Titulación : Grado Ingeniería Electrónica Industrial y Automática

CASA DOMOTICA CON ARDUINO

AUTOR : Oscar Sanclemente Carretero

TUTOR : Roberto Capilla Lladró

20 de Julio 2016

Índice

La idea de tener una casa domótica es algo q desde pequeño siempre me llamo la atención personalmente, por aquel entonces no sabía cómo se podrían hacer todas esas cosas ya que las veía como una película del futuro.

Tras muchos años dedicándome a la electricidad y otros tantos a formarme en electrónica, informática etc...He podido llegar a cumplir este sueño y así poder realizar este proyecto.

Resumen:

La base de este proyecto es la domótica, que se ha dividido en tres bloques diferenciados:

Por un lado el control de llenado de un tanque de agua, para un acuario y el control del nivel de sal de la descalcificadora ya que no posee display y no puede transmitir información al usuario .Toda esta parte del primer bloque manda la información a tiempo real a una pantalla lcd gobernada por un arduino Uno y este la obtiene de varios sensores.

Por otro lado se ha creado un servidor web con un arduino mega y un escudo de Ethernet, para comunicarnos con este arduino que es el que posee la información de la página web a publicar introducimos la ip dinámica de mi router, una vez dentro nos aparecerá la interface de los elementos a accionar de la vivienda.

Por último, se ha creado un control de acceso a la vivienda en el cual nos identificamos para armar y desarmar la alarma con un tag de rfid de 13,5 Mhz y a través de un módulo de voz nos dirá en qué estado se encuentra el control de acceso y nos despedirá o dará la bienvenida mediante voz.

1. Introducción:

1.1 Antecedentes de la Domótica

La domótica se inició a comienzo de la década de los 70, cuando aparecieron las primeras pruebas en pisos piloto.

Ya en los 80 cuando se empezaron a comercializar los integrados, fue cuando la domótica se empezó a expandir al hogar.

Allí es cuando la domótica consigue integrar dos sistemas (el eléctrico y el electrónico) en pos de la comunicación integral de los dispositivos del hogar.

El desarrollo de la tecnología informática permite la expansión del sistema, sobre todo en países de vanguardia como Estados Unidos, Alemania y Japón.

Acorde a los cambios, el auge de la informática hogareña permite incorporar en los edificios el Sistema de Cableado Estructurado (SCE), que facilita la conexión de terminales y redes. Así, estos edificios reciben el nombre de “inteligentes”, por su automatismo al servicio del propietario.

El boom de estos rascacielos de oficinas comerciales fue de gran impacto. La domótica permitía lograr una eficiencia inédita para el servicio de dispositivos.

El primer programa que utilizó la domótica fue el Save. Creado en Estados Unidos en 1984, permite lograr eficiencia y bajo consumo de energía en los sistemas de control de edificios inteligentes.

Estas instalaciones regían bajo el sistema X-10, protocolo de comunicaciones que opera a través del accionar de un control remoto. Desarrollado en 1976 por Pico Electronics (Escocia), sigue siendo la tecnología más utilizada dentro de la domótica. Al transmitir datos por líneas de baja tensión, la relación costo-beneficio sigue siendo la mejor opción en el rubro.

Implantada desde hace más de treinta años, la domótica ha progresado a gran escala desde que se desarrollaron las redes informáticas de comunicación, ya sea por sistema cableado o vía Wi-Fi.

El avance tecnológico vino a suplir las falencias de los comienzos, ya que permite integrar de manera eficiente todos los dispositivos tecnológicos de una casa. Con el fin de la década del ’80 las tecnológicas de un comienzo, destinadas a fines comerciales, comienzan a llegar a los hogares.

Irrumpe la era de la TIC (la tecnología de informática y comunicaciones), que posibilita entender una forma más realista de comprender la instalación domótica en casa.

En la actualidad hay una oferta consolidada en torno a los servicios de domótica. Nuevos protocolos permiten un desarrollo que en un principio era impensado.

Sistemas de desarrollo 2.0 como el ZigBee permiten conformar un protocolo inalámbrico de comunicación domótica. Al requerir una baja tasa de envío de datos, es en la actualidad uno de los protocolos más requeridos para las casas “inteligentes”, ya sea en sensores de movimiento, detectores de humo y otras funciones de seguridad en el hogar. Con la domótica aplicada a la automatización hogareña se mejora en seguridad, confort y ahorro energético, aspectos muy observados por los poseedores de estos sistemas. La llegada de Internet a gran velocidad provocó un giro favorable para su desarrollo.

1.2 Antecedentes de Arduino

Arduino fue inventado en el año 2005 por el entonces estudiante del instituto IVRAE Massimo Banzi, quien, en un principio, pensaba en hacer Arduino por una necesidad de aprendizaje para los estudiantes de computación y electrónica del mismo instituto, ya que en ese entonces, adquirir una placa de micro controladores eran bastante caro y no ofrecían el soporte adecuado; no obstante, nunca se imaginó que esta herramienta se llegaría a convertir en años más adelante en el líder mundial de tecnologías DIY (Do It Yourself). Inicialmente fue un proyecto creado no solo para economizar la creación de proyectos escolares dentro del instituto, sino que además, Banzi tenía la intención de ayudar a su escuela a evitar la quiebra de la misma con las ganancias que produciría vendiendo sus placas dentro del campus a un precio accesible (1 euro por unidad).

El primer prototipo de Arduino fue fabricado en el instituto IVRAE. Inicialmente estaba basado en una simple placa de circuitos eléctricos, donde estaban conectados un micro controlador simple junto con resistencias de voltaje, además de que únicamente podían conectarse sensores simples como leds u otras resistencias, y es más, aún no contaba con el soporte de algún lenguaje de programación para manipularla.
Años más tarde, se integró al equipo de Arduino Hernando Barragán, un estudiante de la Universidad de Colombia que se encontraba haciendo su tesis, y tras enterarse de este proyecto, contribuyó al desarrollo de un entorno para la programación del procesador de esta placa: Wiring, en colaboración con David Mellis, otro integrante del mismo instituto que Banzi, quien más adelante, mejoraría la interfaz de software.
Tiempo después, se integró al "Team Arduino" el estudiante español David Cuartielles, experto en circuitos y computadoras, fue quien ayudó a Banzi a mejorar la interfaz de hardware de esta placa, agregando los microcontroladores necesarios para brindar soporte y

memoria al lenguaje de programación para manipular esta plataforma.
Más tarde, Tom Igoe, un estudiante de Estados Unidos que se encontraba haciendo su tesis, escuchó que se estaba trabajando en una plataforma de open-source basada en una placa de micro controlador pre ensamblada. Después se interesó en el proyecto y fue a visitar las instalaciones del Instituto IVRAE para averiguar en que estaban trabajando. Tras regresar a su país natal, recibió un e-mail donde el mismo Massimo Banzi invitó a Igoe a participar con su equipo para ayudar a mejorar Arduino. Aceptó la invitación y ayudó a mejorar la placa haciéndola más potente, agregando puertos USB para poder conectarla a un ordenador. Además, él le sugirió a Banzi la distribución de este proyecto a nivel mundial.
Cuando creyeron que la placa estaba al fin lista, comenzaron su distribución de manera gratuita dentro de las facultades de electrónica, computación y diseño del mismo instituto. Para poder promocionar el proyecto Arduino dentro del campus, tuvieron que consultar con un publicista que más adelante paso a formar parte del equipo Arduino: Gianluca Martino, quien la distribuyo dentro del instituto y promocionándola a algunos conocidos y amigos suyos. Al ver su gran aceptación por parte de los alumnos y maestros y tomando en cuenta el consejo de Igoe, pensaron en su distribución a nivel mundial, para lo cual contactaron a un amigo y socio de Banzi, Natan Sadle, quien se ofreció a producir en masa las placas tras interesarse en el proyecto.
Un breve tiempo más tarde, al ver los grandes resultados que tuvo Arduino y las grandes aceptaciones que tuvo por parte del público, comenzó a distribuirse en Italia, después en España, hasta colocarse en el número uno de herramientas de aprendizaje para el desarrollo de sistemas autómatas, siendo además muy económica en comparación con otras placas de microcontroladores .

1.3 Ventajas e inconvenientes de la Domótica

Los beneficios que aporta la Domótica son múltiples, y en general cada día surgen nuevos. Por ello creemos conveniente agruparlos en los siguientes apartados:

a) El ahorro energético gracias a una gestión tarifaria e "inteligente" de los sistemas y consumos.

b) La potenciación y enriquecimiento de la propia red de comunicaciones.

c) La más contundente seguridad personal y patrimonial.

d) La tele asistencia.

e) La gestión remota (v.gr. vía teléfono, radio, internet, Tablet, consola juegos, etc.) de instalaciones y equipos domésticos.

f) Como consecuencia de todos los anteriores apartados se consigue un nivel de confort muy superior. Nuestra calidad de vida aumenta considerablemente.

Se podría decir que las desventajas son realmente pocas con respecto a las ventajas pero se pueden mencionar las siguientes:

-El precio aún es demasiado alto.

-Al ser relativamente nueva su aplicación se pueden experimentar fallos en los sistemas, etc.

-Se puede producir el aislamiento del usuario.

-Se puede dar un entorpecimiento del usuario, dependiendo del grado de automatización del sistema

1.4 Ventajas e inconvenientes de Arduino frente a otros microcontroladores

Se dice, que cuando comenzaron a surgir los compiladores en C para sistemas embebidos, los ávidos programadores en Ensamblador (ASM) rechazaron el uso de un lenguaje nuevo en un microcontrolador. ¿Por qué? Quizás por el nivel de optimización menor que ASM, quizás por el innato rechazo humano al cambio, quizás por la desconfianza de un nuevo sistema o quizás porque no era tan seguro programar en C, entre muchas otras cosas más. Las razones previamente mencionadas son en parte ciertas, sin embargo, cualquier programador de sistemas embebidos actual no puede rechazar o negar el impacto que ha llegado a tener el lenguaje C/C++ a pesar del rechazo que tuvo en un principio. Hoy en día C/C++ es el lenguaje más utilizado para programar sistemas embebidos

Ventajas de microcontroladores

Aprender a programar un microcontrolador en C/C++ modificando sus registros internos, fusibles y revisando su hoja de datos, obliga al usuario a conocer mejor el hardware del dispositivo, lo que da una mayor flexibilidad y optimización. Esto es ideal cuando se buscan definir parámetros finos en nuestro programa los cuales pueden ser vitales en algunas aplicaciones. Como el cálculo de tiempos muertos, retardos precisos, aprovechamiento de memoria, etc.

Programar con un microcontrolador y no con una tarjeta de desarrollo, se aprende un poco más acerca de los aspectos analógicos de la electrónica. Esto nos da una ventaja a la hora de crear tarjetas para aplicaciones específicas para nuestros microcontroladores. Como el configurar el capacitor de filtraje (o bypass) correcto para nuestro sistema, el uso de resistencias de pull-up o pull-down, etc.

Usar microcontrolador nos da mayor flexibilidad en los proyectos dedicados. Podemos elegir entre una gama de dispositivos mucho más alta que las diferentes tarjetas Arduino. Parámetros comunes en dispositivos son, que corra a frecuencias mayores o que soporte mayores temperaturas, que tenga más salidas de PWM, que tenga protocolo CAN, etc.

Si se desconoce el Hardware el precio juega un papel muy importante en proyectos donde se requiere la implementación de nuestro programa para escalas mayores a 1,000 piezas, donde veremos que comprar Arduinos no es tan conveniente en precio y tamaño. Sin embargo si podríamos en su debido tiempo solo cargar nuestro programa de Arduino al puro micro.

Ventajas de arduino

Una curva de aprendizaje mucho más rápida. Podemos invertir el tiempo invertido en nuestros proyectos para desarrollar otras aplicaciones o aprender cosas nuevas. El alcance de un proyecto en un tiempo delimitado, partiendo de cero, indudablemente con Arduino llegaremos más lejos. No quiere decir que sea más óptimo, sin embargo, completamente funcional seguramente.

Una gran comunidad con mentalidad “Open Source”. ¿Para qué hacemos algo que alguien más ya hizo? Recordemos: “Solos llegamos más rápido pero juntos llegamos más lejos”. Y aprovechamos para sugerir que compartan sus resultados y/o avances, bibliotecas, programas, ejemplos, etc.

Un entorno de desarrollo minimalista, no es precisamente una ventaja para un proyecto profesional o parar analizar miles de líneas de código, sin embargo, si lo es para la mayoría de desarrollos a los que está orientado Arduino. Podemos descargar el Arduino IDE y comenzar a programar en menos de 20 minutos, en algunos IDEs (si no es que en la mayoría) necesitamos descargar el IDE y además el compilador en C, como el C18 o C30 para los PICs y dsPICs respectivamente, o el WinAVR para los AVRs. Ademas del aspecto de licencias que eso no lo tocaremos como ventaja o desventaja.No necesitas programador! aunque igual puedes instalar el bootloader con casi cualquier microcontrolador, sin embargo, Arduino lanzo a las masas el uso del bootloader y sigue siendo una de las cosas más cómodas a la hora de usar Arduino. Son pocas las tarjetas que hacen uso de esta tecnología, la mayoría de tarjetas de desarrollo acoplan un programador ICSP embebidos en sus tarjetas de prueba o desarrollo.

1.5 Objetivos

1.5.1 Objetivos Generales

Los objetivos generales que se presentaron al comenzar el proyecto fueron el conocimiento y entorno de los microcontroladores , y la posibilidad de poder usarlos para beneficio de la vida cotidiana como bien pueden ser para la automatización en viviendas.

Estos objetivos en primer lugar se cumplieron a base de la documentación encontrada en diferentes webs, blogs, foros y libros dedicados a todo este entorno.

1.5.2 Objetivos Específicos

Los objetivos específicos han sido todo un reto, ya no por la parte de lo q es el desarrollo del proyecto sino también por parte de solucionar todos los problemas que se han ido sucediendo a la hora del montaje real.

Por un lado se ha tenido el reto de compaginar en este proyecto tanto software como hardware.

Por la parte de software se ha creado 2 programas completos desde cero en el entorno de programación c++ que es el lenguaje que utiliza el IDE de arduino, el tercer programa que es el del servidor web , se ha utilizado un programa de ejemplo encontrado en internet el cual se ha tenido que modificar completamente para poder cubrir nuestras necesidades.

En la parte de hardware , se ha utilizado muchos aspectos adquiridos durante el grado, el objetivo era introducir elementos que hemos conocido como bien pueden ser ,resistencias, transistores , un integrado de puertas nor, relés optoaclopadores , soldaduras etc. y unificar todo junto al software para que al unísono se pudiese cumplimentar el proyecto en su totalidad.

2.Metodologia

2.1 Requerimientos del sistema

En este apartado se va a explicar con detenimiento los tres bloques del proyecto y todos los elementos requeridos para poder llevarlo a cabo.

2.1.1 Control llenado y vaciado del tanque

Esta parte va junto con la del control de sal de la descalcificadora pero se separa para ser un poco más específico.

La idea era crear un tanque conectado a un filtro de osmosis el cual cuando estuviese vacío , por medio de una electroválvula se abriese el caudal y cuando el agua llegase a su máximo mediante un sensor de llenado cortase la electroválvula.

La electroválvula funciona con 12 v por lo que mediante una fuente conmutada y un relé optoaclopador la gobernamos a través de nuestro arduino.

Toda esta información se nos imprime en una pantalla lcd colocada en una caja estanca la cual nos muestra en tiempo real la información de nuestro sistema.

Para vaciar el tanque , se introdujo una bomba de acuario dentro de este y mediante un pulsador colocado en la caja de la lcd se procede al vaciado con solo dar un pulso, ya que cuando se creó el programa se taro el tiempo de llenado de una garrafa de 25 litros , y cuando esta se llena el sistema desconecta la bomba de agua.

Con la bomba pasa parecido a la electroválvula, estas bombas funcionan a 230 v ac , entones se hizo lo mismo que con la electroválvula, se conectó a un relé y esté conectado a la red eléctrica de casa.

2.1.2 Control de nivel de sal en una descalcificadora

Esta parte del primer bloque surge de la falta de información del nivel de sal del modelo de descalcificadora con el que se cuenta, debido a esto se le acoplo un sensor de ultrasonidos el cual nos manda la información a nuestro arduino y este dependiendo de la distancia que hay desde la tapa de la descalcificadora a las piedras de sal nos indica si falta sal o no en nuestra lcd.

En concreto ya que es una descalcificadora con poca capacidad , apenas un saco de sal , se taro en el software de tal manera que cuando la distancia del ultrasonido fuese mayor a 30 cm nos saliese en la pantalla que faltaba sal a nuestro sistema, esta parte es solo de información para el usuario y no hay nada automatización.

2.1.3 Servidor web Arduino

En este segundo bloque del proyecto lo que se quería era controlar una serie de circuitos y componentes del hogar de forma automática a través de una interface.

En primer lugar se pensó en hacerlo mediante un módulo de bluetooth , pero se probó el modulo y aparte de que había que depender de una app ya creada para hacer de interface no era muy estable al tener tabiques en casa ya que se distorsionaba la señal y el alcance de transmisión de datos era limitado , aproximadamente unos 10 m dependiendo de fabricantes.

Descartando el bluetooth , el siguiente paso fue pensar en usar la shield de Ethernet para trabajar solo como red local por wiffi, pero claro el inconveniente principal era que solo podríamos interactuar con la casa dentro de esta o en sus inmediaciones hasta donde nos llegase la señal del router . Entonces fue cuando por fin nos decantamos en hacer un servidor web para poder trabajar tanto dentro de casa con y sin wiffi y fuera de casa por medio de la red 4g.

Lo primero que se hizo fue documentarnos sobre html ya que no teníamos conocimiento alguno, después se buscó en internet un programa ya creado el cual lo tuvimos que repasar para entender que era lo que hacía, seguidamente encontrando los puntos como cabeceras, marcos, textos, colores, botones ,etc. lo que se hizo fue modificarlo todo hasta llegar a la interface que queríamos para nuestro proyecto.

Para acceder a nuestro sistema, lo que hacemos es introducir en la barra del navegador la ip dinámica de nuestro router y así nos aparecerá en nuestro dispositivo ya sea el móvil, Tablet o pc la interface que hemos creado.

Una vez en nuestra interface ya podemos interactuar con los circuitos y dispositivos conectados presionando el botón de on u off, la peculiaridad de hacer este servidor y que es una por las cuales nos decantamos por este sistema es que guarda la información en la memoria Eprom de nuestro arduino , esto significa, que si conectamos un circuito y salimos de la página o se nos quedase sin tensión la placa seguiría esa acción conectada , claro la parte de la falta de tensión depende del tiempo que transcurra hasta la vuelta de esta ya que está limitado por la energía acumulada en los condensadores de la placa.

2.1.4 Control de iluminación

En la parte del control de iluminación, lo que controlamos es el encendido y apagado de la luz del salón , una vitrina que hay en el salón con una lámpara led y una tira de led que hay en la bancada de la cocina.

Cuando presionamos el on de cualquier circuito de estos 3, la placa nos dará una salida high de 5v la cual llevamos al relé correspondiente para conectar el circuito.

El circuito del alumbrado del salón son unas pastillas de led unidas entre sí en paralelo por lo que su tensión de trabajo es de 12v, y la de la vitrina es de 230 ya que la lámpara lleva incorporada su transformador y rectificador y la tira de led se alimenta a 230 v por el mismo motivo que la vitrina ,con esto lo que se quiere exponer es que se ha estado trabajando en todo momento con diferentes tipos de tensiones complicando aún más el trabajo de instalación real.

Por otro lado se tuvo una serie de problemas, ya que la placa al conectarla al módulo de relés cuando se accionaba uno o varios caía la tensión y se nos bloqueaba el servidor , la forma de solucionarlo fue aislando el módulo de relés con su propia fuente independiente de la que alimenta al arduino mega y unificando sus masas , y la señal de salida de cada circuito de la placa en lugar de llevarla directamente a su correspondiente relé . La conectamos a la base de un transistor NPN, el emisor a su relé y el receptor a la fuente que alimentaba al módulo de relés.

Cabe anotar que esto lo realizamos en todos los circuitos.

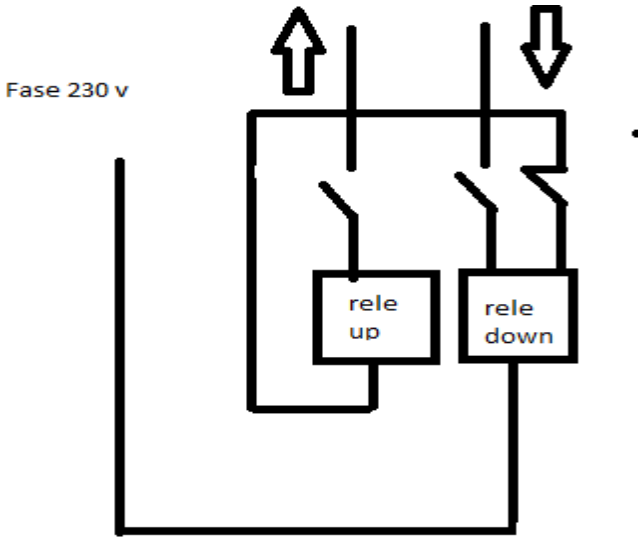
2.1.5 Automatización de persianas

Este bloque es uno más que se controla a través de la interface del servidor que hemos creado, lo que sí que se tuvo que preinstalar fue una persiana con motor.

Cuando accionamos este circuito tenemos 2 opciones en la interface o subir la persiana o bajarla.

La salida correspondiente para cada caso , está conectada a su relé el cual la parte de fuerza trabaja a 230v, la peculiaridad en este circuito es que para evitar problemas si se accionan los don botones de subir y bajar a la vez y poder llegar a quemar el motor lo que se hizo fue lo siguiente:

Representación gráfica del bloqueo entre relés



De esta forma se evita tener cualquier accidente en nuestra instalación.

2.1.6 Automatización de componentes del hogar

En este apartado se engloban cualquier tipo de electrodomésticos eléctricos que tengamos en el hogar, nosotros hemos utilizado el aire acondicionado y una cámara ip a la cual le daremos tensión o se la quitaremos dependiendo de nuestra conveniencia, normalmente están siempre conectadas pero en nuestro caso y por seguridad se ha decidido hacer de esta forma.

Por otra parte el aire acondicionado es muy práctico con este sistema ya que si estamos fuera del hogar lo podemos conectar o apagar ,cabe resaltar que solo actuamos sobre él , es decir solo se le da o se le quita tensión, no podemos cambiar de modo, subir velocidad etc.. Ya que se hace a través de un relé como en los demás circuitos al igual que con la cámara ip, por el cual pasa la fase que proviene de su protección y la cual le alimenta.

Estos tipos de aparatos de la marca Mundo clima tienen la ventaja de la cual nos hemos aprovechado de que si el aparato está conectado ,por ejemplo en modo frio y se le quita

tensión, cuando se le vuelva a dar se conectara en el estado que se encontraba, también cabe decir que para poder controlarlo , siempre abra que utilizar la interface web , ya que si se apaga con su mando luego no se podrá conectar.

2.1.6 Control de acceso por RFID y alarma con voz

La idea de este apartado parte de cómo hacer una alarma de una forma un poco diferente a las que se encuentran en el mercado, un poco más divertida y por supuesto de bajo coste.

Para ello lo que utilizamos fue 5 elementos principales que fueron

-Sensor pir de movimiento

-Detector RFID

-Un módulo de voz (somo 14 d)

-Un altavoz

Y como extra un nano amplificador estéreo del cual se dispone pero no se ha utilizado en el proyecto.

La primera parte del control de acceso se hizo con ultrasonidos y la información de la alarma en lugar de hablar ,nos la mostraba en una pantalla lcd, pero al final nos decantamos por la voz ya que se creyó que era más profesional y más atractivo cara al futuro consumidor.

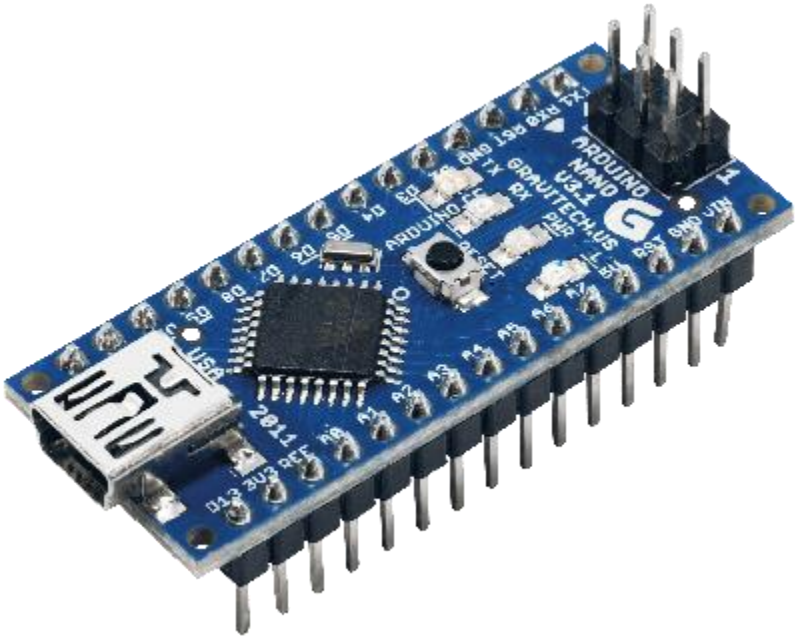
Las funciones de la alarma son las siguientes :

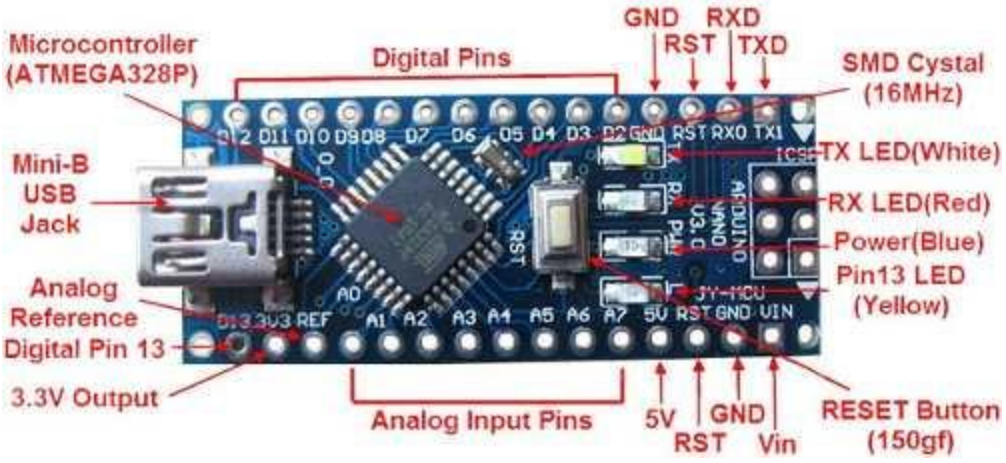
Partimos de la base de que la alarma está conectada y en su estado de reposo o sea desarmada, cuando nos dirigimos a salir de nuestra vivienda , tenemos el lector RFID en el cual nos identificamos mediante un tag, este tag esta memorizado en el programa de nuestro arduino por lo que si se usa otro que no está dado de alta no hará ninguna función.

Una vez nos identificamos, el estado de la alarma pasa a conectado y cuando se regresa a casa , al abrir la puerta el sensor de movimiento lo detecta y la alarma nos pide que nos identifiquemos , lo cual hacemos y esta se desactiva, si no lo hiciésemos saltaría la escena programada como intruso .

2.2 Modelos Arduino y sus características

2.2.1 Arduino Nano





La placa Arduino Nano es una placa de prueba pequeña y completa basada en ATmega328. Tiene funcionalidad similar al modelo Arduino Duemilanove, pero en un módulo DIP. Solo carece de jack de alimentación DC y funciona con un cable Mini-B USB en lugar de uno estándar.

Las características de entrada salida son que cada uno de los 14 pines digitales del Nano pueden ser usados como entrada o salida, usando las funciones pinMode(), digitalWrite(), y digitalRead(). Operan a 5 voltios. Cada pin puede proveer o recibir un máximo de 40mA y

poseen una resistencia de pull-up (desconectada por defecto) de 20 a 50 kOhms. Además algunos pines poseen funciones especializadas:

Serial: 0 (RX) y 1 (TX). (RX) usado para recibir y (TX)usado para transmitir datos TTL vía serie. Estos pines están conectados a los pines correspondientes del chip USB-a-TTL de FTDI.

Interrupciones Externas: pines 2 y 3. Estos pines pueden ser configurados para activar una interrupción por paso a nivel bajo, por flanco de bajada o flanco de subida, o por un cambio de valor. Mira la función attachInterrupt() para más detalles.

PWM: pines 3, 5, 6, 9, 10, y 11. Proveen de una salida PWM de 8-bits cuando se usa la función analogWrite().
SPI: pines 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Estos pines soportan la comunicación SPI, la cual, a pesar de poseer el hardware, no está actualmente soportada en el lenguaje Arduino.
LED: Pin 13. Existe un LED conectado al pin digital 13. Cuando el pin se encuentra en nivel alto, el LED está encendido, cuando el pin está a nivel bajo, el LED estará apagado.

El Nano posee 8 entradas analógicas, cada una de ellas provee de 10 bits de resolución (1024 valores diferentes). Por defecto miden entre 5 voltios y masa, sin embargo es posible cambiar el rango superior usando la función analogReference(). También, algunos de estos pines poseen funciones especiales:

I2C: Pines 4 (SDA) y 5 (SCL). Soporta comunicación I2C (TWI) usando la librería Wire .

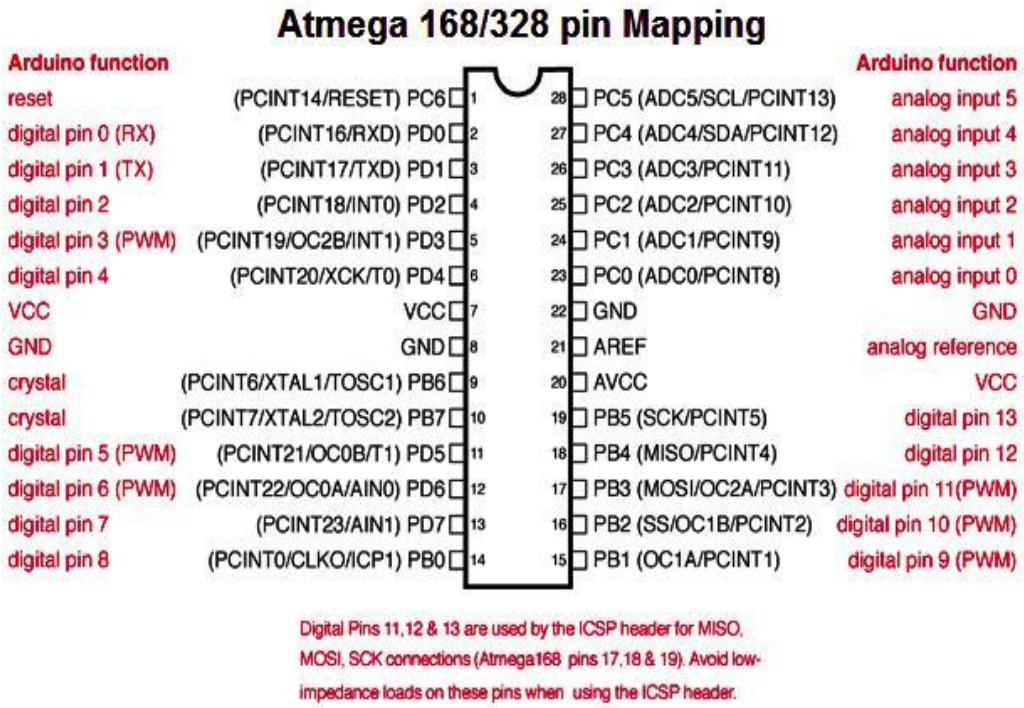
Hay algunos otros pines en la placa:

AREF. Tensión de referencia por las entradas analógicas. Se configura con la función analogReference().
Reset. Pon esta línea a nivel bajo para resetear el microcontrolador. Normalmente se usa para añadir un botón de reset que mantiene a nivel alto el pin reset mientras no es pulsado.

