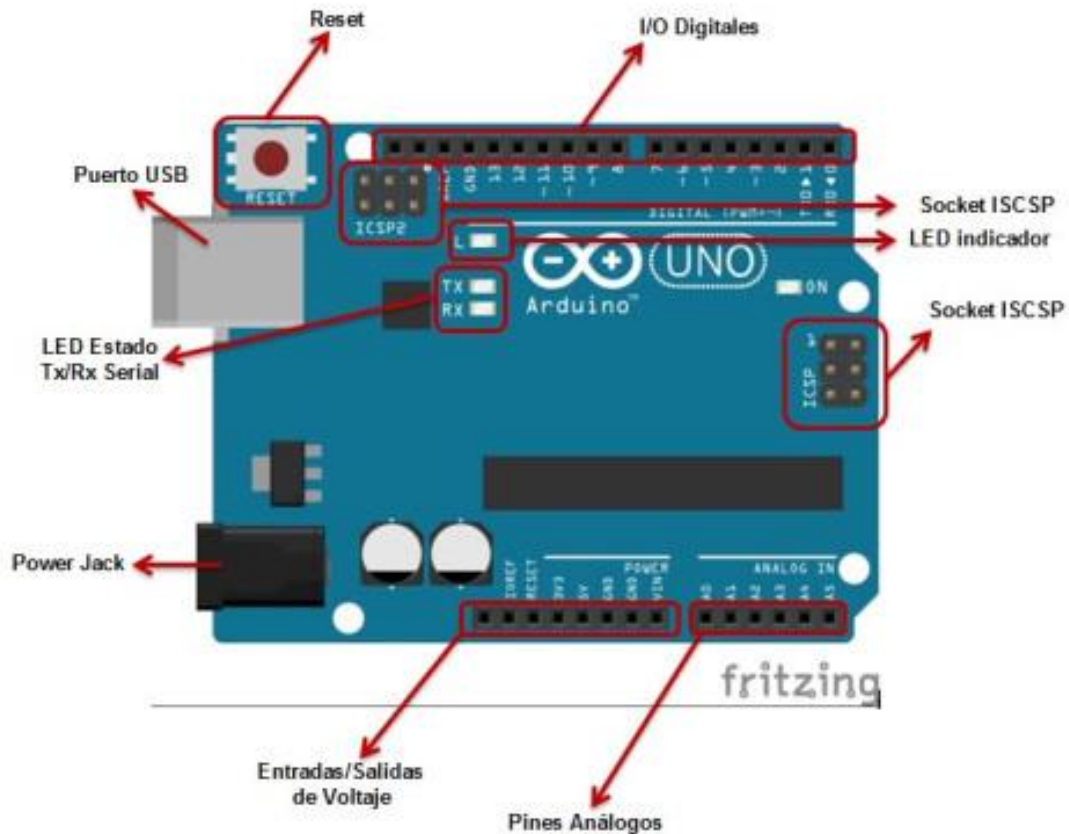


Figura 26: Placa Arduino UNO

- **Microcontrolador:** ATmega328
- **Voltaje Operativo:** 5v
- **Voltaje de Entrada (Recomendado):** 7 – 12 v
- **Pines de Entradas/Salidas Digital:** 14 (De las cuales 6 son salidas PWM)
- **Pines de Entradas Análogas:** 6
- **Memoria Flash:** 32 KB (ATmega328) de los cuales 0,5 KB es usado por Bootloader.
- **SRAM:** 2 KB (ATmega328)
- **EEPROM:** 1 KB (ATmega328)
- **Velocidad del Reloj:** 16 MHZ.

3.6.3 Comunicaciones

3.6.3.1 Recepción de la orden por radiofrecuencia

Este nodo tiene conectado un chip receptor de radiofrecuencia ARM / MCU WL el cual tras recibir una orden, la placa la procesa activando el actuador pertinente.

3.6.4 Actuadores

A continuación, se mostrará una tabla donde aparecen los diferentes actuadores que activara este nodo, conjunto a los pines de conexión y el código pertinente de cada uno.

Actuador	Código	Pin
Puerta garaje	5000	3
Toldo	5001	4
Luz jardín	5002	5
Riego automático	5003	6
luz salón	5004	7
Persianas	5005	8
Chimenea	5006	9
Ventilador	5007	10
Luz baño	5008	11
Calefactor baño	5009	12

3.6.5 Firmware

En esta parte de programación tendremos las siguientes secciones:

3.6.5.1 Activación de librerías.

En esta parte del programa activaremos la librería RCwitch.h la cual nos permite el uso de las radiofrecuencias

3.6.5.2 Una función de inicialización

En esta parte del programa inicializaremos los pines a utilizar, y los baudrates de comunicación.

