**Cambridge University Engineering Department**
**Engineering Tripos Part IIA**
**PROJECTS: Interim and Final Report Coversheet**

IIA Projects

**TO BE COMPLETED BY THE STUDENT(S)**

| | |
|---|---|
| Project: | SF4 Data Logger |
| Title of report: | Specification and Design – Interim Report<br>Group Report |

| Name(s): (capitals) | crsID(s): | College(s): |
|---|---|---|
| Krish Shah | Ks2027 | Girton |
| Jeremias Pohle | Jep78 | Churchill |
| | | |
| | | |

Declaration for: Interim Report

**We confirm that, except where indicated, the work contained in this report is our own original work.**

## Instructions to markers of Part IIA project reports:

### Grading scheme

| **Grade** | A* | A | B | C | D | E |
|---|---|---|---|---|---|---|
| **Standard** | Excellent | Very Good | Good | Acceptable | Minimum acceptable for Honours | Below Honours |

Grade the reports by ticking the appropriate guideline assessment box below, and provide feedback against as many of the criteria as are applicable (or add your own). Feedback is particularly important for work graded C-E. Students should be aware that different projects and reports will require different characteristics.

*Penalties for lateness: Interim Reports: 3 marks per weekday; Final Reports: 0 marks awarded – late reports not accepted.*

### Guideline assessment (tick one box)

| A*/A | A/B | B/C | C/D | D/E |
|---|---|---|---|---|
| | | | | |

| Marker: | | Date: | |
|---|---|---|---|

**Delete (1) or (2) as appropriate (for marking in hard copy – different arrangements apply for feedback on Moodle):**

**(1) Feedback from the marker is provided on the report itself.**

**(2) Feedback from the marker is provided on second page of cover sheet.**

| | Typical Criteria | Feedback comments |
|---|---|---|
| **Project Skills, Initiative, Originality** | Appreciation of problem, and development of ideas | |
| | Competence in planning and record-keeping | |
| | Practical skill, theoretical work, programming | |
| | Evidence of originality, innovation, wider reading (with full referencing), or additional research | |
| | Initiative, and level of supervision required | |
| **Report** | Overall planning and layout, within set page limit | |
| | Clarity of introductory overview and conclusions | |
| | Logical account of work, clarity in discussion of main issues | |
| | Technical understanding, competence and accuracy | |
| | Quality of language, readability, full referencing of papers and other sources | |
| | Clarity of figures, graphs and tables, with captions and full referencing in text | |

# SF4 - Interim Report - Group DL1

## Introduction:

The aim of this project is to build a device recording and analysing some data using an Arduino microcontroller and components bought with a budget of £15. This group decided to attempt to build a rudimentary electrocardiogram. This device will by no means be accurate to the levels required by medical staff, but should be able to give a reasonable indication of potentials across the cardiac vectors.
This report will give an outline on the progress the group has made so far and explain some of the basic electrical and software engineering concepts used.
As suited to their respective skill sets and interests, the team members have decided to divide the task into hardware and firmware design, as well as communications and software design. Despite both requiring vastly different skills, both team members have to communicate requirements to each other and agree on common features, such as communication protocols.

## Communication Protocols

Thinking of the OSI model, we can consider two different layers of communication between the Arduino and the laptop:

**At the Media Layer,** we have protocols that describe the transmission and framing of raw bit streams between two devices. We are transmitting data through a cable attached from the Arduino Uno USB-B serial port to the Laptop. The UART protocol determines how data is split into frames and transmitted through this cable. We set UART to operate at the maximum baud rate of 115200, 8 data bits and one parity check bit.

