A.C.L.A: Active Component Learning Aid

Inalvis Alvarez, Justin Berte, Richard Bielski, Zachary Freer.

Dept. of Electrical and Computer Engineering University of Central Florida, Orlando, Florida, 32816-2450

***Abstract* — A.C.L.A consists of creating a portable system that demonstrates a mechanical representation of active circuit components for students to have an easier visualization of circuit theory. The system is constructed with clear PVC pipes, motorized valves, gauges, flow meters, and an interactive display that allows students to set the circuit parameters. A pump supplies water from the lower retention tank to the main line that feeds the components. Valves are set up in each one of the component lines to control the state of the line by adjusting the rate at which the water flows, representing the current. These valves are controlled utilizing stepper motors, which open at specific angles to allow a controlled current set by the user through the interactive display as well as to change the pressures throughout the system. Measurements are made by the use of pressure sensors, which give students/users a visual representation of the voltage (pressure) as well as the rate of current flow at the various lines. These measurements are correlated to component equations such as Ohm's Law to facilitate the student's understanding of circuit theory. LEDs are allocated through each one of the lines, programmed to display the intensity of water flow, showing an increase or decrease of current**

***Index Terms* — Machine Learning, circuit theory, active components, transducers, flow meters.**

I. INTRODUCTION

Circuit theory is a very challenging topic for those who are being introduced to it for the first time. Current, unlike water is not something that can be seen and students often struggle to understand how it flows and how it affects circuit components. Since water flow is something that can be observed, it is often used as an analogy by instructors when teaching circuit theory. A.C.L.A utilizes this concept and applies it for students to truly visualize and understand electronic concepts. This is done by equating water flow to current flow, and pressure to voltage.

A.C.L.A's serves as a teaching aid for teachers and students alike. It covers active circuit components such as: MOSFET, BJT and diode. These representations are achieved by modeling the various components with the use of pipes, flow limiting valves, pressure transducers, flow meters, and motors.

The valves in the system not only help control the flow of water, but also help control pressure and serve as resistors. These valves are fully motorized through the use of stepper motors, which respond to the user's commands. Pressure readings are obtained via pressure transducers, which return an analog voltage to the signal PCB. These readings are correlated to voltages and displayed in the interactive dashboard. The user chooses which component to run, then proceeds to choose the resistors and voltage values. Once the system starts, the pump supplies water to the chosen components, and the voltage (pressure) as well as current measurements are displayed. Users are able to verify the equations, observe various modes of operations of the different components and change which circuit to implement. The current limited by the valves is easier visualized through the means of LEDs which blink in the direction of the flow and intensity of the current.

II. CIRCUIT REPRESENTATIONS

Strategic points in the mechanical system were chosen to represent, control, and gather data to be able to simulate the various circuit configurations. The diode, BJT and MOSFET configurations represented in the overall design are discussed below. Figure 1 shows an outline of the built mechanical system.

A. BJT

Figure 1a shows the electrical schematic for a basic BJT circuit. The input voltage Vin is represented by the difference in pressures shown by sensors S1 and S5, this pressure is changed by utilizing valve V2 which corresponds to the resistor Rb. The biasing voltage Vcc is represented by the difference in pressure shown by sensors S2 and S5, this pressure is set by varying the opening of the valve V4 which represents the resistor Rc. The valve V3 in between the collector and base branches of the BJT representation is utilized to give a difference in pressure thus ensuring that Vin does not always equal Vcc, however a limitation to this approach is that the pressure "voltage" at Vin will always be higher than the pressure "voltage" at Vcc due to the pressure drop across the valve. The base to emitter (Vbe) voltage shown below is represented by the pressure difference between sensors S4 and S5. Once the BJT is operating, this pressure difference will be set to a specific value equal to the turn on pressure chosen, this will be set by adjusting the opening of the valve V6. Finally, the collector to emitter (Vce) voltage shown below is represented by the difference in pressure shown by sensors S3 and S5, depending on the mode of operation the BJT is set in this pressure drop "voltage" will be either large (representing forward active mode) or

