Lab 3: Motion

EECS 16B Spring 2024

Slides: links.eecs16b.org/lab3-slides

Administrivia

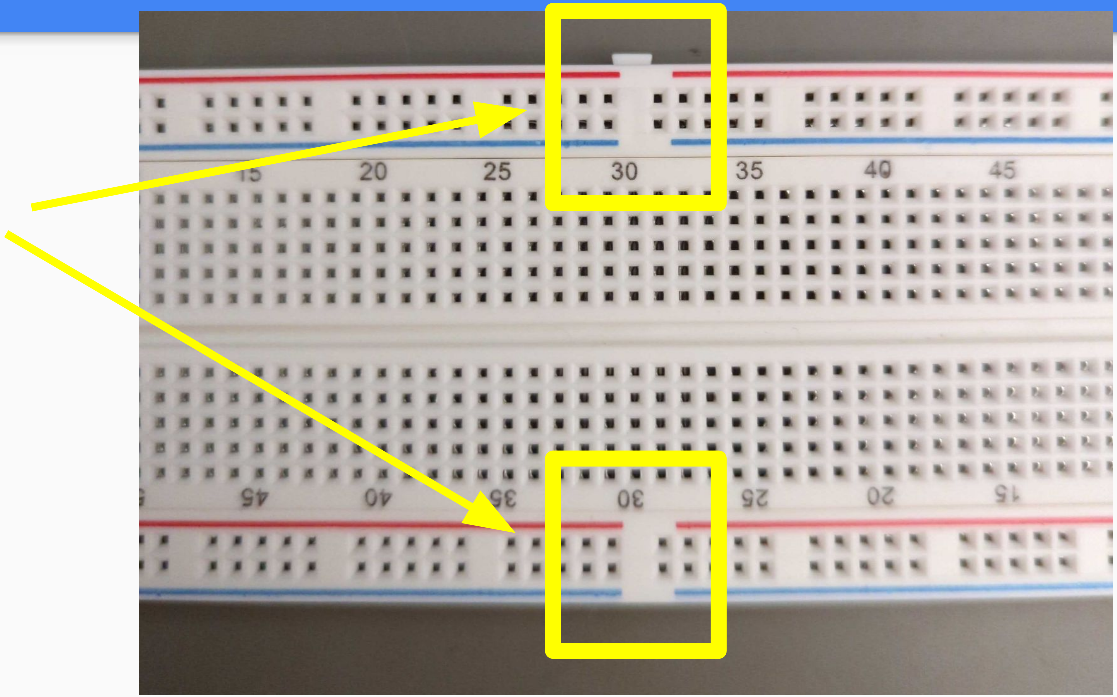
- “Lab checkoffs” assignment on Gradescope
    - Make sure you submitted the email.txt file
        - **Everyone should submit this assignment individually with their own email**
    - If the grade is incorrect/nonexistent ***after the announcement of a fix***, fill out: https://eecs16b.org/lab-checkoff-error

Lab 3 Overview

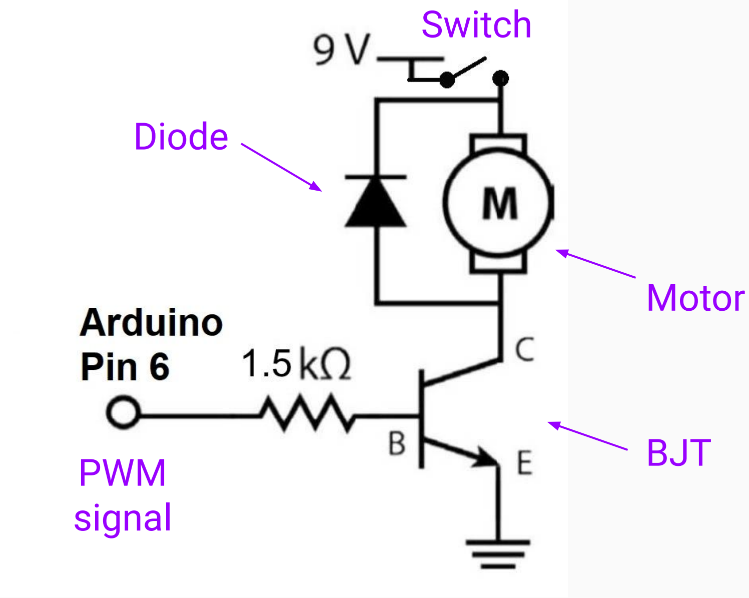
- Build and test motor controller circuits
  - Pulse Width Modulation (PWM) from Arduino
  - Bipolar Junction Transistor (BJT)
  - Switch
  - Diode
- Build 5V Regulator
- Install and test encoders
  - Sensing the distance traveled and speed of the car

