



Universidad Nacional de La Matanza

Florencio Varela 1903 - San Justo - Buenos Aires - Argentina

Sistemas Operativos Avanzados

Trabajo Práctico:

Zapatilla Eléctrica inteligente

Comisión: Miércoles Noche

Integrantes :

Apellido	Nombre	DNI
Barja	Alex	43.180.648
Romero	Cristian	34.297.484

Objetivo:

Desarrollar un sistema de control de consumo eléctrico domiciliario haciendo uso de sensores de corriente eléctrica, con la posibilidad de controlar el encendido y apagado de distintos aparatos reales presentes en un hogar a distancia empleando un celular, manual o de forma automática al superar el límite máximo permitido de consumo.

Software:

- Android studio 2.0:

Android Studio es un entorno de desarrollo integrado para la plataforma Android. Fue anunciado el 16 de mayo de 2013 en la conferencia Google I/O, y reemplazó a Eclipse como el IDE oficial para el desarrollo de aplicaciones para Android. La primera versión estable fue publicada en diciembre de 2014. Está basado en el software IntelliJ IDEA de JetBrains, y es publicado de forma gratuita a través de la Licencia Apache 2.0. Está disponible para las plataformas Microsoft Windows, Mac OS X y GNU/Linux.

- Arduino 1.6.0:

Arduino es un programa para microcontraladores de una sola placa que fue desarrollado en 2005 y comenzó a desarrollarse como un proyecto estudiantil con un costo de solo . Arduino se utiliza para hacer que la creación de aplicaciones interactivas y otros programas sea mucho más accesible para los usuarios. Arduino permite que lo usuarios carguen un programa a un microcontrolador sin la necesidad de un chip u otro dispositivo. Arduino incluye una gama de herramientas y funciones y viene equipado en una plataforma de código abierto con placa que se basó en microcontrolador de 8 bit Atmel AVR.

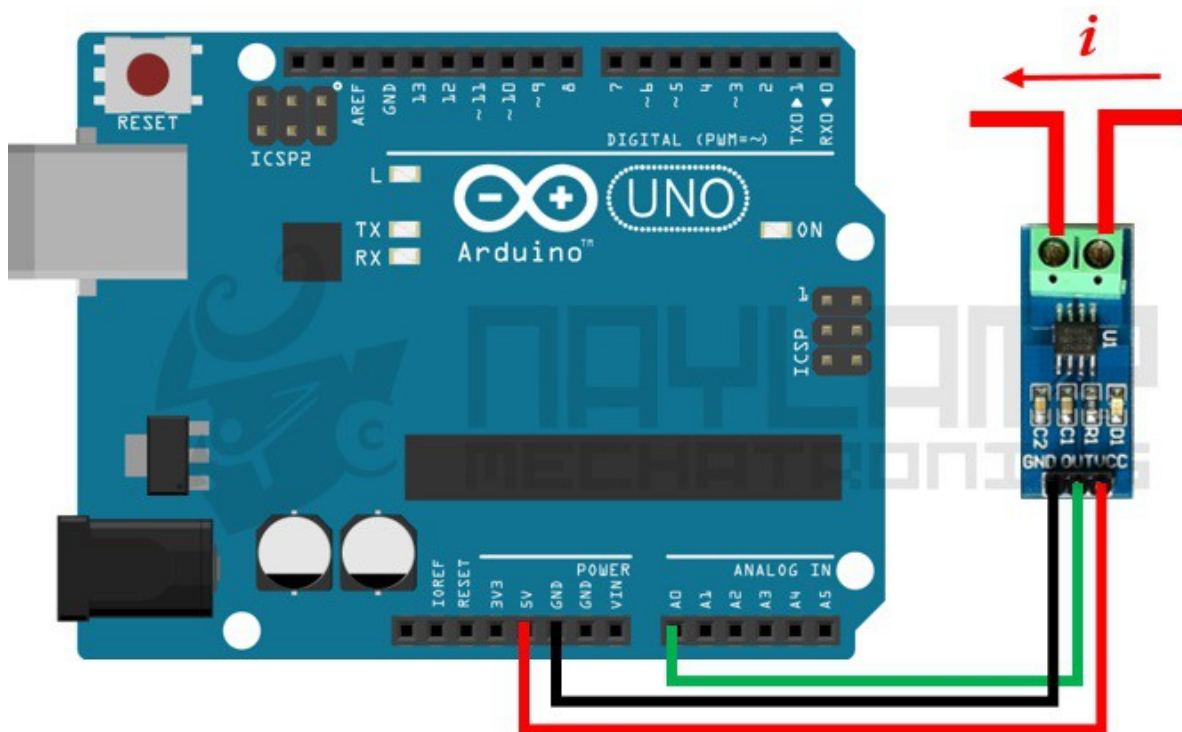
Hardware:

- Arduino Uno R3
- Cable Usb
- Sensor De Corriente $\pm 20a$ Efecto Hall Acs712 Arduino

- Modulo Relay Rele De 4 Canales 5v 10a Arduino
- 40 Cables Macho Macho 20cm Dupont Arduino
- Gabinete con 4 tomas corriente 220v, 5A, 60hz

Circuito Sistema embebido:

- 1_ conectamos primero en serie el sensor de corriente a la entrada A0- entrada analógica 0- para poder censar la corriente total consumida por todas los aparatos conectados en funcionamiento.



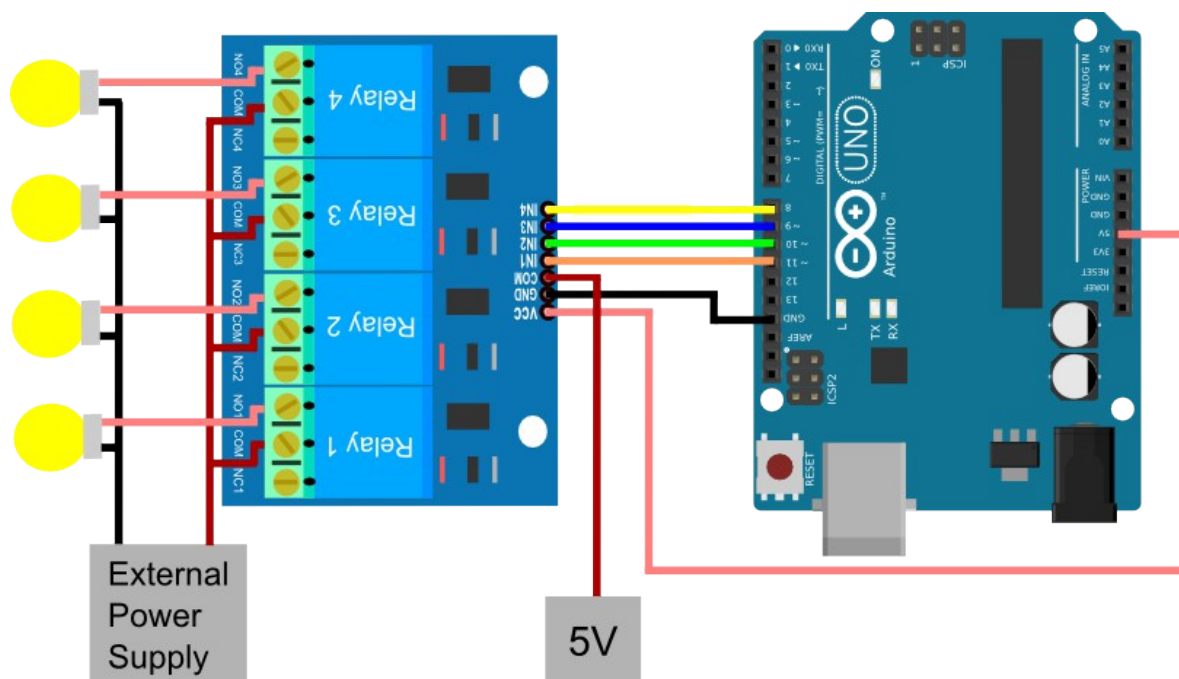
Estructura interna de la zapatilla eléctrica.



Estructura externa final:



2_ Haciendo uso de un módulo de cuatro relés, para controlar el encendido de cargas mayores a las soportadas por arduino. Lo conectamos a las cuatro salidas digitales del arduino uno, para asi tener un total control desde placa de desarrollo.