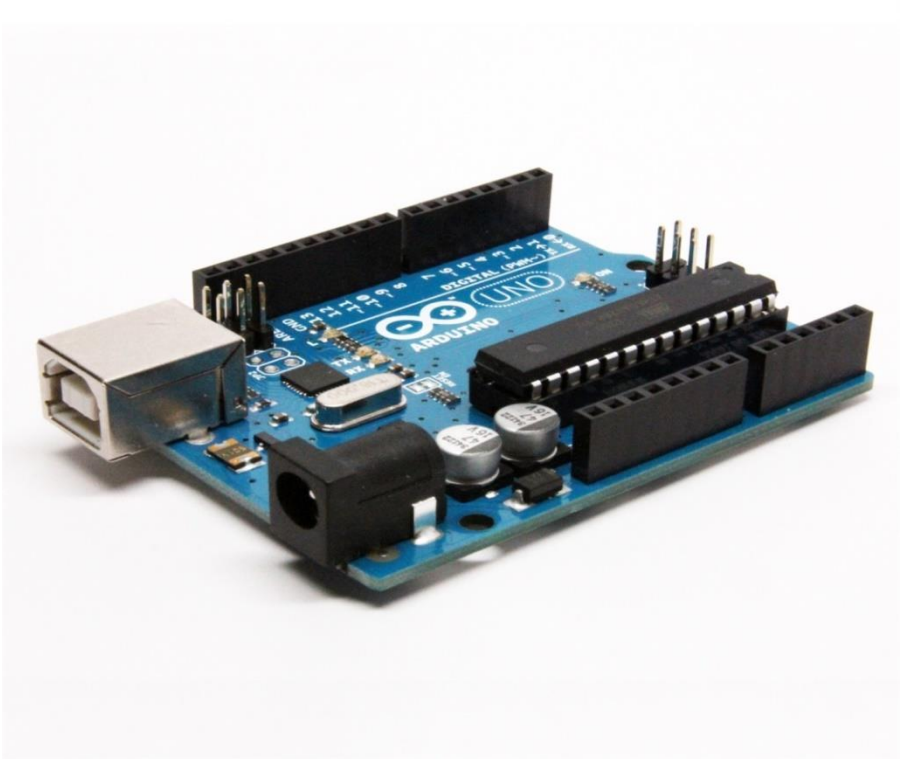
Las características más destacadas son:

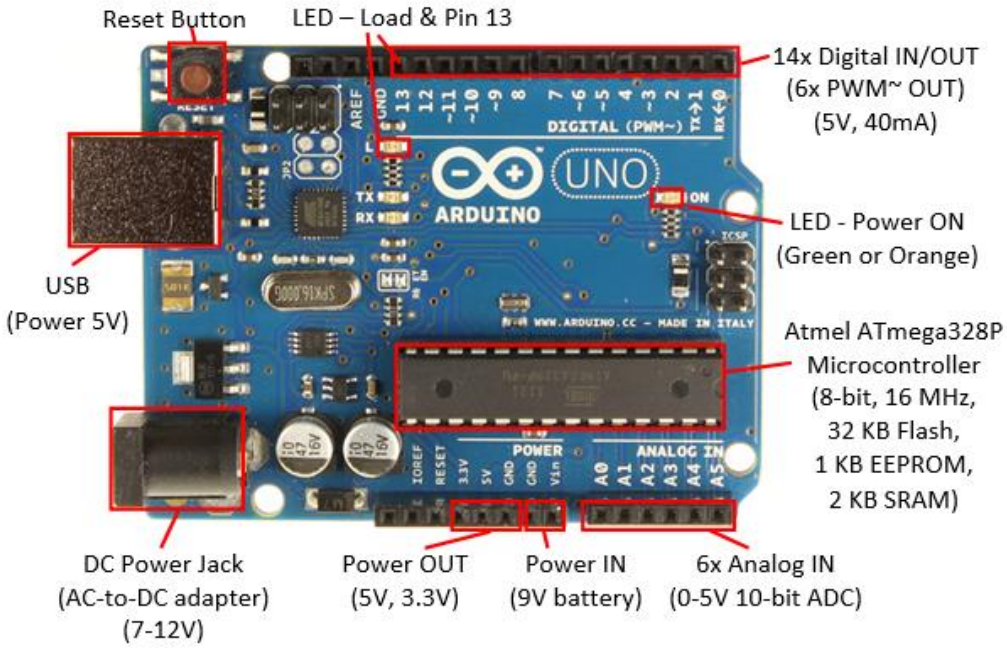
- Microcontrolador ATmega328 con cargador de inicio pre programado.
- Tensión de entrada (recomendada): +7 a + 12 V.
- Tensión de entrada (límites): +6 a + 20 V.
- 14 pines GPIO (de los que 6 ofrecen salida PWM).
- 6 pines de entrada analógica.

- 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”.



2.2.2 Arduino Uno





El Arduino Uno R3 utiliza el microcontrolador ATmega328. En adición a todas las características de las tarjetas anteriores, el Arduino Uno utiliza el ATmega16U2 para el manejo de USB en lugar del 8U2 (o del FTDI encontrado en generaciones previas). Esto permite ratios de transferencia más rápidos y más memoria. No se necesitan drivers para Linux o Mac (el archivo inf para Windows es necesario y está incluido en el IDE de Arduino).

La tarjeta Arduino Uno R3 incluso añade pins SDA y SCL cercanos al AREF. Es más, hay dos nuevos pines cerca del pin RESET. Uno es el IOREF, que permite a los shields adaptarse al voltaje brindado por la tarjeta. El otro pin no se encuentra conectado y está reservado para propósitos futuros. La tarjeta trabaja con todos los shields existentes y podrá adaptarse con los nuevos shields utilizando esos pines adicionales.

El Arduino es una plataforma computacional física open-source basada en una simple tarjeta de I/O y un entorno de desarrollo que implementa el lenguaje Processing/Wiring. El Arduino Uno R3 puede ser utilizado para desarrollar objetos interactivos o puede ser conectado a software de tu computadora (por ejemplo, Flash, Processing, MaxMSP). El IDE open-source puede ser descargado gratuitamente (actualmente para Mac OS X, Windows y Linux).

Nota: Esta plataforma requiere la carpeta de drivers Arduino 1.0 para poder instalarlo de forma apropiada en algunos computadores. Hemos testeado y confirmado que el Arduino Uno R3 puede ser programado en versiones anteriores del IDE. Sin embargo, la primera vez que uses el Arduino en una nueva computadora deberás tener el Arduino 1.0 instalado en la máquina. Si estás interesado en leer más acerca de los cambios en el IDE, revisa las notas oficiales de Arduino 1.0.

Características principales:

Microcontrolador ATmega328.

Voltaje de entrada 7-12V.

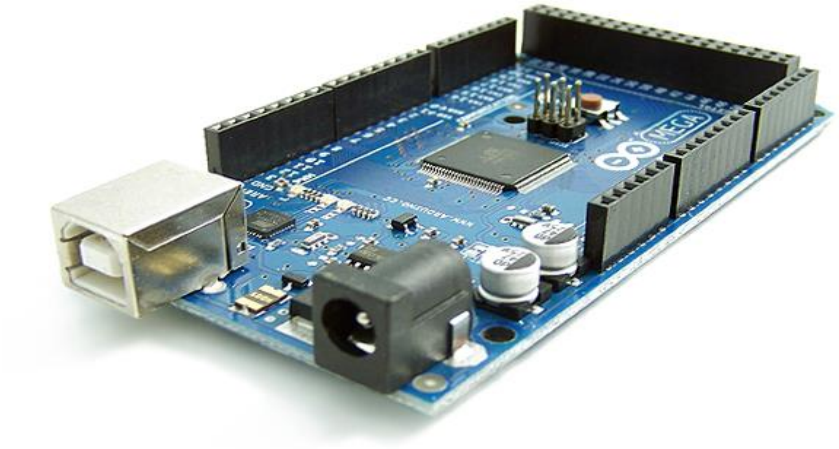
14 pines digitales de I/O (6 salidas PWM).

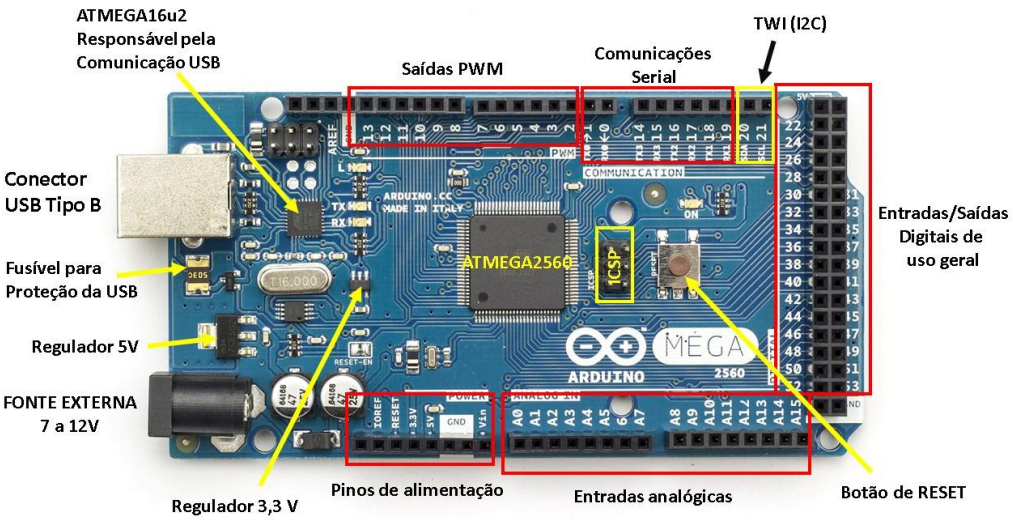
6 entradas análogas.

32k de memoria Flash.

Reloj de 16MHz de velocidad.

2.2.3 Arduino Mega 2560 y Shield Arduino Ethernet





El Arduino Mega está basado en el microcontrolador ATMega2560. Tiene 54 pines de entradas/salidas digitales (14 de las cuales pueden ser utilizadas como salidas PWM), 16 entradas analógicas, 4 UARTs (puertos serial por hardware), cristal oscilador de 16 Mhz, conexión USB, jack de alimentación, conector ICSP y botón de reset. Incorpora todo lo necesario para que el microcontrolador trabaje; simplemente conéctalo a tu PC por medio de un cable USB o con una fuente de alimentación externa. El Arduino Mega es compatible con la mayoría de los shields diseñados para Arduino Duemilanove, diecimila o UNO.

Esta nueva versión de Arduino Mega 2560 adicionalmente a todas las características de su sucesor, el Arduino Mega ahora utiliza un microcontrolador ATMega8U2 en vez del chip FTDI. Esto permite mayores velocidades de transmisión por su puerto USB y no requiere drivers para Linux o MAC (archivo inf es necesario para Windows) además ahora cuenta con la capacidad de ser reconocido por el PC como un teclado, mouse, joystick, etc.

Características principales:

Microcontrolador ATmega2560.

Voltaje de entrada de – 7-12V.

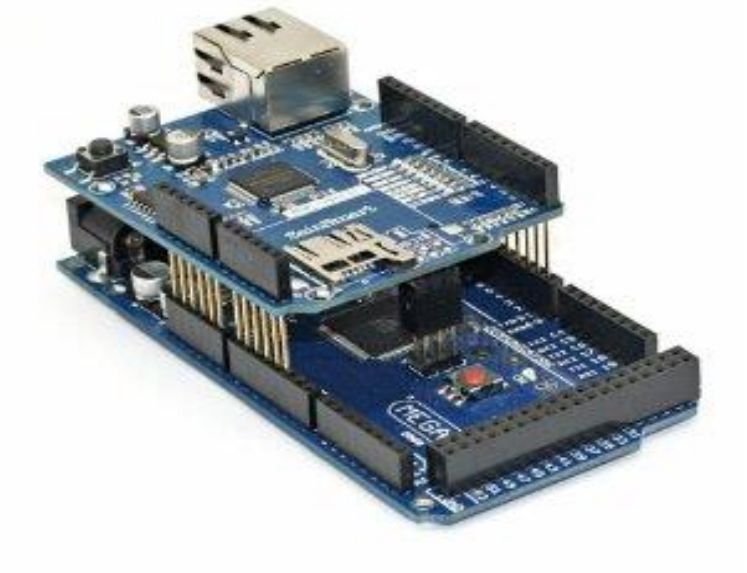
54 pines digitales de Entrada/Salida (14 de ellos son salidas PWM).

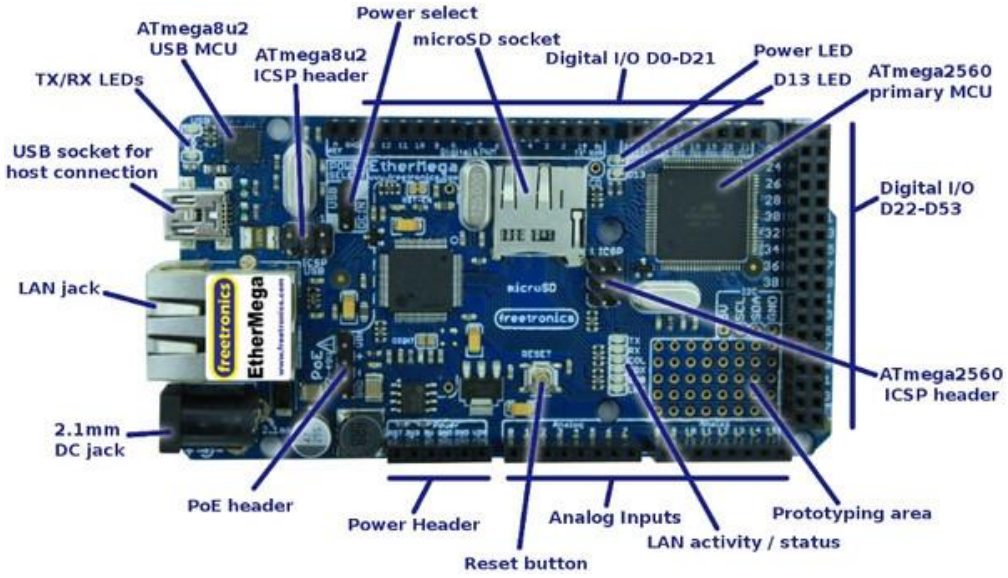
16 entradas análogas.

256k de memoria flash.

Velocidad del reloj de 16Mhz

Shield Ethernet para Arduino mega





La Arduino Ethernet Shield permite a una placa Arduino conectarse a internet. Se basa en la whiznet chip de ethernet . El Wiznet W5100 ofrece una red (IP) capaz de TCP y UDP. Es compatible con hasta cuatro conexiones de socket simultáneas. El escudo de Ethernet se conecta a una placa Arduino usando encabezados por arrollamiento de hilo largos que se extienden a través del escudo. Esto mantiene la disposición de las clavijas intacta y permite que otro escudo pueda ser apilado en la parte superior.

El escudo de Ethernet tiene un estándar de conexión RJ-45, con un transformador de línea integrada y alimentación a través de Ethernet activado.

Dispone de una ranura para micro-SD a bordo, que se puede utilizar para almacenar archivos para servir través de la red. Es compatible con todas las placas Arduino / Genuino. El lector de tarjetas micro SD de a bordo es accesible a través de la Biblioteca SD. Cuando se trabaja con esta biblioteca, SS es por el pin 4. La revisión original del escudo contenía una de tamaño completo ranura para tarjetas SD; esto no es compatible.

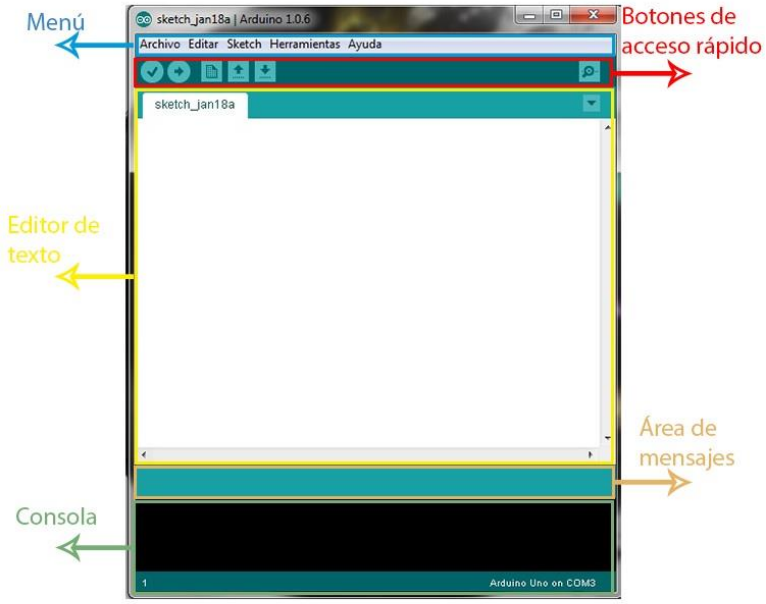
El escudo también incluye un controlador de reajuste, para asegurar que el módulo Ethernet W5100 se restablece correctamente en el encendido. Las revisiones anteriores del escudo no eran compatibles con la Mega y tenía la necesidad de restablecer manualmente después del encendido.

El escudo actual tiene un módulo de alimentación a través de Ethernet (PoE), diseñado para extraer energía de un cable Ethernet convencional de par trenzado Categoría 5

2.3 Entorno de programación

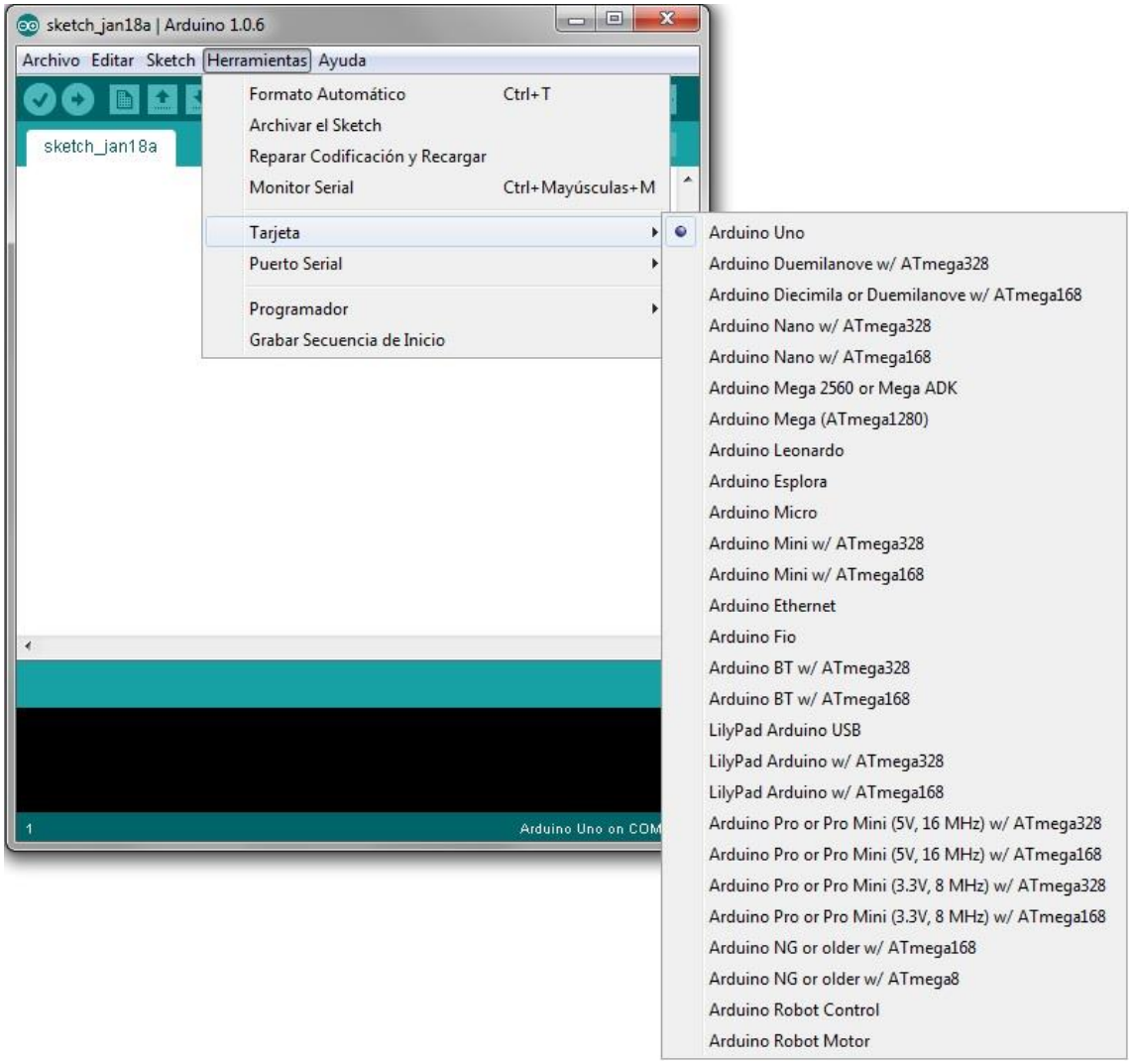
2.3.1 IDE Arduino

Dado que el **Arduino** es como un pequeño ordenador que ejecuta una serie de códigos que previamente le hemos introducido, necesitaremos un programa para poder meter estos códigos a la propia placa. Este programa se llama **IDE**, que significa "Integrated Development Environment" ("Entorno de Desarrollo Integrado")

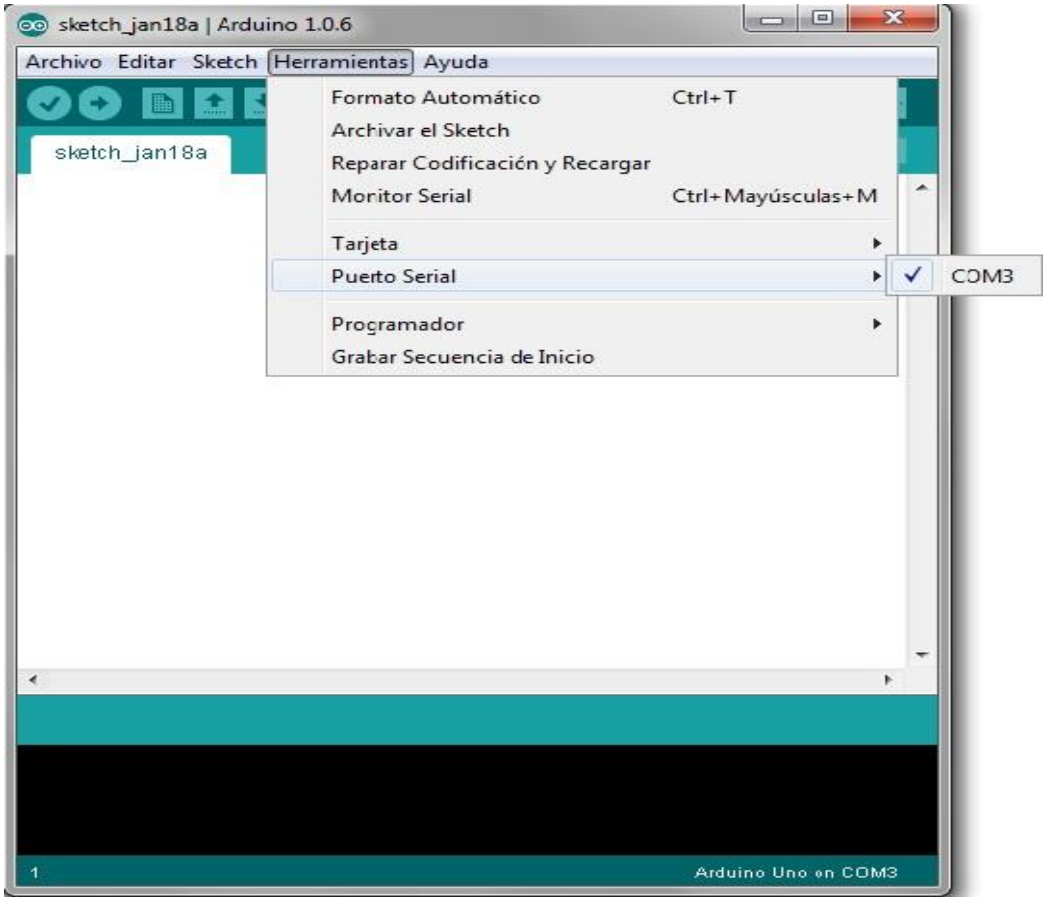


Este IDE estará instalado en nuestro PC, es un entorno muy sencillo de usar y en él escribiremos el programa que queramos que el Arduino ejecute. Una vez escrito, lo cargaremos a través del USB y Arduino comenzará a trabajar de forma autónoma.

El siguiente paso que realizaremos será configurar nuestro IDE para que se comunique con nuestra placa Arduino. Para ello conectaremos nuestro Arduino mediante el cable USB al PC y después de que el sistema operativo haya reconocido e instalado la tarjeta automáticamente, nos dirigimos a la zona de *menú*, pulsamos en *Herramientas* y después en *Tarjeta*. Ahí seleccionamos el modelo de tarjeta Arduino que tengamos, en nuestro caso "Arduino Uno".


sketch_jan18a | Arduino 1.0.6
Archivo Editar Sketch Herramientas Ayuda
sketch_jan18a
Formato Automático Ctrl+T
Archivar el Sketch
Reparar Codificación y Recargar
Monitor Serial Ctrl+Mayúsculas+M
Tarjeta
Puerto Serial
Programador
Grabar Secuencia de Inicio
1
Arduino Uno on COM
Arduino Uno
Arduino Duemilanove w/ ATmega328
Arduino Diecimila or Duemilanove w/ ATmega168
Arduino Nano w/ ATmega328
Arduino Nano w/ ATmega168
Arduino Mega 2560 or Mega ADK
Arduino Mega (ATmega1280)
Arduino Leonardo
Arduino Esplora
Arduino Micro
Arduino Mini w/ ATmega328
Arduino Mini w/ ATmega168
Arduino Ethernet
Arduino Fio
Arduino BT w/ ATmega328
Arduino BT w/ ATmega168
LilyPad Arduino USB
LilyPad Arduino w/ ATmega328
LilyPad Arduino w/ ATmega168
Arduino Pro or Pro Mini (5V, 16 MHz) w/ ATmega328
Arduino Pro or Pro Mini (5V, 16 MHz) w/ ATmega168
Arduino Pro or Pro Mini (3.3V, 8 MHz) w/ ATmega328
Arduino Pro or Pro Mini (3.3V, 8 MHz) w/ ATmega168
Arduino NG or older w/ ATmega168
Arduino NG or older w/ ATmega8
Arduino Robot Control
Arduino Robot Motor

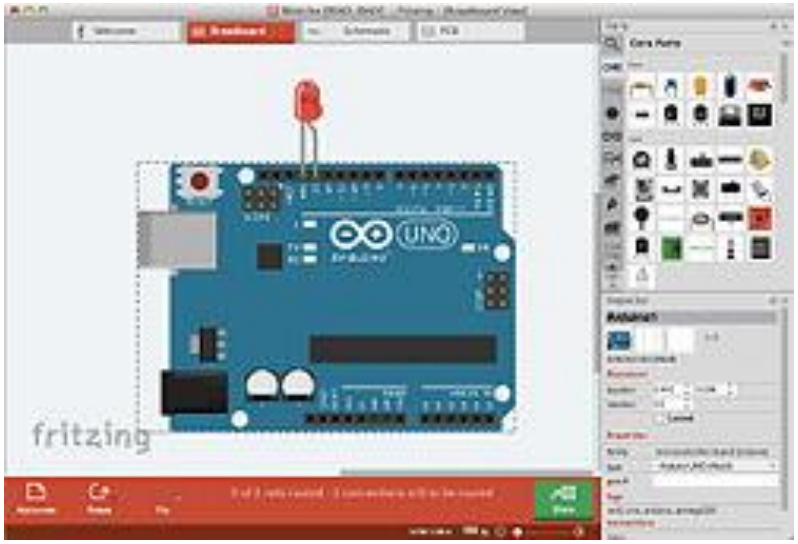
Después vamos a la opción Puerto Serial y elegimos el COM en el que tenemos conectado nuestro Arduino.



Si nos aparecieran varios COM activos, porque estemos usando otros dispositivos serial o por otro motivo, el cual suele pasar muy a menudo para saber cuál de ellos es el que se comunica con nuestra placa, solo tenemos que irnos al Panel de control/Hardware/Administrador de dispositivos. Miramos la pestaña (Puertos COM y LPT)y ahí nos aparecerá nuestro Arduino y el COM en el que está conectado. Con esto, ya podemos empezar a programar nuestro Arduino.

2.3.2 Fritzing

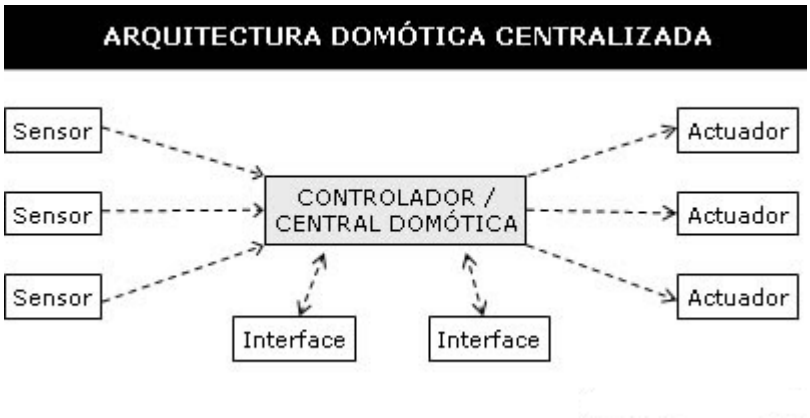
Fritzing fue creado bajo los principios de Processing y Arduino, y permite a los diseñadores, artistas, investigadores y aficionados documentar sus prototipos basados en Arduino y crear esquemas de circuitos impresos para su posterior fabricación, aparte de generar sus esquemas.



Además, cuenta con un sitio web complementario que ayuda a compartir y discutir bosquejos y experiencias y a reducir los costos de fabricación.

2.4 Arquitecturas de control domótico

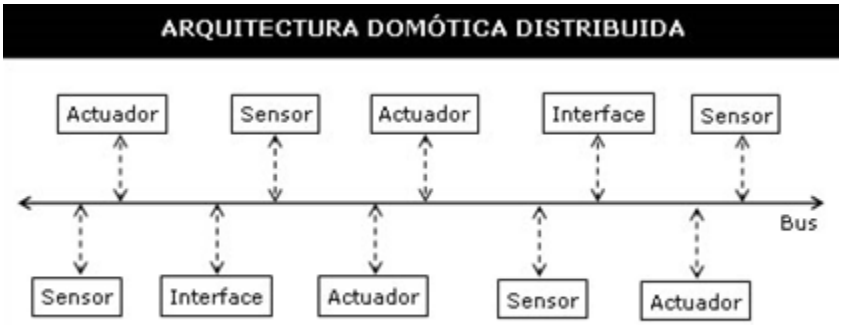
2.4.1 Centralizada



Es la instalación, en la que los elementos que vamos a controlar y supervisar como pueden ser , sensores ,luces ,válvula , etc. se tienen que cablear hasta donde se encuentra el sistema de control, que puede ser un pc , una centralita ,un microcontrolador embebido el cual es nuestro caso.

Si falla nuestro sistema de control, se cae todo ya que todo depende de él, y la instalación se tiene que hacer independiente a la instalación eléctrica y prever eso a la hora de hacer la preinstalación.

2.4.2 Distribuida



Los sistemas de arquitectura distribuida se caracterizan por que cada dispositivo tiene un pequeño procesador propio que gestiona la información que se le ha sido preprogramada por el fabricante en forma de programa de aplicación para ciertas funciones específicas, y actúa según analice la información que le entra por el bus de datos, donde se interconecta con los demás dispositivos, donde se envían información entre todos ellos, tanto las entradas (sensores, pulsadores, interfaces, etc.) como las salidas (actuadores dimmers, relés, persianas, etc.). Este tipo de arquitectura es muy utilizada también en sistemas inalámbricos. **Como knx etc..**

2.4.3 Mixta

Este tipo de arquitectura se da cuando fusionamos los dos tipos anteriores, bien porque el instalador se ve más como trabajando con este tipo o bien por la instalación lo requiere ya pueda ser por ejemplo , si una parte de la instalación está centralizada y se quiere ampliar

2.4.4 Nivel de domotización

Es un nivel de evaluación de una vivienda o instalación domótica, consiste en rellenar un formulario web donde depende de los dispositivos a controlar se genera una puntuación, y se considera que una vivienda es domótica si se alcanza como mínimo el nivel 1 de los 3 que hay, para ello se tiene que obtener en la ponderación del test como mínimo 13 puntos, este

test ha sido realizado en la web www.cedom.es y cumple con la especificación técnica de AENOR EA0026.