Caution!

- Some breadboards may have a break in the power and ground rails
- Make sure to connect them with wires (4 total)!



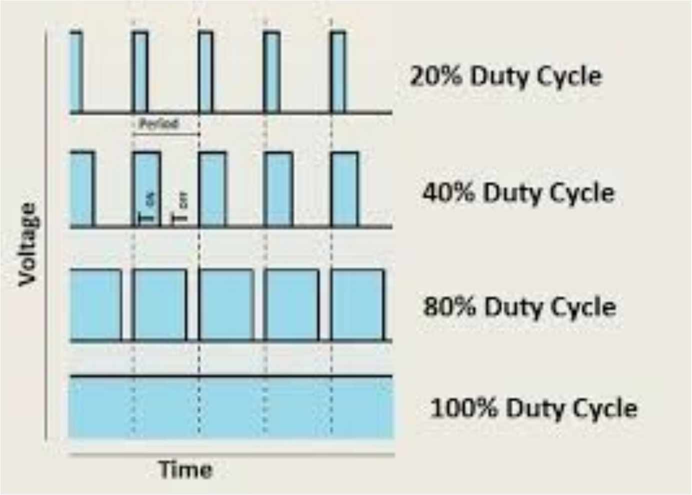
Motor Controller Circuits



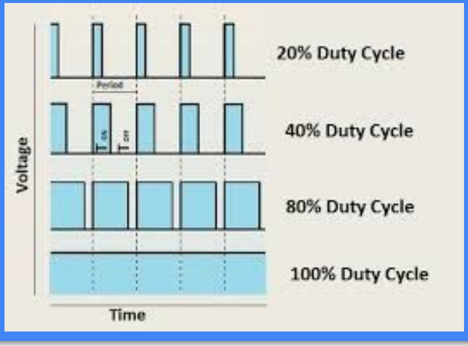
- Built twice (for each motor)
- Many new concepts/components!
  - PWM
  - BJT
  - Switch
  - Diode

Pulse Width Modulation (PWM)

- Square wave with variable “on” time
  - **Duty Cycle:** percentage of time the signal spends “on” or “high” in one period
- Method of supplying variable amounts of power to a component
  - At high frequencies, measurements see an averaged-out analog voltage (depending on duty cycle)
    - i.e. if “on” voltage is 5V and duty cycle is 50%, we will read 2.5V
  - we’ll run over a range of PWM values to see the motors speed up and slow down

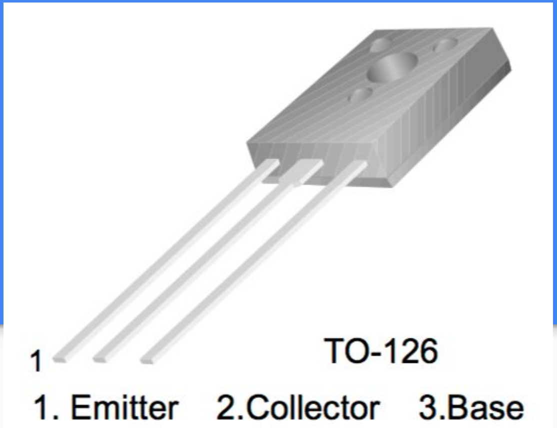


Pulse Width Modulation (PWM)

