



PROYECTO FIN DE CURSO

RADAR ULTRASÓNICO

Décima, Enrique Emanuel
Castro, Oscar Martin
Ortiz, Nicolás Agustín

Profesor : Ing. Rubén Darío Mansilla



Descripción del Proyecto

El proyecto consiste en el desarrollo de un Radar Ultrasónico, diseñado para detectar objetos dentro de un rango de distancia determinado y alertar al usuario mediante mensajes a través de una pantalla LCD y un dispositivo conectado vía Bluetooth.

Este sistema se basa en la medición de distancias utilizando un sensor ultrasónico HC-SR04, montado sobre un servomotor, que le permite escanear su entorno en un rango de 180°.





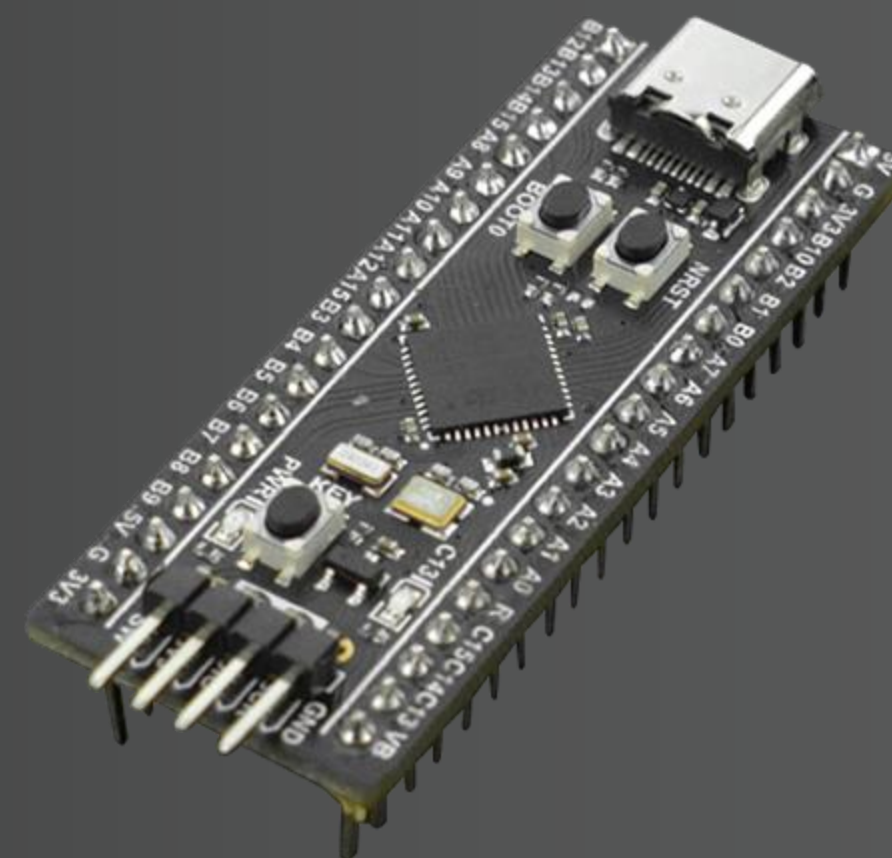
El hardware del Dispositivo y su interconexion fisica

Especificaciones

- *Chip principal: STM32F401CCU6*
- *Núcleo: Cortex-M4*
- *Memoria Flash: 256K*
- *SRAM: 96K*
- *Max. Velocidad de reloj: 84 MHz*
- *USB tipo C*
- *Dimensiones: 20.78mm x 52.81mm*
- *Peso: 4g*

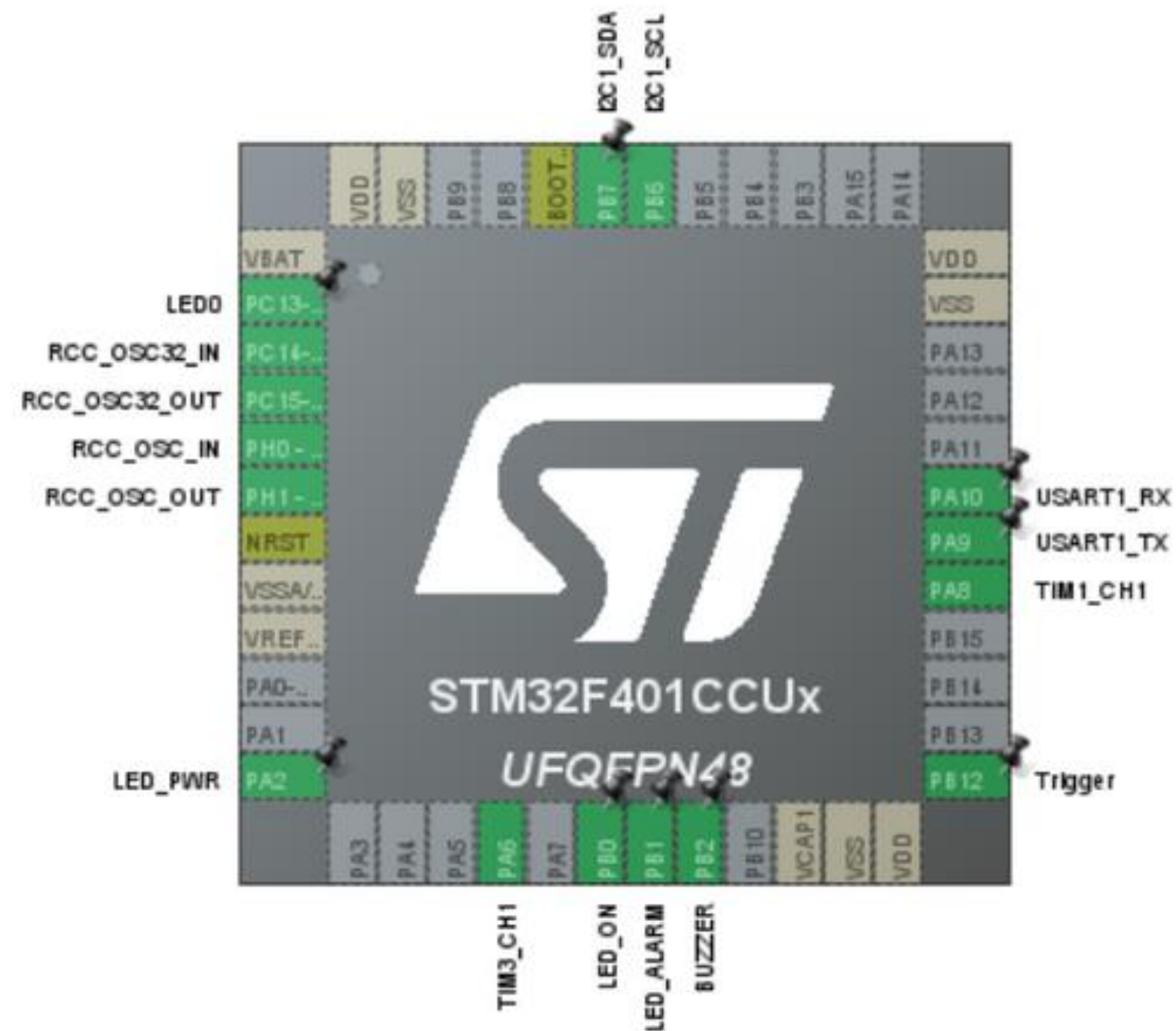
Periféricos

- *32 x GPIOs*
- *1 ADC de 12 bits de Resolución*
- *11 Timers*
- *3 x I2C*
- *3 x USARTs*
- *4 x SPI*
- *1 x SDIO*





