****

**Instituto Tecnológico de Estudios Superiores de Monterrey**

**Campus Tampico**

Problem Oriented Learning **Domótica**

**Ing. Juan Gabino Díaz**

**Jazmín Yalul Tinajero Pineda A00511574**

**José Alberto Valencia Bautista A00512005**

**Alberto Ortega de la Garza A00513420**

**Reyven Raul Segura Nabarret A00513647**

**Carlos Gustavo Lara Rodriguez A00513854**

**Arón Pérez Díaz A01410076**

**Segundo Avance**

Introducción

Podemos entender por el concepto de *domótica* al conjunto de sistemas capaces de automatizar una vivienda, aportando servicios de gestión de energía, seguridad y comodidad. Este conjunto de sistemas puede estar integrado por una red interior o exterior de cableado o una red inalámbrica contando con control dentro del hogar.

A partir de esta definición y con los objetivos marcados por el curso, de los cuales se hablará más adelante, como equipo hemos decidido como nuestro proyecto la automatización de una habitación en una vivienda. Al automatizar esta habitación podremos controlar el encendido y apagado de luces, el control de la intensidad de estas mismas, subir y bajar persianas, sensores de movimiento, control de temperatura, entre otros.

Finalmente todo esto podrá ser controlado a través del desarrollo de una aplicación web, que permita al usuario acceder desde una computadora, tablet o celular de manera remota y recibir retroalimentación de lo que se esta llevando a cabo en la habitación.

Objetivos Generales

1. **Salidas digitales.**

Realizar un movimiento proporcional de encendido y apagado de un objeto físico que se encuentra en una casa. (Intensidad luminosa, velocidad de giro de un motor, posición, etc.) desde un programa en lenguaje de alto nivel en una computadora.

1. **Entradas digitales.**

Recibir la información del exterior en un programa en lenguaje de alto nivel en una computadora. Puede ser temperatura, velocidad de un motor, posición, etc.

1. **Salidas analógicas.**

Realizar un movimiento proporcional de 0 al 100% de un objeto físico que se encuentra en una casa. (Intensidad luminosa, velocidad de giro de un motor, posición, etc.) desde un programa en lenguaje de alto nivel en una Computadora.

1. **Entradas analógicas.**

Recibir la información del exterior en un programa en lenguaje de alto nivel en una computadora. Puede ser temperatura, velocidad de un motor, posición, etc.

1. **Interfaz y comunicación.**

Realizar la comunicación entre la computadora y un dispositivo móvil para controlar y monitorear los puntos 1, 2, 3 y 4.

1. **Presentación final.**

Realizar una exposición del proyecto ante un jurado externo.

Objetivos Específicos

* Automatizar una habitación
* Controlar el encendido y apagado de los focos
* Controlar persianas o en su defecto la pantalla para el cañón
* Para la entrega del primer parcial solo se realizará el encendido y apagado de los focos.

