DATA ACQUISITION AND THE BPM SYSTEM

Abstract

This paper discusses the concept of data acquisition and the procedure involved in designing a data acquisition system to acquire a pulse data with a heart beat sensor. The paper covers the background on designing a prototype to obtain, process, and transmit the data to display the pulse data on to the PC. A thorough explanation and testing of the proposed data acquisition system and software is given based on the project requirements.

1. Introduction

In a world where technology is continuously growing and changing the world and society itself, engineers are always finding ways for innovation and improvement. Data acquisition is the concept of processing an electrical or physical phenomenon like current, voltage, temperature, pressure or sound and converting the results into digital values that can be exploited by a PC. An example of a data acquisition system is music recording. In the 2000s, analog-to-digital convertors were essential to recording music and the reproduction of audio [1]. Sound is recorded through a microphone that converts the analog signal into a digital signal that the computer can understand to produce audio in a digital format such as a compact disk.

Data acquisition has become a revolution in biomedical instrumentation as well. Biomedical measurements require some sort of data acquisition system in order to take measurements from a subject and provide an assessment in order develop or deliver information that can be utilized. FitBit technology is another example of a data acquisition system in the form of a wearable device. FitBit technology uses a 3-axis accelerometer that acquires data in the form of movement or vibrations. With the accelerometer data, it can track the number of steps taken, distance walked, calories burned and heart rate with a tuned algorithm [2].

A data acquisition system follows a simple structure that begins with a transducer which is used to sense a signal in the form of temperature, pressure, light, weight, airflow or humidity. The transducer then converts the signal to an electrical signal that is suitable for the data acquisition system. Once it is passed through a signal conditioning circuit, the analog signal is then converted to a digital signal that is finally processed to the computer as a digital value.

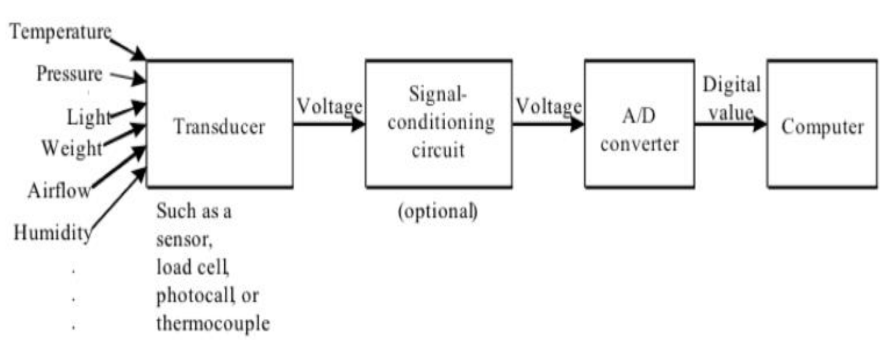

Figure 1: Structure of a Data Acquisition System

This paper will outline the design process of a data acquisition system that samples an individual's heartbeat and converts this analog signal into a digital signal using the Esduino Extreme microcontroller (ESDX). With the ESDX we are configure the data acquisition system using the given project requirements to perform three tasks which include displaying the BPM of a heartbeat using LEDs (BCD), serially transmitting the waveform data to a PC and graphically displaying the data with BPM.

2. System Overview

This section of the paper will outline the overall system and the various components implemented in this project. The system is configured based off the individual's respective student number. The data acquisition system starts and stops with a physical momentary push button. The input of the data acquisition system is the heartbeat sensor or function generator which is used as the transducer component. The ESDX microcontroller is programmed to take the input of the analog signal and convert the signal to a digital conversion that is then serially communicated to the computer using MATLAB to display the BPM signal. The ESDX microcontroller is also programmed to convert the BPM to LEDs using a binary-coded decimal representation.

Project Requirements – Student Number: 001415745	
ADC Channel	AN5
ADC Resolution	12-bit resolution
Bus Speed (E-Clock)	14 MHz