En funcionamiento:



Programa Arduino:

```
#include <SPI.h>
```

```
#include <Ethernet.h>
```

```
byte mymac[] = { 0x90, 0xA2, 0xDA, 0x00, 0x9B, 0x36 };
```

```
byte myip[] = { 192, 168, 0, 8 };
```

```
EthernetServer server(80);
```

```
int led1 = 5;
```

```
int led2 = 2;
```

```
int led3 = 3;
```

```
int led4 = 4;
```

```
String readString = String(30);
```

```
int offset=0;
```

```
float maximo = 0.0;
```

```
String estadoCircu;
```

```
float get_voltage(int);
```

```
void setup() {
```

```
    Ethernet.begin(mymac, myip);
```

```
    pinMode(led1, OUTPUT);
```

```
    pinMode(led2, OUTPUT);
```

```
    pinMode(led3, OUTPUT);
```

```
    pinMode(led4, OUTPUT);
```

```
    digitalWrite(led1,HIGH);
```



```
digitalWrite(led2,HIGH);  
digitalWrite(led3,HIGH);  
digitalWrite(led4,HIGH);  
Serial.begin(9600);  
}
```

```
float get_corriente(int n_muestras){  
    float voltage=0;  
    for(int i=0;i<n_muestras;i++)  
    {  
        voltage =voltage+analogRead(A0) * (5.0 / 1023.0);  
    }  
    voltage=voltage/n_muestras;  
    return(voltage);  
}
```

```
void loop() {  
    EthernetClient client = server.available();  
    if(client) {  
        while(client.connected()) {  
            if(client.available())  
            {  
                char c = client.read();  
                if(readString.length() < 30) {
```



```
    readString += (c);
}
if(c == '\n')
{
    if(readString.indexOf("led1") >= 0) {
        digitalWrite(led1, !digitalRead(led1));
    }
    if(readString.indexOf("led2") >= 0) {
        digitalWrite(led2, !digitalRead(led2));
    }
    if(readString.indexOf("led3") >= 0) {
        digitalWrite(led3, !digitalRead(led3));
    }
    if(readString.indexOf("led4") >= 0) {
        digitalWrite(led4, !digitalRead(led4));
    }
    if((offset = readString.indexOf("maximo0")) >= 0){
        maximo = (readString+offset).toFloat();
    }
    float volta = get_corriente(1);
    delay(100);
    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>Tutorial</title>");
```

```
content="\2,URL=http://192.168.0.8\">");
```

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

```
    if(digitalRead(led1)) {
```

```
        estadoCircu = "ON";
```

```
    } else {
```

```
        estadoCircu = "OFF";
```

```
    }
```

```
client.println("<form action=\"led1\" method=\"get\">");
```

```
client.println("<button type=submit style=\"width:200px;\">Led1 -  
"+estadoCircu+"</button>");
```

```
client.println("</form>volt-");
```

```
client.println(volta);
```

```
client.println("-end<br />");
```

```
client.println("</form>max-");
```

```
client.println(maximo);
```

```
client.println("-end<br />");

if(digitalRead(led2)) {

    estadoCircu = "ON";

} else {

    estadoCircu = "OFF";

}

client.println("<form action=\"led2\" method=\"get\">");

client.println("<button type=submit style=\"width:200px;\">Led2 -  
"+estadoCircu+"</button>");

client.println("</form> <br />");

    if(digitalRead(led3)) {

        estadoCircu = "ON";

    } else {

        estadoCircu = "OFF";

    }

client.println("<form action=\"led3\" method=\"get\">");

client.println("<button type=submit style=\"width:200px;\">Led3 -  
"+estadoCircu+"</button>");

client.println("</form> <br />");


if(digitalRead(led4)) {

    estadoCircu = "ON";

} else {

    estadoCircu = "OFF";
```

```
}

client.println("<form action=\"led4\" method=\"get\">");

client.println("<button type=submit style=\"width:200px;\">Led4 -  
"+estadoCircu+"</button>");

client.println("</form> <br />");

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

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

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

readString = "";

client.stop();

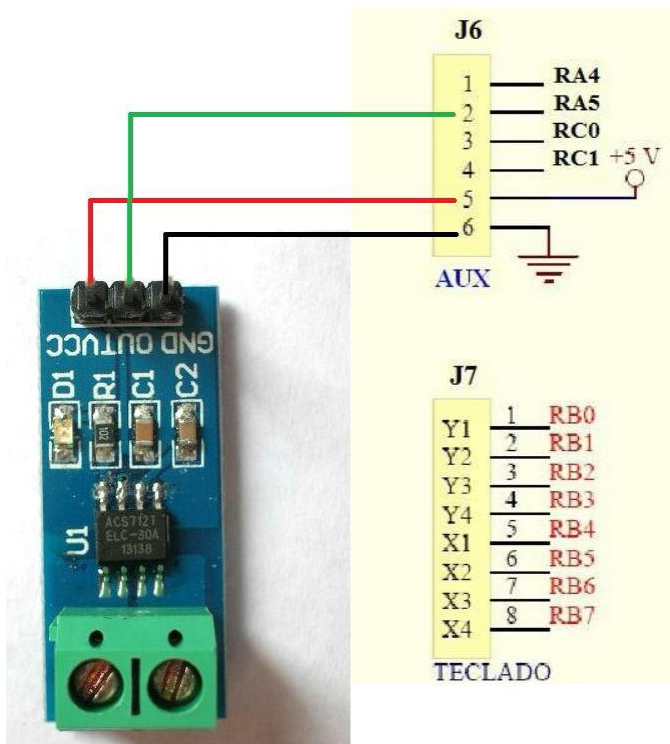
delay(100);

} } } }
```