Planeación del Proyecto

|  |  |
| --- | --- |
| **Fase** | **Description** |
| Salidas Digitales | Realizar un movimiento proporcional de encendido y apagado de un objeto físico que se encuentra en una casa. (Intensidad luminosa, velocidad de giro de un motor, posición, etc.) desde un programa en lenguaje de alto nivel en una computadora. |
| Entradas Digitales | Recibir la información del exterior en un programa en lenguaje de alto nivel en una computadora. Puede ser temperatura, velocidad de un motor, posición, etc. |
| Salidas Analógicas | Realizar un movimiento proporcional de 0 al 100% de un objeto físico que se encuentra en una casa. (Intensidad luminosa, velocidad de giro de un motor, posición, etc.) desde un programa en lenguaje de alto nivel en una Computadora. |
| Entradas Analógicas | Recibir la información del exterior en un programa en lenguaje de alto nivel en una computadora. Puede ser temperatura, velocidad de un motor, posición, etc. |
| Interfaz y Comunicación | Realizar la comunicación entre la computadora y un dispositivo móvil para controlar y monitorear los puntos 1, 2, 3 y 4. |
| Presentación Final | Realizar una exposición del proyecto ante un jurado externo. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Fase** | **Objetivo** | | | | **Coordinador** |
| 1 | Realizar un movimiento proporcional de encendido y apagado de un objeto físico que se encuentra en una casa. (Intensidad luminosa, velocidad de giro de un motor, posición, etc.) desde un programa en lenguaje de alto nivel en una computadora. | | | | Raul Nabarret |
| **Actividades** | | | **Tiempo** | **Responsable** | |
| 1.1 Encender de manera digital objetos electronicos | | | 2 sesiones | Jaz Tinajero | |
| 1.2 Controlar el encendido y apagado con Arduino | | | 2 sesiones | Aron Perez | |
| **Inicio** | | **Final** | | | |
| 20 de enero de 2014 | | 8 de febrero de 2014 | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Fase** | **Objetivo** | | | | **Coordinador** |
| 2 | Recibir la información del exterior en un programa en lenguaje de alto nivel en una computadora. Puede ser temperatura, velocidad de un motor, posición, etc. | | | | Jaz Tinajero |
| **Actividades** | | | **Tiempo** | **Responsable** | |
| 2.1 Adquirir información desde el circuito. | | | 2 sesiones | Jaz Tinajero | |
| 2.2 Desplegar la información en la computadora | | | 2 sesiones | Aron Perez | |
| **Inicio** | | **Final** | | | |
| 9 de febrero de 2014 | | 5 de Marzo de 2014 | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Fase** | **Objetivo** | | | | **Coordinador** |
| 3 | Realizar un movimiento proporcional de 0 al 100% de un objeto físico que se encuentra en una casa. (Intensidad luminosa, velocidad de giro de un motor, posición, etc.) desde un programa en lenguaje de alto nivel en una Computadora. | | | | Aron Perez |
| **Actividades** | | | **Tiempo** | **Responsable** | |
| 3.1 Encender de maneraanalógica objetos electronicos | | | 2 sesiones | Jaz Tinajero | |
| 3.2 Controlar el encendido y apagado e intensidad con Arduino | | | 2 sesiones | Aron Perez | |
| **Inicio** | | **Final** | | | |
| 6 de marzo de 2014 | | 28 de marzo de 2014 | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Fase** | **Objetivo** | | | | **Coordinador** |
| 4 | Recibir la información del exterior en un programa en lenguaje de alto nivel en una computadora. Puede ser temperatura, velocidad de un motor, posición, etc. | | | | Alberto Valencia |
| **Actividades** | | | **Tiempo** | **Responsable** | |
| 4.1 Adquirir información desde el circuito. | | | 2 sesiones | Raul Nabarret | |
| 4.2 Desplegar la información en la computadora | | | 2 sesiones | Alberto Ortega | |
| **Inicio** | | **Final** | | | |
| 6 de abril de 2014 | | 24 de abril de 2014 | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Fase** | **Objetivo** | | | | **Coordinador** |
| 5 | Realizar la comunicación entre la computadora y un dispositivo móvil para controlar y monitorear los puntos 1, 2, 3 y 4. | | | | Alberto Ortega |
| **Actividades** | | | **Tiempo** | **Responsable** | |
| 5.1 Diseñar página web para el control de la casa. | | | 4 sesiones | Jaz Tinajero | |
| **Inicio** | | **Final** | | | |
| 24 de abril de 2014 | | 03 de mayo de 2014 | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Fase** | **Objetivo** | | | | **Coordinador** |
| 6 | Realizar una exposición del proyecto ante un jurado externo. | | | | Carlos Lara |
| **Actividades** | | | **Tiempo** | **Responsable** | |
| 6.1 Exponer el proyecto final | | | 1 sesion | Alberto Ortega | |
| **Inicio** | | **Final** | | | |
| 4 de mayo de 2014 | | Por confirmar | | | |

Diagrama de Gantt

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Fase 1** | | | | | | | | | | | | | | | | | | |
|  | ENERO | | | | FEBRERO | | | | MARZO | | | | ABRIL | | | | MAYO | |
| **Actividades de la fase** | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 |
| 1.1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 1.2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Fase 2** | | | | | | | | | | | | | | | | | | |
|  | ENERO | | | | FEBRERO | | | | MARZO | | | | ABRIL | | | | MAYO | |
| **Actividades de la fase** | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 |
| 2.1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2.2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Fase 3** | | | | | | | | | | | | | | | | | | |
|  | ENERO | | | | FEBRERO | | | | MARZO | | | | ABRIL | | | | MAYO | |
| **Actividades de la fase** | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 |
| 3.1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3.2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

