UNIVERSITY OF CAPE TOWN

EEE3000X

VACATION WORK

EIT: Hardware design

Author:
Matteo Kalogirou

Student Number:
KLGMAT001

14 July 2019

Declaration

1. I know that plagiarism is wrong. Plagiarism is to use another's work and pretending that it is one's own.

2. I have used the IEEE convention for citation and referencing. Each contribution to, and quotation in, this report from the work(s) of other people has been attributed, and has been cited and referenced.

3. This report is my own work.

4. I have not allowed, and will not allow, anyone to copy my work with the intention of passing it off as their own work or part thereof.

Name: Matteo Kalogirou

Student Number: KLGMAT001

Date: 14 July 2019

CONTENTS

1 Introduction **1**
1.1 Background . . . 1
1.2 Problem Statement . . . 2
1.3 Current EIT Hardware . . . 2
1.4 Tomo . . . 2
1.5 Precise Hardware EIT . . . 3
1.6 Report Plan . . . 4

2 Signal Generator **5**
2.1 PWM and Filtering . . . 5
2.1.1 Sallen-Key Bandpass Filter . . . 6
2.1.2 Multiple Feedback Bandpass Filter . . . 7
2.2 Direct Digital Synthesis . . . 7
2.3 555 Timer . . . 8

3 Sampling System **9**

4 Current Source **10**
4.1 Howland Current Source (HCS) . . . 11
4.2 Enhanced Mirrored Howland Current Source (EMHCS) . . . 12
4.3 EMHCS Implementation . . . 14

5 Signal Acquisition **15**
5.1 Voltage Measurement Circuit . . . 16
5.2 Bipolar to Polar Signal Converter . . . 16

6 System Integration **18**

7 Experimental Results **18**
7.1 Rotating measurement electrodes . . . 18

8 Discussion **19**

9 Conclusions **21**

A Software Implementation **i**
A.1 Voltage Measurement Electrode Rotation . . . i

1 INTRODUCTION

This report concerns the design and implementation of a simple, low-cost, time division multiplexed, 16-electrode Electrical Impedance Tomography (EIT) instrument. There are many form of the standard EIT implementation regarding the type and number of current sources, signal stimulation method and measurement process. However, this report focuses on a simple, single current source applying an adjacent injection pattern which is measured using an array of instrumentation amplifiers.

The EIT instrumentation can be divided into six main subsystems. These include:

1. The signal generator
2. The current source
3. The electrode array and test body
4. The voltage measurement circuit
5. The sampling circuit
6. The reconstruction unit

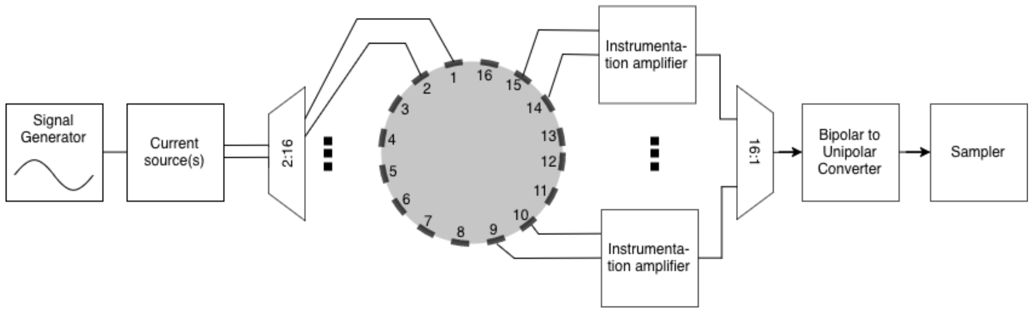


Figure 1: Proposed system architecture.

This report will focus on subsystems 1,2,4 and 5.

1.1 Background

Electrical Impedance Tomography is a non-invasive medical imaging modality which involves measuring a conductive body's impedance to generate a rough pixel image of the body's internal structure. This is achieved through injection of precise amounts of current across an array of electrodes placed on a conductive body's surface. The resulting voltages generated at each of the non-current conducting electrodes are then measured and used in a reconstruction algorithm to determine an approximate impedance map for the internal structure of the conducting body [1].

Clearly, this technology has significant merit based on its potential for a highly portable, non-ionising, and inexpensive to operate biomedical imaging technique [1] [2].

1.2 Problem Statement

There are many technical challenges involved in creating a functioning EIT system. Since the hardware is responsible for creating the data used in the reconstruction, a poor hardware implementation will certainly result in a poor system no matter how complex and impressive the reconstruction algorithm is [1]. For this reason, a robust system must be used.

The most significant factors contributing to a suitable EIT system can be simplified to the following points:

- Implementing a stable current source which is invariable to changes in load impedance
- Voltage measurement instruments must have a large input impedance and good common mode rejection ratio
- Sensitivity of the voltage measurement circuit must be carefully set to detect small changes
- The bandwidth of the system must be sufficient to faithfully represent the sampled signal

The subsequent Section reviews current EIT hardware implementations and their solutions to the challenges of EIT.

1.3 Current EIT Hardware

EIT systems emerged during the early 1980's [1] and were predominantly analog devices. However, technological innovation has seen most current EIT implementations becoming more dependent on digital electronics. Consequently, EIT systems can now be made highly portable to the extent that some authors have developed wearable EIT systems intended for integration into smart watches. This Section presents several imaginings of the EIT system.

1.4 Tomo

Tomo, an abbreviation of the word *tomography*, is a low cost, wearable EIT device developed by Zhang and Harrison of Carnegie Mellon University [2]. The device utilises EIT to examine a cross-section of the user's forearm. Machine learning (ML) is applied to the resulting image to classify the user's wrist position and hand gesture. The authors hope that future works may be able to integrate the technology into smartwatches, expanding the human-computer interface beyond the conventional keyboard and mouse.

This device comprises of an 8-electrode system attached to an adjustable band worn on the forearm. The signal generation, impedance analysis, and ADC is performed by an Impedance Analyser IC, the AD5933. This IC features an onboard direct digital synthesis (DDS) signal generator, 12-bit ADC with anti-aliasing filter with a sample speed of 10^6 samples/s and a programable gain amplifier [3]. The device is shared amongst the electrode by two 8:1 CMOS analog multiplexers (ADG1608) [2].

The AD5933 applies the Discrete Fourier Transform to the measurement samples to determine the impedance between each electrode pair. The authors choose to use the computationally efficient Linear Back Projection Method for image reconstruction. A frame rate of approximately 10fps is achieved. These electrode-pair impedances, along with the differential impedance calculated between each initial electrode-pair impedance formed the basis features for the ML algorithm. The classification problem is solved using a Support Vector Machine.

1.5 Precise Hardware EIT

Iranian authors from various universities have shown that a highly precise, low-cost 32 electrode EIT system can be implemented with relatively simple analog circuit design [4]. The authors argue that due to the challenging nature of the EIT reconstruction problem, accurate and precise hardware design is absolutely necessary to provide data of the highest quality.

The authors describe in detail an EIT system which is subdivided into a voltage controlled current source (VCCS), a multiplexer module, a voltage measurement part and a control unit. Their VCCS is a combination of a voltage controlled oscillator, a Butterworth bandpass filter and a voltage to current converter.