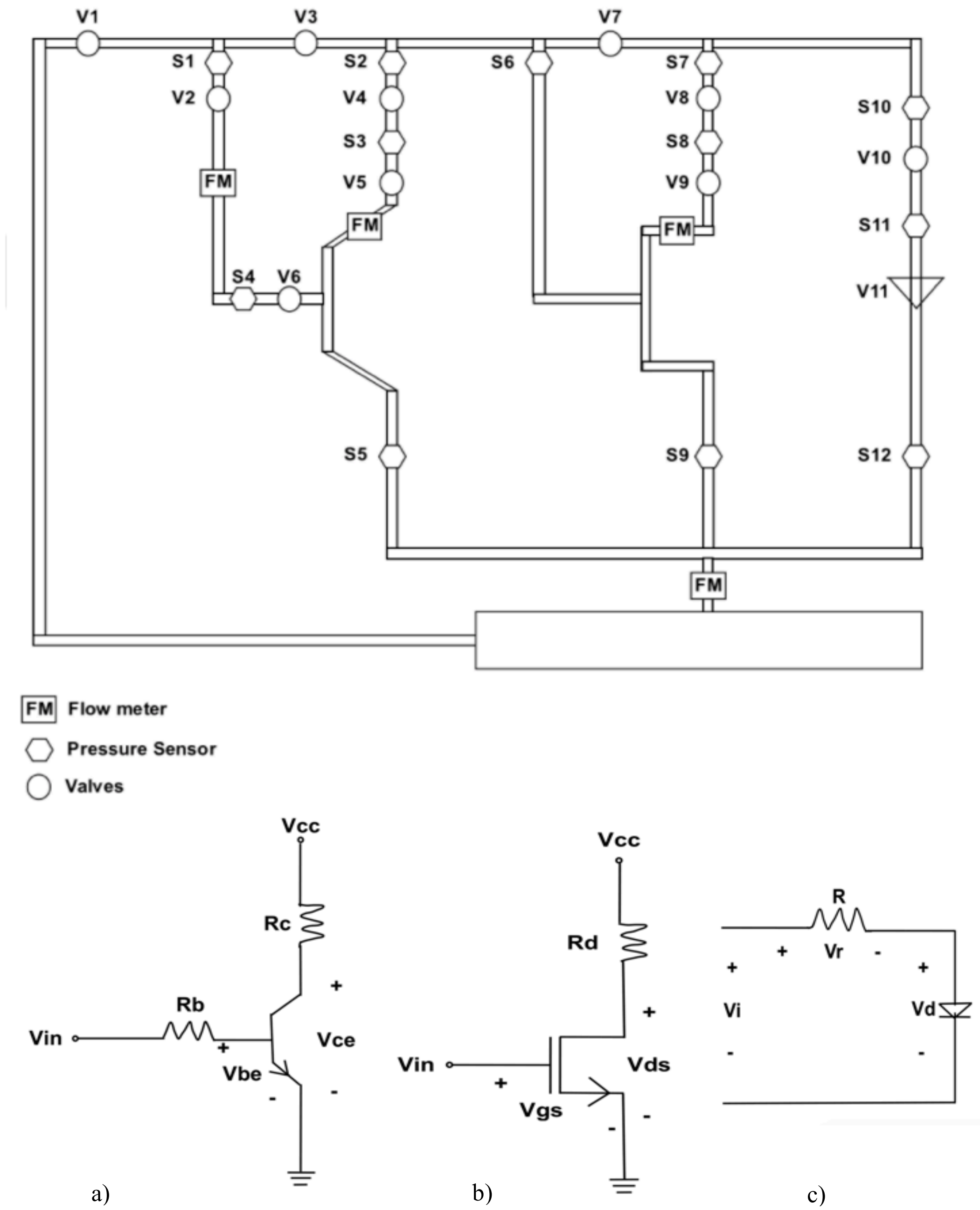

Fig. 1. Overall System Overview

small (representing saturation mode). This pressure will be regulated by varying the opening of the valve V5 which will build up pressure at the sensor S3. The values that will be able to be set by the user for this component will be the resistor values Rb and Rc as well as the two input voltages Vin and Vcc, from these values the mode of operation will be calculated as well as the drops across the resistors.

