Water Saver

Mina Samaan, Evan Glazer, Muhammad Siddiqui, and Mohammed Bazrbachi

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

***Abstract* — This project was successfully completed by three electrical engineers and a computer engineer who dedicated their time and effort to engineer Water Saver. This system aims to conserve rainwater for various user needs, both now and in the future. While the Water Saver project can be implemented on a larger scale, our demonstration showcases it on a smaller scale for demonstration purposes.**

I. INTRODUCTION

Year 2030, with the latest catastrophe in our world where water is a limited resource in the United States and not maintained by government or local ordinances due to new virus taking over the world. Water Saver is here to help residential homes collect and store rainwater for all uses of life. Water Saver is a B2B Business that requires residential or commercial installation using our state-of-the-art design. Additionally, to our hardware and product design, our modern software suite can be used to track and monitor your water system through the next years while water is a scarce resource.

Our approach is dedicated to focusing on the features and technology aspects of the project to meet our senior design goals and apply what we have learned in electrical and computer engineering to produce this business. As you can see on the right, is our water collection system that would be installed on a house, but for our project demo we're focused on a smaller design of the application using fish water tanks to show how collection of water and the system can be automated.

II. PROJECT COMPONENTS

This section will clearly explain the purpose of each component we are using in our project. It aims to make it easy to understand how our system works and to showcase its features.

A. ESP32 wroom Microcontroller:

The ESP32-WROOM microcontroller is an ideal choice for our water management system due to its integrated WiFi and Bluetooth capabilities, which facilitate seamless internet connectivity and HTTP request handling. Its numerous GPIO pins accommodate various sensors and devices, including two distance sensors, a rain sensor, battery level reading, temperature sensors, and relays. The dual-core CPU and higher clock speed allow for concurrent task management, essential for sensor data acquisition, processing, and network communication. Additionally, its low power consumption and power-saving modes extend battery life, making it suitable for remote or off-grid installations. The ESP32-WROOM's ample RAM and flash memory support running multiple libraries, such as those for DHT sensors, OneWire, DallasTemperature, NTPClient, WiFi, HTTPClient, and ArduinoJson, without memory constraints. Compatibility with the Arduino IDE simplifies development, while the robust community and extensive documentation provide valuable support and resources. By leveraging these features, our system can efficiently manage sensors, process real-time data, and communicate effectively with our web server, ensuring a robust and scalable solution for water management and sustainability.

B. Relay:

Using relays in our system provides an efficient and reliable method to control high-power devices like fans based on sensor inputs. The relays offer robust switching capabilities, allowing us to handle the varying loads without compromising the microcontroller's safety. This integration ensures that the system operates seamlessly under different conditions, such as temperature fluctuations detected by the DHT11 sensor, by activating the fans when temperatures exceed 70 degrees Fahrenheit. Additionally, the relays' ability to be controlled through GPIO pins in our code facilitates precise and immediate response to environmental changes, ensuring optimal performance and energy efficiency in our water management system.

C. 20x4 LCD display:

The 20x4 LCD is an ideal choice for our system and code due to its superior display capacity and readability. It allows us to present more information simultaneously, which is critical for monitoring multiple sensors and system states. With four lines of text, we can display distance measurements, temperature readings, rain sensor data, battery voltage, and real-time clock updates without constant screen refreshing or data overwriting. This enhances the user's ability to quickly assess system performance and status briefly, making the interface more intuitive and efficient for real-time monitoring and management tasks. Additionally, the LCD screen serves as a reliable source of information when users do not have access to their phones. While the primary mode of accessing real-time data is through the app, which offers convenience and remote monitoring, the LCD screen ensures that users can still obtain accurate measurements and system status directly from the device itself. This dualmode accessibility is particularly beneficial in situations where mobile device access is limited or impractical. The integration of the LCD screen with our system ensures continuous visibility of crucial data, thereby enhancing the overall user experience and system reliability. This setup guarantees that users have immediate and reliable access to critical information, whether they are near the device or monitoring it remotely through the app.

