//all four us sensors

/\* March 11 2:25am : This program runs 4 ultrasonic sensors round robin. tested with 4 but RB4 has no signal. The trigger for RB4 (RB2) is blinking as expected.

\* (tested with 3 and confirmed in multisensor.c. Reminder: run with KPD disconnected.\*/

/\*March 20 10:50pm : tested and basically works, works without breaking for at least 3 minutes YES\*/

#include <xc.h>

#include <stdio.h>

#include "configBits.h"

#include "I2C.h"

#include "lcd.h"

//#define RS RD2

//#define EN RD3

//#define D4 RD4

//#define D5 RD5

//#define D6 RD6

//#define D7 RD7

#define us\_delay 150

int dist\_final[4][6] = {

{0, 0, 0, 0, 0,0}, // dist\_final[0]

{0, 0, 0, 0, 0,0}, // dist\_final[1]

{0, 0, 0, 0, 0,0}, // dist\_final[2]

{0, 0, 0, 0, 0,0}, // dist\_final[3]

};

int a[4];

int sum[4];

int b;

bool sumf =false;

volatile int meas =0;

volatile bool echoo;

void echo();

void print\_echo();

void \_\_interrupt() ISR(){

if(RBIF == 1){ //Makes sure that it is PORTB On-Change Interrupt

RBIE = 0; //Disab-le On-Change Interrupt

// echoo = true;

// trig();

echo();

// print\_echo();

RBIF = 0; //Clear PORTB On-Change Interrupt flag

RBIE = 1; //Enable PORTB On-Change Interrupt

}

// else{

//// echoo = false;

// }

}

void echo(){

if (b==1){

if (RB7 == 1){ //If ECHO is HIGH

TMR1ON = 1;

} //Start Timer

if (RB7 == 0){ //If ECHO is LOW

TMR1ON = 0; //Stop Timer

a[0] = (TMR1L | (TMR1H<<8))/58.82; //Calculate Distance

// a[0] = (TMR1L | (TMR1H<<8))\*0.034/2;

}

if ((~sumf)&&(dist\_final[0][0]!=0)&&(dist\_final[0][1]!=0)&&(dist\_final[0][2]!=0)&&(dist\_final[0][3]!=0)&&(dist\_final[0][4]!=0)){ //if haven't summed before

sum[0] = (dist\_final[0][0]+dist\_final[0][1]+dist\_final[0][2]+dist\_final[0][3]+dist\_final[0][4]);

sumf = true;

}

if (sumf){

sum[0] = sum[0] - dist\_final[0][meas] + a[0];

dist\_final[0][5] = (sum[0]/5);

}

dist\_final[0][meas] = a[0];

// if (meas ==4){

// meas =0;

// }

// if (meas==0){

// meas=1;

// }

// if (meas==1){

// meas=2;

// } if (meas==2){

// meas=3;

// }

// if (meas==3){

// meas=4;

// }

/\* keep an index for your measurement number that continues in a loop. use this to find out where to subtract and add the new measurement and recalculate the sum and avg

if (dist\_final[0][4]==0){ //if the last element of moving avg array is 0

for (int i=0; i<5; i++){ //runs through a loop to fill the first empty space

if ((dist\_final[0][i]==0)&&(set == 0)){ //if the ith element is 0 and nothing has been set yet

dist\_final[0][i] = a[0]; //set the element the distance just read

set = 1; //set a flag

}

}

}

if ((dist\_final[0][4]!=0)&&(avg == 0)){ //if the last element is not 0 (array is full)

avg = ((dist\_final[0][0]+dist\_final[0][1]+dist\_final[0][2]+dist\_final[0][3]+dist\_final[0][4])/5);

}

if ((dist\_final[0][4]!=0)&&(avg != 0)){

}\*/

}

if (b==2){

if (RB6 == 1){ //If ECHO is HIGH

TMR1ON = 1;

} //Start Timer

if (RB6 == 0){ //If ECHO is LOW

TMR1ON = 0; //Stop Timer

a[1] = (TMR1L | (TMR1H<<8))/58.82; //Calculate Distance

// a[1] = (TMR1L | (TMR1H<<8))\*0.034/2;

}

}

if (b==4){

if (RB4 == 1){ //If ECHO is HIGH

TMR1ON = 1;

} //Start Timer

if (RB4 == 0){ //If ECHO is LOW

TMR1ON = 0; //Stop Timer

a[3] = (TMR1L | (TMR1H<<8))/58.82; //Calculate Distance

// a[3] = (TMR1L | (TMR1H<<8))\*0.034/2;

}

}

if (b==3){

if (RB5 == 1){ //If ECHO is HIGH

TMR1ON = 1;

} //Start Timer

if (RB5 == 0){ //If ECHO is LOW

TMR1ON = 0; //Stop Timer

a[2] = (TMR1L | (TMR1H<<8))/58.82; //Calculate Distance

// a[2] = (TMR1L | (TMR1H<<8))\*0.034/2;

