```
/*################################ Project Header
#################################*/
/*Project Name:  Microcontroller project
 Organization:   Wentao Bi
 Lead Author:    Wentao Bi
 Co-Author:   John Sowlik

; Abstract: This is a collection programming about Pixycam camera, lidar
vite-3,
 Arobotix, lidar servo position, xbee.

;
;******************************** End of Header
********************************


;############################### Copyright Notice
###############################
; Copyright 2016, by Wentao Bi. This work is provided as a part of the
; HCS12 lab manual. Permission to use this work is granted to the
original
; owner of the hcs12 lab manual, for use in conjunction with the hcs.
; Other use of this work or the resultant software is prohibited without
written
; permission from Wentao Bi.
;
; This work is provided "as-is", and without warranty of any kind,
expressed,
; implied or otherwise, including and without limitation, any warranty of
; merchantability or fitness for a particular purpose. In no event shall
the
; author be liable for any special, incidental, indirect or consequential
; damages whatsoever in connection with the use or inability to use this
; software.
;****************************** End of Copyright
*******************************


;################################ Version History
###############################
; Start Date:    2016/12/07  (Date Format: YYYY/MM/DD)
; Latest Date:2016/12/12
; Version:       1.0
;
; Date:                 By/Reviewed:     Description of changes:
; 2016/12/07       Wentao Bi      Initial Release
;******************************* End of History
*******************************/
//when you use Xbee, you have to change sci0 to sci1, sci0 is for Xbee.
/*;***************************** End of History
*******************************/
#include <hidef.h>      /* common defines and macros */
#include <stdlib.h>
#include "derivative.h"      /* derivative-specific definitions */
```

```
#include "p2os.h"
#include "SCI0_Buffered.h"
#include "PLL.H"
#include <stdio.h>
#include "PixyHS12.h"

void delay(uint);
extern void FollowBlock(int);
extern int TrackBlock(int);
extern SRXBuf SCI0RXBuf;
extern SRXBuf SCI1RXBuf;
extern void panUpdate(int32_t);
extern int32_t pan_m_pos, tilt_m_pos;
int TrackNum;
//unsigned int turnMultiplier;



/*;************************ milestone 1 definition
*********************************/
typedef unsigned char unchar;              // define unsigned character
shortcut
typedef unsigned int unint;               // define unsigned integer
shortcut
#define mainClock   8000000               // clock speed on the HCS12
#define SCIBR       =mainClock/(16*38400) // calculate the baud rate at
38400
#define delay1ms    mainClock/(3000) /* Set delay time */

//Function Prototypes
int GetLidardiatance(void);
void StraifForward(void);
void StraifRight(void);
void StraifLeft(void);
void StraifReverse(void);
void TurnRight(void);
void TurnLeft(void);
void RotateRight(void);
void RotateLeft(void);
void SetGait(char);
void Stop();

// Global variables
// Right and Left are stationary at 0x80
unchar Right_V;
unchar Right_H;
unchar Left_V;
unchar Left_H;
unchar Buttons;  //controls gait
unchar CheckSum;

SRXBuf SCI0RXBuf;
SRXBuf SCI1RXBuf;
//Function prototypes
```

```
void Delay (unint num);
/*;*******************************  definition end
*********************************/





//define basic variables for integrated program
