Robert Whitney - rw429
ECE 4960 - Lab 1 Writeup

The first part of lab 1 was simple because I already have the Arduino IDE installed on my computer and am familiar with how these devices work. I uploaded the “Blink” sketch from the examples folder to ensure that sketches could be uploaded to the Artemis. From previous experience, I know that it is successful when the RX and TX lights on the Artemis blink rapidly while the sketch is uploading. To double check that the code was working properly, I changed the timing of the blinking, uploaded the new sketch, and observed the light blinking faster as expected.

```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

  This example code is in the public domain.
 */

// Pin 13 has an LED connected on most Arduino boards.
// Pin 11 has the LED on Teensy 2.0
// Pin 6  has the LED on Teensy++ 2.0
// Pin 13 has the LED on Teensy 3.0
// give it a name:
int led = 19;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);               // wait for a second
  digitalWrite(led, LOW);    // turn the LED off by making the voltage LOW
  delay(500);               // wait for a second
}
```

Fig 1: The blink example modified to blink off quickly (0.5s off time instead of 1s)

The “Serial” example worked as expected, parsing inputs to the Artemis and using those inputs to display messages on the Serial Monitor. I had to change the baud rate to view them properly.

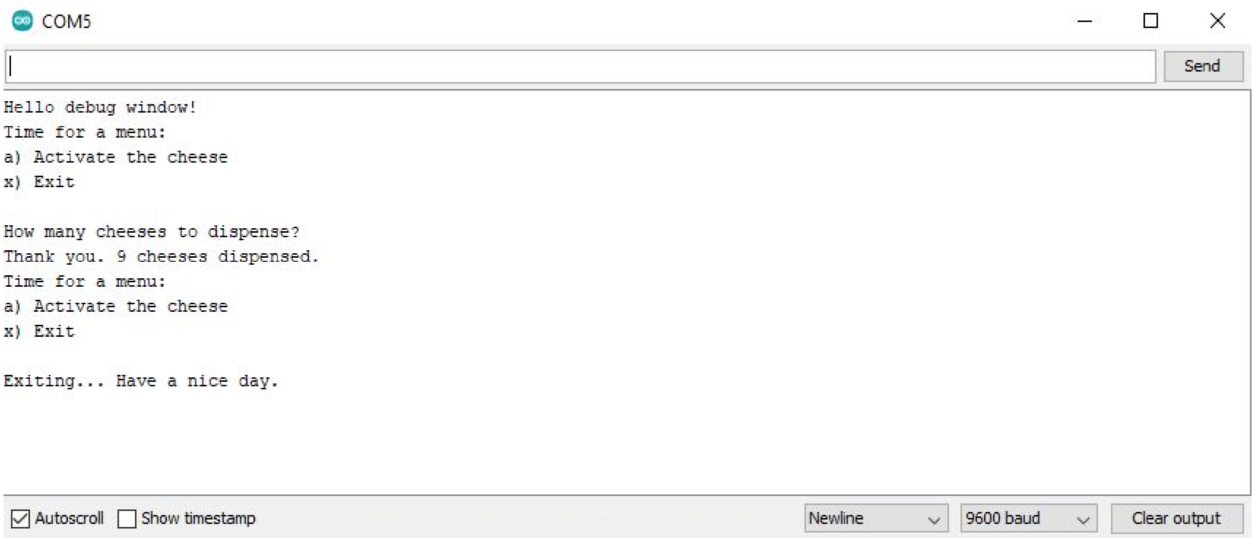

Fig 2: Serial output after entering “a9” then “x”

The “analogRead” example worked as well, although the functionality of the sketch was not fully realized. The temperature sensor reported gradually higher temperatures as I held my thumb to the board. However, the pin being read as an analog input was floating as it was not grounded or connected to an input, as expected.