3.6.5.3 Bucle del programa

En esta parte del programa crearemos un bucle en el cual estaremos escuchando con el receptor de radiofrecuencia de forma que si llega alguno de los códigos adecuados, se activara el pin en cuestión que permitirá cerrar el circuito y activarse una carga configurada con el código determinado.

3.7 Controlador inalámbrico de activación de carga (ciac)

El CIAC es un dispositivo cuya funcionalidad es la de poder activar cualquier carga conectada a un enchufe controlándola con el dispositivo Android

3.7.1 Diagrama de bloques

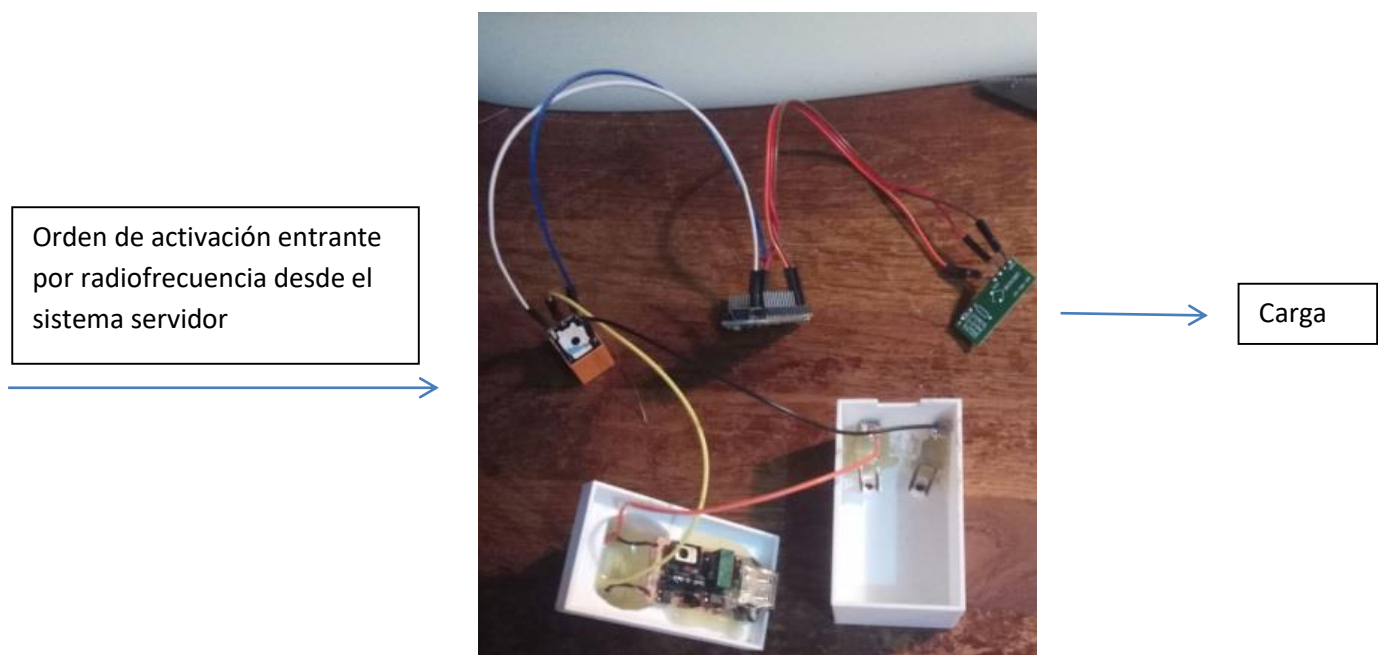


Figura 27: Sensor de luz ultravioleta

3.7.2 Placa Arduino

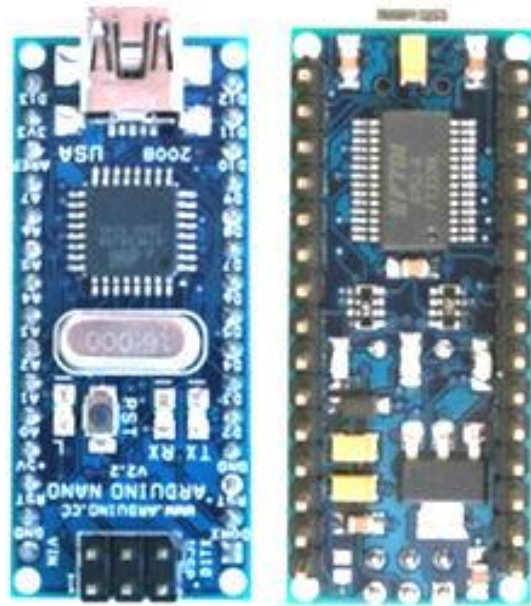


Figura 28: Placa Arduino NANO

- **Microcontrolador** ATmega328 con cargador de inicio programado.
- **Tensión de entrada (recomendada):** +7 a + 12 V.
- **Tensión de entrada (límites):** +6 a + 20 V.
- 14 pines GPIO (de los que 6 ofrecen salida PWM).
- 6 pines de entrada analógica.
- **Corriente DC** por pin de E/S: 40 mA.
- **Memoria Flash** de 32 KB (2 KB para cargador de inicio).
- **SRAM** de 2 KB.
- **EEPROM** de 1 KB.
- Admite comunicación serie IC.
- **Frecuencia de reloj:** 16 MHZ.
- **Dimensiones:** 0,73" x 1,7".

3.7.3 Comunicaciones

La comunicación del CIAC con el Sistema servidor se realiza por medio de radiofrecuencias. El servidor envía una orden de activación de forma que el CIAC permite el paso de la corriente eléctrica.

3.7.3.1 Recepción de la orden por radiofrecuencia