unsigned int length,angel,g_iRtosClock = 0x00;
unsigned char poshighb,poslowb;
//globals
int BlocksInFrame=0;
int total=0;


/*
====================================Start
main=======================================
*/

void main(void)

{
  Delay(2000);
  EnableInterrupts;
  SCI0_Init(38400);               // initialize sci0
  SCI1_Init(38400);               // initialize sci1
  init();                         //allocate memory for blocks
  initSPI();                      //initialize SPI0



  for(;;)//main loop start
    {

       //start by gathering data from peripherals; xBee data also sent at
this time
/*;****************************get pixy objects and configure
****************** ******* **************/
      BlocksInFrame=getBlocks(100);   // Check if there are any blocks
matching a signature
      TrackNum = TrackBlock(BlocksInFrame);      // Track the blocks that
match our signature --> If none, do nothing
          //FollowBlock(TrackNum);
      BlocksInFrame=getBlocks(100);
      TrackBlock(BlocksInFrame);
/*;*************************** lidar
distance*****************************************/
      length=GetLidardiatance();
      poshighb=getcharSCI1();//0xhigh //get servo position
      poslowb=getcharSCI1(); //0xlow
/*;*************************** get lidar servo position
******************** ****** **********/
```

```
      angel= (poshighb<<8)+poslowb;
/*;******************************  Send data to Xbee
***************************************/
      putcharSCI0(0xFF);
      putcharSCI0(poshighb);
      putcharSCI0(poslowb);
      putcharSCI0(length);
      putcharSCI0(0x00);

/*;*****************************Object avoidance algorithm
*************************/
      if (pan_m_pos > 0x220)
      {         //move left
      //SetGait(0x02);
      TurnLeft();
         if((angel > 3900) && (angel <= 800)){
         if (length < 50)
            Stop();
          else
            StraifForward();
        } else if((angel > 2800) && (angel <= 3900)){
          if (length < 50)
            Stop();
          else
            StraifLeft();
        } else if((angel > 2100) && (angel <= 2800)){
          if (length < 50)
            Stop();
          else
            StraifReverse();
        } else if((angel > 800) && (angel <= 2100)){
          if (length < 50)
            Stop();
          else
            StraifRight();
        }
      }
      if ( (pan_m_pos > 0x180) && (pan_m_pos < 0x220) ){    //go straight
      //SetGait(0x02);
      StraifForward();
       if((angel > 3900) && (angel <= 800)){
         if (length < 50)
            Stop();
          else
            StraifForward();
        } else if((angel > 2800) && (angel <= 3900)){
          if (length < 50)
            Stop();
          else
            StraifLeft();
        } else if((angel > 2100) && (angel <= 2800)){
          if (length < 50)
            Stop();
          else
```

```
            StraifReverse();
        } else if((angel > 800) && (angel <= 2100)){
          if (length < 50)
            Stop();
          else
            StraifRight();
        }

      }
      if (pan_m_pos < 0x180){         //move right
      // SetGait(0x02);
      TurnRight();
       if((angel > 3900) && (angel <= 800)){
         if (length < 50)
            Stop();
          else
            StraifForward();
        } else if((angel > 2800) && (angel <= 3900)){
          if (length < 50)
            Stop();
          else
