

“R” Project - UCF Stormwater Academy Rain Simulator Control System Redesign

Adam Cutting, Luke Falls, Brenda Garcia,
David Levy

College of Electrical Engineering and Computer Science, University of Central Florida, Orlando, Florida, 32816-2450

Abstract — This paper defines the design and building process for the UCF Stormwater Academy Rainfall Simulator control system. This rainfall simulation system is a combination of electrical, computer, and mechanical engineering sub-systems. Our team has been challenged with the task of completely redesigning and rebuilding the entire control system for the rain simulator including a wireless rain gauge for real-time verification of the user designated rainfall rate. Other components of this project include: stepper motor control, DC power supply design, microcontroller programming, signal multiplexing as well as wireless communication.

Index Terms — Low Power Electronics, Microcontrollers, Multiplexing, Pulse Width Modulation, Wireless Sensor Networks.

I. INTRODUCTION

The UCF Stormwater Academy Research Lab recruited our group to assist them with their ongoing issues with their Rain Simulator. The research lab is responsible for testing many different materials for both internal research purposes as well as external contracted purposes. They have performed substantial work for the Florida Department of Transportation (FDOT) among other customers. Their primary work is to simulate the short intense rainfalls predominant in Florida and to simulate their effects on many different materials. Included in these materials are concrete used to build Florida highways, asphalt used to pave the streets, as well as soil which is frequently enriched with different chemicals and additives. In the case of the concrete and asphalt, it is the aging process of the substance that is being put to the test whereas with the soil, it is the chemical run-off that is being measured. The entire rain simulation system was designed and created by an external company and cost upwards of \$100,000 to complete. Throughout the past year, researchers at the lab have noticed that the simulator has been producing an inaccurate rainfall output. And, the

system has been having communication problems from its user interface to the motor driving circuitry. The user interface is implemented on an old laptop, which delivers the test parameters to the drivers via a dilapidated RJ-11 cable. Also, one of the systems power supplies intermittently fails. The research lab recruited our group to redesign this control system into a leaner, more reliable system that would correctly drive their designated rainfall rates during simulations.

II. SYSTEM SUB-ASSEMBLIES

At a high level, the system can be broken down into two primary assemblies: mechanical and electrical. The mechanical assembly consists of the pre-constructed rainfall test bed, rain nozzles, stepper motors, and overall support structure of the simulator. The control system which sends the appropriate signals to drive the mechanical assembly is the electrical assembly that we have been prompted to rebuild. The electrical assembly can be broken down into sub-assemblies. These sub-assemblies are: power system, motor drivers, main microcontroller unit (MCU), communication system, user interface and sensors. The power system, communication system, user interface and sensors can be broken down even further. Our design contains two different sensors: a wireless rain sensor located on the rain bed and over-travel sensors located on each of the two stepper motors. Furthermore, the power system consists of a high voltage and a low voltage power supply. The high voltage portion of the power supply is used to drive the large stepper motors, which in turn control the rain nozzles, and the low voltage portion of the power supply is used to drive all of the signals needed for the MCU, motor drivers and user interface. The communication system also consists of two different parts: the wireless integration needed for the aforementioned rain sensor and two signal-carrying transmission lines that travel from the system enclosure, which houses all of the control electronics such as the MCU and the mentioned peripherals, to the stepper motors. Finally, the user interface consists of an input and output device. We have an LCD set to provide output to the user and a keypad for the user to provide input to the system. All of the mentioned sub-assemblies were procured and constructed by our group. Although many components were purchased, the ultimate integration was completed entirely by the team.

III. MAIN SYSTEM MICROCONTROLLER

Since the goal of this project was to simplify the existing control system and eliminate the need for a laptop

computer and all of the interconnecting hardware, a suitable microcontroller had to be chosen. The Texas Instruments MSP430F5438a microcontroller was chosen for this task. The F5438a is a 100 pin microcontroller with many built in peripheral modules that are very useful for HMI systems and motor control applications. Texas Instruments donated an F5438a development board and Field Emulation Tool which made it possible to write, test, and verify the software throughout the design process.