The authors emphasise reducing the output impedance of the oscillator and maximising the output impedance of the voltage to current converter (VCC). A triple-operational amplifier form is chosen as it exhibits an excellent operational bandwidth a $\mathrm{M}\,\Omega$ output impedance and can support a wide range of load impedances [4]. The current source is shared amongst the system electrodes by four 16:1 multiplexers (ADG506AKN).

A fourth-order high-pass Butterworth filter is cascaded with a fourth-order lowpass Butterworth filter to produce a bandpass filter with a bandwidth of 10 kHz to 250 kHz. This BPF separates the oscillator from the VCC.

To measure the boundary voltages, the authors employ a pulse generator with zero and peak current detection, used to demodulate the boundary voltages [4]. This pulse generator is placed after the VCCS subsystem.

The voltage measurement circuit combines a demodulation circuit which uses pulse sample demodulating to measure the load voltage at the current peaks. This extracts the real load voltage. These measurements are then fed to a programable gain amplifier to determine the differential voltage between adjacent electrodes. Another four 16:1 multiplexers are required to read the voltage measurements [4].

A control unit in the form of the ATmega128 is used to co-ordinate the multiplexers and ADC, communicate measurements to a PC and output the voltage reading to an LCD screen.

Due to the precise timing required by the ADC operation, an external ADC is employed in favour of the ATmega's onboard ADC. The AD1674 has a 12-bit resolution and can read bipolar signals in the range pf $\pm 5\,V$.

The authors tested a 16-electrode version of their system with various phantoms in a saline solution using an adjacent current injection pattern and reconstructed the images using the *EIDORS Matlab* package. The authors compare their system's reconstructed images to that of two other 16-electrode EIT systems; one of which is a DSP based multi-frequency system. Visual comparison reveals significantly better results using the author's hardware [4]. The author's note the high resolution and precision achieved by their system with relatively simple circuit design [4].

1.6 Report Plan

The system is divided into functional subsystems which are designed individually according to the requirements of each local application.

Section 2 investigates potential microcontroller based signal generation methods. The sampling requirements of the microcontroller are addressed in Section 3. The generated sinusoid must then be converted to a current by an appropriate voltage controlled current source, discussed in Section 4. Section 5.1 details the hardware used to measure the resulting voltages at each of the non-conducting electrodes due to the excitation. The subsystems are then integrated in Section 6 with the results from the finalised system presented in Section 7.

2 SIGNAL GENERATOR

Ideally, an EIT is a self contained unit with the means to generate an input signal which is applied to the test body.

This section will outline three different methods of generating a sinusoidal voltage waveform used to drive the voltage controlled current source. This system aims to be applied in the biomedical field. For this reason, the applied current is required to be sinusoidal so as to prevent ionisation of the conducting material.

Although it is not strictly necessary to apply a purea sinusoidal signal to the system, pure sinusoids are eigenfunctions. This implies that the system response to a pure sinusoid is expected to be a scaled and shifted version of the input frequency. The complexity of analysing the system response is therefore reduced.

An important considerations for the signal generating system is that a single period is generated at time. In this way, the relative phase shift between input and output signals can be measured. Furthermore, the sinusoid must be spectrally pure with few harmonics. We will assume that the desired sinusoidal frequency is 150 Hz, as explained in Section 3, with a peak-to-peak voltage of 1 mV. The subsequent subsections present methods of generating a pure waveform using a microcontroller.

2.1 PWM and Filtering

This method of generating a sinusoid requires a filtering process to remove the harmonics from the square wave pulses. The MCU generates a pulse width modulated (PWM) signal at a specified frequency. A simple RC filter can be used to remove the harmonics, however, the waveform will have a DC offset. To remove the offset, a DC blocker is required. Alternatively, a narrow bandpass filter can be used to isolate the signal frequency.

This subsystem is shown in the Figure below:

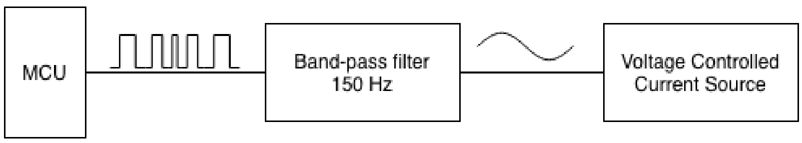


Figure 2: Sinusoidal signal generation by filtering harmonics of a PWM signal.

The process can be controlled directly by the MCU allowing precise control over the waveform application. Additionally, synchronisation with the sampling subsystem can easily be implemented.

Ideally an active bandpass filter is used to isolate the desired frequency. Two bandpass filter designs are considered: the Sallen-Key and Multiple Feedback (MFB) topologies.

2.1.1 Sallen-Key Bandpass Filter

The circuit diagram of a generic Sallen-Key BPS filter is seen in the Figure below [5].

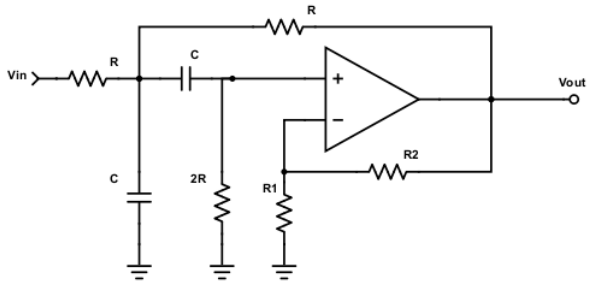


Figure 3: Sallen-Key Bandpass filter topology.

The *Quality Factor* (Q), of the filter is a measure of how narrow the pass-band is. A higher Q results in a narrower pass-band. Since this application is mostly concerned with the purity of the sinusoid, the design begins by specifying Q = 10. The following equations are used to calculate the component values for a topology with a centre frequency, f_m at 150 Hz and a quality factor of 10 [5]:

$$f_m = \frac{1}{2\pi RC} \tag{1}$$

Setting R = 100 kΩ, C = 10.61 nF

$$Q = \frac{1}{3 - G} \tag{2}$$

Allowing Q = 10, G = 2.9

$$G = 1 + \frac{R_2}{R_1} \tag{3}$$

Non-inverting amplifier gain = 2.9. $\therefore \frac{R_2}{R_1} = \frac{19}{10}$

$$A_m = \frac{G}{3 - G} = 29 \tag{4}$$

The centre frequency gain is then 29.

(5)

The large gain associated with a large Q is problematic as the output signal is likely to saturate. This design is restrictive in terms of being able to set f_m and Q independently. The MFB topology allows for independent parameter specification.

2.1.2 *Multiple Feedback Bandpass Filter*

The generic circuit design for the MFB filter is shown in Figure 4 [5]. It should be noted that unlike the Sallen-Key, this topology is based on an inverting amplifier.

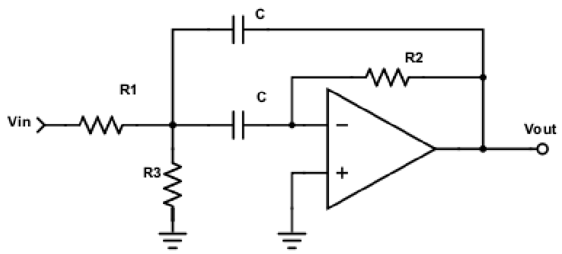


Figure 4: Multiple feedback bandpass filter topology.

