

A dark blue vertical bar on the left side of the page. A blue arrow points to the right from the bar, containing the date.

19-1-2022

# CONTROL DE LUCES DE UNA HABITACIÓN CON PULSADORES, SENSOR DE PRESENCIA Y SENSOR DE TEMPERATURA

Several thin, curved lines in dark blue and light grey originate from the bottom left and curve upwards and to the right.

Luis

Javier Martín Sanz 54728  
Luis Pérez González 54793  
Luis Torres del Nuevo 54887

# Índice

1. Introducción .....	2
1.1. Explicación y funcionamiento del control de luces. ....	2
2. Material utilizado .....	2
3. Configuración pines.....	2
3.1 Reloj.....	3
3.2 TIM 1 (HC-SR04) .....	3
3.3 TIM 3 y TIM4 (led verde) .....	4
3.4 TIM 2 (led rojo).....	4
4. Funcionamiento Código. ....	5
4.1. Función Antirrebotes para los pulsadores. ....	5
4.2. Función para lectura del ultrasonido. ....	5
4.3 Interrupciones del ultrasonido. ....	6
4.4 Lectura de la temperatura.....	6
5. Fotos del Proyecto.....	7
6. Conclusiones.....	8
7. Bibliografía .....	8

# 1. Introducción

## 1.1. Explicación y funcionamiento del control de luces.

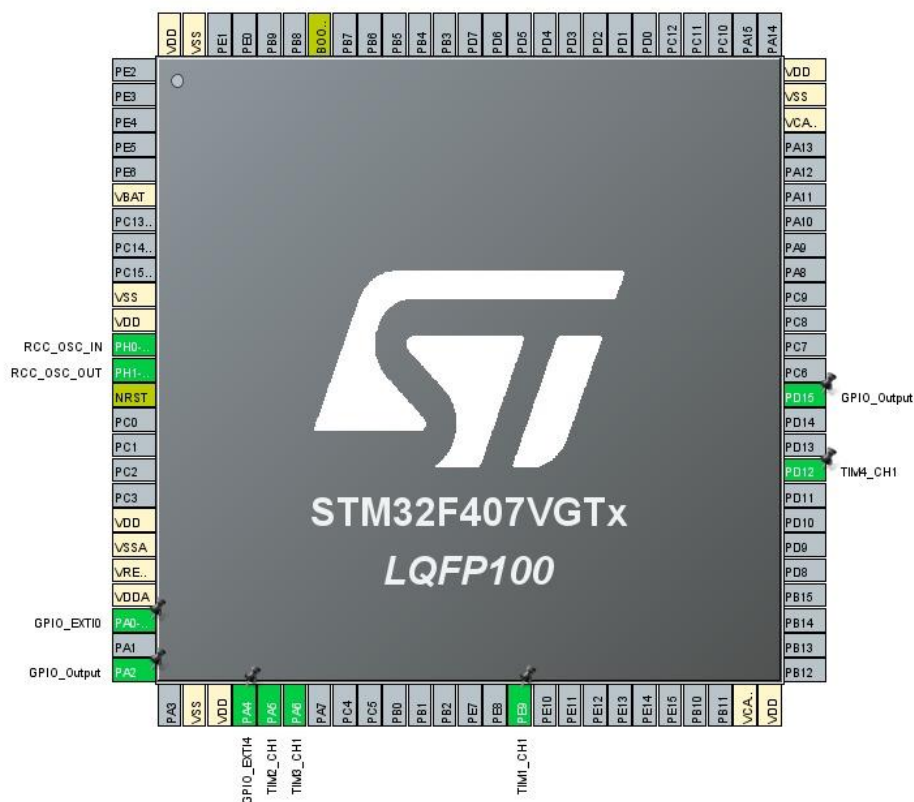
Nuestro proyecto en la STM32 consiste en:

- Encender la luz verde durante 15 segundos cuando se detecta que alguien ha pasado a la habitación.
- Encender la luz roja, de emergencia, cuando se pasan esos 15 segundos.
- Con el pulsador, user, encender la luz verde de la habitación.
- Con un pulsador simulado con los pines, apagar la verde, y encender la luz roja de emergencia.
- Si la temperatura es menor de 30 grados se encenderá en la habitación una luz azul.