En el test realizado se ha sacado una puntuación de 16, por lo que nuestra instalación pertenece al nivel tipo 1, que es el nivel mínimo para que una instalación se considere como domótica.

Tabla de Niveles de Domotización	
Dispositivos	Nº de dispositivos o condición
Detectores de presencia	○ Ninguno ○ 2 ◉ 1 cada 20 m2 ○ 1 por estancia
Teclado codificado, llave electrónica, o equivalente.	○ Ninguno ◉ 1
Sirena interior	○ No ◉ Si
Contactos de ventana y/o impactos	◉ No ○ En puntos de fácil acceso ○ En todas las ventanas
Sistema de mantenimiento de alimentación en caso de fallo de suministro eléctrico	◉ No ○ Si
Módulo de habla/escucha, destinado a la escucha en caso de alarma* También se admite cualquier tipo de control que permita conocer si realmente existe un intruso (cámaras web...)	◉ No ○ Si
Sistema conectable con central de alarmas	◉ No ○ Si
Suma Parcial Alarma de intrusión	5

Detectores de inundación necesarios en zonas húmedas (baños, cocina, lavadero, garaje)	◉ No ○ Los necesarios 1)
Electro válvula de corte agua con instalación para "bypass" manual.	◉ No ○ Las necesarias 1)
Detectores de concentraciones de gas butano y/o natural en zonas donde se prevea que habrá elementos que funcionen con gas	◉ No ○ Los necesarios 1)
Electro válvula de corte gas con instalación para "bypass" manual	◉ No ○ Las necesarias1)
Detector de incendios	◉ No

Suma Parcial Alarmas técnicas	0
Simulación de presencia	◉ No ○ Relacionada con las persianas motorizadas o con puntos de luz. ○ Relacionada con persianas motorizadas y con puntos de luz
Suma Parcial Simulación de presencia	0

Videoportero	◉ No ○ Si
Suma Parcial Videoportero	0
Control de persianas	○ No ◉ Todas las de superficie superior a 2m2 ○ Todas
Suma Parcial Control de persianas	1
Regulación lumínica con control de escenas	◉ No ○ en dependencias dedicadas al ocio ○ En salón y dormitorios
En jardín o grandes terrazas mediante interruptor crepuscular o interruptor horario astronómico	◉ No ○ Si
Conexión/desconexión general de iluminación	◉ No ○ Un acceso ○ Todos los accesos
Control de puntos de luz y tomas de corriente más significativas	○ No
Suma Parcial Control de iluminación	2
Cronotermostato	◉ No ○ 1 en salón ○ zonificando la vivienda en un mínimo de dos zonas ○ Varios cronotermostatos, zonificando la vivienda por estancias
Suma Parcial Control de clima	0
Posibilidad de realizar programaciones horarias sobre los equipos controlados	○ No ◉ Si
Gestor energético	◉ No ○ Si
Suma Parcial Programaciones	2
Consola o equivalente	○ No ◉ Si
Control telefónico bidireccional	◉ No ○ Si ○ Interacción mediante SMS
Equipo para control a través de internet, WAP o equivalente	○ No ◉ Si
Suma Parcial Interfaz usuario	5
Dispositivos conectables a empresas suministradoras a través de redes de comunicación	◉ 0 ○ 1 ○ 2 ○ 3 o más
Suma Parcial Dispositivos conectables a empresas suministradoras	0

Tomas SAT y Tomas Multimedia	◉ No ○ 3 tomas satélite + 3 tomas multimedia ○ 3 tomas satélite +1 toma multimedia en todas las estancias, incluido terraza
Punto de acceso inalámbrico	○ No

Suma Parcial Red Multimedia	1
SUMA TOTAL	16
Número de aplicaciones domóticas cubiertas[2)]	
	Calcular

2.4.5 Normativa

Dado que nuestra instalación es de tipo centralizado, nos centraremos solo en el tipo de normativa que rige este tipo de instalaciones

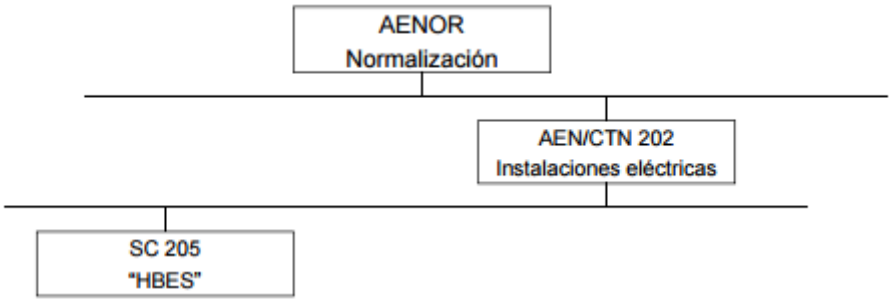
Organismos de normalización

	General	Eléctrico	Telecom.
Internacional	ISO	IEC	UIT
Europeo	CEN	CENELEC	
Nacional	AENOR		

Comités de normalización

Organismo de Normalización	IEC ISO	CENELEC cen	AENOR
Comité	JTC1/SC 25/WG 1	CEN/TC 247 CLC/TC 205 ----	AEN/CTN 100 AEN/CTN 202/SC 205 AEN/CTN 210/SC 215

Comités de Normalización en Domótica



Campo de actividad:

Preparar normas para todos los aspectos de sistemas electrónicos domésticos y en edificios en relación a la sociedad de la información. En más detalle, preparar normas para asegurar integración de un espectro amplio de aplicaciones y aspectos de control y gestión de otras aplicaciones en y entorno a viviendas y edificios, incluyendo las pasarelas residenciales a diferentes medios de transmisión y redes públicas, teniendo en cuenta todo lo relativo a EMC y seguridad eléctrica y funcional.

Visión General de la Normativa Domótica

La legislación actual y la normativa que se dedica a regular los sistemas domóticos no son muy concreta y no está bien definida.

Las normas y la reglamentación son las expuestas en el cuadro anterior.

Estas normas y reglamentaciones nos establecen los requisitos necesarios para una instalación domótica.

Quedan excluidas las instalaciones de telecomunicaciones , sistemas de seguridad y contraincendios que están reglamentados por el Ministerio de Interior y de Fomento.

2.5 Métodos de conexionado

2.5.1 Mediante cableado

Este método empleado en nuestra instalación entre emisores y receptores es mediante cable, por el cual transmiten estos una señal (voltaje) .

En nuestro caso cada señal emitida o recibida tiene una acción o reacción concreta .

Nuestro proyecto comienza a tomar forma principalmente en la placa de prototipo donde mediante los jumpers realizamos la conexión que luego trasladaremos a la instalación real.

En la instalación real se ha utilizado cable normalizado de 1.5 mm para alumbrado y de 2.5 mm para persianas y otros elementos.

Para la comunicación con el servidor y los relés y para la instalación de la parte del tanque y la descalcificadora , se han usado cables multifilares apantallados para evitar los corrientes parasitas generadas por los campos magnéticos en el caso d la descalcificadora , por el motor del grupo de presión que hay junto a ella y en el caso de la vivienda por la instalación de la vivienda ya que no estaba acondicionada desde obra para una ampliación de domótica.

2.5.2 Inalámbrico

El conexionado o comunicación inalámbrica , es la forma de comunicación mediante ondas electromagnéticas entre un objeto y otro.

wiffi

Esta nueva tecnología surgió por la necesidad de establecer un mecanismo de conexión inalámbrica que fuese compatible entre distintos dispositivos. Buscando esa compatibilidad, en 1999 las empresas 3 Com , Airones ,Intersil, Nokia se unieron para crear la Wireless Ethernet Compatibility Alliance, o Weca, actualmente llamada Wi-Fi Alliance. El objetivo de la misma fue designar una marca que permitiese fomentar más fácilmente la tecnología inalámbrica y asegurar la compatibilidad de equipos.

De esta forma, en abril de 2000 WECA certifica la interoperativilidad de equipos según la norma IEEE 802.11b, bajo la marca Wi-Fi. Esto quiere decir que el usuario tiene la garantía de que todos los equipos que tengan el sello Wi-Fi pueden trabajar juntos sin problemas, independientemente del fabricante de cada uno de ellos.

En el año 2002 la asociación WECA estaba formada ya por casi 150 miembros en su totalidad.La familia de estándares 802.11 ha ido naturalmente evolucionando desde su creación, mejorando el rango y velocidad de la transferencia de información, su seguridad, entre otras cosas.

La norma IEEE 802.11 fue diseñada para sustituir el equivalente a las capas físicas y MAC de la norma 802.3 Ethernet. Esto quiere decir que en lo único que se diferencia una red wifi de

una red Ethernet en cómo se transmiten las tramas o paquetes de datos; el resto es idéntico. Por tanto, una red local inalámbrica 802.11 es completamente compatible con todos los servicios de las redes locales LAN de cable 802.3 Ethernet.

Bluetooth

Bluetooth funciona en las frecuencias entre 2.402 y 2.480 MHz o 2400 y 2483,5 MHz incluyendo las bandas de guarda de 2 MHz de ancho en el extremo inferior y 3,5 MHz de ancho en la parte superior. Esto es en el nivel global sin licencia (pero no reglamentada) Industrial, Científica y médica (**ISM**) banda de frecuencia de radio de corto alcance a 2,4 GHz. Bluetooth utiliza una tecnología de radio llamada espectro ensanchado por salto de frecuencia . Bluetooth divide en paquetes de datos transmitidos, y transmite cada paquete en uno de los 79 canales designados Bluetooth. Cada canal tiene un ancho de banda de 1 MHz. Por lo general lleva a cabo a 800 saltos por segundo, con adaptativa salto de frecuencia (AFH) habilitado. baja energía Bluetooth utiliza espaciado 2 MHz, que tiene capacidad para 40 canales.

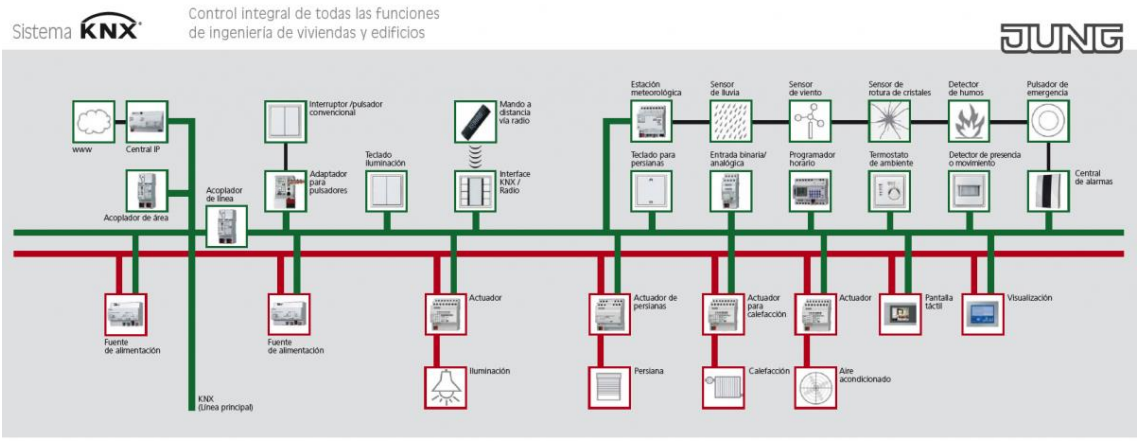
Originalmente, Gaussian modulación por desplazamiento de frecuencia de modulación (GFSK) fue el único esquema de modulación disponibles. Desde la introducción de Bluetooth 2.0 + EDR, (Differential cuadratura Phase Shift Keying) y la modulación 8DPSK también puede ser utilizado entre dispositivos compatibles. Los dispositivos que funcionan con GFSK se dice que están operando en el modo de velocidad básica (BR) cuando una instantánea velocidad de datos de 1 Mbit / s es posible. El término Enhanced Data Rate (EDR) se utiliza para describir π / 4-DPSK esquemas y 8DPSK, cada uno dando 2 y 3 Mbit / s, respectivamente. La combinación de estos modos (EDR) BR y en la tecnología de radio Bluetooth está clasificado como "de radio BR / EDR".

Bluetooth es un protocolo basado en paquetes con una estructura maestro-esclavo . Un maestro puede comunicarse con hasta siete esclavos en una piconet . Todos los dispositivos comparten el reloj del maestro. Intercambio de paquetes se basa en el reloj básico, definido por el maestro, que cumple a intervalos de 312,5 mu s. Dos ciclos de reloj conforman una ranura de 625 mu s , y dos ranuras forman un par ranura de 1250 mu s. En el caso simple de una sola ranura paquetes que el maestro transmite en ranuras y recibe incluso en las ranuras impares. El esclavo, por el contrario, recibe en las ranuras pares e impares transmite en ranuras. Los paquetes pueden ser de 1, 3 o 5 ranuras de tiempo, pero en todos los casos de transmisión del maestro comienza en las ranuras pares y el esclavo de las ranuras impares.

Lo anterior es válido para el "clásico" BT. Bluetooth Low Energy, introducido en la especificación 4.0, utiliza el mismo espectro, pero de manera algo diferente; ver núm.interfaz de radio de baja energía Bluetooth .

2.6 Otros métodos de control

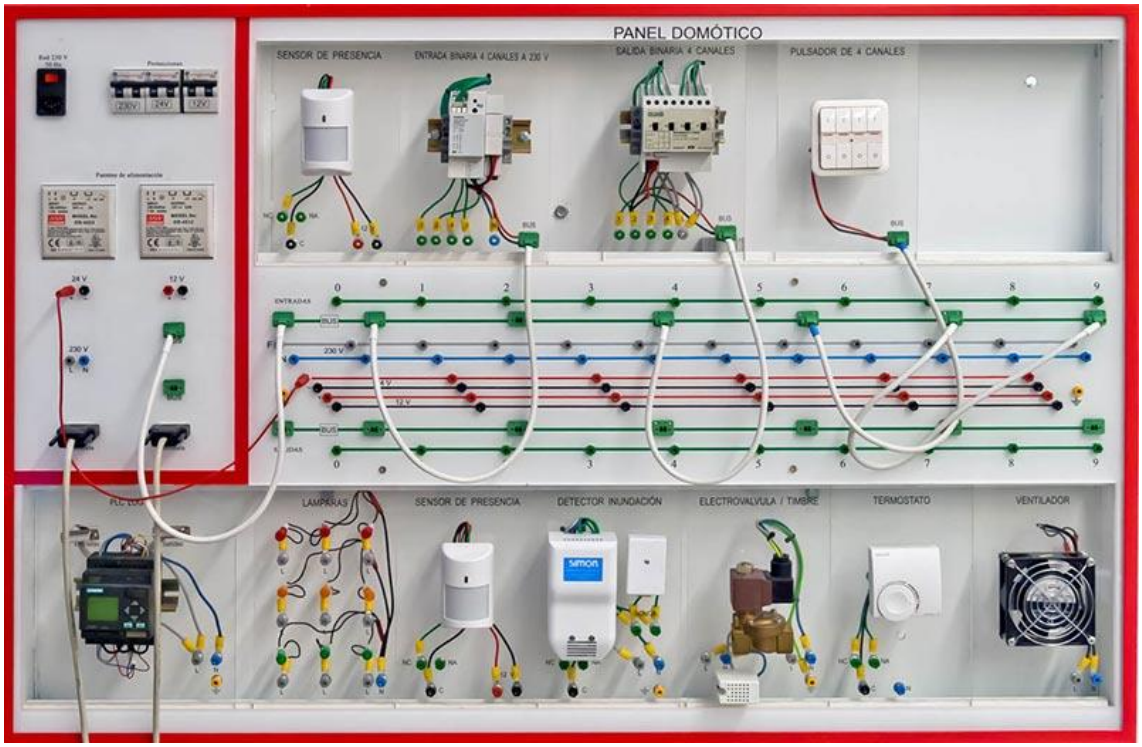
2.6.1 KNX



Básicamente el sistema knx es una instalación en la que unos emisores como pueden ser los pulsadores y unos receptores como puede ser un alumbrado están conectados a un bus y este a una centralita.

A la hora de la puesta en marcha del sistema , lo que se hace es identificar cada elemento en la centralita emitiendo cada uno una frecuencia o señal que se diferenciara del resto y con esta información ya se hace el programa para gestionar los receptores o actuadores de la instalación.

KNX está aprobado como e estándar internacional (ISO/IEC 14543-3), estándar europeo (CENELEC EN 50090 y CEN EN 13321-1) así como estándar nacional en países como China (GB/T 20965). Ello asegura la continuidad de KNX en el futuro. Dispositivos KNX de diferentes fabricantes pueden ser combinados - la marca registrada KNX garantiza la interoperabilidad y el «interworking». En resumen, KNX es un estándar abierto líder a nivel mundial para el control tanto de viviendas como de edificios.



2.6.2 X-10

X10 es un protocolo de comunicaciones para el control remoto de dispositivos eléctricos que utiliza la línea eléctrica (220V o 110V) preexistente para transmitir señales de control entre equipos de automatización del hogar (domótica) en formato digital. Los dispositivos X10 que se comercializan son solo para uso individual y en entornos domésticos de hasta 250 m2, dada su limitación en ancho de banda y en el número máximo de dispositivos a controlar (256). No obstante existen elementos de última generación que incorporan, entre otros, los protocolos X-10 extendidos, para dar funcionalidad a soluciones de comunicación como la bidireccionalidad, solicitud de estados y comprobación de la correcta transmisión de las tramas.

X10 fue desarrollada en 1978 por Pico Electronics of Glenrothes, Escocia, para permitir el control remoto de los dispositivos domésticos. Fue la primera tecnología domótica en aparecer y sigue

siendo la más ampliamente disponible, principalmente por su característica de autoinstalable, sin necesidad de cableado adicional.

Las señales de control de X10 se basan en la transmisión de ráfagas de pulsos de RF (120 kHz) que representan información digital. Estos pulsos se sincronizan en el cruce por cero de la señal de red (50 Hz o 60 Hz). Con la presencia de un pulso en un semiciclo y la ausencia del mismo en el semiciclo siguiente se representa un '1' lógico y a la inversa se representa un '0'. A su vez, cada orden se transmite 2 veces, con lo cual toda la información transmitida tiene cuádruple redundancia. Cada orden involucra 11 ciclos de red (220 ms para 50 Hz y 183,33, para 60Hz).

Primero se transmite una orden con el Código de Casa y el Número de Módulo que direccionan el módulo en cuestión. Luego se transmite otro orden con el código de función a realizar (Function Code). Hay 256 direcciones soportadas por el protocolo.

Se han propuesto distintas alternativas con más banda, incluyendo protocolos como European Home Systems, Lonworks, XD2, CEBus, aunque sigue siendo el más extendido.

Configuración del **sistema de domótica X10**

Para controlar las luces y equipos eléctricos, el sistema x10 utiliza los **comandos** x10 que se envían a través de la red eléctrica. Cada punto de conexión utiliza un receptor x10 para recibir y ejecutar el comando x10. Estos receptores se denominan "módulos x10". Los comandos x10 se envían a través de la red eléctrica de la vivienda por los controladores x10. Todos los módulos x10 deben tener una dirección x10 y podrán ser controlados de forma independiente o conjunta.

1. **Controladores X10**

Los controladores x10 envían una dirección x10 y un comando x10 a través de la red eléctrica ,para controlar los módulos x10. Hay controladores x10 que funcionan como temporizadores, vía línea telefónica, con conexión a internet, etc.

2. Controladores inalámbricos X10

Los controladores inalámbricos x10 transmiten una señal a un receptor x10 que transforma las señales de radiofrecuencia o de infrarrojos a señales X10 que inyecta en la red eléctrica de la vivienda. Hay controladores inalámbricos x10 que funcionan como mandos a distancia, o como interruptores inalámbricos, o como sensores, etc.

3. Módulos X10

Hay 4 tipos diferentes: módulos x10 enchufarles, interruptores x10 integrados, micro módulos x10y módulos x10 para montaje en carril DIN en cuadro eléctrico. Los módulos x10 reciben las órdenes x10 de los controladores x10 a través de la red eléctrica.

4, Direcciones X10

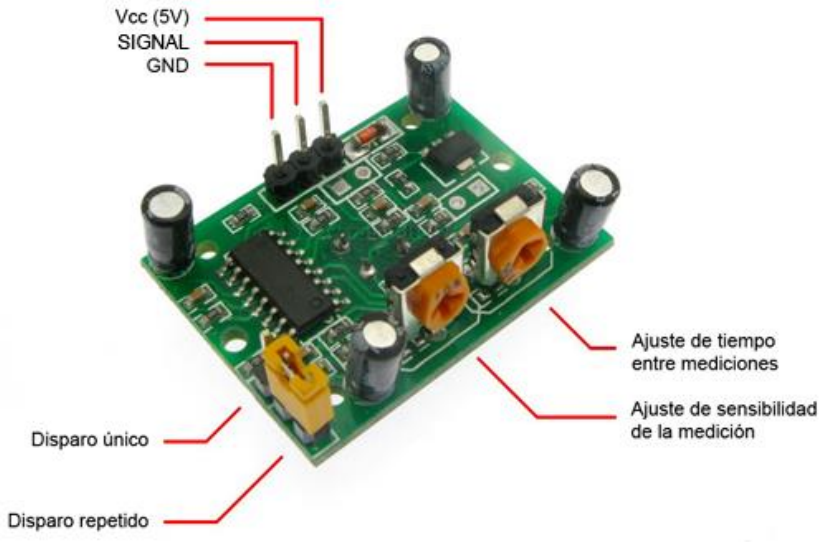
Mediante el uso de dos ruedas de código o por programación sencilla se pueden configurar hasta 256 direcciones x10 diferentes. Estas direcciones x10 se subdividen en un Código de Casa (A-P) y un Código de Unidad (1-16). El Código de Casa puede también ser ajustado en los controladores, lo que significa que los controladores y los módulos forman parte del mismo sistema domótico x10. El sistema X10 también utiliza unos comandos x10 estándar, que controlan todas las unidades con el mismo Código de Casa al mismo tiempo (por ejemplo todas las luces encendidas, apagadas, etc.).

2.7 Sensores

2.7.1 Sensor de movimiento y presencia

En el mercado hay mucha variedad de sensores de movimiento y de presencia, pero para nuestro montaje he instalación basándonos en arduino hay uno que es el más utilizado y es el PIR HC-SR501 .

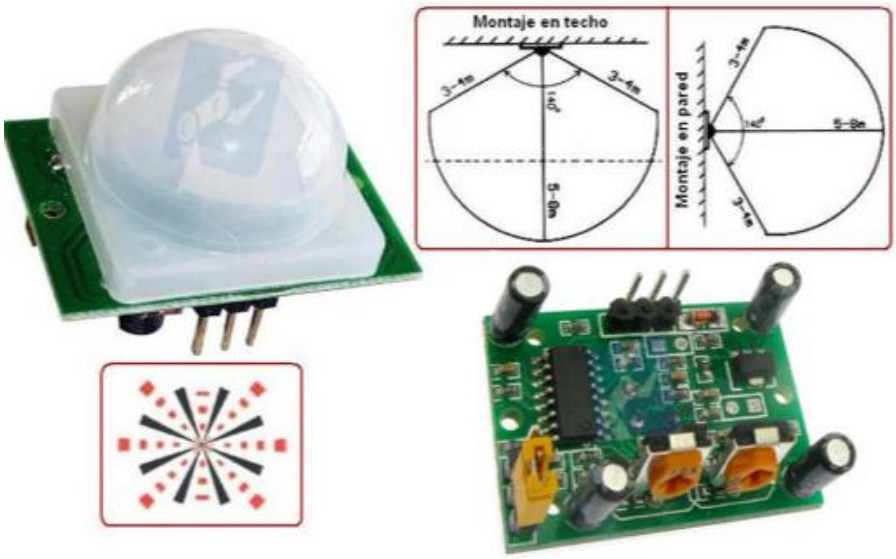
Es un sensor de muy bajo coste y bastante efectivo en su función,

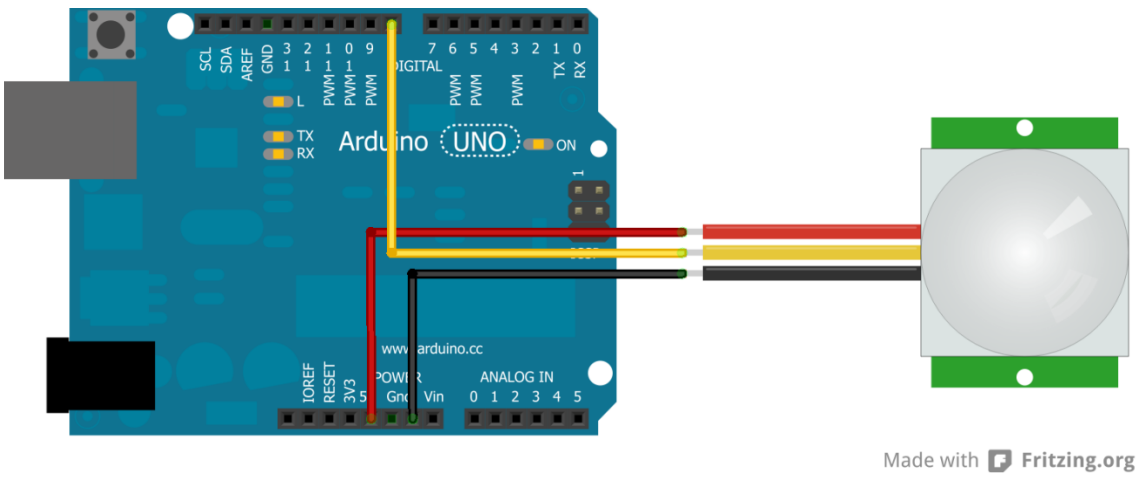


Este sensor detecta movimiento por cambios en el infrarrojo. Es el sensor habitual para detectar intrusiones en áreas cerradas o para encender la luz al pasar sin necesidad de interruptor.

Puede manejar directamente un pequeño relé sin necesidad de micro controlador o leerse desde un arduino o similar para formar una red de sensores múltiples.

Sencillo y resistente detecta el movimiento hasta unos 5 o 6 metros de distancia





2.7.2 Sensor de ultrasonidos

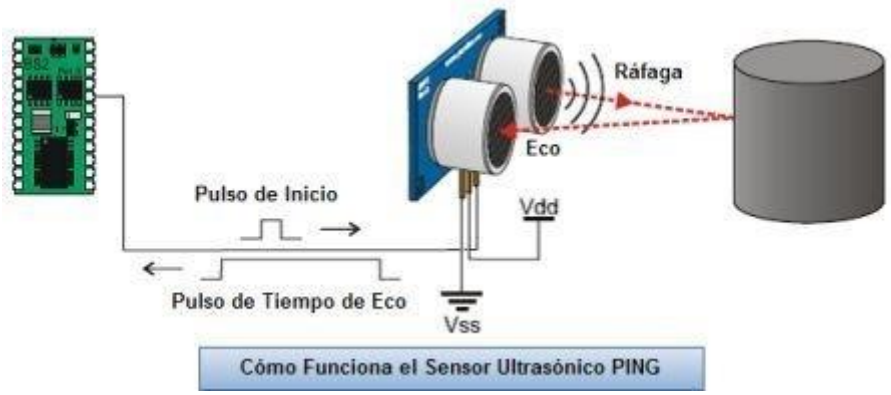
El módulo HC-SR04 o más bien conocido en el entorno de arduino como sensor de ultrasonidos , es el más extendido en este mundo, debido a la gran cantidad de códigos e información que circulan por la red. Este módulo tiene muy pocas dimensiones , lo que lo hace muy práctico y sobre todo un reducido coste.

Tiene un rendimiento estable y una alta precisión.

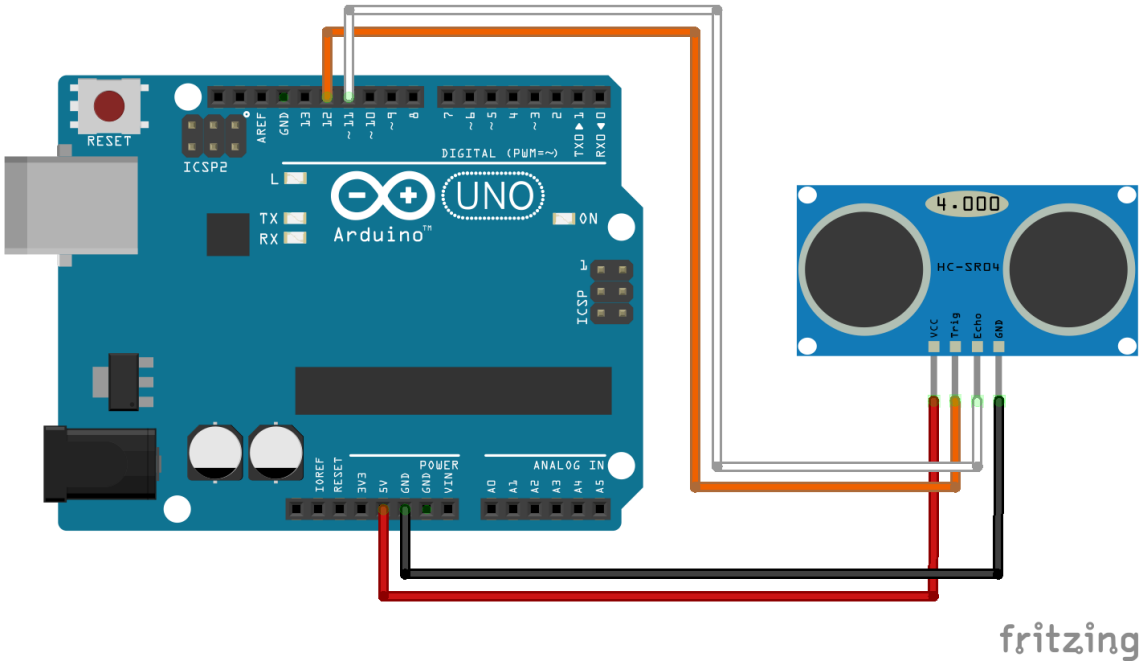
El sensor de ultrasonidos se enmarca dentro de los sensores para medir distancias o superar obstáculos, entre otras posibles funciones.

En este caso vamos a utilizarlo para la medición de distancias. Esto lo consigue enviando un ultrasonido (inaudible para el oído humano por su alta frecuencia) a través de uno de la pareja de cilindros que compone el sensor (un transductor) y espera a que dicho sonido rebote sobre un objeto y vuelva, retorno captado por el otro cilindro.

Este sensor en concreto tiene un rango de distancias sensible entre 3cm y 3m con una precisión de 3mm.



Posee 4 patilla, 2 de alimentación (5v y gnd) y luego ping y eco, por la de eco es la que emite el ultrasonido y por la de eco la recibe , entonces dependiendo del tiempo en que tarda en llegar la onda se calcula la distancia a medir.



2.7.3 Sensor de accionamiento mecánico

La única utilidad que tiene este sensor y de hecho esta creado específicamente para ese fin es controlar el nivel de un fluido como puede ser de un final de carrera la posición de algo que queremos que detenga su movimiento.

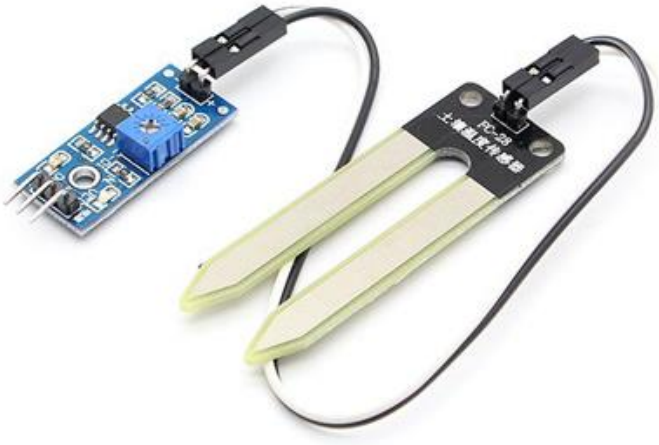
No esta creado exclusivamente para trabajar con un microcontrolador ya que soporta hasta 230v.

Este sensor es un sensor de tipo pasivo ya que estando en su posición de reposo no acciona nada ya que su contacto interno está abierto, dispone de 2 partes, la parte que se instala mediante un racor a la superficie del tanque y la parte móvil que es la flotante, cuando esta se pone en línea con la fija se cierra el contacto.

2.7.4 Sensor de humedad

Es un sensor que utiliza la conductividad entre sus 2 terminales o electrodos para medir parámetros relacionados con el agua y así con su porcentaje de humedad dependiendo de su instalación.