The filter parameters are calculated according to the following rules [5]:

$$f_c = \frac{1}{2\pi C}\sqrt{\frac{R_1 + R_3}{R_1 R_2 R_3}} \tag{6}$$

$$Q = \pi f_c R_2 C \tag{7}$$

$$A_m = -\frac{R_2}{2R_1} \tag{8}$$

$$B = \frac{1}{\pi R_2 C} \tag{9}$$

2.2 *Direct Digital Synthesis*

A simple means of waveform synthesis utilises a direct digital synthesiser (DDS) IC. The DDS operates off a lookup table of stored sinusoidal values in the range of 0 to 2π. By selecting every k^{th} entry, the frequency of the output signal can be set [1].

A DAC then produces an analog waveform from these samples. Finally, the output signal is low-pass filtered to remove any spectral replicas introduced through the sampling process.

These systems are advantageous in that they can be controlled programmatically and will operate separately from the MCU controlling it. This should free the MCU to perform other concurrent operations such as voltage measurements.

2.3 555 Timer

A 555 timer can be configured to oscillate with microsecond precision [6]. This approach is appealing since the signal generation is outsourced to an external module. This decreases the number of operations to be performed by the MCU, freeing up clock-cycles for the sampling process. However, this approach introduces the added complexity of activating the waveform generator for a single period.

This may be performed by holding the **RESET** pin high until the pulse is required. The circuit diagram of the oscillator and resonant circuit is seen in the Figure below.

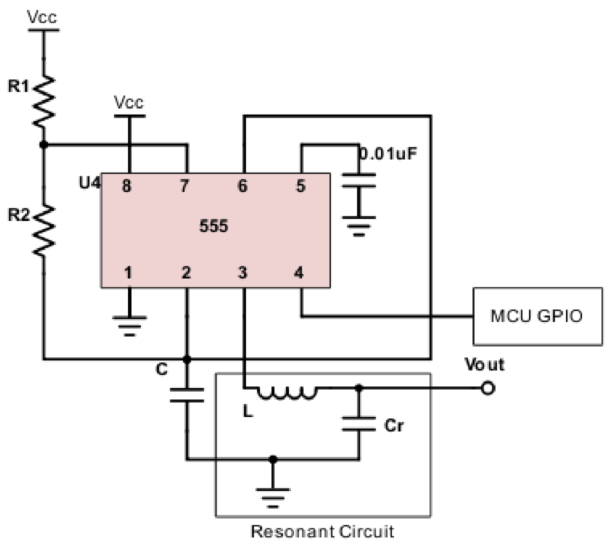


Figure 5: 555 Timer circuit set in astable mode to oscillate at f_c = 150 Hz with a 50% duty cycle. The output is connected to an LC resonant circuit to produce a sine wave at 150 Hz

Resistors R_1 and R_2 are used to set the duty cycle of the waveform. The oscillator's duty cycle is set to approximately 50% using the relationship:

$$\text{Duty Cycle\%} = \frac{R_1 + R_2}{R_1 + 2R_2} \tag{10}$$

Where $R_1 = 1\,k\Omega$ and $R_2 = 100\,k\Omega$ such that the duty cycle $\approx$ 50.02%. The frequency of oscillation is then determined by the following equation:

$$f_c = \frac{1.44}{(R_1 + 2R_2)C} \tag{11}$$

Solving for C at 150 Hz, we have $C = 47.76\,nF$.

The LC circuit must match the frequency of oscillation to cause resonance. The resonant circuit's frequency is governed by the equation:

$$f_r = \frac{1}{2\pi\sqrt{LC}} \implies L = \frac{1}{4\pi^2 f_r^2 C} \tag{12}$$

Therefore at 150 Hz, if we set $C_r = 1000\,\mu F$ then $L = 1.126\,mH$.

3 SAMPLING SYSTEM

As the co-ordinator and sampler, the MCU's clock-speed ultimately determines the bandwidth of the system. Since the bandwidth must be shared by each of the measurement electrodes, the total available system bandwidth is significantly reduced. We can relate the bandwidth of the system to the number of electrodes as:

$$BW \leq \frac{\text{MCU sample speed}}{\text{number of electrodes} \times 2} = \frac{9615}{16 \times 2} = 300.48\,Hz \tag{13}$$

The MCU used in this application has a clock-speed of 16 MHz. The onboard ADC operates at 1/128$^{\text{th}}$ of that and requries 13 clock-cycles. This provides an effective sampling speed of approximately 9615 samples/s. A factor of 2 is required to meet the Nyquist sampling criteria. A signal frequency of 150 Hz will then comfortably satisfy the limited bandwidth of the system as well as remaining sufficiently far from the 50 Hz mains supply frequency. The extra bandwidth should allow for intermediary operations involved in swapping between successive ADC reads.

In order to test practical sampling frequency of the Arduino's ADC, a simple script is written to toggle a digital output pin after successive ADC reads. The ADC value is then converted to a voltage after each read.

The measured frequency is approximately 4.3 kHz - significantly higher than the expected sample rate. However it should be noted that this is an optimistic estimation since only a single operation is performed between successive reads.

4 CURRENT SOURCE

Typically, an EIT system may be designed in one of two configurations: the first requiring N separate and individual current sources for an N-electrode system; and the second requiring a single current source to be shared amongst the N electrodes [1].

This configuration fundamentally determines how the EIT system behaves. For instance, in a frequency division multiplexed system: each of the electrodes is stimulated simultaneously by a slightly different frequency. This is not possible in a system with only a single current source.

Furthermore, the system can employ a variety of different current injection patterns. Each pattern specifies the relative orientation of electrodes which source and sink current. The injection pattern can be configured simply by altering the combination of active electrodes during the injection phase. Therefore, both current source configurations can vary the injection pattern. This system will employ the adjacent injection pattern seen in Figure 6.

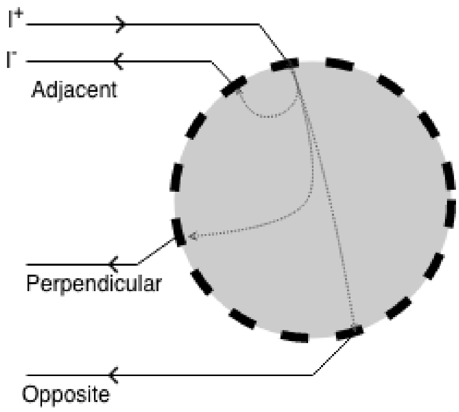


Figure 6: By alternating the source and sink electrodes, a variety of current injection patterns can be applied. The adjacent patterns employs two consecutive electrodes. The perpendicular injection pattern uses electrodes which are spaced at 90°apart and the opposite injection pattern uses electrodes placed 180°apart.

This application will only consider current injection at a single frequency. Therefore, the frequency characteristics of the circuit are not investigated. The quality of the current source is determined by how invariable the current supply is to the load impedance.

To achieve a consistent current supply across a range of loads, the current source must have a large output impedance [7]. This ensures that the system remains impartial to changing loads. The following configurations are discussed: the standard Howland current (HCS) source, the enhanced mirrored Howland current source (EMHCS).