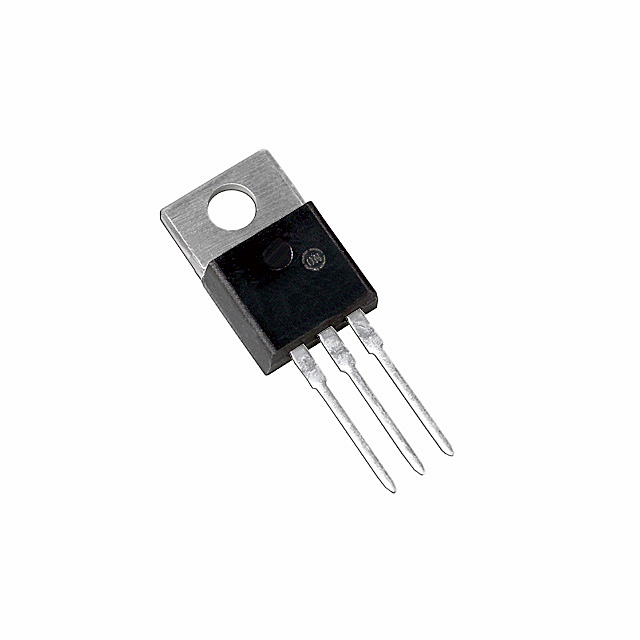
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Fase 4** | | | | | | | | | | | | | | | | | | |
|  | ENERO | | | | FEBRERO | | | | MARZO | | | | ABRIL | | | | MAYO | |
| **Actividades de la fase** | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 |
| 4.1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4.2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Fase 5** | | | | | | | | | | | | | | | | | | |
|  | ENERO | | | | FEBRERO | | | | MARZO | | | | ABRIL | | | | MAYO | |
| **Actividades de la fase** | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 |
| 5.1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

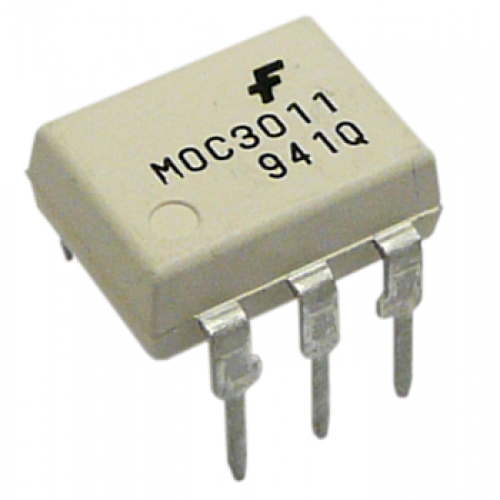
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Fase 6** | | | | | | | | | | | | | | | | | | |
|  | ENERO | | | | FEBRERO | | | | MARZO | | | | ABRIL | | | | MAYO | |
| **Actividades de la fase** | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 |
| 6.1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Marco Teórico

**MAC12D**

Un TRIAC o Triodo para Corriente Alterna es un dispositivo [semiconductor](http://es.wikipedia.org/wiki/Semiconductor), de la familia de los tiristores. La diferencia con un [tiristor](http://es.wikipedia.org/wiki/Tiristor) convencional es que éste es unidireccional y el TRIAC es bidireccional. De forma coloquial podría decirse que el TRIAC es un [interruptor](http://es.wikipedia.org/wiki/Interruptor) capaz de conmutar la [corriente alterna](http://es.wikipedia.org/wiki/Corriente_alterna). Su versatilidad lo hace ideal para el control de corrientes alternas. Una de ellas es su utilización como [interruptor](http://es.wikipedia.org/wiki/Interruptor) estático ofreciendo muchas ventajas sobre los interruptores mecánicos convencionales y los [relés](http://es.wikipedia.org/wiki/Rel%C3%A9). Funciona como interruptor electrónico y también a pila. Se utilizan TRIACs de baja potencia en muchas aplicaciones como [atenuadores](http://es.wikipedia.org/wiki/Dimmer) de luz, controles de velocidad para motores eléctricos, y en los sistemas de control computarizado de muchos elementos caseros.

