Development of a low cost Electroencephalogram patient simulator

Brandon C. Lancaster, George Mason University, Fairfax, VA

Abstract— **Conducting human subjects research has some challenges for bioengineers such as lack of repeatability of bio-signals, IRB approval and recruitment. The COVID-19 pandemic has made it even more difficult for researchers to access research subjects. One biosignal that is economical, portable and quick to collect data from is Electroencephalography (EEG). Here we present a solution to researchers' reduction of access to in vivo EEG signals. The solution is in the form of a patient EEG simulator that is controlled by an ATmega328P, connected to an analog to digital converter and other analog circuitry. Other commercial devices exist but this device was also constructed for educational purposes. The output signal can be adjusted to better match a real bio-potential from a human. Neurophysiology recordings can be loaded onto the device for use. Open source data is readily available for download to add to the database on the device. This simulator has other potential applications such as unit testing for devices under development and more convenient and rapid prototype iteration. Optimization of the device is also discussed.**

I. Introduction

Many studies conducted by bioengineers involve the collection and processing of biosignals. These signals are any metric that can be continuously monitored and sampled from the body. For example, one study uses functional Magnetic Resonance Imaging (fMRI) biosignals to reconstruct what the brain is seeing [1]. Still another study uses four scan lines of ultrasound to determine the volitional motor intent of a prosthesis user [2]. There are also many efforts to build devices that can collect multiple types of biosignals at the same time, such as devices discussed in [3]. Some are wearable and can measure a wide array of items including photoplethysmography, electrocardiography and galvanic-skin response. Most are wearable while others are designed to fit over a mattress or over an automobile seat to monitor sleep and driving, respectively. Thus, access to a patient's biosignals is often very important for research efforts and the development of new medical devices.

To enhance ease of access to different biosignals, many types of patient simulators have been constructed. We will discuss several examples to make clear different use cases. Perhaps one of the most well known is the Human Patient Simulator (Medical Educational Technologies, Inc.). This device is mainly used to train nurses and other physiology students. It can reproduce many features a real patient could such as pulse, pupil dilation and lung sounds. In addition, the accompanying faux patient monitoring screen can produce many of the same indicators that would be seen on a real patient monitoring machine such as blood pressure, an electrocardiogram plot, blood gasses, respiratory rate and more [4]. By contrast, the next patient simulation device discussed has a more narrow use. It simulates pressure changes measured by a Blood Pressure Meter (BPM) machines. Specifically, the pressure signals used oscillometric BPMs. The authors suggest that it can enable more reliable testing of new devices [5]. Still another patient simulator that has been developed is one that mimics specific aspects of a human upper limb. The creators claim it will help physical therapists improve delivery of therapy for muscle spasticity, particularly of stroke patients [6]. Hence, we see that patient simulators as well as biosignal recordings can have a wide range of uses. Further, we see that patient simulators that incorporate biosignals can be useful.

Electroencephalography (EEG) is another example of a widely used biosignal. It is a device that can measure macro electrical properties of the brain. To detect a signal, there must be many neurons firing action potentials at the same time. The smallest neural event is thought to be ~100,000 synchronous pyramidal cells arranged in a similar direction [7]. Measured amplitudes from one review ranged up to 200 μV [8].

EEG has many uses [9, Ch. 3]. For example, it has been discovered that the alpha rhythm has a role in tuning occipito-parietal areas and is associated with regulating attention [10]. It has also been shown that individuals with less left frontal activity (in contrast to right) can help identify patients with major depressive disorder. However, this interhemisphere asymmetry has been observed in other patient groups as well and more research is needed [11]. EEG is also used for neurofeedback for the treatment of ADHD [12]. It would be amiss not to mention the use of EEG in sleep studies as well. Certain features of EEG recordings are associated with different phases of sleep, which is well documented in several different studies as outlined in [13]. In addition, EEG has uses in sports fitness, education, prediction of diseases such as Epilepsy and has even been researched as a form of bio-metric identification [14].