## 2. Material utilizado

- Leds rojo, verde y azul
- Sensor HC-SR04.
- Procesador de computadora STM32.
- Protoboard y cables.

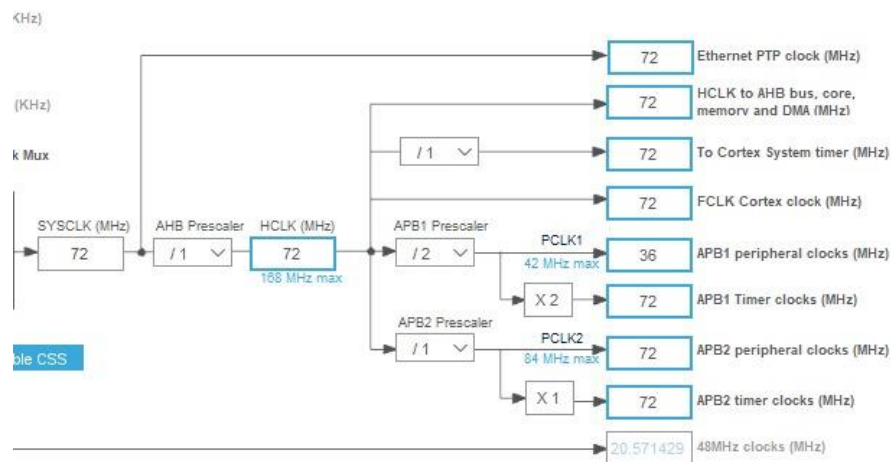
## 3. Configuración pines.



También se ha utilizado el canal interno de sensor de temperatura mediante la conversión digital/analógico.

### 3.1 Reloj.

Se ha utilizado un reloj de 72 MHz.

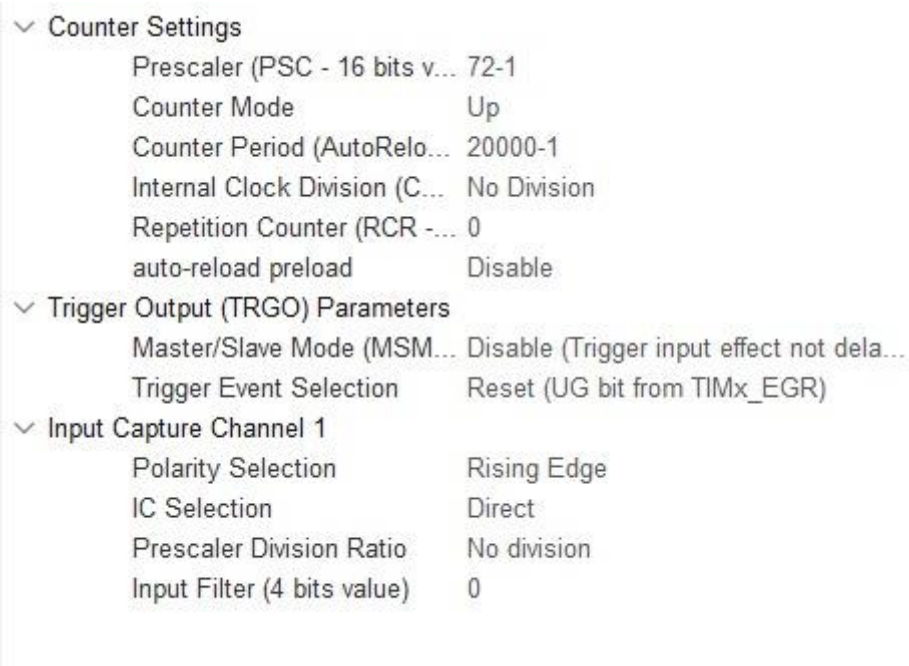


### 3.2 TIM 1 (HC-SR04)

Se ha usado un Input Capture direct Mode.

El timer se configura de esta forma, ya que el sensor de ultrasonido funciona a 50Hz con un periodo entre 1-2ms.

Como el reloj es de 72MHz y con un preescaler de 72 y un period de 20000 nos salen los 50 Hz



### 3.3 TIM 3 y TIM4 (led verde)

Se ha usado un temporizador PWM en el Channel 1 configurado de la siguiente manera.