            StraifLeft();
        } else if((angel > 2100) && (angel <= 2800)){
          if (length < 50)
            Stop();
          else
            StraifReverse();
        } else if((angel > 800) && (angel <= 2100)){
          if (length < 50)
            Stop();
          else
            StraifRight();
        }
      }

    }
}



/*;**************************   *** lidar programming **
*********************************/
 int GetLidardiatance(void){
  /* put your own code here */ //
   unint edge1,edge2,period,meter5,distance;//5000/8=625

    meter5=5000; //500cm*10us/cm=5000
    TSCR1=0x90;  //enable TCNT, fast flag clear
    TIOS &=0xFE; //enable input
    TSCR2=0x03;  //prescale 8, 1us,//but Anna told me to set 64, 8us, I
donot know why for 1us is time-consuming and less programming
    DDRT &=0xFE; //input monitor

   // TCTL4=0x01; //rising, after record rising, set it to falling
```

```
    TFLG1=0x01; //clear c0f flag
    DDRA=0x01; //output trigger
    PORTA=0x01;//ready for 0
    //TC0=TCNT;

          TCTL4=0x01; //currently set to rising, later, after record,
rising, set it to falling
          PORTA=0x00;      //turn on trigger and begin to configure
          while(!(TFLG1&0x01));  //wait
          edge1=TC0;     //rising edge
          //TFLG1=0x01;because fast flag clear
          TCTL4=0x02;//falling edge
          while(!(TFLG1&0x01)&&(TCNT<(edge1+5000)));//wait
          edge2=TC0;       //fall edge
          PORTA=0x01;      //trigger, complete configure

 /*******************************configure two
edges***********************************/
         if (edge2>=edge1)//check overflow
          {
          period=edge2-edge1;   //no overflow
          }
         else
          {
          period=65535-edge1+edge2; //overflow
          }


         if (period>=meter5) //check 5 meters
         {
         period=meter5;      //only get 5 meters
         distance=meter5/10; //get distance
         }
         else
          {
          distance=period/10;//no moer than 5 meters,get distance
(centermeter as default)
          }
          return (distance);
          //while(1);  //just get one distance as example
    }
/*;*********************************** lidar programming end **
*********************************/

/*;********************** *** SCI 0 and SCI 1 initialize
************************************/
/************************************sci 0
*************************************************/
void SCI0_Init(unsigned long Baud)
{
  // initialize buffer
   SCI0RXBuf.In=0;
   SCI0RXBuf.Out=0;
   /* Variable Declarations */
   /* Begin Function InitSCI() */
   SCI0BD = (SysClk / 16) / Baud; /* calculate the SCI Baud register
value */
   /* using the system clock rate */
   SCI0CR2 = SCI0CR2_RIE_MASK + SCI0CR2_TE_MASK + SCI0CR2_RE_MASK ;
   /* enable both the transmitter & receiver, only RX interrupt */
} /* end InitSCI */


int putcharSCI0(char c)
{
      while (!(SCI0SR1&SCI0SR1_TDRE_MASK)) /* check TXbuffer is ready or
not,TDRE */
      ; //OSTimeDly(1); /* wait here */
      SCI0DRL=c; /* put the char in the TX */
} /* end putchar */


int getcharSCI0(void)
{
   while (!(SCI0SR1&SCI0SR1_RDRF_MASK)) /* check RX flag, has data or
not...,RDRF */
   //OSTimeDly(1); /* no data wait here */
   return(SCI0DRL);
} /* end getchar */


 void putsSCI0(char *ptr)                    // output string to SCI0
{
      while(*ptr)
      {
      putcharSCI0(*ptr);
      ptr++;