El hardware del Dispositivo y su interconexion fisica

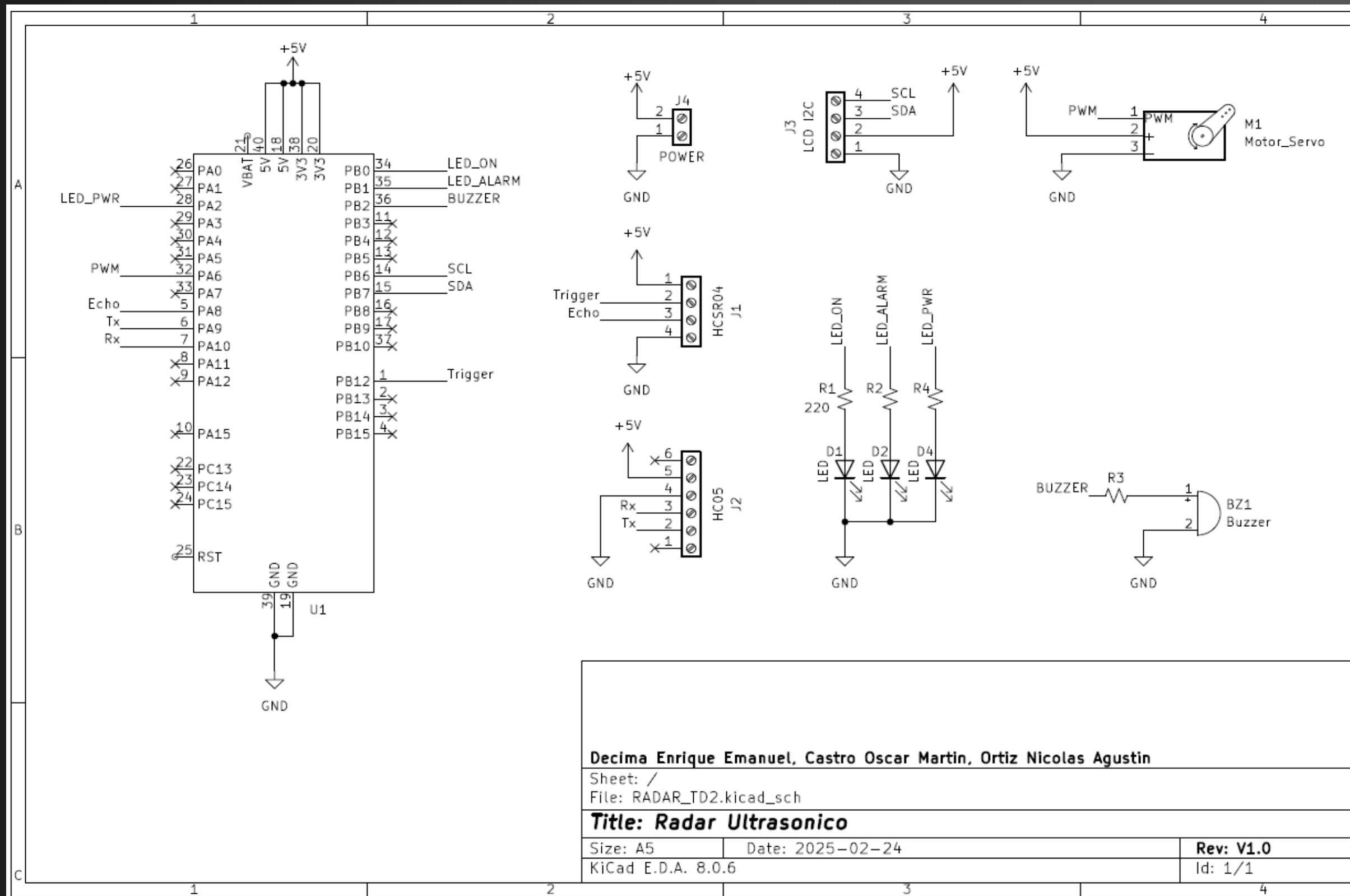


Distribución de Pines (Pinout)

Pin	Etiqueta	Descripcion
PC13	LED0	LED integrado a la placa
PA2	LED_PWR	LED Indicador de Alimentacion
PB0	LED_ON	Indicador de Sistema Activado
PB1	LED_ALARM	LED indicador de alarma
PB2	BUZZER	Buzzer de Alarma
PA6	TIM3_CH1	Servomotor PWM
PB12	Trigger	HCSR04 Trigger Pin
PA8	TIM1_CH1	HCSR04 Echo Pin
PA9	USART_TX	HC-05 Rx
PA10	USART_RX	HC-05 Tx
PB6	I2C_SCL	PCF8574 para Comunicación con LCD
PB7	I2C_SDA	PCF8574 para Comunicación con LCD



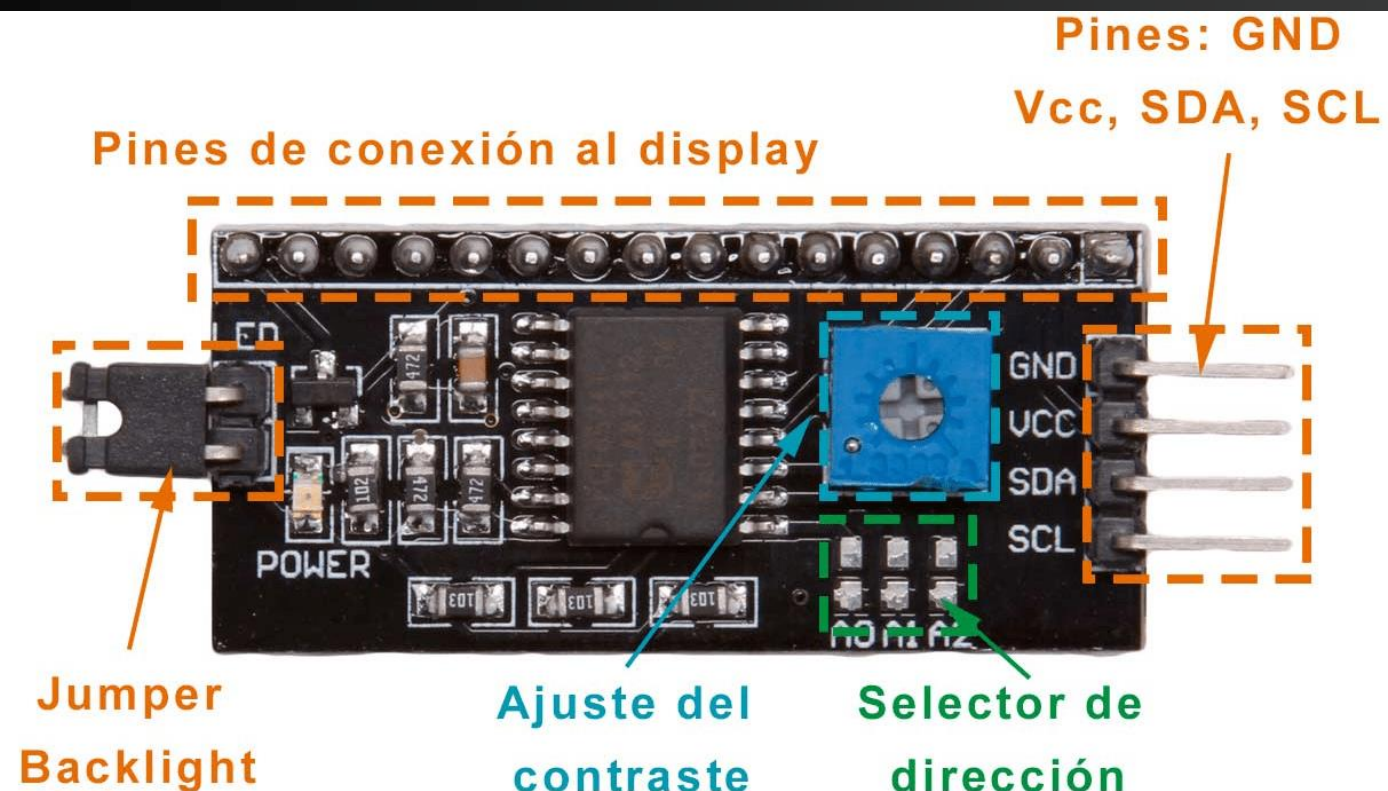
El hardware del Dispositivo y su interconexion fisica



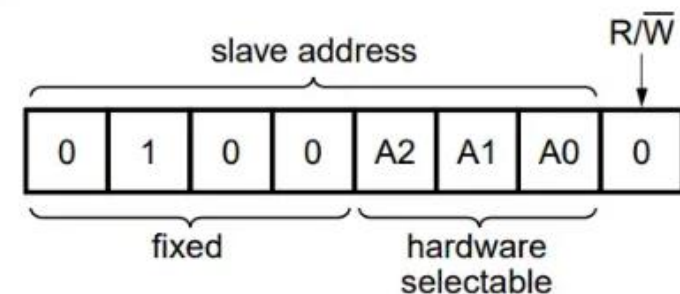


Consideraciones comunicación

I2C – PCF8574



Pin connectivity			Address of PCF8574								Address byte value		7-bit hexadecimal address without R/W
A2	A1	A0	A6	A5	A4	A3	A2	A1	A0	R/W	Write	Read	
V _{SS}	V _{SS}	V _{SS}	0	1	0	0	0	0	0	-	40h	41h	20h
V _{SS}	V _{SS}	V _{DD}	0	1	0	0	0	0	1	-	42h	43h	21h
V _{SS}	V _{DD}	V _{SS}	0	1	0	0	0	1	0	-	44h	45h	22h
V _{SS}	V _{DD}	V _{DD}	0	1	0	0	0	1	1	-	46h	47h	23h
V _{DD}	V _{SS}	V _{SS}	0	1	0	0	1	0	0	-	48h	49h	24h
V _{DD}	V _{SS}	V _{DD}	0	1	0	0	1	0	1	-	4Ah	4Bh	25h
V _{DD}	V _{DD}	V _{SS}	0	1	0	0	1	1	0	-	4Ch	4Dh	26h
V _{DD}	V _{DD}	V _{DD}	0	1	0	0	1	1	1	-	4Eh	4Fh	27h

