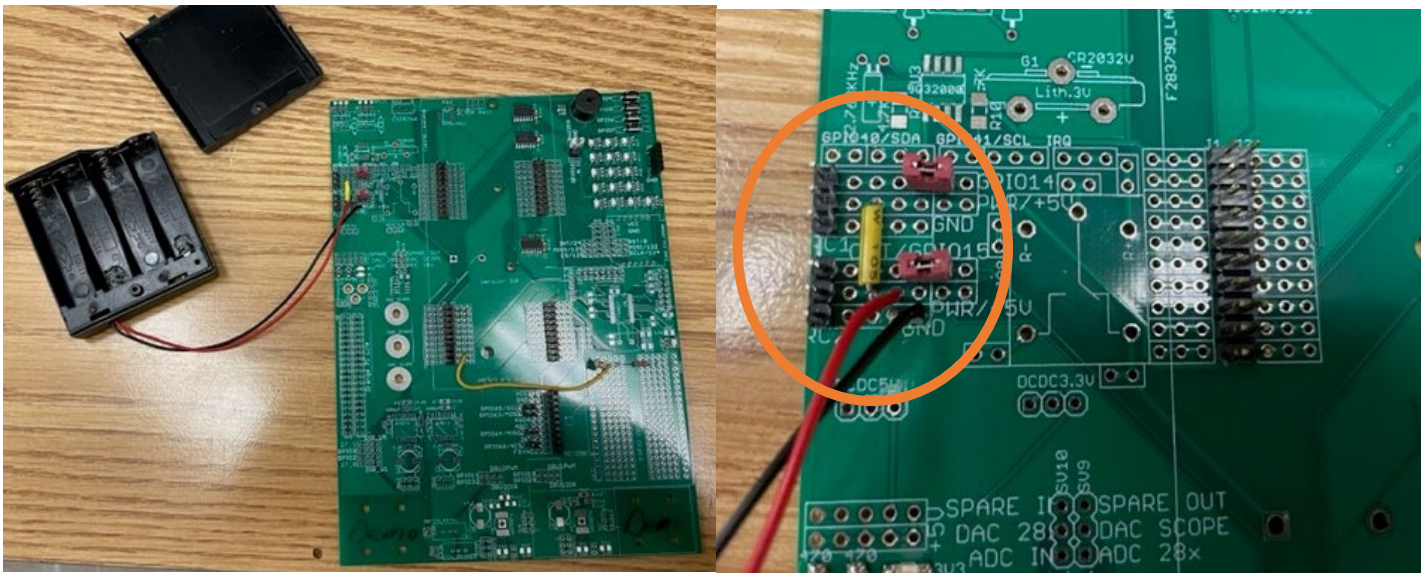
SE 423 Mechatronics Homework Assignment #4

Spring 2024, Due In Lecture April 3rd. The Microcontroller Demonstration Check-Off for Questions 2 and 3 are Due 5pm Tuesday March 26th. The Demonstration Check-Off for Question 4 is Due 5pm Tuesday April 2nd.

1. Give yourself a small introduction working in Linux, especially at the command prompt, by reading through the help at http://community.linuxmint.com/tutorial/view/244. Answer the following questions:
    a. What command along with an option would you type to list all the files in a directory and more information like file date and file size.
    b. What command is used to change to a new directory? What command displays the directory you are currently in?
    c. What command is used to copy a file in Linux? In the /home/pi directory there is a file named "hw4data.dat". Also in the /home/pi directory there is a directory named "hw4". What text would you type to copy "hw4data.dat" to the directory "hw4" with the new name "hw4trial1.dat?"
    d. What does the "less" and "cat" commands do? How are they different?
    e. What do the "ifconfig" and "ping" commands do?
    f. What command line text will set the date to 2:30PM April 25th, 2024 using the "date" command?
2. **Soldering.** Question 3 is going to have you program EPWM8 to drive two RC servos with it's A and B PWM outputs. RC servos are position controlled geared motors and do require a bit of current, especially if they become stalled. For that reason, it is not the best idea to power the RC servos with the USB power of your PC. Instead I am giving you a battery pack that holds 4 AA batteries. (You will have to purchase your own AA batteries.) Each battery is around 1.5V so the RC servos will be powered with 6V. See the picture below to find where to solder the battery pack leads, the 2X1 headers connecting GPIO14 and 15 to the RC servo and the 3X1 headers for plugging the RC servos into your HW board.