**At the Host layer,** we have protocols used for reliable transmission and character encoding:
- TCP-style messages with acknowledgment are used for Laptop to Arduino messages where messages are sparse and reliability is important. UDP-style packets without acknowledgement are used for messages from Arduino to Laptop
- MessagePack is used as the framework for encoding communication between devices as it represents a good tradeoff between implementation ease, reliability and overhead:

| Encoding Format | Message | Encoded Message | Length |
|---|---|---|---|
| Ascii String, comma delimited | 234,567,345,123 | 32 33 34 2c 35 36 37 2c 33 34 35 2c 31 32 33 2c | 16 Bytes |
| MessagePack | 234,567,345,123 | 94 cc ea cd 02 37 cd 01 59 7b | 10 Bytes |
| Raw Binary | 234,567,345,123 | 00 ea 02 37 01 59 00 7b | 8 Bytes |

## Analogue circuitry

The analogue circuit of the ECG consists of multiple op-amp based instrumentation amplifiers, followed by Butterworth 2-pole VCVS filters. With the goal being to record three channels at a time (those being I, II, III and aVF, aVR, aVL, as shown in the figure in the appendix), three sets of equivalent amplifiers are implemented. Due to the Arduino's analogue to digital converters only being able to work with voltages from ground to +5V, the amplifiers need to offset the signal. Further, the supply rail is at a relatively low potential difference for most common op-amps. Hence, the chosen components are required to operate at rail-to-rail output and at low voltages.
The team also chose to only record three traces at a time, keeping the required hardware simple and cheap, as well as reducing the requirements on the communications protocol. To still be able to assess all six different potentials at full accuracy, two different analogue signal paths through the amplifiers are required. This can be achieved by using a digitally controlled analogue SPDT switch on the input signals. A 2:1 analogue multiplexer / demultiplexer is used for this purpose, together with different buffer arrangements to give the required connections.

## Arduino firmware

The firmware running on the Arduino has to be streamlined to reduce the workload on the processor to an achievable level and allow for high sampling rates. Hence, only the most basic of operations are performed prior to sending data to the computer for further processing. Mainly, the firmware sets output pins for the required routing of signals through the analogue circuit described above, and records the amplified signal from the electrodes. The measurements are combined into a single data container together with the time passed since the last measurement and then sent through the serial connection to the computer. If the user requests a change in the measured quantities or decides to stop the measurements, the Arduino reacts to the received serial message accordingly.
A flow chart of this process is detailed in the appendix.