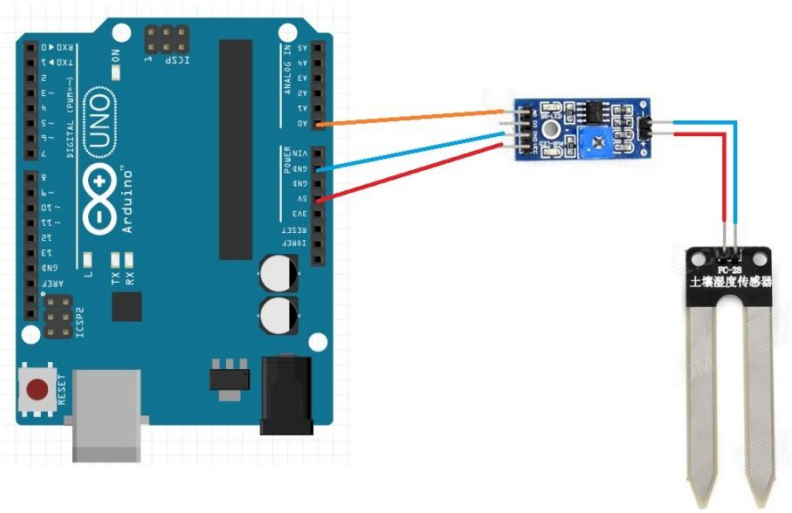
Su funcionamiento se basa en que emite una pequeña corriente entre sus terminales y esta depende de un extremo al otro de la resistencia de la tierra y esta a su vez de la humedad , por lo que si la humedad aumenta la corriente también ya que los electrones lo tienen más fácil para pasar de un extremo al otro.



Tiene 2 tipos de funcionamiento, analógico y digital.

En el analógico tendremos una resolución de entre 0 y 1023 que haciéndole un escalado ya previamente creado en su librería nos lo transformara en el porcentaje de humedad que poseamos en la instalación y luego el digital que básicamente si detecta agua nos dará un 1 o 5v , el cual es el que se usó en el proyecto.

Funciona con 5v.



2.7.5 Otros sensores útiles en domótica

- Foto resistencia ldr

Se puede utilizar para cuando tengamos poca luz en el balcón, que se recoja el toldo, o en el interior de la vivienda para cuando se haga de noche se conecte una escena con las luces que deseemos.

Potencia Max: 90mW
Tensión Max: 150 Vdc
Temperatura: -25 a 75ºC
Resistencia 10 lux: >50K ohms
Resistencia 0 lux: >0.9M ohms

El C-2795, fotorresistencia LDR, disminuye su resistencia en proporción al aumento de la luz ambiente.
Permite el control directo de relés y transistores.
Indicada especialmente para su aplicación como interruptor crepuscular, baliza de encendido automático, dimmers, barreras fotoeléctricas, etc.

-**Sensor de humedad y temperatura dht11**

Este sensor , se puede utilizar para mandar la información a una pantalla lcd o tft y hacerla funcionar como una estación meteorológica para el exterior de la vivienda e incluso para el interior también.

Alimentación: 3-5V
Consumo máximo de corriente 2.5mA
Humedad: 20-80% con precisión del 5%
Temperatura: 0-50ºC precisión +/-2ºC
Tamaño: 15.5mm x 12mm x 5.5mm
4 pines con separación 2.54mm.

-**Modulo sensor de agua**

Su utilidad es muy diversa, colocándolo en el exterior de la vivienda podemos hacer que si llueve se recojan los toldos y además si tenemos un riego por goteo que no se conecte , pero en el interior de la vivienda , se puede colocar debajo de cada toma de agua y se tuviésemos una fuga , el sensor actuaria y a través de una electroválvula y un relé podríamos cortar el suministro de agua de la vivienda hasta que se subsanase la avería.

-**Sensor de consumo eléctrico**

Puede ser útil para saber el consumo real de nuestro hogar almacenando los datos en un sd cada cierto tiempo y hacer nosotros un algoritmo con el código para que nos muestre el consumo real que tenemos en nuestra vivienda en ese mes .

Sensor de consumo de núcleo abierto.
Para corriente alterna.
Máximo 100A.

-Sensor de gas MQ9

Para evitar fugas de gas y posibles incendios conectándolo a una electroválvula a través de un relé para que si el sensor se acciona nos corte el suministro de gas.

Debido a que arduino está muy extendido en todo el planeta la lista de sensores sería prácticamente interminable , ya que existen sensores para casi todas las necesidades de una vivienda y volvemos a hacer hincapié de que son muy económicos de ahí su gran éxito entre los aficionados a la electrónica, informática, mecánica etc…

2.8 RFID

2.8.1 ID-12LA Inovations

Este módulo receptor de radiofrecuencia fue el primer módulo que se probó en el proyecto, pero no tuvo gran éxito ya que era un poco inestable a la lectura de los tags.

Es un módulo más caro que el que se ha utilizado ya que es mucho más robusto y según el fabricante detecta los tags a una distancia de 12 cm ya que posee una antena que posibilita esta distancia .

Es un módulo que trabaja a 125 Khz

- alimentación de 5V

-Frecuencia de lectura de 125 kHz

- EM4001 etiqueta RFID compatible con 64 bits

-9600bps TTL y salida RS232

-salida de emulación de banda magnética

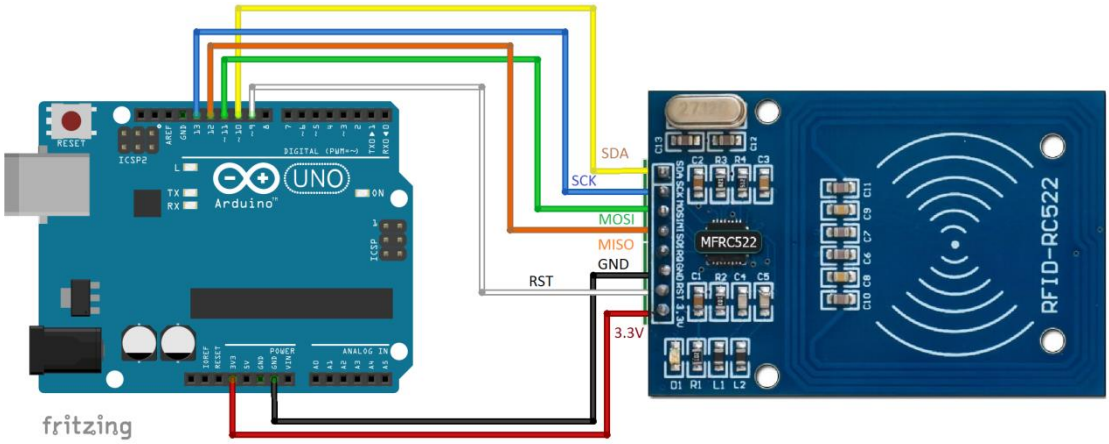
2.8.2 RFID RC522

Este es el modulo utilizado en nuestro montaje, es más económico y estable que el anterior no posee una lectura de tags a tan larga distancia como el anterior pero para la instalación que se ha querido crear ha sido más que suficiente su distancia de reconocimiento.

El lector está basado en el típico chip MFRC522 para lectura y escritura de tags **RFID**

El chip MFRC522. Da soporte a la lectura y escritura de los tags en diferentes condiciones y con control de errores, de una forma sencilla, aunque no puede identificar más que un tag a la vez, a diferencia de otros lectores más profesionales, y requiere que la distancia de lectura sea de una par de centímetros a lo sumo como ya se ha comentado, pero es lo que tiene ser más barato que sus competidores.

La tarjeta presenta un interface SPI para comunicar nuestro Arduino con el MFRC522 y no tiene más dificultad. De hecho creo que os sorprenderá ver lo fácil que resulta leer estas etiquetas RFID.

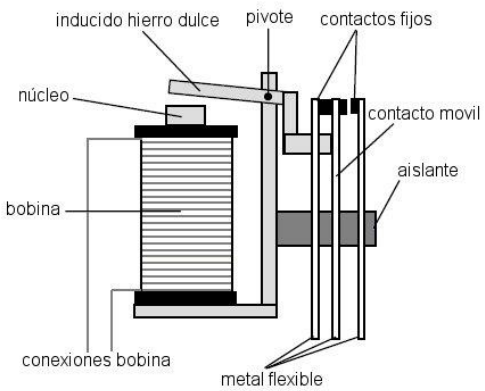


Con esta simpe conexión y bajándonos su librería, sacamos la información o el código del tag que vamos a utilizar para posteriormente acoplarlo a nuestro programa principal de control de acceso

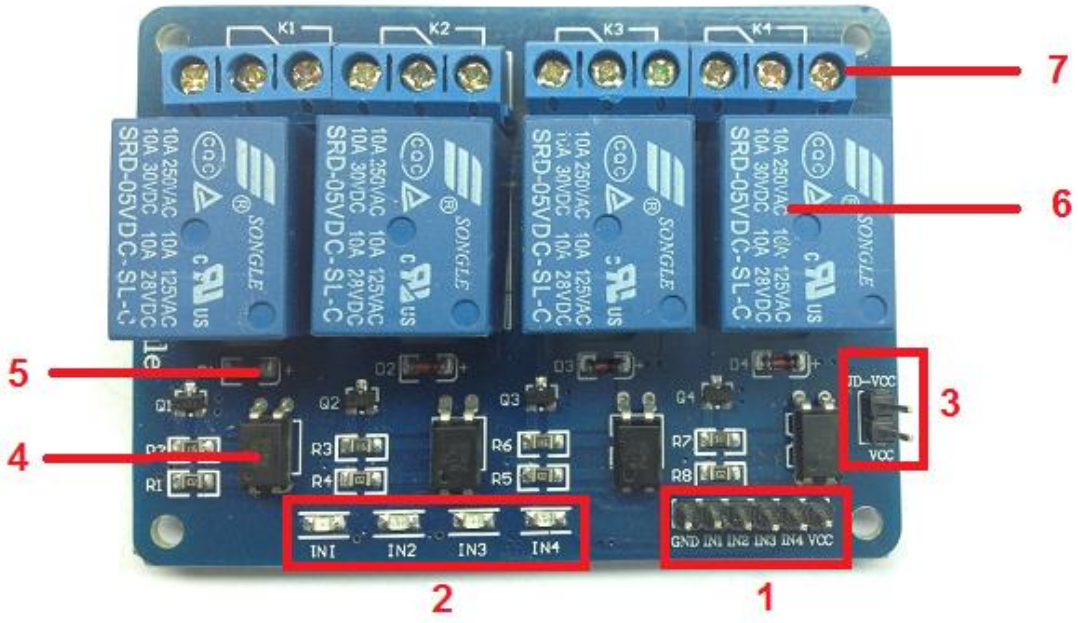
2.9 Relés utilizados

2.9.1 Optoaclopadores

Un relé es un dispositivo electromecánico que funciona como interruptor controlado por un circuito eléctrico en el que por medio de una bobina o electroimán, se acciona un juego de contactos que permiten abrir o cerrar otros circuitos eléctricos.



En nuestro proyecto hemos utilizado módulos de relés de diversos relés , los módulos que se han utilizado llevan un transistor para evitar cargar la placa aunque también los hay que no lo incorporan.



Como se puede apreciar, la placa tiene un conector de entradas (IN1 a IN4) y alimentación (GND es masa o negativo y Vcc es el positivo)

[1] Cuatro leds que indican el estado de la entradas

[2] Un jumper selector para la alimentación de los relés

[3] Cuatro opto acopladores del tipo FL817C

[4] Cuatro diodos de protección

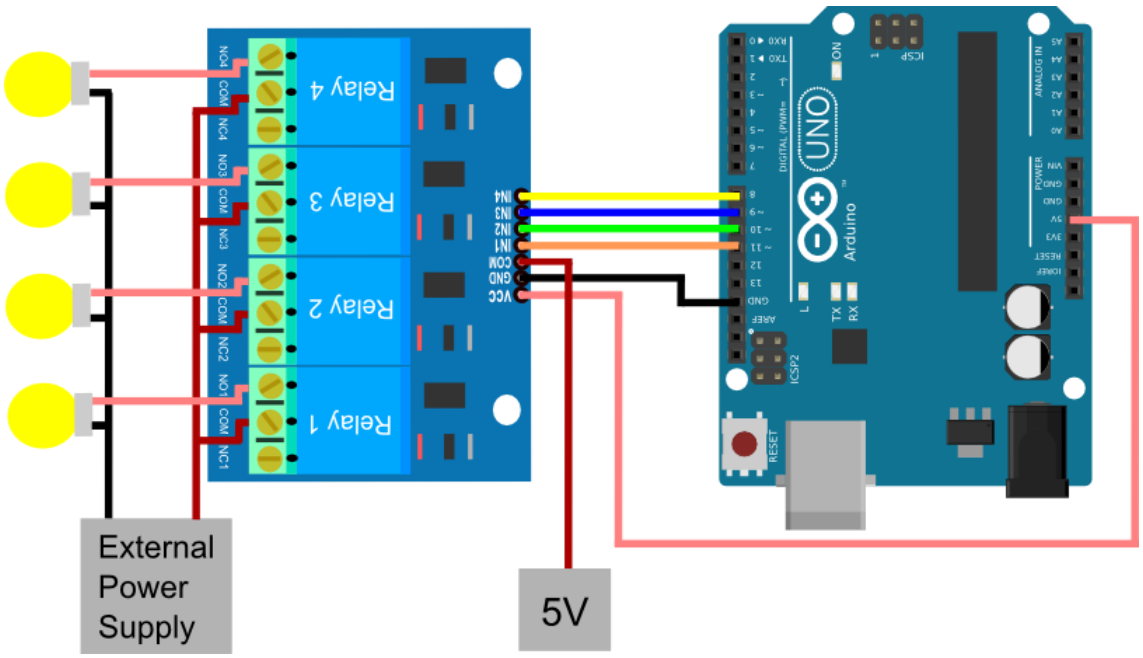
[5] Cuatro relés marca SONGLE con bobinas de 5V y contactos capaces de controlar hasta 10 Amperes en una tensión de 250V

[6]Cuatro borneras, con tres contactos cada una (Común, Normal abierto y Normal cerrado).

[7] Para las salidas de los relés

Con un relé vamos a poder manejar voltajes altos o potencias elevadas con pequeñas tensiones de control.

Tenemos que tener en cuenta que debido a la configuración de los contactos, los relés pueden ser NO (normalmente abiertos) y NC (normalmente cerrados). Los NO en ausencia de tensión en la bobina del relé estarán abiertos, es decir que no dejarán pasar intensidad. Por el contrario los NC se comportaran de manera inversa. Esto es importante porque dependiendo de esta característica tendremos que cambiar nuestra forma de actuar con el microcontrolador Arduino.



En esta imagen podemos apreciar lo sencillo que es poder trabajar con tensiones de 230v con un simple módulo de relés ,un arduino y un sencillo código con el que poder controlar hasta 4 circuitos independientes en este caso.

2.10 Módulos de audio

Antes de comenzar con este apartado, se quiere resaltar que los 3 módulos que a continuación se va a presentar han sido usados en el proyecto, se han tenido que usar los 3 debido a que desde el primero hasta el último nos han dado muchos problemas.

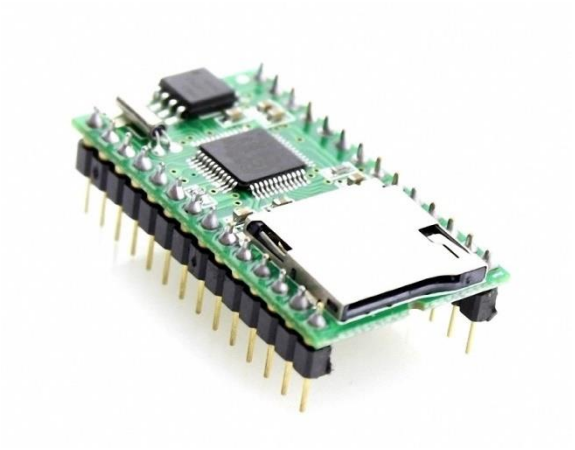
El primero que es el WT5001 ha sido probado sin éxito al igual que el WTV020M01, del cual se han llegado a probar hasta 3 módulos diferentes sin resultado alguno , todos con diversas tarjetas de diferentes marcas y de 1 y 2 gb , como dice el fabricante.

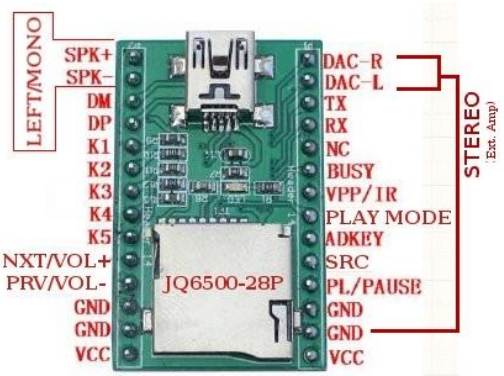
Incluso se ha montado independientemente de arduino con un regulador de tensión a 3.5v sin resultado ,se comenta lo de la tensión de alimentación ya que al ser un módulo chino hay variedad de opiniones circulando por la red, en el módulo pone que se alimente a 5v, pero en el datasheet a 3,3 que es una de las salidas de arduino pero ni así , entonces tras documentarnos bastante llegamos a la conclusión que este módulo no es que tenga conflicto con las micro sd si no con su voltaje de alimentación, entonces se probó a aislar el modulo del microcontrolador y hacerlo funcionar en la función que se le denomina walkman o reproductor de mp3 y alimentándolo a 3,5 v , pero lo más que se llegó a conseguir en esta prueba fue unos pequeños sonidos a través del altavoz cada vez que se reproducía una pista .

Debido a todo este intenso trabajo, nos decantamos por el módulo somo 14d que está fabricado por un fabricante experto y fiable como es Sparkfun, la diferencia está en el chip que monta y claro también en su precio , ya que un módulo chino está alrededor de unos 3e y el somo cuesta 25e.

Para decantarnos por este último modulo nos estuvimos informando y documentando por la red, hasta que dimos con un distribuidor con el cual mantuvimos con el departamento técnico un contacto a la hora del montaje ya que en esta parte del proyecto hubo desesperación por no llegar a la solución final deseada ya que no dependía de nosotros sino del material.

2.10.1 WT5001





Este modulo de sonido, se utiliza como los otros 2 que se van a describir , para implementar sonido o voz a los montajes que se realicen.

Sus caracteristicas son:

-salida de audio estereo

-altavoz 8 hom y 1w

-salida busy

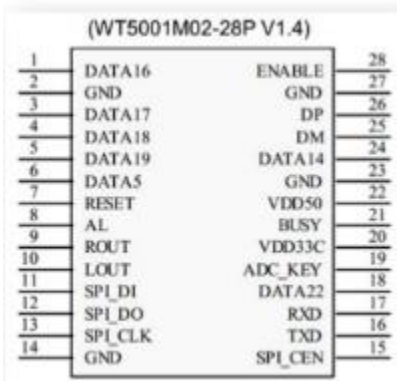
-soporta wma y mp3

-apoyo spi flash para memorizar audio interno

-soporta comunicación rs232 mediante los pines de arduino tx y rx

- posee una fuente anti intenferencias

- voltaje de 3.3 y 5 v



2.10.2 WTV020M01

Este es el modulo más usado y del que más información se puede encontrar en la red debido a su bajo coste como se ha mencionado en la red , pero a su vez también es el más inestable como también se ha mencionado.

El decodificador WTV020SD-16P es un dispositivo de bajo costo (unos 2,35€) que nos va a permitir añadir sonido a nuestros proyectos electrónicos con una calidad de sonido superior a como si lo hiciéramos con el propio Arduino.

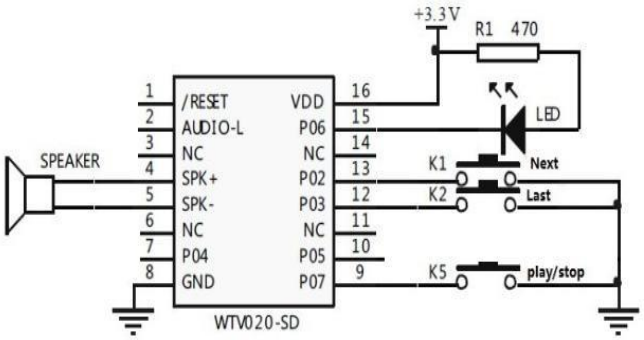
Vamos a explicar cómo es y cómo funciona este dispositivo. En las siguientes imágenes podemos ver cuáles son sus pines y la función de cada uno de ellos:

PIN	SYS.	DESCRIPTION	FUNCTION
1	RESET	RESET	Reset pin
2	AUDIO-L	DAC+	DAC audio output(+) to amplifier
3	NC	NC	NC
4	SP+	PWM+	PWM audio output to speaker
5	SP-	PWM-	PWM audio output to speaker
6	NC	NC	NC
7	P04	K3/A2/CLK	Key /CLK in two line serial
8	GND	GND	Address pin
9	P07	K5/A4/SBT	Key
10	P05	K4/A3/DI	Key /DI in two line serial
11	NC	NC	NC
12	P03	K2/A1	Key
13	P02	K1/A0	Key
14	NC	NC	NC
15	P06	BUSY	BUSY pin
16	VDD	VDD	Power input

Con este módulo podemos trabajar de dos modos:

- De forma autónoma (sólo con una pequeña circuitería). - Conectado a nuestro Arduino y utilizando una librería.

Vamos a ver el primer caso. Para ello deberíamos montar el siguiente esquema:



Básicamente nos valdría con alimentar el dispositivo (3,3V a VDD-> Importante!!!, con más tensión podríais deteriorarlo, y 0V a GND). Luego conectaríamos un pulsador para el RESET y otros dos (P02 y P03) para avanzar/retroceder en las pistas de audio. Por último, conectaríamos un altavoz a los pines SPK+ y SPK-.

El circuito no tiene más dificultad. Ahora sí, debemos tener en cuenta que el dispositivo lee los ficheros de audio en formato .ad4. Para ello debemos convertir nuestros ficheros .mp3 o .wav a dicho formato.

Recomendaciones para no tener problemas con los ficheros de audio:

- La tarjeta microSD debe ser de cómo mucho 2 Gb.

- La tarjeta microSD debe estar formateada como FAT32.

- Los archivos de audio deben ser codificados en 4

-bit ADPCM.

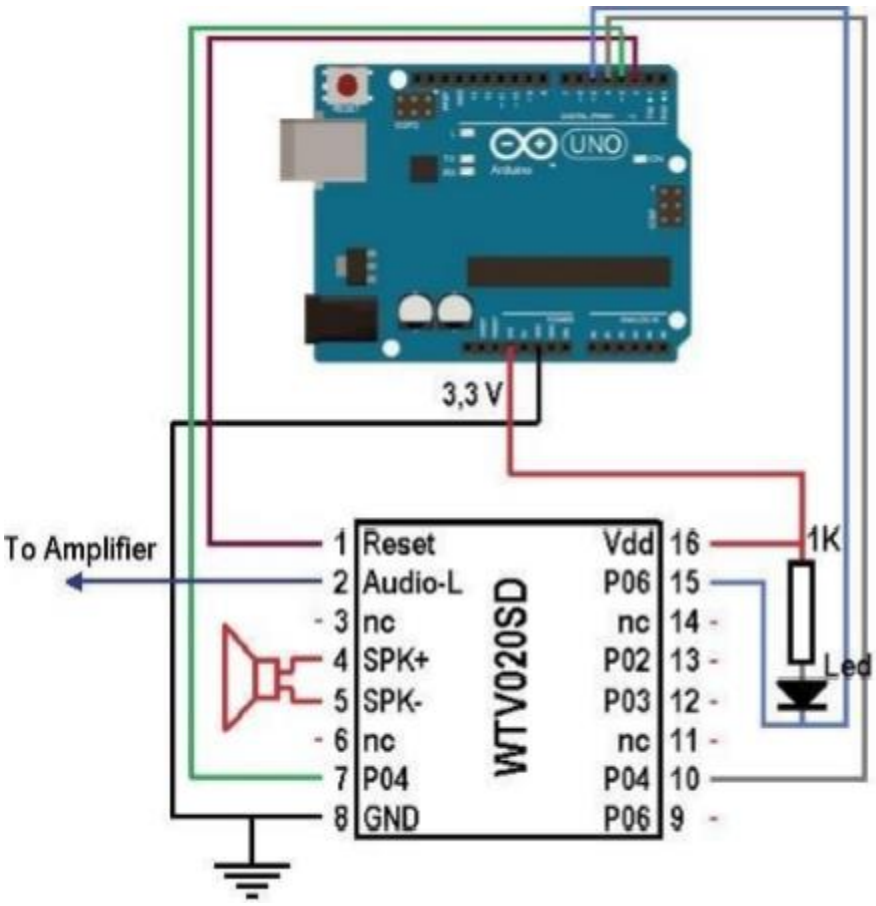
- Los ficheros de audio deben de tener la extensión .ad4

- Los ficheros de audio deben de estar en el raíz de la tarjeta microSD, y no dentro de carpetas (y preferiblemente solos en el raíz, sin más archivos de otro tipo).

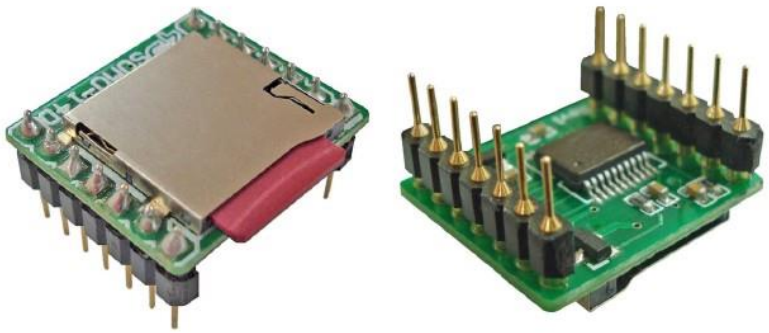
- Los ficheros de audio los debemos renombrar empezando por el 0000.ad4 y hasta el 0511.ad4.

Si optamos por el otro modo (conectado al Arduino), deberemos instalar en nuestro IDE la siguiente librería Wtv020sd16p, y conectar nuestro módulo al Arduino según la siguiente tabla y esquema:

VCC	16 (VDD)	3,3V
GND	8 (GND)	GND
RESET	1 (RESET)	D2
CLOCK	7 (P04)	D3
DATA	10 (P05)	D4
BUSY	15 (P06)	D5

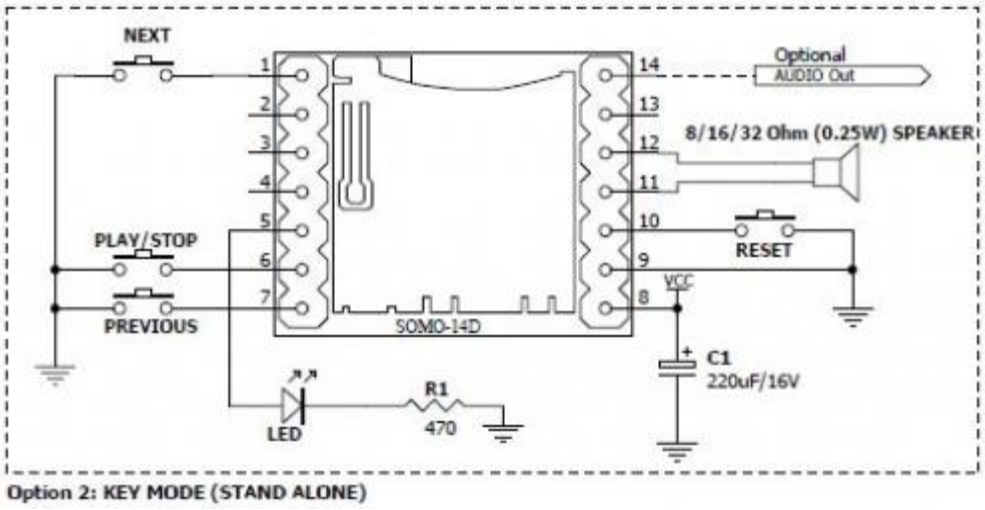


2.10.3 Somo 14 D



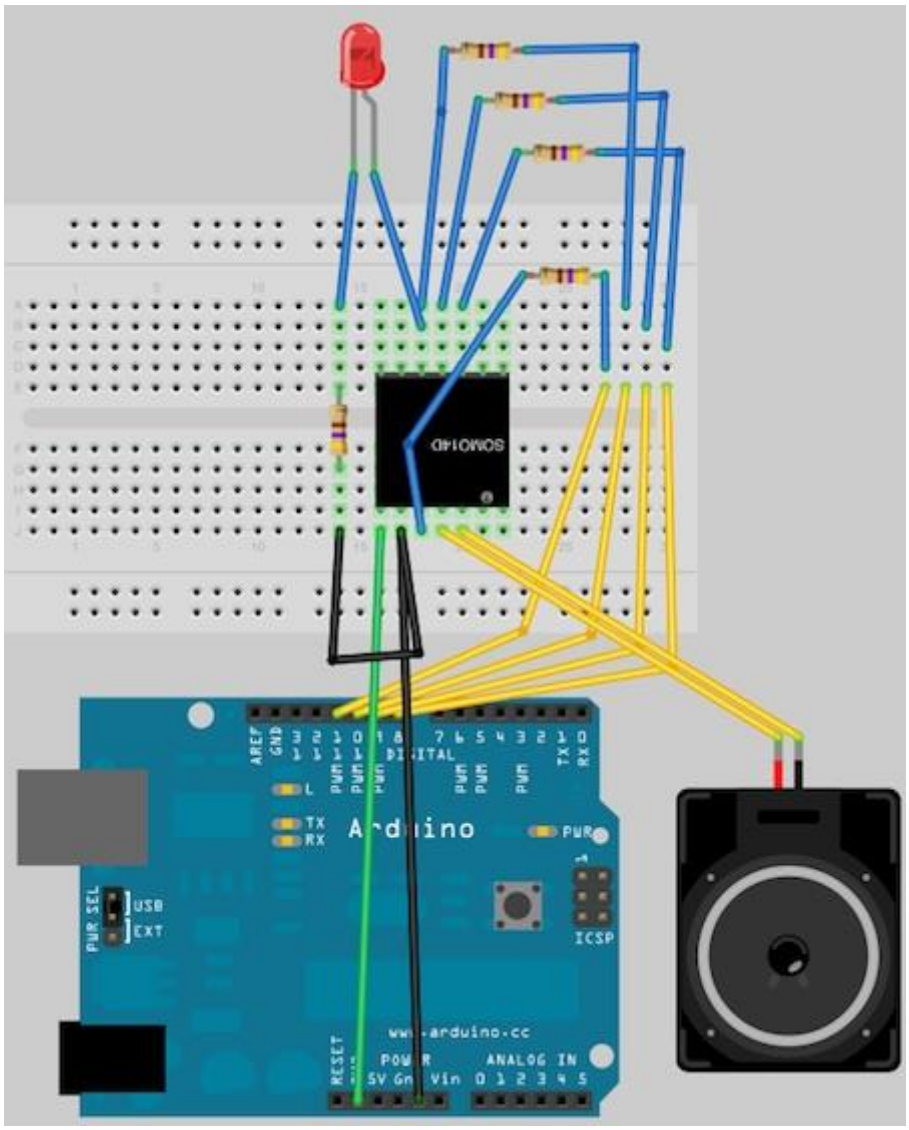
Este módulo es por el que nos hemos decantado tras muchas pruebas, en cuanto a funciones alimentación y ficheros, es exactamente igual que el WTV020M01 , pero el voltaje sí que es exactamente 3,3 que se puede usar directamente el que nos proporciona la placa de arduino y las tarjetas no te restringe el fabricante ni en cuanto a marca ni a capacidad.

Lo único que cambia un poco es su interior y patillaje como vamos a ver ahora:



Option 2: KEY MODE (STAND ALONE)

Y aquí tenemos un ejemplo bastante visual de como sería un montaje con una tableta de arduino, en el que se aprecia el led de la salida busy, que su función es que esta encendido y cuando una pista se está reproduciendo se apaga.