Table 1: Data System Acquisition Requirements

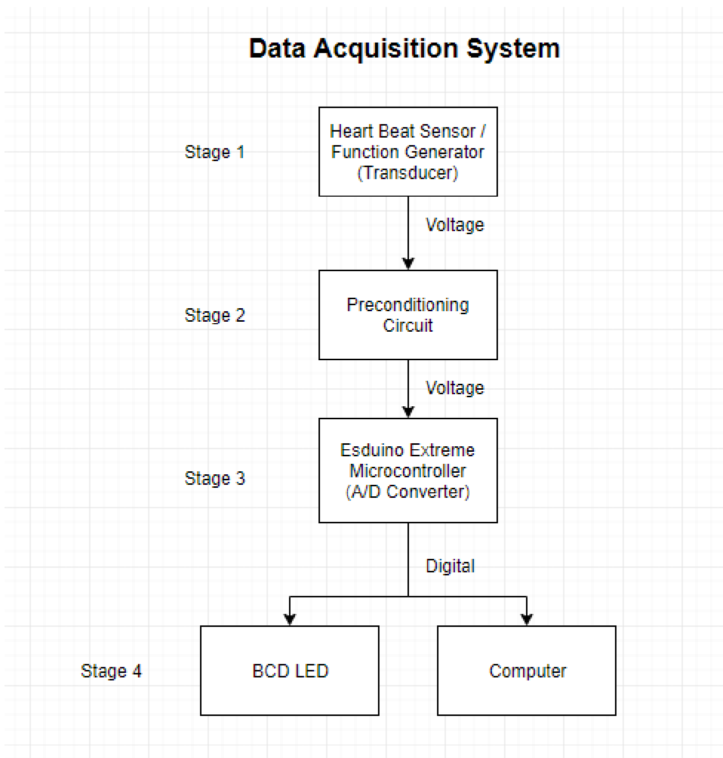

Figure 2: Flowchart of Various Stages in the Data Acquisition System

Final Pin Assignment

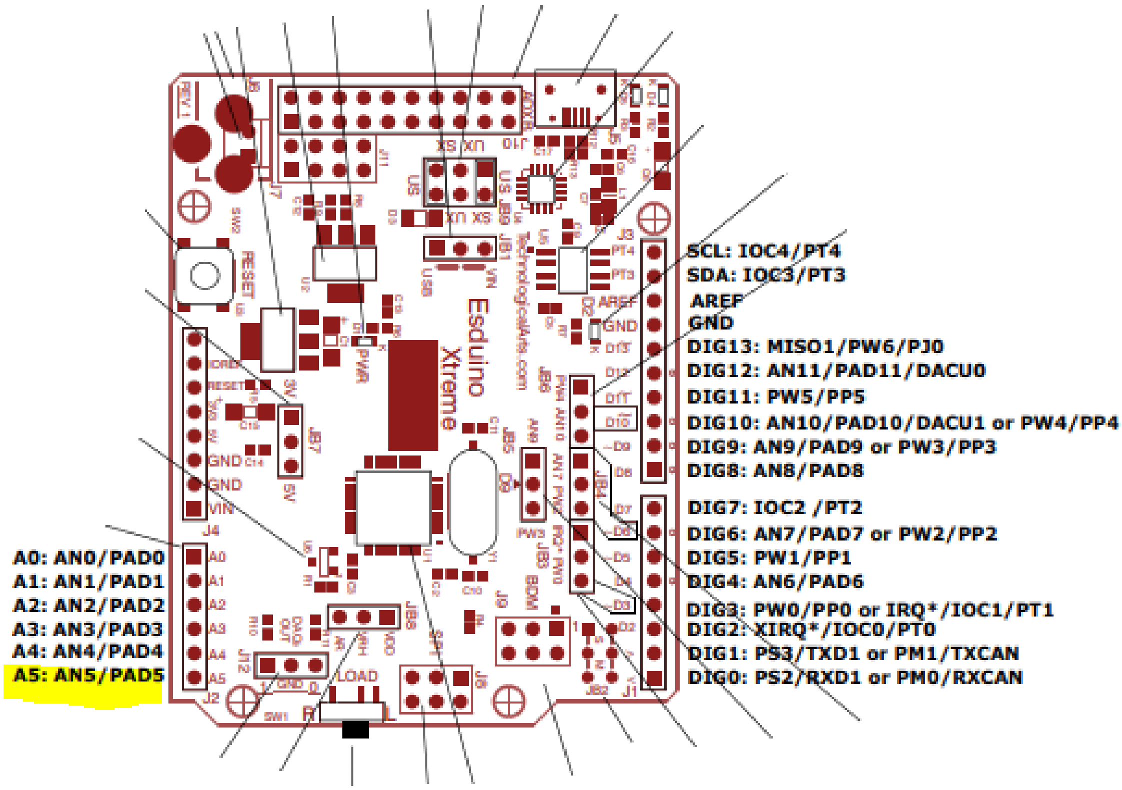

Figure 3: Esduino Extreme Circuit Board

Esduino Extreme PIN Layout

- A[0:3]: Port AD [0:3] = BCD Lights (LED 1-4)
- A[4]: Not Used
- A[5]: AN5 = ADC Channel (Analog signal from Function Generator / Heart Beat Sensor)
- DIG[0]: Port M [0] = BCD Lights (LED 9)
- DIG[1]: Not Used
- DIG[2]: Port T [0] = Button Interrupt
- DIG[3,5,6,9]: Port P[0:3] BCD Lights (LED 5-8)
- DIG[1,4,7,8,10,11,12]: Not Used
- DIG[13]: PJ0 = Positive terminal of LED
- GND: Negative terminal of LED
- AREF: Not Used
- SDA: Not Used
- SCL: Not Used