      }
}


char getcharSCI0buffer(void)
{
   return SCI0RXBuf.buf[SCI0RXBuf.Out++];   // just read data ( no
checking in/out index here

   SCI0RXBuf.Out&=(RXBufferSize-1);         // limit in max index size
}


unsigned char DataInSCI0buffer(void)
{
   return ((SCI0RXBuf.In-SCI0RXBuf.Out)&(RXBufferSize-1)); // get
difference in index
}
```

```
/***********************sci
1*****************************************/

void SCI1_Init(unsigned long Baud)   //Initialize the serial
communication and baud rate
{
   // initialize buffer
   SCI1RXBuf.In=0;
   SCI1RXBuf.Out=0;
   /* Variable Declarations */
   /* Begin Function InitSCI() */
   SCI1BD = (mainClock / 16) / Baud; /* calculate the SCI Baud register
value */
   /* using the system clock rate */
   SCI1CR2 = SCI1CR2_RIE_MASK + SCI1CR2_TE_MASK + SCI1CR2_RE_MASK ;
   /* enable both the transmitter & receiver, only RX interrupt */
} /* end InitSCI */


int putcharSCI1(char c)
{
      while (!(SCI1SR1&SCI1SR1_TDRE_MASK)) /* check TXbuffer is ready or
not,TDRE */
      ;//OSTimeDly(1); /* wait here */
      SCI1DRL=c; /* put the char in the TX */
}


int getcharSCI1(void)
{
   while (!(SCI1SR1&SCI1SR1_RDRF_MASK)); /* chec RX flag, has data or
not...,RDRF */
   //OSTimeDly(1); /* no data wait here */
   return(SCI1DRL);
}


void putsSCI1(char *ptr)  // output string to SCI1
{
      while(*ptr){
      putcharSCI1(*ptr);
      ptr++;
      }
}


char getcharSCI1buffer(void)
{
   return SCI1RXBuf.buf[SCI1RXBuf.Out++]; // just read data ( no checking
in/out index here)


   SCI1RXBuf.Out&=(RXBufferSize-1);       // limite in max index size
}
```

```
unsigned char DataInSCI1buffer(void)
{
   return ((SCI1RXBuf.In-SCI1RXBuf.Out)&(RXBufferSize-1)); // get
difference in index
}

/***********************SCI 0 and SCI 1 initialize
end*****************************/

/*********************** Delay how many ms function
*********************************/
void Delay (unint num) {
unint counter;

while (num > 0) {           //each time this loop runs, it delays the
process 1ms

   counter = delay1ms;      //set the counter

   while (counter > 0) {
    counter = counter - 1;  //decrement the counter
   }
    num = num - 1;          //decrement the required number of delay
milliseconds
 }
}



/************************** Movement Functions
*************************************/
void StraifForward() {
Right_V  = 0x7F;
Right_H  = 0x81;
Left_V   = 0xE6;
Left_H   = 0x81;
Buttons  = 0x00;
CheckSum = (255 - ((Right_V+Right_H+Left_V+Left_H+Buttons)%256));

   putcharSCI1(0xFF);
   putcharSCI1(Right_V);
   putcharSCI1(Right_H);
   putcharSCI1(Left_V);
   putcharSCI1(Left_H);
   putcharSCI1(Buttons);
   putcharSCI1(0x00);
   putcharSCI1(CheckSum);

  // Delay(33); //send this at at most 60Hz
}
```

```
void StraifReverse(){
Right_V  = 0x7F;
Right_H  = 0x81;
Left_V   = 0x80;
Left_H   = 0x1A;
Buttons  = 0x00;
CheckSum = (255 - ((Right_V+Right_H+Left_V+Left_H+Buttons)%256));

   putcharSCI1(0xFF);
   putcharSCI1(Right_V);
   putcharSCI1(Right_H);
   putcharSCI1(Left_V);
   putcharSCI1(Left_H);
   putcharSCI1(Buttons);
   putcharSCI1(0x00);
   putcharSCI1(CheckSum);

 //  Delay(33); //send this at at most 60Hz

}


void StraifRight(){
  Right_V  = 0x7F;
  Right_H  = 0x81;
  Left_V   = 0x80;
  Left_H   = 0xE6;
  Buttons  = 0x00;
CheckSum = (255 - ((Right_V+Right_H+Left_V+Left_H+Buttons)%256));

   putcharSCI1(0xFF);
   putcharSCI1(Right_V);
   putcharSCI1(Right_H);