B. MOSFET

Figure 1b shows the electrical schematic for a basic MOSFET circuit. The input voltage Vin is represented with the pressure difference between the sensors S6 and S9, this voltage is set by varying the opening of the valve V7. The input voltage Vcc is represented by the pressure difference between the pressure sensors S7 and S9, this pressure difference is set by changing the opening of the valve V8 which represents the resistor Rd. Due to the inclusion of the valve V8, the voltage at Vcc can never exceed the voltage at Vin due the inherent pressure drop across V8. The gate to source (Vgs) voltage is set by taking the pressure difference between sensors S6 and S9, it can be seen from the picture below that this pressure "voltage" is simply equal to the input voltage Vin. Finally, the drain to source (Vds) voltage is represented by taking the pressure difference between the sensor S8 and S9, this pressure "voltage" will be manipulated depending on the mode of operation the MOSFET is being operated in (little voltage drop in triode region and large voltage drop in saturation mode), this pressure difference will be set by changing the degree of opening of valve V9. The values that will be controlled by the users will be the resistor value Rd as well as the input voltages to the system Vin and Vcc, the mode of operation as well as voltage drops throughout the system will be set from these values.

C. Diode

Figure 1c shows the electrical schematic for a basic diode circuit. The input voltage Vi is represented by the pressure difference between S10 and S12, this pressure can be controlled by varying the opening of V10. V10 represents the resistor R, the voltage drop across this resistor Vr is represented by the pressure drop between sensors S10 and S11, this drop can be manipulated by varying the openings of both valves V10 and V11. V11 represents the diode in the below circuit, in cutoff mode water will flow towards the closed valve that represents the diode but not through it until the desired pressure difference between S11 and S12 is achieved which corresponds to less than the turn on pressure "voltage" of the diode, at this point V1 will close cutting off the flow of water and the pump will be turned off keeping the voltage across the diode constant and approximately equal to the input voltage. When the diode is on, water will now be allowed to flow through the valve and the opening of V11 and V10 will be changed to maintain a constant pressure drop across the diode from S11 to S12 that will equal the turn on pressure "voltage" of the diode. The values that will be controlled by the users will be the resistor value R and the input voltage Vin from these values the voltage drops as well as if the diode is on will be determined.

III. HARDWARE DESIGN

The hardware diagram can be seen in Figure 2. It provides a breakdown of how the system is powered. The power PCB powers all the components of the system through two voltage rails: 5V and 12V. While the Raspberry Pi is directly connected to the touch screen which displays the system's dashboard.

A. Motorized Valves

About 2 Nm of operating torque is needed to be able to turn the clear PVC Ball Valve. The bipolar stepper motor of choice has only 1.26 Nm. Therefore, gears were designed to compensate for the necessary operating torque. To accomplish this, the number of teeth ratio

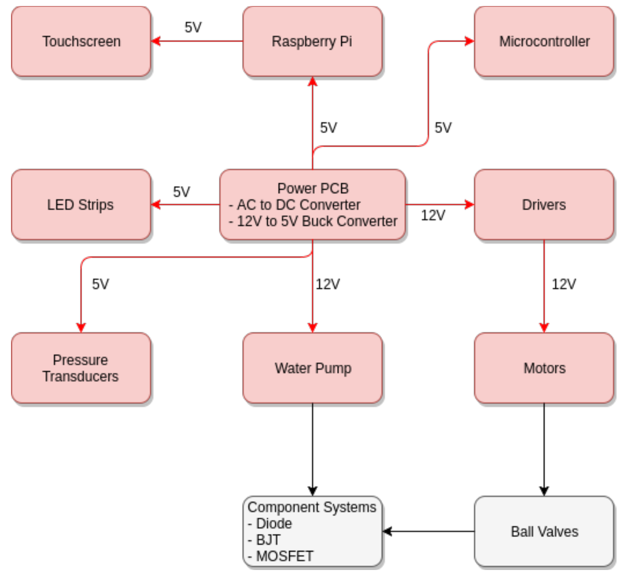

Fig. 2. Hardware Block Diagram

necessary to turn the valve was determined using the following equation:

$$T_D = T_A \frac{Teeth_D}{Teeth_A} \quad (1)$$

where T_D represents the torque required to turn the valve, T_A represents the torque of the motor. This equation gives a teeth ratio of approximately 1.25. As long as the ratio is equal to or greater than this value the motor will be able to turn the valve.

The design of the gears also took into consideration the motor rotor and valve leveler connector dimensions and its shape. Through some testing, it became obvious that due to the water flow the torque to turn the valves needed to be increased, so the teeth ratio was increased to approximately 1.83 to accommodate for this. See the Figure 3 for the gear design for both the motor and the valve respectively.