▼ Counter Settings	
Prescaler (PSC - 16 bits v...	72-1
Counter Mode	Up
Counter Period (AutoRelo...	3922
Internal Clock Division (C...	No Division
auto-reload preload	Disable
▼ Trigger Output (TRGO) Parameters	
Master/Slave Mode (MSM...	Disable (Trigger input effect not dela...
Trigger Event Selection	Reset (UG bit from TIMx_EGR)
▼ PWM Generation Channel 1	
Mode	PWM mode 1
Pulse (32 bits value)	0
Output compare preload	Enable
Fast Mode	Disable
CH Polarity	High

### 3.4 TIM 2 (led rojo)

Se ha usado un teporizador PWM en el Channel 1 configurado de la siguiente manera, análogo al TIM 3 y 4.

▼ Counter Settings	
Prescaler (PSC - 16 bits v...	72-1
Counter Mode	Up
Counter Period (AutoRelo...	3922
Internal Clock Division (C...	No Division
auto-reload preload	Disable
▼ Trigger Output (TRGO) Parameters	
Master/Slave Mode (MSM...	Disable (Trigger input effect not dela...
Trigger Event Selection	Reset (UG bit from TIMx_EGR)
▼ PWM Generation Channel 1	
Mode	PWM mode 1
Pulse (16 bits value)	0
Output compare preload	Enable
Fast Mode	Disable
CH Polarity	High

## 4. Funcionamiento Código.

### 4.1. Función Antirrebotes para los pulsadores.

Esta función mediante un muestreo nos dice si el pulsador se ha pulsado o no.

```
int debouncer(volatile int* button, GPIO_TypeDef* GPIO_port, uint16_t GPIO_number){  
    static uint8_t cb=0; //variable para llevar la cuenta del boton  
    static int cuenta=0;  
  
    if (*button==1){  
        if (cb==0) {  
            cuenta=HAL_GetTick();  
            cb++;  
        }  
        if (HAL_GetTick()-cuenta>=20){  
            cuenta=HAL_GetTick();  
            if (HAL_GPIO_ReadPin(GPIO_port, GPIO_number)!=1){  
                cb=1;  
            }  
            else{  
                cb++;  
            }  
            if (cb==3){  
                cb=0;  
                *button=0;  
                return 1;  
            }  
        }  
    }  
    return 0;  
}
```

### 4.2. Función para lectura del ultrasonido.

Esta función hace que se lea el ultrasonidos

```
void HCSR04_Read (void) //Función de lectura del ultrasonidos  
{  
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_2, GPIO_PIN_SET); //Se envia pulso a la patilla trig del ultrasonidos  
    delay(5);  
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_2, GPIO_PIN_RESET);  
  
    __HAL_TIM_ENABLE_IT(&htim1, TIM_IT_CC1);  
}
```

### 4.3 Interrupciones del ultrasonido.

Esta función nos da la distancia que hay desde el objeto al ultrasonidos.

```
void HAL_TIM_IC_CaptureCallback(TIM_HandleTypeDef *htim) {
    if (htim->Channel == HAL_TIM_ACTIVE_CHANNEL_1)
    {
        if (Is_First_Captured==0)
        {
            IC_Val1 = HAL_TIM_ReadCapturedValue(htim, TIM_CHANNEL_1); //lee el primer valor
            Is_First_Captured = 1; // selecciona la primera captura
            __HAL_TIM_SET_CAPTUREPOLARITY(htim, TIM_CHANNEL_1, TIM_INPUTCHANNELPOLARITY_FALLING);
        }

        else if (Is_First_Captured==1) // si ha leído el primer valor
        {
            IC_Val2 = HAL_TIM_ReadCapturedValue(htim, TIM_CHANNEL_1); // lee el segundo valor
            __HAL_TIM_SET_COUNTER(htim, 0); // reset the counter

            if (IC_Val2 > IC_Val1)
            {
                Difference = IC_Val2-IC_Val1;
            }

            else if (IC_Val1 > IC_Val2)
            {
                Difference = (0xffff - IC_Val1) + IC_Val2;
            }

            Distance = Difference * .034/2;//conversion a centimetros
            Is_First_Captured = 0;
            __HAL_TIM_SET_CAPTUREPOLARITY(htim, TIM_CHANNEL_1, TIM_INPUTCHANNELPOLARITY_RISING);
            __HAL_TIM_DISABLE_IT(&htim1, TIM_IT_CC1);
        }
    }
}
```