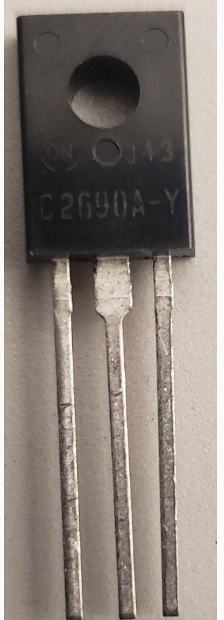


- Square wave with variable “on” time
  - “Duty cycle” is the percentage of time the signal spends “on” or at its “high” in one period
  - If period T = 1s, then 50% duty cycle means it spends .5s “on” and .5s “off”
- Method of supplying variable amounts of power to a component
  - We will be using this to control our motors
- Motors and multimeters unable to “react fast enough” to the rapid turning on and off, so see averaged-out analog voltage depending on duty cycle
  - Variable frequency, anywhere from a few hundreds to thousands of Hz
  - If “on” voltage is 3.3V, then 50% duty cycle means you see 1.65V

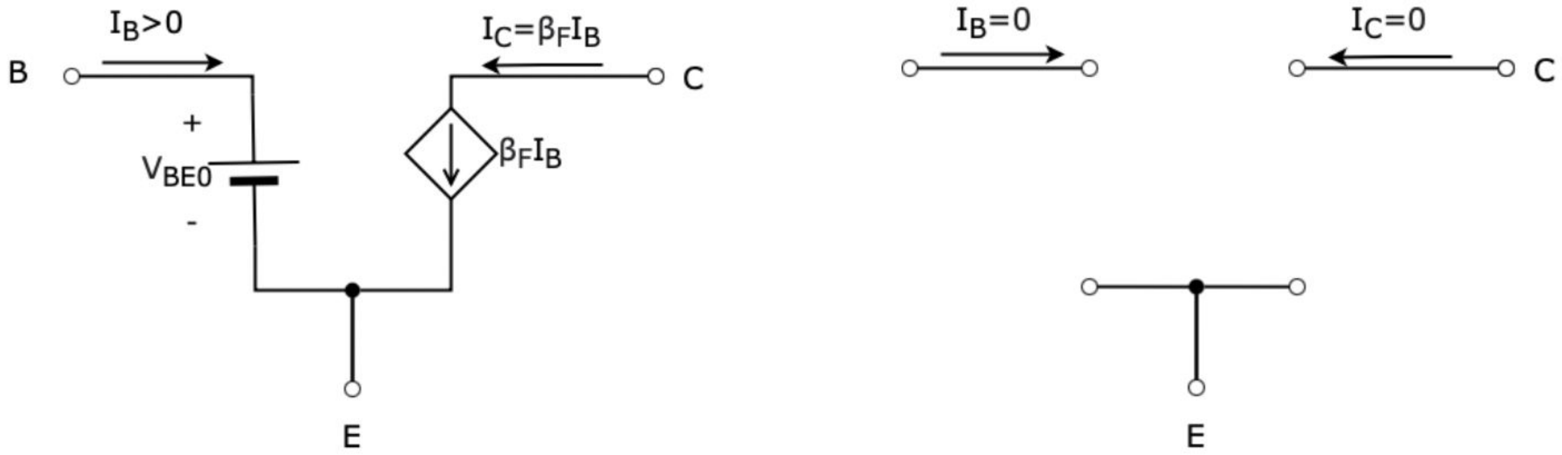
New Component: BJT



- Bipolar Junction Transistor
  - 3 pins: Base (B), Collector (C), Emitter (E)
  - Analogous to MOSFETs: Base -> Gate, Collector -> Drain, Emitter -> Source
- NPN BJT behaves similarly to NMOS
  - High Base voltage turns BJT “on” and conducts current from Collector to Emitter
  - High Gate voltage turns NMOS “on” and conducts current from Drain to Source
  - More accurate model description in lab note
- NOT the voltage regulator component
  - Body is fully black plastic, does not have a metal tab sticking up
  - Orientation of the picture is with the 3 dots on the plastic body facing you



NPN BJT Model



(a) Model of BJT in ON mode (when Arduino output pin is HIGH)

(b) Model of BJT in OFF mode (when Arduino output pin is LOW)

