piRover Builds with K2

Digital Outputs – Driving High Current Loads Rev 1.0

Overview:

In the last activity we investigate the electrical characteristics associated with the Raspberry Pi GPIO ports. The maximum output current was specified as 16 mA. This meant that our LED with two 330-ohm parallel resistors (165 ohms) creating an LED current of around 10 mA was fine. But what do you do when a high output current is required?

This activity will review solid state drivers introduced in EET204. We'll review the theory of both BJT transistors and MOSFETs. You'll then wire the large amber LED indicator into the piRover and create blink code. This device will function as a warning when the piRover is in motion.

Prerequisites:

Prior to beginning the instruction provided in this lesson you must have completed the following:

1. Digital Outputs - Introduction

Performance Outcomes:

1. Review theory associated with solid state devices including both BJT transistors and MOSFETS.
2. Measure current requirements of high-intensity amber LED.
3. Install high-intensity amber LED along with MOSFET driver component.
4. Create a warning.py solution that configures both the amber LED and the buzzer and uses PWM outputs to create continuous flashing and beeping.

Resources:

1. Sparkfun transistor tutorial
2. Adafruit Transistors 101
3. Transistor Basics - MOSFETs : 6 Steps - Instructables
4. MOS Driver Module

Materials:

1. piRover with fully charged battery
2. RAM155 Digital Multimeter
3. RAM205 Parts Kit

piRover Builds with K2

Directions:

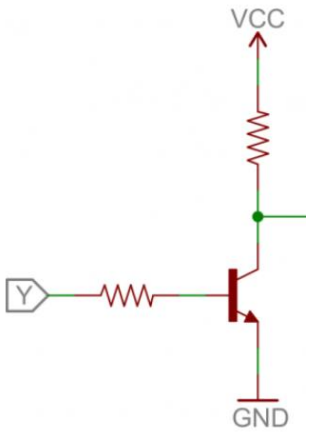
1. The instructor will the electrical characteristics of the GPIO port including high-side and low-side switching.
2. You should be able to calculate the current through an LED given the current limiting resistor value. Assume the voltage drop on an LED is 1.5 V
3. Have you completed EET204? The instructor will review concepts of BJT and MOSFET transistors using the links provided in the Resources section.
4. Why did low-side switching not work with the servo cable that we connected to the servo port? What is the value of VCC? What is the value of Vout? Will a high out on the GPIO prevent the LED from being forward biased and conduct?
5. Review the high-side switching example in the Sparkfun resource. What is wrong with this example?
6. Use your test leads to connect the high intensity LED to your piRover battery. How much current does this device draw from the 12 volt source? Note the polarity of this device. Which side is connected to the positive and which side is connected to the negative?

   I_L = _____
7. Based on the collector current and Beta(hfe) values, which BJT transistor do you recommend – 2N3904 or 2N2222? Why?

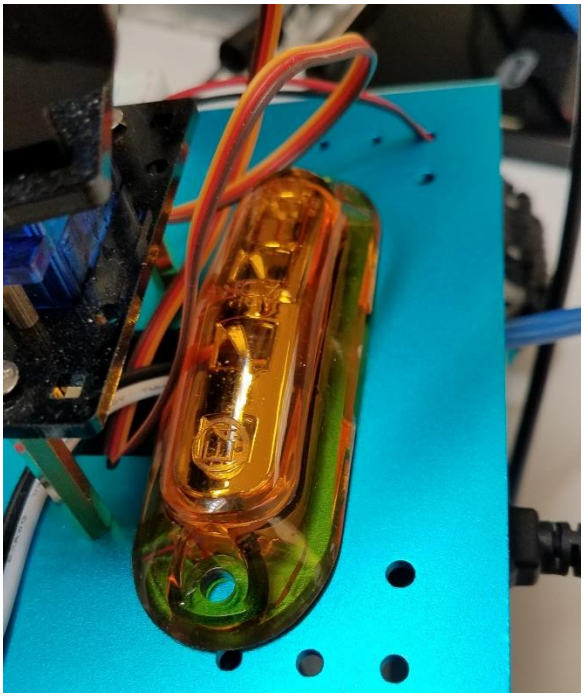
   Q = _____
8. What value of base resistor should be used? Recall that you want to drive the device into saturation to minimize the voltage across the device which minimizes power dissipation.

