

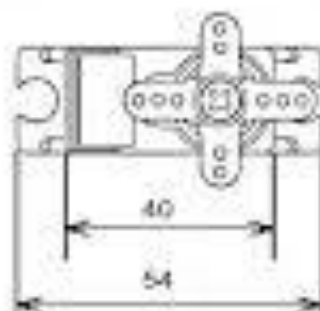
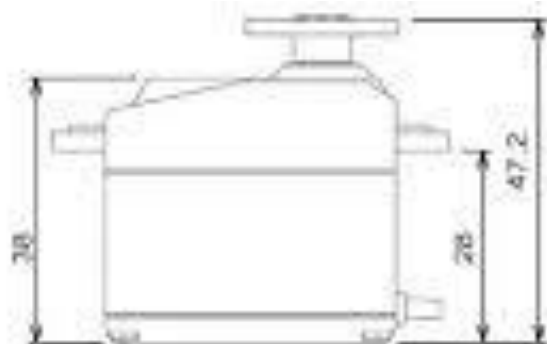
MICROS 32 BITS STM - SERVOMOTOR

ROBINSON JIMENEZ MORENO

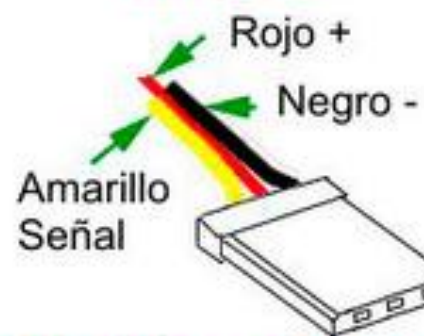


SG5010

Dual Ball Bearing high speed servo

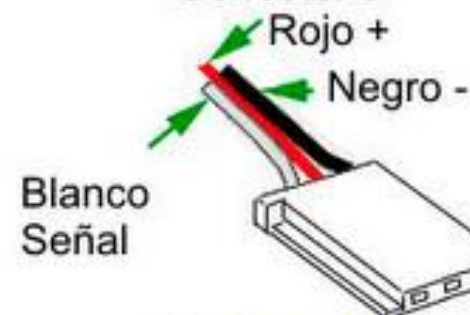


hitec

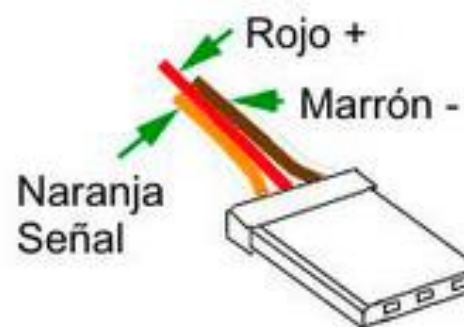


Futaba

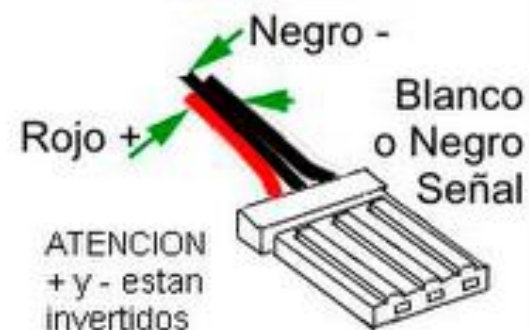
Conector J

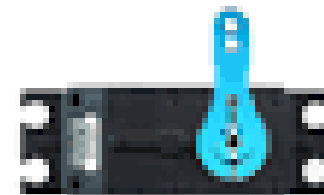
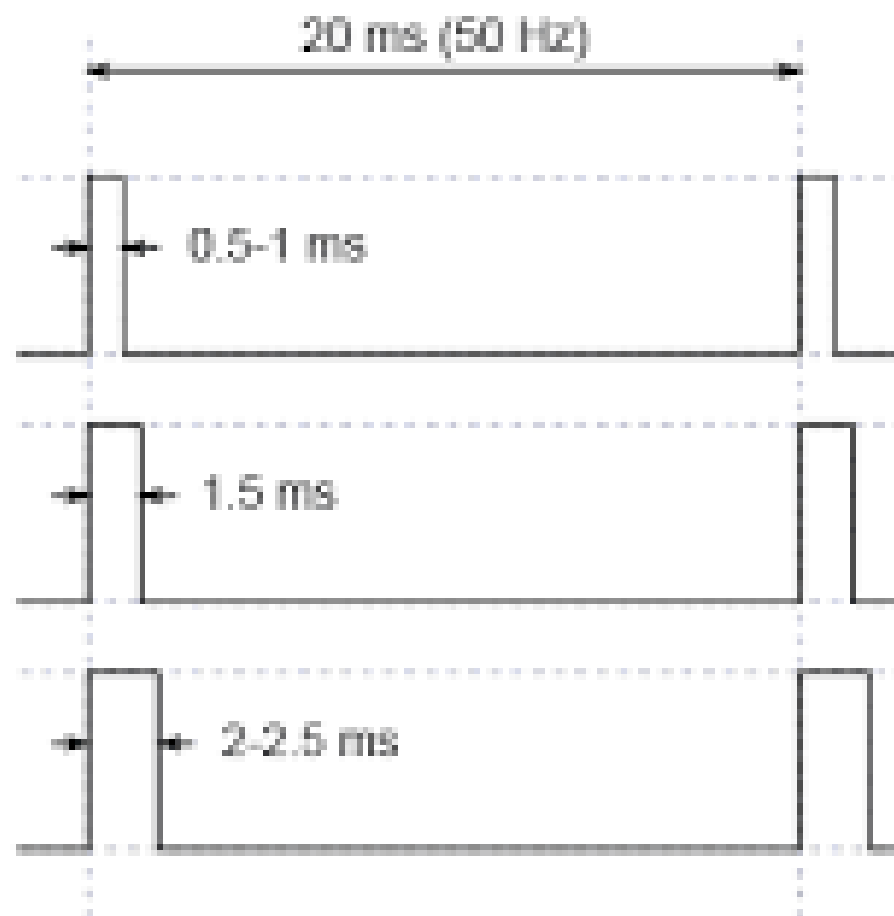


JR Radios Maxx



AIRTRONICS

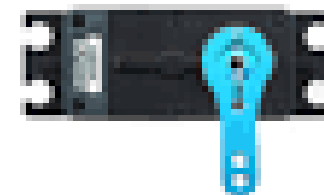




0°



90°



180°

```

1  #include <stdio.h>
2  #include "STM32F7xx.h"
3  #include <math.h>
4
5  int paso=0;
6
7  extern "C" {
8      void SysTick_Handler ( void )
9      {
10         paso++;
11     }
12 }
13
14 int main(void) {
15
16     RCC->AHB1ENR =0xFF; //TODOS LOS RELOJES ON -> Puerto A, B, C, E, F.
17
18     GPIOC->MODER = 0x000055;          //Pines del PB0 al PB11 como salida
19     GPIOC->OTYPER = 0;
20     GPIOC->OSPEEDR = 0x555555;        //medium speed
21     GPIOC->PUPDR = 0x10000000;        //pull up
22     //*****
23
24     SystemCoreClockUpdate();
25     SysTick_Config(SystemCoreClock/10000); //velocidad //tiempo mínimo
26
27     while(true) {
28         paso=0;
29         GPIOC->ODR=1;
30         while(paso<10);
31         paso=0;
32         GPIOC->ODR=0;
33         while(paso<200);
34     }