D. Ultrasonic sensors:

Ultrasonic sensors are optimal for our water-saving system due to their precise and reliable distance measurement capabilities, which are crucial for accurately monitoring water levels. These sensors emit ultrasonic waves that bounce back upon hitting an object, allowing the system to calculate the distance based on the time taken for the echo to return. In the provided code, two ultrasonic sensors are set up using trigger and echo pins (trigPin, echoPin, newTrigPin, newEchoPin) to measure distances. The code includes functions to get distance readings, averages multiple readings for accuracy, and uses these measurements to control the relay and other components, ensuring efficient water management and system responsiveness. This water level sensor is particularly well-suited for our project because of its compact size. We needed a sensor that could fit discreetly in a space where the wires would remain hidden, contributing to a more professional look. The sensor fits perfectly within a 3D-printed mold that integrates seamlessly into our design. This setup prevents any disturbances to the ultrasonic waves, ensuring accurate detection when the water level approaches the top. The sensor was specifically chosen for its compatibility with our mold, which is implemented inside the fish tank to stop waves from interfering with the measurements. This precise and reliable sensor significantly enhances our system's ability to manage water levels effectively.

E. Water pump:

Widely adopted brushless DC water pumps maintain operating voltages between 9V and 24V, well below the safety threshold of 36V for human safety. This substantial reduction in voltage significantly reduces the risk of electric shock incidents. In regions with erratic voltage, stability is ensured through power adapters (transformers) that provide consistent power and the option to rely on batteries during power outages. Brushless DC pumps rectify issues of low energy efficiency, bulkiness, and noise. brushless DC water pumps are the best option making them capable of enduring high pressures. The water pump integrates seamlessly with our system and code due to its relay-based control, which allows for precise activation in response to specific environmental conditions detected by various sensors. The system monitors rain levels, distances from ultrasonic sensors, and other environmental factors. The relay connected to the pump (pin 23) is activated or deactivated based on conditions like rain detection and distance readings from the ultrasonic sensors. When rain is detected, the system intelligently manages the pump to prevent water wastage. This setup ensures efficient water management and conservation, driven by real-time sensor data processed in the code.

F. DHT11 temperature sensor:

The DHT11 sensor is optimal for our system and code due to its simplicity, reliability, and low-cost temperature and humidity measurements, making it well-suited for monitoring the PCB box environment. The sensor is set up using the DHT library and connected to pin 25, with initialization in the setup function through dht.begin(). During operation, the sensor reads temperature in Celsius using dht.readTemperature(), and the value is converted to Fahrenheit for display and control purposes. The DHT11 sensor integrates seamlessly with our other components and provides essential environmental data for system functionality. We initially chose these fans because we did not anticipate our components being at risk of overheating, and these fans can operate at just three volts. Multiple tests showed that this setup did not affect the motor in any way. However, we observed that when we increased the motor's power, the fans also sped up significantly, putting them at risk of burning out. To address this issue, we switched from the Raspberry Pi Pico to the ESP32 microcontroller. The ESP32 operates at 3.3 volts, allowing us to manage the board's temperature more effectively. This change ensures that our fans run efficiently without overheating, keeping our system safe and maintaining optimal performance.

G. Rain sensor detector:

The rain sensor detector integrates seamlessly with our system due to its ability to provide real-time rain detection, enabling automated control of the relay and other connected devices based on weather conditions. In the setup, the rain sensor is connected to pin 34, where it continuously monitors moisture levels. The analogRead() function retrieves the sensor data, and the system logic checks if the rain value exceeds a threshold (e.g., 1000). If rain is detected, the system updates the relay state to manage connected devices, accordingly, enhancing the system's responsiveness to environmental changes. Additionally, the sensor data is transmitted to a server for remote monitoring and analysis. This rain sensor will continuously monitor the weather, and as soon as it starts raining, it will trigger the relay to turn on, which in turn activates the motor to fill up the storage for the user to utilize as needed. We initially considered using a weather app to pull information for the microcontroller to predict rain, but in Florida, the weather can be highly unpredictable, making such predictions unreliable. Due to this bipolar weather, we opted for a rain detector. The sensor operates after being soaked in water for 30 seconds, ensuring accurate detection before triggering the relay to turn on the motor. This integration ensures that the entire project works together seamlessly, efficiently managing water based on real-time conditions.