3. RC servos are popular in RC airplane and RC cars. RC servo motors are devices that you can command to move to a desired angle. Normally they only have a range of -90 degrees to 90 degrees. To command these motors, a PWM signal with a slow carrier frequency of 50Hz (20 millisecond period) is used. Then to command the angle of the motor you change the duty cycle

commanding the motor between about 4% duty to 12% duty cycle. -90 degrees is approximately 4% duty cycle, 0 degrees is close to 8% duty cycle and +90 degrees is close to 12% duty cycle. Any other angle desired is linear in between those values. First, modify the initialization code for these two EPWM channels. Looking at your breakout board's labeling you should see that GPIO14 (EPWM8A) and GPIO15 (EPWM8B) are the pins connected to the RC Servo "3 pin" connectors. In homework #2 and lab #3 you setup PWM channel A to drive an LED and then drive the two motors of the robot. In this exercise you will be setting up EPWM8 in a similar fashion but now you also have to setup the B output of the EPWM8 peripheral. So first get the EPWM8A output working. You can copy the setup code you used for EPWM12 in HW#2 as a start to the setup for EPWM8. But you will need to change the carrier frequency of the PWM to 50Hz instead of the 5000Hz we used in HW#2. Remember here when you set the carrier frequency that the TBPRD register is only 16 bits so its value cannot be greater than 65535. So in other words, you may need to set CLKDIV to something else than 0. Before you plug in your RC servo to your green board, use a digital channel of the oscilloscope to scope the PWM signal and make sure it is the signal you expect. Then when you have EPWM8A working setup EPWM8B. Most of the setups for EPWM8A also effect EPWM8B. For example 8A and 8B have to use the same carrier frequency since there is only one TBPRD register. The only additional registers you have to use for EPWM8B are AQCTLB and CMPB. AQCTLB is set very similar to AQCTLA but NOTE there is a CBU event. So setup EPWM8 so that it has the required RC Servo carrier frequency of 50Hz and start CMPA and CMPB's value such that it is commanding the RC Servos to 0 degrees (8% duty cycle).

Similar to the PWM functions you created in Lab #3, create two functions

"void setEPWM8A_RCServo(float angle)" and "void setEPWM8B_RCServo(float angle)".

The parameter "angle" is a value between -90 and 90 degrees where -90 equates to 4% duty cycle, 0 equates to 8% duty cycle and 12% equates to 90. Make sure to first saturate "angle" between -90 and 90 just in case a value outside of the range is passed. Test that your functions work by writing some code that gradually changes the value passed to "angle" so that the RC servo is driven back and forth. If you are only given one RC servo, make sure to plug it into both "3 pin" RC servo connectors to make sure both your EPWM8A and EPWM8B functions are working. **Show this working code to your TA.**

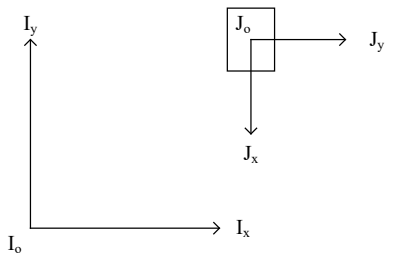
4. For this exercise I would like you to look into the external interrupt capabilities of the F28379D pins and also give you another exercise in merging code from one piece of example code into another project's code. The F28379D allows you to setup any of its GPIO pins as an input and in addition can generate an interrupt in the F28379D when the signal entering the GPIO pin transitions from High to Low (or Low to High, you can pick). The best use case for this is an external chip needing to tell the F28379D that it has new data and wishes to be communicated with. An external slave chip that uses SPI to communicate with the master is a perfect use case for this. If you remember with SPI the only way the master receives data from a slave chip is for the master to send data to the slave. Using an external interrupt the slave chip could set a signal high indicating that it had new data. This signal would come into a F28379D GPIO and cause an interrupt. In the interrupt function could issue the SPI send/receive commands to read the new data from the slave device. For this exercise, instead of having an external SPI slave chip interrupt the F28379D, we are going to have two of the push buttons on your green board cause two different external interrupts. In the same folder as HWstarter and Labstarter there is another starter project called XINTstarter. (XINT is eXternal INTerrupt) Go ahead and create a new project using this project starter. Build and run the program. Pull up Tera Term and notice that the number of times you have pressed push button 1 and push button 2 is displayed. Press buttons 1 and 2 a number of time and see how many counts occur. These are cheap push buttons and they can have contact bounce that could cause more than one interrupt each time your press the button.

