

```
#include <stdio.h>
#include <htc.h>

__CONFIG( CP_OFF & WDTE_OFF & PWRTE_OFF & FOSC_HS & MCLRE_ON );

void ServoTimerInit(void);
void InitTimer0(void);
void InitInterrupts(void);

//HXT 900 servo
//Pulse Cycle: 20 ms

#define TWO_SEC          153
#define INIT_TIMER0     0b11010111
#define MAX_TIMER_VAL   0x01
#define TIMER2_INIT     0b00000101
#define CLEAR           0
#define SET              1
#define RECT_HIGH_TIME  6
#define ATCK_HIGH_TIME 13

static unsigned char Two_Seconds_Timer = 0;
static unsigned char Servo_Overflows = 0;
static unsigned char ShouldAttack = 0;

void main(void)
{
    ServoTimerInit();
    InitInterrupts();
    InitTimer0();
    while(1)
    {
        static unsigned char LastBitState = 0;
        static unsigned char CurrentBitState = 0;

        CurrentBitState = RC1;
        if ((CurrentBitState == SET) && (CurrentBitState != LastBitState))
        {
            ShouldAttack = 1;
            TMR0 = MAX_TIMER_VAL; //2s complement of max value = 0xFF
            Two_Seconds_Timer = CLEAR;
        }
        else if ((CurrentBitState == CLEAR) && (CurrentBitState != LastBitState))
        {
            ShouldAttack = 0; //PWM retract
        }
        if ((Two_Seconds_Timer >= TWO_SEC) && (ShouldAttack))
        {
            Two_Seconds_Timer = CLEAR;
            ShouldAttack = 0; //PWM retract
        }
        if(TOIF == SET)
    }
}
```

```
{
    T0IF = CLEAR;
    Two_Seconds_Timer++;
}
LastBitState = CurrentBitState;
}
}

void interrupt ISR(void)
{
    if(TMR2IF)
    {
        Servo_Overflows++;
        if(Servo_Overflows >= 200)
        {
            RC5 = 1;
            Servo_Overflows = 0;
        }
        else if((Servo_Overflows == ATCK_HIGH_TIME) && (ShouldAttack))
        {
            RC5 = 0;
        }
        else if((Servo_Overflows == RECT_HIGH_TIME) && (!ShouldAttack))
        {
            RC5 = 0;
        }
        TMR2IF = 0;
    }
}

void ServoTimerInit(void)
{
    ANSEL = CLEAR;
    ANSELH = CLEAR;
    TRISC1 = SET;           //monitor pin RC1 set to input
    TRISC5 = CLEAR;       //CCP1 pin/ RC5 pin, set to output
    PR2 = 124;
    T2CON = TIMER2_INIT;
}

void InitTimer0(void)
{
    TMR0 = MAX_TIMER_VAL;
    OPTION_REG = INIT_TIMER0; //Initialize Option Register for Timer0, With internal clock and
    Presecale of 256
}

void InitInterrupts(void)
{
    GIE = 1;               //global enable interrupts
    PEIE = 1;             //peripheral enable interrupts
    TMR2IE = 1;           //pr2 match interrupt enable
}
```