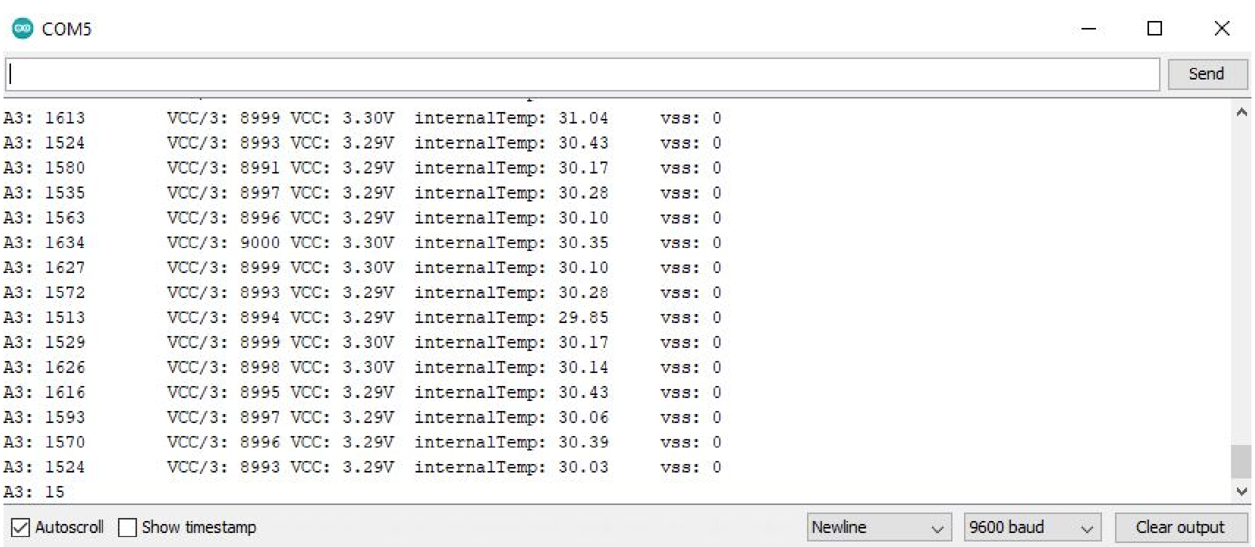

Fig 3: Analog output before holding thumb on Artemis chip. Internal temp hovers around 30.

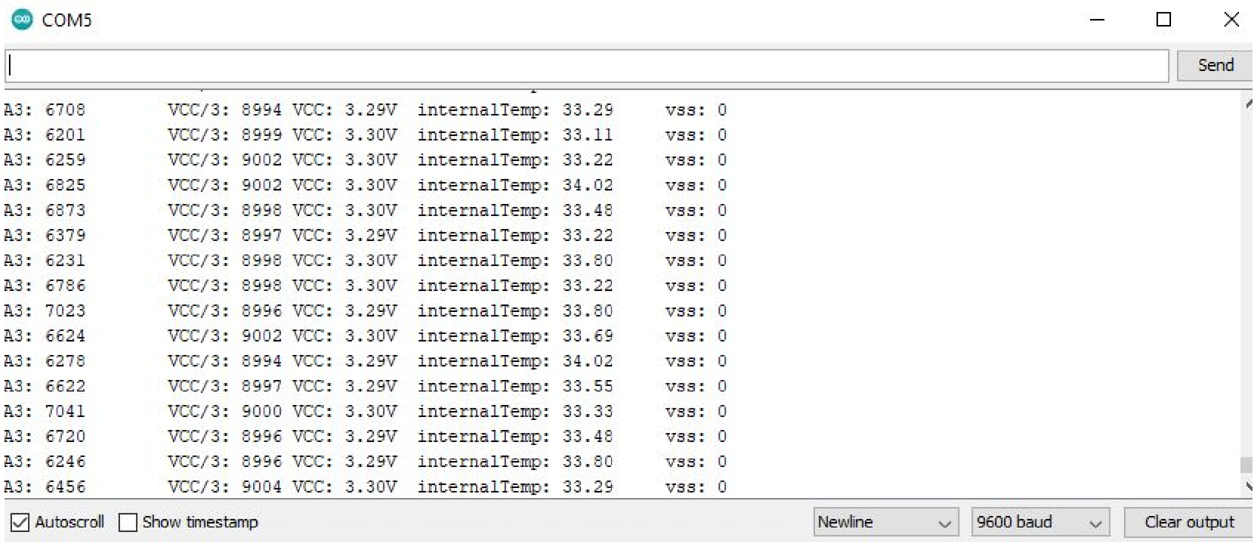

Fig 4: Analog output after holding thumb on Artemis chip for 20s. Internal temp hovers around 33.

The "microphoneOutput" example sketch also worked as described, returning the frequency of the loudest sound. When whistling loudly, I could change the output on the serial monitor by changing the pitch of my whistling in a range from 100 to 2500.

