Universidad Politécnica de Madrid

Grado en Ingeniería Electrónica y Automática

Asignatura

SISTEMAS ELECTRÓNICOS DIGITALES

CONTROL DE LUCES DE UNA VIVIENDA

STM32F411

Curso 2019-2020

Universidad Politécnica de Madrid

Grado en Ingeniería Electrónica y Automática

Datos del Grupo

DIEGO ESCALONA IRALA	*53355*
ELENA ACUÑA MATEOS	*52215*
EDUARDO FRANCISCO AZNAR TORRES	*52236*

Contenido

1. INTRODUCCIÓN

Nuestro proyecto se basa en la programación del comportamiento de las luces de una vivienda.

Para ello hemos construido una maqueta de la vivienda con diferentes estancias, donde controlaremos las diferentes luces que se encuentran en ella.

Para ello contaremos con dos botones uno de forma manual y otro de forma automática, a través de un sensor ultrasónico.

2. MATERIALES

- Placa STM32F411.
- Cables y resistencias para las conexiones entre los elementos y la placa.
- Interruptores de botón.
- Sensor ultrasónico HC-SR04.
- Resistencia LDR y potenciómetro.
- Leds y un led RGB.

3. ELEMENTOS Y FUNCIONES PROGRAMADAS

3.1. FUNCIONES PRINCIPALES

- **void ctrl_casa(void)**

  A través de esta función nos permite realizar el toggle de las distintas estancias de la casa.
  - Cocina
  - Baño
  - Salón
  - Habitación
  - Buhardilla
  - Jardín

  Donde dentro de ella se encuentra **ctrl_led** la cual realiza el encendido de los distintos leds.

- **void off_casa(void)**

  A partir de esta función se realiza el apagado de las estancias de la casa. Donde llamaremos a la función **off_led,** la cual realiza el apagado de los leds cuando se pulsa el interruptor.

- **void ModeSelect (int estado, GPIO_PinState led_azul, GPIO_PinState led_naranja)**

  Esta función nos permite conocer en qué estado nos encontramos:
  Estado1=Modo manual. (LED Azul, placa)
  Estado2=Modo automático. (LED Naranja, placa)

3.2. FUNCIÓN SENSOR

En la estancia del salón tenemos un sensor, el cual detectará presencia y encenderá el LED, y se mantendrá encendido durante unos segundos tras dejar de detectar presencia.

La función principal de este sensor es: **void ctrl_hcsr04()**

La cual tiene unos parámetros específicos de este sensor, para que realice la rutina de medición:

1. Activación 10uSec del Trigger.
2. Espera del flanco de subida de Echo.
3. Comienzo medida ancho de pulso de Echo uSec.
4. Conversión a cm.(***distancia = (numTicks + 0.0f)*2.8*VelSonido***)

```
//variables Sensor hcsr04
const float VelSonido = 0.0343/2;
float distancia;
uint32_t numTicks = 0;
_Bool flag=0;
```

A continuación se realiza la rutina para el encendido de los leds, según la detección de la distancia, la cual será menor que 7 cm.

La espera de este tiempo tras desactivarse la detección de presencia se realiza con la siguiente función donde tiene la siguiente estructura:

```
void usDelay(uint32_t uSec){ //funcion para la espera en uSec del sensor hsrc04
  if(uSec < 2) uSec = 2;
  usTIM->ARR = uSec - 1;  /*añadir valores al registro*/
  usTIM->EGR = 1;         /*Reinicializar el contador*/
  usTIM->SR &= ~1;     //RESET del flag
  usTIM->CR1 |= 1;     //Habilitar contador
  while((usTIM->SR&0x0001) != 1);
  usTIM->SR &= ~(0x0001);
}
```

3.3. TEMPORIZADOR

Hemos realizado la programación de dos temporizadores:

1. TIM necesario para el PWM, el cual tendrá varios canales:

```
  //inicializacion de los canales PWM
  HAL_TIM_PWM_Start(&htim1,TIM_CHANNEL_1);
  HAL_TIM_PWM_Start(&htim1,TIM_CHANNEL_2);
  HAL_TIM_PWM_Start(&htim1,TIM_CHANNEL_3);
```

Con los siguientes parámetros característicos:

Prescaler: 769

Periodo: 255

2. El siguiente temporizador será utilizado para el ultrasonido.

3.4. FUNCION LDR, CONVERTIDOR ADC

Para la estancia del jardín tendremos un LDR el cual simulará cuando se haga de noche con el encendido de Led´s, éste también contara con un botón para su encendido manual.

Al ser una variación de la tensión es necesario el convertidor el cual le realizaremos con DMA, utilizaremos un I2C1 de manera circular.

La función utilizada de forma global es: **void ctrl_ldr(void)**

Donde dentro se encuentra la función necesaria para ADC con DMA:

HAL_ADC_Start_DMA(&hadc1, &buffer, 1);

```
//variables ADC
uint32_t adc_val;
uint32_t buffer;
```

3.5. FUNCION RGB

Para la realización de esta señal utilizaremos un PWM (Modulación Por Ancho de Banda), la cual es una técnica para simular una salida analógica con una entrada digital. La señal de onda cuadrada tendrá un ciclo de trabajo del 100%, con un periodo de 255, tendremos un reloj de 100MHz el cual es el reloj interno de la placa.

