Electronics inside an electric fan motor.

Electronics inside a Tesla Model 3 Motor.
Current Project

Our first few projects introduce the use of a microprocessor to control simple electronic devices.

Last Week: Control LED with pushbutton.

This Week: Control LED with a microcontroller.

Next Week: Implement “Simon” game.
Microcontroller Hardware

We will use a Teensy 3.2 microcontroller.
We will use a Teensy 3.2 microcontroller.
Teensy 3.2

The Teensy 3.2 has four dedicated pins:

- power input $V_{\text{in}}$ to be connected to a 5V power supply,
- power output (3.3V) which we will use to power our circuits,
- analog ground (AGND) for analog circuits, and
- digital ground (GND) for digital logic.

All of the other 24 pins are programmable.
Any of these 24 can be used for digital input or output.
Any of 10 pins labeled A0–A9 can be used for analog input or output.
Driver Circuits

There are also two “driver” circuits to control LEDs and coils.
Driver Circuits

There are also two “driver” circuits to control LEDs and coils.
Driver Circuits

Each driver circuit controls four output pins.
Driver Circuits

The output pins are organized in pairs – e.g., coilA+ and coilA–.

We refer to these as coil pins because they will ultimately be used to excite the coils in a brushless motor.
Driver Circuits

These outputs pins are controlled by input pins AIN1 and AIN2.

The driver’s input pins are connected to Teensy output pins, which are ultimately controlled by a Teensy program.
**Driver Circuits**

We control Coil A outputs with Teensy pins 0 and 1:

<table>
<thead>
<tr>
<th>Teensy pin</th>
<th>Coil A</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 HIGH</td>
<td>+ gnd</td>
</tr>
<tr>
<td>1 HIGH</td>
<td>– gnd</td>
</tr>
<tr>
<td>0 HIGH</td>
<td>+ +5 V</td>
</tr>
<tr>
<td>1 LOW</td>
<td>– gnd</td>
</tr>
<tr>
<td>0 LOW</td>
<td>+ gnd</td>
</tr>
<tr>
<td>1 HIGH</td>
<td>+ +5 V</td>
</tr>
<tr>
<td>0 LOW</td>
<td>float*</td>
</tr>
<tr>
<td>1 LOW</td>
<td>float*</td>
</tr>
</tbody>
</table>

* “float” means that both +5 V and gnd are disconnected

The other output pins are controlled similarly:

<table>
<thead>
<tr>
<th>Teensys pins</th>
<th>Output pins</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 1</td>
<td>Coil A+</td>
</tr>
<tr>
<td>2 3</td>
<td>Coil B+</td>
</tr>
<tr>
<td>4 5</td>
<td>Coil C+</td>
</tr>
<tr>
<td>6 7</td>
<td>Coil D+</td>
</tr>
</tbody>
</table>
Sensing External Inputs

The microcontroller card has provisions for four external inputs.

Voltages at these input screws are converted to integers (0-255) that are available to the microcontroller.
Microcontroller Software

Teensy programs are written in C/C++ organized in four parts.

// declarations

void setup(){
   // initialization code
}

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

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

Use the declarations area to set up symbolic names for constants such as pin numbers.

// Output Control Pins

#define AIN1 0
#define AIN2 1
#define BIN1 2
#define BIN2 3
#define CIN1 4
#define CIN2 5
#define DIN1 6
#define DIN2 7

// Input Access Pins

#define SENSOR1 A0
#define SENSOR2 A1
#define SENSOR3 A2
#define SENSOR4 A3
The initialization area contains a single function called setup. The setup function runs just once: right after power is turned on.

```c
void setup(){
    // Teensy pins are INPUTs by default, which is fine for SENSOR pins. 
    // However, output control pins must be set as digital output signals.
    pinMode(AIN1,OUTPUT);
    pinMode(AIN2,OUTPUT);
    pinMode(AIN3,OUTPUT);
    pinMode(AIN4,OUTPUT);
    pinMode(AIN5,OUTPUT);
    pinMode(AIN6,OUTPUT);
    pinMode(AIN7,OUTPUT);
    pinMode(AIN8,OUTPUT);
    // Teensy will convert SENSOR pins to integers.
    // Use 8-bit resolution to get integers between 0 and 255.
    analogReadResolution(8);
}
```
The Loop

Use loop area contains a single function called `loop`. The loop function is called repeatedly, as long as power remains on.

```c
void loop(){
    // example code to set COIL A voltages based on voltage at SENSOR1 input.
    // if SENSOR1 voltage is less than half of 3.3V ...
    if(analogRead(SENSOR1)<128){
        // set COIL A+ to HIGH and COIL A- to LOW
        digitalWrite(AIN1,HIGH);
        digitalWrite(AIN2,LOW);
    } else {
        // set COIL A+ to LOW and COIL A- to HIGH
        digitalWrite(AIN1,LOW);
        digitalWrite(AIN2,HIGH);
    }
}
```
Events

Use events area for code that should run when external events such as messages from the laptop occur.

There are special keywords for each kind of external event that can happen.

For example, if you include a function called `serialEvent`, that function will run every time the microcontroller receives a character from the laptop.
Breakout Groups and Homework

We will divide up now to work in small groups to work on this week’s projects, which are described under the week 3 lab tab:

http://mit.edu/6.a01

Homework: Upload a video to demonstrate this week’s project.

Next week, we will build on today’s results to construct the “Simon” game.