### 4.4 Lectura de la temperatura.

Hacemos la lectura de la temperatura por Polling, como nos mide la temperatura interna de la placa, y nos da muy alto, se ha hecho una interpolación para ver la temperatura que hay en el exterior. Por eso hay 1650.0/adcvalue.

```
if(HAL_ADC_PollForConversion(&hadc1, 100) == HAL_OK)
{
    adcvalue=HAL_ADC_GetValue(&hadc1);
    temp=1650.0/((float)adcvalue);
}
if(temp<30)
    HAL_GPIO_WritePin(GPIOD, GPIO_PIN_15,1);
else
    HAL_GPIO_WritePin(GPIOD, GPIO_PIN_15,0);
```

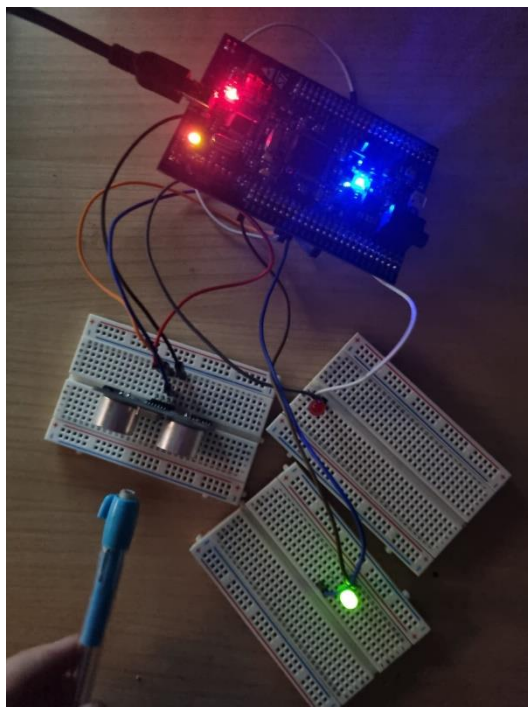


## 5. Fotos del Proyecto.

- FOTO AL SENSOR DE TEMPERATURA

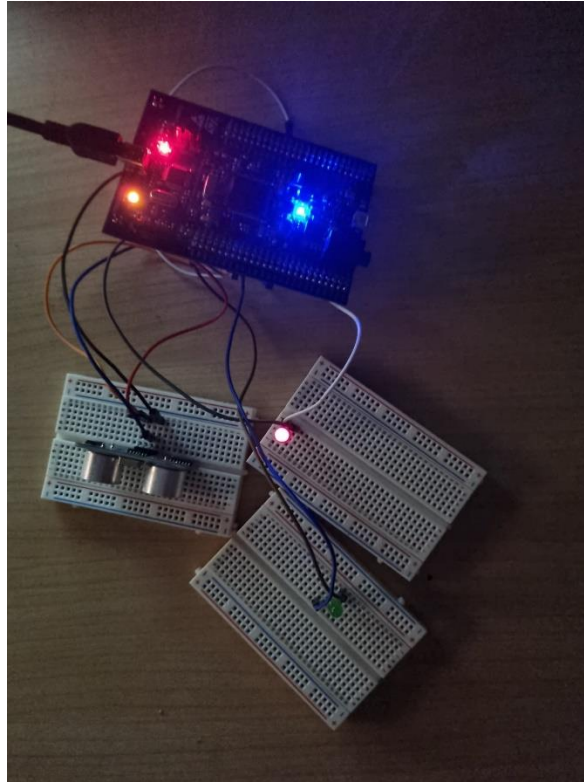


- FOTO AL LED VERDE CUANDO SE DETECTA PRESENCIA EN 15 mm





- FOTO AL LED ROJO PASADOS LOS 15 SEGUNDOS



## 6. Conclusiones

Este trabajo ha servido de ayuda para incentivarnos en el aprendizaje de microprocesadores, ya que hemos aprendido acerca de estos componentes, desconocidos anteriormente.

Nos hemos dado cuenta que están más en nuestro día a día de lo que pensábamos.

## 7. Bibliografía

<https://controllerstech.com/hcsr04-ultrasonic-sensor-and-stm32/>

<https://respuestas.me/q/lectura-del-sensor-de-temperatura-interno-stm-32-61959599003>

<https://bitbucket.org/abrunete/sistemas-electronicos-digitales/src/master/>