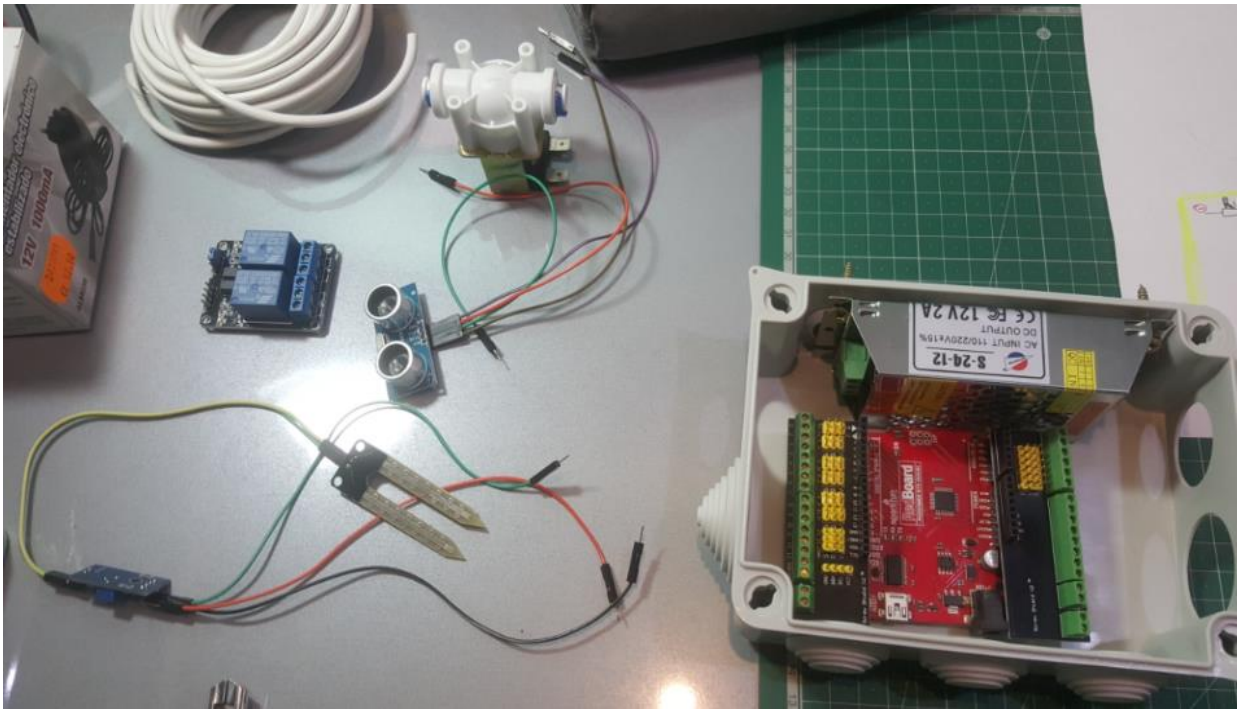
3 Visualización real de los montajes y de sus componentes

En este apartado nos vamos a centrar en explicar los componentes utilizados en cada apartado del proyecto y en su montaje en la placa board.

3.1 Control de nivel de llenado del tanque y nivel de sal descalcificadora

Esta es la primera parte de las 3 que consta el proyecto de la casa domótica con arduino.

Esta parte del proyecto surgió por la necesidad de controlar el llenado de un tanque de agua y el nivel de sal de una descalcificadora.



Los componentes que se han utilizado son:

- Arduino uno con su módulo de expansión para poder atornillar el cableado

- Modulo de 2 relés

- Electroválvula para el corte del llenado del tanque

- Transformador de 12v para la alimentación del arduino

- Sensor de ultrasonidos

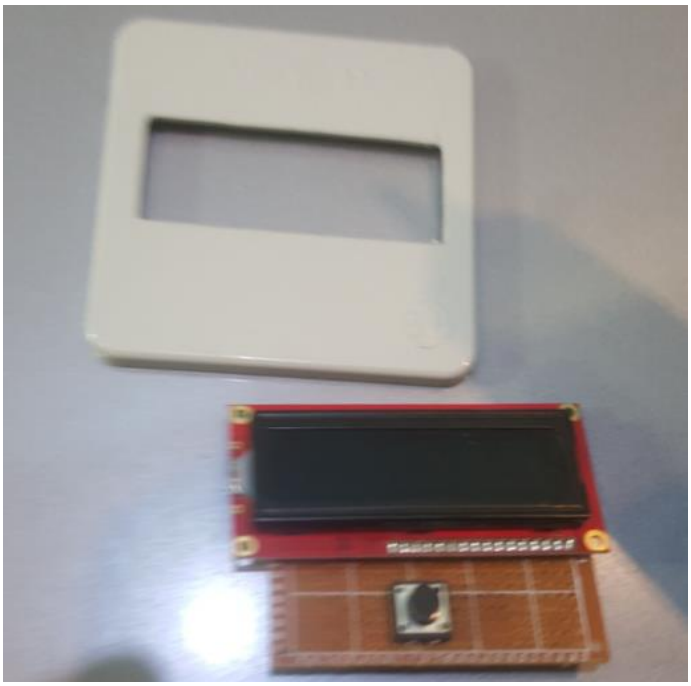
- Sensor de humedad aunque en el montaje real se sustituyó por un sensor de rebose

- Fuente de alimentación conmutada 24v cc para alimentar la electroválvula

- Electroválvula 24v cc

- Cable multifilar apantallado

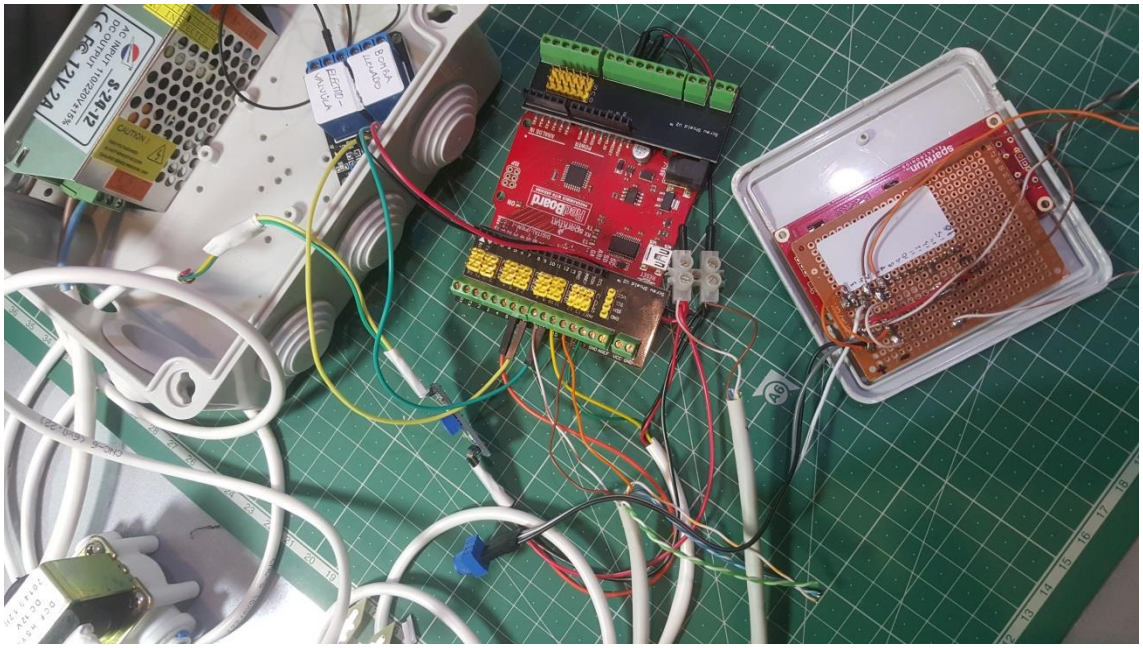
- 3 cajas estancas de superficie y una placa multiperforadora



-Una pantalla lcd

-Un potenciómetro para regular la intensidad de la pantalla.

-Un pulsador para activar la bomba de agua para poder evacuarla del tanque



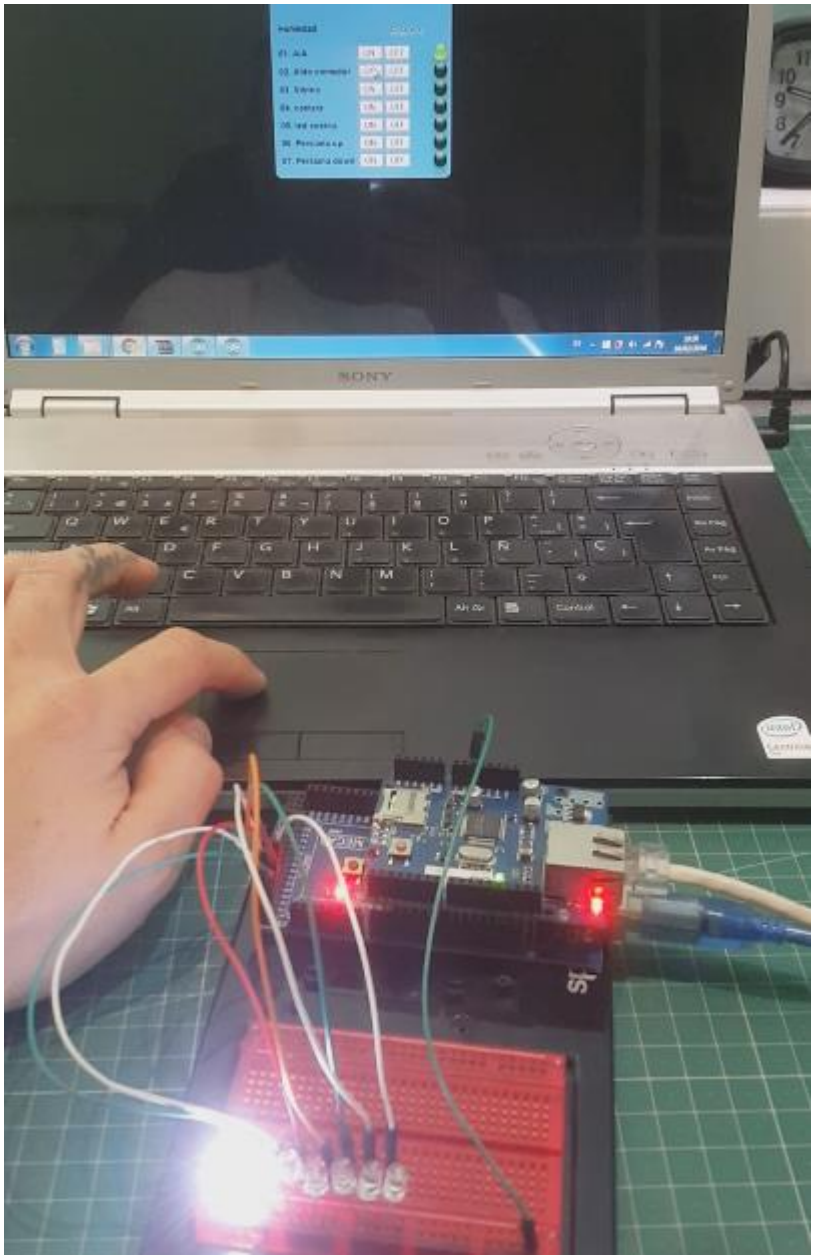
Este es el aspecto del montaje en la mesa de pruebas sin conectar

Y este es el aspecto del montaje funcionando

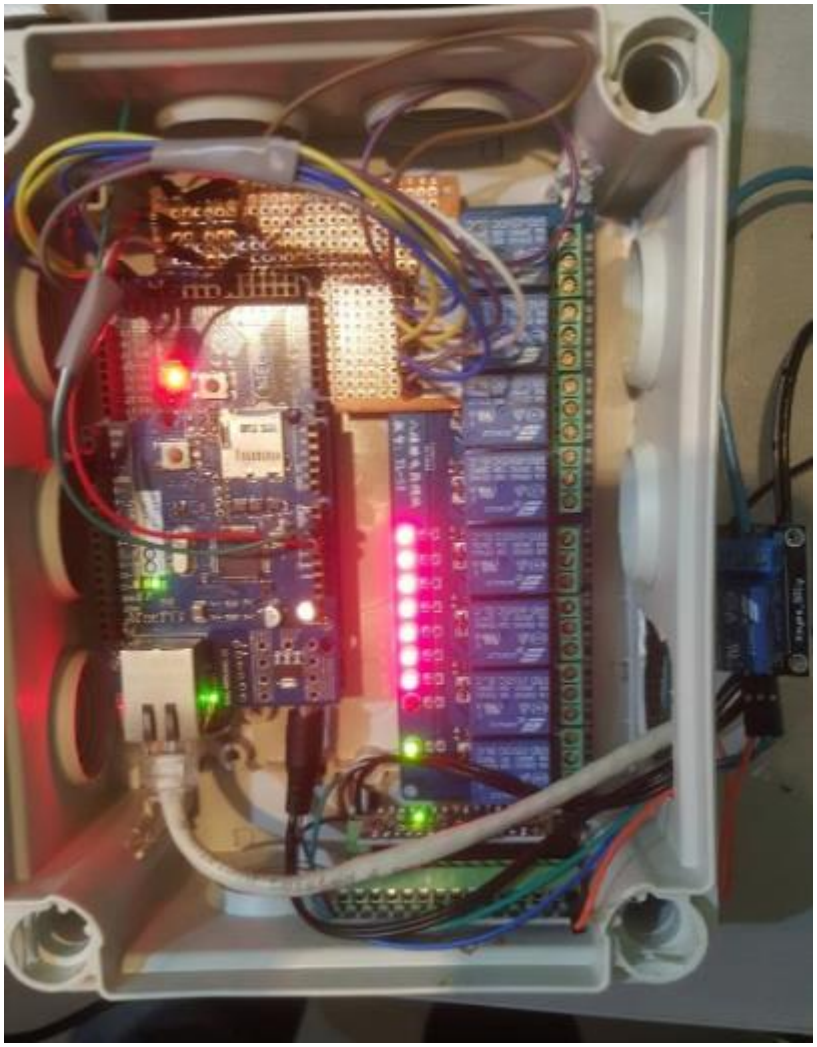
3.2 Control Demótico a través del servidor web Arduino

Para crear esta parte de la instalación, partimos de un programa que se encontró en la red, el cual se modificó y se ajustó a nuestras necesidades.

En el punto siguiente nos centraremos más en profundidad en cada apartado explicando detalladamente cada elemento.



Esta primera captura , fue la primera prueba realizada con nuestro servidor ya creado y en lugar de los circuitos de casa , se colocaron leds simulándolos



En este apartado, para lo que es el servidor donde está la información de nuestra web se han utilizado los siguientes materiales:

-Un arduino mega

-Una shield de Ethernet

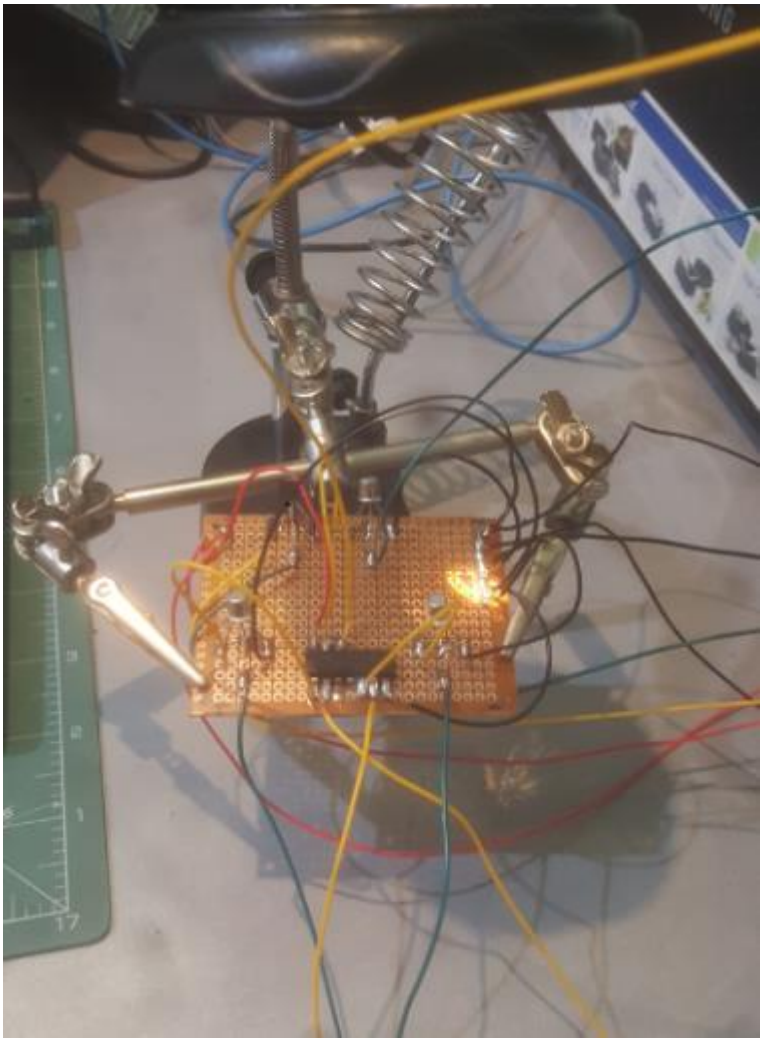
-2 módulos de relés de 4 relés por modulo

-2 placa multiperforadas

-Un arduino nano

-1 relé para cortar la tensión al mega

- Una fuente de alimentación para alimentar a los arduinos



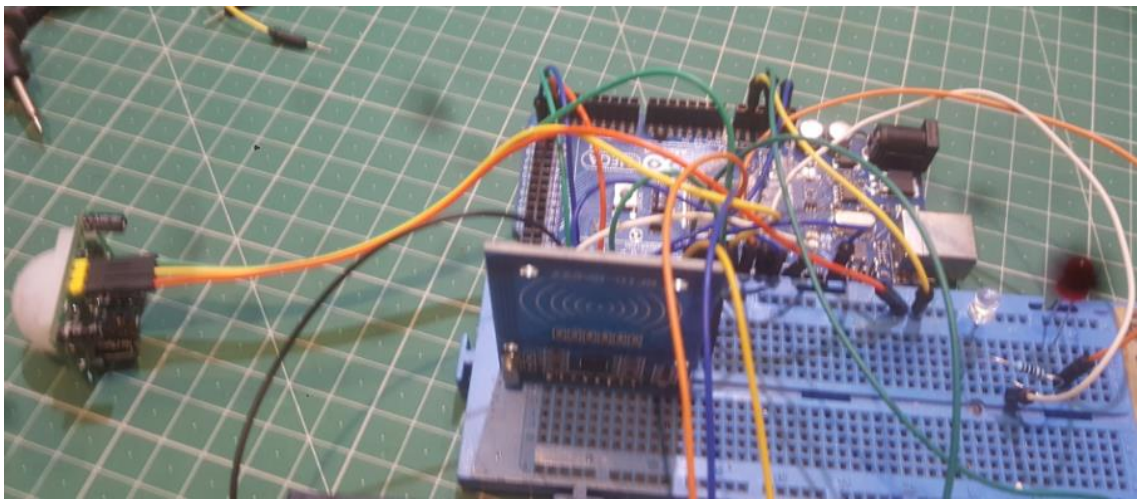
-4 transistores NPN para no sobrecargar la placa

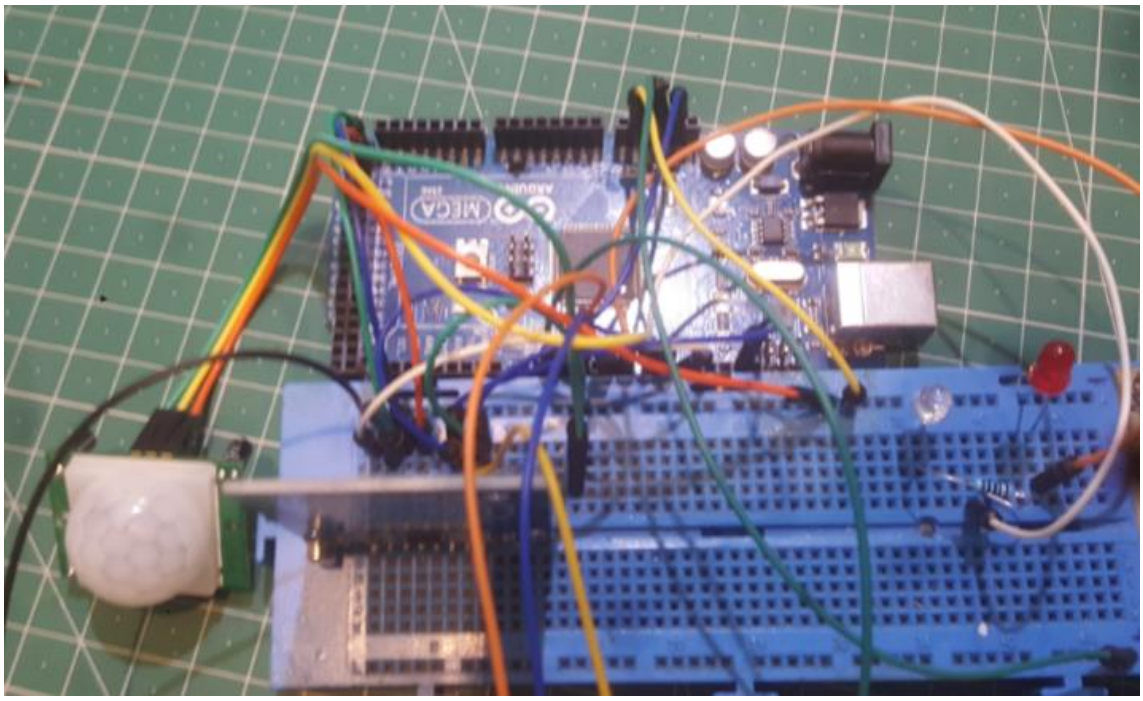
- Una segunda fuente de alimentación para alimentar los relés y transistores

- Un integrado de puertas Not para invertir algunas salidas

3.3 Acceso mediante Control de RFID y estados por voz

Esta última parte del proyecto se dividió en la mesa de montaje por un lado se creó lo que sería la identificación por radiofrecuencia y por otro el módulo de voz para cuando estuviese todo correcto poder fusionarlos.



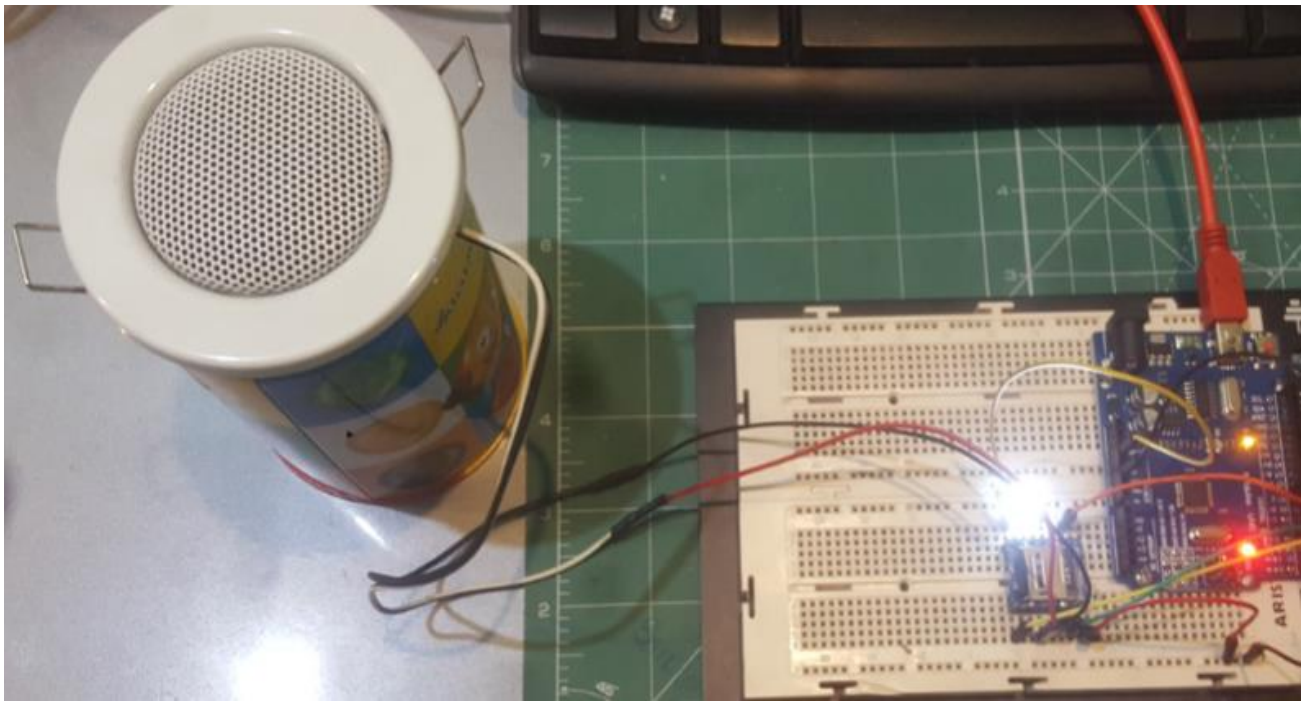


Esta es la parte del montaje de identificación por radiofrecuencia, en la cual hemos utilizado los siguientes materiales:

-Un arduino mega

-Un módulo RFID

-Un sensor pir para simular la detención d la puerta de acceso a la casa



En esta parte que es la de la voz , en la board se probó con un Arduino nano pero que luego no se utilizó ya que se usa el mega para todo.

Aquí se ha utilizado:

-Módulo de voz Somo 14-d

-Arduino uno

-Altavoz 1w y 8 hom

-Cableado

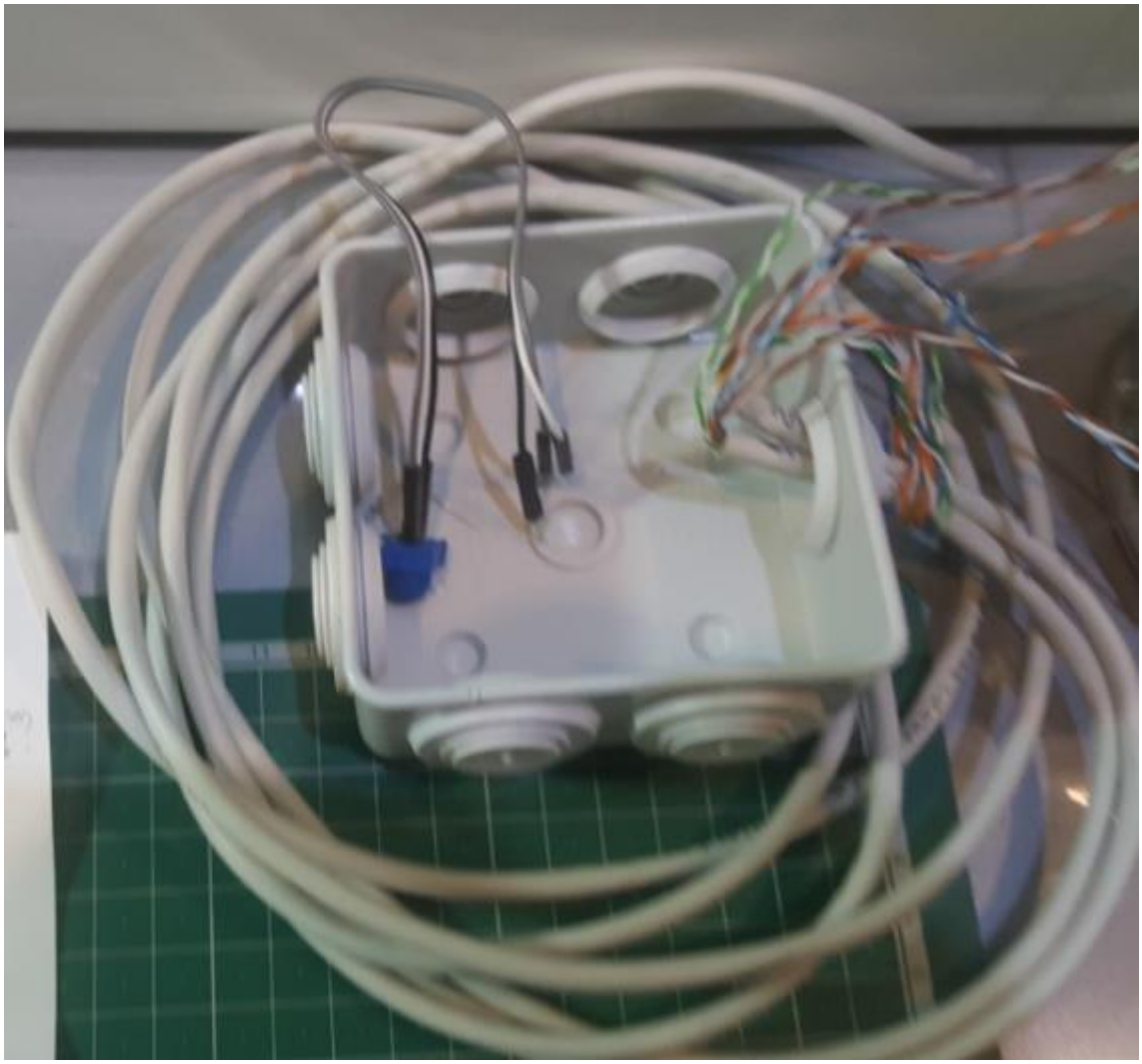
4 Diseño del sistema

4.1 Bloque de control de llenado de tanque y nivel de sal descalcificadora

4.1.1 Arquitectura del sistema y descripción del mismo

Nuestro tipo de arquitectura es de tipo centralizada, como ya se comentó en el apartado de tipología de arquitectura, esta es la instalación, en la que los elementos que vamos a controlar y supervisar como pueden ser , sensores ,luces ,válvula , etc. se tienen que cablear hasta donde se encuentra el sistema de control, que puede ser un pc , una centralita ,un microcontrolador embebido el cual es nuestro caso.

Si falla nuestro sistema de control, se cae todo ya que todo depende de él, y la instalación se tiene que hacer independiente a la instalación eléctrica y prever eso a la hora de hacer la preinstalación.



Comenzamos cableando los cables que van a comunicar el arduino con la pantalla, para mostrar la información que registre de los 2 sensores , el de ultrasonidos y el de la descalcificadora

Aquí tenemos montada la caja que nos hará de display, se ha colocado un pulsador el cual está conectado al arduino y cuando se pulse , este lo leerá y nos activara la bomba de agua que tenemos introducida dentro del tanque , durante 1,45 min que es el tiempo que tarda en llenarse una garrafa de agua de 25l con el caudal que nos da esta pequeña bomba.

El dispositivo tiene 5 estados

-Descalcifica: ok que significa que hay suficiente sal

-Descalcifica: sal , cuando falta sal

-Tanque lleno,

-Tanque llenándose

-Pulsador activado vaciando tanque

Aquí tenemos centralizada lo que seria la parte de potencia, donde tenemos la fuente de alimentacion de 24v , para alimentar el rele de la electrovalvula ya que esta funciona a 24 v en cc.

Aquí , se puede apreciar la electrovalvula por un lado conectada al filtro de osmosis y por el otro a su rele.

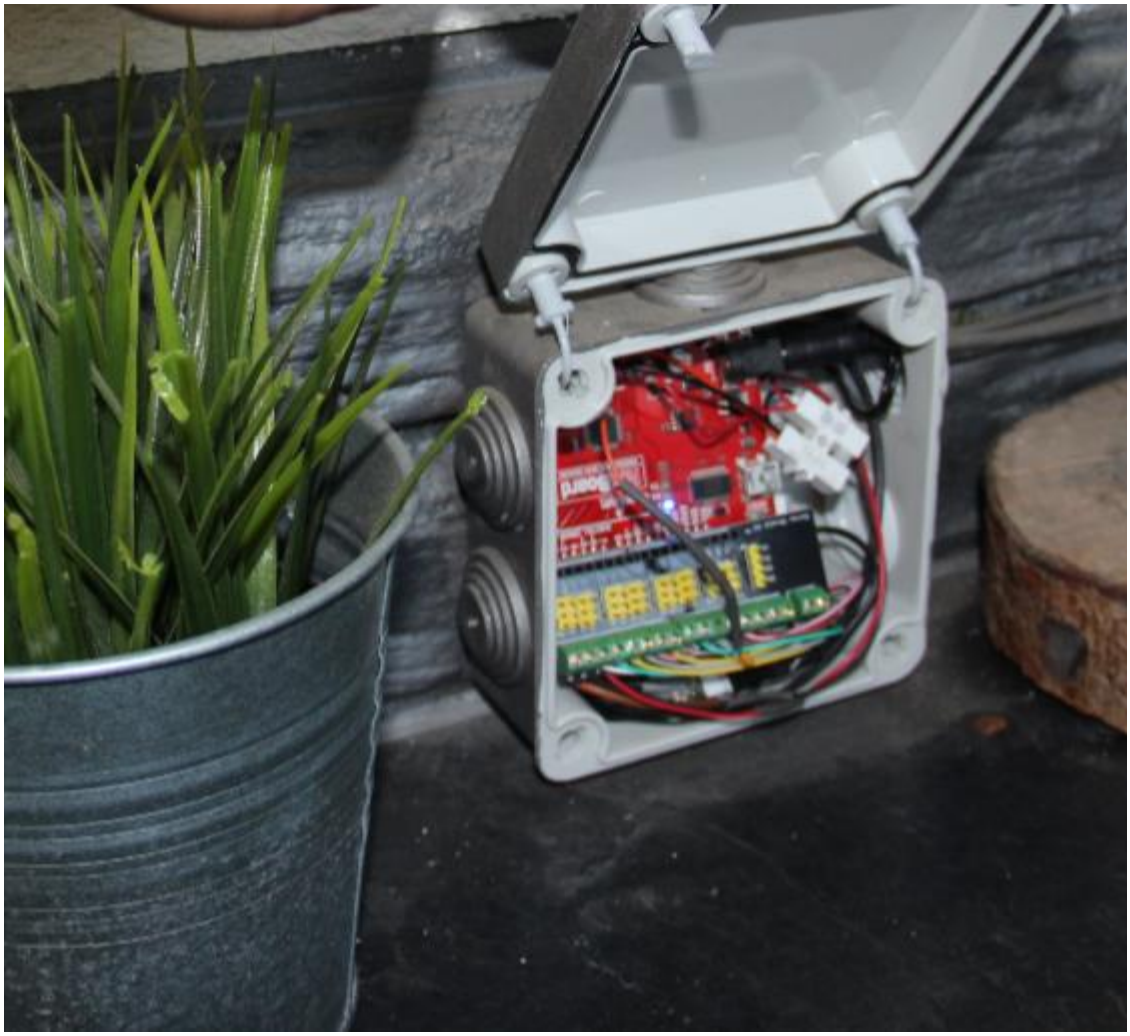
El otro rele es para la activacion de la bomba de agua y esta conecta a 230v el ac.