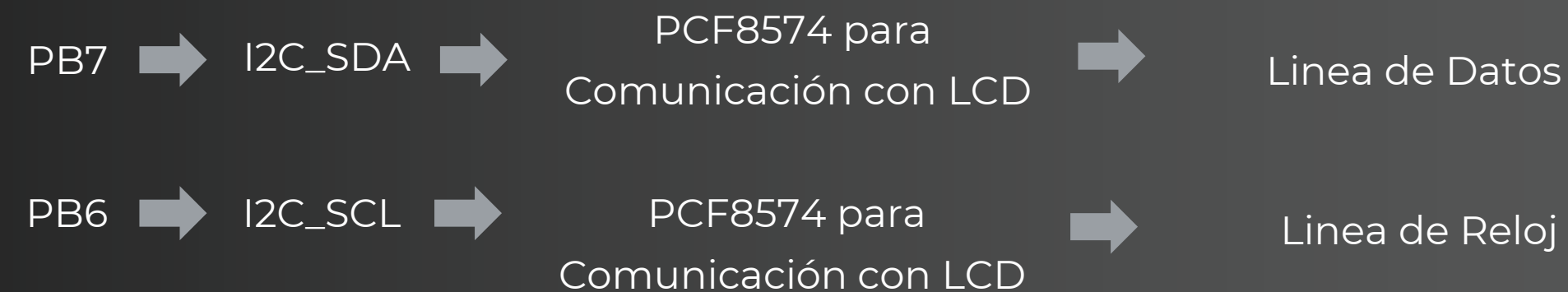
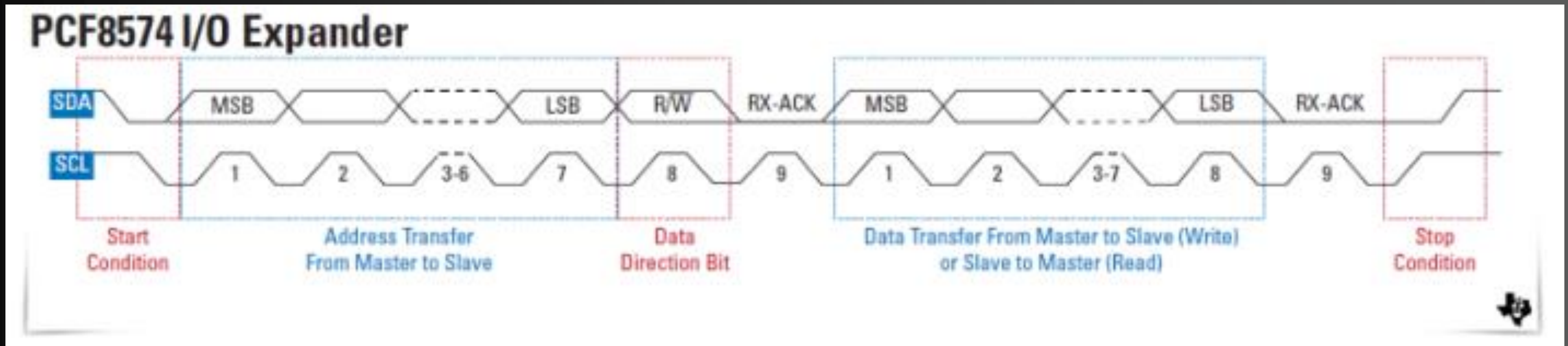


002aad628

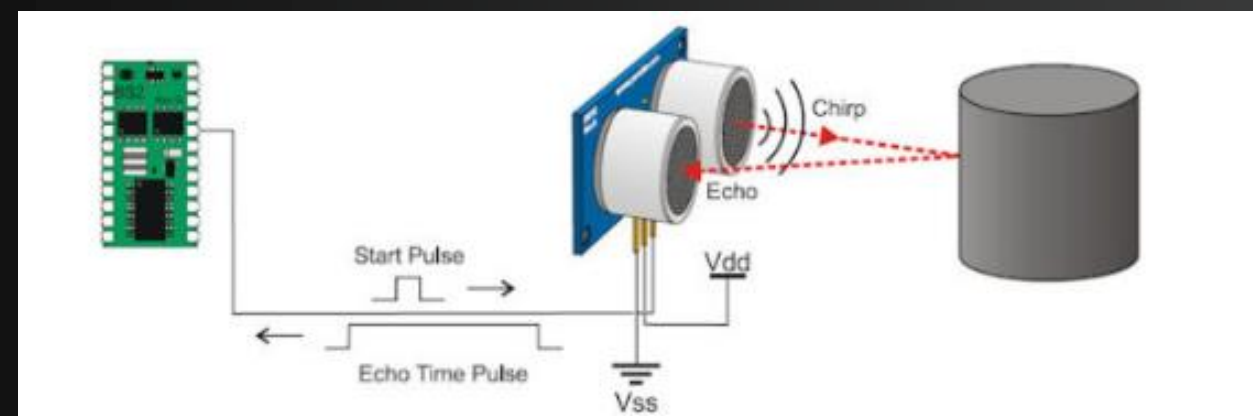
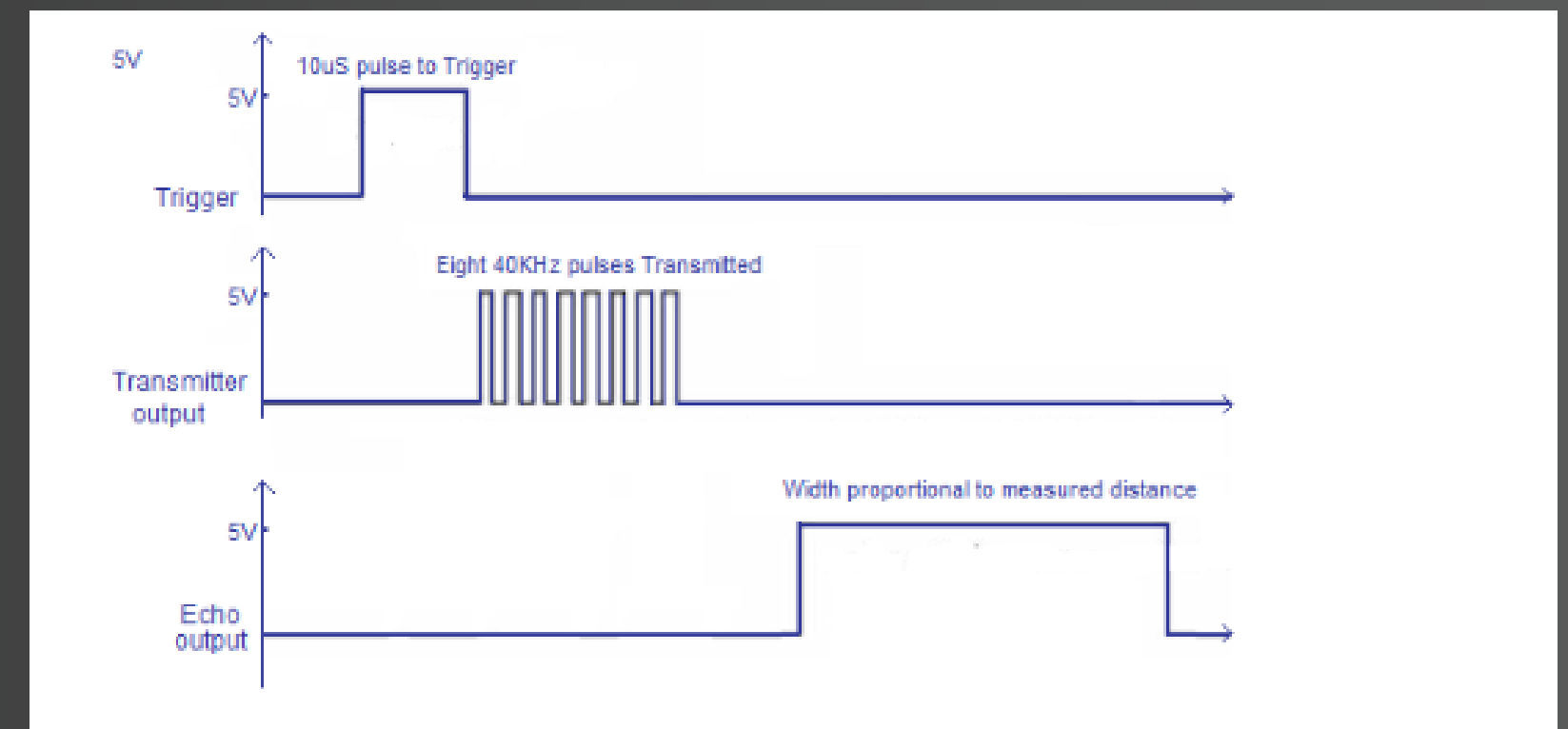
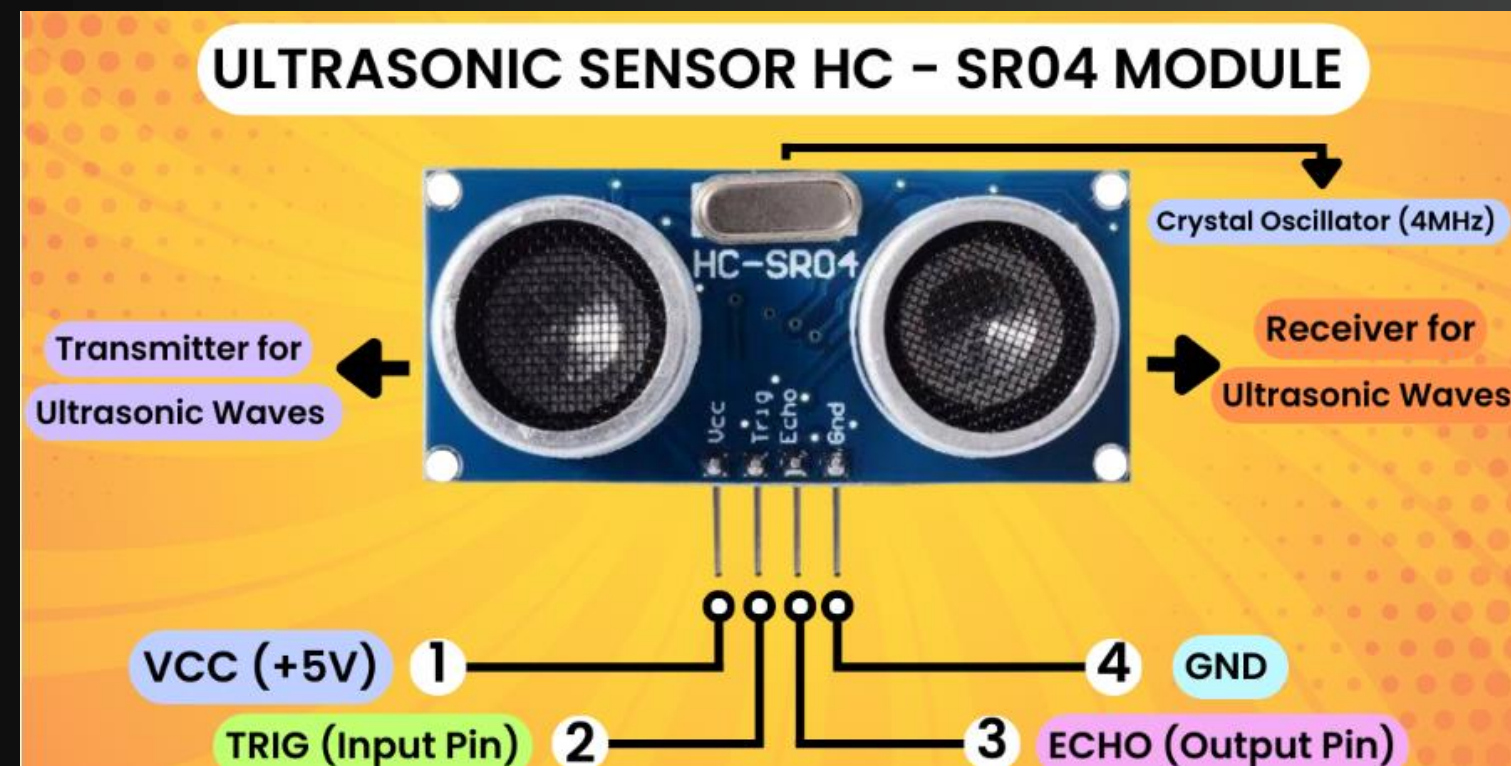
a. PCF8574