In a digital motor control system, the microcontroller must be capable of monitoring internal and external sensors or events while concurrently providing the necessary signals to control the motor. That is why the built in peripheral modules are important, because they can be set up to work independent of the microcontroller's main processor. The built in peripheral modules that are used in this project consist of the following:

- Pulse Width Modulated Outputs
- UART Serial Communication
- Real Time Clock
- Timer Generated Interrupts

The pulse width modulated signals are created using the built in timer modules which can automatically generate PWM outputs.

Two UART Serial Communication modules are utilized in this project. One is a 9600 BAUD output to the serial enabled LCD for the user interface. The second one is also 9600 BAUD, and it is used to receive the rainfall levels from the wireless XBEE module.

The Real Time Clock (RTC) was used in this project to easily keep track of the simulation duration and update the LCD with the correct amount of remaining minutes. The RTC automatically refreshes the seconds, minutes, and hours registers, and these registers can be read for a time reference.

The internal timers are used to generate interrupts for the control of the motors. Specifically, the step rate period and the rain nozzle pause duration are generated with the timers. This allows the processor to work on other tasks while the timers are counting down in the background.

IV. MOTOR CONTROL TECHNIQUES

The motor driver chips each require four pulse width modulated (PWM) inputs that are supplied by the microcontroller. The purpose of the PWM is to limit the current that is delivered to the motors. The PWM signals are fed directly to the gates of the full bridge FETs that supply the current to the motor windings. Since we are driving the motors with a voltage that is much higher than

they are rated for, care had to be taken to avoid overamping the motors. The transient response of the motor winding current needed to be limited to the shortest time possible, so by applying the full 42 VDC signal to the motors, the current will rise exponentially to the rated value very quickly. Then the PWM signal from the microcontroller needs to effectively chop that 42 VDC signal to the rated voltage of the motor (2.7 VDC) so that the steady state current of 5.6 amps can be obtained. The speed of the transient response is very important because it determines the maximum stepping rate of the motor. Each motor has two windings and each winding requires two PWM signals for a total of eight. However, to reduce software complexity, we decided to parallel the PWM signals for both motors which reduces the count to four. This is acceptable because the two motors need to be synchronous with each other for proper operation. Since this system has no feedback, simulations were necessary to determine the amount of time required for the motor winding current to reach the rated value.

(1) is the expression for motor winding current as a function of time (I_0 is the initial current). This equation was used in Matlab to simulate a PWM input. And, Figure 1 shows the simulated response of the current in a motor winding.

$$i(t) = I_0 e^{-\frac{R}{L}t} + \frac{V}{R} (1 - e^{-\frac{R}{L}t}) \text{ Amps} \quad (1)$$

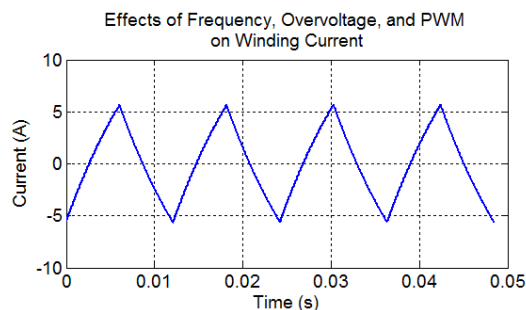


Fig. 1 Current in one winding of one stepper motor. 42 VDC input, 25% duty cycle, 40 kHz PWM, 16 steps @ 330 steps per second.

Ideally, when full stepping a motor the winding current should look like a square wave. In Fig. 1 notice that the motor winding current reaches 5.6 amps but the waveform is not the desired square wave.

A peak-to-peak transition represents two steps of the motor. The problem with the triangle wave is that every other step of the motor is either very low torque or missed entirely because the winding current is passing through

zero when it should be at 5.6 amps. To remedy this problem the duty cycle must be manipulated to start at 100% just long enough for the winding current to reach the rated value of 5.6 amps and then reduced to 6% to remain at that value of current. Figure 2 shows the result of the Matlab simulation. Notice that now every step of the motor receives the rated 5.6 amps.