4.1 Howland Current Source (HCS)

The Howland current source provides a simple solution to the problem employing a minimum number of components. This current source can be configured to provide a very high output impedance by carefully selecting the resistor values. This implementation is especially appealing due to its simplicity [1].

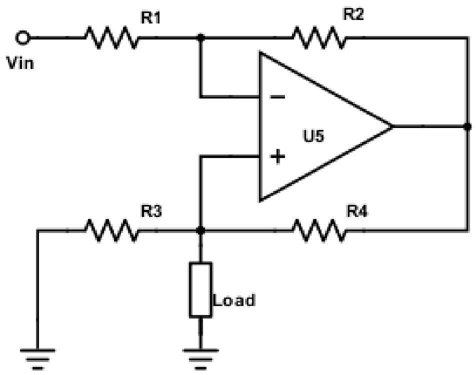


Figure 7: Simple Howland Current Source configuration.

If we assume a sufficiently ideal op-amp: infinite input impedance and zero output impedance, then the source can be made to have an infinite output impedance by satisfying the relationship [7]:

$$\frac{R_4}{R_3} = \frac{R_2}{R_1} \tag{14}$$

Which then implies that the load current can be defined as:

$$I_{load} = -\frac{V_{in}}{R_3} \tag{15}$$

This design requires two multiplexers to route current from the source to an electrode and then to route the sunk current from the adjacent electrode through to ground. This design can be improved somewhat to provide a more reliable current draw.

4.2 *Enhanced Mirrored Howland Current Source (EMHCS)*

This system expands on the standard HCS, by implementing a mirrored current source which provides current at an equal magnitude but opposite direction. Therefore, this creates a push-pull effect so that the current sourced equals the current sank [8].

This configuration makes slight modifications to the multiplexing circuit in that the current is now routed to the source electrode from one side of the mirror and then routed through the adjacent sink electrode to the other side of the mirror. This ensures a more consistent and precise current injection.

However, since this design utilises two active components, there is an unavoidable mismatch between output impedances of the two sources. The result is a finite impedance seen by the current between the conducting electrodes, which will then generate some common mode voltage. This is undesirable as the common mode voltage manifests as an artefact in the reconstruction process, thereby producing a less faithful reconstruction [8].

The circuit diagram of the EMHCS is seen in the Figure below [8].

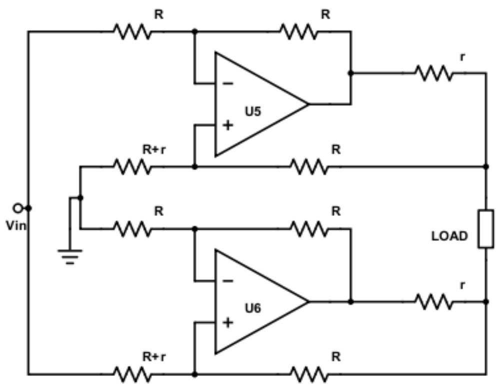


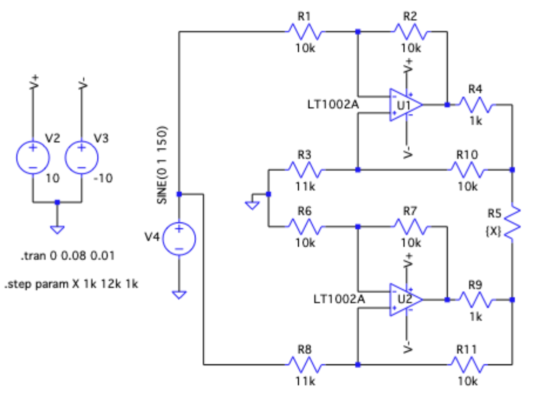
Figure 8: Enhanced Mirrored Howland Current source configuration.

The output current of the circuit is described by the following relationship [8]:

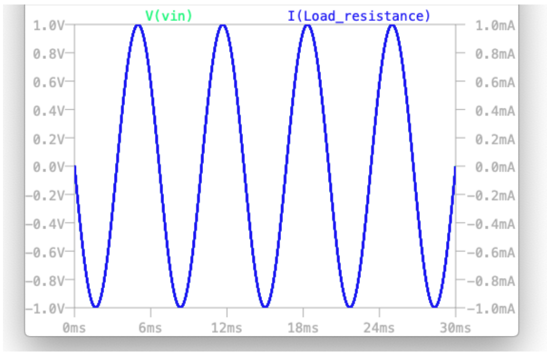
$$I_{load} = V_{in}/r \tag{16}$$

Applying an input voltage of $1V_{pp}$ with $r = 1k\,\Omega$ therefore produces a $1\,mA_{pp}$ current signal.

A simulation of the circuit is performed in LTSpice to verify the consistency of the supplied current signal. In the simulation, the load resistance is stepped from $1\,k\Omega$ through to $12\,k\Omega$ with no discernible change to the output signal.



(a) EMHCS simulated in *LTSpice* using JFET amplifiers and a 1 V_{pp} sinusoidal input voltage. The load resistance is varied to determine the current source stability.



(b) The input voltage signal (green) is completely masked by the output current signal (blue) for each load resistance. The output current has a constant swing of 1 mA regardless of the load resistance. This indicates a stable and effective current source.

Figure 9: LTSpice simulation of the EMHCS.

Theoretically, the EMHCS has a relatively high output impedance, which makes the current supply invariable of the load impedance. The output impedance of the system can be estimated by modelling the current source as an ideal current source with a parallel impedance, Z_{out}, as in Figure 10.

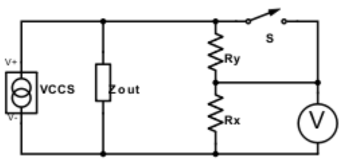


Figure 10: Circuit configuration to determine the VCCS output impedance. The VCCS is modelled by an ideal current source in parallel with the output impedance, Z_{out}.

The circuit's output impedance is then calculated according to [4]:

$$Z_{out} = \left[R_y - R_x \right] \frac{V_1}{V_2 - V_1} \tag{17}$$

Where V_1 is the voltage measured by the voltmeter with a full load, (S = open) and $R_L = R_y + R_x$. V_2 is the voltage measured when R_L is reduced to R_x (S = closed).

Applying the above theory with different resistor combinations, the EMHCS's Z_{out} was calculated to vary between 128 kΩ and 450 kΩ. The circuit's output impedance is approximately an order of magnitude greater than the expected load impedance, which implies that the current source should perform adequately [4].

4.3 EMHCS Implementation

The EMHCS has been implemented and the prototyped circuit can be seen in Figure 11.