Given the widespread uses of EEG, patient simulators which are capable of mimicking these signals can have many uses. In this manuscript, we will discuss the design of such a patient simulator. Given that there already exist patient simulators for EEG [15], this device was created as part of the course requirements at George Mason University with the primary goal of teaching its creators about EEG and electronics.

II. Component Selection

The device consists of an Arduino Uno development board, an SD card shield, an external momentary button and a digital to analog converter (DAC).

The Arduino Uno was selected due to its ubiquity and the ability to remove the microcontroller from the development board to use in a stand alone configuration. The microcontroller is easy to remove as it is in a dual in line

B. C. Lancaster is a graduate student in the Bioengineering Department at George Mason University, Fairfax, VA 22030 USA (e-mail: blancas2@gmu.edu).

package (DIP). The part number of the microcontroller is ATmega328P. The only external components that are needed for the ATmega328P to function independently from the development board is the 16 MHz oscillator, two supporting capacitors and a resistor to connect the reset pin to Vcc so it is not floating. The microcontroller has hardware SPI communication to interface with the SD card and also hardware I2C communication to interface with the digital to analog converter.

A SD card shield was selected to add flexibility to the design. Storing data on an SD card allows the user to easily place new EEG recordings on the device without having to reprogram the ATmega328P. In addition, the microcontroller only has a limited amount of memory which would likely not be suitable for many EEG recordings. It has 2 KB SRAM, 32 KB of flash memory (which is not writable by the program, only the programmer) and 1 KB of EEPROM. The EEPROM is writable by the program and would be the most logical place to store such data but is not very large. Assuming a 12 bit value needs to be stored for each sample, this would only amount to 682 EEG samples [(1024×8)÷12= 682.67]. In contrast, even a 4 GB SD card is about 4 million times larger.

The digital to analog converter comes in a 6 lead small outline transistor (SOT-23) package. It is a 12 bit resolution, rail to rail. It was selected because the package has a hardware pin that allows the I2C address of the device to be changed depending on if the pin is set to a high or a low voltage level. This is important because it allows for multi channel DAC output of the device. This is possible by keeping all the DACs address pins low except for the one that needs to be addressed at a certain time. The DAC to be addressed has its I2C address pin set to high. As such, it will be the only device with a specific address on the bus. Once it is set, its I2C address pin can be lowered and the next DAC's pin can be brought high. This type of configuration is limited only by the available pins on the master microcontroller. With enough channels, in theory this type of addressing could cause slower refresh rates but this should not be an issue. More research and testing would be needed to determine the maximum number of allowable channels to maintain a minimum desired refresh rate. A prototype of the system can be seen in figure 1.

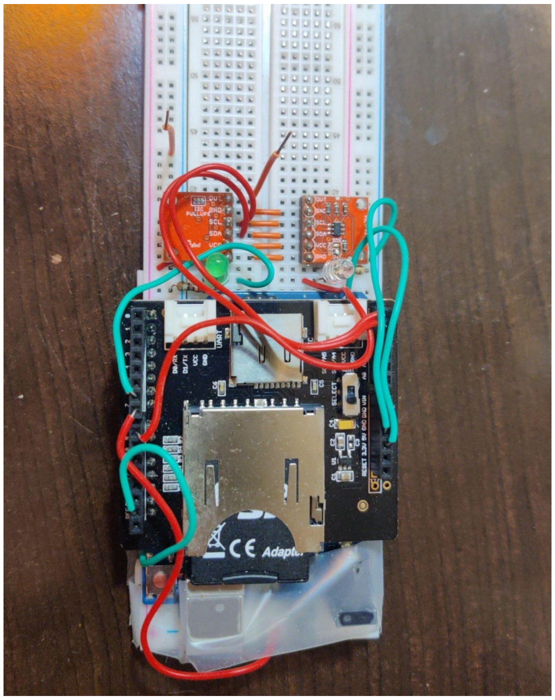

Figure 1: Photograph of the system

III. SYSTEM DESIGN & FEATURES

The system was coded in Arduino IDE V 1.8.16. It is capable of optionally "writing" the EEG voltage value to pulse width modulation (PWM) pin in addition to the DAC. This is controlled via a boolean flag set at programming time. However, the PWM on the ATmega328p has only 8 bit resolution so this is likely not suitable for direct output as a signal. It is more intended to drive a status indicator such as an LED. The program currently allows a gain value to be set dynamically by sending an integer over the microcontroller's serial port.