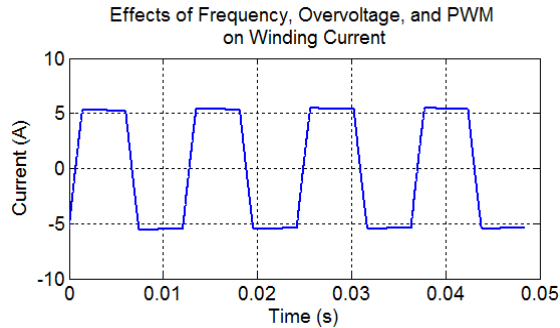


Fig. 2 Current in one winding of one stepper motor. 42 VDC input, 100% duty cycle for 1.5 ms, reduced to 6% duty cycle, 40 kHz PWM, 16 steps @ 330 steps per second.

Another important consideration when trying to drive a stepper motor at a high speed is a linear velocity profile. This is true especially with large motors that have high rotor inertia and are driving a load. To accomplish the linear velocity profile using timer interrupts, the time between successive interrupts must be multiplied by a continuously changing ratio [1].

$$Ratio = \frac{\sqrt{1 + \frac{1}{n}} - 1}{1 - \sqrt{1 - \frac{1}{n}}} \quad (2)$$

(2) would have been very time consuming to calculate in real time, so it was approximated with a second order Taylor Series approximation.

$$Approx. Ratio = \frac{4n - 1}{4n + 1} \quad (3)$$

In both (2) and (3), the 'n' is the value that goes in the timer register that generates the stepping interrupts. (3) is very simple to calculate in real time. The present timer value is multiplied by the approximate ratio during every step to generate a new step time which corresponds to a linear velocity ramp.

V. STEP SPEED CALCULATIONS

To determine the actual step rates of the motors for the range of rainfall intensities we came up with a method for calculating the step rates that the existing system utilizes, such that we could match and then exceed the level of accuracy of the existing system. To do so, we recorded the time that it took for the nozzles to pan across the exposure windows at intensities varying from 0 to 12 inches. We found the rotation angle of the output of the 10:1 motor gearhead to be 105.5° . One motor step correspondes to 1.8° rotation, before the gearhead. Thus, using the sweep times for each intensity and the rotation angle the step rates were determined per (4).

$$StepRate = \frac{105.5^\circ * 10}{(1.8 \frac{Degrees}{Step}) * t} \quad (4)$$

The measured step rates were then graphed and found to be linearly correlated. We used this information to extrapolate the function of the step rate vs. the rainfall intensity. Figure 3 depicts the results of our findings. The resulting linear function was then used in our software to generate the motor step rates for each rainfall intensity.

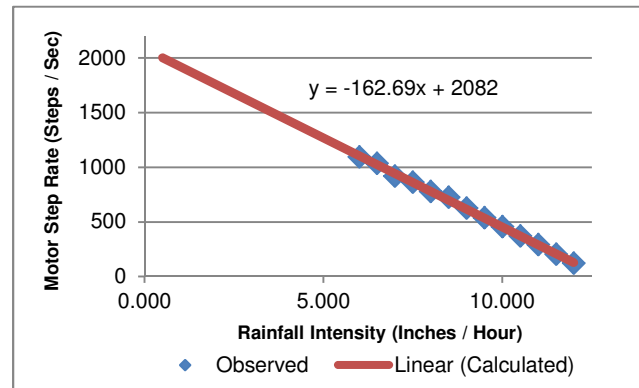


Fig. 3 A plot of the motor stepping rates that correspond to the range of rainfall intensities for the existing system.