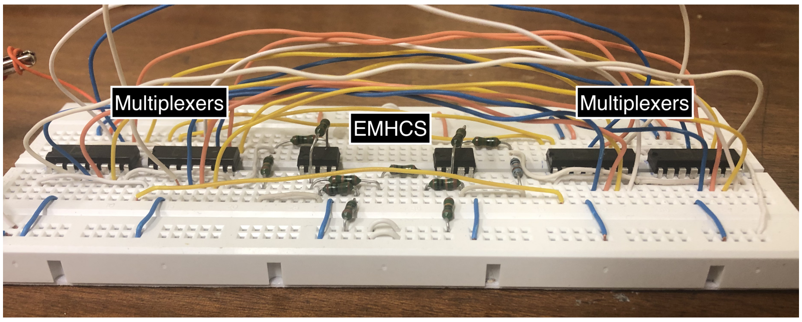
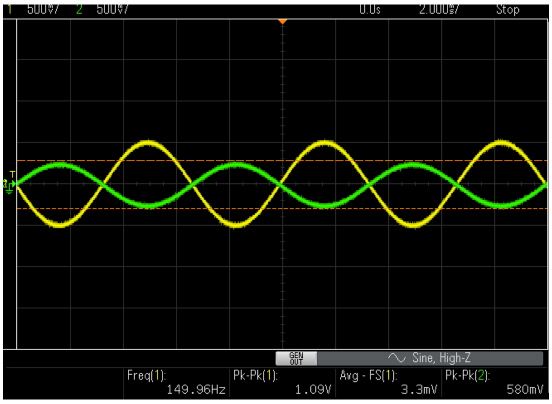


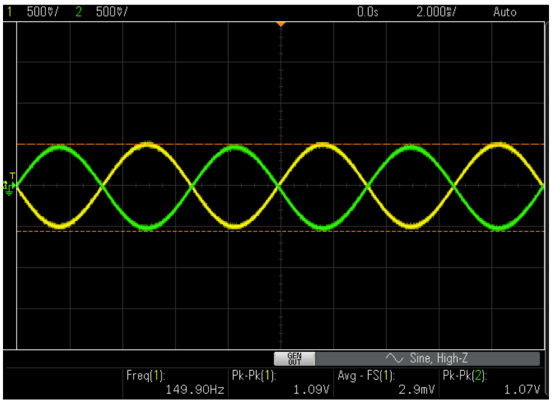
Figure 11: The EMHCS prototype circuit. Two LF351 FET amplifiers form the basis of the current sources. Two ADG508, 8:1 multiplexers are combined on either side of the current source to create two 16:1 current routing multiplexers. These connect the current source to the electrode array.

The CS has been constructed using FET based LF351 amplifiers which provide a high input impedance.

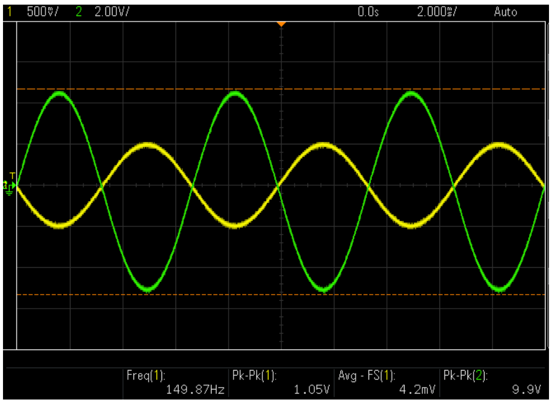
The EMHCS has been setup to mimic the simulation. The experimental setup involved loading the current source through the multiplexers with different load resistors, R_L of 5% tolerances. A signal generator is used to generate a 1 V_{pp} sinusoidal input signal at 150 Hz. The load current, I_L, is measured indirectly by measuring the voltage across R_L. The experimental results seen in Figure 12 verify the stability of the CS and prove its ability to maintain a constant current.



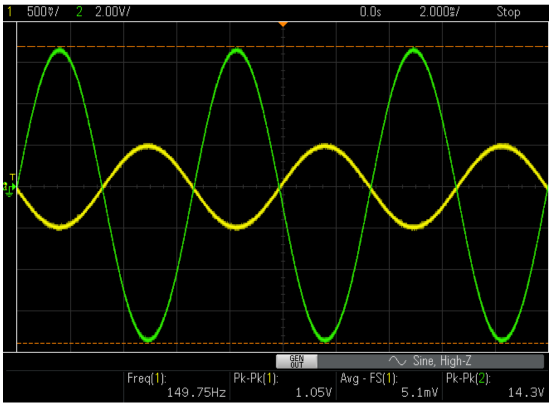
(a) $R_L = 560\,\Omega$. $I_L = 1.036\,mA$



(b) $R_L = 1\,k\Omega$. $I_L = 1.07\,mA$



(c) $R_L = 10\,k\Omega$. $I_L = 0.99\,mA$



(d) $R_L = 15\,k\Omega$. $I_L = 0.953\,mA$

Figure 12: Experimental results obtained by loading the EMHCS with different resistor values. The input signal (yellow) is a $1\,V_{pp}$ sinusoid at 150 Hz obtained from a signal generator and the output signal (green) is measured across R_L.

5 SIGNAL ACQUISITION

The differential voltage measured at the electrodes is created by the source current's interaction with the impedance of the conductive body. These voltage measurements are used to estimate the load impedances, and form the raw data used in the reconstruction process. Therefore the quality of these measurements directly relates to the quality of the reconstruction.

Typically in EIT systems, differential or instrumentation amplifiers are used to measure the voltages between two of the electrodes [1].

Since this system is integrated with an Arduino Mega, and the differential voltage is to be measured using the onboard ADC; a bipolar to unipolar converting circuit must be implemented.

This Section describes the voltage measurement circuit made up of the instrumentation amplifiers and the bipolar-polar signal converter.

5.1 Voltage Measurement Circuit

This system utilises 16 AD620 instrumentation amplifiers with a set gain of 2 to measure the differential voltages. The measurement electrodes are interleaved with the current source electrodes to allow for voltage readings between the conducting electrodes. The output of each amplifier is multiplexed such that a single level-shifting circuit is used.

A circuit diagram of the subsystem is shown in Figure 13.

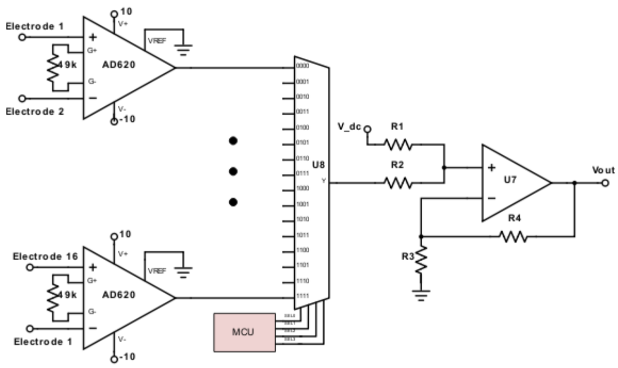


Figure 13: Circuit diagram of the voltage acquisition circuit.

5.2 Bipolar to Polar Signal Converter

The function of this circuit is simply to scale and shift the bipolar input signal to a condensed unipolar signal. A simple summing amplifier can be used to scale the input signal and apply a constant DC offset to map the input signal to an acceptable level. This circuit diagram is shown in Figure 13 [9].

The input current to the electrode array is assumed to be a 1 mA sinusoid with a variable load resistance $R_{load} \in (1; 10)\, k\Omega$ which implies an expected differential voltage in the range of $V_d = \pm 10\, V$

Since the Arduino's ADC can withstand a maximum of 5 V, the input signal must be mapped to the space $V_{ADC} \in (0; 5)\,V$.