H. PCB fans and heat sink:

The PCB fans and heat sink are integral to our system, providing essential cooling to prevent overheating of components during operation. Our PCB faces significant thermal challenges due to the components we've chosen, which are not very “thermally” efficient. After consulting with our professor, we determined that a heat sink and two fans are essential for cooling our board. The fans blow air through the heat sink, effectively lowering the temperature. While our power board can operate at temperatures up to the mid-80s, it becomes dangerously hot when temperatures reach 100°F or higher. During testing, we observed our board overheated, forcing us to halt the demonstration. We haven't determined the maximum operating temperature, but we plan to conduct further testing after the demo to identify this critical limit. Therefore, we are using a temperature sensor to monitor the internal temperature of the box. Two fans are connected to the microcontroller via specific pins and are programmed to activate only when the temperature reaches a dangerous level. Once activated, the fans work to lower the temperature, effectively cooling the system and preventing overheating. This setup ensures that the fans operate only when necessary, maintaining a safe operating temperature for our components. In our code, the fans are controlled by pins 26 and 27, and they activate when the temperature exceeds a specified threshold (70°F) detected by the DHT11 sensor. This setup ensures efficient thermal management, maintaining optimal performance and longevity of the electronics. The fans are triggered by reading the temperature and sending the appropriate signals to the designated pins, automatically adjusting the cooling based on real-time temperature data.

I. DS18B20 Water temperature sensor:

The DS18B20 water temperature sensor is ideal for our system due to its precision, digital output, and ease of integration with the OneWire protocol, which simplifies connectivity by requiring only one data pin. In our setup, the DS18B20 is connected to pin 32, and its readings are processed using the DallasTemperature library. The code initializes the sensor in the setup() function and periodically requests temperature updates in the loop(), converting the values to Fahrenheit for display on an LCD and logging. This ensures reliable and accurate water temperature monitoring, critical for the system's functionality.

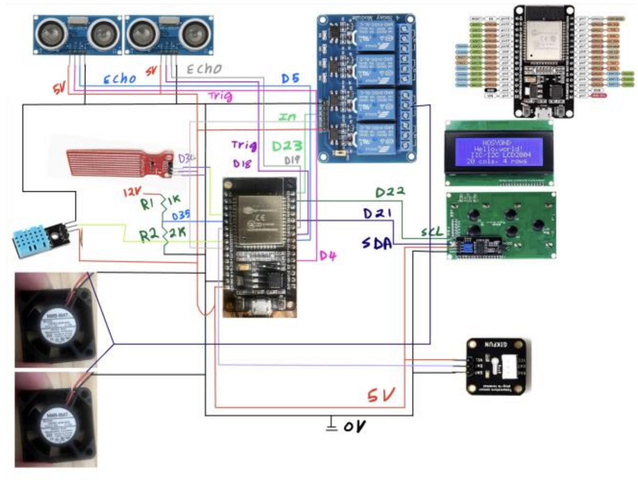


Fig. 1

The system diagram of the project has different components as you can see in the diagram above

III. SYSTEM SETUP

In this section, we are showing how our system works. We built our Arduino code based on this diagram.

A. Embedded system in our design:

Our system starts working either when the rainwater sensor detects rain or when a user activates the system; otherwise, it remains off. We have two water tanks: one at the bottom to collect rainwater from the house roof via a gutter and tube, which passes through a water filter to remove large objects like leaves. The water is then transferred to the top tank until the water level sensor detects that the tank is full, at which point the system stops sending water until the user activates it again. The reason for placing one tank on top is to allow gravity to distribute the water immediately to the user. When both tanks are full, the overflow water is diverted away from the bottom tank. To make our project more efficient, we decided to use solar power and a 12V battery to power the system.

