SEP
SECRETARÍA DE EDUCACIÓN PÚBLICA

TECNOLÓGICO NACIONAL DE MÉXICO
INSTITUTO TECNOLÓGICO DE LEÓN

Ingeniería en Sistemas Computacionales

Sistemas Programables

Practica 2: Alarma Visual

PRESENTA:

Andrea Sarahi Pérez Rodríguez

NOMBRE DEL PROFESOR:

Ing. Levy Rojas Carlos Rafael

LEÓN, GUANAJUATO **Periodo: Enero-Julio 2018**

Practica 2: Alarma Visual

Programa de una alarma visual, que usa un LED RGB para cambiar de color y un REED SWITCH como un pulsador. Cuando la puerta se abre, la cual tiene un imán, el LED cambia a color rojo y parpadea dando aviso de que la puerta fue abierta, cuando esta todo normal, es decir, la puerta cerrada el LED esta de color azul. También tiene agregado el MODO SLEEP, para el bajo consumo del arduino.

Que usa las funciones:

- SLEEP_xS: Fija el tiempo que estará "dormido" usando para ello el Watchdog.
- ADC_OFF: Apaga los convertidores Analógico al Digital.
- BOD_OFF: Apaga el circuito de Brown Out Detection, que es un circuito que sirve como detector de niveles de tensión peligrosamente bajos que podrían incluso llegar dañar los circuitos.

Materiales:

- 1 LED RGB (Es un led que se conforma con el Rojo(Red), Verde(Green) y Azul(Blue), y en combinación de estos 3 conforma otros colores)
- 1 REED SWITCH (es como un botón)
- 2 resistencias de 220 ohms (pueden ser de 330 ohms)
- 1 resistencia de 1000 ohms (1kilo)
- 1 Arduino UNO
- Varios cables de colores

Software:

- Programa Arduino: para realizar la programación del Arduino UNO
- fritzing: para realizar los esquemas del circuito.

Código:

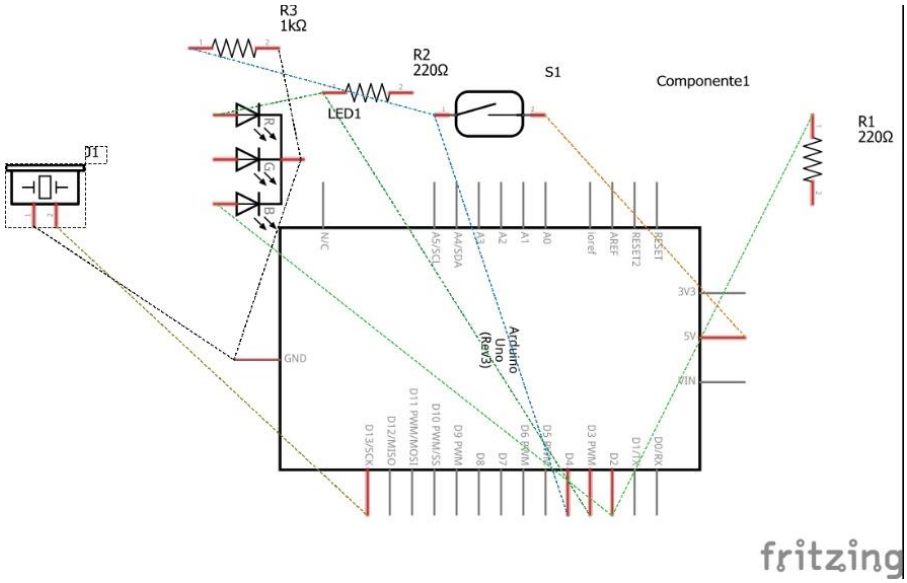
```
/* Programa de una alarma visual, que usa un LED RGB para cambiar de color y un
REED SWITCH como un pulsador.
 *  Cuando la puerta se abre, la cual tiene un imán, el LED cambia a color rojo y parpadea
dando aviso de que la puerta fue abierta, además se escucha un sonido dando a entender
esto,
 *  Cuando esta todo normal, el LED esta de color azul y no se escucha ningún sonido.
 *  También tiene agregado el MODO SLEEP, para el bajo consumo del arduino.
*/

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

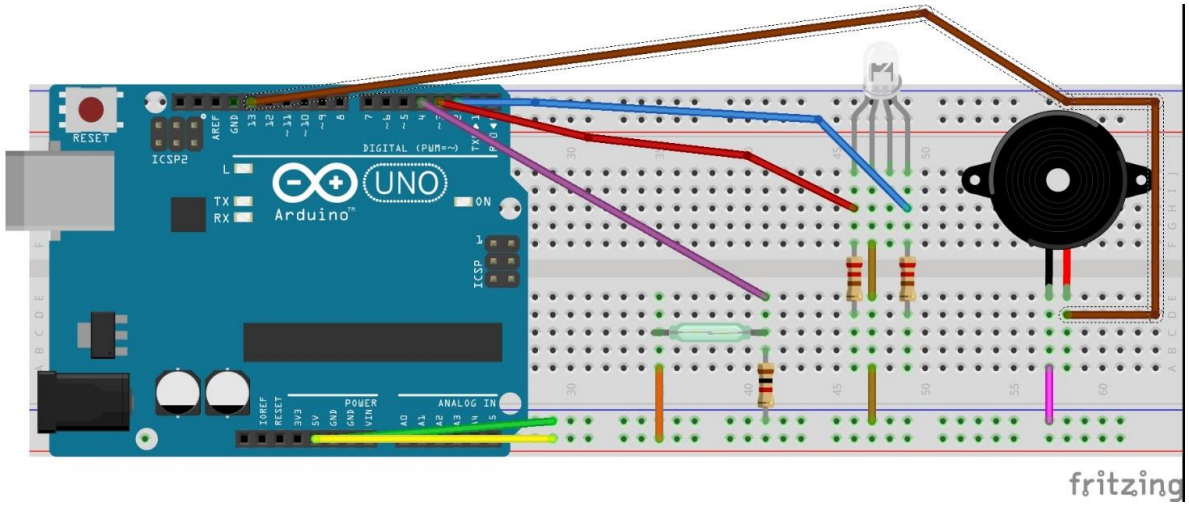
```
//Incluye la librería
#include <LowPower.h>