   I_B = _____

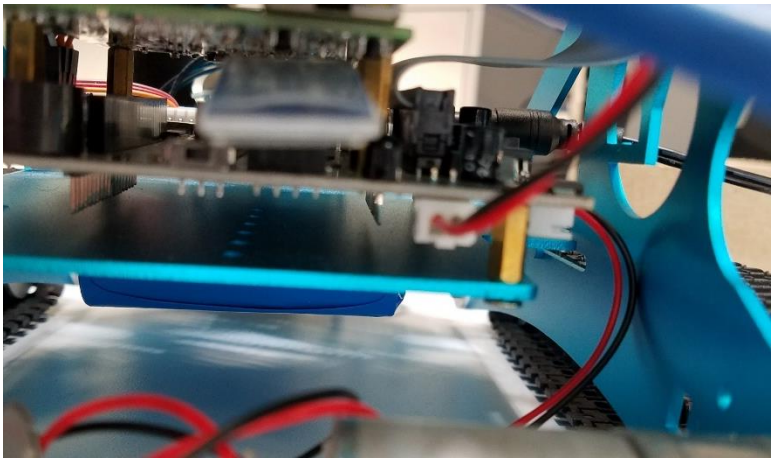
   R_B = _____



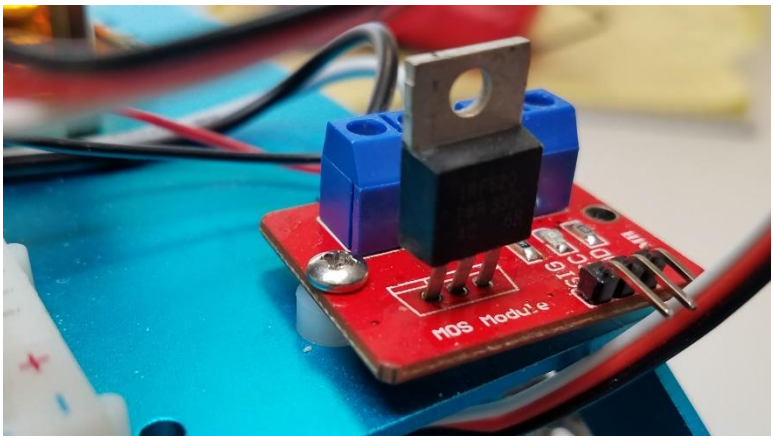
9. The instructor will review the concept of MOSFETs using the link in the resource section. Again, not the option of both high-side and low-side switching.
10. Review the MOS Driver Module provided in your kit. See the link in the Resources section.
11. Locate the two adhesive pads in the RAM parts kit. Use these to attach the high-intensity LED to piRover as shown below.



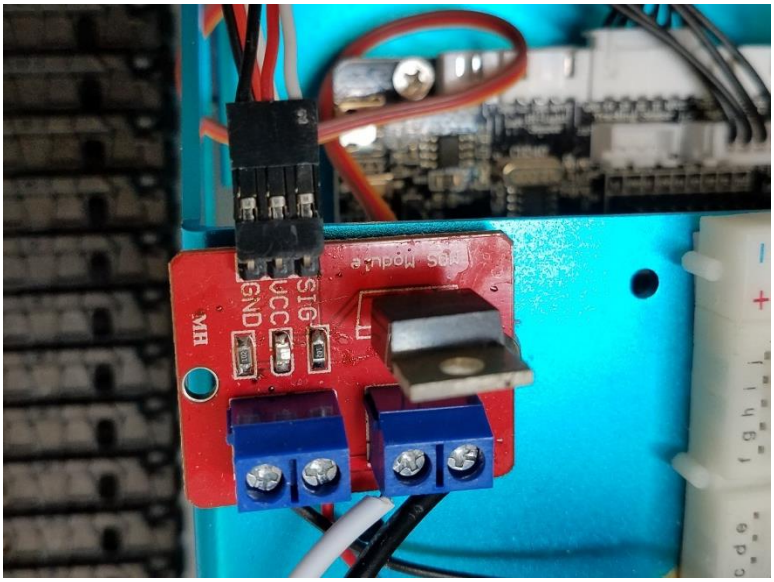
12. Locate the 12-volt power connector on the bottom of the controller board. See the image below. Insert the power lead provided in your parts kit. This will power the MOSFET module and LED.



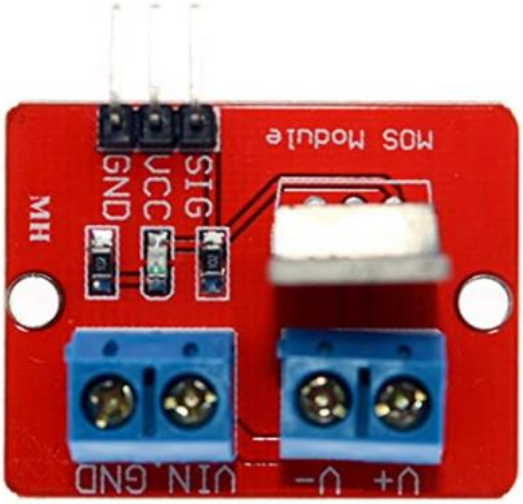
13. Attach the MOSFET module to the top of the piRover using the #4 screw, locknut, and nylon spacer. Only one screw is used due to hole placement.



14. Locate the female-to-female servo cable extension. Replace the male-to-female used on the breadboard during the last activity with this cable. Attach to the MOSFET module as shown.



15. Review the power and load connection for the MOSFET module. Connect the 12-volt power to the module. Connect the LED as the load. An image of the device is provided on the next page.



16. Copy the digital output code used during the last activity to blink the LED on the breadboard. Rename this file to warning.py.
17. Verify your circuit build by running the code. The code was verified during the last activity. So, if you do not have a blinking LED, the issue is with your circuitry.
18. Modify the code by adding the buzzer. Blink the LED and sound the buzzer at a 1-2 Hz rate.
19. Be prepared to demonstrate your circuit during the next class period.

Assessment:

Demonstrate your warning system to the instructor during the next class period. Note that you will need a camera to demonstrate. Also be prepared to share your screen to display your warn code. Additional items will be submitted to Moodle using instructions detailed during next class period.