Nick Marchuk Northwestern Medical Makers 2019 Workshop 2 – Introduction to Arduino

The Arduino we are using is the Feather M4 Express from Adafruit: https://www.adafruit.com/product/3857. This is a powerful 32bit microcontroller running at 120MHz with 512k flash (program space) and 192k RAM (variable space). It runs on 3.3V and has a built in LiPo battery charger. I'll refer to it as the Feather.

The Feather is roughly 4 times faster and has 16 times as much memory as the classic Arduino Uno, and costs about the same. Note that the Feather runs at 3.3V, not 5V like the Uno. If you are looking at instructions for Uno circuits, be sure to replace any 5V sources with 3.3V or you may damage the Feather.

The Feather can be programmed using C++ using the Arduino IDE, or using Python using the Mu editor. Python doesn't run very fast compared to C++, so we'll stick with C++.

First, you need the Arduino IDE from https://www.arduino.cc/en/Main/Software

Next, follow the instructions at https://learn.adafruit.com/adafruit-feather-m4-express-atsamd51/setup to add Arduino SAMD and Adafruit SAMD support to the IDE.

Plug the Feather into your computer. Open the example called Blink (File->Examples->Basics->Blink). Select the Adafruit Feather M4 Express from Tools->Boards. Select the correct Port from Tools->Port (usually the largest number or the last one in the list, it is also the port that disappears if you unplug the board). Click the Upload button (rightwards facing arrow), and the code will compile and upload to the board. If it works, the red LED near the USB connector will blink 1 second on and one second off. Change the code so that the LED blinks 1/10th of a second on and 1/10th of a second off, and verify that the LED blinks 10 times faster than it did before.

Next, try the example called AnalogReadSerial (File->Examples->Basics->AnalogReadSerial). This code will create a new port and print the value of the voltage applied to pin A0 at 1000Hz. Upload the code and view the printed values in the Serial Monitor and Serial Plotter from the Tools menu (nothing is plugged into A0 so the voltage will just be noise). Change the delay to be 100 times slower and upload the code again.

Note that you need to change the port each time you leave the monitor and upload the code. You may also need to put the Feather into "bootloader mode" to upload the code by double clicking the Reset button on the board.

Snap 4 header pins off of one of the strips of pins, and press the long strips into your breadboard so that the Feather is at the top of the board and the USB connector faces out. Solder the Feather into the header pins. Here are some soldering tips: http://hades.mech.northwestern.edu/index.php/EDI_Bootcamp

Connect the 3V pin (it's actually 3.3V) to the red power rail of your breadboard, and the GND pin to the blue

power rail. Take the output of your pulse detector circuit and plug it into the A0 pin. Verify that you can see your pulse in the Serial Plotter.

Arduino style C++ code generally has three regions:

- the top of the code has the definitions of global variables
- the setup() function calls initialization functions
- the loop() function contains code that runs after setup() and until the board loses power

Common code tips:

- The delay() function stops the code in milliseconds
- Serial.println() can print the value of a variable or text if it is in quotes, like “hello!”
    - To print a statement like “the number is 14”, use several print() statements, with the last statement being a println(), like Serial.print(“the number is”); Serial.println(variable_name);
- The analogRead() function returns an integer from 0 to 1023, where 0 is 0V and 1023 is 3.3V
- The pinMode() function is used to make a pin an INPUT (to read digital voltages like buttons) or OUTPUT (to set voltages to turn things like LEDs on and off). This function is only needed once, in setup()
- The digitalWrite() function controls a pin that has been setup as an OUTPUT
- The millis() function returns how many milliseconds have passes since the Feather turned on

---

Exercise:

- Edit the AnalogReadSerial example so that it prints the voltage (like 1.65) instead of the raw analogRead() value, and print at 100Hz
- Continue to edit the code so that it also will turn on the red LED on pin 13 if the voltage goes above (or below) some threshold, so that the LED blinks once per heartbeat
- Continue to edit the code so that the time between the heartbeats is printed, every time a heatbeat is detected