## Software

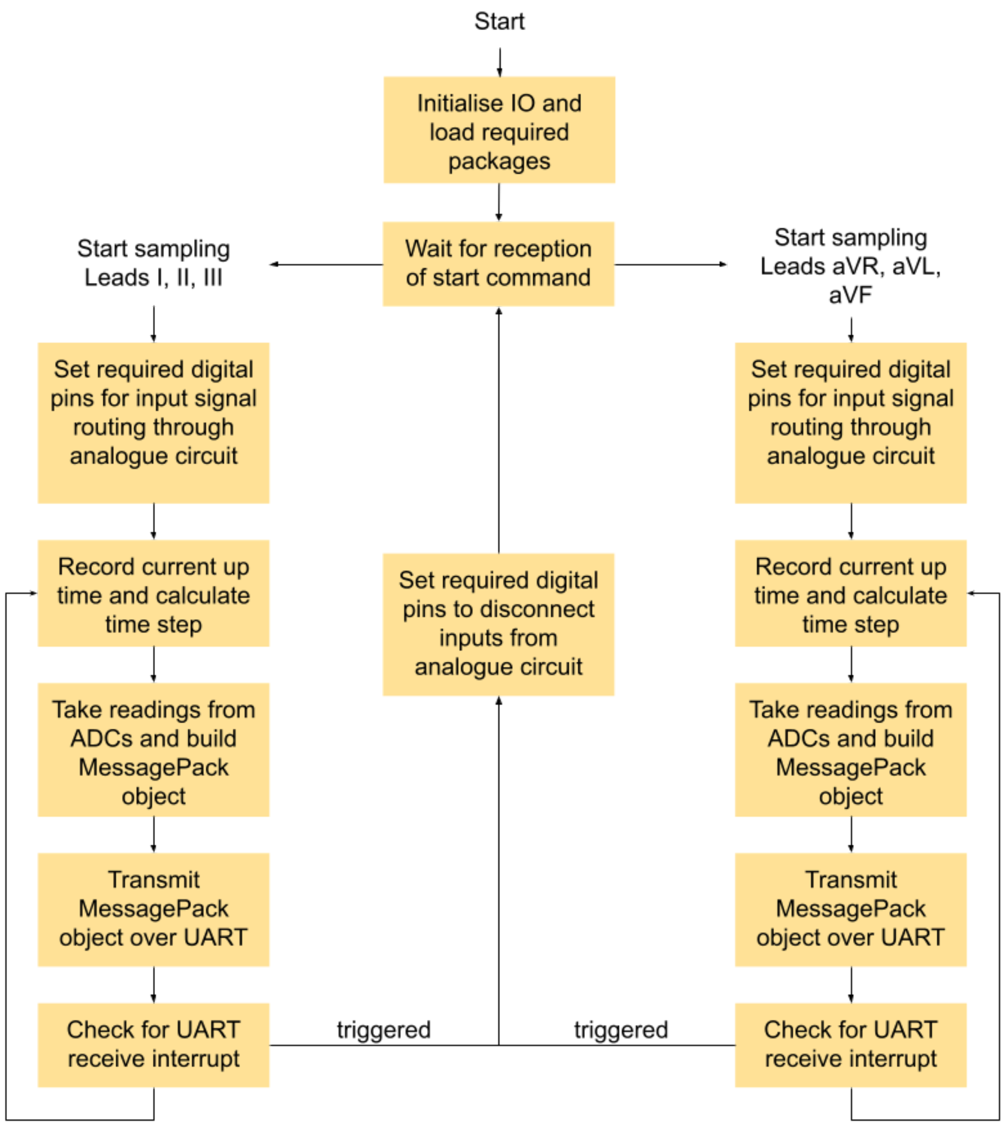
Unique to our project is the choice of browser based software to view data from the data logger. By simply visiting [https://sf4-data-logger.netlify.app/](https://sf4-data-logger.netlify.app/), end users can view the full instructions, download compiled firmware and operate the device without any installation requirements.
The browser uses Javascript as an execution environment, which is inherently better at parallel execution than Python. This allows us to better manage asynchronous events, eg. writing / reading from serial ports, awaiting user button presses, awaiting signal processing results.

## Conclusion

This report outlines the current progress and intentions of the team. Satisfactory progress has been made, and we expect to produce a working prototype by the end of next week.