Figure 3: Model of NPN BJT in Different Modes

New Component: Switch

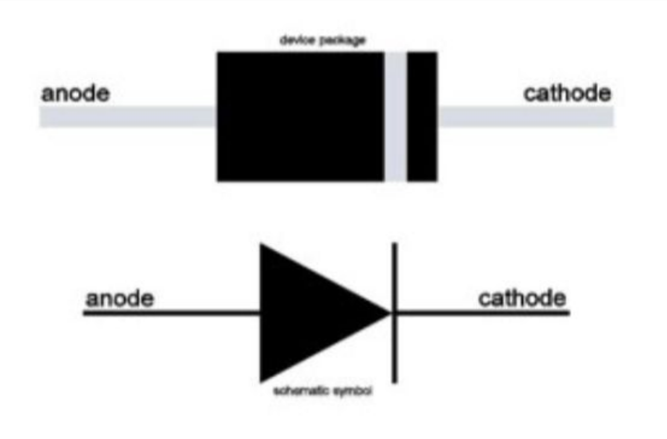
- Allows you to change circuit connections (to change input voltage)
  - If switch is in the left position, left and middle pins are shorted together
  - If switch is in right position, right and middle pins are shorted together
- Note: Middle pin is ALWAYS shorted (connected) to something
  - Connect your motors to middle pin, 9V to side pin, GND to other side pin
  - Toggle your motors being connected to 9V and GND
- **DO NOT CONNECT 9V OR GND TO MIDDLE PIN**
  - Fastest way to say goodbye to your circuit, battery, arduino, and sometimes your usb port

New Component: Switch

- Middle pin is ALWAYS shorted (connected) to something
  - If switch is in the left position, left and middle pins are shorted together
  - If switch is in right position, right and middle pins are shorted together
- Connect your motors to middle pin, 9V to side pin, GND to other side pin
  - Toggle your motors being connected to 9V and GND
- **DO NOT CONNECT 9V OR GND TO MIDDLE PIN**
  - You will short 9V to GND if you flip the switch
  - Fastest way to say goodbye to your circuit, battery, arduino, and sometimes your usb port
  - **DON'T DO IT!**

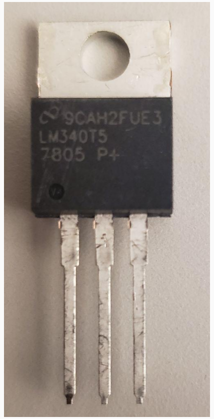
New Component: Diode

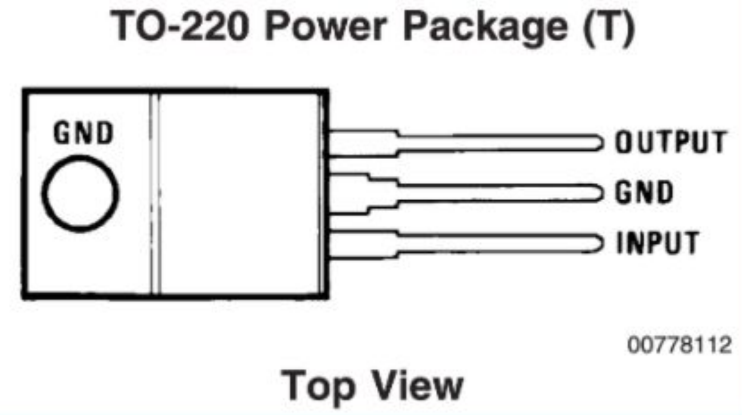
- Conduct current one way but not the other
- **Direction is important**
  - Anode is +
  - Cathode is -
  - Make sure you double check that you have connected it in correct direction
- **not light emitting!**

