// Reflectance.c

// Provide functions to take measurements using a QTR-8RC reflectance

// sensor array (Pololu part number 961). This works by outputting to

// the sensor, waiting, then reading the digital value of each of the

// eight phototransistors. The more reflective the target surface is,

// the faster the voltage decays.

// Student version of GPIO lab

// Daniel and Jonathan Valvano

// July 2, 2017

/\* This example accompanies the books

"Embedded Systems: Introduction to the MSP432 Microcontroller",

ISBN: 978-1512185676, Jonathan Valvano, copyright (c) 2017

"Embedded Systems: Real-Time Interfacing to the MSP432 Microcontroller",

ISBN: 978-1514676585, Jonathan Valvano, copyright (c) 2017

"Embedded Systems: Real-Time Operating Systems for ARM Cortex-M Microcontrollers",

ISBN: 978-1466468863, , Jonathan Valvano, copyright (c) 2017

For more information about my classes, my research, and my books, see

http://users.ece.utexas.edu/~valvano/

Simplified BSD License (FreeBSD License)

Copyright (c) 2017, Jonathan Valvano, All rights reserved.

Redistribution and use in source and binary forms, with or without modification,

are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice,

this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice,

this list of conditions and the following disclaimer in the documentation

and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"

AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE

IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE

ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE

LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL

DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;

LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED

AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,

OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE

USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The views and conclusions contained in the software and documentation are

those of the authors and should not be interpreted as representing official

policies, either expressed or implied, of the FreeBSD Project.

\*/

// reflectance LED illuminate connected to P5.3

// reflectance sensor 1 connected to P7.0 (robot's right, robot off road to left)

// reflectance sensor 2 connected to P7.1

// reflectance sensor 3 connected to P7.2

// reflectance sensor 4 connected to P7.3 center

// reflectance sensor 5 connected to P7.4 center

// reflectance sensor 6 connected to P7.5

// reflectance sensor 7 connected to P7.6

// reflectance sensor 8 connected to P7.7 (robot's left, robot off road to right)

**#include** <stdint.h>

**#include** "msp432.h"

**#include** "..\inc\Clock.h"

// ------------Reflectance\_Init------------

// Initialize the GPIO pins associated with the QTR-8RC

// reflectance sensor. Infrared illumination LEDs are

// initially off.

// Input: none

// Output: none

**void** **Reflectance\_Init**(**void**){

// write this as part of Lab 6

//Clock\_Init48MHz();

P5->SEL0 &= ~0x08; //P5.3 set to GPIO

P5->SEL1 &= ~0x08; //P5.3 set to GPIO

P5->DIR |= 0x08; //P5.3 set to output

P5->OUT &= ~0x08; //P5.3 set to low

P9->SEL0 &= ~0x04;

P9->SEL1 &= ~0x04;

P9->DIR |= 0x04;

P9->OUT &= ~0x04;

P7->SEL0 &= ~0xFF; //P7.0 set to GPIO

P7->SEL1 &= ~0xFF; //P7.0 set to GPIO

//P7->DIR &= ~0xFF; //P7.0 set to input

}

/\*

int main(void){

Reflectance\_Init();

while(1){

P5->OUT |= 0x01; //P5.3 set to high

P7->DIR |= 0x01;

P7->OUT |= 0x01;

Clock\_Delay1us(10);

P7->DIR |= 0x00;

for(int i = 0; i < 10000; i++){

P4->OUT &= P7->IN&0x01;

}

P5->OUT &= ~0x01;

Clock\_Delay1ms(10);

}

}

\*/

// ------------Reflectance\_Read------------

// Read the eight sensors

// Turn on the 8 IR LEDs

// Pulse the 8 sensors high for 10 us

// Make the sensor pins input

// wait t us

// Read sensors

// Turn off the 8 IR LEDs

// Input: time to wait in usec

// Output: sensor readings

// Assumes: Reflectance\_Init() has been called

uint8\_t **Reflectance\_Read**(uint32\_t time){

//uint8\_t result;

// write this as part of Lab 6

uint8\_t result;