Consideraciones comunicación

I2C – PCF8574



Consideraciones comunicación HC - SR04



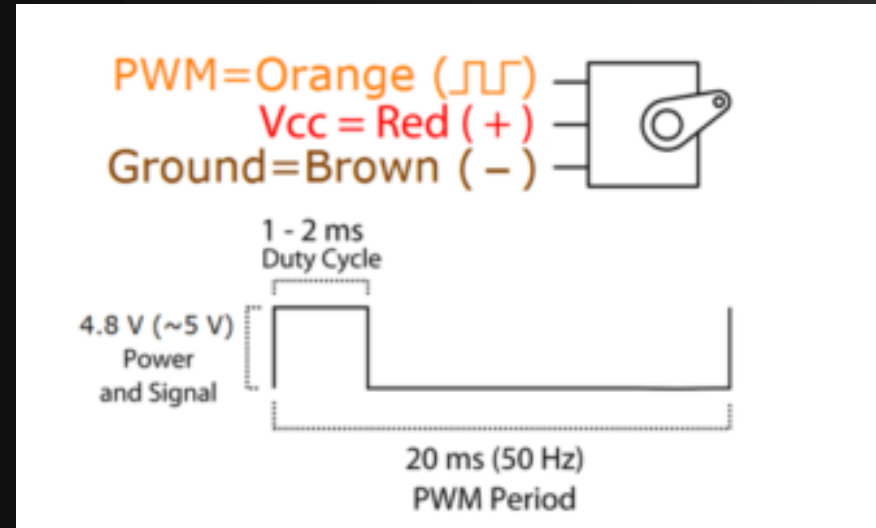
Puede detectar objetos desde 2 centímetros hasta 400 cm, con una precisión de 3 mm y un angulo de cobertura de medición de 15 grados.