# Appendix

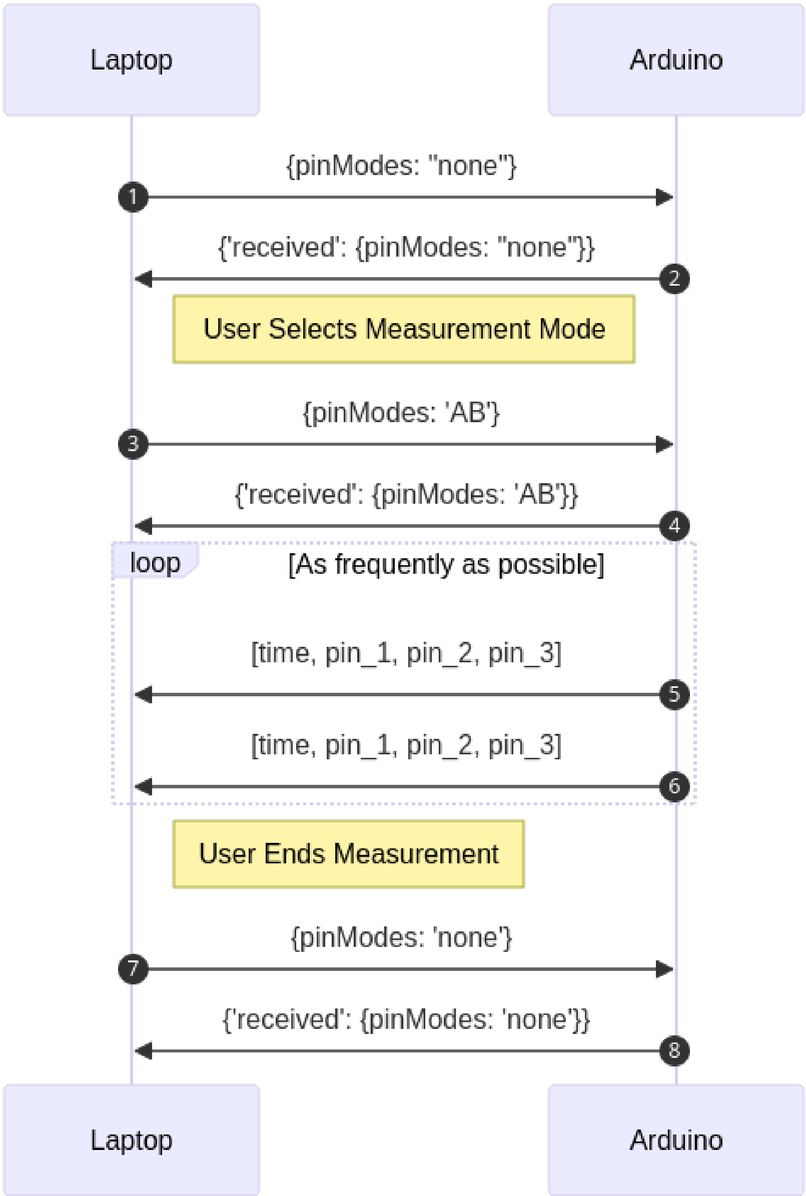
Arduino firmware flow chart:



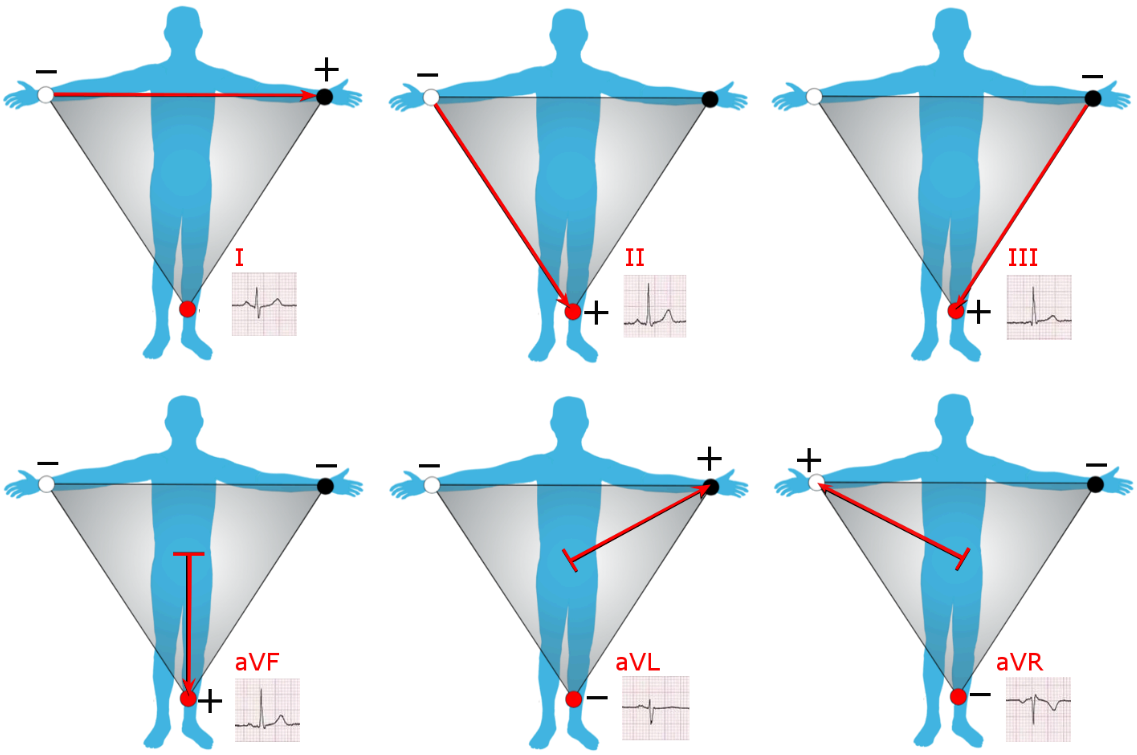

The firmware on the Arduino will be following this flow diagram. Executable actions are shown in yellow boxes, white boxes indicate context for states.

Communications Diagram:

An example exchange between devices. When the connection is first established, the laptop sends a request (1) to the arduino and checks the response (2). When the user