Estos 2 reles claro esta, estan gobernados por la señal de arduino

Como los cables multifilares que se han utilizado son apantallados para evitar el ruido o armonicos generados por el motor del grupo de presion que se encuentra en el mismo lugar de la instalacion , y que nos dio problemas en su dia , se unificaron las mallas de los cables y se llevaron a tierra o sea a la pared de la casa a traves del cable marron que se aprecia en la captura.

Aquí tendríamos el sensor de ultrasonidos conectado a la descalcificadora, el cual nos manda la información al arduino midiendo la distancia desde donde se encuentra el y donde está la sal , cuando esta distancia supera los 40 cm , el arduino nos imprimirá en la pantalla que nos falta sal, como se aprecia en la captura anterior de la caja de la lcd.

En esta imagen se puede apreciar el nivel de cableado utilizado y en la parte izquierda tenemos la alimentación de arduino atreves de un transformador de 12v.

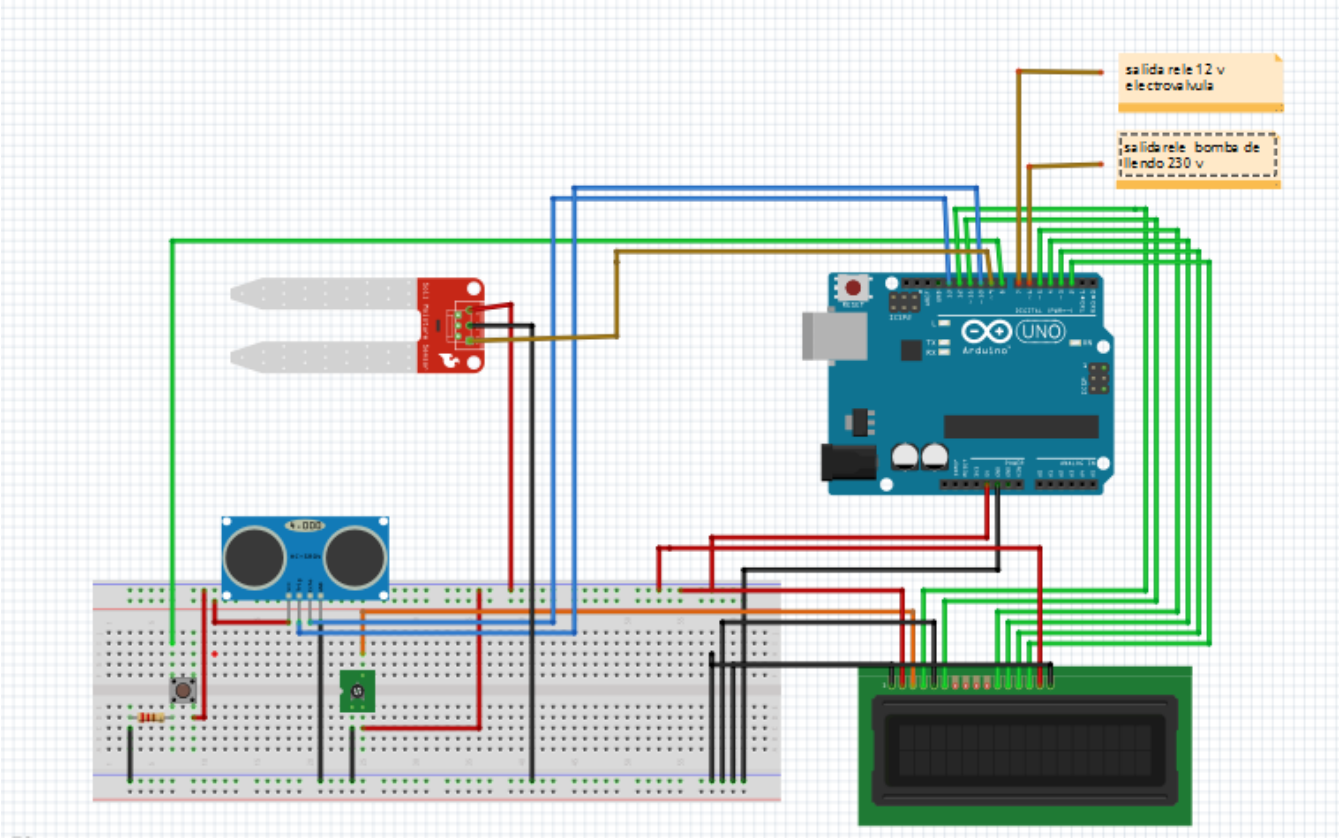


En esta última imagen, vemos la arquitectura centralizada de esta parte del proyecto, donde se aprecia el arduino conectado con todos los elementos, sensores, relés, pantalla.

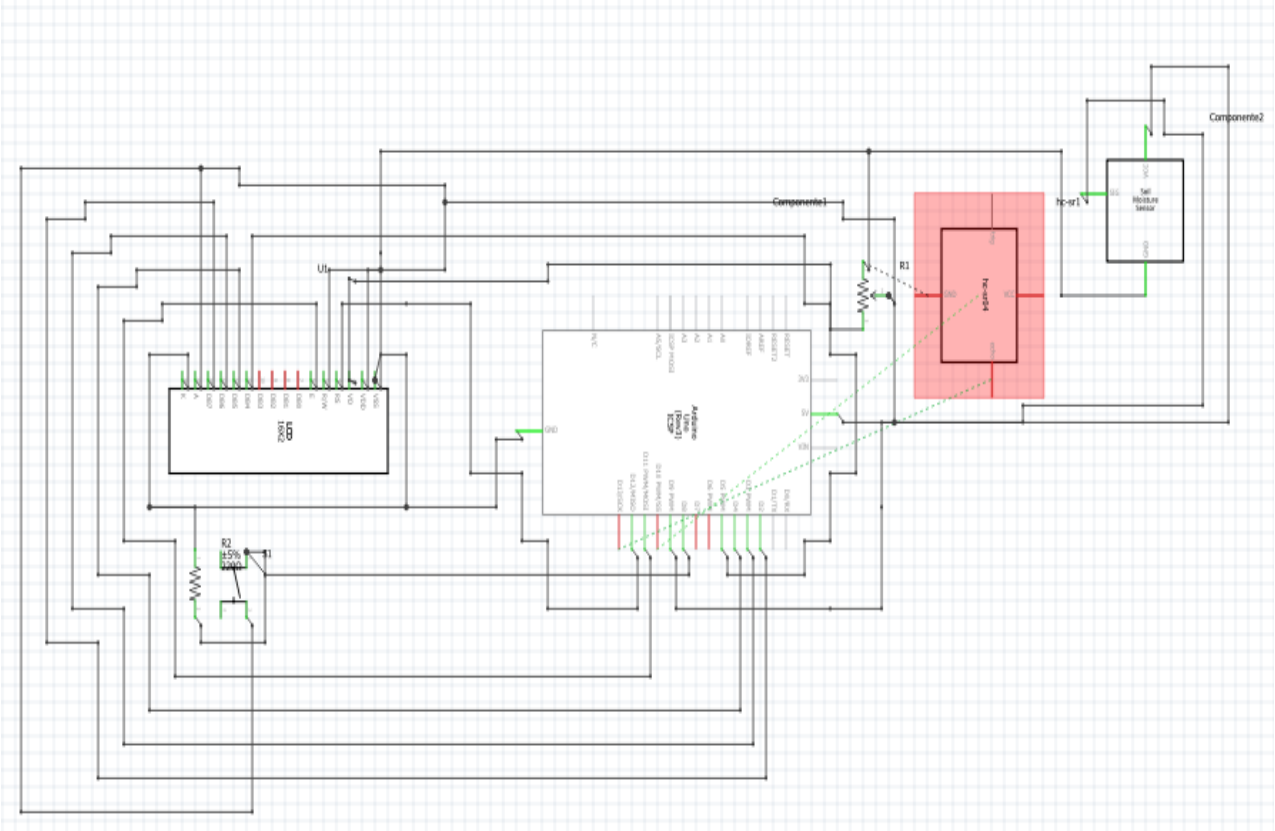
Se ha tenido que usar un módulo de expansión tipo escudo para poder atornillar todos los cables ya que son muchos y de no ser así sería imposible.

4.1.2 Esquemas

Esquema de montaje

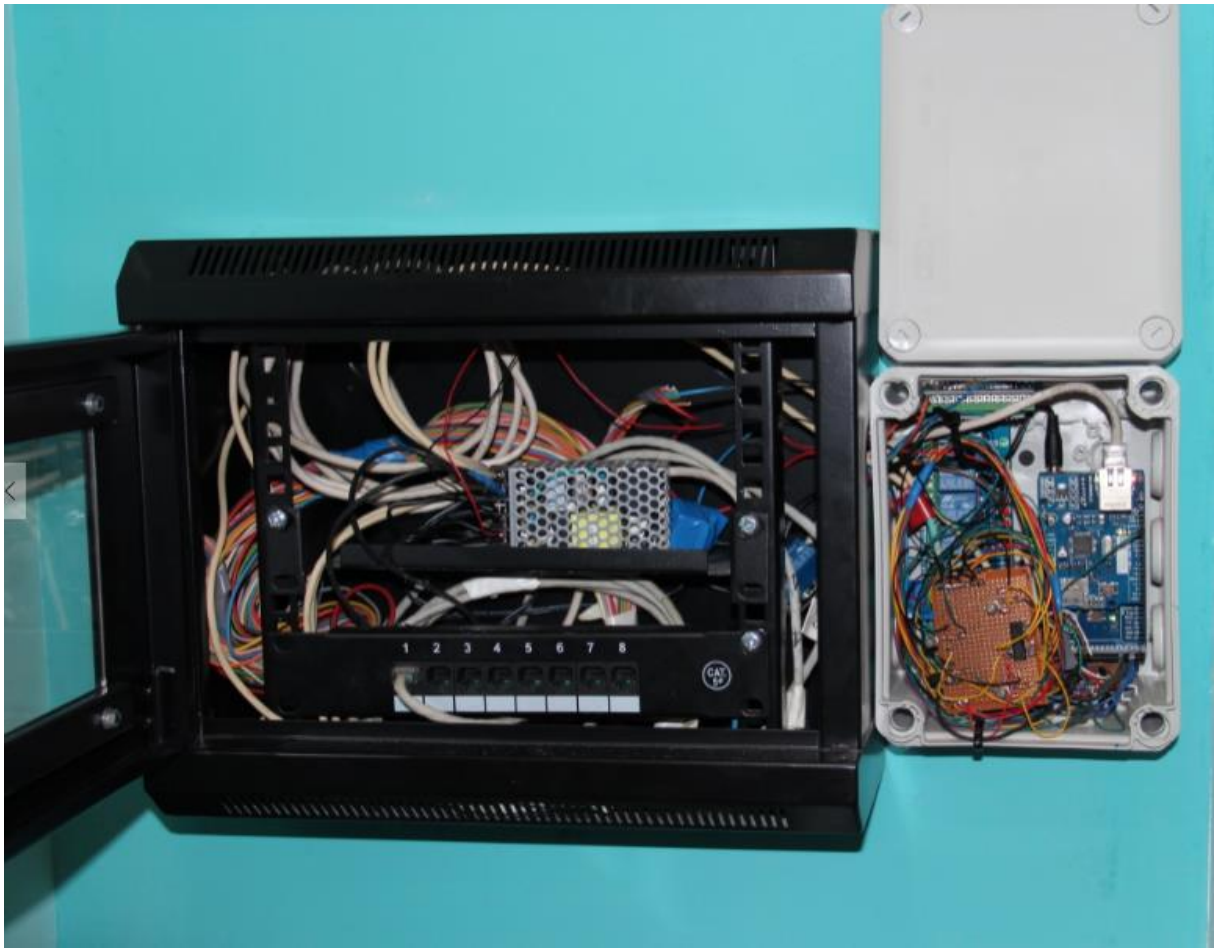


Esquema electrónico



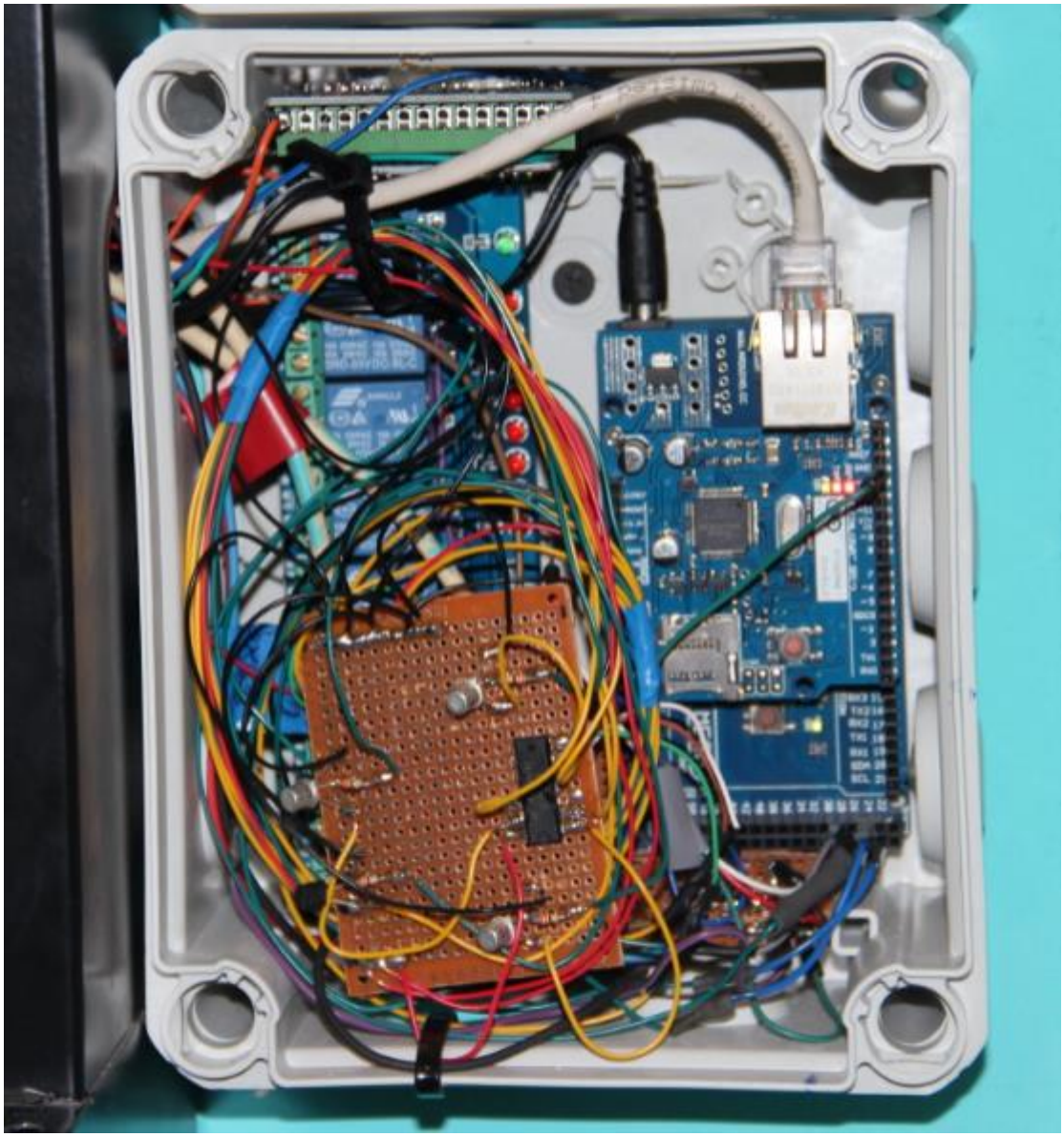
4.2 Servidor web Arduino

4.2.1 Arquitectura del sistema y descripción del mismo



La arquitectura de esta parte del proyecto también es centralizada, como en las otras dos partes, quiere decir que todo el sistema va conectado al arduino mega que tenemos en el interior de la caja estanca.

En esta caja tenemos 4 elementos diferenciados:



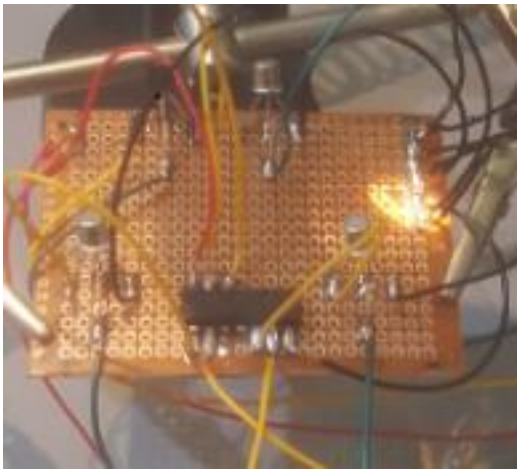
Tenemos el arduino mega con su escudo de Ethernet con el que se ha creado el servidor web , para utilizarlo como interface y gobernar a través de los circuitos de la vivienda.

También tenemos un módulo de relés el cual se cabio de ubicación y se trasladó debajo del cuadro eléctrico de la casa donde está el cuadro de telecomunicaciones, así nos resultó más fácil acceder a los circuitos de la casa.

También se puede apreciar la placa multiperforadora donde se encuentran los 4 transistores y el integrado 7404 de puertas not, los transistores se colocaron debido a que si

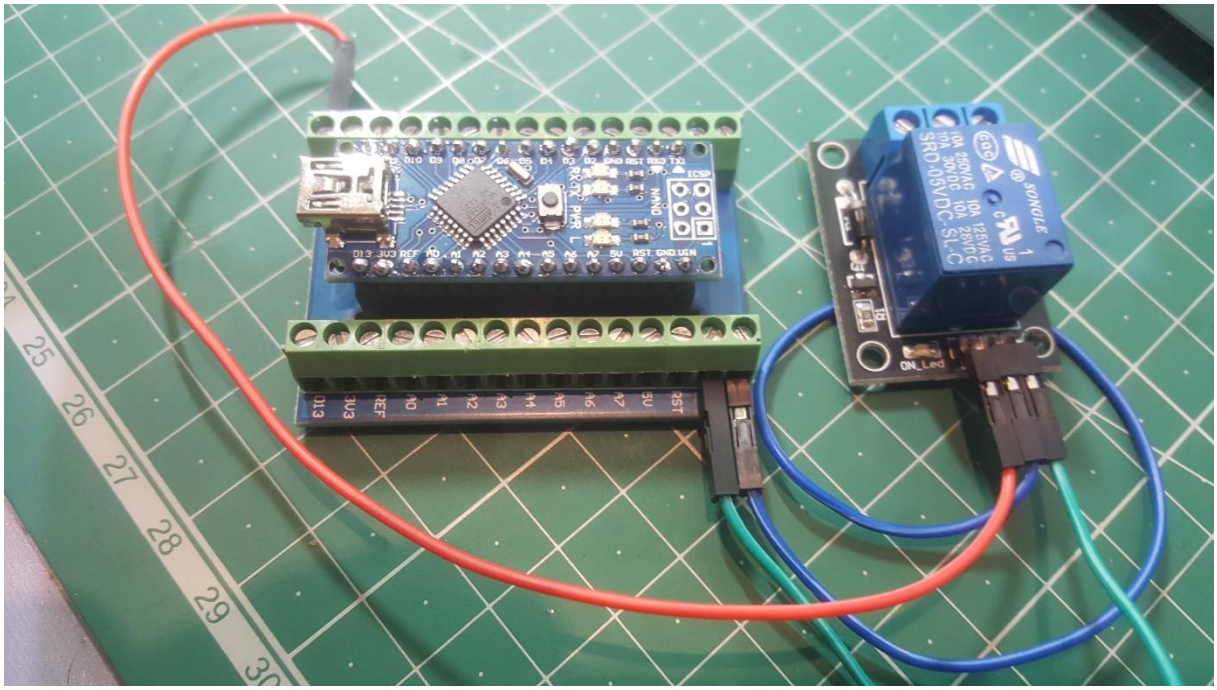
alimentábamos desde la placa directamente los relés que van a los circuitos la tensión del arduino caía y se nos bloqueaba, entonces la solución fue alimentar con una fuente externa a la del arduino , los 2 módulos de relés y el colector de los transistores, y se unificaron masas con la fuente de arduino. Con esto , lo que se consiguió fue que arduino no soportase carga , que la señal que sale de la placa de cualquier pin pase estimule la base del transistor y este se accione mandándole la señal a su correspondiente relé.

El 7404 , se colocó debido a que el servidor nos mandaba las señales con 0v en lugar de con 5v cuando tenía que activar una señal, se revisó todo el código y al no encontrar ninguna anomalía se declinó por esta solución la cual nos dio el resultado que se deseaba.



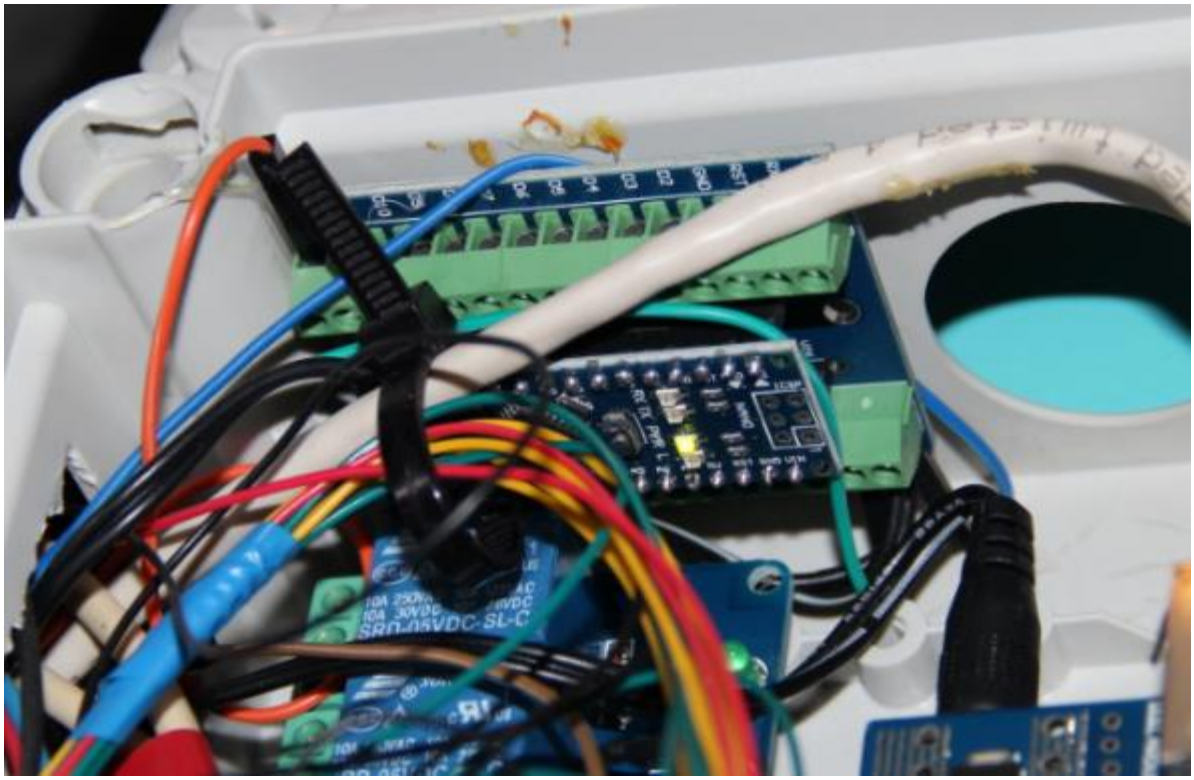
A la parte izquierda de la captura se observa el rac de telecomunicaciones de la vivienda, el cual se ha utilizado para dar servicio de internet a la shield de Ethernet y para alojar las 2 fuentes de alimentación, a través de este rac , se ha instalado el cableado multifilar que comunica con los relés de los circuitos que se encuentran en el cuadro de telecomunicaciones.

Debido a una serie de complicaciones ,con el servidor , ya que cuando se encontraba mucho tiempo esperando una petición de cliente este se bloqueaba y había que hacerle un reset desde la tarjeta y era un problema ya que la caja estaba cerrada, lo que se hizo fue colocar un arduino nano con el cual haciendo un sencillo código reseteamos la alimentación del mega cada 10 minutos, a través de un relé.

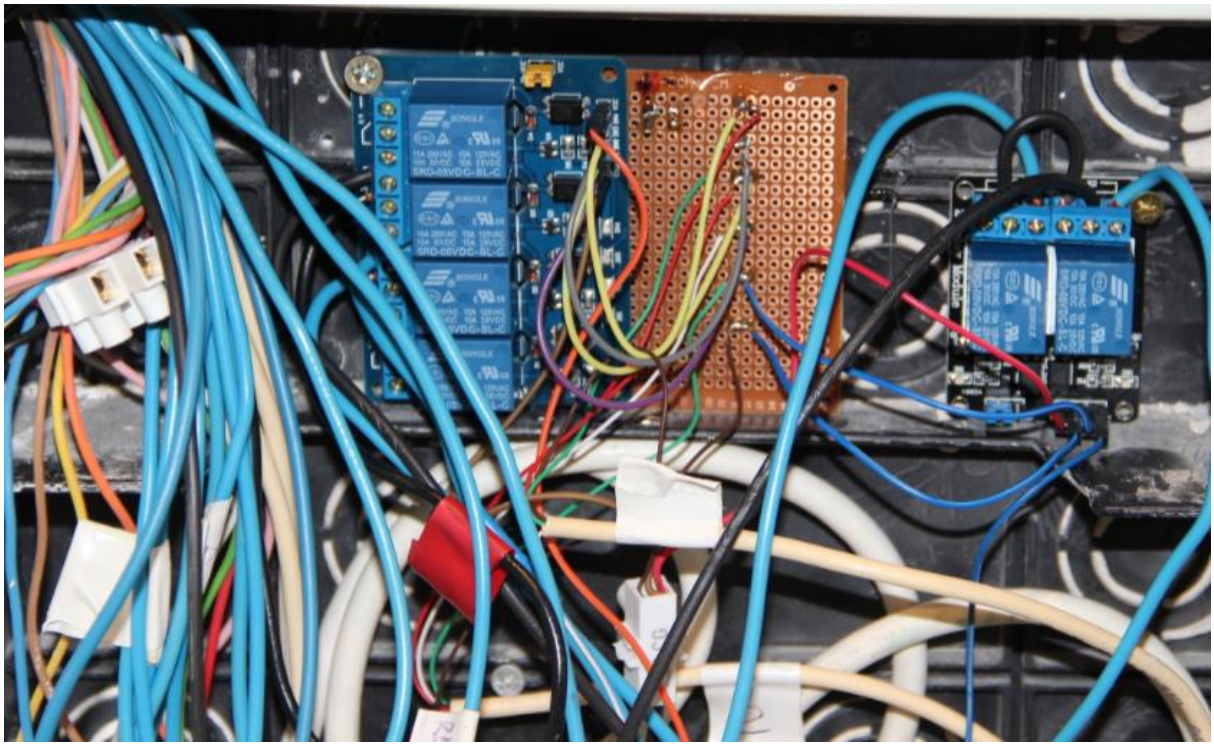


Esta captura es de las primeras pruebas que se hicieron con el nano, y el la de abajo se aprecia cómo se instaló en la parte superior de la caja de maniobra del servidor.

El relé para cortar el suministro del mega se encuentra en el rac de telecomunicaciones.

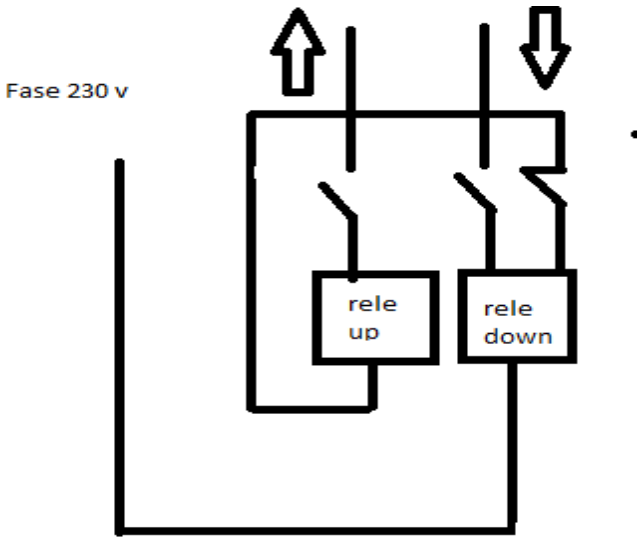


Como se puede observar aquí estaría lo que podríamos denominar como la parte de la fuerza de la instalación domótica



Aquí se puede apreciar la placa multiperforadora ,donde se encuentra soldado el cable que viene del arduino previamente pasando por el transistor y un cable hembra para la estimulación del relé, y en la parte de potencia de los relés se observan las fases de los circuitos.