Pinout & Configuration

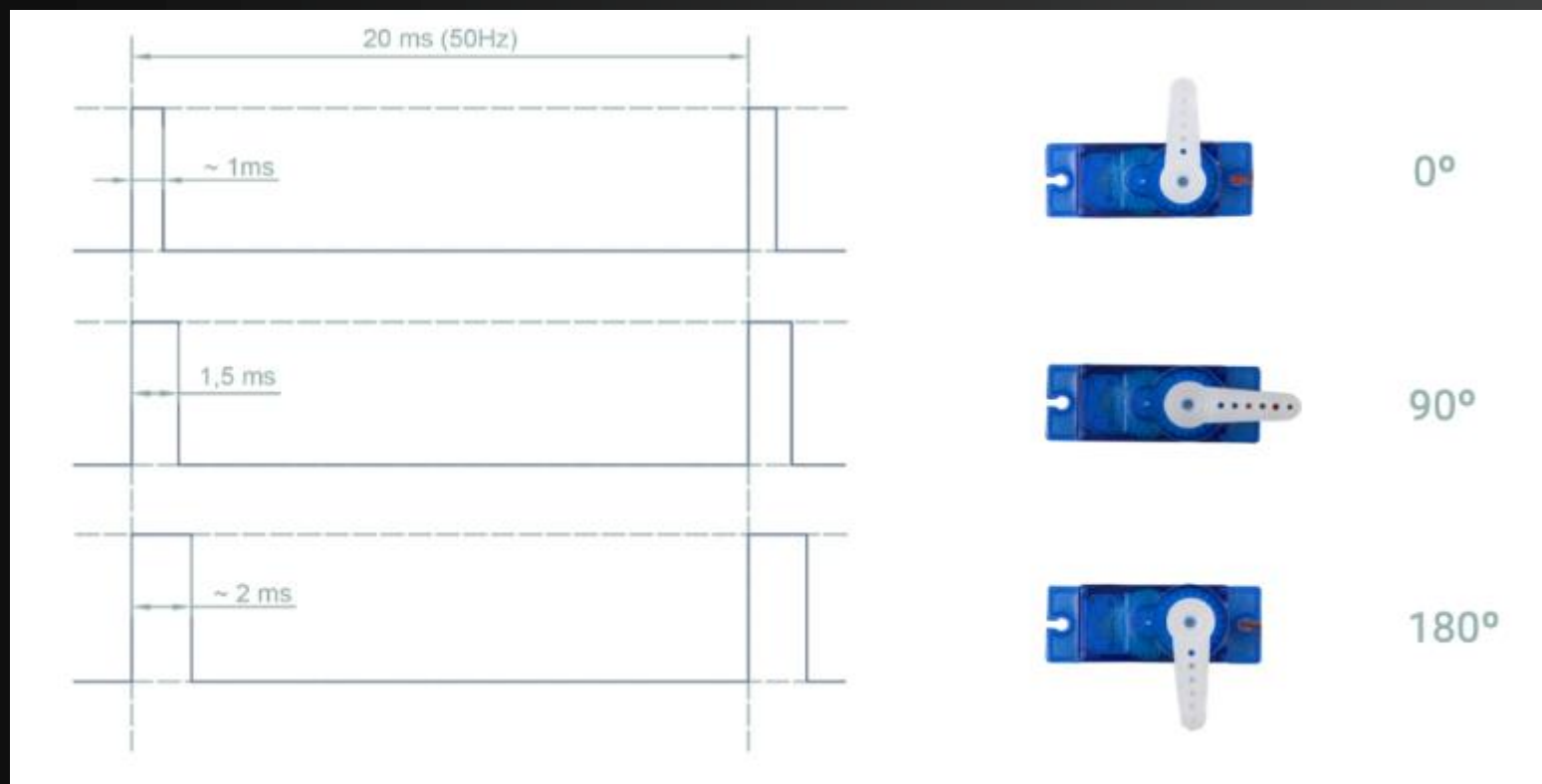
PB12 → Trigger → HCSR04 Trigger Pin

PA8 → TIM1_CH1 → HCSR04 Echo Pin

Consideraciones comunicación Servo SG90



POSICIÓN DEL SERVO	PULSO
0° (todo a la izquierda)	~ 1 ms
90° (en medio)	1,5 ms
180° (todo a la derecha)	~ 2 ms



$$\text{Timer Tick Frequency} = \frac{\text{Timer input clock}}{\text{Prescaler}+1}$$

$$\text{Counter Frequency} = \text{Frequency Required} * \text{Pwm Resolution(Steps)}$$

$$\text{Timer Prescaler Value} = (\text{Timer input clock} / \text{Counter frequency}) - 1$$

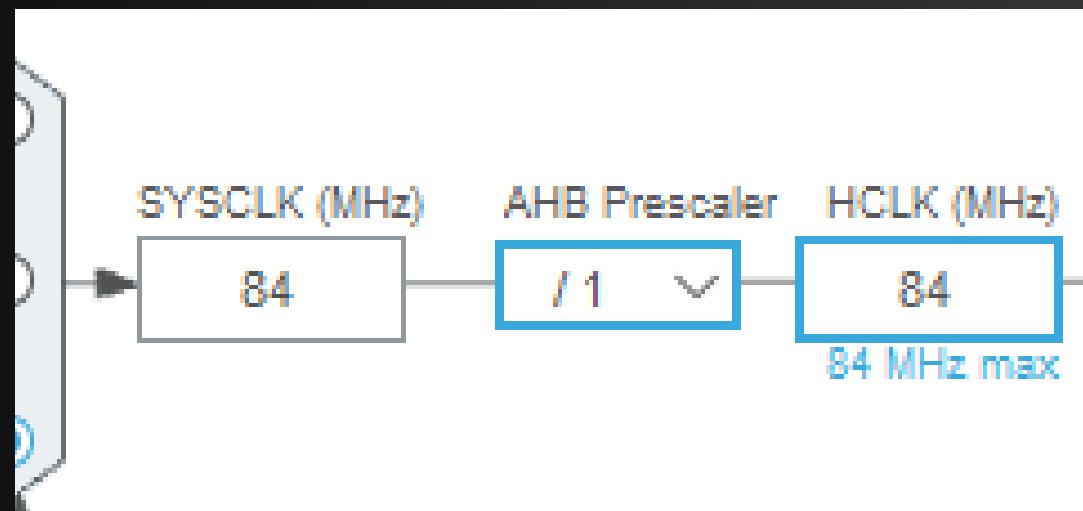
Pinout & Configuration

PA6 → TIM3_CH1 → Servomotor PWM

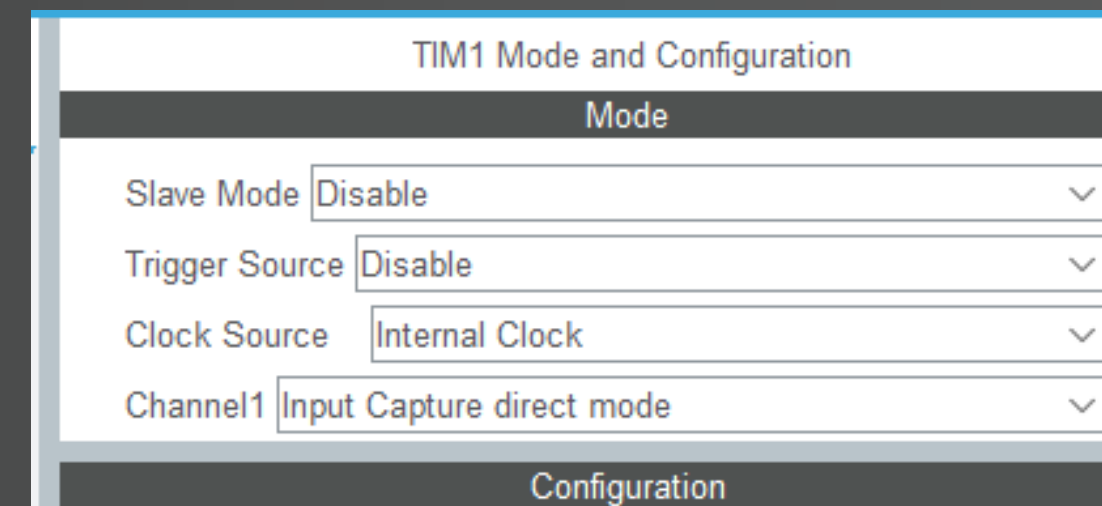
Consideraciones sobre el software

Configuracion del Modulo TIM 1

Configuracion del Reloj



Configuracion Principal del Modulo



Configuration window for the TIM1 module. The Mode section includes the following settings:

- Slave Mode: Disable
- Trigger Source: Disable
- Clock Source: Internal Clock
- Channel1: Input Capture direct mode

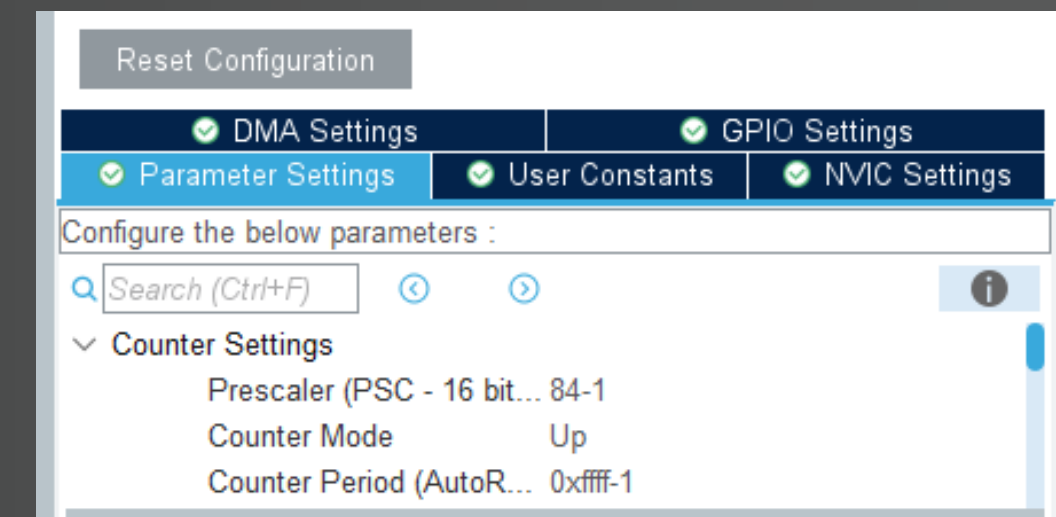
The Configuration section is also visible but not detailed in the image.