selects a certain measurement mode, a message is sent to the arduino (3) and the arduino acknowledges this message (4). The Arduino sends the corresponding data (5, 6…) at a high data rate. The user ends measurement, and the laptop sends the corresponding message to the Arduino (7), which is acknowledged (8).



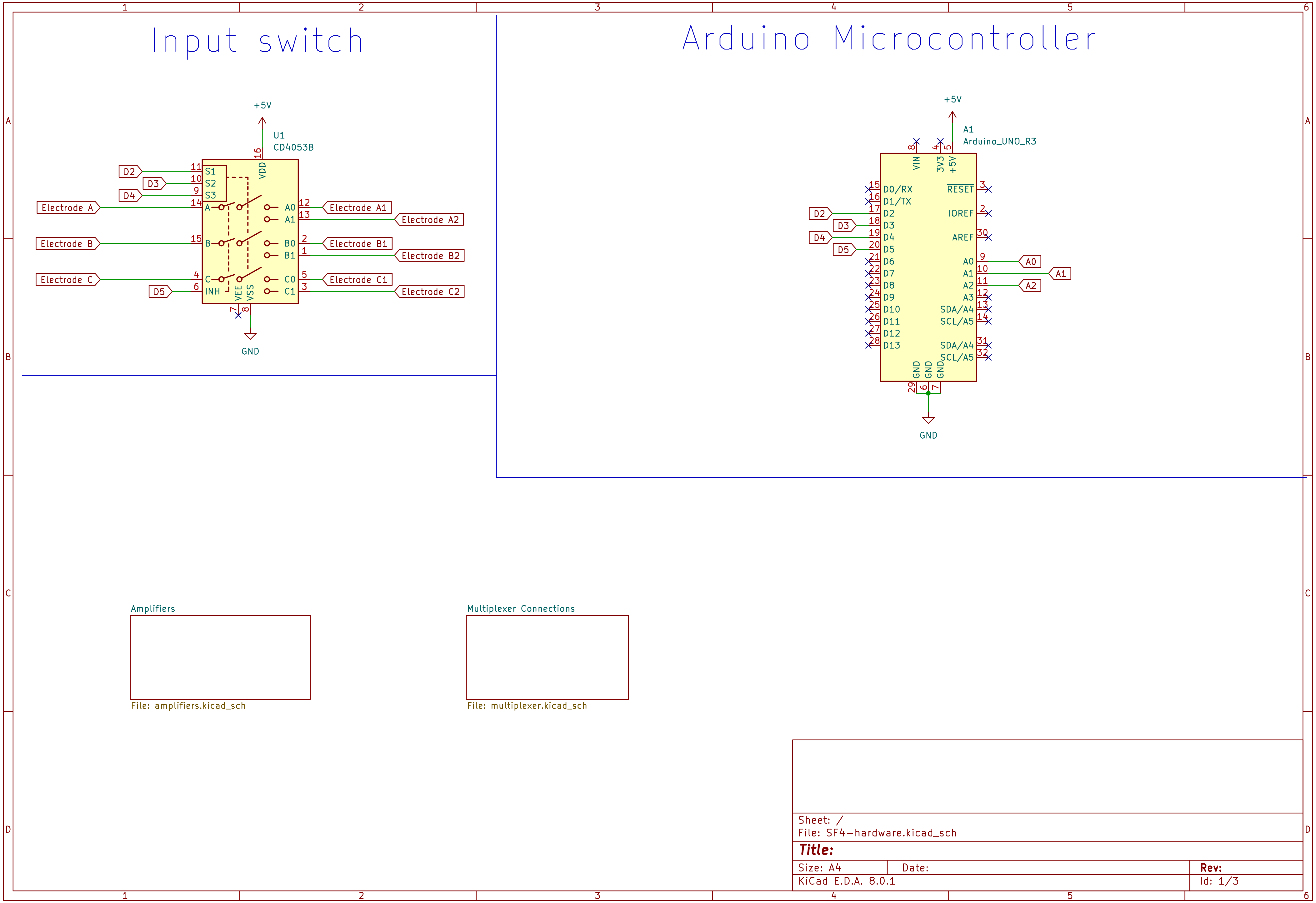
Monitored connections on the ECG:

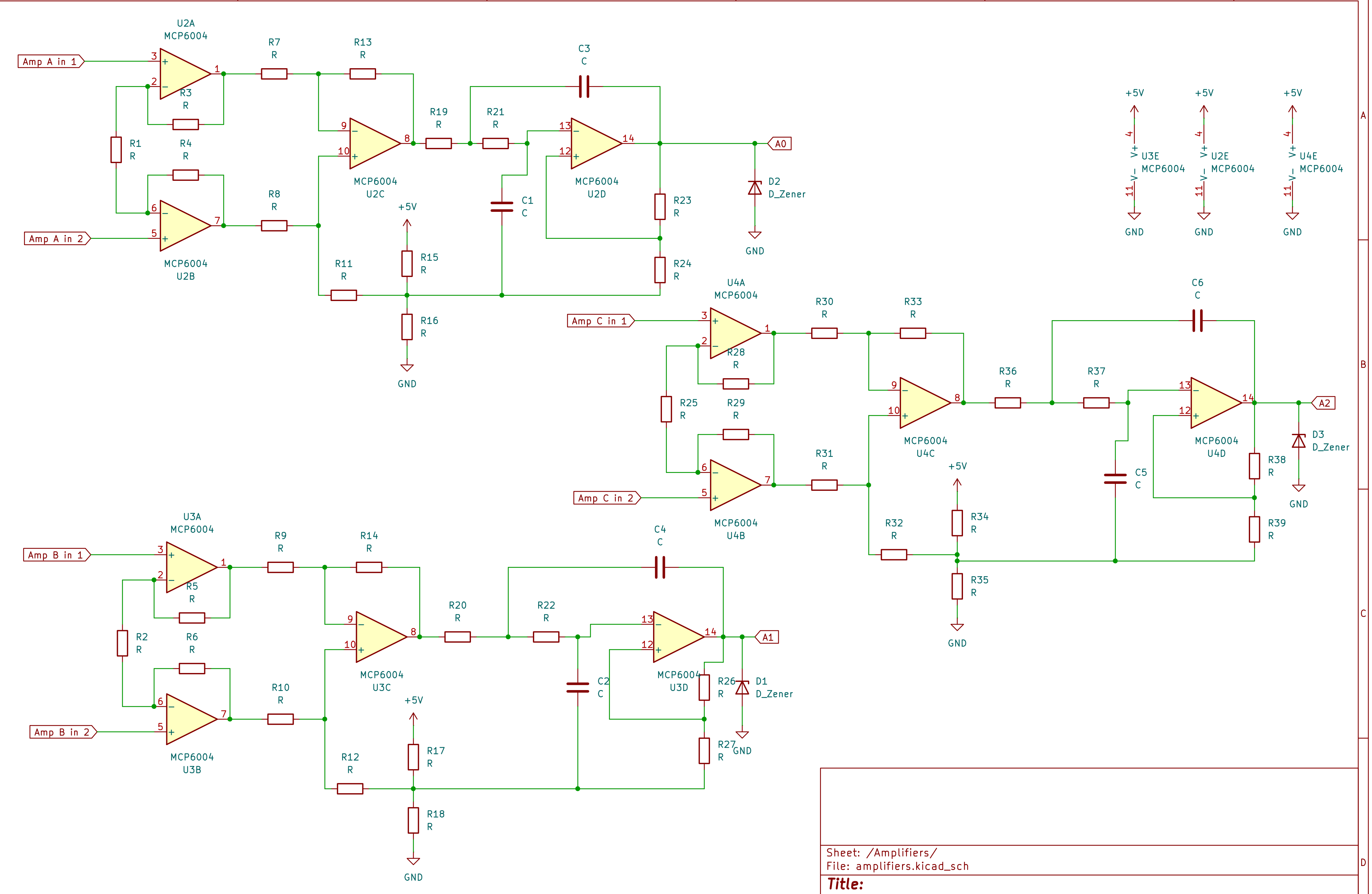



Accessed on Wikipedia 18/05/2024 -
https://en.wikipedia.org/wiki/Electrocardiography#/media/File:Limb_leads_of_EKG.png

Parts list for purchasing:

| Onecall order code | Component description | Qty | Unit price | Total price / £ |
|---|---|---|---|---|
| 1605571 | MCP6004-I/P Quad Op-Amp | 5 | 0.478 | 2.39 |
| SC18013 | ONSEMI IN5231CTR | 5 | 0.1 | 0.5 |
| SC16576 | TI CD4053BE | 1 | 0.48 | 0.48 |
| | | | | 3.37 |