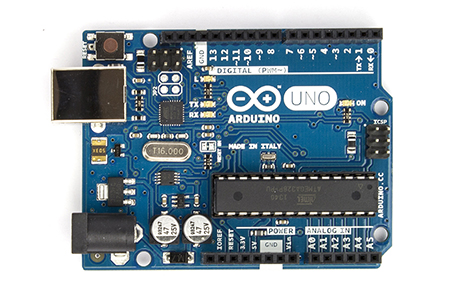
**MOC3011**

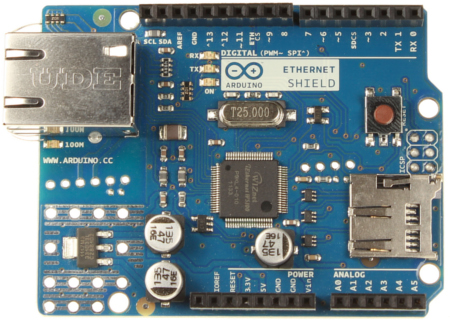
****Un optoacoplador, también llamado *optoaislador* o aislador acoplado ópticamente, es un dispositivo de emisión y recepción que funciona como un interruptor activado mediante la luz emitida por un [diodo LED](http://es.wikipedia.org/wiki/Diodo_LED) que satura un componente [optoelectrónico](http://es.wikipedia.org/wiki/Optoelectr%C3%B3nica), normalmente en forma de [fototransistor](http://es.wikipedia.org/wiki/Fototransistor) o fototriac. De este modo se combinan en un solo dispositivo semiconductor, un fotoemisor y un fotorreceptor cuya conexión entre ambos es óptica.

**LDR 10MOhms**

Un fotorresistor o LDR (por sus siglas en inglés "*light-dependent resistor*") es un [componente electrónico](http://es.wikipedia.org/wiki/Componente_electr%C3%B3nico) cuya [resistencia](http://es.wikipedia.org/wiki/Resistencia_el%C3%A9ctrica) varia en función de la luz. El valor de resistencia eléctrica de un LDR es bajo cuando hay luz incidiendo en él (puede descender hasta 50 ohms) y muy alto cuando está a oscuras (varios megaohmios).

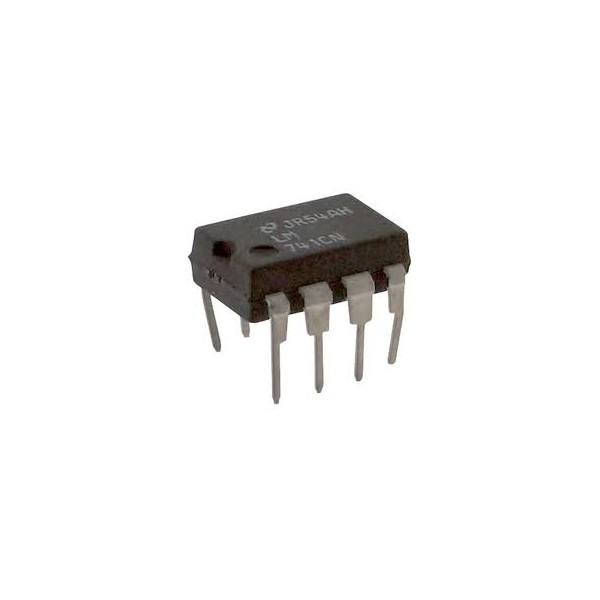
**Arduino UNO**

El Arduino Uno es una placa electronica basada en el ATmega328. Cuenta con 14 pines digitales de entrada / salida (de los cuales 6 pueden utilizarse para salidas PWM), 6 entradas analógicas, un 16 MHz resonador cerámico, una conexión USB, un conector de alimentación, una cabecera ICSP, y un botón de reinicio. El Arduino Uno puede ser alimentado a través de la conexión USB o con una fuente de alimentación externa. Cada uno de los 14 pines digitales en el Uno se puede utilizar como una entrada o salida. Funcionan a 5 voltios.

