Introduction

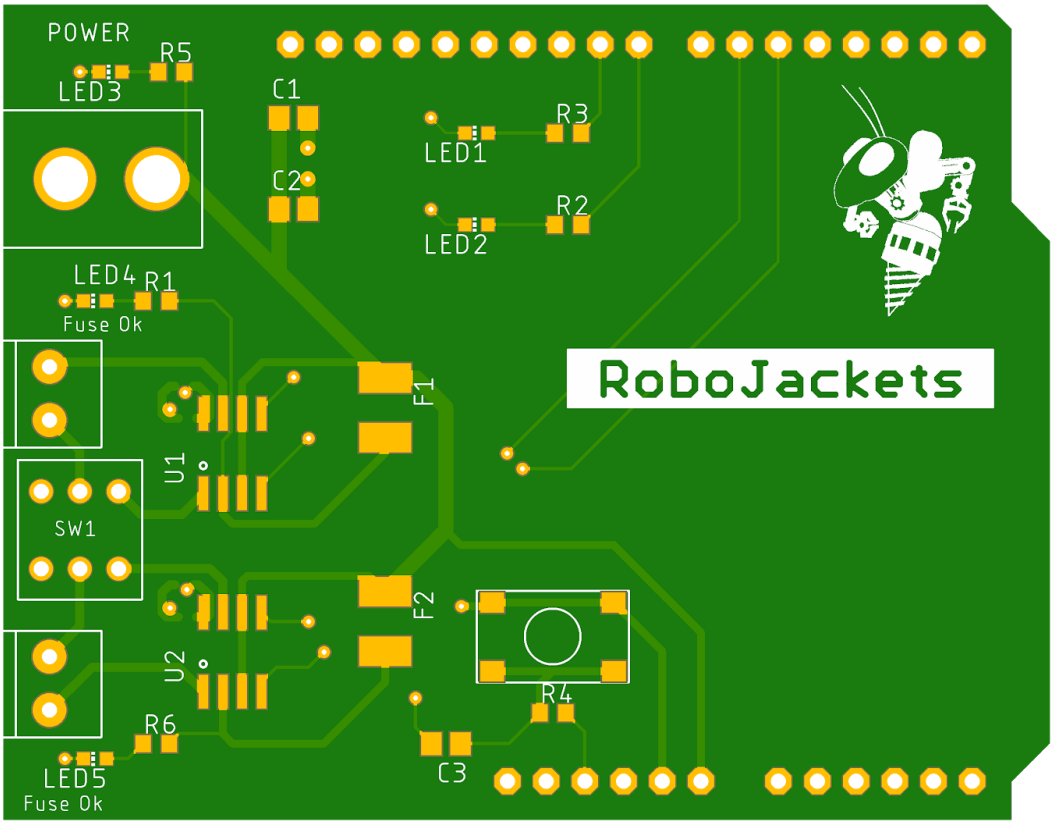
Firmware represents the link between hardware and software; thus it is important to be able to combine knowledge of both when completing firmware tasks. This requires reading schematics, datasheets, and code documentation. This lab will practice those skills by programming functionality onto the Electrical Training boards.

Objective

To do this lab, you will be implementing the software to properly control two DC motors. The hardware component is already done, so you will be using the knowledge from prior labs as well as the datasheet for the motor driver chip to correctly write firmware for the board.

A successful board will have a motor turning at a variety of speeds due to a time-varying voltage induced across one of the output terminals. In addition, it will have the LEDs blinking to indicate the motor speed.

Background



Training Board Functionality

This board contains a dual-motor-driver circuit, meaning it can be used to control two motors at once. This functionality is achieved through a pair of ZXBM5210 motor driver chips, which are each attached to screw terminals that your motor can attach to. A certain signal must be sent to the pins on the motor driver to cause it to apply a voltage across the output terminals.

A schematic for the board is available on the github repository.

Instructions

1. Attach a training board PCB to your Arduino Uno. Screw in a motor into the screw terminals. Attach a jumper to short the pins on the top of the SW1 header.
2. Open up and look at the schematic of the training board (located on the master branch). Pay close attention to the connected pins on the Arduino, and note their function.

3. First, start by creating a program that blinks both LEDs on the training board. This requires writing simple digital signals to each of the pins the LEDs are connected to.
    a. The blink sketch is the first program that gets run on any new board or microcontroller, just to verify that we can successfully write to the microcontroller's flash memory.
4. Take a look at the ZXBM5210 Motor Driver datasheet. Understand its functionality by reading the application note. In our hardware configuration, we connect Vref to VDD, so pay attention to the function of the input signals in this mode. Use this to determine a plan for how to control the motor.
5. Write code to create a 9-volt output between the terminals on the motor driver.
    a. This requires noting that analogWrite takes in a value between 0 and 255 and outputs a PWM signal with duty cycle value/255.
    b. Also, remember that the boards will ultimately take a 12V power supply.
6. Now, output -9 volts between the output terminals.
    a. Understand what signal must be sent to cause the motor driver to output reverse rather than forward.
7. Create code that cycles through the sequence 0V, 9V, -9V, -5V. It should pause at each of these for 1 second. Ask an instructor to verify with a multimeter.

Code Functions

- `digitalWrite(pin, value)`
  - Writes a high or low voltage to a specified pin
  - Arduino Website Documentation
- `analogWrite(pin)`
  - Writes an arbitrary voltage to the pin using PWM
  - Arduino Website Documentation
- `delay(milliseconds)`
  - Stops program execution for a specified number of milliseconds
  - Arduino Website Documentation