//P5->SEL0 &= ~0x08;

//P5->SEL1 &= ~0x08;

//P5->DIR |= 0x08; //set 5.3 as output

P5->OUT |= 0x08; //set 5.3 as high

//P9->DIR |= 0x04; //set 9.2 as output

P9->OUT |= 0x04; //set 9.2 as high

//P7->SEL0 &= ~0xFF;

//P7->SEL1 &= ~0xFF;

//Clock\_Delay1us(10);

P7->DIR |= 0xFF; //set P7.7-P7.0 as output

P7->OUT |= 0xFF; //set P7.7-P7.0 as high

Clock\_Delay1us(10);

P7->DIR = 0x00; //set P7.7-P7.0 as input

Clock\_Delay1us(time);

result = (P7->IN & 0xFF);

P5->OUT &= ~0x08; //turn off IR LED

P9->OUT &= ~0x04; //turn off IR LED

**return** ~result;

}

// ------------Reflectance\_Center------------

// Read the two center sensors

// Turn on the 8 IR LEDs

// Pulse the 8 sensors high for 10 us

// Make the sensor pins input

// wait t us

// Read sensors

// Turn off the 8 IR LEDs

// Input: time to wait in usec

// Output: 0 (off road), 1 off to left, 2 off to right, 3 on road

// (Left,Right) Sensors

// 1,1 both sensors on line

// 0,1 just right off to left

// 1,0 left left off to right

// 0,0 neither lost

// Assumes: Reflectance\_Init() has been called

uint8\_t **Reflectance\_Center**(uint32\_t time){

// write this as part of Lab 6

**return** 0; // replace this line

}

// Perform sensor integration

// Input: data is 8-bit result from line sensor

// Output: position in 0.1mm relative to center of line

int32\_t **Reflectance\_Position**(uint8\_t data){

// write this as part of Lab 6

/\*

P5->DIR |= 0x08;

P5->OUT |= 0x08; //turn on IR LED

P7->DIR = 0xFF; //set P7.7-P7.0 as output

P7->OUT = 0xFF; //set P7.7-P7.0 as high

Clock\_Delay1us(10);

P7->DIR = 0x00; //set P7.7-P7.0 as input

\*/

**int** W[8] = {-332, -237, -142, -47, 47, 142, 237, 332};

uint8\_t Mask[8] = {1, 2, 4, 8, 16, 32, 64, 128};

**int** sum, count, result;

sum = 0;

count = 0;

result = 0;

**for**(**int** i = 0; i < 8; i++){

**if**(data & Mask[i] ){

count++;

sum = sum + W[i];

}

}

**if**(data == 0){

**return** 333;

}

**else**{

result = sum / count;

**return** result;

}

// replace this line

}

// ------------Reflectance\_Start------------

// Begin the process of reading the eight sensors

// Turn on the 8 IR LEDs

// Pulse the 8 sensors high for 10 us

// Make the sensor pins input

// Input: none

// Output: none

// Assumes: Reflectance\_Init() has been called

**void** **Reflectance\_Start**(**void**){

// write this as part of Lab 10

P5->OUT |= 0x08; //set 5.3 as high

P9->OUT |= 0x04; //set 9.2 as high

P7->DIR |= 0xFF; //set P7.7-P7.0 as output

P7->OUT |= 0xFF; //set P7.7-P7.0 as high

Clock\_Delay1us(10);

P7->DIR = 0x00; //set P7.7-P7.0 as input

}

// ------------Reflectance\_End------------

// Finish reading the eight sensors

// Read sensors

// Turn off the 8 IR LEDs

// Input: none

// Output: sensor readings

// Assumes: Reflectance\_Init() has been called

// Assumes: Reflectance\_Start() was called 1 ms ago

uint8\_t **Reflectance\_End**(**void**){

// write this as part of Lab 10

Clock\_Delay1us(1000);

uint8\_t end;

end = (P7->IN & 0xFF);

P5->OUT &= ~0x08; //turn off IR LED 5.3

P9->OUT &= ~0x04; //turn off IR LED 9.2

**return** ~end; // replace this line

}