**Arduino Ethernet Shield**

La Arduino Ethernet Shield permite que una placa Arduino se conecte a internet. Se basa en el chip ethernet Wiznet W5100 (hoja de datos). El Wiznet W5100 proporciona una red (IP) apilar capaz de TCP y UDP. Soporta hasta cuatro conexiones de socket simultáneas. Hay una ranura de tarjeta micro-SD a bordo, que se puede utilizar para almacenar archivos del servidor a través de la red. Es compatible con el Arduino Uno y Mega (utilizando la librería Ethernet). El lector de tarjetas microSD a bordo es accesible a través de la Biblioteca SD.

**LM741**

La serie LM741 son amplificadores de propósito general operativo que ofrecen un rendimiento mejorado sobre los estándares de la industria. Los amplificadores ofrecen muchas características que hacen que su aplicación casi infalible: la protección de sobrecarga en la entrada y salida, no latch-up cuando el común se supera rango de modo, así como la libertad de oscilaciones. Tiene en su desempeño garantizar de 0 ° C a + 70 ° C.

**Tarjeta SD**

Las tarjetas MicroSD o Transflash corresponden a un formato de tarjeta de [memoria flash](http://es.wikipedia.org/wiki/Memoria_flash) más pequeña que la [MiniSD](http://es.wikipedia.org/wiki/MiniSD), desarrollada por [SanDisk](http://es.wikipedia.org/wiki/SanDisk); adoptada por la[Asociación de Tarjetas SD](http://es.wikipedia.org/wiki/Asociaci%C3%B3n_de_Tarjetas_SD)bajo el nombre de «microSD» en julio de [2005](http://es.wikipedia.org/wiki/2005). Mide tan solo 15 × 11 × 1 milímetros, lo cual le da un área de 165 mm.

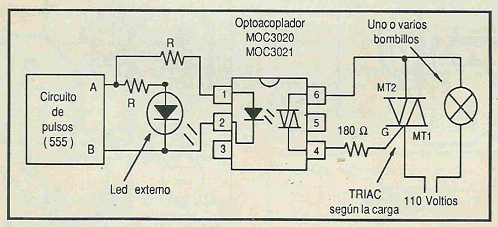
**Router TP-LINK TL-WR741ND**

El TL-WR741ND es un dispositivo que integra un switch de 4 puertos y un router para compartir la conexión a Internet. El TL-WR741ND ofrece la posibilidad de conectar los dispositivos tanto de forma inalámbrica como fija a través de un cable.

Para manejar focos debemos utilizar SCRs o TRIACs de la ca­pacidad adecuada en corriente y voltaje y debidamente montados sobre disipadores de calor. La capacidad de corriente en amperios y el voltaje del SCR o TRIAC los determi­nan la cantidad de focos que se van a conectar. Estos valores deben ser por lo menos un 50 % mayores al consu­mo de la carga.