We can therefore express the transfer function of the circuit as:

$$V_{ADC} = \frac{1}{4} V_d + 2.5 \tag{18}$$

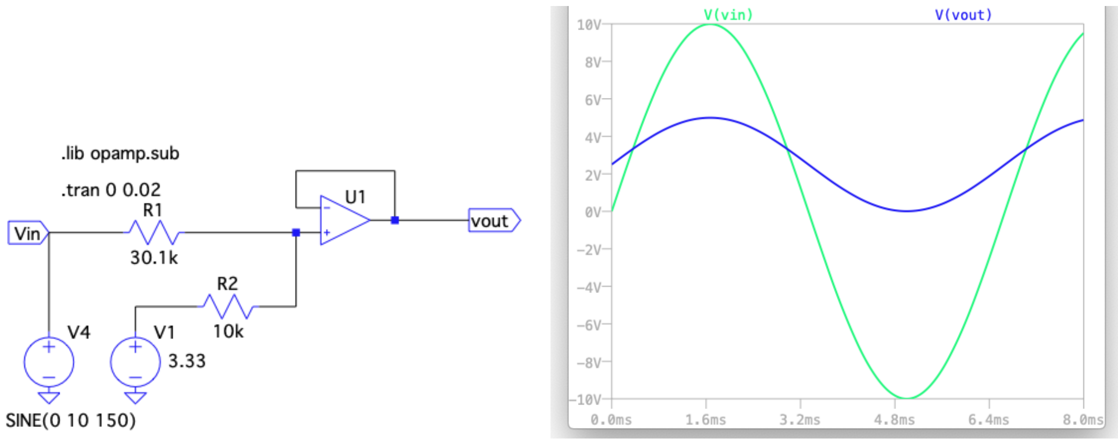
The original bipolar input signal can be determined by rearranging the equation to solve for V_d in software. The appropriate component values are determined by comparing Equation 18 to the summing amplifier's general equation [9].

$$V_{sum} = \left[V_1 \frac{R_2}{R_1 + R_2} + V_2 \frac{R_1}{R_1 + R_2} \right] \left[1 + \frac{R_4}{R_3} \right] \tag{19}$$

By open circuiting R_3 and short circuiting R_4, we can remove the non-inverting amplifier gain of the system, essentially resulting in a unity gain buffer. We let $V_1 = V_d$ which implies that $1/4 = R_2/R_1 + R_2$. Therefore, we choose $R_2 = 10\,k\Omega$ and $R_1 = 30\,k\Omega$. This then requires $V_2 = 3.3\,V$

A variable voltage regulator circuit is used to produce the constant offset voltage.

The circuit is simulated using an ideal operational amplifier and ideal voltage sources. The simulated circuit and output can be seen in Figure 14.



(a) Summing amplifier with a sinusoidal input signal and a constant DC voltage.

(b) Simulated input output waveform relationship. The input waveform has a swing of $\pm 10\,V$ and has no DC offset. The output signal has a swing of $\pm 5\,V$ and has a DC offset of $2.5\,V$.

Figure 14: Simulation of unipolar to bipolar conversion circuit in *LTSpice*.

6 SYSTEM INTEGRATION

The system has been integrated and tested. The final system configuration is depicted in Figure 15 below.

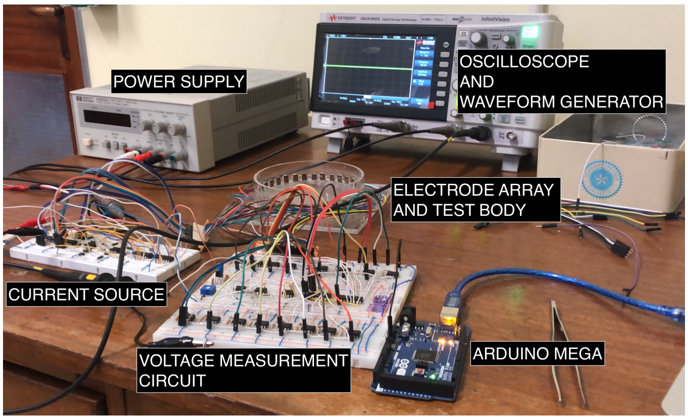


Figure 15: Integrated single source EIT system comprising of an enhanced mirrored Howland current source, current routing multiplexers, the electrode array and conductive body, 16 instrumentation amplifiers used to measure the differential electrode voltages, the level shifting circuit and finally the Arduino Mega.

The following Section describes the experimental procedure used to test the functionality of the system.

7 EXPERIMENTAL RESULTS

The integrated system's functionality is tested experimentally. This experiment asserts whether the voltage readings provide intelligible results.

The main objective of the experimental testing is to confirm whether the system's measurement subsystem produces intelligible results. This includes determining whether: appropriate gains for the instrumentation amplifiers have been set, and measurements are obtainable.

7.1 Rotating measurement electrodes

An Arduino script has been generated to test the system integration. The current source is kept constant. The Arduino then co-ordinates the measurement multiplexers to cycle through each of the differential voltages and sample the voltage measured by the ADC.

For each electrode position, the voltage reading is sampled 120 times and stored in an array. The resulting array is then broadcast serially to Matlab where the data is processed and visualised. The Arduino and Matlab scripts can be found in the Appendix A.

The Figures below represent three frames of data. In each Figure, the electrode position is changed which corresponds to a different pattern exhibited across each of the electrode pairs. The output voltage is highest near the current injecting electrodes and attenuates rapidly as the distance increases. Notice how the amplitude and phase of the output waveform varies with each electrode position.

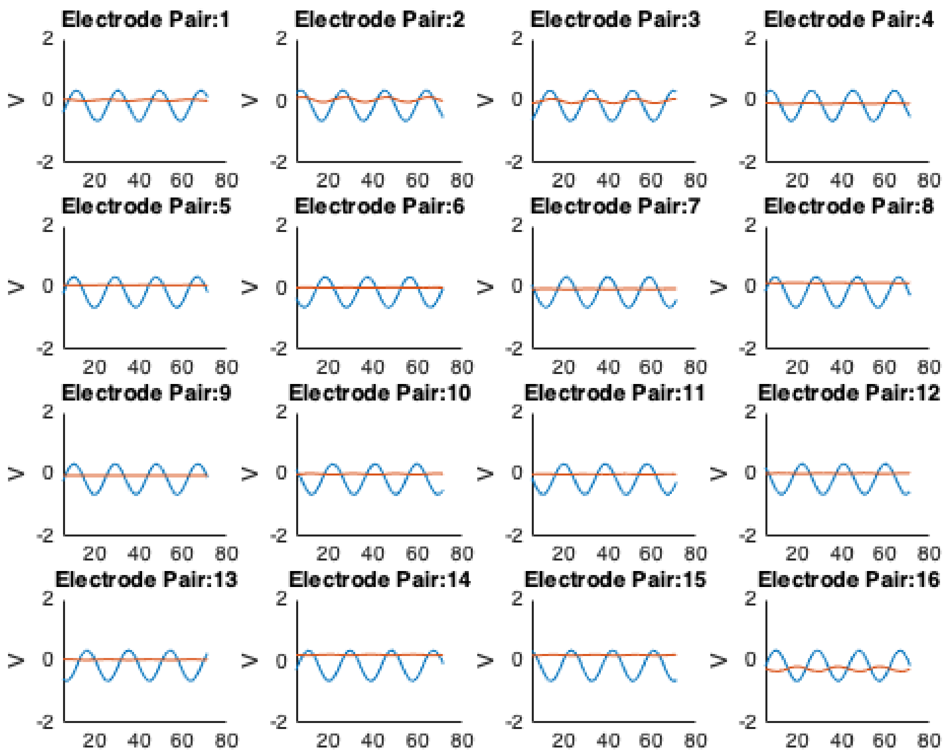


Figure 16: Current stimulating electrode at electrode position 1.