Signal Properties

The input provided to the Esduino Extreme (ESDX) microcontroller can vary between a pulse generated from the heartbeat sensor or an equivalent sinusoid generated from a function generator with 60 to 120 BPM. The generated analog signal is received by the microcontroller through port AN5. Furthermore, the signal generated can vary between 0V and 5V. There are multiple signals that can be passed through such as a square, triangle or sine wave with frequency dependent on 60 to 120 BPM. Frequency can be calculated with Hz = BPM / 60. The frequency range of the signal generated can be between 1 Hz to 2 Hz.

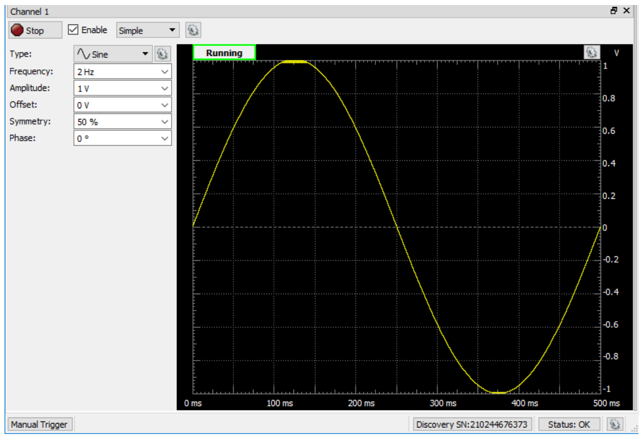

Figure 4: Input Analog Sine Wave with 1 V Amplitude and 2 Hz Frequency

Transducer

The transducer used in the following data acquisition system is the heart beat sensor. The heart beat sensor uses a reflected light in the finger tip or wrist to measure changes in light absorption from the skin related to blood perfusion. As the green light measures the changes, it sends an analog signal to port AN5. In the case when the heart beat sensor does not work or senses an undistinguishable pulse, a function generator is used as a substitute transducer to send a sine wave that is around 60 to 120 BPM.

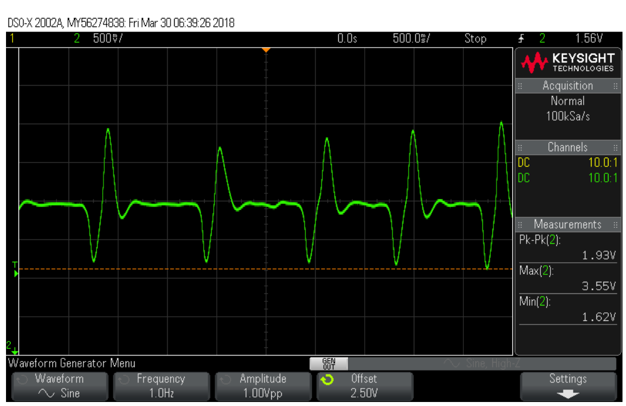

Figure 5: Oscilloscope reading of heart beat sensor pulse

Precondition / Amplification / Buffer

Preconditioning is the stage of the data acquisition system that transforms the transducer signal to be buffered, filtered, level-shifted or amplified. Most data acquisition systems require a preconditioning circuit in order to effectively transmit the signal for analog-to-digital conversion without distortion or unwanted frequencies. An op-amp may be used as the preconditioning circuit to amplify small signals to produce a sufficient analog signal to input to the ADC Channel. The heart beat sensor / function generator is the selected transducer, so for our preconditioning circuit it is optional for this system as configuring the signal can be done on the transducer itself.

ADC

Analog-to-Digital Conversion are essential in data acquisition systems as there are many different types of analog signals that are measured such as temperature, humidity, pressure, mass, light and speed. Analog-to-Digital Conversion makes it possible to consider various analog signals and transmit them to computer by digitalizing the signal. In the data acquisition system designed, the ADC channel being utilized is port AN5 and the ADC resolution is 12 bits.

BPM LED Display (BCD)

The data acquisition system is responsible of taking an analog signal and converting the signal into a digital format such that it can be processed by the computer. Once the signal is processed by the PC it is converted to Beats Per Minute (BPM). The BPM of an individuals heart beat can be displayed with 9 LEDs in a Binary Coded Decimal representation. As an example, an individual has an approximate BPM of 107. The LED display of the BMP is displayed in BCD format as 1 0000 0111.