B. Embedded Engineering:

The code initializes various libraries and hardware configurations, connects to WiFi using provided credentials, and sets up an NTP client for time synchronization. It defines pin configurations for sensors and actuators, including ultrasonic sensors, a rain sensor, a temperature sensor, and a relay. The setup function initializes serial communication, connects to WiFi, initializes the LCD, sets pin modes, and begins sensor operations. It creates a queue for sensor data and a task to send HTTP POST requests with sensor data to a specified server URL. The loop function updates the time, reads and processes data from the sensors, displays the data on the

LCD, and controls the relay based on specific conditions involving rain detection and distance measurements. It also debounces the relay state to prevent rapid switching. Sensor data includes distance measurements from two ultrasonic sensors (one using averaging), rain sensor values, battery voltage, and temperatures from the DHT11 and DS18B20 sensors, with the results displayed on both the LCD and the serial monitor. Construction delves into the intricate process of bringing theoretical designs to life through the creation of functional prototypes. It outlines the systematic approach undertaken to finalize the design, source and select components, and meticulously assemble.

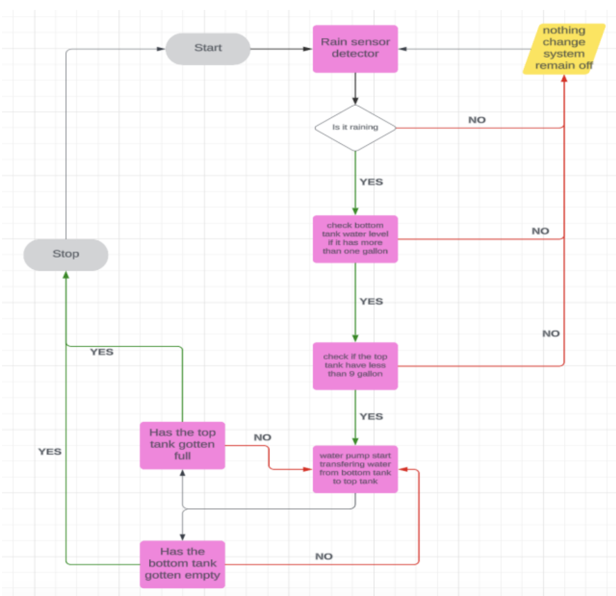


Fig. 2 Code setup functionality

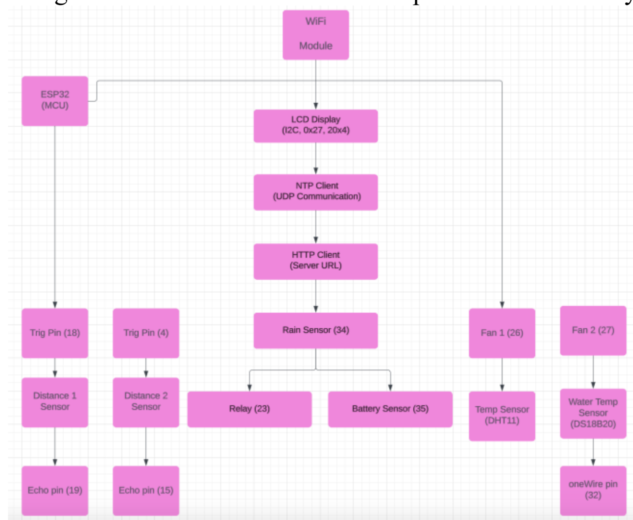


Fig. 3 Hardware integration block diagram

III. SOFTWARE DESIGN

A. Embedded Platform Functionality (refer to Fig. 1)

The Water Saver System is a secure and efficient platform for monitoring and managing water usage. It offers a user-friendly dashboard, real-time data insights, and robust security measures. Key components include user interaction, API services, password protection, testing and provisioning, backend infrastructure, traffic management, database management, data storage, frontend development, and monitoring and control software. The system is designed to meet the needs of both individual users and enterprises. The backend is hosted on AWS EC2 instance, containerized for efficient deployment and scalability, and Nginx is employed for traffic and bandwidth optimization. The front end is developed using React for a responsive and dynamic user interface. Route 53 is used for reliable DNS management.

Real-time insights are delivered to help users optimize their water consumption effectively. The embedded system software is designed to perform several crucial functions. It continuously acquires data from all connected sensors, including ultrasonic sensors, the DHT11 sensor, the rain sensor, and the DS18B20 water temperature sensor. This data is then processed to extract meaningful information, such as current water levels, temperature, and humidity percentages. Based on the processed data, the system employs control logic to make decisions, such as activating the water pump if the water level is low or turning on the fans if the ESP32 is overheating. The processed data and system status are displayed on the LCD, providing real-time information for user monitoring. Additionally, the system can be configured to upload sensor data to a server, enabling remote monitoring and logging. This integrated approach ensures that the system operates efficiently and that users have access to up-to-date information about the system's status.

