Steve Guerrero

ECPE 155

Pre-lab 3

The QEI you should be able to determine the direction of rotation of the wheels and the wheels position given the number of pulses. Since there are 128 pulses per revolution, you add in a timer interrupt and you should be able to acquire phi or wheel speed for each wheel. From there you can find the angular velocity and the forward velocity using the kinematic formulas. Then you can calculate the robots final pose.

/\* Write pseudocode to implement this approach, identifying where you will want specific functions \*/

/\*

// Initialize the QEI peripheral in the TM4C123

// Enable sysctl clock module for QEI

SYSCLOCK\_QEI\_ENABLE();

// Enable clock for the appropriate GPIO module

RCGC\_GPIO\_ENABLE();

// Choose AFSEL for desired pins

// Set port control for the chosen pins

Config\_Pin\_type ();

// configure QEI to capture edges for both A & B signals

// write to QEICTL register for reset mode and capture mode

QEICTL |= ( 0x18 << 0 ); // sets reset mode and capture mode

// write to QEIMAXPOS register for # of pulses per revolution

MaxPosSet(QEI\_BASE0[QEIMAXPOS] |= ( 0x80 << 0 )); // sets max position to

// 128 pulses per rev

// Enable QEI module by setting bit 0 in QEICTL

QEIEnable(QEI\_BASE0);

// Configure QEI Timer along with QEIMAXPOS above to get the wheel speed Phi

QEIConfigTimer();

// Once everything is configured you can get the wheel speed along

// with the direction of the wheel’s rotation

QEIGetSpeed();

QEIGetDir();

// With the acquired wheel speed ( Phi ) you can use the kinematic

// equations to find the robot’s current position and determine where

// the robot would need to go next.

Change\_direction(float phi)

// depending on the targets position form the robot’s current position the robot will rotate to aim at the target.

Robot\_forward(int time\_in\_ms, float phi)

// the robot will travel forward using the kinematics to monitor its current position and check if it matches the target’s position.