```

```

1  #include <stdio.h>
2  #include "STM32F7xx.h"
3  #include <math.h>
4
5  int paso=0;
6  int limite=0;
7
8  extern "C" {
9      void SysTick_Handler ( void )
10     {
11         paso++;
12     }
13 }
14
15 int main(void) {
16
17     RCC->AHBLENR = 0xFF; //TODOS LOS RELOJES ON -> Puerto A, B, C, E, F.
18
19     GPIOC->MODER = 0x000055;          //Pines del PB0 al PB11 como salida
20     GPIOC->OTYPER = 0;
21     GPIOC->OSPEEDR = 0x555555;        //medium speed
22     GPIOC->PUPDR = 0x10000000;        //pull up
23     //*****
24
25     SystemCoreClockUpdate();
26     SysTick_Config(SystemCoreClock/10000); //velocidad //tiempo mínimo
27
28     while(true) {
29         for(int i=0;i<10;i++){
30             paso=0;
31             GPIOC->ODR=1;
32             limite=i+10;
33             while(paso<limite);
34             paso=0;
35             GPIOC->ODR=0;
36             while(paso<250);
37         }}
38 }

```

```

////////////////////////////////////
#include <stdio.h>
#include "STM32F7xx.h"
#include <math.h>

int paso=0;
int servo=0;
int limite=1;

extern "C" {
    void SysTick_Handler ( void )
    {
        paso++;
    }
}

int main(void){

    RCC->AHB1ENR =0xFF; //TODOS LOS RELOJES ON -> Puerto A, B, C, E, F.

    GPIOC->MODER = 0x000055;          //Pines del PB0 al PB11 como salida
    GPIOC->OTYPER = 0;
    GPIOC->OSPEEDR = 0x555555;        //medium speed
    GPIOC->PUPDR = 0x10000000;        //pull up
    //*****
    SystemCoreClockUpdate();
    SysTick_Config(SystemCoreClock/10000); //velocidad //tiempo mínimo
    while(true)
    {
        if((GPIOC->IDR & 0X2000)==0X2000){
            servo+=servo;}
        paso=0;
        GPIOC->ODR=1;
        limite=servo+10;
        while(paso<limite);
        paso=0;
        GPIOC->ODR=0;
        while(paso<250);
    }
}

```

```

#include <stdio.h>
#include "stm32f7xx.h"
#include <stdlib.h>

int dato=500;

int main(void){

    RCC->AHB1ENR |= 0x47; //PUERTO A,B,C,G
    GPIOB->MODER |= 0x10004001; //Colocar en salida para encender los leds
    GPIOB->OTYPER |=0x0; //PUSH PULL
    GPIOB->OSPEEDR |=0x10004001; //Velocidad Medio
    GPIOB->PUPDR |=0x10004001;//PULL UP
    GPIOC->MODER &= ~(3UL<<2*13);
    //-----PTA6 -> TIM3_CH1-----//
    GPIOA->MODER = 0x02000; //PTA6 en MODO ALTERNO
    GPIOA->OTYPER = 0; //PUSH PULL -> PTA6
    GPIOA->OSPEEDR = 0x01000; //MEDIUM SPEED -> PTA6
    GPIOA->PUPDR = 0; //PULL-UP -> PTA6
    GPIOA->AFR[0] = 0x2000000; //PTA6 funcion alterna AF2= TIM3_CH1
    //-----TIM3_CH1-----//
    RCC->APB1ENR |= (1UL << 1); //HABILITA CLOCK TIM3
    TIM3->EGR |= (1UL<<0); //UG = 1 , RE-inicializar el contador
    TIM3->PSC = 15; //señal de reloj HSI=16Mhz, se necesita generar 1Mhz por lo tanto PSC=15
    TIM3->ARR = 25000; //con una frecuencia de 1Mhz -> T=1uS :
    TIM3->DIER |= (1UL<<0); //UIE = 1, update interrupt enable
    //conteo hasta 20000 significa 20000*1uS = 20ms //periodo de la señal de control del servo

    TIM3->CR1 |= (1UL<<0); //Enable counter
    TIM3->CCMR1 = 0x60; //PWM modo 1, preload del CCR1 deshabilitado, CH1 configurado como salida
    TIM3->CCER |= (1UL<<0); //OC1 signal is output on the corresponding output pin

    while(true) { GPIOB->ODR=0xffffffff;
if(GPIOC->IDR &= 0X2000){
    dato=dato+50;
for(int i=0;i<20000;i++){
    if(dato>2400){dato=2400;}
    TIM3->CCR1=dato;
    }//WHILE
} //MAIN

```