Data Processing

For the data acquisition system to process data properly the baud rate of the system had to coexist with the bus clock of the microcontroller. In the case where a 14 MHz bus clock is required, the microcontroller was configured with a baud rate of 115200 bits per second as it was the highest rate possible to transmit the data without going over an error of +/- 6%. The data also had to be processed through MATLAB to figure out the number of BPM in a given signal.

Control / Communication

The embedded system was processed and serially communicated with a HP Envy 17 running Windows 10 Pro. The HP Envy system was 64-bit Operating System with specs which included an Intel® Core™ i7-4710MQ CPU processor @ 2.50 GHz with 16.0 GB of RAM. The data acquisition system was serially communicated with MATLAB, as MATLAB is great for manipulating data to be measured graphically. To successfully transmit the analog to digital conversion into MATLAB, a specific baud rate and correct COM port was required. The COM port was configured to COM2 which was set up under Device Manager. The baud rate to serially transmit the signal from the ESDX to the HP Envy system was 115200 bits per second given a bus speed of 14 MHz.

Esduino Extreme Hardware

The Esduino Extreme (ESDX) circuit board is an Arduino-style based microcontroller used as the data acquisition system for this project. The ESDX is 16-bit microcontroller that features a CPU12 core, 25 MHz bus speed (PLL enabled), 1 MHz oscillator, 240K Flash, 4K EEPROM, Flash security, 11K RAM, 12-channel, 12 bit analog-to-digital converter subsystem. In addition, the Esduino Extreme belongs to the HCS12 microprocessor family with a specific MC9512GA240 CPU. The microcontroller has a default 6.25MHz bus speed and can operate at 3.15V to 5.5V. For this project, the Esduino will serve its purpose to perform analog to digital conversion on channel AN5. The Esduino Extreme has the capability to manually change the bus speed of the microcontroller which is essential in configuring the Esduino based on our student number. The Esduino and Accessories costs around $100 and was analyzed with an Analog Discovery Module. The reason as to why the Esduino was the right choice to implement the data acquisition system is due to the need of an analog-to-digital convertor and serial communication.

Esduino Extreme - Program

The software used to program the Esduino Extreme to handle data acquisition is CodeWarrior. CodeWarrior is an integrated development environment (IDE) created by NXP to allow development of embedded applications. CodeWarrior can be programmed by Assembly language or C/C++. For simplicity and time, the data acquisition is programmed in C language.

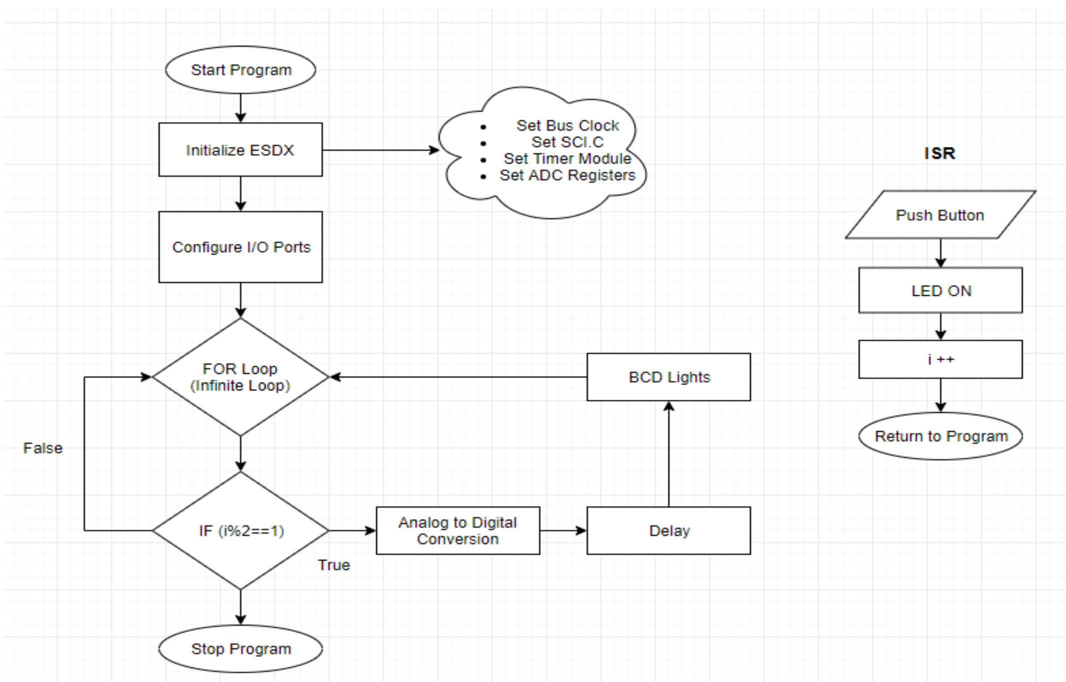

Figure 6: CodeWarrior Flowchart

3. Results / Design Process

CodeWarrior

The first step in designing the data acquisition system is to set the Bus Clock of the Esduino to 14 MHz. Using the Reference Manual given on and equations provided, the e-clock parameters were set to:

- PLL Locked (LOCK = 1)
- fREF = 1 MHz
- PLL = 2*BusClock = 2*14MHz = 28 MHz
- Set POSTDIV = 0, PLLSEL = 1, OSCE = 0, VCOFRQ = 00
- fVCO = fPLL*(POSTDIV+1) = 28 MHz * (0+1) = 28 MHz
- VCOCLK = 28 MHz
- fVCO = 2* fREF * (SYNDIV+1), SYNDIV = 13

Clock Register	Value
CPMUCLKS	0x80
CPMUOSC	0x00
CPMUSYNR	0x0D
CPMUPOSTDIV	0x00

Table 2: Clock Parameters Configuration

To verify our E-Clock of 14 MHz, we tested the bus speed by using a oscilloscope and by creating a test function to turn on/off the LED with a 1s delay. As it is evident in Figure 3, the width is 996.6 milliseconds, which confirms that our bus speed is equivalent to 14 MHz with little to no error.

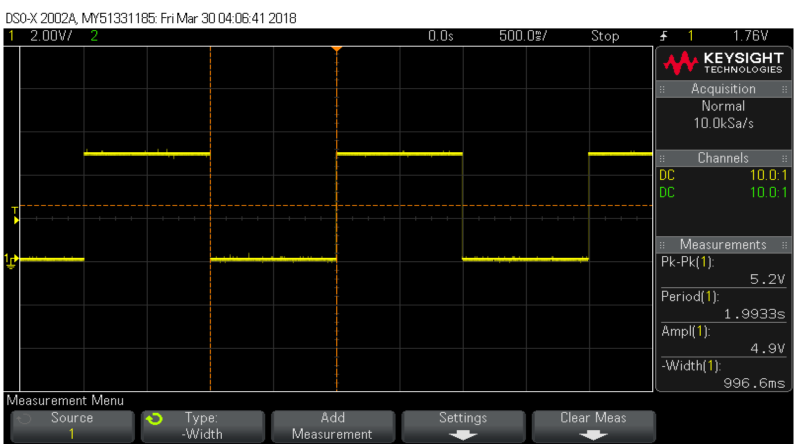

Figure 7: Oscilloscope Verification of Bus Clock

The next step of the program configuration is setting the SCI Baud Rate and Baud divisor. Using the equations provided in the Reference Manual:

- Baud divisor = Bus Clock / (16* baud rate)
- Baud Rate = (Bus Clock / divisor) / 16

- Percent Error = (Theoretical – Actual) / (Actual) * 100

Baud Rate	Baud Divisor	Percent Error (%)
2400	365	0.114155251
4800	182	0.16025641
9600	91	0.16025641
19200	13	0.928442029
38400	23	0.928442029
57600	15	1.273148148
115200	8	5.056423611

Table 3: SCI Baud Rate Calculations

Referring to Table 3, the SCI Baud Rate chosen was 115200 bits per second as it was the highest baud rate within the acceptable error of 6%. Once the baud rate and divisor were found, the SCI.C file in the Project was updated to the correct values to allow serial communication with the PC.