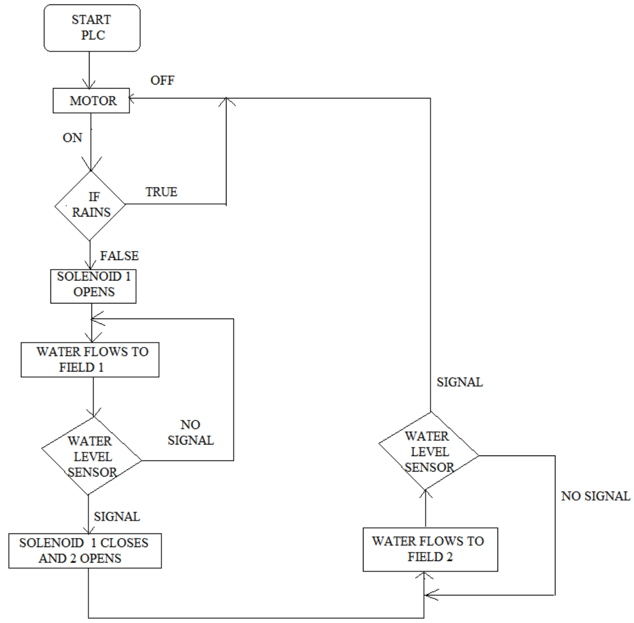
IV. SYSTEM IMPLEMENTATION

A. System Performance

The rainwater harvesting system effectively collects and stores water, achieving conservation goals despite challenges like unpredictable weather and component integration, which were addressed through rigorous testing and reliable sensor use. The system offers significant environmental and economic benefits by reducing reliance on municipal water and lowering costs, and it is adaptable for various contexts, including households and developing communities, where it can improve water availability and quality. This innovative approach supports sustainable water management and addresses global water scarcity.

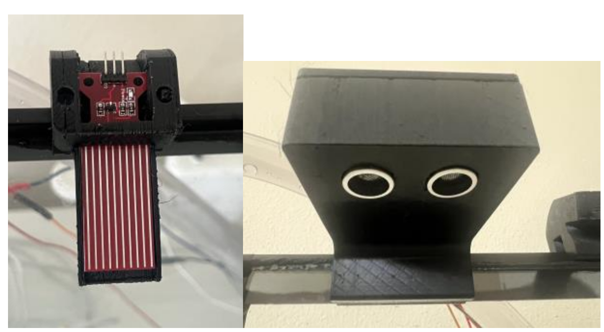
V. FIGURES

The system demonstrated efficiency in collecting, filtering, and storing rainwater, offering significant environmental and economic benefits. However, limitations such as unpredictable weather and component integration posed challenges, which were addressed through rigorous testing and reliable sensor use. Future improvements could focus on enhancing system scalability and incorporating advanced filtration techniques. Rainwater harvesting remains crucial for sustainable water management, particularly in regions facing water scarcity. Our implementation includes installing a gutter system along the roof to capture rainwater efficiently, using a First Flush Diverter to ensure clean water enters the storage tanks by diverting initial contaminants. This project underscores the importance of innovative solutions in addressing global water challenges.

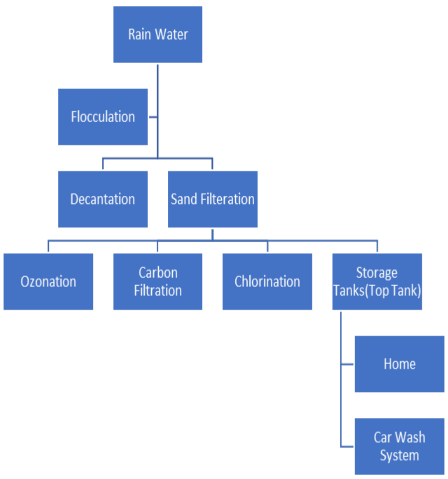


Choosing the perfect storage tanks is key. We'll select durable, non-toxic, and corrosion-resistant tanks with capacities tailored to household size and water usage needs. To ensure top-notch water quality, a two-stage filtration process will be implemented. First, pre-filters or mesh screens will catch larger debris before the water enters the storage tanks. Then, more sophisticated filtration systems, like sand filters or cartridge filters, will ensure the water is crystal clear and safe for various uses.

