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.
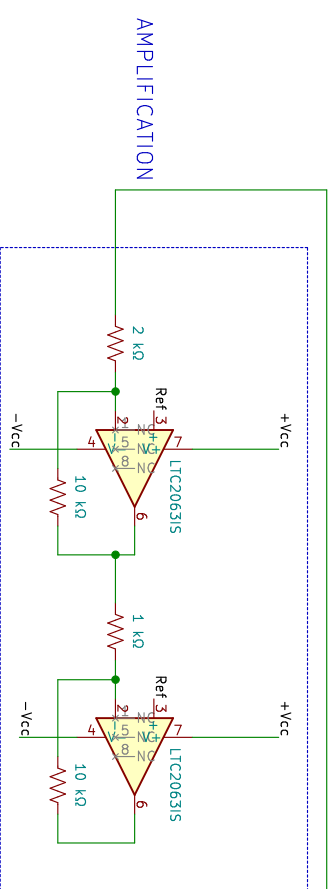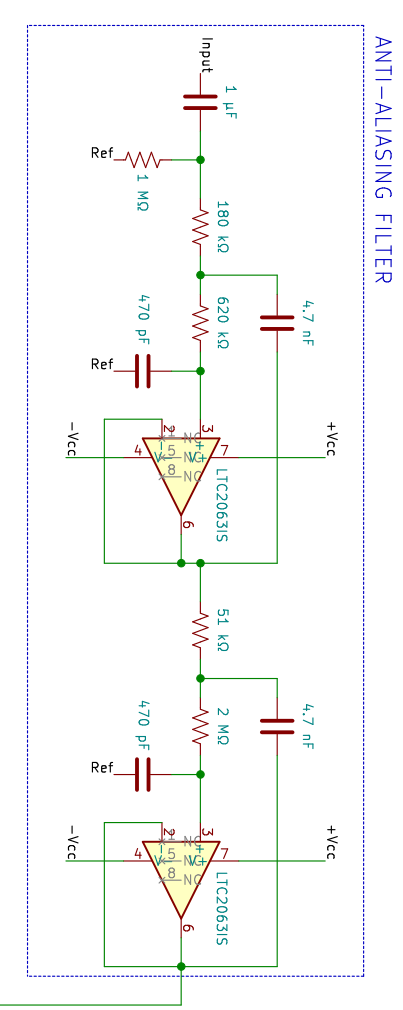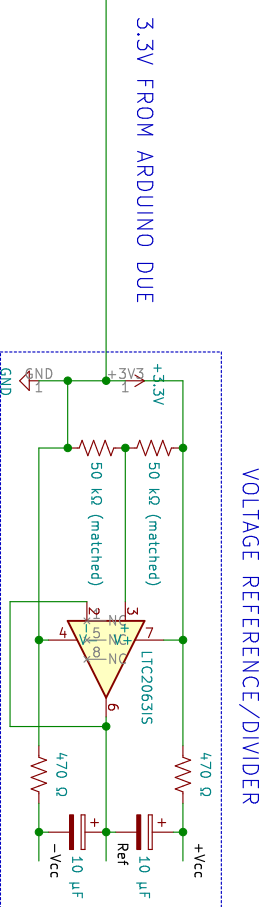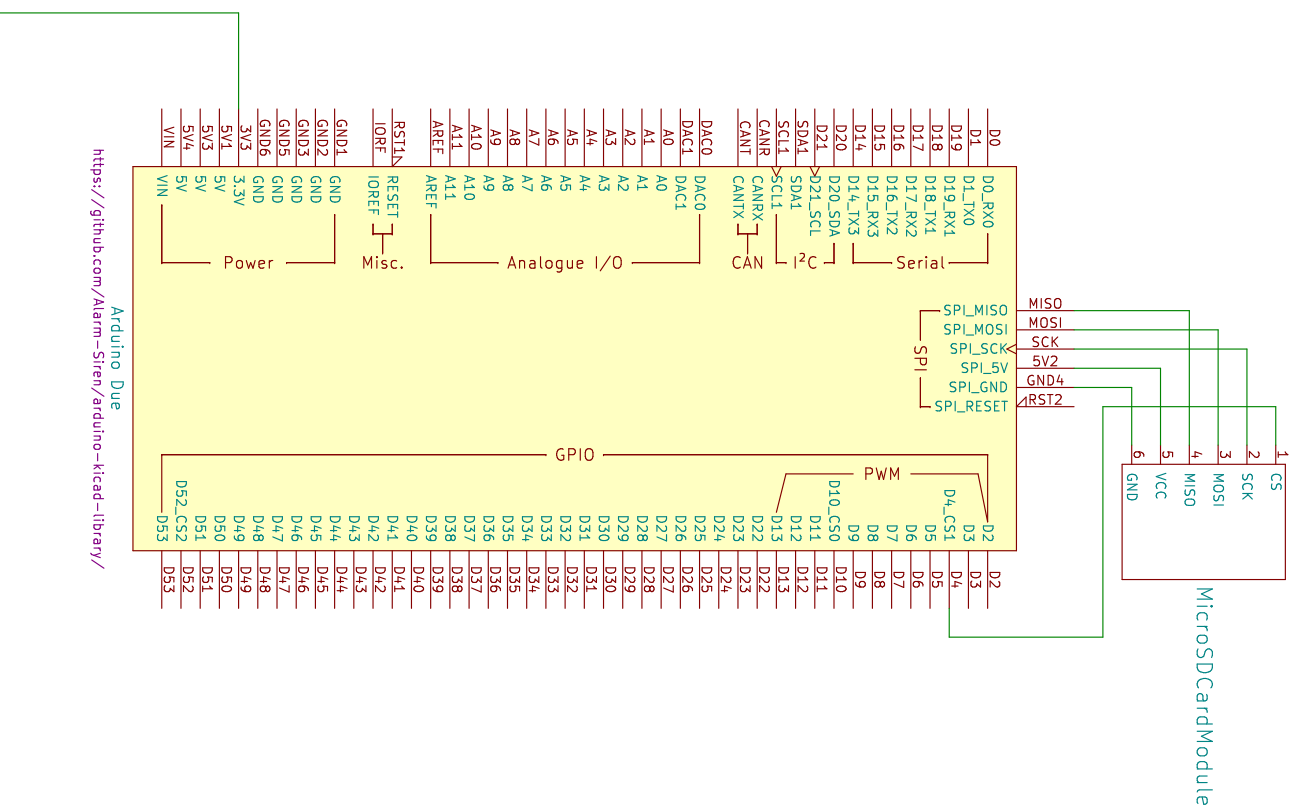
Hardware Setup