8 DISCUSSION

From the results obtained, it is clear that the system is partially operational. A sinusoidal current signal has been injected into the system and a corresponding voltage waveform has been observed and measured at the output.

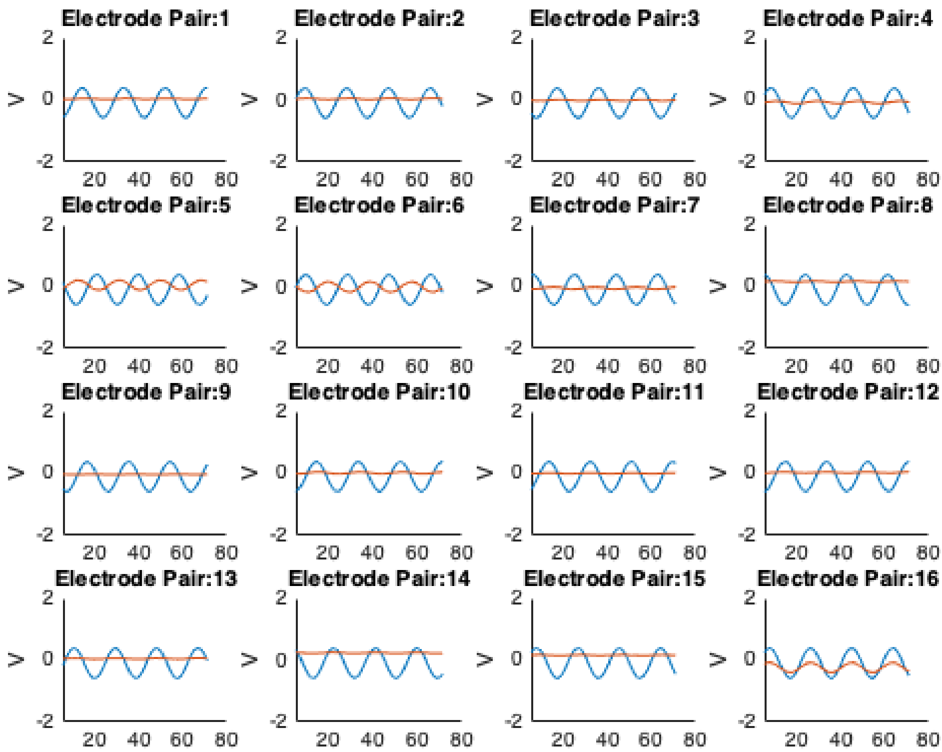


Figure 17: Current stimulating electrode at electrode position 5.

Although these results are promising, there are several limitations of its operation. Namely, the sampling speed of the Arduino. To capture a single frame of data (measuring 120 points at the output of each instrumentation amplifier) takes almost a minute to complete. The prospect of real-time monitoring is therefore not possible on the system as such.

Furthermore, the Arduino's ADC has a resolution of 10-bits, implying a precision of approximately 4.9 mV. This may not be sufficient to capture the minute differential voltages far from the current carrying electrodes. Moreover, the ADC only accepts unipolar signals in the range of 0-5 V which requires a conditioning circuit to format the bipolar output signal. This may introduce additional phase delays and imprecise gains that reduce the accuracy of the system.

9 CONCLUSIONS

A simple EIT system has been designed and built using using standard components. Methods concerning potential voltage signal generation from a microcontroller have been explored but ultimately a signal generator has proved to be most effective. An enhanced mirrored Howland Current source has been implemented and is effective at producing a consistent 1 mA load current. The output impedance of the EMHCS has been evaluated experimentally, and is suitable for the expected range of load impedances. A voltage gain of two has been applied to the instrumentation amplifiers on each electrode pair. Experimental observation has proven this gain to be sufficient. Finally, the system has been integrated and tested with partial success. The results indicate that the system can effectively sample both the input and output signals for low frequencies near 150 Hz. A scaling and phase shift of the outputs signal indicates some ability to detect impedance. However, the system may be improved upon significantly by replacing the Arduino with a more appropriate sampling system and a bipolar ADC.

REFERENCES

[1] D. Holder, Ed., *Electrical Impedance Tomography Methods, History and Applications*. Institute of Physics Publishing, 2005.

[2] Y. Zhang and C. Harrison, "Tomo: Wearable, low-cost, electrical impedance tomography for hand gesture recognition," *Carnegie Mellon University*, 2015.

[3] Analog electronics - ad5933. [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/AD5933.pdf

[4] M. Khalighi, B. Vosoughi-Vahdat, M. Mortazavi, and M. M, "Design and implementation of precise hardware for electrical impedance tomography (eit)," *IJST, Transaction of Electrical Engineering*, vol. 38, no. E1, pp. 1–20, 2014.

[5] B. Carter, *Op Amps for everyone*. Newnes, 2013.

[6] Texas instruments - lm555. [Online]. Available: http://www.ti.com/lit/gpn/lm555

[7] P. Bertemes-Filho, A. Felipe, and V. Vincenze, "High accurate howland current source: Output constraints analysis," *Circuit and Systems*, vol. 4, no. 7, pp. 451–458, 2013.

[8] V. Sirtoli, K. Morcelles, and V. Vincenze, "Design od current sources for load common mode optimization," *Journal of Electrical Bioimpedance*, vol. 9, pp. 59–71, 2018.

[9] P. Horowitz and W. Hill, *The art of electronics*, 3rd ed. Cambridge Univ. Press, 2015.

APPENDIX

A SOFTWARE IMPLEMENTATION

The Arduino scripts used in the experimental procedures is documented in this Section.