//Declaracion de los puertos de entradas y salidas. Variables
int contacto = 4; //Pin asignado al REED SWITCH
int ledAzul = 2; //Pin asignado al LED Azul
int ledRojo = 3; //Pin asignado al LED Rojo
int zumbador = 13; //Pin asignado al BUZZER
int frecuencia = 220; //Constante con la que se repetira el sonido

//Funcion Principal, se ejecuta cada vez que el arduino se inicia
void setup()
{
  pinMode(contacto, INPUT); //El REED SWITCH como una entrada
  pinMode(ledAzul, OUTPUT); //El LED Azul como una salida
  pinMode(ledRojo, OUTPUT); //El LED Rojo como una salida
  pinMode(zumbador, OUTPUT); //El BUZZER como salida
}

//Funcion ciclica, esta funcion se mantiene ejecutando cuando este energizado el Arduino
void loop()
{
  //Si el iman se aleja al REED SWITCH
  if(digitalRead(contacto) == LOW)
  {
    for(int a = 0; a < 50; a++)
    {
      color(255,0,0); //Prende el LED Rojo
      delay(50); //Tiempo
      color(0,0,0); //Para que se apague el LED
      delay(50); //Tiempo
      tone(zumbador, frecuencia); //Sueno o se escucha el sonido
    }
  }
  else
  {
    //Si el iman se acerca al REED SWITCH, el LED Azul, permanece encendido
    color(0,0,255);
    delay(50); //Tiempo
    noTone(zumbador); //No se escucha nada
  }
  //Le envio un valor al sleep, que son los segundos que queremos que duerma el arduino
```

```
  sleep(15);
}

//Funcion Color
void color(int rojo, int verde, int azul){
  //Escritura del PWM del color Rojo
  analogWrite(ledRojo, 255 - rojo);

  //Escritura de PWM del color Azul
  analogWrite(ledAzul, 255 - azul);
}

//Se hace uso de intervalos de 8 segundos y hace comprobaciones en intervalos pequeños, como esto lo hace en microsegundo, no se ve afectado entre cuando se duerme y se vuelve a despertar
//el arduino, su funcion continua como si nunca se durmiera.
void sleep(int sec)
{
  while(sec >= 8)
  {
    LowPower.powerDown(SLEEP_8S, ADC_OFF, BOD_OFF);
    sec -= 8;
  }

  if(sec >= 4)
  {
    LowPower.powerDown(SLEEP_4S, ADC_OFF, BOD_OFF);
    sec -= 4;
  }

  if(sec >= 2)
  {
    LowPower.powerDown(SLEEP_2S, ADC_OFF, BOD_OFF);
    sec -= 2;
  }

  if(sec >= 1)
  {
    LowPower.powerDown(SLEEP_1S, ADC_OFF, BOD_OFF);
    sec -= 1;
  }
```

}

Diagrama:





Aplicación Arduino, en donde se hace la programación.

AlarmaVisual Arduino 1.6.8

Archivo Editar Programa Herramientas Ayuda

AlarmaVisual

```
}

//Se hace uso de intervalos de 8 segundos y hace comprobaciones en intervalos pequeños, como esto lo hace en microsegundo, no se ve afectado entre cuando se duerme y se vuelve a despertar
//el arduino, su funcion continua como si nunca se durmiera.
void sleep(int sec)
{
  while(sec >= 8)
  {
    LowPower.powerDown(SLEEP_8S, ADC_OFF, BOD_OFF);
    sec -= 8;
  }

  if(sec >= 4)
  {
    LowPower.powerDown(SLEEP_4S, ADC_OFF, BOD_OFF);
    sec -= 4;
  }

  if(sec >= 2)
  {
    LowPower.powerDown(SLEEP_2S, ADC_OFF, BOD_OFF);
    sec -= 2;
  }

  if(sec >= 1)
  {
    LowPower.powerDown(SLEEP_1S, ADC_OFF, BOD_OFF);
    sec -= 1;
  }
}
```

Subido

```
El Sketch usa 3.424 bytes (10%) del espacio de almacenamiento de programa. El máximo es 32.256 bytes.
Las variables Globales usan 43 bytes (2%) de la memoria dinámica, dejando 2.005 bytes para las variables locales. El máximo es 2.048 bytes.
```

47 Arduino/Genuino Uno en COM6