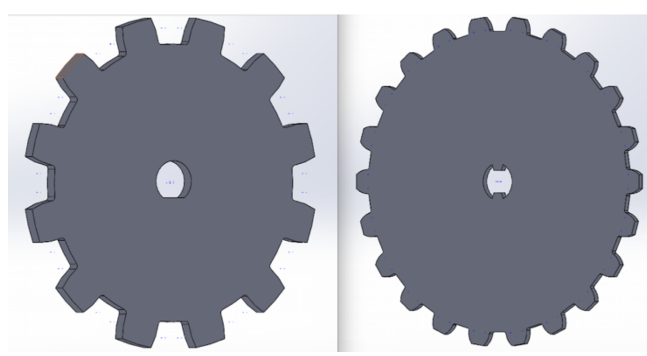

Fig. 3. Motor and Valve Gear Design

The gears image is not to scale, due to the higher quantity of teeth existing in the Valve Gear this one has a radius of 52mm, while the Motor Gear has a radius of 27mm. The teeth fit perfectly in one another, giving an increase of 0.31Nm of torque.

The bipolar stepper motor of choice provides high precision (1.8 degrees), reliability and the necessary torque. It requires a max of 2.4A of current and 12 V, as well as a driver for easy operation.

B. Transducers

Transducers provided the ability to adjust the pressure differences by manipulating the valves through feedback. Through the use of the microcontroller, the pressures are sampled once the user inputs the values and starts the system, the program will calculate the pressure differences correlate it to a voltage drop and adjust the valves to obtain the necessary values.

The pressure transducers are used to adjust the mechanical valves so that the corresponding pressure differential represents voltage in the system. This will be performed by relaying the pressure differential to the microcontroller. Once the microcontroller receives the pressure differential it will process the data to determine if the correct relationship has been applied. If the relationship is as it should be, the microcontroller issues no commands. If the relationship is not correct, the microcontroller adjusts the valves correspondingly in an attempt to create the proper pressure differential. This will be done until the proper pressure is received by the microcontroller to reflect voltage in the system. The Eyourlife Universal 30 PSI pressure transducer is designed to work with water among other fluids.

C. Flowmeters

The Fill-Rite In-line Digital Flow Meter meets has a maximum operating pressure of 999.7kPa (145 PSI), flow rate in the range of 189.3-1640.4 cm3s (3-26 GPM), and an accuracy of 1% . The LCD display has many features such as displaying battery condition, calibration mode, rate of flow and total flow along with the unit of measurement. It is important to perform calibration on this product to eliminate as much error as possible in the readings. This is done by determining the correction factor using the following equation:

$$CF_P = CF_C * \frac{Actual\ Value}{Displayed\ Value} \quad (2)$$

where CF_P is the proper correction factor and CF_C is the current correction factor. This flow meter has an inlet/outlet of 2.54 cm (1 in), since the diameter of the PVC pipes is of 1.27 cm (0.5 in) two 1 in to 0.5 in NPT adaptors per flow meter are needed. [1] It is battery powered, therefore it had no external current/voltage requirements.

D. Pump

The “neutral ground” is represented by the use of a tank where all the water supplied to the system is stored. A main pipe is connected to the tank and used to distribute the water through the components in the system and then is then drained back to the tank. Due to this a submersible pump is necessary to transport all the water from the tank up to the main line. The length of the main pipe from the tank to the top is about 2.1m (7ft), therefore it is required that the pump is able to have enough force to pump the water to the top. It is also necessary for the pump to be able to build enough pressure to acquire measurements to represent the voltage drops through the system. Therefore, after careful considerations, ¼ HP Submersible Sump Pump was chosen. The pump connection is controlled by supplying a high signal from the signal PCB to a solid state relay.

E. Raspberry Pi 3

The Raspberry Pi 3 is used mainly as the front end of the system. It is responsible for running the graphical user interface where the user can interact with the system. The Raspberry Pi 3 is also responsible for relaying the corresponding components values from the graphical user interface to the microcontroller for calculations. The Raspberry Pi 3 supporting a full Linux operating system produces an ad hoc network for a client to connect to. This allows for the client to remote desktop into the Raspberry Pi 3 essentially mirroring the graphical user interface for an audience. In addition, the Raspberry Pi 3 receives transmissions from the microcontroller with information regarding the pressure of the transducers. This information is displayed on the graphical user interface so the voltage drops across components are visible.