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.

Arduino Due

https://github.com/Alarm-Siren/arduino-kicad-library/

Power   Misc.   Analogue I/O   CAN   I²C   Serial

VIN 5V4 5V3 5V1 3V3 GND6 GND5 GND3 GND1 IORF RST1N AREF A11 A10 A9 A8 A7 A6 A5 A4 A3 A2 A1 A0 DAC0 DAC1 CAN1 CANR SDA1 SCL1 SDA1 D21 D20 D14 D15 D16 D17 D18 D19 D1 D0

VIN 5V 5V 5V 3.3V GND GND GND GND GND RESET IOREF AREF A11 A10 A8 A9 A7 A6 A5 A3 A2 A1 A0 DAC1 DAC0 CANTX CANRX SCL1 SDA1 D21_SCL D20_SDA D14_TX3 D15_RX3 D16_TX2 D17_RX2 D18_TX1 D19_RX1 D1_TX0 D0_RX0

SPI
SPI_MISO   MISO
SPI_MOSI   MOSI
SPI_SCK   SCK
SPI_5V   5V2
SPI_GND   GND4
SPI_RESET   RST2

GPIO   PWM

D53 D52_CS2 D51 D50 D49 D48 D47 D46 D45 D44 D43 D42 D41 D40 D39 D38 D37 D36 D35 D34 D33 D32 D31 D30 D29 D28 D27 D26 D25 D24 D23 D22 D13 D12 D11 D10_CS0 D9 D8 D7 D6 D5 D4_CS1 D3 D2