Sensores y actuadores empleados:

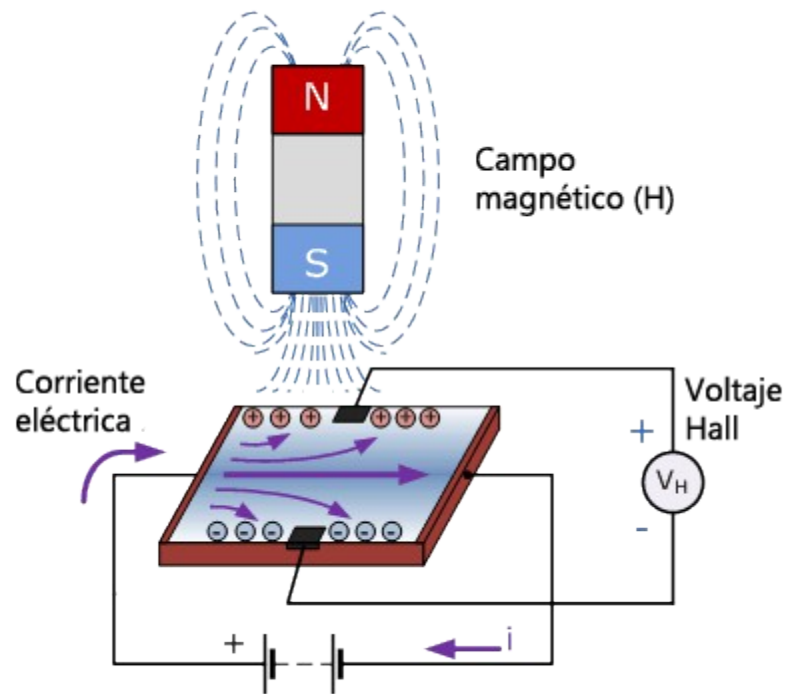
Sensor De Corriente ACS712:

El sensor ACS712 de Allegro, es un sensor de corriente por efecto hall, que provee un solución económica y precisa para medir corriente en AC o DC, ya sea en ambientes industriales o comerciales. Este Sensor funciona transformando un campo magnético surgido de el paso de la corriente por un alambre de cobre interno en el sensor, y convirtiendo este campo en un voltaje variable. Esto significa que a mayor cantidad de corriente que tengamos, mayor voltaje vamos a tener en un pin. Este sensor viene en 3 modelos distintos: ACS712ELCTR-05B-T que mide hasta 5A, el ACS712ELCTR-20A-T que mide hasta 20A y el ACS712ELCTR-30A-T que mide hasta 30A. Las diferencias entre cada uno de los modelos es que las variaciones de voltaje en su pin de salida es siempre la misma, por ende para cualquier modelo su salida analógica variara entre 0 y 5v dandonos una mejor precision en el modelo de 5A que en el de 30A.



Sensores Hall:

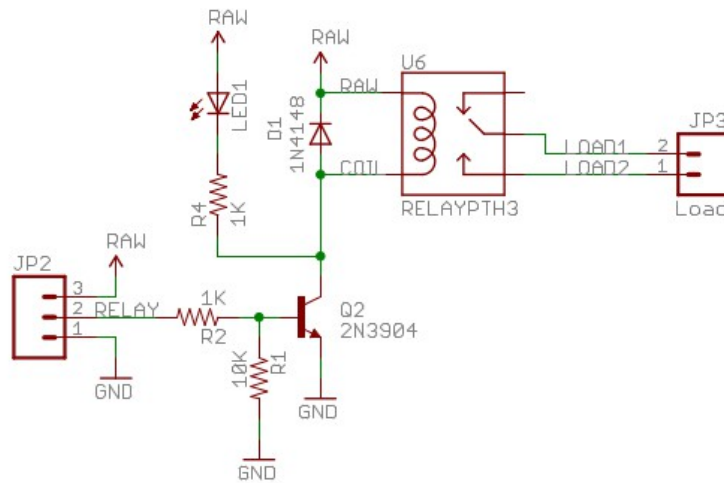
Los sensores Hall son ampliamente utilizados. Por ejemplo, en la industria del automóvil se usan para funciones tan dispares como para el accionamiento de los cinturones de seguridad, o la medición de la posición del árbol de levas. También se usan para medir velocidades de fluidos, detección de metales, factores de inducción, entre otras muchas aplicaciones.