Este Controlador Inalámbrico De Activación De Carga tiene conectado un chip receptor de radiofrecuencia ARM / MCU WL el cual, tras recibir una orden, la placa la procesa activando el actuador pertinente.

3.7.4 Firmware

En esta parte de programación tendremos las siguientes secciones:

3.7.4.1 Activación de librerías.

En esta parte del programa activaremos la librería RCwitch.h la cual nos permite el uso de las radiofrecuencias

3.7.4.2 Una función de inicialización

En esta parte del programa inicializaremos los pines a utilizar, y los baudrates de comunicación.

3.7.4.3 Bucle del programa

En esta parte del programa crearemos un bucle en el cual estaremos escuchando con el receptor de radiofrecuencia de forma que, si llega el código adecuado, se activará el pin en cuestión que permitirá cerrar el circuito y activarse una carga determinada.

4 Validación

Como prueba de que lo realizado en este proyecto además de ser factible, es funcional, a continuación, marcaremos las diferentes etapas de la señal mostrando cada parte para un mejor entendimiento del proceso.

4.1 Dispositivo Smartphone

Desde la aplicación de Ex Control del dispositivo móvil, podremos observar los diferentes actuadores que podremos comandar, así como la visualización de los sensores como el sensor de temperatura DHT22 el cual en la siguiente imagen nos marca 33,2°C

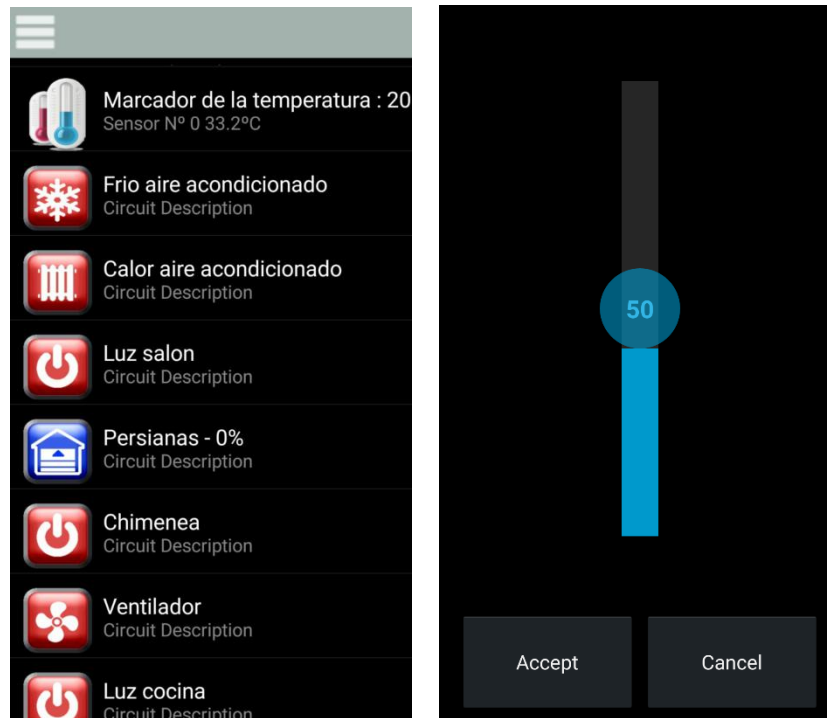


Figura 29: Dispositivo Android

Además, en esta aplicación podremos también observar la variación de los parámetros de los diferentes sensores mediante la graficación de los mismos.



