What to Expect

A typical workflow consists of the following steps:

1. Load ***RealTime_Detector.ino*** script into the IDE
2. Adjust script parameters (specified later in this document) based on experiment requirements
3. Ensure that the Analog pin **A1** is fed the pre-processed signal (raw LFP input filtered by protoshield)
4. Ensure the external hardware driving the experiment (ex: laser) is configured to accept input from **digital pin 51.**
5. Set the stopping condition parameter to determine the length of the experiment
6. Retrieve the SD card post experiment and use ***byte_data_reader.py*** to convert relevant byte data into a readable format.

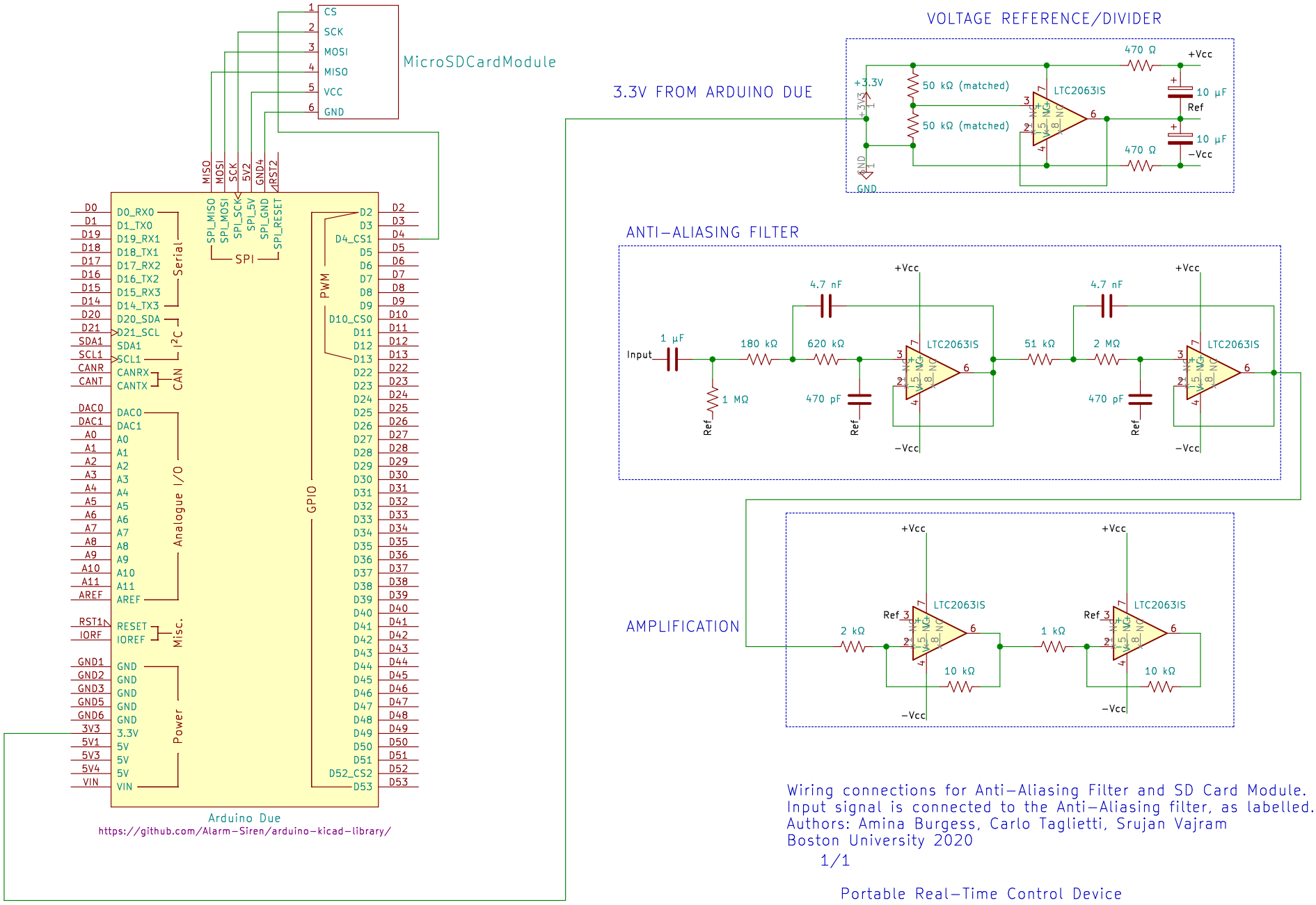
Hardware Setup

Component List

- Arduino, Due
- SanDisk Ultra 16GB MicroSD card
- SD card adapter (Model number and build)
- Protoshield (see circuits diagrams)

Arduino Wiring Diagram

- Technical Schematic for the Anti-Aliasing Filter Circuit, and the SD card connections on the next page.


