/\*

\* File: Blink.c

\* Author: petrusn & ahtouw

\*

\* Created on September 26, 2016, 8:56 AM

\*/

#include <stdio.h>

#include <stdlib.h>

#include <plib.h>

/\* all ports have names: \*/

/\* e.g. IOPORT\_A IOPORT\_B, etc. \*/

/\* all port bits have names: \*/

/\* e.g. BIT\_0, BIT\_1,... BIT\_14, BIT\_15, etc. \*/

/\* led 1 port RB10 \*/

/\* led 2 port RB11 \*/

/\* led 3 port RB12 \*/

/\* led 4 port RB13 \*/

/\*

\* This program is to control the motors on the MRK-LINE Robot, using feedback

\* from the optical sensors in the line-following accessory.

\* Here are the pertinent port connections:

\* Left wheel (assuming sensors are on front bumper)

\* DIR - RD6 (1: forward)

\* EN - RD2 active high

\* SA - RD10 SA and SB are quadrature encoded feedback

\* SB - RC2

\* Right wheel

\* DIR - RD7 (0: forward)

\* EN - RD1

\* SA - RD9

\* SB - RC1

\* Optical Sensors

\* S1 - RB0 S1 is rightmost sensor

\* S2 - RB1 S2 is middle right sensor

\* S3 - RB2 S3 is middle left sensor

\* S4 - RB3 S4 is leftmost sensor

\*/

int main(int argc, char\*\* argv) {

INTConfigureSystem(INT\_SYSTEM\_CONFIG\_MULT\_VECTOR);

INTEnableInterrupts();

//Configure ports for onboard LEDs as outputs

PORTSetPinsDigitalOut(IOPORT\_B, BIT\_10 | BIT\_11 | BIT\_12 | BIT\_13);

PORTSetPinsDigitalOut(IOPORT\_D, BIT\_6 | BIT\_2 | BIT\_7 | BIT\_1 );

PORTSetPinsDigitalIn(IOPORT\_C, BIT\_2 | BIT\_1);

// Useful functions: (see PeripheralLibraries pdf file for more)

// PORTSetPinsDigitalOut(IOPORT\_B, BIT\_10 | BIT\_11 | BIT\_12 | BIT\_13);

// PORTSetPinsDigitalIn(IOPORT\_B, BIT\_10 | BIT\_11 | BIT\_12 | BIT\_13);

// PORTClearBits(IOPORT\_B, BIT\_10 | BIT\_11 | BIT\_12 | BIT\_13); // clear all bits

// PORTSetBits(IOPORT\_B, BIT\_10 | BIT\_11 | BIT\_12 | BIT\_13); // set all bits

// PORTToggleBits(IOPORT\_B, BIT\_10 | BIT\_11 | BIT\_12 | BIT\_13); // toggle state of the bits

// Configure built in buttons as inputs

// On ProMX4 butons are on RA6 and RA7

PORTSetPinsDigitalIn(IOPORT\_B, BIT\_0 | BIT\_1 | BIT\_2 | BIT\_3);

PORTSetPinsDigitalIn(IOPORT\_A, BIT\_6 | BIT\_7);

// Useful function: see DigitalIO project for more

// PORTReadBits(IOPORT\_A, BIT\_6); // read the state of button on RA6

while (1) // continuous loop

{

if ((PORTReadBits(IOPORT\_B, BIT\_2)) == 0) // if middle-left sensor S3 detects black

{

PORTSetBits(IOPORT\_D, BIT\_1); //Right Wheel Enabled

PORTClearBits(IOPORT\_D, BIT\_2 | BIT\_7); // Left Wheel Disabled, Right Direction Forward

}

else if ((PORTReadBits(IOPORT\_B, BIT\_1)) == 0) // if middle-right sensor S2 detects black

{

PORTSetBits(IOPORT\_D, BIT\_2 | BIT\_6); // Left Wheel Enabled, Left Direction Forward

PORTClearBits(IOPORT\_D, BIT\_1); // Right Wheel Disabled

}

else

{

PORTSetBits(IOPORT\_D, BIT\_1 | BIT\_2 | BIT\_6);

PORTClearBits(IOPORT\_D, BIT\_7);

}

}

return (EXIT\_SUCCESS);

}

Note:

* We used the two middle sensors to detect black line
* We only enabled one wheel at a time if a black line is detected by either one of the sensors
* We demonstrated that our program works