Figura 30: Graficado de los sensores

Una vez activado el actuador pertinente, la señal viaja hasta el router por una IP de red local o vía internet hasta el router previamente configurado.



Figura 31: Router Utilizado

Tras llegar al router, la orden viajará vía Ethernet hasta el Sistema servidor, el cual se encargará de leer y procesar la orden.

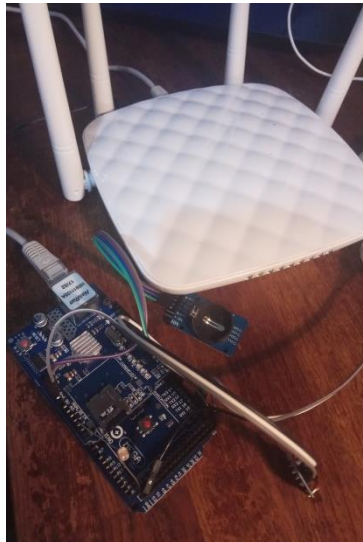


Figura 32: Conexión del Sistema Servidor

En el supuesto caso de que la orden fuese activar un actuador del nodo esclavo o el CIAC, el Sistema servidor enviaría la orden pertinente vía radiofrecuencia a la placa correspondiente para que activase el actuador deseado el cual, en este caso, al tratarse de una maqueta vendría referenciado con un led.



Figura 33: Conexión del Sistema Servidor

5 Conclusiones

Gracias a este TFM he podido aplicar e incrementar mis conocimientos sobre el funcionamiento y la aplicación de los sensores y de la plataforma Arduino, además de mejorar mis conocimientos sobre la programación en C.

Este tipo de instalaciones domóticas podrían aplicarse en otros sectores como la monitorización de procesos industriales ya que los actuales PLCs instalados, además de ser mucho más costosos, son sistemas cerrados con opciones considerablemente más limitadas y que consumen más energía que estos sistemas alternativos.

Otro tema a tener en cuenta es el ahorro energético, ya que gracias a la domótica el riego se activará cuando el césped lo necesite y no de forma rutinaria cada día.

Dentro del diseño domótico existe un algoritmo de seguridad que enciende las luces de la vivienda de forma aleatoria durante 10-15 min para aparentar que la vivienda está habitada.

La programación está protegida con contraseña para dificultar la intrusión de cualquier individuo indeseado en el sistema de programación.

6 Futuras Líneas De Trabajo

Existen varias formas de mejorar la domótica de este proyecto.

Primeramente, añadiría una pantalla LCD que me informase de los valores de los diferentes sensores.

Añadiría un sistema de seguridad vía SMS, el cual es imposible de hackear, de forma que, con enviar un SMS, se activaría el router y, por lo tanto, el sistema domótico. Por supuesto tendría que añadir un actuador de apagado del router en la programación.

Otro sensor que agregaría al sistema de sensorización sería el sensor de gas MQ-35.

Para completar el tipo de control sobre la vivienda, también programaría un control por Infrarrojos a dispositivos como: Televisión, aire acondicionado, etc...

7 Bibliografía

<https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT22.pdf>

<https://www.luisllamas.es/comunicacion-inalambrica-en-arduino-con-modulos-rf-433mhz/>

<http://panamahitek.com/arduino-mega-caracteristicas-capacidades-y-donde-conseguirlo-en-panama/>

<http://www3.gobiernodecanarias.org/medusa/ecoblog/ralvgon/files/2013/05/Caracter%C3%ADsticas-Arduino.pdf>

<http://electronica-teoriaypractica.com/caracteristicas-arduino-nano-328/>

<https://programarfacil.com/blog/arduino-blog/reloj-con-arduino-rtc/>

<https://aprendiendoarduino.wordpress.com/2016/07/04/ethernet-shield/>

<https://www.abcelectronica.net/productos/reles/5vdc/>

Temario del master de electrónica de la asignatura de Sistemas electrónicos avanzados.

<http://sherlin.xbot.es/microcontroladores/introduccion-a-los-microcontroladores/que-es-un-microcontrolador>

<http://www.electronicaestudio.com/microcontrolador.htm>

<http://www.dc.fi.udc.es/~so-grado/current/Varios/CursoC.pdf>

<http://www.cetronic.es/sqlcommerce/disenos/plantilla1/seccion/Catalogo.jsp?idIdioma=&idTienda=93&cPath=>