VI. SYSTEM SOFTWARE

The software for the rain simulator was designed to implement a state machine. It is entirely interrupt driven, and in an effort to reduce processor overhead, it does not use loops or software delays. To ensure that all of the interrupts are handled in the proper order and at the proper times, interrupt vectors were used in the software. This provided a fast and easy way to decode and prioritize all

of the interrupts. This was one of the built in features of the F5438a microcontroller that was used in our system.

The software has to handle all of the user input from the interface buttons, as well as the faults and warnings that are automatically generated by the motor drivers, and it

has to receive and analyze data from the wireless rain sensor. A sufficient level of error checking was built in to avoid erroneous user input. Figure 4 shows a high level view of the system software.

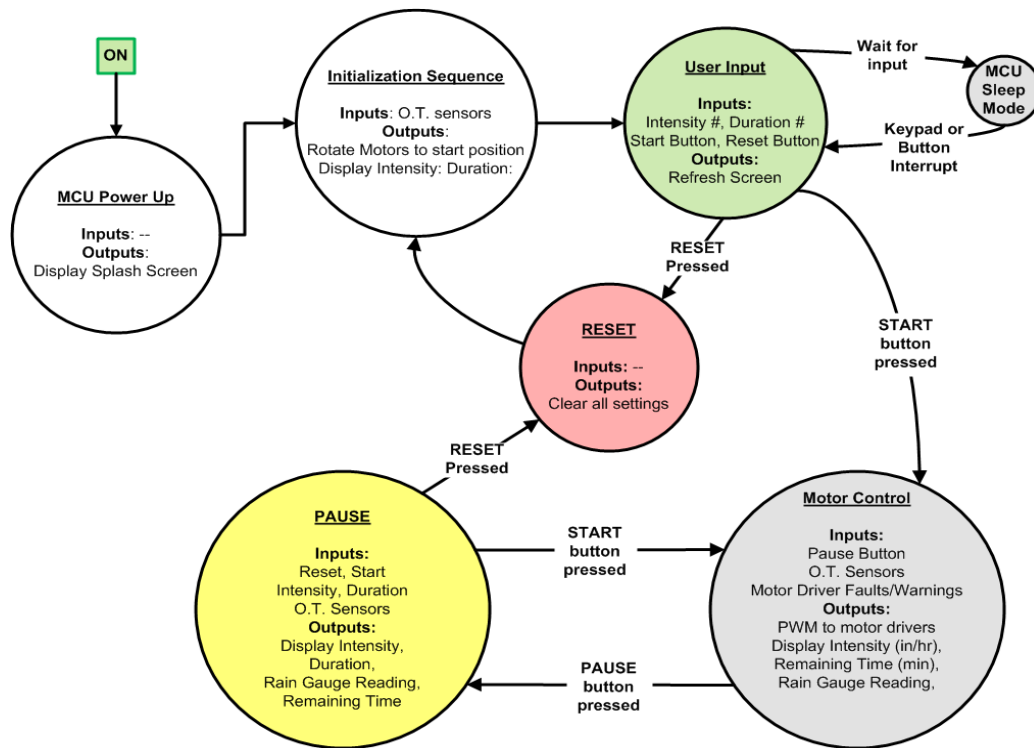


Fig. 4 System Software Flow Chart.

VII. MOTOR DRIVERS

The Texas Instruments DRV8432 integrated motor driver is capable of handling the current demand of the motors at the rain simulator. The motors that are in place at the rain simulator will draw up to 5.6 amps per phase each, and the DRV8432 is capable of handling 7 amps per phase. The DRV8432 is a dual full bridge motor driver with advanced protection circuits built into the device. [2] In dual full bridge mode, the motor driver is capable of driving a motor with 50 volts and 7 amps. As was stated previously, the motor drivers have built in protection systems for under voltage, over temperature, over current, and short circuit conditions. In addition to those protections, the drivers do not require any external snubber or Schottky flyback diodes. As discussed in the

power supply section, there is a need for two supply voltages for the motor driver chips. They each require a motor supply voltage as well as a supply voltage for the motor driver chip itself. The operating conditions for the motor drivers as used in this project are shown in Table 1.

