```
//------------------------------------------------------------------------------
// use Scara final 1.py
//
// This code controls two servos and a dc motor in a scara configuration.
// The two servo positions are received over UART from the Python code.
//------------------------------------------------------------------------------



//libraries
#include "project.h"
#include "math.h"

//------------------------------
//union for UART data
//------------------------------

union forUartCon {    //used to convert uint8 to int
    uint8 int8val;
    int intvalue;
    } fuc;

//------------------------------
//function for dc motor
//------------------------------

void DCupORdown(int Target_Count_f)     //move dc motor to up or down position
{
    int time=0;
    int Count;                          //encoder count
    int Error;                          //dif of up/down position and encoder
    int Speed;                          //0-100, 0 = max speed, 100 = stop
    float Kp=0.3;                       //proportional gain

    while(time<400)                     //0.4s to move rack up or down
    {
        Count=QuadDec_1_GetCounter();   //get encoder value
        Error=Target_Count_f-Count;     //up/down position compared to encoder
        if (Error>0)                    //rotate clockwise
        {
            Speed=Kp*Error;             //proportional control, calc speed
            if (Speed>100)
            {
                Speed=0;                //0 is max speed
            }
            else
            {
                Speed=100-Speed;
            }
            PWM_2_WriteCompare1(Speed); //set dc motor clockwise speed
            PWM_2_WriteCompare2(100);   //turn off counter clockwise speed
        }
```

```
        else                            //or rotate counterclockwise
        {
            Speed=-Kp*Error;            //proportional control, calc speed
            if (Speed>100)
            {
                Speed=0;                //0 is max speed
            }
            else
            {
                Speed=100-Speed;
            }
            PWM_2_WriteCompare1(100);   //turn off dc motor clockwise speed
            PWM_2_WriteCompare2(Speed); //set counter clockwise speed
        }

        CyDelay(10);                    //makes position correction every 1 ms
        time=time+10;
    }

}

int main(void)
{

    //------------------------------
    //variables
    //------------------------------
    int Target_Count1=750; //rack and pinion, up value
    int Target_Count2=250; //..., down value
    int tempDelay=1700;
    uint8 Receive;
    unsigned char testA[8];
    int m=0;
    int testB[8];
    uint8 startB=100;
    int Xcompare;
    int Ycompare;
    int Xmm=0;
    int Ymm=0;
    int homeS=4385;
    //------------------------------
    //start
    //------------------------------
    QuadDec_1_Start();          //rack and pinion encoder
    LCD_Char_1_Start();
    PWM_1_Start();              // for rc servos
    PWM_2_Start();              //for DC motor
    UART_1_Start();
    QuadDec_1_SetCounter(250);  //set rack down position
```

```
//------------------------------
//initialize with Python
//------------------------------

LCD_Char_1_ClearDisplay();
LCD_Char_1_Position(0,0);
LCD_Char_1_PrintString("Initializing:");
LCD_Char_1_Position(1,0);
LCD_Char_1_PrintString("Start Python now");

Receive=UART_1_GetChar();
while (Receive==0)              //wait for Python
{
    Receive=UART_1_GetChar();   //initialize
}

LCD_Char_1_ClearDisplay();
LCD_Char_1_Position(0,0);
LCD_Char_1_PrintString("Press button");
LCD_Char_1_Position(1,0);
LCD_Char_1_PrintString("to home scara");

//------------------------------
//initial homing sequence
//------------------------------
m=0;
while (m==0)                            //wait for user button press-
{
    Switch_1_Read();
    if (Switch_1_Read()==1)
    {
        CyDelay(150);
        if (Switch_1_Read()==1)
        {
            m=1;
            UART_1_PutChar(startB);     //Python sets up camera
            DCupORdown(Target_Count1);  //rack up
            PWM_1_WriteCompare1(homeS); //rc servos to home position
            PWM_1_WriteCompare2(homeS);
            CyDelay(1200);
        }
    }
}

LCD_Char_1_ClearDisplay();
LCD_Char_1_Position(0,0);//print 1st row
LCD_Char_1_PrintString("Press button");
LCD_Char_1_Position(1,0);//print 2nd row
LCD_Char_1_PrintString("set background");
```

```
//------------------------------
//infinite loop
//------------------------------
for(;;)
{

//------------------------------
//user sets camera background
//------------------------------
m=0;
while (m==0)                            //wait for user button press
{
    Switch_1_Read();
    if (Switch_1_Read()==1)
    {
        CyDelay(150);
        if (Switch_1_Read()==1)
        {
            m=1;
            UART_1_PutChar(startB);     //Python sets camera background
        }
    }
}

//---------------------------------
//Python calculates object position
//---------------------------------
LCD_Char_1_ClearDisplay();
LCD_Char_1_Position(0,0);//print 1st row
LCD_Char_1_PrintString("Put object");
LCD_Char_1_Position(1,0);//print 2nd row
LCD_Char_1_PrintString("in workspace");


//------------------------------
//receive position data
//------------------------------
m=0;
Receive=UART_1_GetChar();
while (Receive==0)
{
    Receive=UART_1_GetChar();           //wait
    if (Receive>0)
    {
     while (m<8)                        //receive 8 bytes
        {
            if (Receive>0)
            {
                testA[m]=Receive;       //testA array stores bytes
                m=m+1;
            }
            CyDelay(1);                 //loop needs improvement but works
```

```
            Receive=UART_1_GetChar();
        }
        Receive=1;
    }


}



//------------------------------------------------------
//convert data, move to position, print position to LCD
//------------------------------------------------------
m=0;
while (m<8)
    {
        fuc.int8val=testA[m];               //uint8 to union
        testB[m]=fuc.intvalue;              //get int from union
        m=m+1;
    }

    DCupORdown(Target_Count1);              //rack up

    Xcompare=256*testB[0]+testB[1];         //2 bytes for servo 1 position
    Ycompare=256*testB[2]+testB[3];         //2 bytes for servo 2 position

    Xmm=testB[5];                           //for LCD, x position
    Ymm=testB[7];                           //for LCD, y position

    PWM_1_WriteCompare1(Xcompare);          //move servo 1 to position
    PWM_1_WriteCompare2(Ycompare);          //move servo 2 to position

    LCD_Char_1_ClearDisplay();
    LCD_Char_1_Position(0,0);               //for LCD, +/-x
    if (testB[4]==100)
    {
        LCD_Char_1_PrintString("X(mm)=-");
        LCD_Char_1_Position(0,7);
    }
    else
    {
        LCD_Char_1_PrintString("X(mm)=");
        LCD_Char_1_Position(0,6);
    }
    LCD_Char_1_PrintNumber(Xmm);

    LCD_Char_1_Position(1,0);               //for LCD, +/-y
    if (testB[6]==100)
    {
        LCD_Char_1_PrintString("Y(mm)=-");
        LCD_Char_1_Position(1,7);
    }
```

```
        else
        {
            LCD_Char_1_PrintString("Y(mm)=");
            LCD_Char_1_Position(1,6);
        }
        LCD_Char_1_PrintNumber(Ymm);

        CyDelay(tempDelay);                     //delay to reach position
        DCupORdown(Target_Count2);              //rack down

    }   //end of for loop
}
```