A network of pipes will be strategically placed to deliver rainwater to designated areas such as irrigation, laundry, toilets, and car washing. To maximize efficiency and control the flow, control valves will be incorporated at key points within the system. This setup ensures effective and efficient water distribution, promoting sustainable water management.

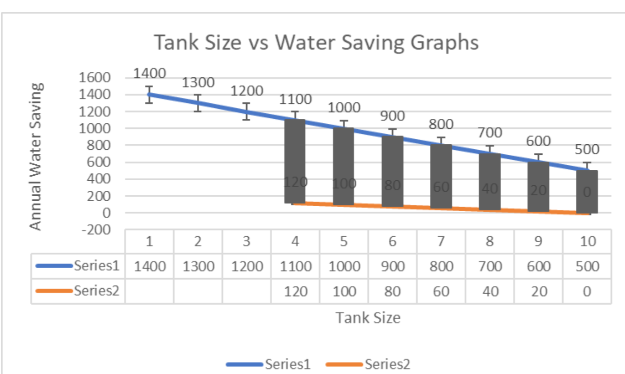


For accurate and reliable liquid level measurement, especially in sensitive environments like food production and chemical processing, choosing the right sensor technology is crucial. Capacitive sensors offer a non-invasive solution using changes in capacitance to detect liquid levels through tank walls. This eliminates contamination risks and simplifies installation, making them ideal for such applications.



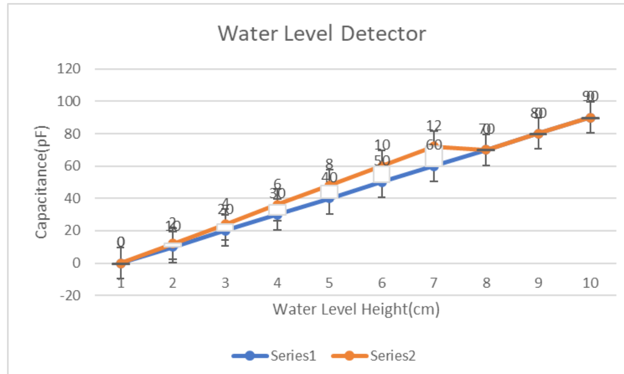
To distribute the collected rainwater effectively, we'll install a water pump sized specifically for our system's needs. This pump will be the workhorse, ensuring a smooth flow of water throughout the system.

This graph shows the height or percentage of water level on the y-axis and the remaining gallons of water on the x-axis. This would be a straight line with a slope of -1, starting at (100%, 10 gallons) and ending at (0%, 0 gallons).

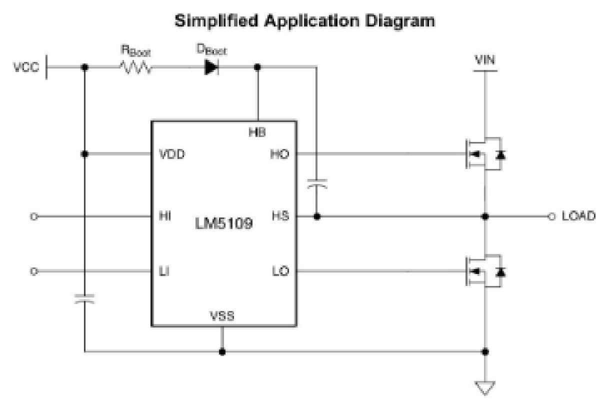


The relationship between water level height and capacitance in a capacitive water level sensor is not perfectly linear, but it can be approximated with a graph. Here's a breakdown:

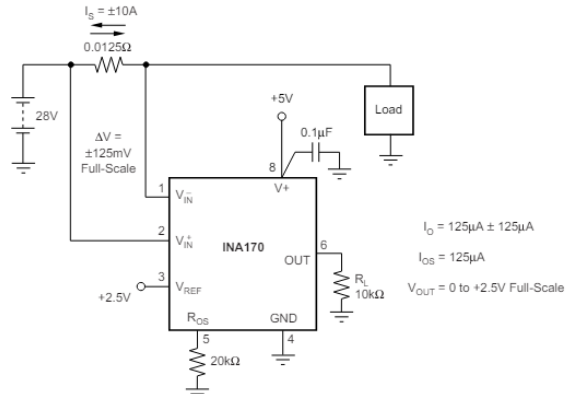
The graph would generally show an increasing capacitance as the water level rises. This is because water has a higher dielectric constant compared to air. As the water level increases, the sensor plates come closer to a material with a higher dielectric constant, leading to a higher capacitance.



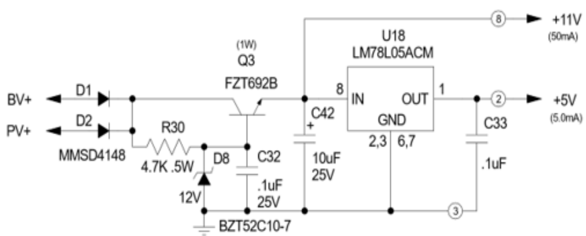
This graph shows the time elapsed on the x-axis and the gallons of water used on the y-axis. This would depend on the rate at which water is being used from the tank. It could be a linear graph if the water usage is constant (e.g., a faucet left open), or a curved line if the water usage is variable (e.g., filling a container, taking showers).



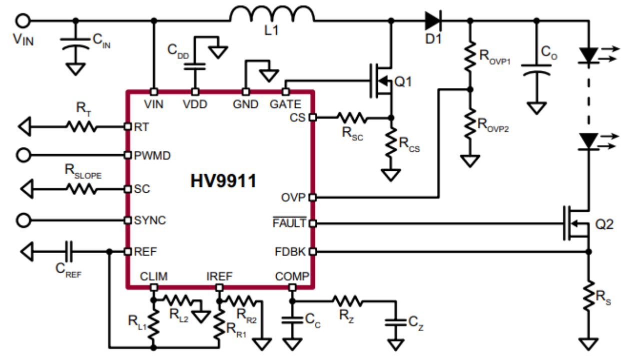
We are using an IC called LM5109A in our circuit. The setup includes two MOSFETs and a PV for our solar panel, with a capacitor connected to stabilize the PV voltage. The purpose of these two MOSFETs is to prevent the PV from directly charging the battery until certain conditions are met. These MOSFETs are controlled by the LM5109A IC. We supply 11V to the VDD pin of the IC to power it. The 'CHARGON' signal goes into the IC; when 'CHARGON' is high, it drives the 'HO' (high-side output) high as well. This high output at terminal 7 is mirrored at the gates of the two MOSFETs, causing them to shorten. When the MOSFETs are short, voltage from the PV is blocked from flowing directly to the battery. Consequently, current is diverted to charge the battery. This precise control ensures efficient and safe charging of the battery by regulating the flow of current and voltage through the system.



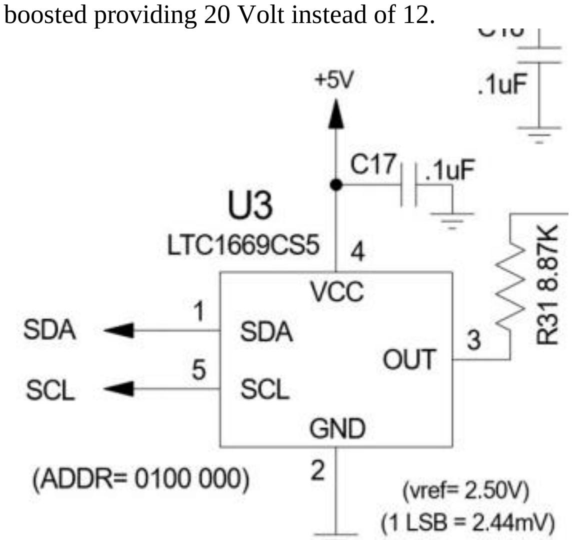
We also utilize the High-Side, CURRENT SHUNT MONITOR INA170 current sensor. This IC monitors the voltage difference between inputs IN+ and IN-. When there's a significant voltage difference, the INA170 produces an output voltage signal. If IN+ exceeds IN-, indicating that the PV voltage is higher than the battery voltage, the output signal goes high. This high signal instructs the system to proceed with charging the battery. This operation is powered by our 11-volt supply, ensuring precise control over the charging process based on voltage differentials detected by the INA170. This approach ensures efficient and reliable battery charging, optimizing energy management within the system.