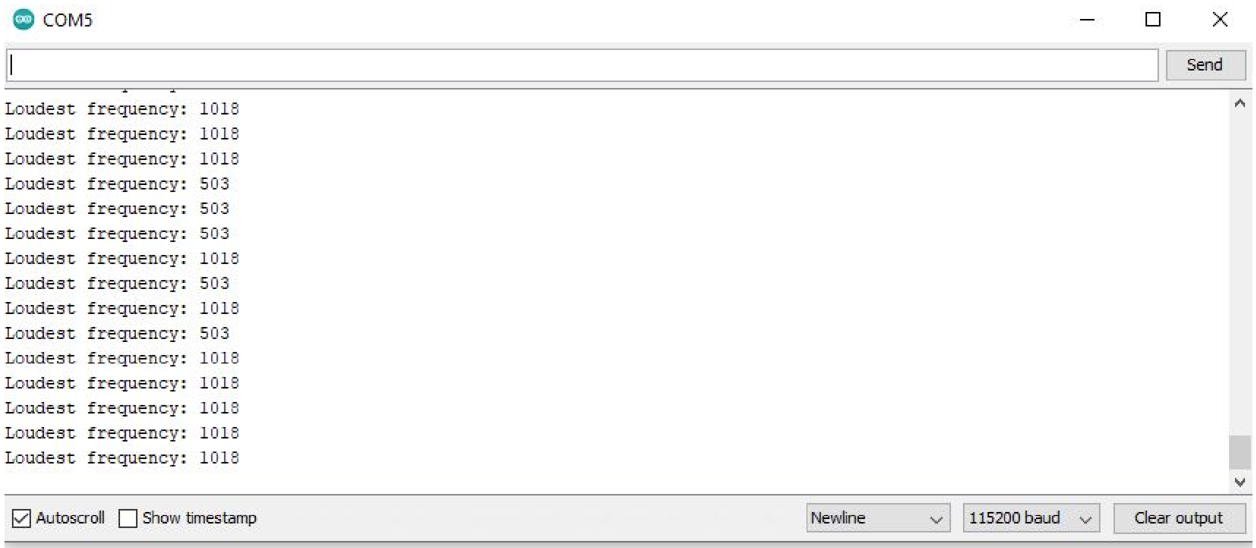

Fig 5: Serial output in the absence of whistling.

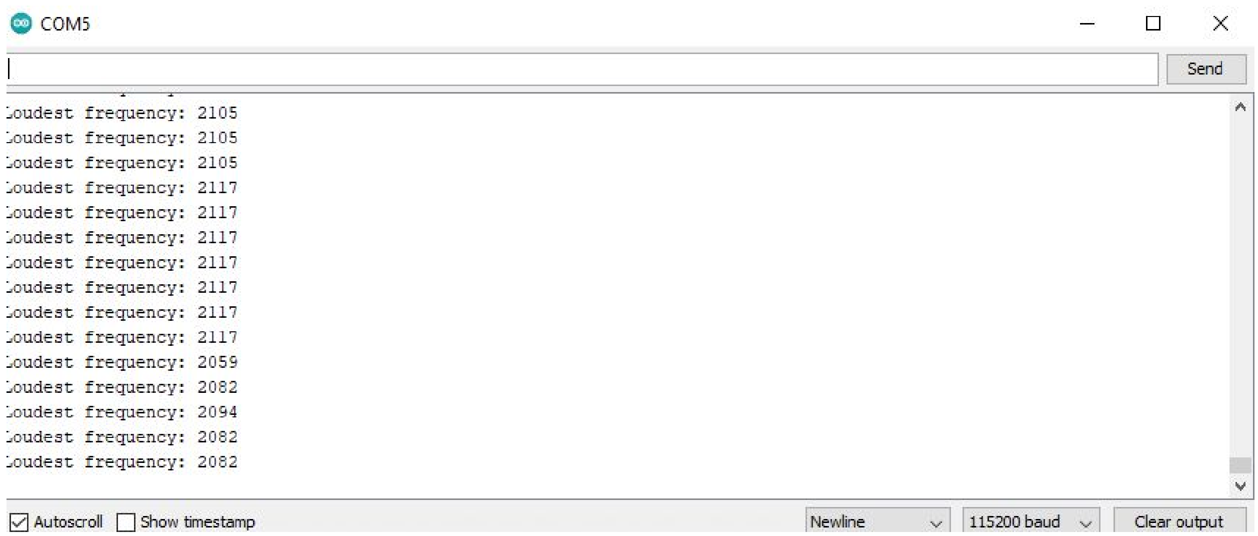

Fig 6: Serial output in the presence of high pitched whistling

To test the battery, I inserted an "if" statement to turn on the onboard LED (pin 19) if the loudest frequency was above 2000 and turn it off otherwise. I also had to set pin 19 as an output in the setup function. This modification to the sketch worked as expected. I could whistle a low tune while watching the serial monitor, slowly raise the output above 2000, and watch the LED turn on.

```
ui32LoudestFrequency = (sampleFreq * ui32MaxIndex) / pdmDataBufferSize;
if (ui32LoudestFrequency > 2000) {
  digitalWrite(19, HIGH);
}
else {
  digitalWrite(19, LOW);
}
```

Fig 7: modification made to "Mic Example" to turn on the LED in the presence of high frequency whistling.

I also unplugged the Artemis from the USB cable and plugged in the battery. The power LED turned on, indicating the sketch was probably running. I tested the sketch with whistling again, this time with no serial monitor to check the frequency. Again, the LED turned on when I whistled a high enough frequency and turned off when it was below the threshold.