F. ATMEGA2560

The ATMEGA2560 is an 8-bit RISC based microcontroller. It has 256KB of flash memory, 8KB of RAM, and 54 general purpose I/O pins 14 of which are capable of pulse width modulation [2]. This Microcontroller is utilized because of its sheer number of General Purpose I/O pins. This is important to the system because of the number of components which need to be controlled, 11 motor drivers (pulse, direction, enable pins), 12 transducers (analog pins), several LED strips, communication with the Raspberry Pi 3, and the pump. The ATMEGA2560 providing 54 GPIO pins allows for flexibility on how to communicate with these components without the hindrance of number of available pins. The ATMEGA also is implemented on the Arduino Mega 2560 development board, which allowed for easy development of the MCU and testing before being implemented off the board. The MCU operates at 5 volts.

It is capable of providing 40 mA of DC current through Input Output pins and a clock speed of 16MHz [3]. The variety of GPIO pins on this microcontroller also offers a wide variety of ways to communicate with the Raspberry Pi 3 both in parallel and serial methodologies. "I2C: 20 (SDA) and 21 (SCL). Support I2C (TWI) communication using the Wire library.", the ability to communicate with the device through a serial communication such as I2C will allow for better synchronization, and more free GPIO pins to communicate with components. [3]

The ATMEGA2560 is also responsible for receiving voltages from the pressure transducers. The pressure transducers will convert the pressure at a given point in the system to analog voltage which the microcontroller can read from an analog input pin. The ATMEGA2560 will read from the input pin it will then convert the corresponding voltage to a pressure and pass that pressure using I2C protocol to the Raspberry Pi 3 for display on the graphical user interface.

The analog input for the ATMEGA2560 is capable of reading voltages up to 5V. The analog input reads voltage for 1024 units. This allows for differentials in .0049 V giving a very high accuracy in analog read executions. The analog pin is capable of reading at a rate of 100 microseconds or up to 10 thousand times per second. This will allow for a high rate of sampling for the application.

G. LED Strips

The Alitove WS2812B is an individually addressable LED strip. The led strip features 5050 components which allows for a clear display for individual LEDs. The LED strip is capable of transmission of a signal with a distance of 5 meters without any increase circuit. This is crucial to allow for transmission of the "blink" function through a branch in the system. The transfer protocol used for this LED strip is a single NZR communication mode. The LED strip is capable of full-color mode. It is waterproof and is capable of supporting all of the purposes of the system. [4] It consists of 30 segments per meter, 24-bit full color, operating voltage of 5V DC, built-in IC short circuit protection, and signal reshaping circuit (ensure distortion of the wave does not accumulate), and consumes 0.3W/LED.

They are configured in a cascaded method, which transmits data sent by the microcontroller in a single line. This allows for a smooth flow of data transmission through the LED strip. Each pipeline representing the connection between the various components have LEDs running behind it. The higher the current set by the user, the faster the LEDs will blink in a flow like manner to indicate direction. Each pipeline is color coded to indicate where the current comes from and how is separated and/or merged after each branch. For circuit representations such as the BJT, the LEDs represent the current relation between beta and the base current with respect to the collector current. The LEDs provide a visual guide when it comes to current flow since in electronics this is something that is impossible to observe and is only possible to measure.

IV. Software

The signal PCB houses the brain of the system, ATMEGA2560 microcontroller. It is responsible for initializing the system, this entails providing all of the motors attached to the valves of the pipe with the corresponding inputs for correct rotation angle. This allows the system to function in an automated fashion. The microcontroller is also responsible for acquiring data from the pressure transducers, communicating the data to the Raspberry Pi and controlling the LEDs. Figure 4 displays the signal distribution of the system.