In our setup, we have two inputs, Bv+ and Pv+, feeding into a voltage regulator. A transistor controls current flow into the regulator, while a zener diode and resistor combination stabilizes the voltage at or below 12 volts between the output and ground. This setup ensures that even with inputs like a 24-volt battery or higher from the PV array, the voltage remains regulated. A capacitor smooths out any voltage fluctuations. The transistor's base-to-emitter voltage drop maintains the output around 11.3 volts, powering the initial IC reliably at 11 volts. The remaining voltage powers another regulator, which outputs a stable 5 volts for the microcontroller, ensuring consistent and safe operation throughout the system.



We are using Boost Converter LED Drivers Using the Supertex HV9911. This boost converter circuit takes input from the battery and uses it to increase voltage. A capacitor stabilizes voltage, while a current sampling mechanism protects against excessive current. The main section features an inductor that stores energy as a magnetic field when current flows through it. When the gate signal goes high, it activates a MOSFET to create a path for current to ground through resistors, limiting current flow. This action allows the inductor to store energy. When the gate signal goes low, the circuit closes, causing the inductor to release stored energy, boosting the voltage from 12V to 20V efficiently. We're also utilizing the LED's that was provided by the data sheet to use to indicate when the circuit is being boosted providing 20 Volt instead of 12.



The LTC1669CS5 is a 10-bit digital-to-analog converter (DAC) with a low operating current, designed for micropower applications. It uses an I2C-compatible interface for digital control. The digital input and the reference voltage (VREF) determine the output voltage determined by the digital input and the reference voltage (VREF). When using the LTC1669CS5, you can program the current and voltage output by adjusting the digital inputs according to the datasheet specifications. The reference voltage is provided internally or can be an external source, and voltage dividers can be used for precise control of output voltages. The DAC outputs-controlled voltages as determined by the machine controller, ensuring flexibility and precision in various applications. We required a device capable of dynamically adjusting voltage and current while ensuring operational safety. Therefore, we opted for this integrated circuit (IC) due to its advanced features.

VI. CONCLUSION

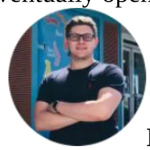
This project served as a unifying force, fostering collaboration among all team members on a singular initiative. It provided a platform not only to apply the knowledge acquired through our electrical and computer engineering degrees but also to refine our team-working skills. Unlike individual assignments that assess knowledge in isolation, this project mirrored real-world scenarios where teamwork is paramount for accomplishing tasks. Through this collaborative effort, we honed our teamwork skills, gaining valuable insights into time management and the importance of regular communication to ensure alignment, minimize errors, and meet deadlines effectively. Furthermore, teamwork enriched the final project, incorporating diverse perspectives from each member's background, resulting in the multifaceted Water Saver project.

VI: ACKNOWLEDGEMENT

We offer our thanks and gratitude to the professors who have so kindly agreed to review our project we have worked so hard on. Dr. Weeks and Dr. Wei had large impacts on our architecture and the successes of the project to produce Water Saver demonstrations.

VII: BIOGRAPHY

Mina Samaan, 27 years old, is graduating in Electrical Engineering summer of 2024. He owns an luxury transportation company and is looking forward to gaining experience in thd field after graduating to eventually open my own electronics company.



Evan Glazer, is a 29-year-old graduating in Computer Engineering summer of 2024. He has a job working in the industry as a Software Engineer specializing in architecting big data, algorithmic and complete solutions.

Mohammed Bazrbachi is an electrical engineer who is graduating in the summer of 2024 and is currently doing an internship at Cuitronics helping CEO's with development problems.

Muhammad Siddiqui is graduating bachelor's in electrical engineering in summer 2024. He is a doing internship with Kapsch helping them in changing the LED of highways signs and cameras of highways in Central Florida