Configuracion de Interrupciones

Reset Configuration			
<input checked="" type="checkbox"/> DMA Settings	<input checked="" type="checkbox"/> GPIO Settings		
<input checked="" type="checkbox"/> Parameter Settings	<input checked="" type="checkbox"/> User Constants	<input checked="" type="checkbox"/> NVIC Settings	
NVIC Interrupt Table		Ena...	Sub F
TIM1 break interrupt and TIM9 global interrupt		<input type="checkbox"/>	0
TIM1 update interrupt and TIM10 global interr...		<input type="checkbox"/>	0
TIM1 trigger and commutation interrupts and ...		<input type="checkbox"/>	0
TIM1 capture compare interrupt		<input checked="" type="checkbox"/>	0

PB12 → Trigger → HCSR04 Trigger Pin

Parametros del Modulo



Parameter Settings window for the TIM1 module. The Counter Settings section includes the following parameters:

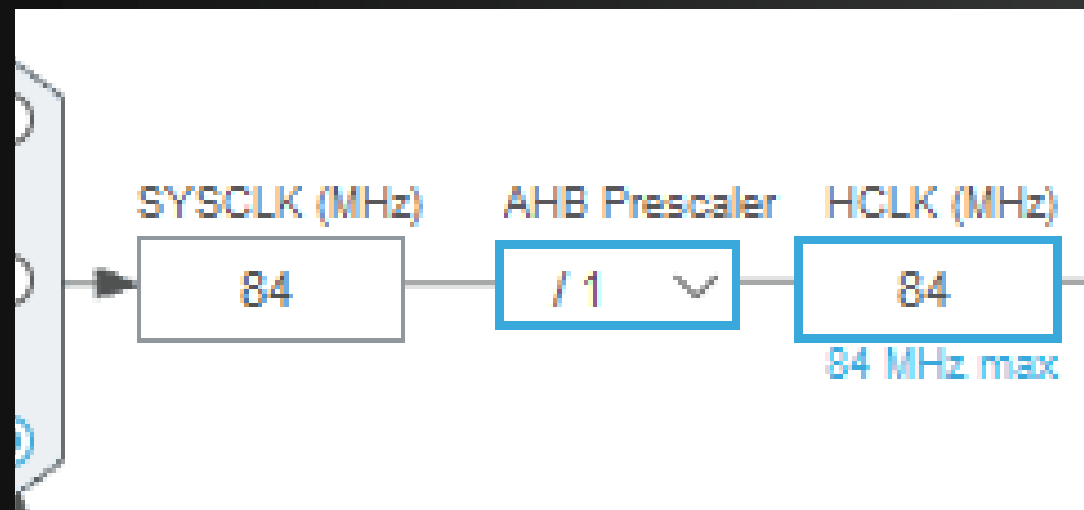
- Prescaler (PSC - 16 bit... 84-1)
- Counter Mode: Up
- Counter Period (AutoR... 0xffff-1)

PA8 → TIM1_CH1 → HCSR04 Echo Pin

Consideraciones sobre el software

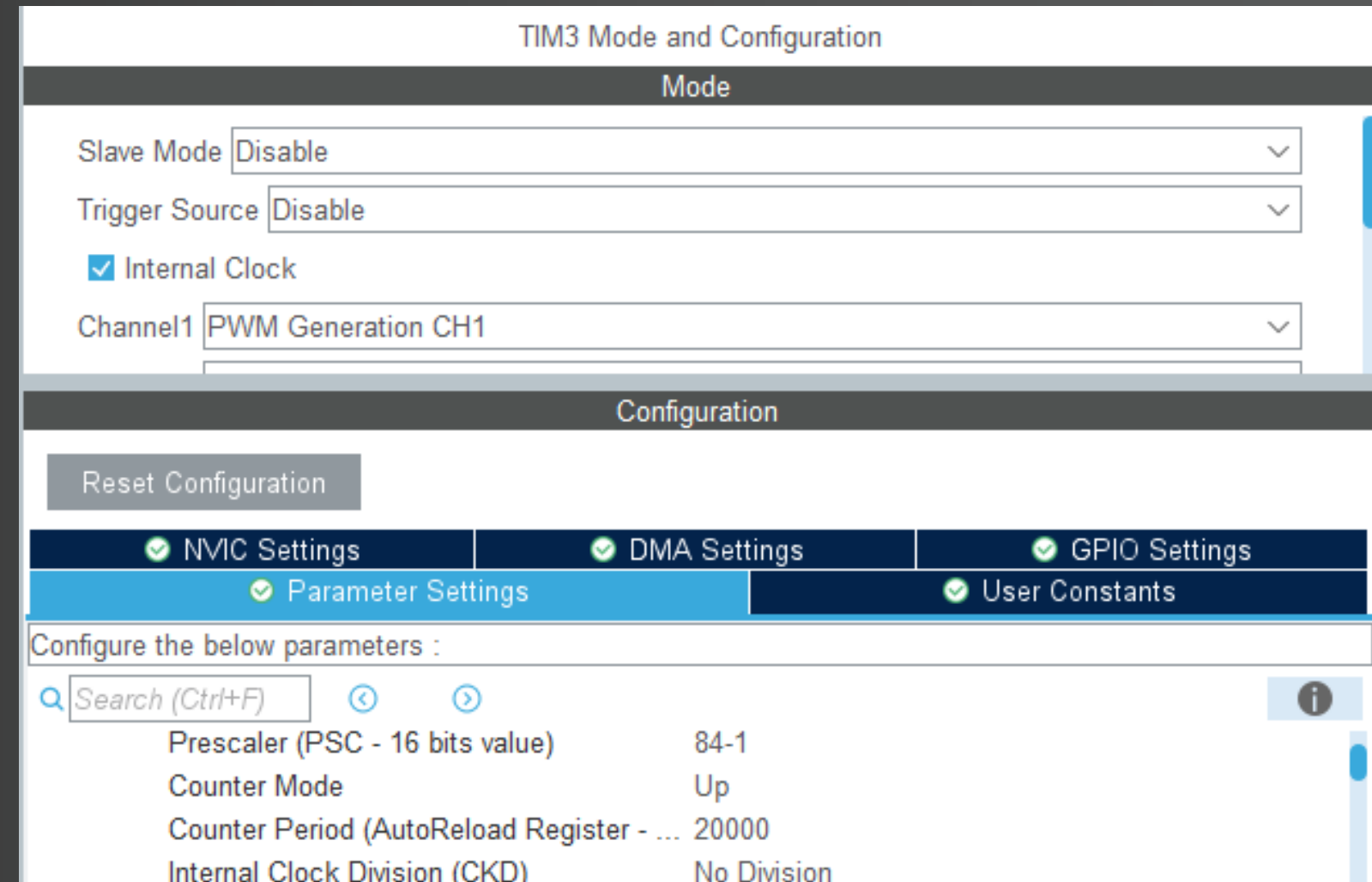
Configuracion del Modulo TIM 3

Configuracion del Reloj



PA6 → TIM3_CH1 → Servomotor PWM

Configuracion Principal de Modulo y Parametros



TIM3 Mode and Configuration

Mode

- Slave Mode: Disable
- Trigger Source: Disable
- ☒ Internal Clock
- Channel1: PWM Generation CH1

Configuration

Reset Configuration

- ☒ NVIC Settings
- ☒ DMA Settings
- ☒ GPIO Settings
- ☒ Parameter Settings
- ☒ User Constants

Configure the below parameters :

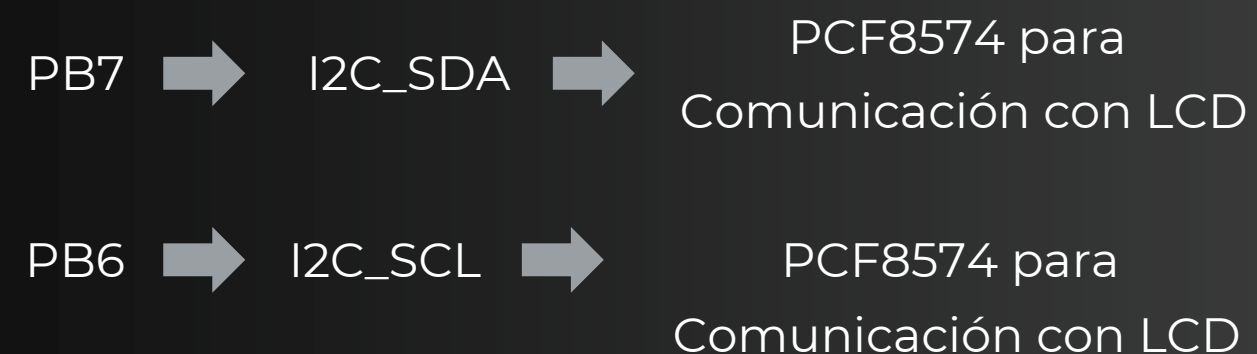
Search (Ctrl+F) [Navigation icons] [Info icon]