The simulator reads a previously saved comma separated value file from an SD card. The data has no header, only a time value (read into the program as an int datatype) and an EEG voltage value (read into the program as a float datatype). It was decided that the system would skip samples if it detects that it is too far behind the time stamp in the file. This way, if the system is unable to attain the sample rate of the file, at least the envelope of the signal is preserved as well as the temporal aspect. The system dynamically updates how many samples to skip and converges on an appropriate value for the given data. The program went through several iterations but the current version's execution is as follows:

- Initialize the DAC object and set the voltage to zero.
- Initialize the serial port for sending status messages back to a host computer (if present, not required for operation)
- Turn on a pull up resistor within the microcontroller for button pin
- Initialize output pins as low impedance output instead of high impedance input pins

- Initialize the SD card, halt execution if failure
- Waits for button press to begin execution
- After button is pressed, opens SD card file
- Within a loop while the file still has data:
    - Check serial terminal for a new gain value
    - Read in the current line of the file and parse it
    - Compare the time stamp of the EEG data and output immediately if behind, if ahead wait until proper time and then "write"
    - If the system detects it has been behind for two samples or more: skip a certain amount of bytes in the file and increase the amount of bytes to skip [note 1]
    - If it was ahead of the time stamp in the file the last time, decrease the amount of bytes skipped
    - After skipping, find the next valid line in the file
- Close the file and send statistics about samples to serial port

Note 1- the system currently does not skip any part of the file unless it has been behind for a minimum number of samples. This prevents this logic from running and slowing the system down if it has been successfully able to keep up with the sample rate in the file.

IV. System Speed Tests & Optimization

To test system performance, a file was loaded that had a 1925.926 Hz sample rate and was 27.47198 seconds in length. The sampling rate of commercial EEG recorders range from 128 Hz to 16 kHz [14]. The file was selected as a typical recording file of EEG systems. It was well above the minimum sample rate but not close to the maximum possible either. A future version of the prototype should be tested with a file recorded at 16 kHz. The portion of the code which checks skipped samples was disabled for this test and the code was allowed to just write samples as fast as possible.

The code took an average of 138 seconds to execute over 4 trials. This is approximately 5 times too slow. Dividing the original sample rate by this slow down, we get a rate of ~383 Hz. Clearly some improvements were required.

The goal of the first optimization was to remove all non-essential parts of the program such as status indicator LEDs and messages sent over a serial port. The only thing done in the loop was to read the data from the SD card and write that value to the DAC. This decreased the time to 85.7 seconds, which is still ~3.1 times too slow.

The next optimization step executed was to eliminate code that wrote the values to the DAC. This was done to ascertain if the SD card reads or the DAC writes were taking up the most time. The program took an average of 77.9 seconds to execute over 4 trials-- an improvement yet still ~2.8 times too slow.

Next, the portion of the program that read the SD card was optimized. Research indicated that the functions used to parse the integers and floats from the SD card file consume a significant portion of time. To test this, these functions were replaced with functions that simply read a certain number of bytes from the file into a buffer. With a one byte buffer, execution time was finally reduced to an acceptable level – 26.0 seconds. When the buffer size was increased to 100 bytes, the execution time was reduced to 3.8 seconds. This is ~34 times faster than the fully featured code. It is ~6.9 times faster than the code that has only a 1 byte buffer. Thus, we can conclude that parsing numbers from the file takes the bulk of the time in the loop. Also interesting to note is that speed increase does not scale linearly with buffer size because increasing the buffer size from 1 byte to 100 would have yielded a 100 times speed increase instead of the 6.9 times increase that was seen experimentally.

V. Future Work

The most important feature to continue this project is to implement multichannel output. Commercial systems have between 2 and 256 channels [14]. The second feature would be to have the system parse the data before execution and save the data to a binary data file so that the 34 times speed increase outlined above could be realized.