   putcharSCI1(Left_V);
   putcharSCI1(Left_H);
   putcharSCI1(Buttons);
   putcharSCI1(0x00);
   putcharSCI1(CheckSum);

   //Delay(33); //send this at at most 60Hz

}


void StraifLeft(){
  Right_V  = 0x7F;
  Right_H  = 0x81;
  Left_V   = 0x80;
  Left_H   = 0x1A;
  Buttons  = 0x00;
CheckSum = (255 - ((Right_V+Right_H+Left_V+Left_H+Buttons)%256));

   putcharSCI1(0xFF);
   putcharSCI1(Right_V);
   putcharSCI1(Right_H);
   putcharSCI1(Left_V);
   putcharSCI1(Left_H);
   putcharSCI1(Buttons);
   putcharSCI1(0x00);
   putcharSCI1(CheckSum);

   //Delay(33); //send this at at most 60Hz

}


void Stop(){
  Right_V  = 0x80;
  Right_H  = 0x80;
  Left_V   = 0x80;
  Left_H   = 0x80;
  Buttons  = 0x80;
  CheckSum = (255 - ((Right_V+Right_H+Left_V+Left_H+Buttons)%256));

  putcharSCI1(0xFF);
  putcharSCI1(Right_V);
  putcharSCI1(Right_H);
  putcharSCI1(Left_V);
  putcharSCI1(Left_H);
  putcharSCI1(Buttons);
  putcharSCI1(0x00);
  putcharSCI1(CheckSum);

  //Delay(33); //send this at at most 60Hz

};

void TurnRight(){
  Right_V  = 0x7F;
  Right_H  = 0xE6;
  Left_V   = 0xE6;
  Left_H   = 0x81;
  Buttons  = 0x00;
  CheckSum = (255 - ((Right_V+Right_H+Left_V+Left_H+Buttons)%256));

  putcharSCI1(0xFF);
  putcharSCI1(Right_V);
  putcharSCI1(Right_H);
  putcharSCI1(Left_V);
  putcharSCI1(Left_H);
  putcharSCI1(Buttons);
  putcharSCI1(0x00);
  putcharSCI1(CheckSum);

  //Delay(33); //send this at at most 60Hz

};
void TurnLeft(){
  Right_V  = 0x7F;
  Right_H  = 0x1A;
  Left_V   = 0xE6;
  Left_H   = 0x81;
  Buttons  = 0x00;
  CheckSum = (255 - ((Right_V+Right_H+Left_V+Left_H+Buttons)%256));

  putcharSCI1(0xFF);
  putcharSCI1(Right_V);
  putcharSCI1(Right_H);
  putcharSCI1(Left_V);
  putcharSCI1(Left_H);
  putcharSCI1(Buttons);
  putcharSCI1(0x00);
  putcharSCI1(CheckSum);

  //Delay(33); //send this at at most 60Hz

};
void RotateRight(){
  Right_V  = 0x7F;
  Right_H  = 0xE6;
  Left_V   = 0x80;
  Left_H   = 0x81;
  Buttons  = 0x00;
  CheckSum = (255 - ((Right_V+Right_H+Left_V+Left_H+Buttons)%256));

  putcharSCI1(0xFF);
  putcharSCI1(Right_V);
  putcharSCI1(Right_H);
  putcharSCI1(Left_V);
  putcharSCI1(Left_H);
  putcharSCI1(Buttons);
  putcharSCI1(0x00);
  putcharSCI1(CheckSum);

  //Delay(33); //send this at at most 60Hz

};
void RotateLeft(){
  Right_V  = 0x7F;
  Right_H  = 0x1A;
  Left_V   = 0x80;
  Left_H   = 0x81;
  Buttons  = 0x00;
  CheckSum = (255 - ((Right_V+Right_H+Left_V+Left_H+Buttons)%256));

  putcharSCI1(0xFF);
  putcharSCI1(Right_V);
  putcharSCI1(Right_H);
  putcharSCI1(Left_V);
  putcharSCI1(Left_H);
  putcharSCI1(Buttons);
  putcharSCI1(0x00);
  putcharSCI1(CheckSum);

  //Delay(33); //send this at at most 60Hz

};



/*Set Gait changes the robots Gait Mode
0x00: Don't Change Gait
0x01: 1 leg at a time, slowly
//0x02: 2 legs at a time, slowly
0x04: 1 leg at a time
0x08: 2 legs at a time, medium pace
0x10: 3 legs at a time
//0x20: 2 legs at a time, quickly
0x40: Do Nothing
0x80: Do Nothing
*/
void SetGait(char NewGait){
  Right_V  = 0x80;
  Right_H  = 0x80;
  Left_V   = 0x80;
  Left_H   = 0x80;
  Buttons  = NewGait;
  CheckSum = (255 - ((Right_V+Right_H+Left_V+Left_H+Buttons)%256));

  putcharSCI1(0xFF);
  putcharSCI1(Right_V);
  putcharSCI1(Right_H);
  putcharSCI1(Left_V);
  putcharSCI1(Left_H);
  putcharSCI1(Buttons);
  putcharSCI1(0x00);
  putcharSCI1(CheckSum);

   Delay(33); //send this at at most 60Hz


}
/************************** Movement Functions
end*********************************/
```