Prescaler (PSC - 16 bits value)	84-1
Counter Mode	Up
Counter Period (AutoReload Register - ...)	20000
Internal Clock Division (CKD)	No Division

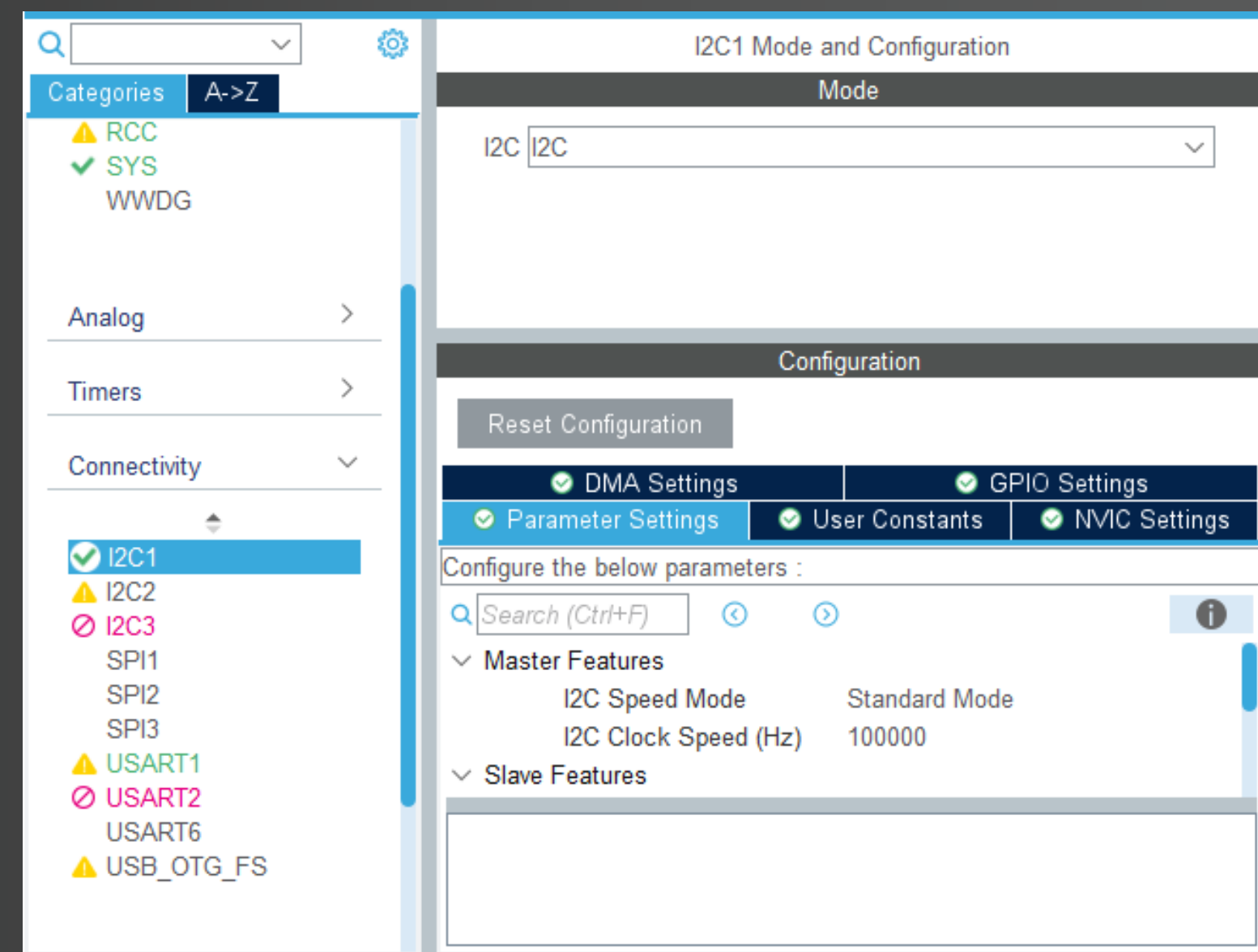


Consideraciones sobre el software

Configuración del Módulo I2C



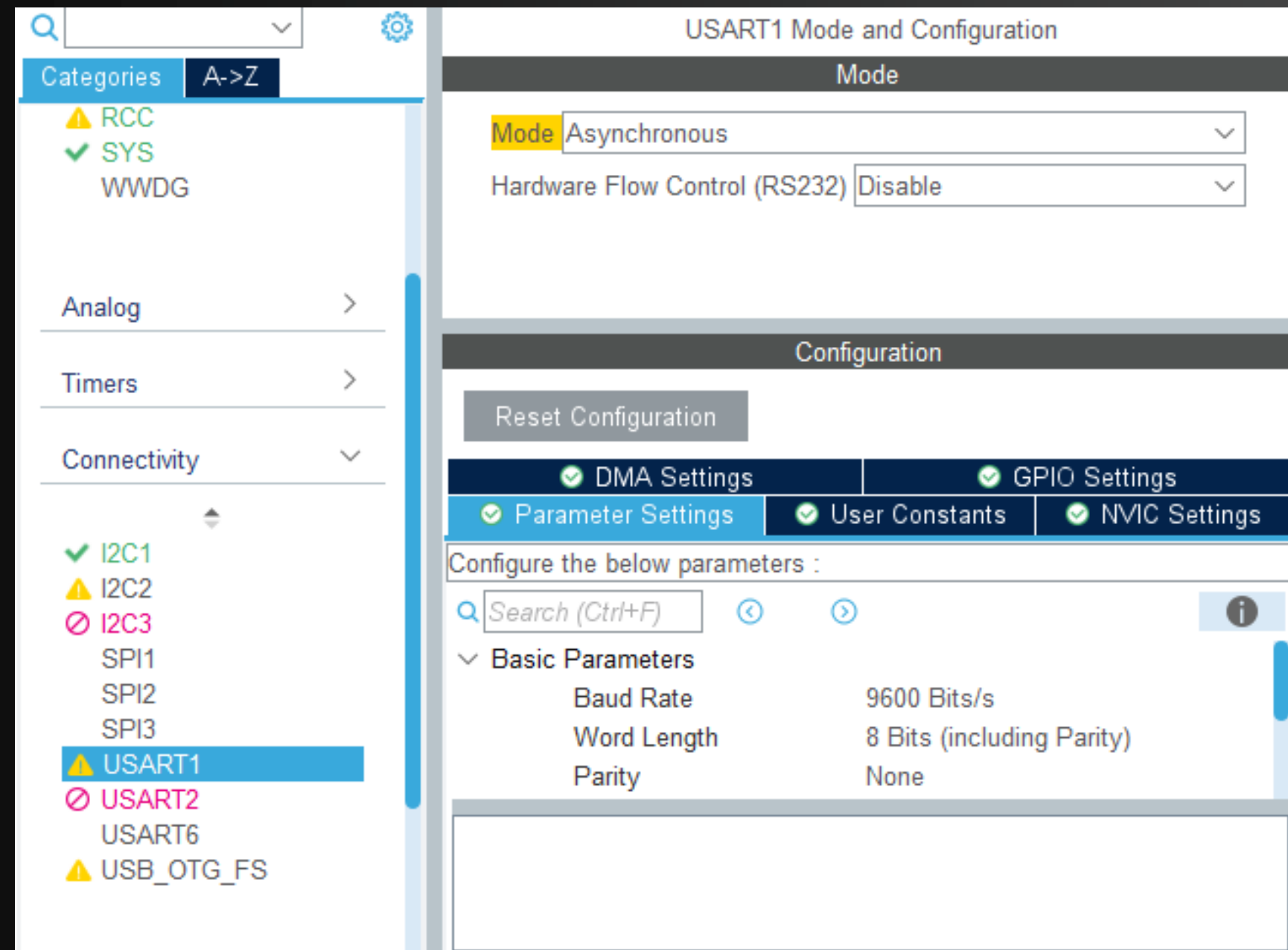
Configuración Principal del Módulo



Consideraciones sobre el software

Bluetooth HC-05

Configuración de Parametros del Modulo



PA9 → USART_TX → HC-05 Rx
PA10 → USART_RX → HC-05 Tx



Consideraciones sobre el software

Salidas Digitales

GPIO

Categories

A->Z

System Core

DMA

GPIO

IWDG

NVIC

RCC

SYS

WWDG

Analog

>

Timers

>

Connectivity

>

Multimedia

>

Computing

>

GPIO Mode and Configuration

Configuration

Group By Peripherals

GPIO

I2C

RCC

TIM

USART

Search Signals

Search (Ctrl+F)