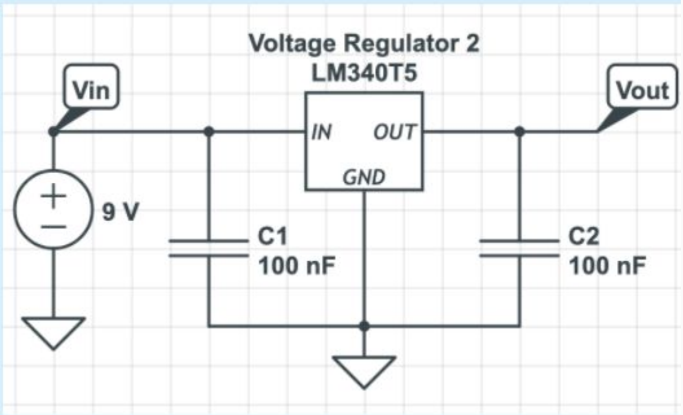


New Component: Regulators

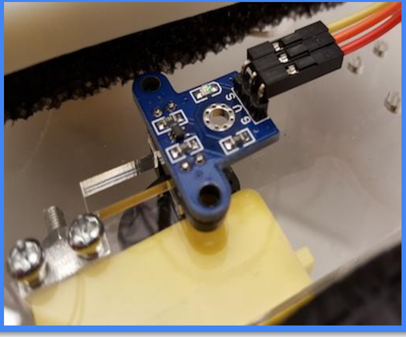
- To later allow S1X33N to be powered by 9V batteries, we will create 9V -> 5V circuits
  - For today, these will be powered using the power supply!
- The 9V-5V Regulator is labeled **LM340T5**
  - Orientation of diagram is with the metal tab facing away from you





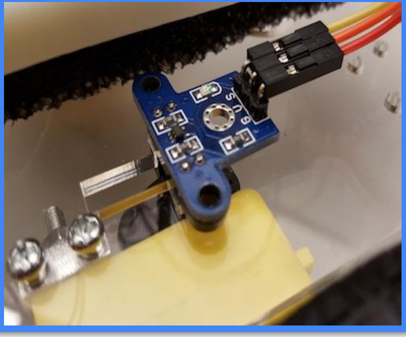


New Component: Encoders (Photointerrupters)



- Beam of light between 2 “legs”
  - outputs voltage based on whether the beam of light is blocked or unblocked
  - Mounted on “encoder wheels,” which have many holes
    - As wheel rotates, spokes block and holes unblock the beam of light
- Can calculate velocity of car from rate of encoder value change
- 3 pins
  - **“G” or “GND”** = ground
  - **“V” or “Vcc”** = voltage (connect to breadboard positive rail, NOT Arduino’s 5V pin)
  - **“S” or “OUT”** = encoder signal (connected to Arduino)

New Component: Encoders (Photointerrupters)



- Beam of light between 2 “legs”
- As wheel turns, rotates encoder wheel with it
    - Encoder wheel has many holes in it
    - As wheel rotates, spokes block and holes unblock the beam of light
- Can calculate velocity of car from rate of encoder value change
- 3 pins
    - **“G”** = ground, connect to breadboard 0V negative rail
    - **“V”** = voltage, connect to breadboard 3.3V positive rail from output of 3.3V regulator, NOT Arduino’s 3.3V pin
    - **“S”** = encoder signal; input to voltage divider whose output goes to Arduino pins (2 and 3)

