Controlling a Brushless Motor

Calendar for ”Making with Technology”

- Week 1: Making and measuring electromagnets
- Week 2: Designing a brushless motor
- Week 3: Fabricating the parts
- Week 4: Programming the Teensy ← Today!
- Week 5: Motor demonstrations

December 3, 2018
Parts: Layout and Fabrication

Try to finish fabrication of parts today:

- base plate
- electromagnets
- Hall-effect sensors
- rotor

Fabrication tips:

- Laser cut acrylic for planar parts.
- 3D print non-planar parts.
- Power supplies provide 5V to drive electromagnets.
  - Limit power to 2 watts per electromagnet to avoid overheating.
  - Use 32 gauge wire for simple designs with plastic bobbins.
- Use screws to hold pieces together – NO glue or tape.
Teensy 3.2 Microcontroller

The Teensy3.2 is a powerful and inexpensive controller. Compatible with Arduino, supported on Linux, Windows, and Macs. Programmable in C/C++. Controllable in any language.

Four dedicated pins:
- two power pins: power in (5V) and power out (3V, 0.1A max)
- two grounds: analog and digital.

24 programmable pins:
- any of these can be used for digital in/out
- 10 pins (A0-A9) can be used for analog in/out
Switches for Electromagnets

Adafruit TB6612 board: 2 H-bridges to control electromagnets.

Each H-bridge can supply up to 1.2A at 5V to one or multiple coils
• connected in parallel (so that currents add) or
• connected in series (so that voltage is divided among them).
External Power Supply

Power for motors is provided from an external microUSB connector. To use the same source for Teensy (so that motor can run without a laptop) we must disconnect power from the Teeny’s USB port.

Then power comes from external microUSB connector, and USB connector on the Teensy is only used to control the Teensy.
**Wiring**

Solder wire-wrap connectors to Teensy, TB6612, and USB boards.

Use wire-wrap to connect power and ground leads as follows.

![Wiring Diagram](image-url)
Test Power Wiring

Test power connectors to avoid damage to your laptop or circuits.

**Step 1:** Plug USB power supply into Teensy USB connector and then into wall power. Since we cut the USB power supply, the integrated led shown below *should NOT blink*. If it does, recut the power trace on the backside of the Teensy and repeat this test.

**Step 2:** Remove USB power supply from Teensy USB connector and plug it into the microUSB board. Now the integrated led *should blink*. If it does not, then power is not getting to the Teensy from the microUSB board. Check your power connectors and repeat test. Do not proceed until you get no blink in step 1 and blink in step 2.
Programming the Teensy 3.2

The Teensy 3.2 is compatible with Arduino.

Use "Teensyduino" software. Installation is described at https://www.pjrc.com/teensy/td_download.html.

Program examples are included in today's lab.
Our Teensy programs will typically have four parts.

// declarations

void setup(){
    // initialization code
}

void loop(){
    // code to do repeatedly
}

void serialEvent(){
    // code to process messages to and from laptop
}

We will describe each of these parts in the examples that follow.
Connecting Hall-effect sensors

Connect Hall-effect sensors.

left pin (red): +3.3V
center pin (black): gnd
right pin (blue): one of Teensy’s analog pins (A0 to A9)

Multiple sensors can share +3.3V and gnd, but each must have a separate analog pin.
Hall Sensor Code

Reading sensor outputs with Teensy.

// simple program to read output from one Hall device

int HallA = A8; // analog channel for Hall A device
int HallAv; // 0 <= answer < 16384

void setup(){
    analogReadResolution(14); // set A/D converter to 14 bits
}

void loop(){
    HallAv = analogRead(HallA)
}

Connecting Electromagnets

Each TB6612 board has two H-bridges.

Multiple electromagnets can share the same H-bridge if they are always switched on and off at the same time.

- If connected in series, the voltage across each coil will be halved.
- If connected in parallel, the total current required will double.

Voltage from H-bridge $= 5V$.
Maximum current from each H-bridge is 1.2A.
Controlling an H-Bridge

Switches in each H-bridge are controlled by input pins: IN1 and IN2.

OFF
IN1: L; IN2: L

BRAKE
IN1: H; IN2: H

CW
IN1: H; IN2: L

CCW
IN1: L; IN2: H

H = high = +3.3V; L = low = gnd.
Unused inputs: PWM and STBY should be tied H.
Wiring an H-Bridge

Any of the Teensy pins 0 to 23 can be used as a digital control line.

![Diagram of Teensy and TB6612](image)

Connect coils to TB6612 using wire-wrapped connections.

Use very fine sandpaper (600 grit) to remove the insulation from the last 2” of magnet wire. To prevent accidental short circuits, wrap 3” of magnet wire (1” with insulation and 2” without) so that ALL of the bare magnet wire is wrapped.
Coil Control Code

Controlling electromagnets with Teensy.

// simple program to set coil A to north polarity

int coilA1 = 13; // coil A, input pin 1
int coilA2 = 14; // coil A, input pin 2

void setup(){ // setup control pins as Teensy outputs
  pinMode(coilA1,OUTPUT);
  pinMode(coilA2,OUTPUT);
}

void loop() {
  digitalWrite(coilA2,LOW); // turn second control pin off
  digitalWrite(coilA1,HIGH); // then turn first control pin on
}
Sample Code

Examples of code are contained in today’s lab.

Tailor these examples to match the number of sensors and coils that you are using – as well as your pin assignments.