/\*\*

\* ECPE 155 Autonomous Robotics

\* Spring 2018

\* Lab 2 - Pulse Width Modulation (PWM)

\* Lab Associates: Paul Vuong

\* Steve Guerro

\* 1. To drive the HUB-ee motors/wheels, the PWM modules need to be configured for

\* dead-band generation.

\* 2. PWM Generator 1 will supply a signal to the right wheel.

\* 3. PWM Generator 2 will supply a signal to the left wheel.

\* 4. In addition to the PWM dead-band signals, two additional digital control signals

\* will give added functionality to the wheels.

\* 5. Digital signals will be sourced from the many GPIO ports as a digital output

\* 6. The additional digital inputs will allow the motor to move forwards, backwards and stop.

\*/

**#include** <stdbool.h>

**#include** <stdint.h>

**#include** "SYSCTL.h"

**#include** "GPIO.h"

**#include** "OSC.h"

**#include** "PWM.h"

**#include** "M1PWMn.h"

// Control interface system prototypes initialization

**void** **initGPIO**(**void**);

**void** **moveForward**(uint32\_t in1, uint32\_t in2);

**int** **main**(**void**)

{

//Initiate Oscillator. Direct register writes to RCC and RCC2 register.

//Establish system clock and enable both system clocks and PWM clock.

//Use of PWM module 1 will require register writes to RCGCPWM register.

initOSC();

//Initiate GPIOs. Enable clocks to ports D and E (RCGCGPIO). PD1 and PE4 will

//be used for the PWM output pins while PE1, PE2, PE3, and PE5 are the digital

//signals out to the HUBee wheels.

initGPIO();

//Initiate PWM module. Assign clock signal to PWM module (RCGCPWM).

//1. Load start value into PWMnCTL register

//2. Set pulse width values in PWMnGENA and PWMnGENB registers

//3. From PWM clock signal determine the amount of clock ticks and load value

// into PWMnLOAD register.

//4. Set duty cycles in PWMnCMPA and PWMnCMPB register

//5. Start the timers in PWMnCTL register.

//6. Enable PWM outputs PWMENABLE register.

initM1PWMn();

**while**(1){

moveForward(0, 1);

wait();

moveForward(1, 1);

wait();

}//end main while loop

}