MicroSDCardModule

1 CS   2 SCK   3 MOSI   4 MISO   5 VCC   6 GND

ANTI-ALIASING FILTER

Input
1 µF
Ref 1 MΩ
180 kΩ
470 pF
620 kΩ
4.7 nF
Ref
LTC2063IS
-Vcc   +Vcc
LTC2063IS
51 kΩ
470 pF
2 MΩ
4.7 nF
Ref
-Vcc   +Vcc
LTC2063IS

AMPLIFICATION
2 kΩ
Ref
10 kΩ
-Vcc   +Vcc
LTC2063IS
1 kΩ
Ref
10 kΩ
-Vcc   +Vcc
LTC2063IS

3.3V FROM ARDUINO DUE

VOLTAGE REFERENCE/DIVIDER
GND   +3.3V
50 kΩ (matched)   50 kΩ (matched)
LTC2063IS
470 Ω   470 Ω
-Vcc   Ref   +Vcc
10 µF   10 µF

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.
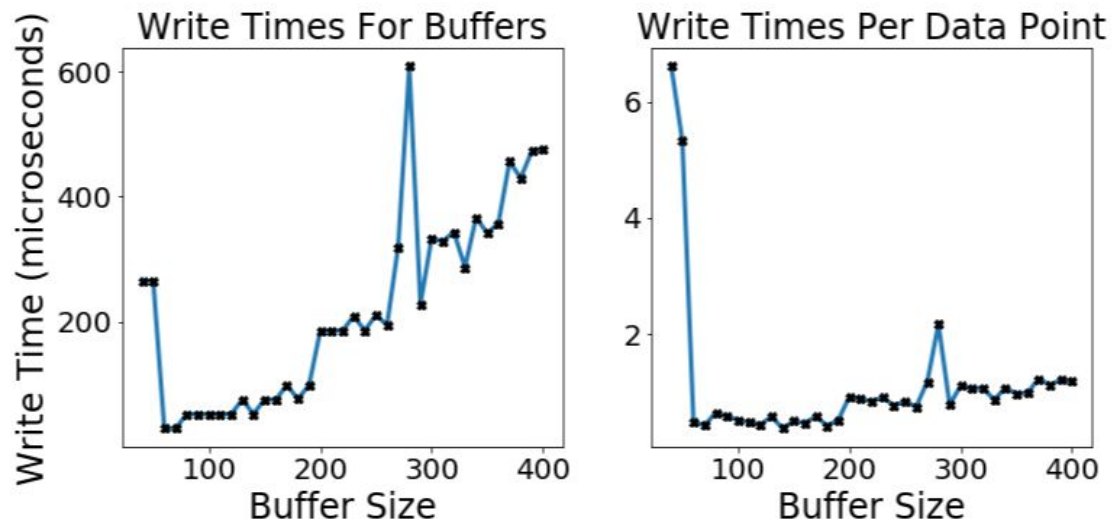
Results from Simulation Data and Other Metrics:

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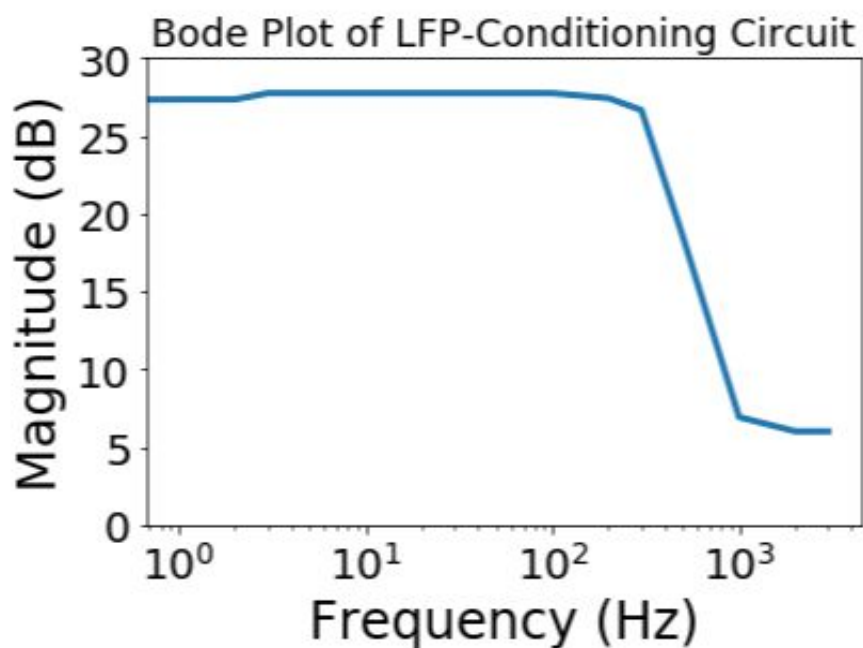
Figure 1 *Bode plot of the LFP-Conditioning Circuit, with magnitude in dB on the y-axis and frequency in Hz on the x-axis. Not shown here is the -32.04 dB attenuation for a DC (0 Hz) signal by the AC coupler.*

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:
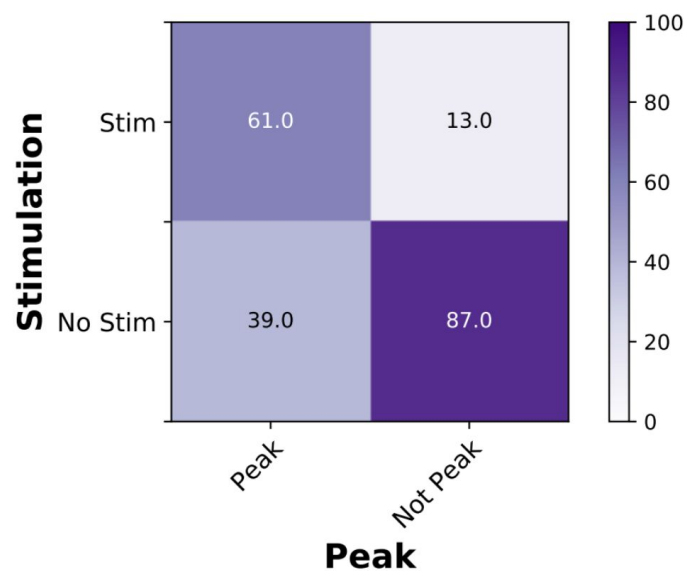
Figure 3: *Confusion Matrix of the Arduino Due implementation based on simulation data.*
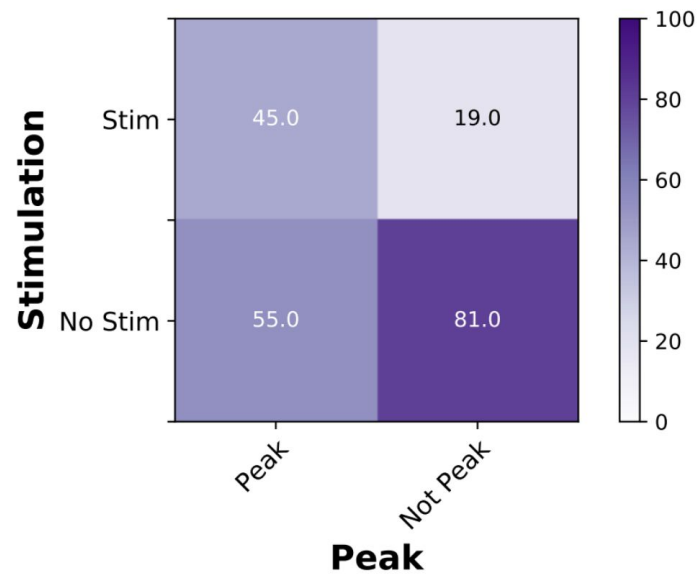
Figure 4: *Confusion Matrix of RTXI software implementation based on real - time experimental data.*

Published Code

(Will have a link once ownership of repository is transferred)