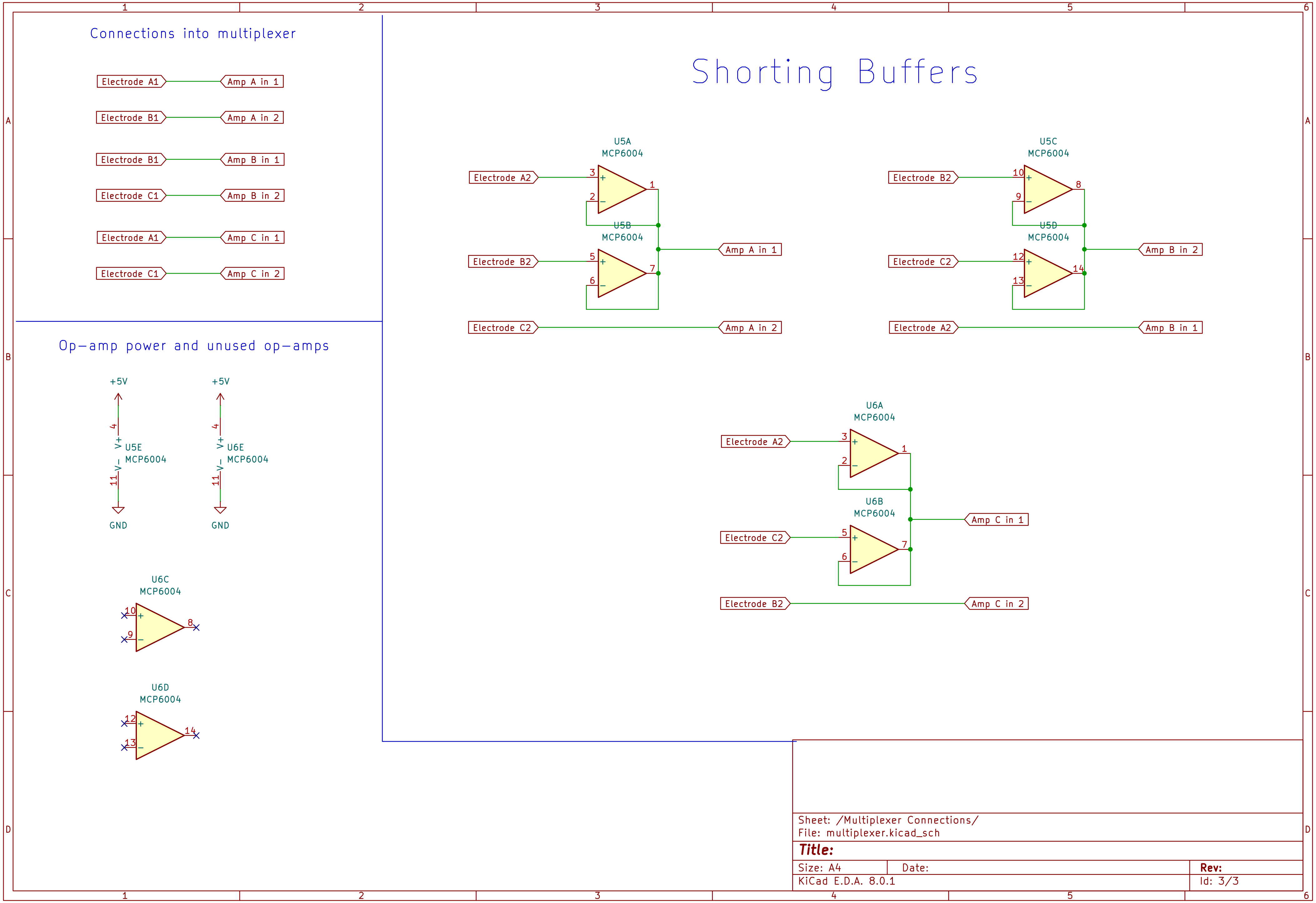

Input switch
Arduino Microcontroller
+5V
U1
CD4053B
VDD
D2
D3
D4
D5
S1
S2
S3
A
B
C
INH
VEE
VSS
A0
A1
B0
B1
C0
C1
Electrode A
Electrode B
Electrode C
Electrode A1
Electrode A2
Electrode B1
Electrode B2
Electrode C1
Electrode C2
GND
A1
Arduino_UNO_R3
VIN
3V3
+5V
D0/RX
D1/TX
D6
D7
D8
D9
D10
D11
D12
D13
RESET
IOREF
AREF
A2
A3
SDA/A4
SCL/A5
Amplifiers
File: amplifiers.kicad_sch
Multiplexer Connections
File: multiplexer.kicad_sch
Sheet: /
File: SF4-hardware.kicad_sch
Title:
Size: A4
Date:
Rev:
KiCad E.D.A. 8.0.1
Id: 1/3




Sheet: /Amplifiers/
File: amplifiers.kicad_sch

**Title:**

| Size: A4 | Date: | Rev: |
|---|---|---|
| KiCad E.D.A. 8.0.1 | | Id: 2/3 |



Shorting Buffers
Connections into multiplexer
Electrode A1
Amp A in 1
Electrode B1
Amp A in 2
Electrode B1
Amp B in 1
Electrode C1
Amp B in 2
Electrode A1
Amp C in 1
Electrode C1
Amp C in 2
Op-amp power and unused op-amps
+5V
+5V
U5E
MCP6004
U6E
MCP6004
GND
GND
U6C
MCP6004
U6D
MCP6004
U5A
MCP6004
Electrode A2
U5B
MCP6004
Electrode B2
Amp A in 1
Electrode C2
Amp A in 2
U5C
MCP6004
Electrode B2
U5D
MCP6004
Electrode C2
Amp B in 2
Electrode A2
Amp B in 1
U6A
MCP6004
Electrode A2
U6B
MCP6004
Amp C in 1
Electrode C2
Electrode B2
Amp C in 2
Sheet: /Multiplexer Connections/
File: multiplexer.kicad_sch
Title:
Size: A4
Date:
Rev:
KiCad E.D.A. 8.0.1
Id: 3/3