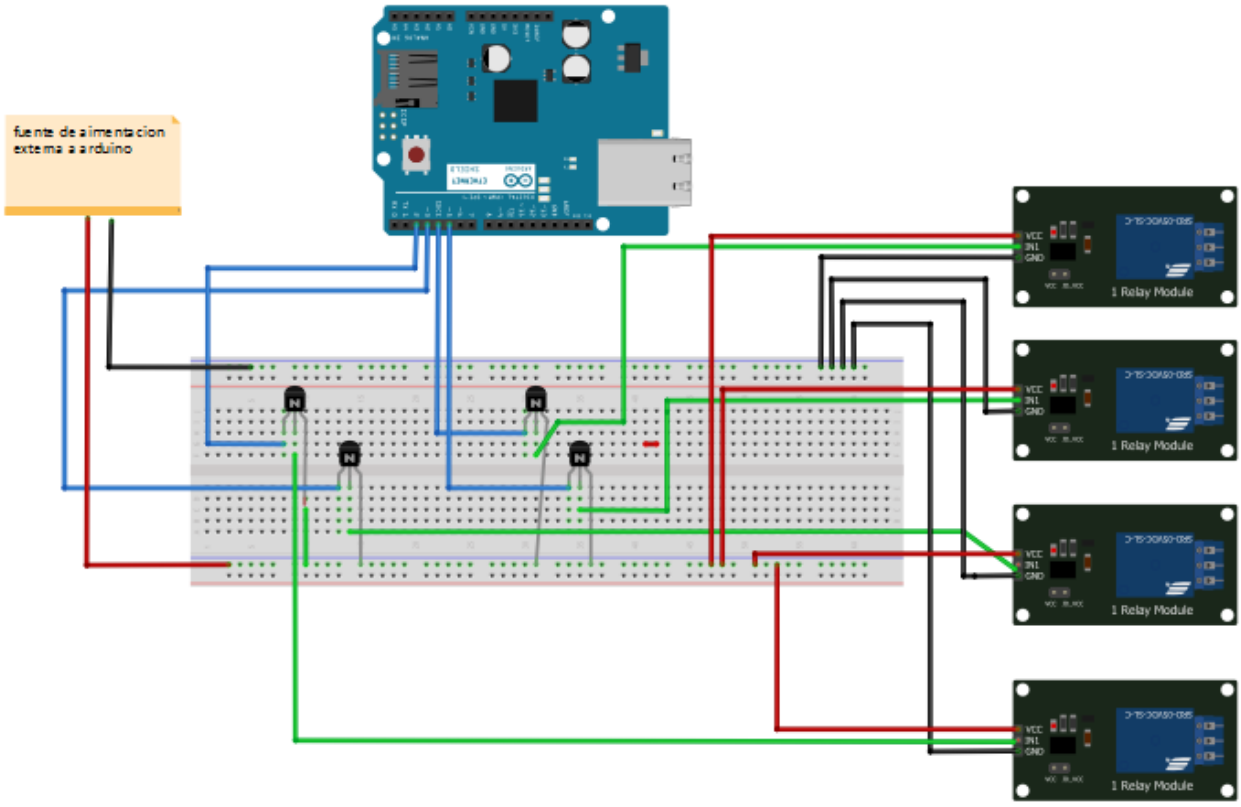
Los relés de la parte derecha , son los del accionamiento de la persiana, donde se tuvo que hacer lo siguiente



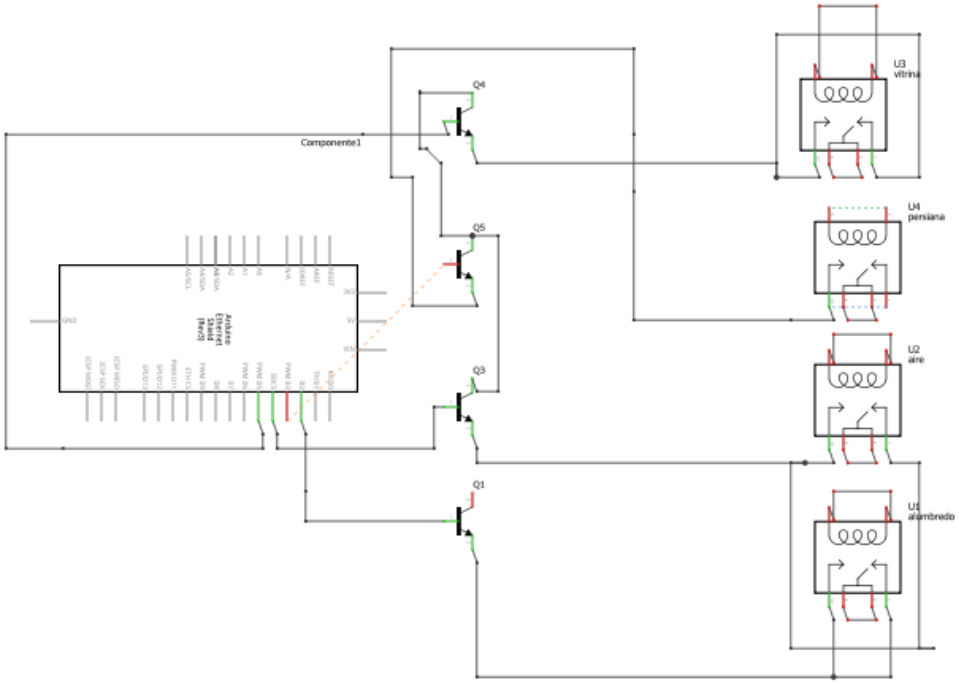
Para evitar tener problemas y quemar el motor.

4.2.2 Esquemas

Esquema de montaje



Esquema electrónico



4.2.4 Servidor web Arduino

La idea de hacer la domótica basándonos en una interface alojada en un servidor con arduino, vino de que se probaron otras instalaciones como bluetooth y a través de wiffi, pero el resultado no me convenció ya que la primera era limitado el alcance y era muy inestable además de que necesitabas una apps ya creada por un tercero, y la de wiffi te condicionaba a estar dentro d casa , por lo que nos decantamos por esta opción.

Nos basamos en un ejemplo encontrado en la red, el cual se estudió todo su código y modifico para obtener el siguiente resultado.

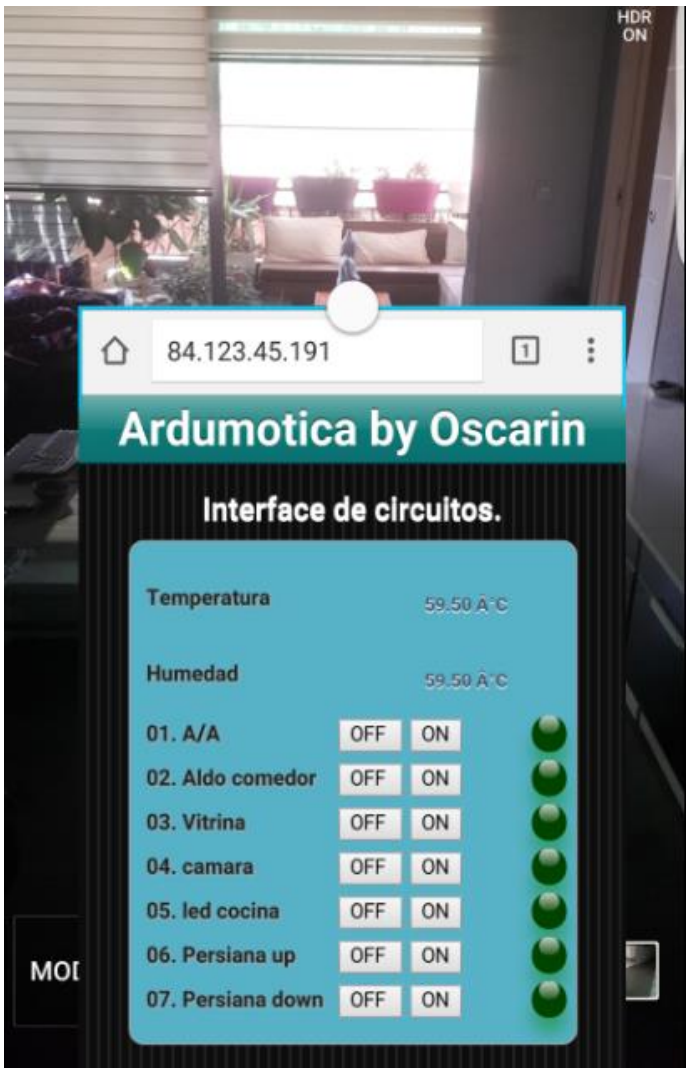
Podemos acceder desde wiffi o desde 3,4g, el codigo se encuentra dentro de nuestro arduino, en formato html, entonces cuando accedemos a nuestro servidor introduciendo

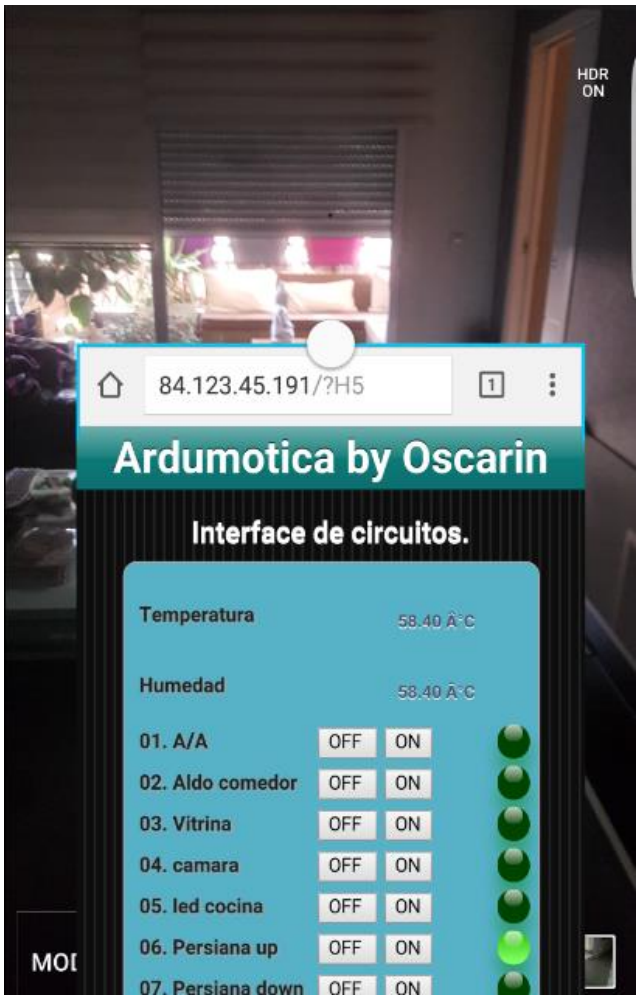
la ip de nuestra casa se nos habre esta ventana en el navegador, la cual es nuestra interface para poder controlar los circuitos de nuestra casa.

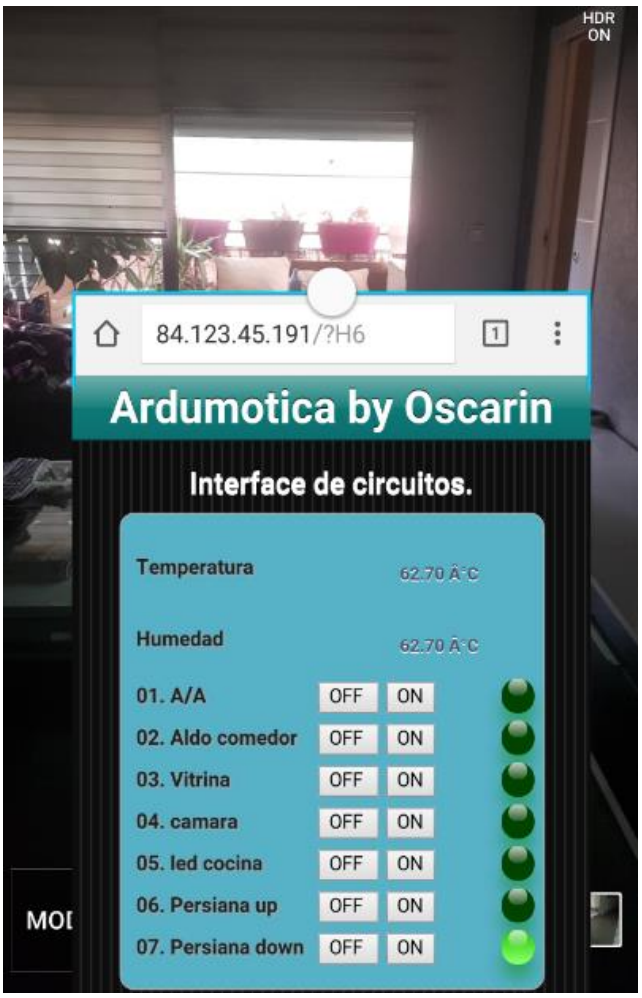
Cabe destacar que para poder acceder desde fuera de casa al servidor hay que habilitar el puerto 80 del router, entrando dentro de este y cambiando el peine y tambien indicarlo en el codigo de arduino.

4.2.5 Control de persianas

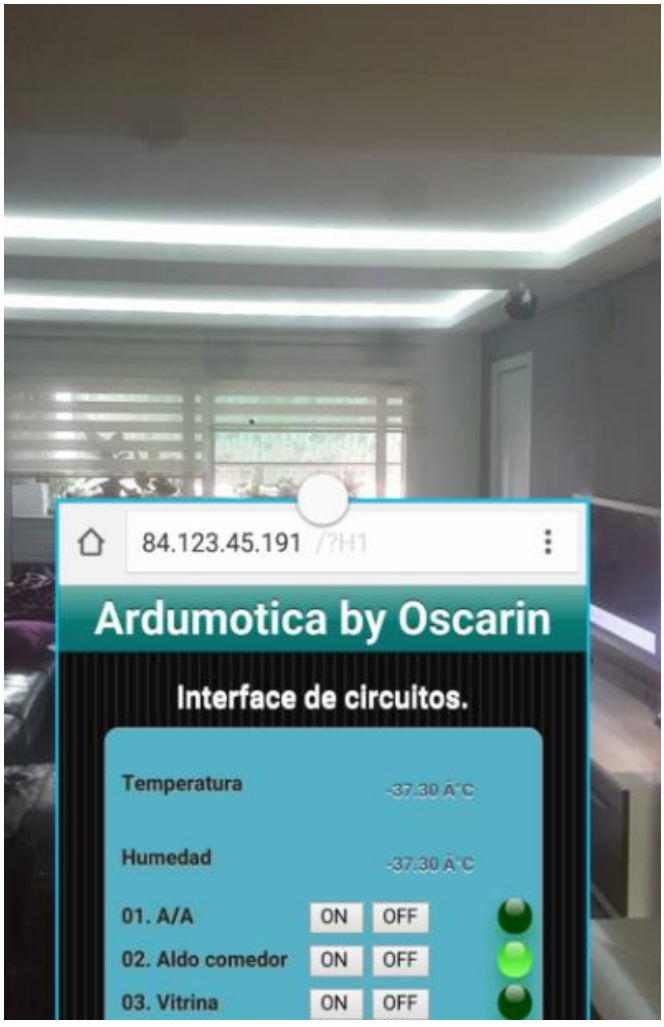
Cuando entramos dentro de nuestra interface , nos encontramos con los circuitos o dispositivos a controlar, pues bien si lo que queremos es subir o bajar la persiana , basta con presionar su circuito y desconectarlo para que la persiana se quede a la altura que deseamos.




HDR
ON
84.123.45.191/?H5
Ardumotica by Oscarin
Interface de circuitos.
Temperatura
58.40 Â°C
Humedad
58.40 Â°C
01. A/A OFF ON
02. Aldo comedor OFF ON
03. Vitrina OFF ON
04. camara OFF ON
05. led cocina OFF ON
06. Persiana up OFF ON
07. Persiana down OFF ON


HDR
ON
84.123.45.191/?H6
Ardumotica by Oscarin
Interface de circuitos.
Temperatura
62.70 Â°C
Humedad
62.70 Â°C
01. A/A OFF ON
02. Aldo comedor OFF ON
03. Vitrina OFF ON
04. camara OFF ON
05. led cocina OFF ON
06. Persiana up OFF ON
07. Persiana down OFF ON

4.2.6 Control iluminación



En estas capturas se puede apreciar que cuando estimulamos el circuito de alumbrado del comedor este se nos conecta y lo mismo pasa con el de la luz de la vitrina.

y lo mismo pasa con el de la luz de la vitrina.

4.2.7 Control aire acondicionado

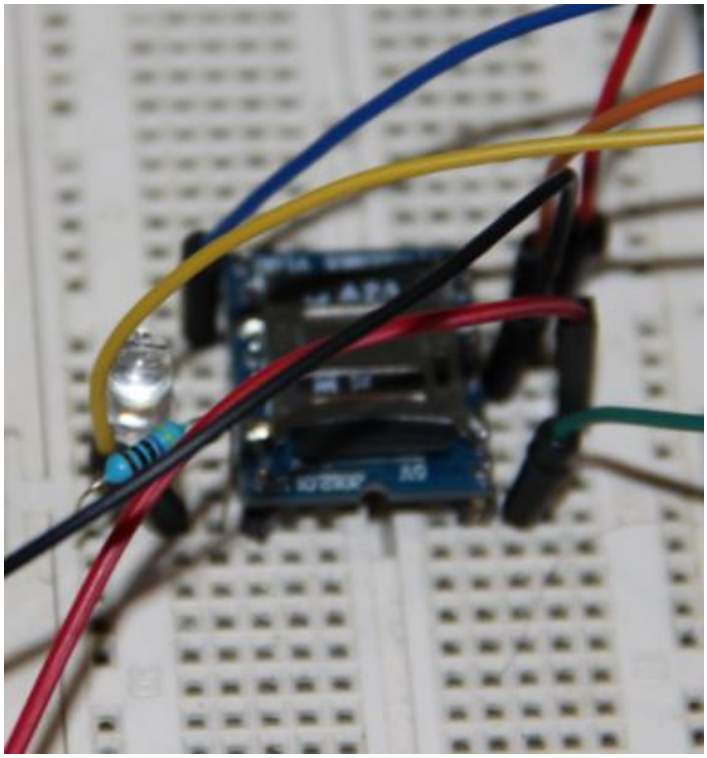
Este circuito funciona igual que todos los demás, pero varia un poco ya que el aire acondicionado solo lo podemos gobernar para encenderlo y apagarlo no para cambiar de modo , velocidad etc.., esto se debe a que solo actuamos sobre la tensión que lo alimenta , pero al ser un aparato de salvador escoda tiene la peculiaridad de que si está conectado por ejemplo en modo frio y a 24 grados y se le quita la tensión , cuando esta vuelve el aparato se conecta tal cual estaba funcionando con anterioridad.

4.3 Control de acceso mediante RFID y estados por voz

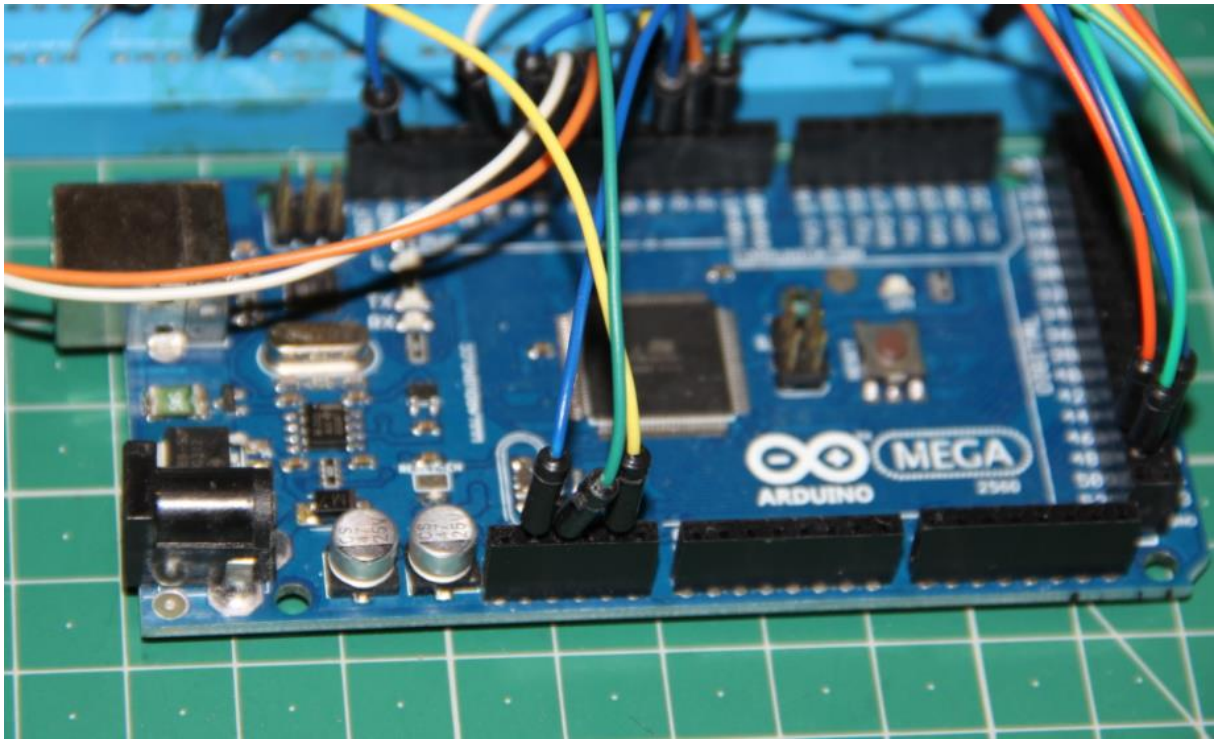
4.3.1 Arquitectura del sistema y descripción del mismo

La arquitectura de esta parte del proyecto al igual que las otras también es centralizada, dado que este montaje se va a llevar el día de la exposición no hay capturas del montaje real en la instalación ya que se conserva en la board pero si se puede indicar como va a ser el montaje y donde va a ir instalada cada parte del sistema.

El Arduino mega junto con el módulo de voz lo colocaremos en la caja de telecomunicaciones donde se encuentran los relés del servidor creado.



Este sería el módulo de voz

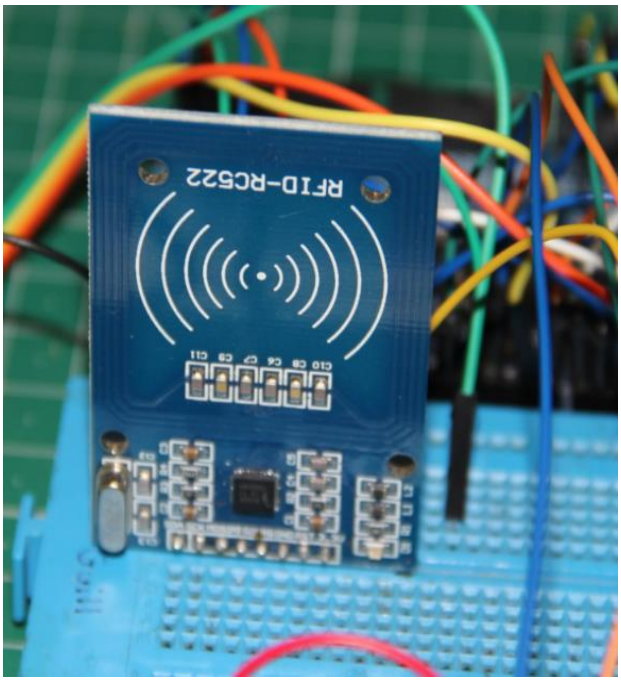


Y este el arduino.

Cabe destacar que todas las conexiones se harán con cable multifilar apantallado para evitar posibles problemas ya que el cuadro eléctrico de la vivienda eta justo encima de donde irán estos 2 dispositivos instalados.

Irían instalados dentro de la caja de telecomunicaciones como ya se ha mencionado.

Por otro lado tenemos el detector de RFID que se colocara en el lugar del interruptor de la entrada , el cual se va a desplazar a otro lugar partiendo el mecanismo .



En lugar de la tecla del interruptor se colocara una tapa ciega y lo más pegada a ella el lector de RFID para no tener problemas cuando nos identifiquemos.

Solo quedaría conectar el Pir que iría en este lugar

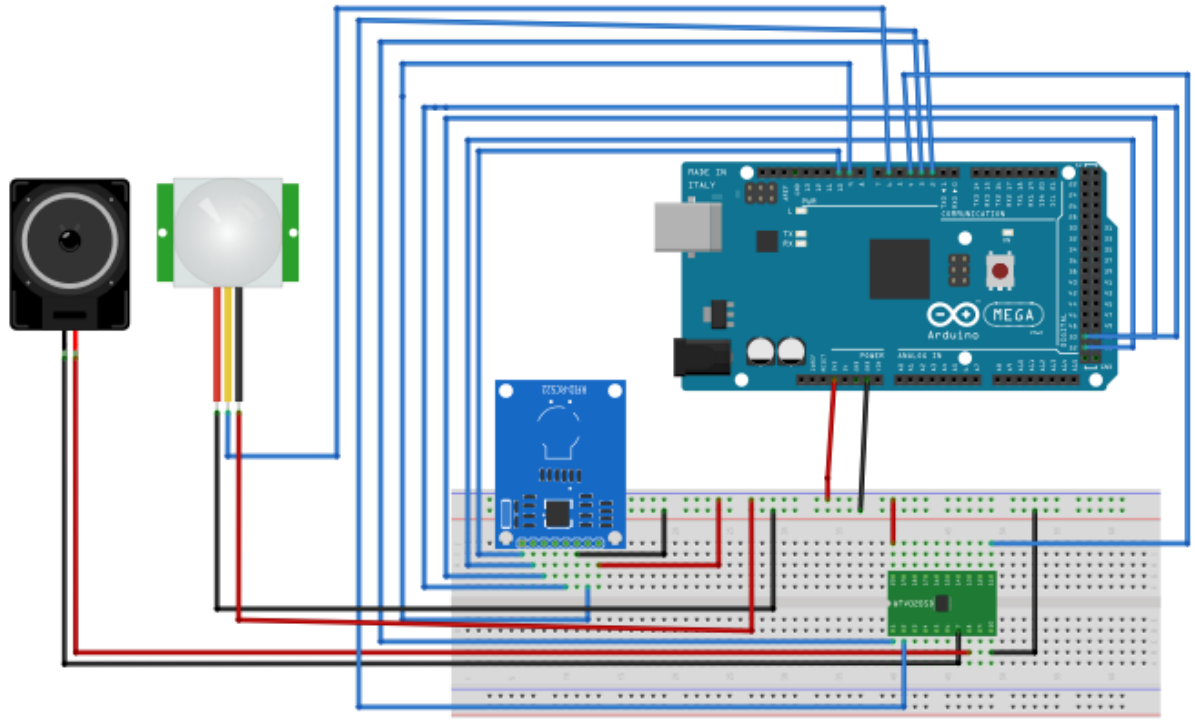
para solo hacer que actué cuando se abra la puerta y la alarma este armada.

Y por último nos quedaría solo la instalación del altavoz

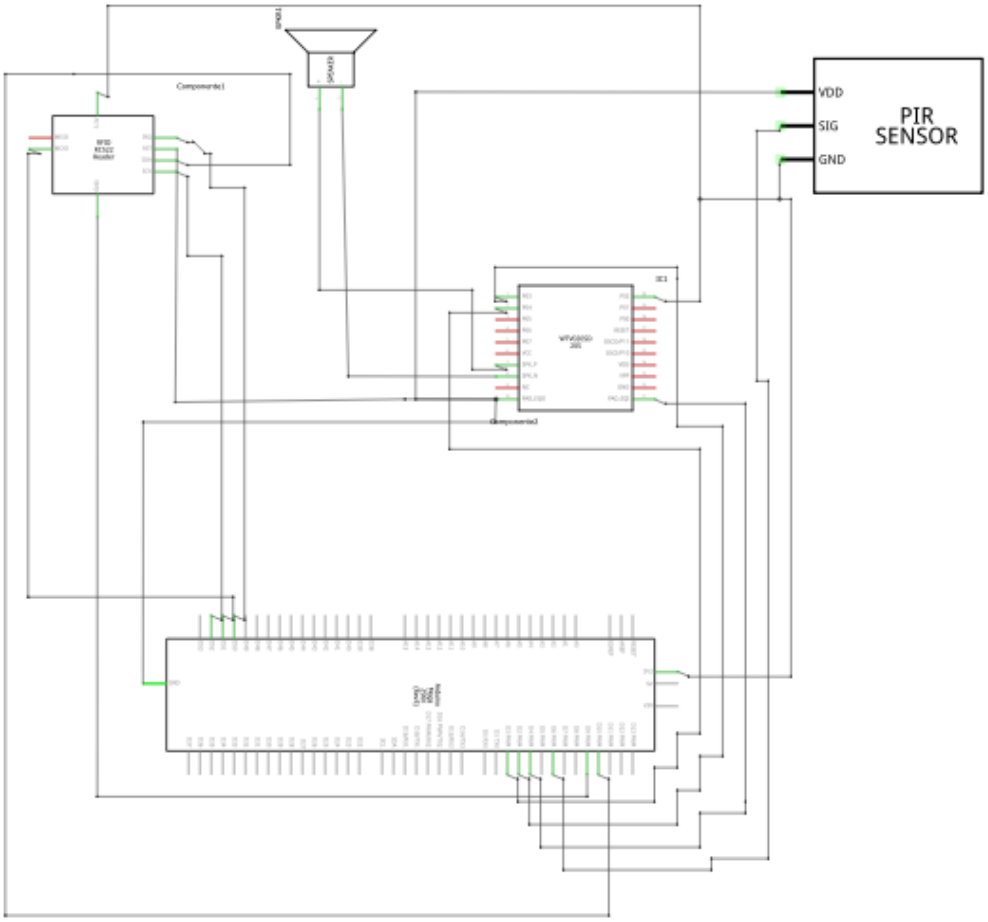
Que como se puede apreciar en la captura, es un altavoz empotrable para techo que iría ubicado en la entrada a la vivienda.

4.3.2 Esquemas

Esquema de montaje



Esquema electrónico



5 Planos

6 Anexos

6.1 Anexo codigo tanque

```
#include <Ultrasonic.h>
#include <LiquidCrystal.h>
#include <Wire.h>

Ultrasonic ultrasonic(6,10);// (Trig PIN,Echo PIN)
LiquidCrystal milcd(12,11,5,4,3,2);

int distancia,estadotanque,estadopulsador, pantalla=1;

unsigned long tiempo = 0;
unsigned long t_actualizado = 0;
unsigned long t_delay = 36000000;   //10 min


void setup()
{
  Serial.begin(9600);
  pinMode(13,INPUT); //sensor agua
  pinMode(8,INPUT); // pulsador vaciado
  pinMode(7,OUTPUT); //salida para la electrovalvula
  pinMode(9,OUTPUT); //salida bomba
```

```
 milcd.begin(16,2);     // decimos de cuantas columnas y filas es nuestra pantalla




}

void loop()

{




 distancia=(ultrasonic.Ranging(CM)); // leemos ultrasonidos



 estadotanque=digitalRead(13);  //leemos sensor tanque



estadopulsador=digitalRead(8); // leemos pulsador

delayMicroseconds(1000);
```

```
tiempo=millis() ; //aqui almacenamos el tiempo desde que se encendio el arduino


if(tiempo>t_actualizado + t_delay)


{

  t_actualizado=tiempo;              //en este algoritmo lo q hacemos es q cada 10 min
reseteamos la lcd para evitar los ruidos

  reset_lcd();

}




 if(distancia>=20&&estadotanque==0) // falta sal y tanque llenandose

 {


   estado1();


 }

  if(distancia<20&&estadotanque==0)//descalcificadora ok y tanque llenandose

 {


   estado2();


 }
```

```
if(distancia>=20&&estadotanque==1)  // falta sal y el tanque lleno
{

  estado3();
}



  if(distancia<20&&estadotanque==1)//descalcificadora ok y tanque lleno
{

  estado4();
}

// aqui con el pulsador para vaciar

 if(estadopulsador==HIGH)
 {
   pulsadoractivado();
 }



// final del loop
```

```
//comenzamos con las funciones


}
 void estado1()
 {
    digitalWrite(7,LOW);


    milcd.setCursor(0,0);            //colocamos el cursor en el prmer digito de arriba a la izquierda
    milcd.write("DESCALCIFICA:SAL");
    delay(1000);


    milcd.setCursor(0,1);
    milcd.write("TANQUE:LLENANDOSE");
    delay(1000);


 }


void estado2()
{
    digitalWrite(7,LOW);


    milcd.setCursor(0,0);
    milcd.write("DESCALCIFICA: OK");
    delay(1000);
```

```
    milcd.setCursor(0,1);
    milcd.write("TANQUE:LLENANDOSE");
    delay(1000);
}

void estado3 ()
{
   digitalWrite(7,HIGH); // tanque lleno desconectamos la electrovalvula


   milcd.setCursor(0,0);            //colocamos el cursor en el prmer digito de arriba a la
izquierda
   milcd.write("DESCALCIFICA:SAL");
   delay(1000);

   milcd.setCursor(0,1);
   milcd.write("TANQUE:    LLENO");
   Serial.println(" EL TANQUE ESTA LLENO" );
   Serial.println(" " );
   delay(1000);


}
```

```
void estado4()

{

   digitalWrite(7,HIGH); // tanque lleno desconectamos la electroválvula conectamos la
electrovalvula



   milcd.setCursor(0,0);

   milcd.write("DESCALCIFICA: OK");

   delay(1000);


   milcd.setCursor(0,1);

   milcd.write("TANQUE:    LLENO");

   Serial.println(" EL TANQUE ESTA LLENO" );

   Serial.println(" " );

   delay(1000);




}


void pulsadoractivado()

{

   digitalWrite(9,HIGH);

   milcd.clear();

  // milcd.begin(16,2);

   milcd.setCursor(0,0);
```

```
    milcd.print("PULSADOR ACTIVO");
   //  milcd.begin(16,1);
    milcd.setCursor(0,1);
    milcd.print("VACIANDO TANQUE");
    Serial.println(" pulsador activo ,vaciamos deposito durate 20 segundos" );
    Serial.println(" " );
     delay(110000);
     digitalWrite(9,LOW);
}

void reset_lcd()
{
  milcd.begin(16,2);

}
```

6.2 Anexo codigo servidor web

```
#include <Ethernet.h>

#include <SPI.h>

#include <EEPROM.h>

//////////////////////////////////////////////////////////////////////////

//CONFIGURACION

//////////////////////////////////////////////////////////////////////////

//IP CONFIGURACION MANUAL

byte ip[] = {192, 168, 1, 200 };

byte gateway[] = {192, 168, 1, 1 };

byte subnet[] = {255, 255, 255, 0 };

byte mac[] = {0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
```

//PUERTO ETHERNET

EthernetServer server = EthernetServer(80); // aqui decimos que el acceso a nuestro servidor arduino se hace a traves de nuestro ruter por el puerto 80

//numero de salidas

int outputQuantity = 7; // nº de circuitos

//invertimos la salidas

boolean outputInverted = false; //true or false

// esto se hace en el caso de que la placa de reles dispare el rele en negativo enlugar de positivo o sea dependiendo de si se conecta en N.A o N.C

//Html el refresco de la pagina

int refreshPage = 15; // el defecto son 10 seg.