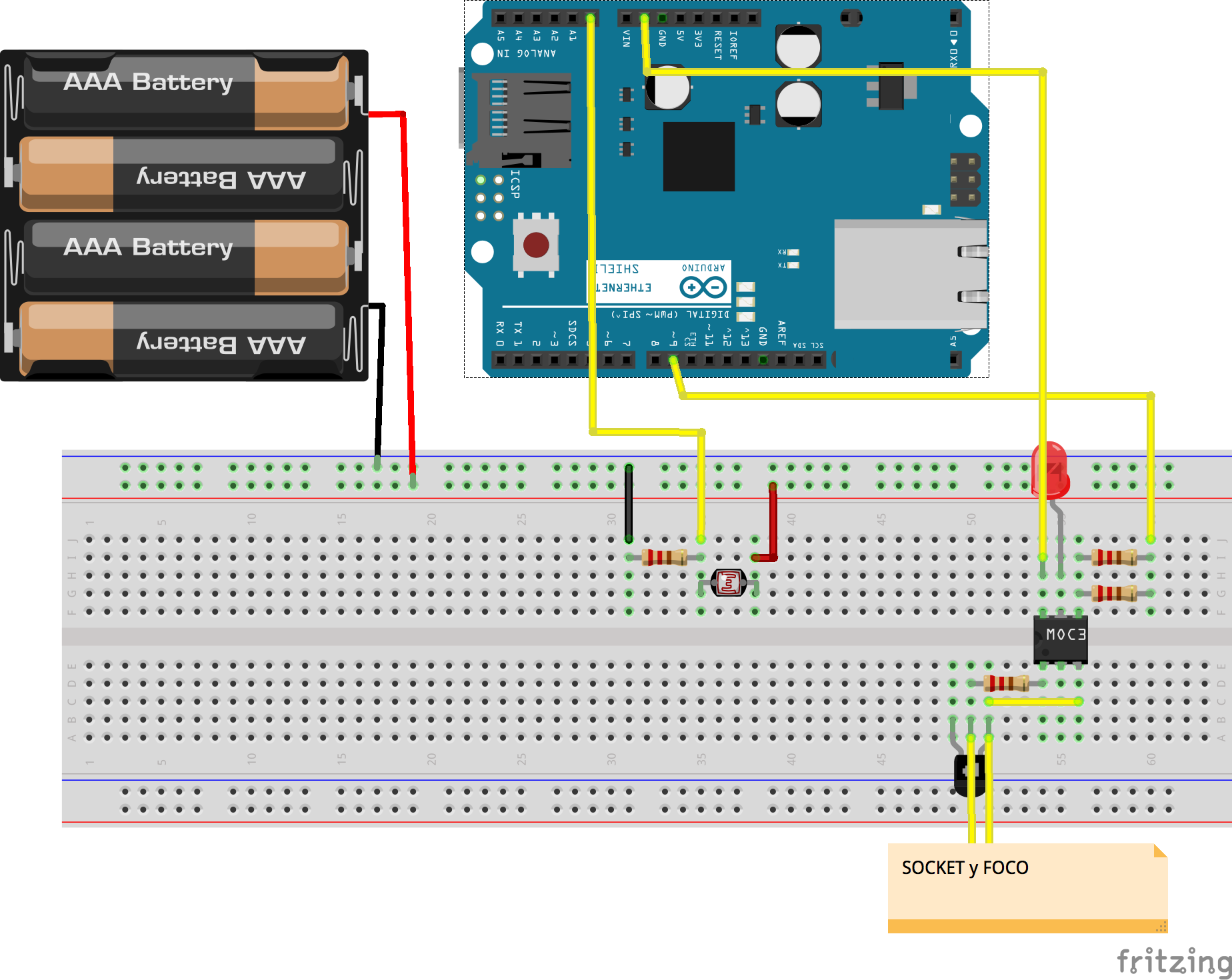
Para acoplar un SCR o TRIAC y su respectiva carga a un circuito electrónico de control que produce pulsos, ya sea por me­dio de un circuito integrado como el 555 o similares, o por medio de transistores, el método más fácil y seguro es utilizar uno o varios optoacopladores del tipo de TRIAC.

Para cada salida del circuito que alimenta un LED, o sean dos terminales, se emplea el siguiente circuito:



Listado de Materiales y Equipo Utilizado

* Arduino uno
* MAC12D
* MOC3011
* LDR 10MOhms
* Foco Incandescente
* Socket
* Protoboard
* Cable
* Batería 5V
* Ethernet Shield para Arduino UNO
* Amplificador LM741
* Tarjeta SD
* Cable Ethernet
* Router TP-LINK TL-WR741ND

Diagrama del circuito

Programa utilizado

#include <SD.h>

#include <SPI.h>

#include <Ethernet.h>

int led = 5;

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

byte ip[] = { 192, 168, 1, 71 };

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

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

EthernetServer server(80);

String readString;

void setup() {

Serial.begin(9600);

pinMode(led, OUTPUT);

Ethernet.begin(mac, ip, gateway, subnet);

server.begin();

Serial.print("server is at ");

Serial.println(Ethernet.localIP());

}

void loop() {

EthernetClient client = server.available();

if (client) {

while (client.connected()) {

if (client.available()) {

char c = client.read();

if (readString.length() < 100) {

readString += c;

