//Check Ultrasonic sensor for signal for pole or tire position. \*\*Note one pin IN and one OUT for sensor.

// Provide TRIGGER to ultrasonic module

// Listen for Echo

// Start Timer when ECHO HIGH is received

// Stop Timer when ECHO goes LOW

// Read Timer Value

// Convert it to Distance

// Display it

//This code runs two ultrasonic sensors. Tested. 3/10: 10:57pm

#include <xc.h>

#include <stdio.h>

#include "configBits.h"

#include "I2C.h"

#include "lcd.h"

int a = 0;

int dist\_final[3]; // distance array

#define RS RD2

#define EN RD3

#define D4 RD4

#define D5 RD5

#define D6 RD6

#define D7 RD7

//#include <xc.h>

//#include "lcd.h";

//#include <pic16f877a.h>

//// BEGIN CONFIG

//#pragma config FOSC = HS

//#pragma config WDTE = OFF

//#pragma config PWRTE = OFF

//#pragma config BOREN = ON

//#pragma config LVP = OFF

//#pragma config CPD = OFF

//#pragma config WRT = OFF

//#pragma config CP = OFF

////END CONFIG

int a1;

int a2;

int b;

void \_\_interrupt() echo(){

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

{

RBIE = 0; //Disable On-Change Interrupt

if (b==1){

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

TMR1ON = 1;

} //Start Timer

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

{

TMR1ON = 0; //Stop Timer

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

a1 = (TMR1L | (TMR1H<<8))\*0.034/2;

}

}

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 = (TMR1L | (TMR1H<<8))/58.82; //Calculate Distance

a2 = (TMR1L | (TMR1H<<8))\*0.034/2;

}

}

// RBIE = 0; //Disable On-Change Interrupt

// if ((RB5 == 1) || (RB6 == 1) ){ //If ECHO is HIGH

// TMR1ON = 1;

// } //Start Timer

// if((RB5 == 0) || (RB6 == 0) ) //If ECHO is LOW

// {

// TMR1ON = 0; //Stop Timer

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

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

//

// }

}

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

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

}

void main(){

TRISB = 0b01100000; //RB5 6 as Input PIN (ECHO)

TRISD = 0b00000000; // LCD Pins as Output

GIE = 1; //Global Interrupt Enable

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

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

initLCD();

// int b;

\_\_delay\_ms(3000);

lcd\_clear();

T1CON = 0x10; //Initialize Timer Module

while(1){

for(b=1; b<3; b++){

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

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

if(b==1){

RB3 = 1; //TRIGGER HIGH

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

RB3 = 0; //TRIGGER LOW

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

//a = a + 1; //Error Correction Constant

// printf("Distance = %d",a);

}

// if(a1>=2 && a1<=400){ //Check whether the result is valid or not

// lcd\_clear();

// printf("Distance = %d",a1);

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

// printf("%d", a1%10 + 48);

// }

// else{

// lcd\_clear();

// printf("Out of Range");

// }

if(b==2){

RD0 = 1; //TRIGGER HIGH

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

RD0 = 0; //TRIGGER LOW

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

//a = a + 1; //Error Correction Constant

// printf("Distance = %d",a);

}

lcd\_clear();

if(a1>=2 && a1<=400){ //Check whether the result is valid or not

lcd\_clear();

printf("Distance = %d",a1);

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

// printf("%d", a1%10 + 48);

}

if(a2>=2 && a2<=400) //Check whether the result is valid or not

{

// lcd\_clear();

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

printf("Distance 2= %d",a2);

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

// printf("%d", a2%10 + 48);

// a = a/10;

// Lcd\_Set\_Cursor(1,13);

// Lcd\_Write\_Char(a%10 + 48);

//

// a = a/10;

// Lcd\_Set\_Cursor(1,12);

// Lcd\_Write\_Char(a%10 + 48);

//

// Lcd\_Set\_Cursor(1,15);

// Lcd\_Write\_String("cm");

}

// else

// {

// lcd\_clear();

//

// printf("Out of Range");

// }

\_\_delay\_ms(400);

}

}

}

//void main(void)

//{

// int a;

//

// TRISB = 0b00100000; //RB4 as Input PIN (ECHO)

// TRISD = 0x00; // LCD Pins as Output

//

//

//

// T1CON = 0x10; //Initialize Timer Module

//

//

// while(1)

// {

// initLCD();

//

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

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

//

// RB0 = 1; //TRIGGER HIGH

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

// RB0 = 0; //TRIGGER LOW

////

// while(!RB5); //Waiting for Echo

// TMR1ON = 1;

//// printf("yhh");

//// } //Timer Starts

//

// \_\_delay\_ms(1000);

// lcd\_clear();

// printf(" what;er");

// while(RB5); //Waiting for Echo goes LOW

// TMR1ON = 0; //Timer Stops

// lcd\_clear();

// printf(" fadsg: %d");

// a = (TMR1L | (TMR1H<<8)); //Reads Timer Value

// a = a/58.82; //Converts Time to Distance

// a = a + 1;

//lcd\_clear();

// printf(" Distance: %d", a);

// //Distance Calibration

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

// lcd\_clear();

// printf(" Distance: %d", a);

// for (int i=0; i<3; i++){

// dist\_final[i] = a%10 + 48;

// a = a/10;

// }

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

// printf("%d%d%d", dist\_final[2],dist\_final[1],dist\_final[0]);

// }

// else {

// lcd\_clear();

// printf(" out of range: ");

// }

// \_\_delay\_ms(400);

// }

//}