TABLE I
OPERATING CONDITIONS OF DRV8432

VDD – Digital supply voltage	12 VDC
GVDD_X – Logic supply voltage	12 VDC
PVDD_X – Half bridge supply voltage	42 VDC
Peak current output per pin	5.6 A
Continuous current per output pin	5.6 A
PWM switching frequency	40 kHz

A. Basic Operating Modes

The motor driver IC is designed to run in one of four operating modes; the dual full bridge mode with two pulse width modulation inputs each full bridge and cycle-by-cycle current limiting (CBC), dual full bridge mode with two pulse width modulation inputs each full bridge and over current latching shutdown, parallel full bridge mode with cycle-by-cycle current limiting, and dual full bridge mode with one pulse width modulation input each full bridge and cycle-by-cycle current limiting. For the first two modes, the PWM_A input controls half bridge A, PWM_B input controls half bridge B, and so on. For this project, the dual full bridge mode with two PWM inputs each full bridge and cycle-by-cycle current limiting will be used. This mode allows the most versatility in control.

B. Protection Devices

Another nice feature of the DRV8432 was the error reporting functionality of the device. Faults that result in a shutdown of the motor driver are signaled by the FAULT pin going low. The faults that can result in a driver shutdown are over temperature, over current, and under voltage. The other fault pin is the over temperature warning (OTW) pin which goes low when the junction temperature reaches 125°C. Along with these two error reporting pins, the DRV8432's internal protections will immediately set the half bridge outputs to a high impedance state when a FAULT occurs. As well as reporting the error to the FAULT pin, the device is able to recover from some faults automatically. The over current protection of the DRV8432 has a programmable trip threshold on all high side and low side FETs. In CBC mode the detector outputs are monitored by two protection systems. One protection system controls the power stage to prevent the output current from increasing further by limiting the current in each and every cycle preventing a motor driver shut down. The second protection system would be used during a short to ground or short to power condition. In either of these conditions, the CBC limiting circuit might not be able to maintain the current at an appropriate level. This second protection system triggers a latching shutdown and sets the half bridge in a high impedance state. Each half bridge has its own independent current limiting and over current protection. The DRV8432 has multiple thermal sensors on different areas of the device to monitor its temperature. The over temperature protection also has two stages. The first stage is an over temperature warning, and the motor driver will set the over temperature warning pin low when the junction temperature rises above 125°C. The second stage of the thermal protection in the DRV8432 is the thermal

shut down mode. If the junction temperature rises above 150°C the motor driver sets all the half bridge outputs to a high impedance state and sets the FAULT and OTW pins low. Some faults are automatically recoverable while others require the micro-controller to assert the reset lines to low. In the event that a fault condition is reported, the simulation will be aborted.

VIII. LOW VOLTAGE POWER SUPPLY

The second power supply provides the low voltage specifically needed by the control circuitry. This supply needs to be capable of providing the appropriate voltages to the user interface, the micro-controller, motor driver, cooling fans and over-travel sensors. After considering the voltage level requirements of each of the components we determined that the low voltage supply would need to be capable of providing 3.3V, 5.0V, and 12V. The LT1083 linear regulator from Linear Technology was used for the 12V supply, a MC7805CTG from On Semiconductor was used for the 5V supply and a UC78M33DCY from Texas Instruments was used for the 3.3V supply. The only difficulty in designing this control voltage supply was thermal considerations for the regulators. Due to the low noise requirements of the DRV8432, these regulators had to be linear regulators and the 12V regulator will have to supply up to 2.7A. As shown in (5), this regulator's junction temperature with a heat sink will rise to 138.3°C. Without a heat sink, the junction temperature exceeds the rated value of the device quite easily, so it is absolutely necessary to have a heat sink on this regulator.