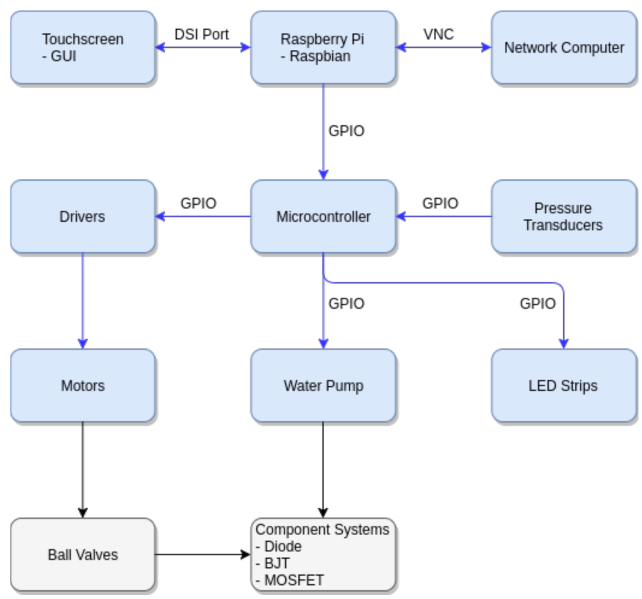

Fig. 4. Software Block Diagram

A. I2C

Inter-Integrated Circuit allows for multiple masters and multiple slaves. I2C is a synchronous protocol that only requires two pins: a clock wire (SCL) and a data wire (SDA). [5] The ATMEGA2560 microcontroller is the only slave to the Raspberry Pi master, transmission and addressing conflicts would not be an issue.

A script in Python was created for the Raspberry Pi which initialized a serial connection with a specific address that referenced the ATMEGA2560 slave. It then allows to input data through the console which the Raspberry Pi would transmit to the ATMEGA2560 using the I2C communication protocol. Whenever the Raspberry

Pi receives data, it automatically displays it on the console. On the microcontroller side, a program was flashed to initialize a serial connection with a specific address that references the Raspberry Pi master. When the program runs, the ATMEGA2560 stores any data received from its data line in a variable. When the transmission is complete, it sends the data in the variable back to the master via the data line.

B. Wireless Communication

An Ad Hoc network powered by the Raspberry Pi was chosen to allow for communication to the system through a remote connection without the dependency on other variables such as other Wi-Fi network connections or access to the internet. The Raspberry Pi 3 having an integrated network card it can receive connection from other devices. Tools used for the Raspberry Pi to act as an access point include hostapd, isc-dcp-server, and udhcpd. These tools can be used to configure the access point's name, passwords, and selection of the wireless card acquiring values from a helpful tutorial [6]. Furthermore, the network can be started and stopped with a single terminal command. This terminal command is configured in the rc.local file and making the file executable, allows the command to run at the startup every time the Raspberry Pi is booted. The implementation allows for an access point to be immediately available upon startup without any interaction with the device.

The remote desktop application chosen for this system is Realvnc. The server has been implemented on the Raspberry Pi 3 to allow connections to it and the client will need to installed on the user's external device. Once the user has the client installed on their device they will enter the Raspberry Pi's static IP address and insert the password. This will instantly allow the user access to Raspberry Pi's interface. The implemented system allows for this ability to mirror the user interface without the use of unpredictable external variables.

C. Machine Learning

Machine learning is used to allow the system to fine tune itself and find the corresponding inputs to reflect the desired output of the pressure transducers. The system implements a supervised learning technique where available inputs will be provided to the microcontroller as well as the desired outputs. The inputs of the system will consist of the valves available for each component, while the desired outputs will be the pressure drops between transducers. The microcontroller is the device physically executing each run and test, however, the algorithm and data are processed and stored on the Raspberry Pi 3. Each state of the system is recorded and tested by the Raspberry PI 3 in connection to the microcontroller. When the feedback from the transducers becomes available it is plotted with the corresponding state and given a value of acceptance. The value is better the closer to the desired result and worse as it gains distance from it. The algorithm then processes this data to find a pattern and ultimately reach its goal for pressure drop (with a certain percent tolerance). The program is developed in python as that is the recommended language on the Raspberry Pi 3 as well as favored by data scientists for the art of machine learning. The data set and results are stored and processed on the Raspberry PI 3 allowing for updating of the algorithm as well as recalibration of the system. Overall this provides the system with a way to adapt to changes and provide better performance. [7]

D. User Interface

GUI development is a key aspect of designing the system. Unlike the rest of the software and a lot of the hardware needed to operate the system, the GUI is the key methods for a user to interact with the system (whether physically or through VNC). Because of this, the GUI was designed to be simple, lightweight, and with obvious, well-defined buttons.

The "Setup" menu is where the user initializes the system before running. This menu displays a image of the system using circuit schematic symbols where their respective components are located. The user is then prompted to select one of the circuit symbols to modify that component's parameters. For each component, there is a set of drop-down menus with preset values (to prevent user error) for each parameter of the component. Once each parameter is set, the user is prompted to push "Start". This would lock the screen until the user pushes "Stop" or "Reset" which pauses the system so the user can adjust the parameters or stop the system and return to the main menu.