   Study the external interrupt starter code and then change the code so that instead of PB1 and PB2 causing the interrupts, have PB3 and PB4 cause the two interrupts XINT1 and XINT2. Then as one final exercise to give you some more experience merging example code, move all necessary code from this application that causes external interrupts when PB3

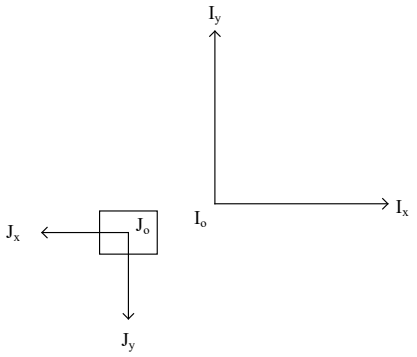
and PB4 are pressed into your RC servo problem 3 project. Your final code should be able to drive the two RC servos and also display the number of times PB3 and PB4 have been pressed. You do not have to, but you could do something fun like move the RC servo to a new position each time PB3 is pressed. **Show this working code to your TA.**

5. Obtain the rotation matrix, which converts base J's coordinates into base I's coordinates (R_J^I), for the following cases:

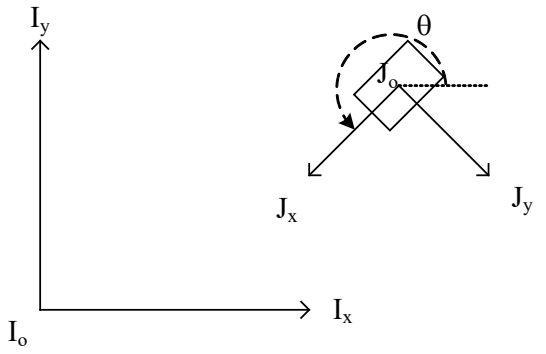
a.



b.



c.



6. Given the following translation vectors:
   a. $\overrightarrow{I_0J_0}$= (3.83,-1.75,0)$_I$
   b. $\overrightarrow{I_0J_0}$= (11.1,6.7,0)$_I$
   c. $\overrightarrow{I_0J_0}$ = (5.3,-1.4,0)$_I$

and using the results on the previous a, b, and c questions respectively, obtain the homogeneous transformation matrix, which coverts frame J coordinates to frame I coordinates.

Using the homogeneous transformation matrix found in part c. above, solve the following two questions:

i. If a golf ball's coordinates are (2.8,-2.3,0) in the robot's (J) frame, what are the golf ball's coordinates in the world (I) frame as a function of θ? What are the golf ball's coordinates if θ equals 35°? To check that you are doing this problem correctly you should find that x is equal to 8.9129. Show your work and find this x value and also the y coordinate.

ii. If a golf ball's coordinates are (2.54,1.45,0) in the world (I) frame, what are the golf ball's coordinates in the robot (J) frame as a function of θ? Note that the inverse of a homogeneous transformation matrix $H = \begin{bmatrix} R & d \\ 0 & 1 \end{bmatrix}$ is $H^{-1} = \begin{bmatrix} R^T & -R^T d \\ 0 & 1 \end{bmatrix}$. If θ equals 120°, is the golf ball on the right or left side of the robot? Remember that robot positive x points straight ahead of the robot and therefore positive y points to the robot's left. To check that you are doing this problem correctly you should find that x is equal to 3.8482. Show your work and find this x value and also the y coordinate and determine if the ball is on the right or left.