Acknowledgment

B. C. L. thanks Nathalia Peixoto for help starting the project and encouragement to write this paper.

References

[1] Y. Miyawaki *et al.*, "Visual image reconstruction from human brain activity: A modular decoding approach," *J. Phys.: Conf. Ser.*, vol. 197, p. 012021, Dec. 2009, doi: 10/d2pfnx.

[2] N. Akhlaghi *et al.*, "Sparsity Analysis of a Sonomyographic Muscle–Computer Interface," *IEEE Transactions on Biomedical Engineering*, vol. 67, no. 3, pp. 688–696, Mar. 2020, doi: 10.1109/TBME.2019.2919488.

[3] Q. Lin *et al.*, "Wearable Multiple Modality Bio-Signal Recording and Processing on Chip: A Review," *IEEE Sensors Journal*, vol. 21, no. 2, pp. 1108–1123, Jan. 2021, doi: 10.1109/JSEN.2020.3016115.

[4] W. M. Nehring, W. E. Ellis, and F. R. Lashley, "Human Patient Simulators in Nursing Education: An Overview," *Simulation & Gaming*, vol. 32, no. 2, pp. 194–204, Jun. 2001, doi: 10.1177/104687810103200207.

[5] G. Geršak *et al.*, "Physiology-based patient simulator for blood pressure meter testing," *Measurement: Sensors*, vol. 18, p. 100260, Dec. 2021, doi: 10.1016/j.measen.2021.100260.

[6] N. A. Cz, T. Komeda, and C. Y. Low, "Design of Upper Limb Patient Simulator," *Procedia Engineering*, vol. 41, pp. 1374–1378, Jan. 2012, doi: 10.1016/j.proeng.2012.07.324.

[7] M. X. Cohen, “Where Does EEG Come From and What Does It Mean?,” *Trends Neurosci.*, vol. 40, no. 4, pp. 208–218, Apr. 2017, doi: 10.1016/j.tins.2017.02.004.

[8] G. Buzsaki, C. A. Anastassiou, and C. Koch, “The origin of extracellular fields and currents - EEG, ECoG, LFP and spikes,” *Nat. Rev. Neurosci.*, vol. 13, no. 6, pp. 407–420, Jun. 2012, doi: 10.1038/nrn3241.

[9] B. He, Ed., *Neural engineering*, Third edition. Cham: Springer, 2020. doi: 10.1007/978-3-030-43395-6.

[10] G. Thut and C. Miniussi, “New insights into rhythmic brain activity from TMS-EEG studies,” *TRENDS COGN. SCI.*, vol. 13, no. 4, pp. 182–189, Apr. 2009, doi: 10.1016/j.tics.2009.01.004.

[11] A. A. Fingelkurts and A. A. Fingelkurts, “Altered Structure of Dynamic Electroencephalogram Oscillatory Pattern in Major Depression,” *Biol. Psychiatry*, vol. 77, no. 12, pp. 1050–1060, Jun. 2015, doi: 10.1016/j.biopsych.2014.12.011.

[12] M. E. Toplak, L. Connors, J. Shuster, B. Knezevic, and S. Parks, “Review of cognitive, cognitive-behavioral, and neural-based interventions for Attention-Deficit/Hyperactivity Disorder (ADHD),” *Clinical Psychology Review*, vol. 28, no. 5, pp. 801–823, Jun. 2008, doi: 10.1016/j.cpr.2007.10.008.

[13] S. Astori, R. D. Wimmer, and A. Lüthi, “Manipulating sleep spindles – expanding views on sleep, memory, and disease,” *Trends in Neurosciences*, vol. 36, no. 12, pp. 738–748, Dec. 2013, doi: 10.1016/j.tins.2013.10.001.

[14] M. Soufineyestani, D. Dowling, and A. Khan, “Electroencephalography (EEG) Technology Applications and Available Devices,” *Applied Sciences*, vol. 10, no. 21, Art. no. 21, Jan. 2020, doi: 10.3390/app10217453.

[15] “Biomedical Test Equipment - Netech Corporation.” https://www.netechbiomedical.com/ (accessed May 02, 2022).