A.1 Voltage Measurement Electrode Rotation

The following code is used to run the electrode measurement experiment of Section 7.1 on an Arduino.

```
/*
 * This script is used to circulate through the voltage readings.
 * All votlages are measured using an AD620 instrumentation amplifier with unity gain.
 * The bipolar signal is then converted to a unipolar signal by a summing amplifier and
     constant DC voltage.
 *
 * Current injection electrodes are kept constant.
 *
 * No ADC readings are taken on this run. This script's purpose is to control the signal
 * gating of the differnetial voltages measured on the conductive body.
 *
```

```
 * The scipt will alternately enable and disable the respective MUXs to route signals
     from each of the
 * diff voltages at intervals of 2 seconds. The output is measured on the oscilloscope
     before reading the voltage using the ADC to ensure
 * no hazardous voltages are present. Amen.
 *
 * NB 2 8:1 muxs are connected to create a 16:1. The MUX outputs are tied together so
     both MUXs MUST be disabled
 * before changing channels.
 * MUX1: 0-7
 * MUX2: 8-15
 *
 * S2 |  S1 | S0 | Channel
 * ---------------------------
 * 0  |  0  |  0 |  0
 * 0  |  0  |  1 |  1 etc
 *
 * MUXEN = HIGH to enable
 *
 * Matteo Kalogirou
 */

// Need: 2 MUX EN, 3 select lines. (The mux select lines are tied together.)

#define S0 24
#define S1 26
#define S2 28

#define MUX1 32
#define MUX2 34
#define MUX3 36
#define MUX4 38

float v_adc = 0;               //Analog read from A0

int channel = 0;             //ADC channel to read from


void setup() {
  // Initialise the pins
  pinMode(MUX1, OUTPUT);
  digitalWrite(MUX1, LOW);
  pinMode(MUX2, OUTPUT);
  digitalWrite(MUX2, LOW);
  pinMode(S0, OUTPUT);
  pinMode(S1, OUTPUT);
  pinMode(S2, OUTPUT);

  Serial.begin(9600);        //Open a serial port

}

void loop() {

  //Select channel
  Serial.print("Channel: "); Serial.println(channel);
  selectChannel(channel);
  //Enable MUX
  if(channel < 8){
    digitalWrite(MUX1, HIGH);
  }else if(channel >= 8 ){
```

```
    digitalWrite(MUX2, HIGH);
  }

  //Read the voltages and store them into the array

  //Increment channel
  ++channel;
  if(channel >15)
    channel = 0;

  //Read voltage
//   recordADC();          //voltages readings stored in v[]
//   maxV = getMax(v, 10);
//   Serial.print("Maximum Voltage = "); Serial.println(maxV);
//   minV = getMin(v, 10);
//   Serial.print("Minimum Voltage = "); Serial.println(minV);
//   Serial.print("Range = "); Serial.println(maxV-minV); Serial.println();

    ADCtoSerial();

  //Delay
  delay(1000);

  //Disable MUXs
   digitalWrite(MUX_1, LOW);
   digitalWrite(MUX_2, LOW);

}

/*
 * Read the voltage at the output port (Period of one waveform = 1/150 = 6.67ms)
 * vout = 1/4 vin -2.5
 * Store the results of one period in an array
 */
void recordADC()
{
    //Store 10 readings into v_int
    for(int i =0; i< 10; i++)
        v[i] =   analogRead(v_adc);

     for(int i=0; i<10; i++)
        v[i] = (v[i]/1023)*5 - 2.5;      //Convert to an actual voltage
}

void ADCtoSerial()
{
    for(int i =0; i< 1000; i++)
    {
      m = analogRead(v_adc);
      m = (m/1023) * 5;
       Serial.println(m);
    }
}


//Return the maximum element from an array
float getMax(float a[], int size)
{
   float maximum = 0;
   for (int i = 0; i< size; i++)
   {
```

```
      if(a[i] > maximum)
          maximum = a[i];
   }
   return maximum;
}

float getMin(float a[], int size)
{
   float minimum = 9999;
   for (int i = 0; i< size; i++)
   {
      if(a[i] < minimum)
          minimum = a[i] ;
   }
   return minimum;
}


//Change the S0-2 Output pins to select the channel;
void selectChannel(const int channel){
    int c = channel%8;
    switch(c)
    {
      case 0:
            digitalWrite(S0,LOW);
            digitalWrite(S1,LOW);
            digitalWrite(S2,LOW);
            break;
      case 1:
            digitalWrite(S0,HIGH);
            digitalWrite(S1,LOW);
            digitalWrite(S2,LOW);
            break;
      case 2:
            digitalWrite(S0,LOW);
            digitalWrite(S1,HIGH);
            digitalWrite(S2,LOW);
            break;
      case 3:
            digitalWrite(S0,HIGH);
            digitalWrite(S1,HIGH);
            digitalWrite(S2,LOW);
            break;
      case 4:
            digitalWrite(S0,LOW);
            digitalWrite(S1,LOW);
            digitalWrite(S2,HIGH);
            break;
      case 5:
            digitalWrite(S0,HIGH);
            digitalWrite(S1,LOW);
            digitalWrite(S2,HIGH);
            break;
      case 6:
            digitalWrite(S0,LOW);
            digitalWrite(S1,HIGH);
            digitalWrite(S2,HIGH);
            break;
      case 7:
            digitalWrite(S0,HIGH);
            digitalWrite(S1,HIGH);
```

```
            digitalWrite(S2,HIGH);
            break;
    }
}
```

This data is then serially transmitted to Matlab which is used to process and store the data. The script is seen below.

```
function [ s ] = setupSerial( comPort )
% Manually sets the serial object to interface with the arduino

s = serial(comPort);
set(s, 'DataBits', 8);
set(s, 'StopBits', 1);
set(s, 'BaudRate', 9600);
set(s, 'Parity', 'none');

s.InputBufferSize = 10000;
s.ReadAsyncMode = 'continuous';

fopen(s);            %connect the serial object to the arduino


end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%   EEE4022
%   EIT: hardware
%   This script will be used to open and manage the serial link between
%   the arduino and the PC.
%
%   2019
%   Matteo Kalogirou
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%---------Cleanup the junk
delete(instrfind({'Port'}, {'/dev/tty.usbmodem14101'}));
delete(instrfind({'Port'}, {'/dev/tty.usbmodem14201'}));
clear arduino;
clear comport;
clear all;
close all;
clc;
%%

%---------Setup the Serial Link
comport = '/dev/tty.usbmodem14101';
[ arduino ] = setupSerial(comport);


%% ---------Variables
bytesPerSample  = 6;
numberSamples = 200;
output_v_meas = zeros(numberSamples, 16);
input_v_meas = zeros(numberSamples,16);

%% -------- Read alternating voltages
```

```
% Send the electrode pair to read from, wait until there are some bytes,
% store the results. Incrememt the electrode pair and repeat

for electrodePair = 0:15

    flushinput(arduino);                    %Clear the serial buffer
    fprintf(arduino, electrodePair);        %Select electrode pair to read

    while(arduino.BytesAvailable < numberSamples*bytesPerSample*2)
        %                   WAIT
    end

    %READ from the buffer
    receivedValues = arduino.BytesAvailable/(bytesPerSample*2); %2 channels (input and
        output)
%
    if(receivedValues ~= numberSamples)
        display('Error: Incorrect number of samples read.');
    end

    for i = 1:receivedValues
        input_v_meas(i, electrodePair+1) = fscanf(arduino, '%f', bytesPerSample);
        output_v_meas(i, electrodePair+1) = fscanf(arduino, '%f', bytesPerSample);
    end

end

display('WE DONE');

%% ------- View the JUNK

figure();

%Shift the output signal
shift = 10;
shifted_v_out = [zeros(shift, 16); output_v_meas(1:end-shift,:)];

for i = 1:16
    subplot(4,4,i);
    hold on;
%     plot(output_v_meas(:,i));
    plot(input_v_meas(:,i));
    plot(shifted_v_out(:,i));
    hold off;
    ylabel('Voltage');
    title(strcat('Electrode Pair:', num2str(i)));
end
legend('Input', 'Output', 'Location', 'NorthEastOutside');
hold off;
```