The "VNC" menu provides instructions on how initialize VNC for remote control from any Wi-Fi-compatible device. It describes how to connect to the ad-hoc Wi-Fi network the Raspberry Pi broadcasts and sign in using the WPA2 passphrase. It also instructs on how to install RealVNC Viewer and connect to the now-local Raspberry Pi VNC server.

The "Settings" tab allows the user to control other various aspects of the system. A "Help" menu is accessible from the top left of the screen. This menu offered general information about the system and its designers and a guide on how to configure and run the system, describing all of its menus, buttons, and results.

Finally, a startup shell script for the Raspberry Pi is implemented, so that whenever it boots up, the computer automatically launches the GUI. This was intended to reduce the initial steps to turn on the system for the first time.

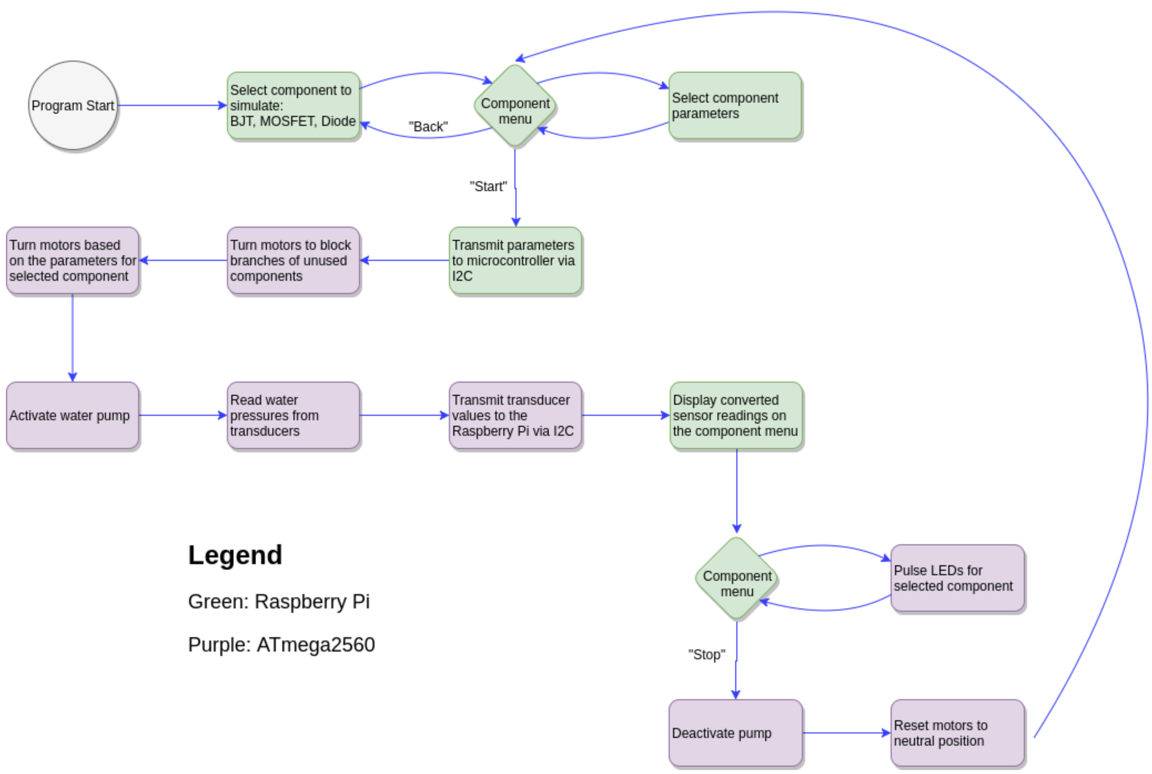

Fig. 5. System Flowchart

E. Flowchart

When the Raspberry Pi boots up, a startup script automatically loads the Python 3 GUI script which shows the user three options of components. From there the user can select either the BJT, MOSFET, or diode to simulate. For each of the respective components, the application transitions to a menu of parameters for the selected electronic device. At this point, the user can select "Back" to return to the main menu. If the user is satisfied with the parameters selected, he/she pushes "Start" for the Pi to transmit the values via I2C protocol to the ATMEGA2560 microcontroller. The microcontroller realizes based on the data received which of the two devices are unused and send signals to the respective motor drivers to turn the valves to those branches, blocking any water from flowing through them. Then the microcontroller directs the valves of the desired branch to rotate so that it will emulate the results of the user selected component with the given parameters. After all the motors are aligned, the microcontroller signals the relay to activate the water pump, then reads the analog pressure transducers. Upon storing values, the microcontroller transmits them to the Pi which displays the pressures (converted to the corresponding voltage levels based on prior calculations) as voltages on the GUI. After the transmission, the microcontroller blinks the LEDs to help illustrate the current flowing. Whenever the user wants to stop the current representation, they may select "Stop" on the GUI. This causes the Pi to signal the microcontroller to stop the pump then reset all the motors to the neutral position. (See Figure 5)