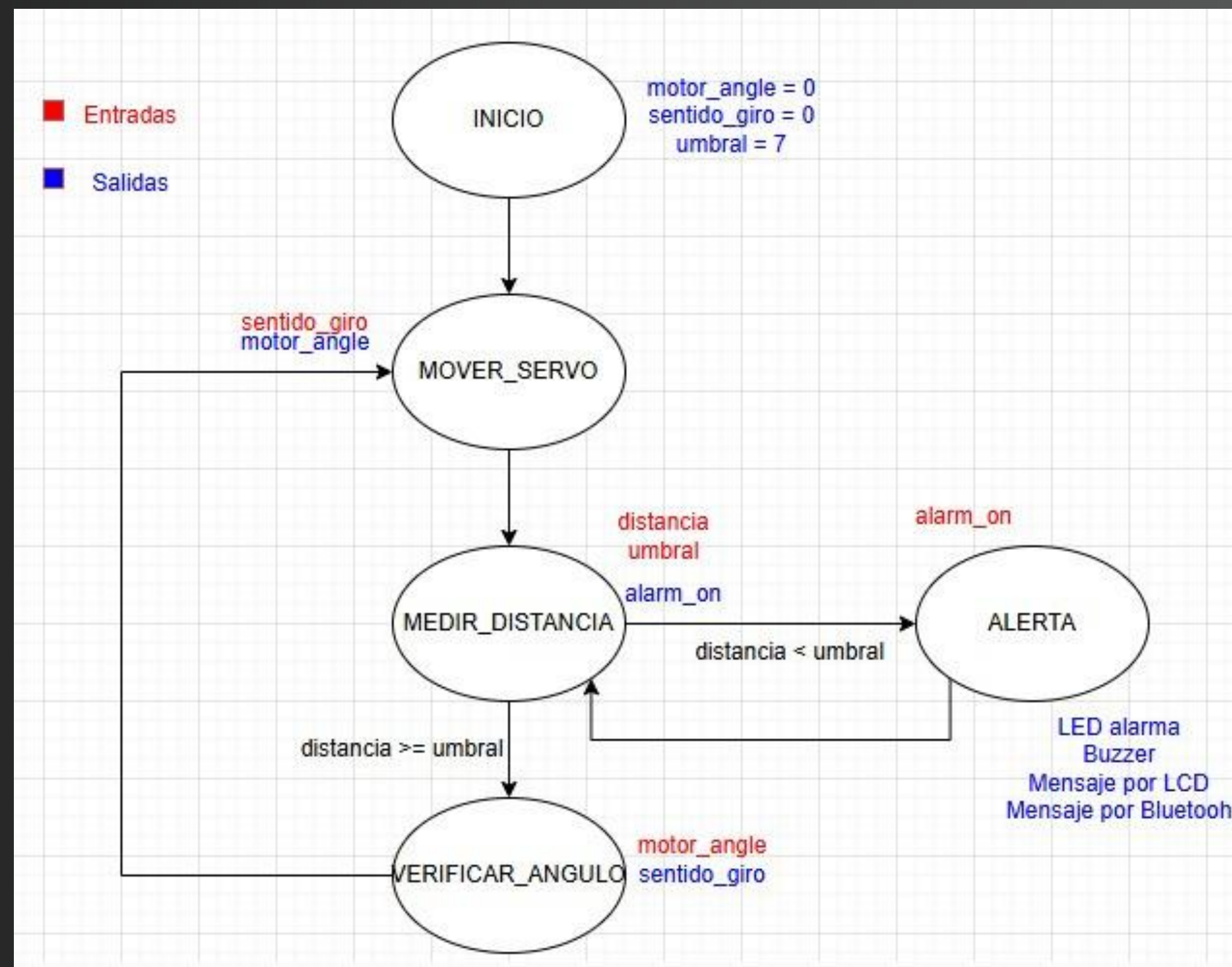
☐ Show only Modified Pins

Pin Name	Signal...	GPIO outp...	GPIO mode	GPIO Pul...	Maximum ...	User Label	Modified
PA2	n/a	High	Output Pu...	No pull-u...	Low	LED_PWR	
PB0	n/a	Low	Output Pu...	No pull-u...	Low	LED_ON	
PB1	n/a	Low	Output Pu...	No pull-u...	Low	LED_ALA...	
PB2	n/a	Low	Output Pu...	No pull-u...	Low	BUZZER	
PB12	n/a	Low	Output Pu...	No pull-u...	Low	Trigger	
PC13-ANTI_TAMP	n/a	Low	Output Pu...	No pull-u...	Low	LED0	

Select Pins from table to configure them. **Multiple selection is Allowed.**



Consideraciones sobre el software





Breve Repaso por el Código Main .c

```
18  /* USER CODE END Header */
19  /* Includes -----*/
20  #include "main.h"
21  #include "i2c.h"
22  #include "tim.h"
23  #include "usart.h"
24  #include "gpio.h"
25
26  /* Private includes -----*/
27  /* USER CODE BEGIN Includes */
28  #include "Headers.h"
29  /* USER CODE END Includes */
```

```
98      delay_t LCD_Refresh;
99      delayInit(&LCD_Refresh, REFRESH_RATE); //Cada 1s se refresca la pantalla LCD
100
101      //Inicializar Modulos
102      LCD_Init();
103      HCSR04_Init();
104      Servo_Init();
105      MEF_Init();
106      LCD_Clear();
107
108      HC05_SendString("Sistema Encendido\n");
109
110
```



Breve Repaso por el Código Main .c

```
while (1)
{

    if(delayRead(&LCD_Refresh) && !(MEF_GetAlarmState())){
        LCD_ShowData(MEF_GetDistance(), MEF_GetAngle());
    }

    MEF_Update();

    /* USER CODE END WHILE */

    /* USER CODE BEGIN 3 */

}
/* USER CODE END 3 */
```




Breve Repaso por el Código Maquina de Estados

```
case INICIO:

    //Valores Iniciales de Variables
    motor_angle = 0;
    sentido_giro = 0;
    umbral = 7;
    distancia = 10;
    alarm_on = 0;

    Servo_SetAngle(&htim3, TIM_CHANNEL_1, motor_angle);
    Secuencia_Inicio();

    MEF_Actual = MOVER_SERVO;
    break;
```



Breve Repaso por el Código Maquina de Estados

```
case MOVER_SERVO:  
    if(sentido_giro){  
        motor_angle = motor_angle - VEL_MOTOR;  
    }else{  
        motor_angle = motor_angle + VEL_MOTOR;  
    }  
    Servo_SetAngle(&htim3, TIM_CHANNEL_1, motor_angle);  
  
    MEF_Actual = MEDIR_DISTANCIA;  
    break;
```



Breve Repaso por el Código Maquina de Estados

```
case MEDIR_DISTANCIA:
    distancia = HCSR04_GetMeasure();
    if(distancia < umbral){
        MEF_Actual = ALERTA;
    }else{
        alarm_on = 0;
        HAL_GPIO_WritePin(LED_ALARM_GPIO_Port, LED_ALARM_Pin, 0);
        MEF_Actual = VERIFICAR_ANGULO;
    }
    break;
```




Breve Repaso por el Código Maquina de Estados

```
case VERIFICAR_ANGULO:  
    if(motor_angle == 180){  
        sentido_giro = 1;  
    }  
    if(motor_angle == 0){  
        sentido_giro = 0;  
    }  
    if(motor_angle > 180 || motor_angle < 0){  
        Error_Handler();  
    }  
    MEF_Actual = MOVER_SERVO;  
    break;
```



Breve Repaso por el Código Maquina de Estados

```
case ALERTA:
    alarm_on = 1;
    Mensaje_Alerta();
    MEF_Actual = MEDIR_DISTANCIA;
    break;
```

```
void Mensaje_Alerta(){
    //Salidas Digitales
    HAL_GPIO_WritePin(LED_ALARM_GPIO_Port, LED_ALARM_Pin, GPIO_PIN_SET);
    HAL_GPIO_WritePin(BUZZER_GPIO_Port, BUZZER_Pin, GPIO_PIN_SET);
    //Mensaje por LCD
    LCD_Clear();
    LCD_PutCur(0, 0);
    LCD_SendString("ALARMA");
    LCD_PutCur(1, 0);
    LCD_SendString("ACTIVADA!");
    //Mensaje por Bluetooth
    HC05_SendString("ALARMA ACTIVADA: OBJETO DETECTADO\n");
    //BUZZER
    HAL_Delay(500);
    LCD_Clear();
    HAL_GPIO_WritePin(BUZZER_GPIO_Port, BUZZER_Pin, GPIO_PIN_RESET);
    HAL_Delay(500);
}
```



**¡Muchas
Gracias!**