MicroSDCardModule
CS
SCK
MOSI
MISO
VCC
GND
MISO
MOSI
SCK
5V2
GND4
RST2
SPI_MISO
SPI_MOSI
SPI_SCK
SPI_5V
SPI_GND
SPI_RESET
SPI
Serial
I²C
CAN
Analogue I/O
Misc.
Power
PWM
GPIO
Arduino Due
https://github.com/Alarm-Siren/arduino-kicad-library/
3.3V FROM ARDUINO DUE
VOLTAGE REFERENCE/DIVIDER
+3.3V
GND
50 kΩ (matched)
50 kΩ (matched)
LTC2063IS
470 Ω
470 Ω
+Vcc
−Vcc
10 µF
10 µF
Ref
ANTI−ALIASING FILTER
Input
1 µF
180 kΩ
620 kΩ
4.7 nF
1 MΩ
470 pF
51 kΩ
2 MΩ
4.7 nF
470 pF
AMPLIFICATION
2 kΩ
10 kΩ
1 kΩ
10 kΩ
Wiring connections for Anti−Aliasing Filter and SD Card Module.
Input signal is connected to the Anti−Aliasing filter, as labelled.
Authors: Amina Burgess, Carlo Taglietti, Srujan Vajram
Boston University 2020
1/1
Portable Real−Time Control Device
Fall 2019 − Spring 2020

Program Setup and Initialization

The program is run as the script ***RealTime_Detector.ino***. We use two additional scripts to initialize our SD card and ensure we can read byte data off the SD card: ***SdFormatter.ino*** and ***byte_data_reader.py*** respectively. An additional script ***Constant_Stimulation.ino*** can be used for a seperate constant stimulation mode.

RealTime_Detector.ino: This program performs the real-time detection and subsequent stimulation of either peaks or troughs. The important parameters to set include

a) ***const boolean peakDetection***: Set this value to **true** to ensure peak detection, and **false** to ensure trough detection.
b) ***const int refractoryLength***: Value in microseconds, which determines how long the stimulation will last once the feature has been detected
c) ***const int sDsize*** = *150:* Value that affects the number of data points sent out at a time to the SD card. Changing this affects the data-transmission performance. (Bigger buffer sizes mean more data , but potentially longer times to send.)
d) const int STOP_ITER: Variable that determines how many times the loop will execute (i.e the length of the experiment.)
e) ***const int outputPin*** = 51: Value that sets the digital output pin (stimulation pin)

The implementation uses the **SdFat** library written by Bill Greiman, which can be searched and installed using the ‘manage libraries’ tab in the Arduino I.D.E.

Storing and interpreting data

Troubleshooting

Occasionally the script may fail due to improper SD Card format. In such situations, the following steps should be taken:

1. Retrieve any important data already on the SD card and save it in a separate location.
2. Go to ***File => Examples => SDFat => SDFormatter***
3. Set ***const uint8_t chipSelect*** to **4**
4. Run the script and open the serial monitor. The card should initialize successfully and provide a few options.

5. If the card does not initialize, there may be a connection issue between the card and the board. Recheck all the wire connections.
6. Choose the **Option F “erase and then format the card”**
7. This should set the card to the correct format.

Results from Simulation Data and Other Metrics:

Bode Plot for External Filter:

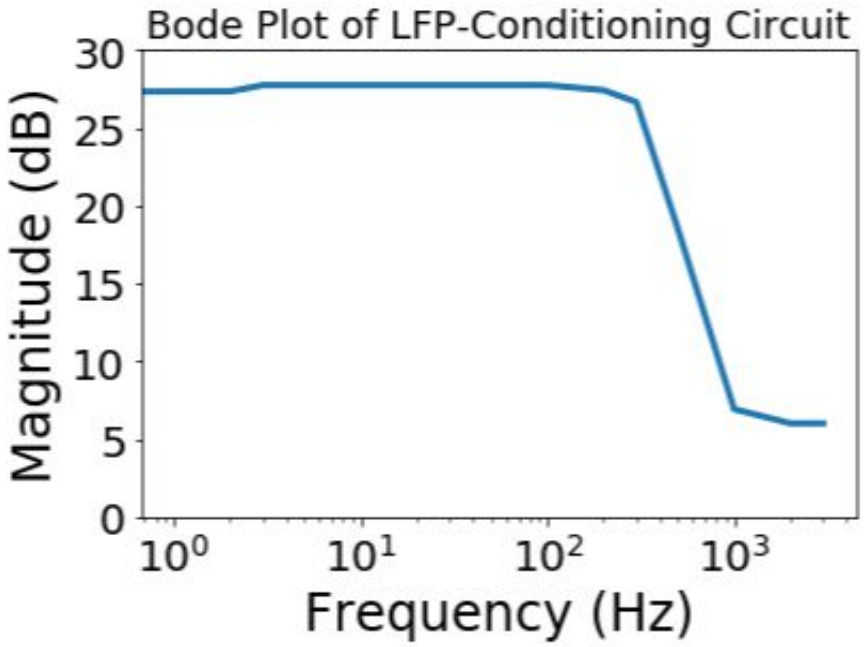


Figure 1 *Bode plot of the LFP-Conditioning Circuit, with magnitude in dB on the y-axis and frequency in Hz on the x-axis. Not shown here is the -32.04 dB attenuation for a DC (0 Hz) signal by the AC coupler.*