Modulo Relay Rele De 4 Canales 5v 10a Arduino:

Es un **interruptor** que podemos activar mediante una señal eléctrica. Un relé está compuesto por una bobina que al circular una pequeña corriente (3.3V, 5V o 12V) genera un campo magnético y este a su vez hace mover una placa metálica abriendo o cerrando un circuito eléctrico independiente que es por el que circula un voltaje superior (220V).

Esquema interno modulo:



Android: Actividades conexión y control

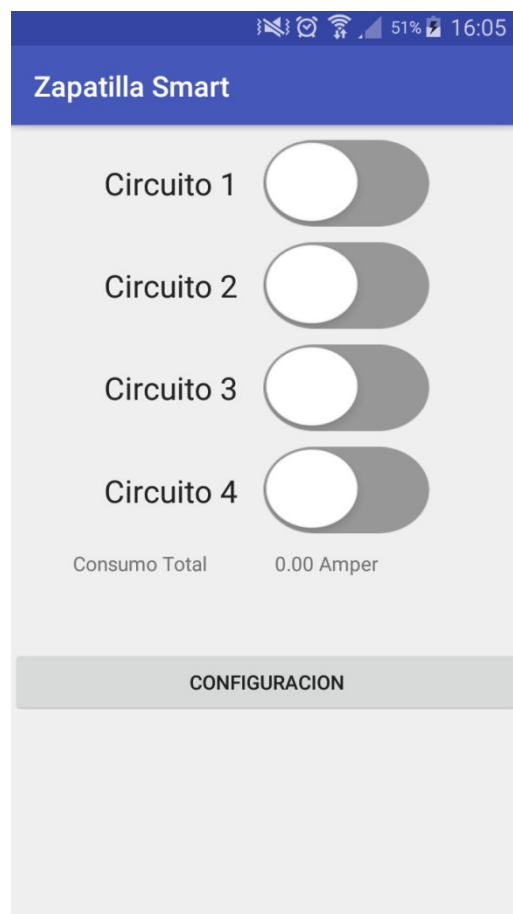
Activities conexión:

Con esta actividades el usuario podrá atreves de una interfaz intuitiva indicar la ip correspondiente al arduino y establecer el nivel de corriente máximo permitida de consumo. Luego con el uso de button e intents se pasara a la activities principal.



Activities control:

En esta actividades el usuario podrá activar y desactivar de manera independiente y a distancia los aparatos conectados a arduino. Desplazándola a la derecha activa el dispositivo. A su vez puede visualizar el consumo que dichos aparatos están haciendo de la red eléctrica. Opcionalmente puede volver a reconfigurar la tensión máxima permitida. Además podrá usar el sensor de proximidad para poder desactivar todos los aparatos conectados a la zapatilla eléctrica. En caso de superarse el valor máximo permitido de consumo el usuario recibirá una notificación informándole del exceso de consumo.



Conclusiones:

Con este trabajo practico a diferencia de en otras materias más teóricas pudimos poner a prueba los conocimientos adquiridos tanto en esta materia como en anteriores, y así poder concluir en un producto que pudiéramos utilizar cualquiera de los integrantes del grupo en el hogar.

El grupo comenzó con distintas temas para encarar el proyecto aplicando los conocimiento de la catedra, la primera idea fue la de realizar un sistema que permita una comunicación utilizando el código morse entre el arduino y un celular pero al ver que no era algo muy útil para los integrantes del grupo, planteamos otras alternativas como fueron la detección de rostro utilizando las bibliotecas opencv en Linux Yocto pero tuvimos la limitante del poco tiempo para practicar con la placa Intel Galileo debido a los horarios. Finalmente elegimos llevar a cabo la idea más práctica y útil para todos en el grupo.

Al comienzo el grupo empezó con muchas dificultades por nunca haber utilizado ninguna de las placas de desarrollo ni sensores, pero luego con la ayuda de la catedra, y la numerosa bibliografía que tiene la placa de desarrollo se pudo avanzar más rápido, y terminar de forma satisfactoria el trabajo practico.