}

if (c == '\n') {

Serial.println(readString);

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

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

client.println();

client.println("<!doctype html>");

client.println("<html>");

client.println("<head>");

client.println("<title>Domotica</title>");

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

client.println("<style>");

client.println("html{font-family:'Trebuchet MS';font-size:14px;height:100%;width:100%;}");

client.println("a{text-decoration:none;}");

client.println(".header{background:blue;height:100px;width: 100%;position:fixed;}");

client.println(".button{color:black}"); client.println(".content{background:white;height:900px;width:100%;}"); client.println(".footer{background:gray;height:100px;width:100%;}");

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

client.println("<body>");

client.println("<header class='header'>");

client.println("<h1 class='title'>Proyecto de Domotica</h1>");

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

client.println("<body class='content'>");

client.println("<a class='button'href='\'/?button1on\'\'>Turn On LED</a>");

client.println("<a class='button'href='\'/?button1off\'\'>Turn Off LED</a>");

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

client.println("<footer class='footer'>");

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

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

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

if (readString.indexOf("?button1on") >0){

digitalWrite(led, HIGH);

}

if (readString.indexOf("?button1off") >0){

digitalWrite(led, LOW);

}

client.stop();

delay(1);

readString="";

}

}

}

}

}

#include <SPI.h>

#include <Ethernet.h>

#include <SD.h>

// MAC address from Ethernet shield sticker under board

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

IPAddress ip(192, 168, 1, 92); // IP address, may need to change depending on network

EthernetServer server(80); // create a server at port 80

File webFile;

void setup()

{

Ethernet.begin(mac, ip); // initialize Ethernet device

server.begin(); // start to listen for clients

Serial.begin(9600); // for debugging

// initialize SD card

Serial.println("Initializing SD card...");

if (!SD.begin(4)) {

Serial.println("ERROR - SD card initialization failed!");

return; // init failed

}

Serial.println("SUCCESS - SD card initialized.");

// check for index.htm file

if (!SD.exists("index.htm")) {

Serial.println("ERROR - Can't find index.htm file!");

return; // can't find index file

}

Serial.println("SUCCESS - Found index.htm file.");

}

void loop()

{

EthernetClient client = server.available(); // try to get client

if (client) { // got client?

boolean currentLineIsBlank = true;

while (client.connected()) {

if (client.available()) { // client data available to read

char c = client.read(); // read 1 byte (character) from client

// last line of client request is blank and ends with \n

// respond to client only after last line received

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

// send a standard http response header

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

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

client.println("Connection: close");

client.println();

// send web page

webFile = SD.open("index.htm"); // open web page file

if (webFile) {

while(webFile.available()) {

client.write(webFile.read()); // send web page to client

}

webFile.close();

}

break;

}

// every line of text received from the client ends with \r\n

if (c == '\n') {

// last character on line of received text

// starting new line with next character read

currentLineIsBlank = true;

}

else if (c != '\r') {

// a text character was received from client

currentLineIsBlank = false;

}

} // end if (client.available())

} // end while (client.connected())

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

client.stop(); // close the connection

} // end if (client)

}

Conclusiones generales

Para esta primera entrega del proyecto fuimos capaces de controlar el encendido y apagado de un foco incandescente mediante la utilización de los componentes ya mencionados.

Para esta segunda entrega pudimos aprender a utilizar el Shield de Ethernet para el Arduino, esto representó una curva de aprendizaje ya que ninguno de los integrantes del equipo lo había utilizado anteriormente. El uso de este shield del Arduino nos facilitó el uso del mismo como servidor, que al conectarlo dentro de una red local nos permite desplegar una página web en una IP asignada y a través de dicha página web se logra controlar el encendido y apagado de las luces, así como regular la intensidad de luz variando por porcientos. El uso de este shield nos ayudó a facilitar el manejo de la aplicación web con la cual controlamos ahora el sistema.

Bibliografía

Malvino, Albert Paul (2000). *Principios de Electrónica*. McGraw-Hill/Interamericana de España,S. A. U

<http://www.alldatasheet.es/datasheet-pdf/pdf/172082/ONSEMI/MAC12D.html>