La asignación de los distintos colores será a través de la función:

void set_rgb(uint8_t red, uint8_t blue, uint8_t green)

La función que asignara los colores según los pulsos de PWM será:

void ctrl_rgb(void)

3.6. INTERRUPCIÓN

Llevamos a cabo tres interrupciones:

1. La cual tras pulsar el botón PA0 se produce el apagado de las luces, como simulación de que han saltado los fusibles, donde se indicará con el encendido del LED rojo de la placa.
2. Con esta realizaremos la selección del **modo manual**, el cual será indicado con el LED azul de la placa.
3. Y por último esta será el **modo automático**, indicado con LED naranja.

```
void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin){

if (GPIO_Pin==GPIO_PIN_0){
  HAL_GPIO_WritePin(GPIOD, GPIO_PIN_14,1); //se
  off_casa();
  espera(2000);
  HAL_GPIO_WritePin(GPIOD, GPIO_PIN_14,0);
}
if (GPIO_Pin==GPIO_PIN_2){
  ModeSelect(1,1,0); //Seleccion Modo Manual
                     //Encendemos el led azul en
}
if (GPIO_Pin==GPIO_PIN_12){
  ModeSelect(2,0,1); //Seleccion Modo Automatic
                     //Encendemos el led naranja
}
```

4. DISEÑO ESTRUCTURA Y CONEXIÓN

A continuación, presentamos unas imágenes del diseño realizado para la estructura de la casa, así como del conexionado de los cables.

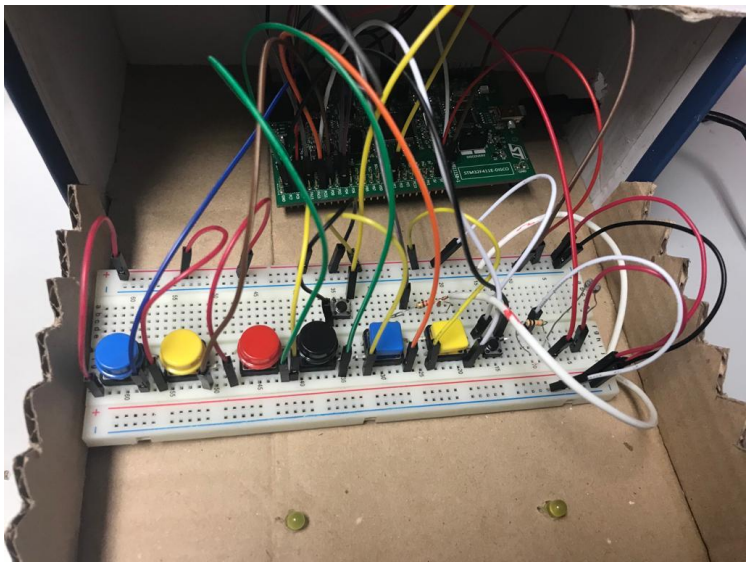
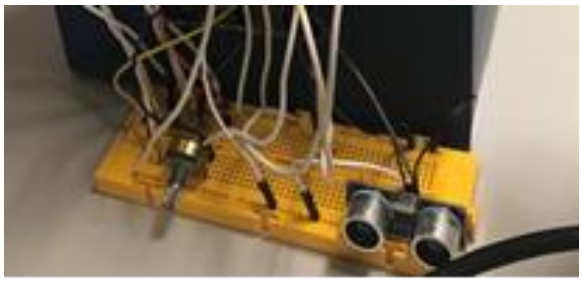


Imagen 1.Conexión botones y LDR



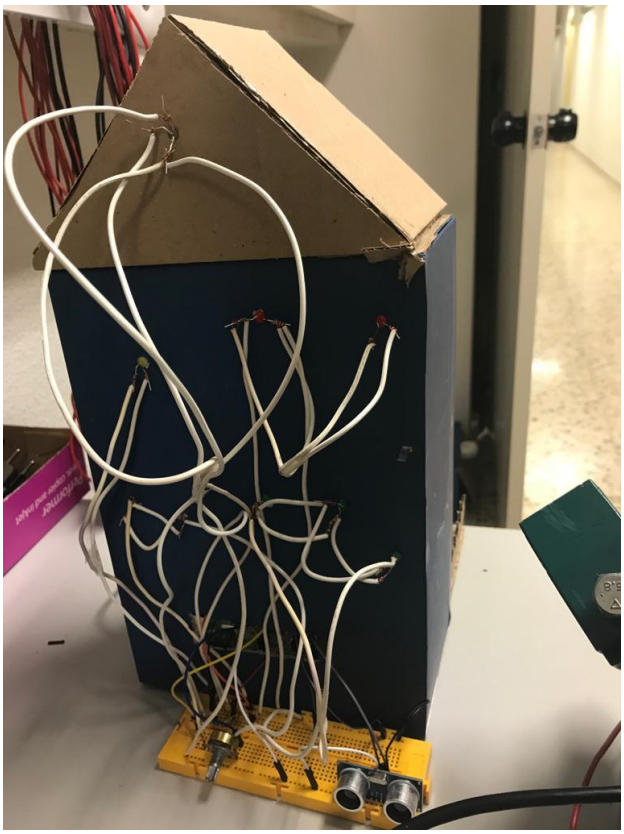


Imagen 2.Conexión LED´s y Sensor ultrasónico.

Imagen 3. Estructura de la vivienda.

5. ENLACE REPOSITORIO GITHUB

https://github.com/edwarttorres/Trabajo-SED

Debido a que Vivado en las simulaciones va generando archivos de mas de 100MB que no se pueden subir a Github debido a las restricciones. Para poder solucionar este problema hemos marcado los archivos .wdb como git lfs utilizando el Bash de Git.

El comando utilizado ha sido el siguiente:

```
$ git lfs track "*.wdb"
```