SD Card write-time performance:

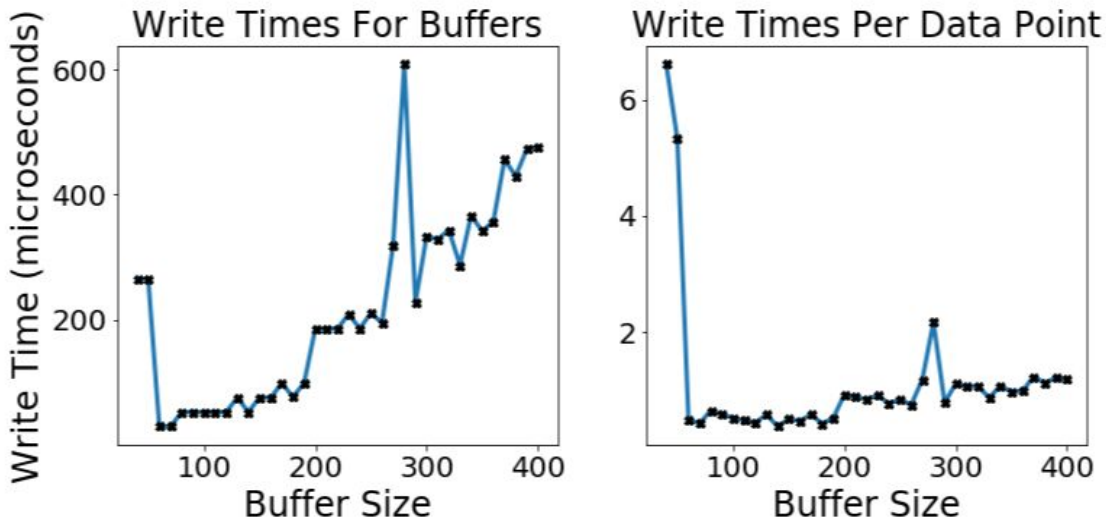


Figure 2 *Average write time in milliseconds as a function of the buffer size in two representations; the graph on the left represents the total average write time for each buffer size, while the graph on the right depicts the average write time per data point for each buffer size.*

Confusion Matrices:

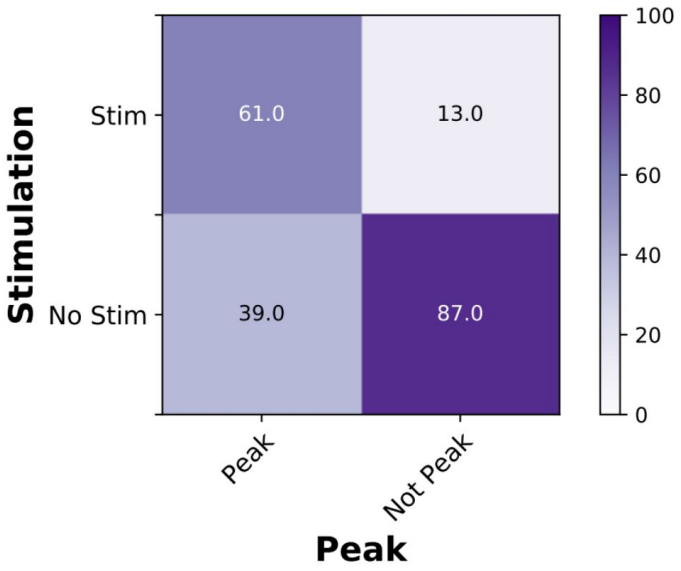


Figure 3: *Confusion Matrix of the Arduino Due implementation based on simulation data.*

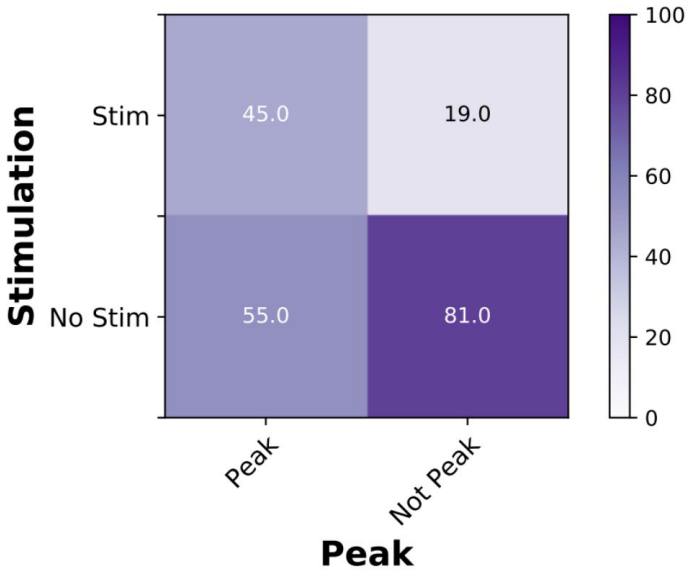


Figure 4: *Confusion Matrix of RTXI software implementation based on real - time experimental data.*