VI. Conclusion

The overall purpose of this system is to provide a teaching aid for students pursuing educations in science and engineering fields. Professors will be able to visually represent the current flow and voltage drops across Bipolar Junction Transistors, Metal–Oxide–Semiconductor Field-Effect Transistors, and Diodes. The components operating such that the water flow through the pipes acts as current flow through the devices and pressure drops between pressure transducers represent voltage drops throughout the circuit. This will allow for the professors to represent these components using a new method to those students who may benefit from visual aids. The project also poses an amazing learning experience for both the electrical and computer engineering students who participated in the design as well as implementation of this device.

Acknowledgement

The authors wish to acknowledge the assistance and support of Dr. Chung-Yong Chan, for his guidance, dedication and patience. Also, many thanks to Siemens Energy, Inc. for the funding that made this project possible.

References

[1] Sotera Systems. FR1118-P10 User's Manual. Tuthill.

[2] "AtmelATmega640/V-1280/V-1281/V-2560/V-2561/V," Atmel ATmega640/1280/1281/2560/2561 Datasheet. [Online]. Available: http://www.atmel.com/Images/Atmel-2549-8-bit-AVR-Microcontroller-ATmega640-1280-1281-2560-2561_datasheet.pdf. [Accessed: 30-Nov-2017].

[3] "Arduino MEGA 2560," Arduino MEGA 2560. [Online]. Available: http://www.mantech.co.za/datasheets/products/A000047.pdf. [Accessed: 30-Nov-2017].

[4] Alitove WS2818 LED Strip Datasheet

[5] Stevenvh, "Re: USART, UART, RS232, USB, SPI, I2C, TTL, etc. what are all of these and how do they relate to each other?," Stack Exchange, 13-Aug-2012. [Online]. Available:https://electronics.stackexchange.com/questions/37814/usart-uart-rs232-usb-spi-i2c-ttl-etc-what-are-all-of-these-and-how-do-th. [Accessed: 09-Oct-2017].

[6] P. Martin, "Using your new Raspberry Pi 3 as a WiFi access point with hostapd," Frillip's Blog, 13-Jun-2016. [Online]. Available:https://frillip.com/using-your-raspberry-pi-3-as-a-wifi-access-point-with-hostapd/. [Accessed: 30-Nov-2017].

[7] V. Maini, "Machine Learning for Humans, Part 2.1: Supervised Learning," Medium, 19-Aug-2017. [Online]. Available:https://medium.com/machine-learning-for-humans/supervised-learning-740383a2feab. [Accessed: 30-Nov-2017].

Biographies

Inalvis Alvarez Fernandez, senior Electrical Engineering student at the University of Central Florida. Currently an undergraduate research assistant. Will be attending graduate school in Fall 2018 to obtain a Master's Degree in Electrical Engineering with a concentration in Power Systems.

Justin Berte, currently a senior at the University of Central Florida. He plans to graduate with his Bachelor's of Science in Electrical Engineering in May of 2018. He is currently working as a Developmental Engineer for the United States Air Force. He plans to attend graduate school at the location in which he is stationed. His primary interests lie in semiconductor devices as well as control systems.

Richard Bielski, a senior Computer Engineering student at the University of Central Florida. He pursued an internship with Nielsen Media research in their ETP-Engineering internship program. Upon graduation with this Bachelor's of Science, he will continue his career as a full time ETP-Engineer in Nielsen Media Research's Engineering department.

Zachary Freer, a senior Computer Engineering student at the University of Central Florida. He currently works as software engineer with Lockheed Martin's Missiles and Fire Control in the CWEP program. Upon graduating with his Bachelor's of Science, he will continue his career as a Systems Integration & Test Engineer in Harris Corporation's Space and Intelligence Systems Segment.