}

}

if (b==4){

if (RB4 == 1){ //If ECHO is HIGH

TMR1ON = 1;

} //Start Timer

if (RB4 == 0){ //If ECHO is LOW

TMR1ON = 0; //Stop Timer

a[3] = (TMR1L | (TMR1H<<8))/58.82; //Calculate Distance

// a[3] = (TMR1L | (TMR1H<<8))\*0.034/2;

}

}

// print\_echo();

}

void trig(){

for(b=1; b<5; b++){

TMR1H = 0; //Sets the Initial Value of Timer

TMR1L = 0; //Sets the Initial Value of Timer

// RBIE = 0;

if(b==1){

RBIE = 0;

RD1 = 1; //TRIGGER HIGH

\_\_delay\_us(10); //10uS Delay

RD1 = 0; //TRIGGER LOW

// \_\_delay\_ms(100); //Waiting for ECHO

RBIE = 1;

\_\_delay\_ms(us\_delay); //Waiting for ECHO

//

// if (echoo){

// echo();

// }

}

if(b==2){

RBIE = 0;

RD0 = 1; //TRIGGER HIGH

\_\_delay\_us(10); //10uS Delay

RD0 = 0; //TRIGGER LOW

// \_\_delay\_ms(100); //Waiting for ECHO

RBIE = 1;

\_\_delay\_ms(us\_delay); //Waiting for ECHO

// if (echoo){

// echo();

// }

}

if(b==3){

RBIE = 0;

RB3 = 1; //TRIGGER HIGH

\_\_delay\_us(10); //10uS Delay

RB3 = 0; //TRIGGER LOW

// \_\_delay\_ms(100); //Waiting for ECHO

RBIE = 1;

\_\_delay\_ms(us\_delay); //Waiting for ECHO

// if (echoo){

// echo();

// }

}

if(b==4){

RBIE = 0;

RB2 = 1; //TRIGGER HIGH

\_\_delay\_us(10); //10uS Delay

RB2 = 0; //TRIGGER LOW

// \_\_delay\_ms(100); //Waiting for ECHO

RBIE = 1;

\_\_delay\_ms(us\_delay); //Waiting for ECHO

// if (echoo){

// echo();

// }

}

// \_\_delay\_ms(100); //Waiting for ECHO

// RBIE =1;

// \_\_delay\_ms(100); //Waiting for ECHO

}

}

void print\_echo(){

lcd\_clear();

if(a[0]>=2 && a[0]<=400){ //Check whether the result is valid or not

printf("%d,%d,%d,%d,%d",a[0],dist\_final[0][0],dist\_final[0][5],sum[0],meas);

}

if(!(a[0]>=2 && a[0]<=400)){

printf("Distance 1 = X");

}

if(a[1]>=2 && a[1]<=400){ //Check whether the result is valid or not

lcd\_set\_ddram\_addr(LCD\_LINE2\_ADDR);

printf("Distance 2 = %d",a[1]);

}

if(!(a[1]>=2 && a[1]<=400)){

lcd\_set\_ddram\_addr(LCD\_LINE2\_ADDR);

printf("Distance 2 = X");

}

if(a[2]>=2 && a[2]<=400){ //Check whether the result is valid or not

lcd\_set\_ddram\_addr(LCD\_LINE3\_ADDR);

printf("Distance 3= %d",a[2]);

}

if(!(a[2]>=2 && a[2]<=400)){

lcd\_set\_ddram\_addr(LCD\_LINE3\_ADDR);

printf("Distance 3 = X");

}

if(a[3]>=2 && a[3]<=400){ //Check whether the result is valid or not

lcd\_set\_ddram\_addr(LCD\_LINE4\_ADDR);

printf("Distance 4= %d",a[3]);

}

if(!(a[3]>=2 && a[3]<=400)){

lcd\_set\_ddram\_addr(LCD\_LINE4\_ADDR);

printf("Distance 4 = X");

}

}

void main(){

TRISB = 0b11110000; //RB5 6 as Input PIN (ECHO)

TRISD = 0x00; // LCD Pins as Output

GIE = 1; //Global Interrupt Enable

RBIF = 0; //Clear PORTB On-Change Interrupt Flag

RBIE = 1; //Enable PORTB On-Change Interrupt

ADCON1=0x0F;

TRISBbits.RB0 = 0;

LATBbits.LATB0 = 1; //ENABLES KPD

initLCD();

lcd\_clear();

T1CON = 0x10; //Initialize Timer Module

while(1){

for (meas=0; meas<5;meas++){

trig();

// \_\_delay\_ms(100);

RBIE = 0;

// \_\_delay\_ms(100);

print\_echo();

// \_\_delay\_ms(100);

RBIE = 1;

// \_\_delay\_ms(100);

}

}

}