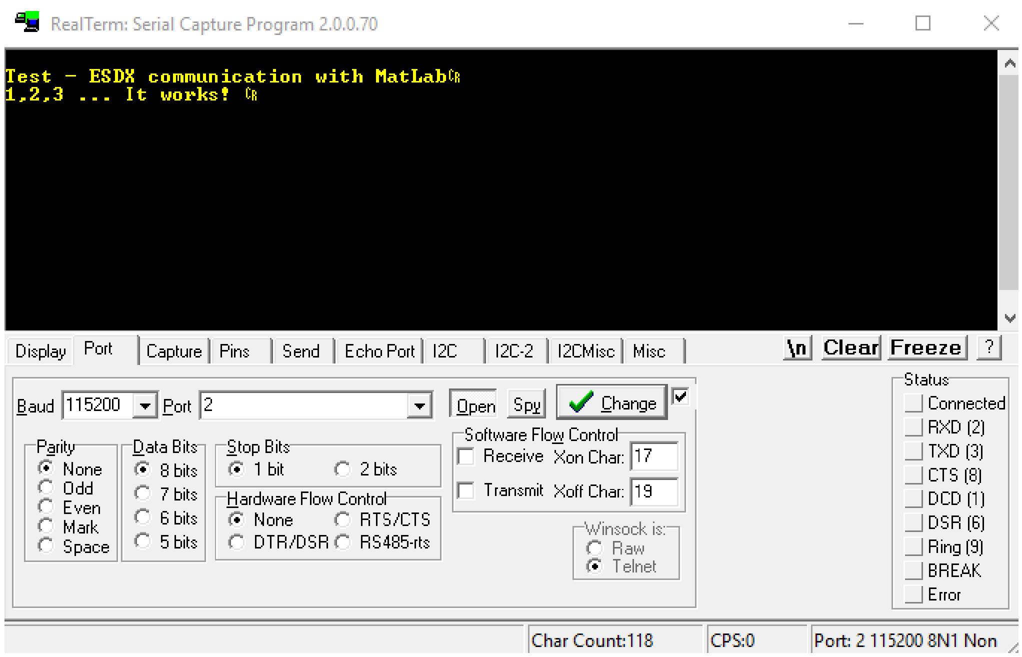

Figure 8: Testing Serial Communication with ESDX

Next, the ADC Channel had to be configured on the Esduino Extreme to produce analog-to-digital conversion. Based on the required ADC Channel of Port AN5 and ADC Resolution of 12-bits the ADC Registers were configured with the following:

ADC Register	Value
ATDCTL1	0x4F
ATDCTL3	0x88
ATDCTL4	0x06
ATDCTL5	0x25

Table 4: ADC Register Configuration

The ATDCTL1 register is set to 0x4F as it is equivalent to 12-bits. ATDCTL3 is set to 0x88 for right justified, ATDCTL4 I set to 0x06 as the prescaler value is calculated with the ATD Clock formula (ATD clock = 14 Mhz / (2*(6+1)) = 1 MHz) and finally ATDCTL5 is set to 0x25 for continuous conversion on channel 5.

Once the parameters are setup in the code, a push button program is implemented to start and stop data acquisition. The button is implemented as an interrupt with an Interrupt Subroutine (ISR) that increments a counter every time the button is pressed. If the counter value is an odd number, then the program executes the analog to digital conversion, serial communication and BDM calculation to light the BCD LEDs. If the counter variable is an even number, this indicates that the button is pressed again, and data acquisition is stopped.