7. This question is mainly a For Your Information item. You will not be soldering transistors or relays to your green board or writing any program to turn on and off a transistor or relay. All I want you to turn in for this problem is to read through this information and write me a synopsis of the information given here.

Look at the top left corner of your green board and see there are two spots for soldering TIP122 transistors to your green board. Also close to the transistor there is a spot for one relay. For this exercise you will learn how to wire/use both transistors and relays to allow the F28379D's digital outputs to turn on and off high current devices that the digital outputs alone would not be able to drive, and if attempted to drive these loads, would damage the digital outputs. Below explains and illustrates how to wire GPIO outputs to the given NPN transistors and then in turn use the high current output of the transistors to drive an ultra-bright LED and a relay.

If you are unfamiliar with transistors, refer to any introductory electronic circuits text or check the web; a good transistor switch tutorial is given at http://electronicsclub.info/transistorcircuits.htm. In a transistor, the (possibly large) collector-emitter current is controlled by the small base current through the relationship $I_C = h_{FE} I_B$. Since we are interested in using the transistors as switches, we will be operating in only the "on" (saturation) and "off" (cutoff) modes of the transistor. We can "turn off" the load circuit by setting $I_B = 0$. Likewise, if we know the desired current I_C and the gain h_{FE} (from datasheet), we can choose I_B so that the transistor is fully "on" (saturated). When the transistor is saturated, the voltage drop between collector and emitter is small and the collector-emitter junction can be modeled as a short circuit.

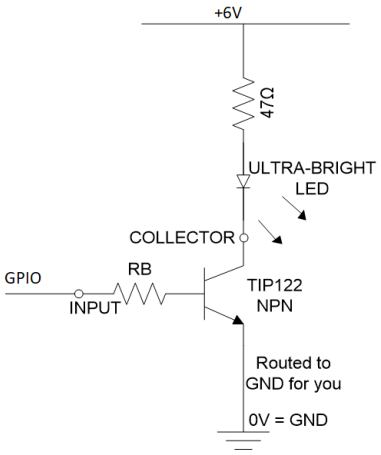


Figure 1

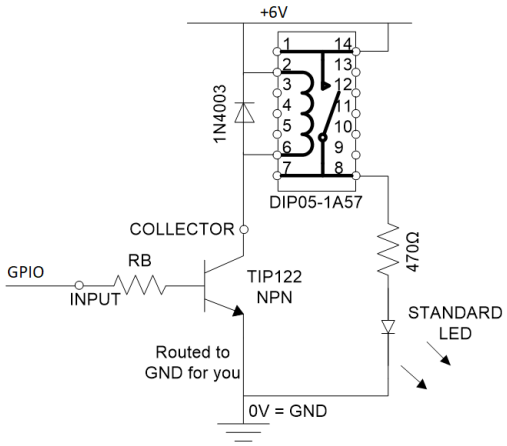


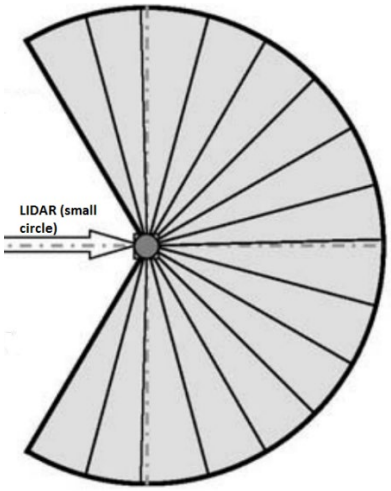
Figure 2

Refer to Figure 1 for the following example. If we want to use 5V to drive an ultra-bright LED in series with a 47Ω resistor, we can calculate. $I_B \geq \frac{I_C}{h_{FE}} = \frac{(6-0.7)V/47\Omega}{h_{FE}}$ For the TIP122 NPN Darlington Transistors you will use, $h_{FE} \geq 1000$. Therefore,

$I_B \geq 0.11mA$. To realize this current with our microcontroller's 3.3V output, we need to determine an appropriate value for the resistor R_B. When $V_B > 1.4V$, the base-emitter junction of a Darlington transistor behaves like a diode with a 1.4V drop. Therefore, the drop across R_B is $V_R = 3.3V-1.4V = 1.9V$. Now $R_B \leq \frac{V_R}{I_B} = \frac{1.9V}{.11} \approx 17k\Omega$ Therefore, any resistor with a value less than 17kΩ would suffice.