$$\begin{aligned}
 V_{Drop} &= 6.4V \\
 I_{Out} &= 6.4V \\
 P_{Reg} &= 17.3W \\
 T_{Amb} &= 38^{\circ}C \\
 \theta_{Total} &= \theta_{Junc-Case} + \theta_{Case-HS} + \theta_{HS-Amb} = 5.8^{\circ}C/W \\
 T_{junc} &= T_{Amb} + \theta_{Total} \times P_{Reg} = 138.3^{\circ}C < 150^{\circ}C
 \end{aligned} \tag{5}$$

IX. HIGH VOLTAGE POWER SUPPLY

Each motor on the simulator consumes approximately 30 VA of power under full operation. Thus, the motor power supply must supply 60 VA, at minimum. An unregulated power supply was designed such that the back EMF from the motors would not damage the system. The supply is composed of two identical electrically isolated legs, to reduce the electrical noise between the motor driver circuits. Each leg is composed of a transformer,

bridge rectifier and a capacitor bank. The power is transferred to the motors via the motor drivers, which constrain the supply specifications. The motor drivers require a voltage level no higher than 50 VDC, nominally. Thus, the secondary voltage level on the transformers needed to have an RMS value of 35 VAC or less. Transformers with secondary voltages of 30 VAC were chosen, for the sake of convenience and to compensate for fluctuations in the line. This yields a rectified voltage of 42 VDC, which is adequate for the motor driver design. Under full operation each motor effectively draws 7.9 amps. Thus, we calculated the required capacitance for each leg of the supply as shown in (6), to maintain the voltage level on the supply within 10% of 42 VDC at 60Hz.

$$C = \frac{5 * I}{V_p * f} = 15,558 \mu F \quad (6)$$

This was realized by using four 3,900 μF capacitors in parallel for a total capacitance of 15,600 μF per leg. Our final design is capable of providing 150 VA of power. This allows a good safety margin and ensures that the supply will not overheat due to extensive use.

X. TRANSMISSION LINES

The transmission lines from the control system to the motors have been known to cause communication problems. Because the lines are left outdoors and are roughly treated they have incurred mechanical failures. These failures were fixed by our team such that the existing system and the new system can effectively control the simulator.

XI. USER INTERFACE

The control system will be located close to the rain simulator and exposed to the environment including the over-spray from the simulated rain. The user interface components that are being used are an LCD with a 4 line by 20-character display, a 4-row by 3-column keypad, and a display bezel for the LCD. With the need to display up to three pieces of information and on screen instructions this 4-line display is ideal. The rain intensity and the rain duration, which are the two pieces of input information needed to run the simulation, are being displayed. Along with these two input parameters, the users have requested that we also display the measured rainfall from the wireless rain sensors. With these three pieces of information, the users should be able to determine the condition of the current simulation. The PC2004-A display is back-lit with an LED and requires a nominal 5-volt

supply. Upon completion of a simulation, the display will show the set intensity, that the simulation is complete, and the measured rainfall.

The keypad that is needed for the input on the final product needs to be water resistant and needs to be rugged. The one that we have chosen to be suitable for our use is the 1K12T103. The 1K12T103 is a 3x4 matrix keypad with an Ingress Protection 65 rating. This rating means that the device is completely protected from dust intrusion and exposure to electrical contacts. The Ingress Protection level from water is sufficient to ensure reliable operation. The keypad has an operating voltage of up to 24 VDC and a max operating current of 50 milliamps.

The Storm-Interface 5000 Series display bezel is another impact resistant device with similar IP ratings to the 1K12T103 Keypad. The bezel data sheet also states that this device is weather sealed for outdoor use with a scratch resistant, anti-reflective window. As well as being protected from the environment, the bezel is resistant to most cleaning chemicals. The bezel is designed for an operating temperature range of -20°C (dry) to 60°C which is well within the expected environmental conditions of the Rain Simulator. The 4x20 character LCD is one of the displays that bezel was designed for. The three integrated buttons on the face of the bezel are in the perfect location to be used as Start, Pause, and Reset buttons. These three buttons, along with the 3x4 keypad will provide all the input necessary for the project.