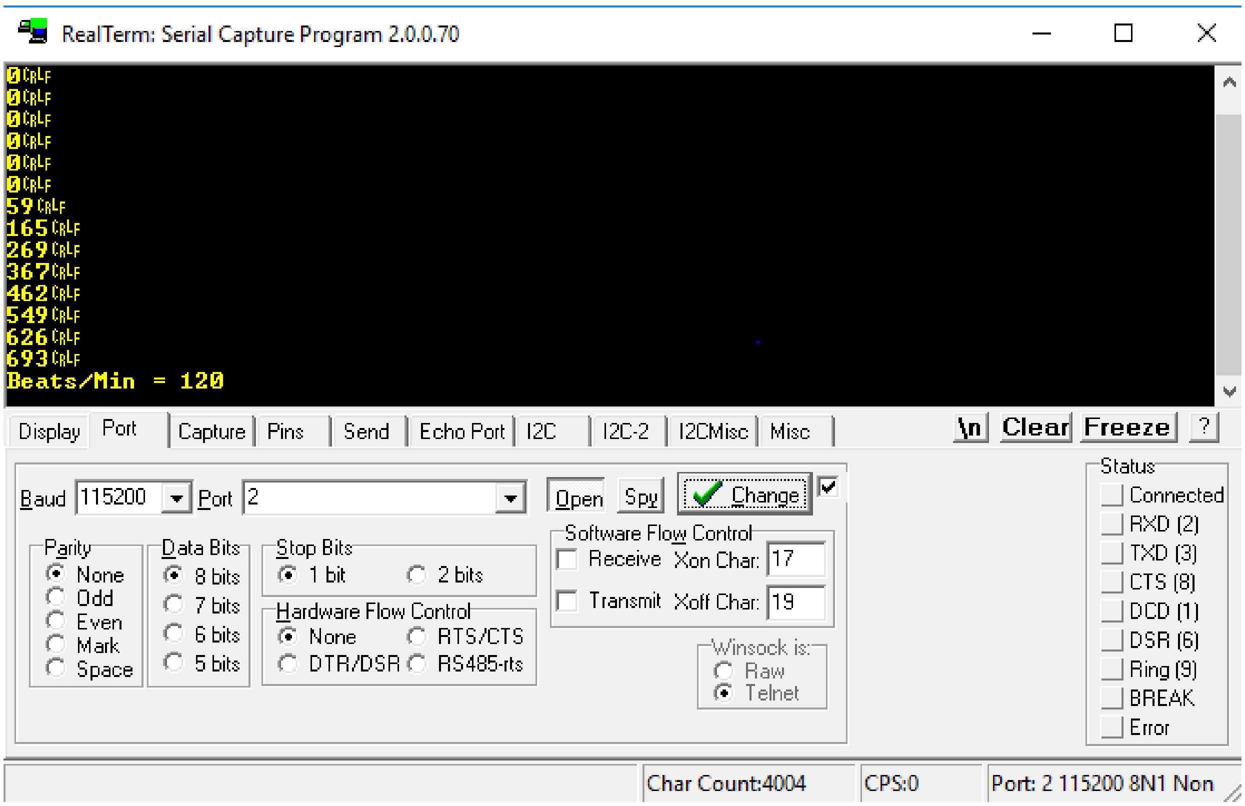

Figure 9: Serial Communication with Real Term Displaying ADC values

MATLAB

The data acquisition system collects the data that is serially communicated through MATLAB and plots a continuous waveform as long as the button is not pressed again. To set up the MATLAB file, the correct COM port and SCI baud rate needs to be selected to allow ADC values to be collected in order to graph a receiving signal on a Voltage .vs. Time plot.

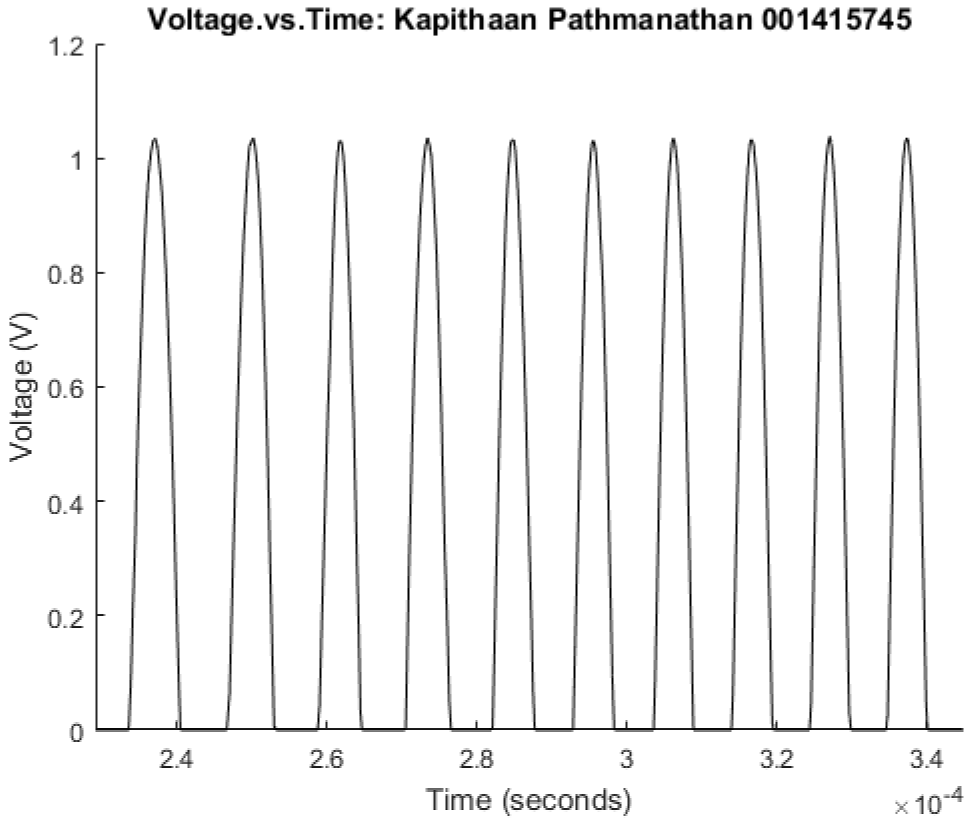

Figure 10: MATLAB Plot of Signal

4. Discussion

1. Maximum Quantization Error Calculation:

   Q = VFS / (2^bits) = (Max – Min) / (2^bits) = 5 / 2^12 = 0.0012207

2. With an assigned bus speed of 14 MHz, the maximum standard serial communication rate that could be implemented is a Baud Rate of 115200 bits per second and Baud Divisor of 8. Using the equations:
   - Baud divisor = Bus Clock / (16*baud rate)
   - Baud rate = (Bus Clock / divisor) / 16
   - Percent Error = (Theoretical Baud Divisor – Actual Baud Divisor)/(Actual baud Divisor) x 100%