// refrescamos por que si intentamos acceder demasiado rapido , puede que la pagina se haga inaccesible

//asociamos los botones con los pines del arduino y les damos el valor de false al iniciar el programa para que esten apagados

```
int switchOnAllPinsButton = false; //true or false

int outputAddress[] = { 22,23,24,25,26,27,28}; //pines de salida

String buttonText[7] = {

"01. A/A","02. Aldo comedor","03. Vitrina","04. camara","05. led cocina","06. Persiana up","07. Persiana down"};

// Ahora lo que hacemos es retener el ultimo estado de las salidas

int retainOutputStatus[7] = {0,0,0,0,0,0,0};

////////////////////////////////////////////////////////////////////////

//DECLARACION DE VARIABLES

////////////////////////////////////////////////////////////////////////

int outp = 0;

boolean printLastCommandOnce = false;

boolean printButtonMenuOnce = false;
```

```
boolean initialPrint = true;

String allOn = "";

String allOff = "";

boolean reading = false;

boolean outputStatus[16]; // Creamos una matriz boleana

String rev = "";

unsigned long timeConnectedAt;

boolean writeToEeprom = false;

////////////////////////////////////////////////

// lectura de la temperatura

const int tempInPin = A1;

int tempInValue = 0; //lectura de la temperatura

int tempScaleOutValue = 0; // escalado
```

```
int tempOutValue = 0; // la temperatura que se manara al cliente

float tempOutDeg = 0.0; // esta variable tipo float es para mas precision dando variables

////////////////////////////////////////////////////////////////////////

//RUNEA SOLO UNA VEZ CUANDO ARRANCAMOS PARA INICIALIZAR ,VELOCIDAD EN
BAUDIOS ETC ...

////////////////////////////////////////////////////////////////////////


void setup(){

Serial.begin(9600);

initEepromValues();

readEepromValues();

//inicializacion de pines de salida

boolean currentState = false;
```

```
int var;

for (int i = 0; i < outputQuantity; i++){

pinMode(outputAddress[i], OUTPUT);

var = outputAddress[i];



if(outputInverted == true) {

//digitalWrite(outputAddress[var], HIGH);

if(outputStatus[i] == 0){currentState = true;}else{currentState = false;} //check
outputStatus if off, switch output accordingly

digitalWrite(var, currentState);

}

else{

//digitalWrite(outputAddress[var], LOW);

if(outputStatus[i] == 0){currentState = false;}else{currentState = true;}//check
outputStatus if off, switch output accordingly
```

```
digitalWrite(var, currentState);

}

}

//Setting up the IP address. Comment out the one you dont need.

//Ethernet.begin(mac); //for DHCP address. (Address will be printed to serial.)

Ethernet.begin(mac, ip, gateway, subnet); //for manual setup. (Address is the one
configured above.)

server.begin();

Serial.print("Server started at ");

Serial.println(Ethernet.localIP());

}

////////////////////////////////////////////////////////////////////

//BUCLE LOOP
```

//

void loop(){

//Lectura del sensor de temperatura

tempInValue = analogRead(tempInPin);

// Connecting a 10K3 Thermistor to the Arduino Input

// +5V ﷺ[10Kohms]——————[Thermistor]——ﷺ 0V

// To Arduino IP ﷺ|

tempScaleOutValue = map(tempInValue, 0, 1023, 1023, 0); // pasamos con map el valor del transistor en mv a un valor entre 0 y 1023

tempOutValue = map(tempScaleOutValue, 130, 870, -170, 730); //rango de valores de arduino comparando con la temperatura

tempOutValue = tempOutValue -45; //ajustamos

```
tempOutDeg = tempOutValue / 10.0;

checkForClient();

}

////////////////////////////////////////////////////////////////////////

//checkForClient Funcion

////////////////////////////////////////////////////////////////////////

//

void checkForClient(){

EthernetClient client = server.available();

if (client) {

boolean currentLineIsBlank = true;

boolean sentHeader = false;

int temp,temp1;
```

```
while (client.connected()) {

if (client.available()) {

char c = client.read();

if(c == '*'){

printHtmlHeader(client); //esta es la respuesta al cliente de la cabecera de la pagina html

printLoginTitle(client);

printHtmlFooter(client);

break;

}

if(!sentHeader){

printHtmlHeader(client);

printHtmlButtonTitle(client); //imprimimos el titulo del boton
```

```
sentHeader = true;

}

if(reading && c == ' '){

reading = false;

}

if(c == '?') {

reading = true;

}

if(reading){

// si la entrada del cliente es higth  salimos como encendido 1

if(c == 'H') {

outp = 0;

}
```

```
if(c == 'L') {

outp = 1;

}

// Serial.println(c); //print the value of c to serial communication

//--------------------------------------------------------------------------------

// ? H 1 0

// ^ ^ ^ ^

// | | | |_____________read 4 ( 10,11,12,13....)

// | | |_______________read 3 ( 1....9)

// | |_________________read 2 if user input is H set output to L

// |___________________read 1

//--------------------------------------------------------------------------------
```

```
if( c == '1'){

char c = client.read();

switch (c) {

case '0':

triggerPin(outputAddress[10], client, outp);

break;

case '1':

triggerPin(outputAddress[11], client, outp);

break;

case '2':

triggerPin(outputAddress[12], client, outp);

break;

case '3':
```

```
triggerPin(outputAddress[13], client, outp);

break;

case '4':

triggerPin(outputAddress[14], client, outp);

break;

case '5':

triggerPin(outputAddress[15], client, outp);

break;

default:

char c = client.read();

triggerPin(outputAddress[1], client, outp);

}
```

```
}

else {

switch (c) {

case '0':

triggerPin(outputAddress[0], client, outp);

break;

// case '1':

// triggerPin(outputAddress[1], client, outp);

// break;

case '2':

triggerPin(outputAddress[2], client, outp);

break;

case '3':
```

```
//add code here to trigger on 3

triggerPin(outputAddress[3], client, outp);

break;

case '4':

//add code here to trigger on 4

triggerPin(outputAddress[4], client, outp);

break;

case '5':

//add code here to trigger on 5

triggerPin(outputAddress[5], client, outp);

//printHtml(client);

break;
```

```
case '6':

//add code here to trigger on 6

triggerPin(outputAddress[6], client, outp);

break;

case '7':

//add code here to trigger on 7

triggerPin(outputAddress[7], client, outp);

break;

case '8':

//add code here to trigger on 8

triggerPin(outputAddress[8], client, outp);

break;

case '9':
```

```
//add code here to trigger on 9

triggerPin(outputAddress[9], client, outp);

break;

} //end of switch case

}

}//end of switch switch the relevant output

//if user input was blank

if (c == '\n' && currentLineIsBlank){

printLastCommandOnce = true;

printButtonMenuOnce = true;

triggerPin(777, client, outp); //Call to read input and print menu. 777 is used not to update any outputs

break;
```

```
}

}

}

printHtmlFooter(client); //Prints the html footer

}

else{

if (millis() > (timeConnectedAt + 60000)){

if (writeToEeprom == true){

writeEepromValues(); //write to EEprom the current output statuses

Serial.println("No Clients for more then a minute - Writing statuses to Eeprom.");

writeToEeprom = false;

}
```

```
}

}

}// END

////////////////////////////////////////////////////////////////////////

//triggerPin Function

////////////////////////////////////////////////////////////////////////

//

void triggerPin(int pin, EthernetClient client, int outp){

if (pin != 777){

// Serial.println(pin);

if(outp == 1) {

if (outputInverted ==false){

digitalWrite(pin, HIGH);
```

```
}

else{

digitalWrite(pin, LOW);

}

}

if(outp == 0){

if (outputInverted ==false){

digitalWrite(pin, LOW);

}

else{

digitalWrite(pin, HIGH);

}
```

```
}

}

//Refresh the reading of outputs

readOutputStatuses();

//Prints the buttons

if (printButtonMenuOnce == true){

printHtmlButtons(client);

printButtonMenuOnce = false;

}

}

////////////////////////////////////////////////////////////////////////

//FUNCION PARA IMPRIMIR LOS BOTONES EN HTML

////////////////////////////////////////////////////////////////////////
```

```
void printHtmlButtons(EthernetClient client){

//Start to create the html table

client.println("");

//client.println("<p>");

client.println("<FORM>");

client.println("<table border=\"0\" align=\"center\">");

//IMPRIMIMOS LA TEMPERATURA

client.print("<tr>\n");

client.print("<td><h4>");

client.print("Temperatura");

client.print("</h4></td>\n");
```

```
client.print("<td></td>");

client.print("<td>");

client.print("<h3>");

client.print(tempOutDeg);       // es es el valor analogico de nuestra temperatura que se vera en la pagina

client.print(" °C</h3></td>\n");

client.print("<td></td>");

client.print("</tr>");

//AQUI IMPRIMIMOS LA HUMEDAD

client.print("<tr>\n");

client.print("<td><h4>");

client.print("Humedad");

client.print("</h4></td>\n");
```

```
client.print("<td></td>");

client.print("<td>");

client.print("<h3>");

client.print(tempOutDeg);       // esto en su dia se cambiara por la variable para leer la humedad

client.print(" °C</h3></td>\n");

client.print("<td></td>");

client.print("</tr>");


//iniciamos la impresion de boton por boton

for (int var = 0; var < outputQuantity; var++) {

//set command for all on/off

allOn += "L";

allOn += outputAddress[var];
```

```
allOff += "H";

allOff += outputAddress[var];

//Print begining of row

client.print("<tr>\n");

//Prints the button Text

client.print("<td><h4>");

client.print(buttonText[var]);

client.print("</h4></td>\n");

//Prints the ON Buttons++++++++++++++++++++++++++++++++++++++++++++++++++

client.print("<td>");

client.print("<INPUT TYPE=\"button\" VALUE=\"OFF ");

client.print("\" onClick=\"parent.location='/?L");
```

```
client.print(var);

client.print("'\"></td>\n");

//Prints the OFF Buttons ------------------------------------------

client.print(" <td><INPUT TYPE=\"button\" VALUE=\"ON");

client.print("\" onClick=\"parent.location='/?H");

client.print(var);

client.print("'\"></td>\n");

//Invert the LED display if output is inverted.

if (outputStatus[var] == false ){ //If Output is ON

if (outputInverted == true){ //and if output is not inverted

client.print(" <td><div class='black-circle'><div class='glare'></div></div></td>\n"); //Print
html for OFF LED

}
```

```
else{ //else output is inverted then


client.print(" <td><div class='green-circle'><div class='glare'></div></div></td>\n");
//Print html for ON LED




}


}


else //If Output is Off


{


if (outputInverted == false){ //and if output is not inverted


client.print(" <td><div class='black-circle'><div class='glare'></div></div></td>\n"); //Print
html for OFF LED


}


else{ //else output is inverted then


client.print(" <td><div class='green-circle'><div class='glare'></div></div></td>\n");
//Print html for ON LED
```

```
}

}

//Print end of row

client.print("</tr>\n");

}

//Display or hide the Print all on Pins Button

if (switchOnAllPinsButton == true ){

//Prints the ON All Pins Button

client.print("<tr>\n<td><INPUT TYPE=\"button\" VALUE=\"Switch ON All Pins");

client.print("\" onClick=\"parent.location='/?");

client.print(allOn);

client.print("'\"></td>\n");
```

```
//Prints the OFF All Pins Button

client.print("<td><INPUT TYPE=\"button\" VALUE=\"Switch OFF All Pins");

client.print("\" onClick=\"parent.location='/?");

client.print(allOff);

client.print("'\"></td>\n<td></td>\n<td></td>\n</tr>\n");

}

//Closing the table and form

client.println("</table>");

client.println("</FORM>");

//client.println("</p>");

}

////////////////////////////////////////////////////////////////////////

//readOutputStatuses Function
```

```
////////////////////////////////////////////////////////////////////////

//Reading the Output Statuses

void readOutputStatuses(){

for (int var = 0; var < outputQuantity; var++) {

outputStatus[var] = digitalRead(outputAddress[var]);

//Serial.print(outputStatus[var]);

}

}

////////////////////////////////////////////////////////////////////////

//FUNCION DE LA LECTURA DE LOS VALORES DE LA EPROM

////////////////////////////////////////////////////////////////////////

//Read EEprom values and save to outputStatus
```

```
void readEepromValues(){

for (int adr = 0; adr < outputQuantity; adr++) {

outputStatus[adr] = EEPROM.read(adr);

}

}

////////////////////////////////////////////////////////////////////////

//FUNCION DE ESCRITURA DE LOS VALORES EN EPROM

////////////////////////////////////////////////////////////////////////

//Write EEprom values

void writeEepromValues(){

for (int adr = 0; adr < outputQuantity; adr++) {

EEPROM.write(adr, outputStatus[adr]);

}
```

```
}

////////////////////////////////////////////////////////////////////////

//INICIALIZAR LOS VALORES DE LA EPROM

////////////////////////////////////////////////////////////////////////

//Initialiaze EEprom values

//if eeprom values are not the correct format ie not euqual to 0 or 1 (thus greater then 1) initialize by putting 0

void initEepromValues(){

for (int adr = 0; adr < outputQuantity; adr++){

if (EEPROM.read(adr) > 1){

EEPROM.write(adr, 0);

}

}
```

```
}

////////////////////////////////////////////////////////////////////////

//FUNCION DE LA CABECERA DE LA PAGINA HTML

////////////////////////////////////////////////////////////////////////

//Prints html header

void printHtmlHeader(EthernetClient client){

// Serial.print("Serving html Headers at ms -");

timeConnectedAt = millis(); //Record the time when last page was served.

// Serial.print(timeConnectedAt); // Print time for debbugging purposes

writeToEeprom = true; // page loaded so set to action the write to eeprom

// send a standard http response header

client.println("HTTP/1.1 200 OK");
```

```
client.println("Content-Type: text/html");

client.println("Connnection: close");

client.println();

client.println("<!DOCTYPE HTML>");

client.println("<head>");

// TITULO DE LA PAGINA

client.println("<title>ARDUMOTICA BY OSCARIN</title>");

client.println("<meta name=\"description\" content=\"ARDUMOTICA BY OSCARIN\"/>");

// add a meta refresh tag, so the browser pulls again every x seconds:

client.print("<meta http-equiv=\"refresh\" content=\"");

client.print(refreshPage);

client.println("; url=/\">");

// add other browser configuration
```

```
client.println("<meta name=\"apple-mobile-web-app-capable\" content=\"yes\">");

client.println("<meta name=\"apple-mobile-web-app-status-bar-style\"
content=\"default\">");

client.println("<meta name=\"viewport\" content=\"width=device-width, user-
scalable=no\">");

//inserting the styles data, usually found in CSS files.

client.println("<style type=\"text/css\">");

client.println("");

//lo siguiente es para ver la pagina de forma grafia

client.println("html { height:100%; }");

client.println(" body {");

client.println(" height: 100%;");

client.println(" margin: 0;");

client.println(" font-family: helvetica, sans-serif;");
```

```
client.println(" -webkit-text-size-adjust: none;");

client.println(" }");

client.println("");

client.println("body {");

client.println(" -webkit-background-size: 50% 21px;");

client.println(" background-color: #0E0D0E;");    // COLOR DEL FONDO DE h2

client.println(" background-image:");

client.println(" -webkit-gradient(linear, left top, right top,");

client.println(" color-stop(.75, transparent),");

client.println(" color-stop(.75, rgba(255,255,255,.1)) );");

client.println(" -webkit-background-size: 7px;");

client.println(" }");
```

```
client.println("");

client.println(".view {");

client.println(" min-height: 100%;");

client.println(" overflow: auto;");

client.println(" }");

client.println("");

client.println(".header-wrapper {");

client.println(" height: 44px;");

client.println(" font-weight: bold;");

client.println(" text-shadow: rgba(0,0,0,0.7) 0 -1px 0;");

client.println(" border-top: solid 1px rgba(255,255,255,0.6);");

client.println(" border-bottom: solid 1px rgba(0,0,0,0.6);");

client.println(" color: #fff;");
```

```
client.println(" background-color: #0E7B77;");      //EN ESTA LINEA LE CAMBIAMOS EL COLOR A LA CABECERA PRINCIPAL

client.println(" background-image:");

client.println(" -webkit-gradient(linear, left top, left bottom,");

client.println(" from(rgba(255,255,255,.4)),");

client.println(" to(rgba(255,255,255,.05)) ),");

client.println(" -webkit-gradient(linear, left top, left bottom,");

client.println(" from(transparent),");

client.println(" to(rgba(0,0,64,.1)) );");

client.println(" background-repeat: no-repeat;");

client.println(" background-position: top left, bottom left;");

client.println(" -webkit-background-size: 100% 21px, 100% 22px;");

client.println(" -webkit-box-sizing: border-box;");
```

```
client.println(" }");

client.println("");

client.println(".header-wrapper h1 {"); // cabecera 1

client.println(" text-align: center;");

client.println(" font-size: 30px;");

client.println(" line-height: 44px;");

client.println(" margin: 0;");

client.println(" }");

client.println("");

client.println(".group-wrapper {");

client.println(" margin: 9px;");

client.println(" }");
```

```
client.println("");

client.println(".group-wrapper h2 {");                    // cabecera 2

client.println(" text-align: center;");

client.println(" color: #fff;");                          // color blanco

client.println(" font-size: 20px;");                      //tamaño de la fuente

client.println(" line-height: 0.8;");

client.println(" font-weight: bold;");

client.println(" text-shadow: #fff 0 1px 0;");

client.println(" margin: 20px 10px 12px;");

client.println(" }");

client.println("");

client.println(".group-wrapper h3 {");                    //Cabecera 3

client.println(" color: #4c566c;");                       //verde oscuro de los leds apagados
```

```
client.println(" font-size: 12px;");

client.println(" line-height: 1;");

client.println(" font-weight: bold;");

client.println(" text-shadow: #fff 0 1px 0;");

client.println(" margin: 20px 10px 12px;");

client.println(" }");

client.println("");

client.println(".group-wrapper h4 {");          //h4

client.println(" color: #212121;");       //colo de las fuentes de mi interface

client.println(" font-size: 14px;");

client.println(" line-height: 1;");

client.println(" font-weight: bold;");
```

```
client.println(" text-shadow: #aaa 1px 1px 3px;");

client.println(" margin: 5px 5px 5px;");

client.println(" }");

client.println("");

client.println(".group-wrapper table {");

client.println(" background-color: #57B1C7;");

client.println(" -webkit-border-radius: 10px;");

client.println(" -moz-border-radius: 10px;");

client.println(" -khtml-border-radius: 10px;");

client.println(" border-radius: 10px;");

client.println(" font-size: 17px;");

client.println(" line-height: 20px;");

client.println(" margin: 9px 0 20px;");
```

```
client.println(" border: solid 1px #a9abae;");

client.println(" padding: 11px 3px 12px 3px;");

client.println(" margin-left:auto;");

client.println(" margin-right:auto;");

client.println(" -moz-transform :scale(1);"); //Code for Mozilla Firefox

client.println(" -moz-transform-origin: 0 0;");

client.println(" }");

client.println("");

//Lo siguiente es para ver el led en verde claro cuando este en on el boton

client.println(".green-circle {");   // aqui definimos el led como un circulo verde

client.println(" display: block;");

client.println(" height: 23px;");
```

```
client.println(" width: 23px;");

client.println(" background-color: #0f0;");

//en estas lineas lo que hacemos es definir el led encendido con el color rgb como verde claro

client.println(" -moz-border-radius: 11px;");

client.println(" -webkit-border-radius: 11px;");

client.println(" -khtml-border-radius: 11px;");

client.println(" border-radius: 11px;");

client.println(" margin-left: 1px;");

client.println(" background-image: -webkit-gradient(linear, 0% 0%, 0% 90%, from(rgba(46, 184, 0, 0.8)), to(rgba(148, 255, 112, .9)));@");

client.println(" border: 2px solid #ccc;");

client.println(" -webkit-box-shadow: rgba(11, 140, 27, 0.5) 0px 10px 16px;");

client.println(" -moz-box-shadow: rgba(11, 140, 27, 0.5) 0px 10px 16px; /* FF 3.5+ */");
```

```
client.println(" box-shadow: rgba(11, 140, 27, 0.5) 0px 10px 16px; /* FF 3.5+ */");

client.println(" }");

client.println("");

//LAS SIGUIENTES LINEAS NOS HACEN QUE EL LED SE VUELVA VERDE OSCURO CUANDO ESTA APAGADO

client.println(".black-circle {");

client.println(" display: block;");

client.println(" height: 23px;");

client.println(" width: 23px;");

client.println(" background-color: #040;");

client.println(" -moz-border-radius: 11px;");

client.println(" -webkit-border-radius: 11px;");

client.println(" -khtml-border-radius: 11px;");
```

```
client.println(" border-radius: 11px;");

client.println(" margin-left: 1px;");

client.println(" -webkit-box-shadow: rgba(11, 140, 27, 0.5) 0px 10px 16px;");

client.println(" -moz-box-shadow: rgba(11, 140, 27, 0.5) 0px 10px 16px; /* FF 3.5+ */");

client.println(" box-shadow: rgba(11, 140, 27, 0.5) 0px 10px 16px; /* FF 3.5+ */");

client.println(" }");

client.println("");

//LAS SIGUIENTES LINEAS AÑADEN EL RESPLANDOR A LOS LEDS

client.println(" .glare {");

client.println(" position: relative;");

client.println(" top: 1;");

client.println(" left: 5px;");

client.println(" -webkit-border-radius: 10px;");
```

```
client.println(" -moz-border-radius: 10px;");

client.println(" -khtml-border-radius: 10px;");

client.println(" border-radius: 10px;");

client.println(" height: 1px;");

client.println(" width: 13px;");

client.println(" padding: 5px 0;");

client.println(" background-color: rgba(200, 200, 200, 0.25);");

client.println(" background-image: -webkit-gradient(linear, 0% 0%, 0% 95%,
from(rgba(255, 255, 255, 0.7)), to(rgba(255, 255, 255, 0)));");

client.println(" }");

client.println("");

// estos son los datos del estilo de la cabecera

client.println("</style>");
```

```
client.println("</head>");

//now printing the page itself

client.println("<body>");

client.println("<div class=\"view\">");

client.println(" <div class=\"header-wrapper\">");

client.println(" <h1>Ardumotica by Oscarin</h1>");

client.println(" </div>");

//////

} //end of htmlHeader

////////////////////////////////////////////////////////////////////////////

//htmlFooter Function

////////////////////////////////////////////////////////////////////////////
```

```
//Prints html footer

void printHtmlFooter(EthernetClient client){

//Set Variables Before Exiting

printLastCommandOnce = false;

printButtonMenuOnce = false;

allOn = "";

allOff = "";




client.println(rev);

client.println("</h3></div>\n</div>\n</body>\n</html>");

delay(1); // give the web browser time to receive the data

client.stop(); // close the connection:

Serial.println(" - Done, Closing Connection.");
```

```
delay (2); //delay so that it will give time for client buffer to clear and does not repeat multiple pages.

} //end of htmlFooter

//////////////////////////////////////////////////////////////////////

//printHtmlButtonTitle Function

//////////////////////////////////////////////////////////////////////

// esta es la segunda cabecera

void printHtmlButtonTitle(EthernetClient client){

client.println("<div class=\"group-wrapper\">");

client.println(" <h2>Interface de circuitos.</h2>");

client.println();

}

//////////////////////////////////////////////////////////////////////
```

```
//printLoginTitle Function

////////////////////////////////////////////////////////////////////////////

//Prints html button title

void printLoginTitle(EthernetClient client){

//client.println("<div class=\"group-wrapper\">");

client.println(" <h2>Please enter the user data to login.</h2>");

client.println();

}
```

6.3 Anexo codigo control de acceso

```
#include <SPI.h>

#include <RFID.h>

#include <Wtv020sd16p.h>




RFID rfid(10,9); // Pines de conexión del RFID


int codigo,EstadoAlarma=0,i,a,b,c,contador;  //estado de alarma es el pulsadory en su dia sera el sensor de la puerta


int sensorpir = 6;

int ledsensor = 3;

//pines modulo de voz

int resetPin = 2;

int clockPin = 3;

int dataPin = 4;

int busyPin = 5;


Wtv020sd16p wtv020sd16p(resetPin,clockPin,dataPin,busyPin);

String orden;
```

```
void setup()
{
  Serial.begin(9600);

  wtv020sd16p.reset();
  SPI.begin();
  rfid.init();

  pinMode(sensorpir,INPUT);//sensor pir

  pinMode(ledsensor,OUTPUT);//led de pruebas sensor pir

  pinMode(7,OUTPUT);     //led ROJO alarma ACTIVADA

  pinMode(8,OUTPUT);     //led amarillo alarma desactivada




}

void loop()
{
```

```
// wtv020sd16p.reset();

 // el siguiente if es el que nos lee la tarjeta

  if (rfid.isCard())
  {

      // El siguiente IF muestra en la Pantalla Serial el nº de serie de la tarjeta "escaneada"

      if (rfid.readCardSerial()) {


            codigo = rfid.serNum[4],DEC;    //es el ultimo bloque de los 5 bloques de
numeros del tag que se sacan del prorama rfid original


            if(codigo==234)
            {
              Serial.println(" ");
              Serial.println(" ");
              Serial.println(" Hola Oscarin,,activando la Alarma en 10seg ");
              Serial.println(" ");
              Serial.println(" ");
```

```
                pista1();


                }


              // codigo==0;
                        delay(1000);


              }


      }
  rfid.halt();




 if(EstadoAlarma==0)
 {
  desactivada();
 }



 if  (((codigo==234&&EstadoAlarma==0)&&(digitalRead(sensorpir)==LOW))  ||
((codigo==234&&EstadoAlarma==0)&&(digitalRead(sensorpir)==HIGH)))


  {
    activandose();
```

```
  }


if(EstadoAlarma==1)
{
  activada();
}



if  ((digitalRead(sensorpir)==HIGH)&& codigo==234 &&EstadoAlarma==1)

{
  desactivandose();
}



if((digitalRead(sensorpir)==HIGH)&&EstadoAlarma==1)
{
  pista2();
  bienvenido();

}
```

```
  if(digitalRead(sensorpir) ==HIGH)
  {
   sensor();
  }
  // aqui empiezan las pistas del voz
}

// a partir de aqui bienen los bloques de funciones

 void pista1()
 {
   wtv020sd16p.playVoice(1);
```

```
  delay(5000);
}

void pista2()
{
  wtv020sd16p.stopVoice();
  wtv020sd16p.playVoice(2);
}

void pista4()
{
  wtv020sd16p.stopVoice();
  wtv020sd16p.playVoice(4);
}




void sensor()
{



 digitalWrite(ledsensor,HIGH);
 delay(1000);
 digitalWrite(ledsensor,LOW);
}
```

```
void desactivada ()
 {
   EstadoAlarma=0;



   digitalWrite(7,LOW);
   digitalWrite(8,HIGH);

   delay(100);



 }

 void activandose()
 {

    wtv020sd16p.stopVoice();
    wtv020sd16p.playVoice(1);
   EstadoAlarma=1;
   codigo=0;

  for(i=0;i<10;i++)
```

```
  {

   delay(1000);            //tiene que activarse en 10 seg pq si no coincidia  alarma a1 pueta habierta y codigo

  }

}

void activada()
{

  codigo=0;
  digitalWrite(7,HIGH);
  digitalWrite(8,LOW);

  delay(1000);

}

void bienvenido()
{
```

```
    Serial.println(" ");
    Serial.println(" ");
    Serial.println("BIENVENIDO IDENTIFICATE TIENES 5 SEGUNDOS O SE ACTIVARA LA
ALARMA  ");

    Serial.println(" ");
    Serial.println(" ");
    delay(6000);



   if (rfid.isCard())
  {



        if (rfid.readCardSerial()) {


                codigo = rfid.serNum[4],DEC;


                if(codigo==234)
                {
```

```
                Serial.println(" ");
                Serial.println(" ");
                Serial.println(" WELCOME TO THE HOUSE OSCARIN...DESACTIVANDO
ALARMA ");
                wtv020sd16p.playVoice(3);
                digitalWrite(8,HIGH);
                digitalWrite(7,LOW);
                Serial.println(" ");
                Serial.println(" ");
                codigo=0;
                EstadoAlarma=0;
                delay(1000);

              }

            }

      }
    else
    {
    pista4();
    intruso ();
    }
```

```
    rfid.halt();



}

 void intruso()

{


   for(a=0;a<5;a++)

   {

     Serial.println("intruso ");

     Serial.println(" ");

     Serial.println(" ");

     delay(1000);

   }

  EstadoAlarma=0;   //pongo 0 para que no siga dando por saco,,pero en la realidad hay q aumentar el for

}


  void desactivandose()

 {


    EstadoAlarma=0;
```

```
   codigo=0;


   for(a=0;a<5;a++)
  {
    delay(1000);
  }
   EstadoAlarma=0;
}
```

7 Presupuesto

NUMERO	DESCRIPCIÓN	PRECIO UNI/E	CANTIDAD	TOTAL
1	cableado macho /hem	2,5	5	12,5
2	bobina cable	4,6	4	18,4
3	leds	25	0,3	7,5
4	resistencias pack	1	10	10
5	potenciometro	1	1,23	1,23
6	caja estanca	6,2	3	18,6
7	estaño	3,25	2	6,5
8	placa mutiperforada	0,25	10	2,5
9	board	9	2	18
10	pantalla lcd	8,5	1	8,5
11	modulo 1 rele	2,5	1	2,5
12	modulo 2 reles	3,2	2	6,4
13	modulo 4 reles	5,6	1	5,6
14	transformador 5v	12	1	12
15	fuente alimen 24v	25,6	1	25,6
16	fuente alimen 12v	18	2	36
17	sensor ultrasonidos	4,95	1	4,95
18	sensor humedad	3,7	1	3,7
19	pulsador	0,61	1	0,61
20	Arduino uno	14,95	1	14,95
21	Arduino nano	7,5	1	7,5
22	Arduino mega	17,9	1	17,9
23	shield ethernet	14	1	14
24	modulo expansion	6	1	6
25	transistor npn	1,2	4	4,8
26	integrado 7404	1,5	1	1,5
27	modulo RFID RC522	5,6	1	5,6
28	sensor movimiento PIR	3,2	1	3,2
29	modulo voz somo14D	25	1	25
30	altavoz	20	1	20
31	tanque agua	15	1	15
32	sensor rebose	12,35	1	12,35
33	armario telecomunica	65	1	65
				0
34	numero de horas	15	120	1800
				Total /e
				2213,89
				Total 21 % iva
				2678,8

Como se puede apreciar, el presupuesto total con el 21% de IVA es de 2678,80 euros, es un precio un poco elevado ya que aunque el materia es relativamente barato, se han utilizado una gran cantidad de horas para el desarrollo y el montaje de este proyecto, cabe añadir que si se tuviese que instalar en otra vivienda su precio bajaría considerablemente ya que todo el código necesario que es en la parte que más horas se han dedicado ya está creado.

8 Referencias

En este apartado , se quiere hacer referencia y agradecer a todas las webs, blogs, chats, libros etc.,,las cuales se han consultado y que gracias a ellas se ha podido concluir este proyecto.

-Libro Arduino curso práctico, por Oscar Torrente Artero

- **www.arduinodhtics.weebly.com**
- **www.domoactualidad.bogspot.com.es**
- **www.hetpro-store.com**
- **www.electronica-teoriaypractica.com**
- **www.Arduino.cl//Arduino.cc**
- **www.openwebinaris.net**
- **www.wikipwdia.org**
- **www.Raulcarretero.com**
- **www.KNX.org**
- **www.latiendadomotica.com**
- **www.electan.com**
- **www.Prometec.net**
- **www.elcajondeardu.blogspot.com.es**
- **www.profetoloka.com.ar**
- **www.Tallerpacticodearduino//LuisTorreñoPeromingo**
- **www.Arkiplus.com**