For the sake of universality, we will use a resistor with a much smaller value (470Ω) so that we are able to drive larger loads if the need arises. You are not asked to do this for this homework assignment but if you would want to use the transistor spots on your green board in the future follow these guidelines. Solder the two TIP122 chips onto the green board. Solder two 470Ω resistors in the places labeled "R_B". Also solder the LED circuit to the "COLLECT" terminal of one of the transistors as shown in Fig. 1. The only power you currently have to drive the transistor loads is the 6V battery pack voltage. Do not use the USB power, 3.3V and 5.0V to drive the transistor and relay loads. Solder the relay circuit shown in Fig. 2 to the "COLLECT" terminal of the second transistor. (Note the current rating of this small relay's switch is 0.5 Amps so obviously you could drive a larger current load than a standard LED.) Solder a GPIO to the INPUT of the ultra-bright LED circuit and a different GPIO to the INPUT of the relay circuit.

8. For this assignment I would like you to expand on the state machine and pseudocode of HW #3. Now instead of just having two distance readings from the LIDAR sensor, (one front distance and one front-right distance) you have access to the 228 distance readings from the LIDAR. The robot's LIDAR calculates 228 distance readings starting at -120 degrees on its right and sweeping to 120 degrees on its left. So the LIDAR produces a distance reading every 1.05 degrees starting at -120 and ending at 120. The LIDAR readings are given to you in an array of 228 elements. Element 113 is the distance reading straight in front of the robot. The below picture gives you an idea of the sweep of measurements, but note this picture is only showing 17 measurements.



Given these new measurements from the LIDAR, modify your obstacle avoidance section of your state machine with decisions and needed states to have the robot either left wall follow or right wall follow when it recognizes an obstacle with the LIDAR. Make sure to think about what you should do if you find an obstacle on your left, on your right and pretty much straight in front of you. As a final task, think about how you could recognize human legs and what would you have your robot do?

9. This homework assignment is up to you. Use your creativity to build something using one or two RC servo motors, the buzzer if you want to and one or more of the following sensors: photo sensor, MPU-9250 IMU, joystick, microphone (I will have to give you one) any other sensor you already own or want to purchase. You will be able to send me STL files that I will print on the lab's 3D printers. Make something that you will be proud of and put on your desk at home or something that will scare your friends when they try to raid your refrigerator. Anything goes but keep in mind that you also have a final project with the Robot to complete by the end of the semester. So in other words, I am not expecting it to be an elaborate and finely polished design.

Items that you **CAN** use that are in the Mechatronics Lab: *(This is not a complete list so ask if there is a part that you need).*

a. Any of the parts (resistors, capacitors, sensors, etc) used in this and previous homework assignments.
b. Anything (hardware, sensor, actuator or integrated circuit) that you purchase.
c. Raw plastic and aluminum, "Super Velcro", nuts and bolts. Cheap items that can be purchased from McMaster-Carr. I will be the judge of what is cheap.

Items that you **CANNOT** take from the lab.

a. Pretty much any of the pre-made parts for the RC servos. Unfortunately these items are relatively expensive and I can't give them to you.
b. IR and Ultrasonic sensors used by the Robot.
c. Gears, pulleys, belts. Ask though because I have some old ones you can have.
d. Other items? Ask before you plan on using them.

This assignment spans both HW #4 and HW #5 and HW #5 is only this project. Your finished product should be completed and checked off by the due date of HW #5 (April 19th).

What needs to be turned in for HW #4 question 9?

a. A description of what you are planning to build. What will it do? What parts are you going to need?
b. Mechanical drawings of your device. Pencil and paper is fine as long as it is neat.
c. A wiring schematic of all the electronic parts of your project. Pencil and paper is fine as long as it is neat.
d. Any source code you have developed to this point.