   The serial communication rate is found by setting up a table with various Baud Rates to calculate the highest possible Baud Rate with a percent error of +/- 6%. With a baud rate of 115200 bits per second the percent error is equal to 5.056423611, which is suitable to implement for serial communication. The maximum value that can be implemented is 115200 bits per second because anything above that value will be too much for the microcontroller to handle. To verify this, a verifySerialCommunication() function was created to test various SCI baud rates. RealTerm was used to test the SCI baud rate of 115200 bits per second and it is evident in Figure 8 that it successfully communicated with the PC.

3. In the data acquisition system, the primary limitation on speed is due to the baud rate and bus speed of the system. The bus speed is a factor in calculating the baud rate of a specific data acquisition system. The baud rate also has a limitation on speed as it is the speed at which transmission between the ESDX and MATLAB occurs. This was tested through MATLAB as having smaller baud rates would result in a delay with the MATLAB plot whereas higher baud rates would result in a delay that is marginal.
4. Nyquist theorem states that the maximum frequency the analog signal can effectively reproduce is half of the sampling frequency. In the case when the input signal has a frequency that is greater than half of the sampling frequency, the signal become inaccurately sampled and aliasing occurs.
5. Input signals that contain sharp transitions such as triangle, square, sawtooth can accurately be reproduced with lower frequencies as high frequencies do not capture the sharp transitions due to the sampling frequency being so small. This was justified through testing the various signals on the waveform generator and by varying the frequencies from small to large.

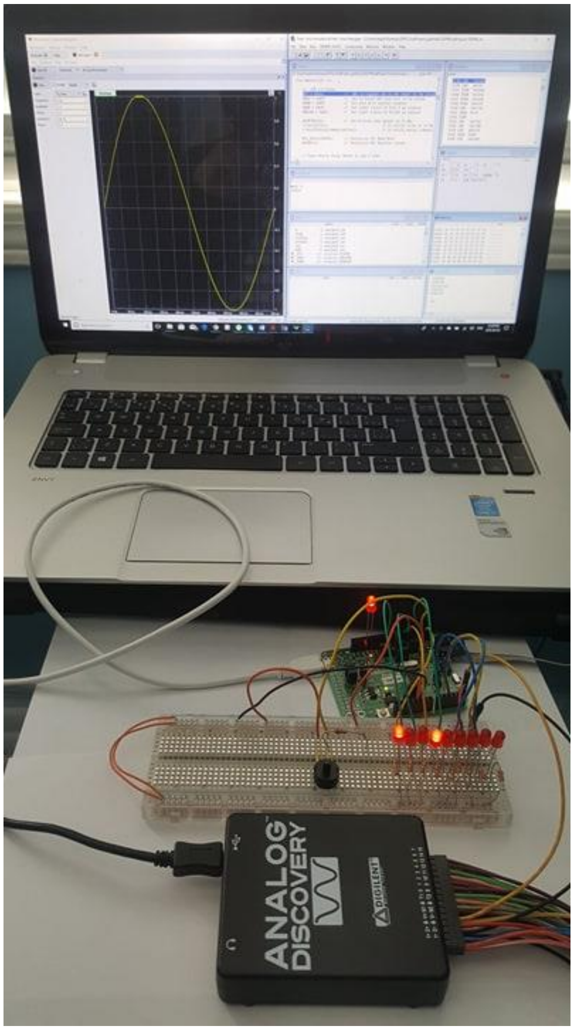

Figure 11: Data Acquisition System – Final Product

5. Conclusion

In conclusion, the purpose of this project was to understand the various stages of the data acquisition system and how acquiring data is possible through serial communication between a microcontroller and the PC. This system was used to collect data based off an individuals heart beat to convert and calculate the beats per minute in a certain time frame. Furthermore, the project was essential in developing skills to program a complex embedded system to perform various tasks such as lighting the LEDs in binary coded decimal format or to transmit the analog to digital conversion signal to a software like MATLAB. Overall, data acquisition systems have become an integral part of technology and providing useful information for different types of analysis.

6. References

[1] "ADC (Analog-to-Digital Converter) Definition", *Techterms.com*, 2008. [Online]. Available: https://techterms.com/definition/adc [Accessed: 27- Mar- 2018].

[2] M. Tung, "How it Works- Fitbit", *Jameco.com*. [Online]. Available: https://www.jameco.com/Jameco/workshop/Howitworks/how-it-works-fitbit.html [Accessed: 27- Mar- 2018].