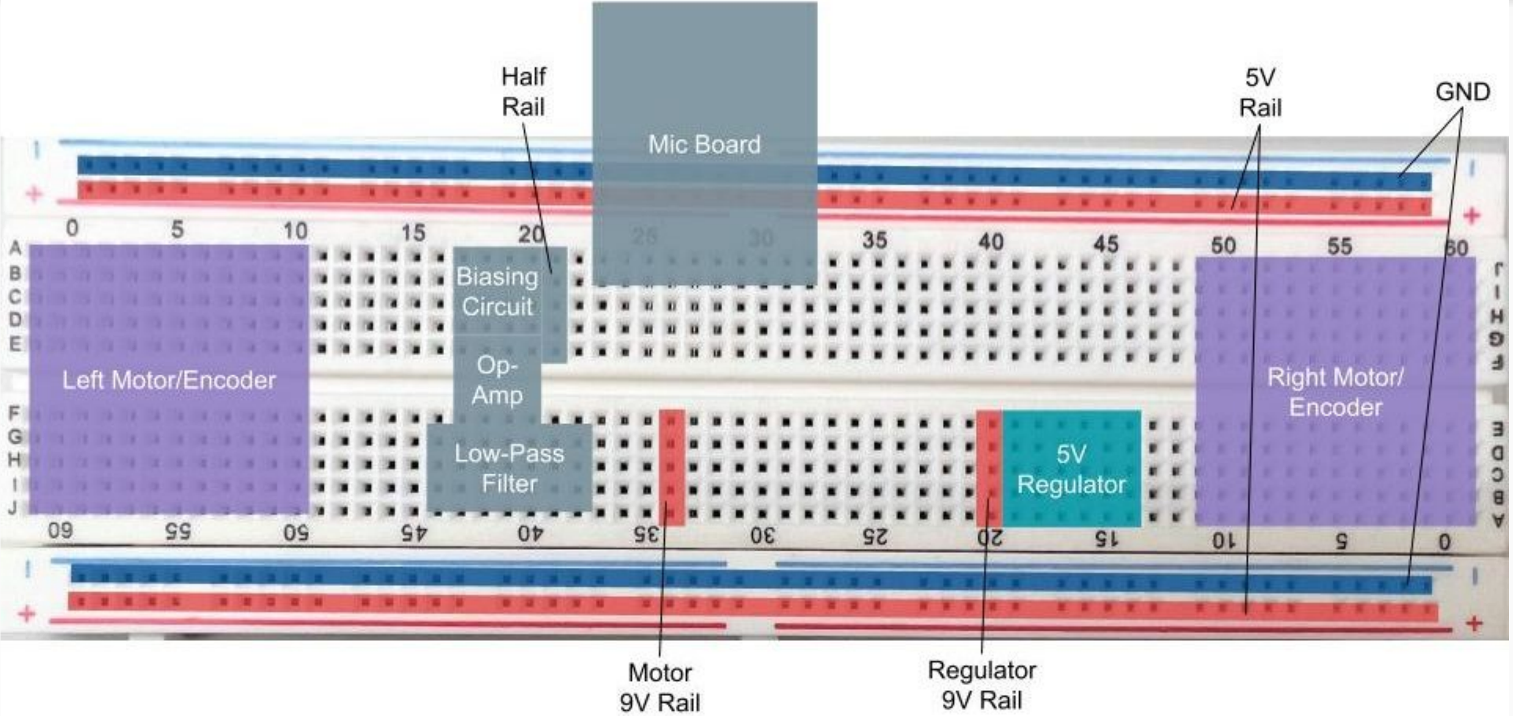
Testing Encoders

- Encoder “S” pins connected to voltage divider
  - Using resistors of ≥1k, divide the 3.3V max voltage down to between 2V - 2.5V
  - Can accomplish with one 1k ohm and one 2k ohm resistor
- Pass something between encoder legs or turn car wheel, red LED on encoder should blink if powered correctly
- encoder_test_0_ticks.ino
  - 4 phases: both wheels unpowered, L wheel powered, R wheel powered, both wheels powered
  - Test once with encoder wheels OFF
    - Checks if encoder readings are from noise or from actual car movement
  - Then test with encoder wheels ON
    - Rules out false negative

General Reminders/Habits

- Make sure to connect one of your Arduino GND pins (any one works) to your breadboard negative (-) rail (which carries GND)
- Encoders and Arduino pins can only tolerate voltages between 0V and 5V; they should never be connected to 9V
- Make sure the regulator metal tabs don’t touch any other components!
  - They are conductive and will short your circuit if they’re too close together.
- 9V from power supply goes to reserved 9V rows on breadboard, NOT positive rails

BREADBOARD LAYOUT



Important Forms/Links

- Help request form: https://eecs16b.org/lab-help
- Checkoff request form: https://eecs16b.org/lab-checkoff
- Slides: links.eecs16b.org/lab3-slides
- Lab Groups: https://eecs16b.org/lab-groups
- Anon Feedback: https://eecs16b.org/lab-anon-feedback
- Lab Grade Discrepancy: https://eecs16b.org/lab-checkoff-error