/\* File: pwm8bits.c

\* Author: Craig Bacon

\* Created on 30 October 2014, 14:11

\* PWM pin 13 right hand motor, mark space ratio is 8 bit (255) and the PWM

\* frequency is 610 Hz PWM Period = (PR2+1)x4 x 1/Osc x TMR2 Prescaler

\* e.g (255+1)x4 x1/10Mhz x 16 =1.638ms or 610Hz.PR2 changes frequency.

\*

\* Need to control pins 13(CCP1) 21(RB0) and 22(RB1)

\*/

#include <xc.h>

#include <stdio.h>

#include <stdlib.h>

#pragma config OSC = HS

#pragma config WDT = OFF

#pragma config LVP = OFF

#pragma config PWRT = ON

#define LED1 LATBbits.LATB2 //LED1

#define LED2 LATBbits.LATB3 //LED2

#define LED3 LATBbits.LATB4 //LED3

#define LED4 LATBbits.LATB5 //LED4

#define RA4 LATAbits.LATA4

#define RA5 LATAbits.LATA5

#define RB0 LATBbits.LATB0

#define RB1 LATBbits.LATB1

#define \_XTAL\_FREQ 10000000 // define clock frequency for \_\_delay\_10ms()

#define RA2 PORTAbits.RA2 // left beacon sensor

#define RA3 PORTAbits.RA3 // right beacon sensor

void wait10ms(int del); //generates a delay in multiples of 10ms

int main(void)

{

int markspace=127; //mark space value for 8 PWM (50% mark space ratio)

TRISCbits.RC1=0; //set CCP1(pin13) to an output pin

TRISCbits.RC2=0; //set CCP1(pin13) to an output pin

TRISA = 0b11001111;

TRISB = 0b00000000;

PR2 = 0b11111111 ; //set period of PWM

T2CON = 0b00000111 ; //Timer 2(TMR2) on, Prescaler = 16

CCP2CON = 0b00001100; //0x0c enables PWM module CCP1

CCP1CON = 0b00001100; //0x0c enables PWM module CCP1

CCPR2L = markspace; //Load duty cycle into CCP1CON, PWM begins

CCPR1L = markspace; //Load duty cycle into CCP1CON, PWM begins

ADCON1 = 0b00001101; // disable analogue inputs to enable digital inputs

RA4 = 1;

RA5 = 1;

RB0 = 1;

RB1 = 1;

LED1 = 0;

LED2 = 0;

LED3 = 0;

LED4 = 0;

while(1){

while (RA2 == 0)

{

LED1 = 1;

LED2 = 0;

}

while (RA2 == 1)

{

LED1 = 0;

LED2 = 1;

}

while (RA3 == 0)

{

LED3 = 1;

LED4 = 0;

}

while (RA3 == 1)

{

LED3 = 0;

LED4 = 1;

}

};

} //end main()

void wait10ms(int del){ //delay function

unsigned char c;

for(c=0;c<del;c++)

\_\_delay\_ms(10);

return;

}