XII. WIRELESS RAIN GAUGE

The wireless rain sensor is best described as a sophisticated integration between a manually read rain gauge, electronics, and a wireless antenna to convert it into a fully-automated rain gauge. This rain gauge has been designed to hang from the main horizontal beam of the nozzle support system. Due to the variation in the incline that the rain bed will be subjected to, we initially considered constructing a self-balancing base to hold the rain gauge. However, due to the large variation in materials tested on the rain bed, the rain gauge would have had to be able to attach to all of the different types of materials ranging from hard concrete to porous soil equally well. The group would have had to construct different mounts for the base of the rain gauge that would be changed depending on the type of material being tested. To avoid this additional effort, the rain gauge will be hung very low from the horizontal beam to the point where it almost reaches the rain bed but far enough to not be touching the rain bed when put at an incline. This design will not need a base support system and will automatically

always be held in the appropriate direction to catch and measure the incoming rainfall.

The rain gauge was also designed to surpass the rainfall measurement accuracy of 0.25 inches. From 0 to 1.4 inches, the gauge will measure to 0.1-inch accuracy and anything above 1.4 inches will be measured to an accuracy of 0.2 inches. The water level sensing system simply consists of 32 bare conductors that are submerged into the water with an open circuit in between that is later bridged by the rising water. Each conductor resides at a specific water level and will only conduct current when the water reaches that specific level.

Thus by applying a small voltage of 3.3VDC to one side of the conductors, we will only see this small voltage when the water has reached this discrete level. Through the use of two 16:1 multiplexers, we were able to connect and control the voltage through all 32 conductors. A small Texas Instruments MSP430-G2553 microcontroller has been programmed to scan all 32 conductors and determine how many are under water. For every high conductor that the microcontroller sees, it will increment a counter. Once all 32 conductors have been cycled through, the final number on the counter will denote the amount of high lines read. This amount of high lines will be the amount of water levels reached and from this we can convert to the reading of the water level in inches. This final reading in inches is then sent to the XBEE antenna which in turn sends the signal to another XBEE module located on the MCU PCB. This receiving XBEE then sends the signal to the MCU where it is then communicated to the end user via the LCD screen on the outside of the enclosure.

Due to the fact that rain gauge is a stand-alone system, it needs its own power supply. To power the rain gauge we are using two AA batteries each of which are 1.5VDC. The two multiplexers need a minimum of 3.1 volts to function and the microcontroller will only take a maximum of 3.3 volts so we also integrated a 3.3VDC boost converter to ensure that our voltage level is always at the necessary magnitude to power these two crucial components. Additionally, an LED was integrated to the system to give a visual notification of when the system transmits data. All of the electronics will be stored in a small water-resistant box attached to the water-catcher. In order to turn the rain gauge on and off, a waterproof switch will be mounted to the outside of the small plastic box making it easily accessible for the researchers to turn on the rain gauge to prepare for a simulation and turn it off in between simulations. The researchers will be responsible for keeping track of the battery life of the rain gauge and changing them when needed. All in all, the rain gauge will be a stand-alone system, which will take a water level and convert it to a signal that will ultimately be

communicated to the researcher through the LCD on the main enclosure. Figure 5 is a depiction of the components of the rainfall sensor system.

