

```

#include <htc.h>
#include <stdlib.h>
#include <stdio.h>
#include <string.h>

#define _XTAL_FREQ 4000000L

#define testbit(var,bit) ((var&(1<<bit)))
#define setbit(var,bit) ((var|= (1<<bit)))
#define clrbit(var,bit) ((var &=~(1<<bit)))
__CONFIG(FOSC_INTOSC & WDTE_OFF & PWRTE_ON & MCLRE_OFF & CP_OFF & CPD_OFF
&
BOREN_ON & CLKOUTEN_OFF & IESO_ON & FCMEN_ON);
__CONFIG(WRT_OFF & PLLEN_OFF & STVREN_ON & BORV_LO & LVP_OFF);

//__CONFIG(FOSC_INTOSC & WDTE_SWDTEN & PWRTE_ON & MCLRE_OFF & //CP_OFF &
CPD_OFF &
//BOREN_ON & CLKOUTEN_OFF & IESO_ON & FCMEN_OFF);
//__CONFIG(WRT_OFF & PLLEN_OFF & STVREN_ON & BORV_19 & LVP_OFF);

unsigned short i;
char counter;
char pos_data[15];
int el, az, az_temp, top_az, bottom_az, el_wrk;

void init(void)
{
    OSCCON = 0b11101000;           //4xPLL, 4MHz, Internal clock
    TRISA = 0b00000000;           //all output

    WPUB = 0b00000000;           //set weak pull up on B port
    ANSELA = 0b00000000;         //all digital
    ANSELB = 0b00000000;         //all digital
    OPTION_REG = 0b000000000;

//set up USART
    TRISB = 0b00100100;           //RB2, RB5 input for USART,
                                   //0=I/O output, 1= input

    SPBRGH = 0;
    SPBRGL = 25;                  //9600 baud at 4MHz clock
    BRGH = 1;
    BRG16 = 0;
    TXEN = 1;                     //enable transmitter
    CREN=1;                       //enable receiver
    SYNC = 0;
    SPEN = 1;
    TXCKSEL=1;                    //puts TX function on RB5, pin 11
    RXDTSEL=1;                    //puts RX function on RB2, pin 8
                                   //this allows SCL1 to be on
    RB4, pin 10
    PORTA = 0b00000000;           //set all PORTA to 0
    PORTB = 0b00000000;
    TMR1ON=1;

```

```

}

void hundred_ms_delay(int t)          //100 ms intervals
{
    for(i=0;i<t;i++)
    {
        __delay_ms(100);
    }
}

void pulse_width(int width)          //uses timer 1 to delay for the
desired pulse width
{
    TMR1=65565-(width);              //adjust for overhead
    TMR1IF=0;
    while (!TMR1IF) continue;
}

int ascii_to_dec(char x)              //converts the ascii number from
SatPC32 into a decimal value
{
    int decimal;
    decimal=(pos_data[x]-48)*100+(pos_data[x+1]-48)*10+(pos_data[x+2]-
48);
    return decimal;
}

void get_point (void)
{
    //SatPC32 format:  AZ###.#
    EL###.# #'s are in ascii format
    pos_data[0]=0;
    do                                //wait for the letter A to
come from SatPC32
    {
        while (!RCIF) continue;
        pos_data[0]=RCREG;
    }while (pos_data[0] != 'A');
    RA4=1;
    for (i=1;i<15;i++)
    {
        while (!RCIF) continue;
        pos_data[i]=RCREG;
    }
    el = ascii_to_dec(10);            //convert el
    az = ascii_to_dec(2);            //convert az
    RA4=0;
    // to view data from Putty, open this section
    // for(i=0;i<15;i++)
    // {
    //     TXREG=pos_data[i];
    //     while (!TXIF) continue;
    // }
}

```

```

}

void main(void)
{
    init();

while(1)
{
    get_point(); //listen to SatPC32 for new
data

    if (az<181) //if less than 181 degrees
move bottom az servo
    {
        top_az=2250;
        bottom_az=2250-(az*10);
    }
    else //if greater than 180
degrees move top az servo
    {
        bottom_az = 450;
        top_az = 450+((360-az)*10);
    }
    el_wrk = 2250-(el*10);

    for(counter=0;counter<100;counter++)//100 counts ensures complete
turn, probably can be less
    {
        RA2 = 1;
        pulse_width(top_az);
        RA2 = 0;
        RA1 = 1;
        pulse_width(bottom_az);
        RA1 = 0;
        RA0 = 1;
        pulse_width(el_wrk); //move el servo
        RA0 = 0;
        __delay_ms(17); //delay an additional 17 mS
    }
}
}

```