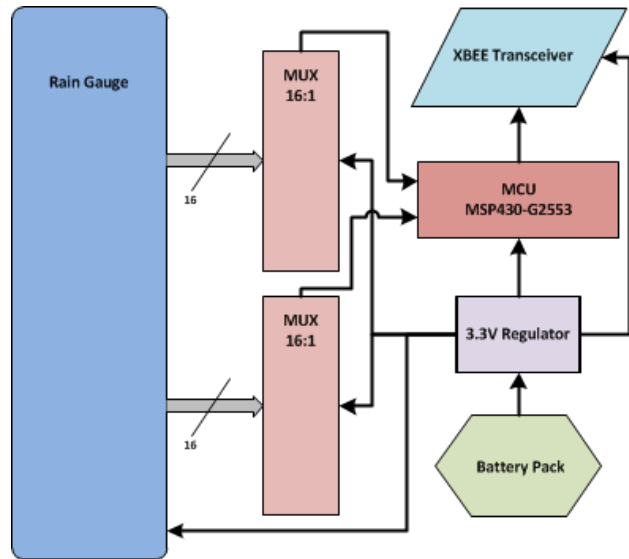


Fig. 5 Rain Gauge System Block Diagram

XIII. SYSTEM ENCLOSURE

The entire control system needed to be housed in an enclosure conducive to the rainfall simulator environment. Thus, it needed to be rugged, water resistant and portable. Sunshine Metals, a metal shop in Altamonte, graciously donated a custom built stainless steel enclosure to our project, per our specifications. The stainless steel enclosure provides an impressive look, undoubtedly rugged structure, water resistance and it will not corrode after being subject to many years of use. The whole system weighs in at about 63 pounds; thus, it remains portable. Figure 6 is a depiction of the enclosure.



Fig. 6 Rainfall Simulator Control System Enclosure

XIV. CONCLUSION

Throughout the past year, our group has managed to successfully redesign and build the entire rain simulator electrical control system. The division of the complete system into many sub-assemblies definitely aided in the victorious completion of this project. Due to the thorough testing and troubleshooting done after the construction of each sub-assembly, we were able to ensure that each sub-assembly would communicate and work with another correctly. We hope that this modularized design will lengthen the life of our system as well as help the UCF Stormwater Research lab with any future maintenance that may need to be performed.

XV. BIOGRAPHIES



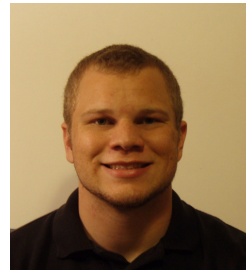
Adam Cutting is an Electrical Engineering student at UCF. Adam participated in the co-op program at United Space Alliance at Kennedy Space Center. His projects at the space coast included a three-phase portable solar power system and other energy saving projects. Adam's focus in the senior design project is the user interface and its integration to the rest of the system. He has also done extensive research on the motor drivers and their specific application to the rain simulator.



Luke Falls is an electrical engineering major with experience in the area of air conditioning and refrigeration systems. During his junior year at UCF Luke began a co-op at United Space Alliance at the Kennedy Space Center. At his co-op Luke worked on projects like power systems planning and layout. Luke's main focus in the senior design project has been in the area of microcontrollers and their various characteristics and applications to the rain simulator. Additionally, Luke has also taken the major task in putting together the smaller sub-systems into one central communicating control system, much like a systems engineer.



Brenda Garcia is an Electrical Engineering major at UCF. Brenda has participated in a co-op experience at Lockheed Martin where she worked with the turreted systems program. She is currently working at Siemens Energy in generator supply chain management. Brenda's focus in this senior design project has been in the area of the wireless integration of the rain gauges and various administrative tasks.



David Levy is an Electrical Engineering major at UCF. David has participated in a co-op at the United Space Alliance at Kennedy Space Center. David's main focus in the senior design project has been the stepper motors and the power supply design. David has successfully dissected the inner workings of the stepper motor and applied their functionalities to the specific application of the rain simulator. Additionally, David has done extensive research and design of the power supply needed for the control system.

ACKNOWLEDGEMENTS

The group would like to acknowledge support given by Erik Stuart and Mike Hardin on behalf of the UCF Stormwater Research Lab. We would also like to thank Storm Interface, Sunshine Metals, Texas Instruments and Cadsoft for their gracious donations in the form of the display bezel and keypad, the system enclosure, and various surface mount electronic components as well as development tools, respectively.

REFERENCES

- [1]David Austin, "Generate stepper-motor speed profiles in real time," *www.eetimes.com*, November, 2011
- [2]Texas Instruments, DRV8432 Data Sheet, *www.ti.com*, July 2011