12-2018

Smart Surge Irrigation Using Microcontroller Based Embedded Systems and Internet of Things

Prashant Dinkar Borhade
University of Arkansas, Fayetteville

Smart Surge Irrigation Using Microcontroller Based Embedded Systems and Internet of Things

A thesis submitted in partial fulfilment
of the requirements for the degree of
Master of Science in Computer Engineering

by

Prashant Dinkar Borhade
K. J. Somaiya Institute of Engineering and Information Technology
Bachelor of Science in Electronics and Telecommunication, 2010

December 2018
University of Arkansas

This thesis is approved for recommendation to the Graduate Council.

____
Christophe Bobda, Ph.D.
Thesis Director

____
Christopher G Henry, Ph.D.
Committee Member

____
David Andrews, Ph.D.
Committee Member

____
Dale R. Thompson, Ph.D.
Committee Member

Abstract

Surge Irrigation is a type of furrow irrigation and one of many efficient irrigation techniques. It is one of the economical techniques and requires minimum labor for monitoring it. In surge irrigation, water is applied intermittently to a field to achieve uniform distribution of water along the furrows, which is important while irrigating, as it ensures that there is enough water near the root zone of the crop. The uneven distribution can cause a loss in crop productivity.

Surge irrigation uses a surge valve, which is an electro-mechanical device that irrigates a field. The commercial surge valves available on the market are made to control only the time for the irrigation. However, their functionality is limited and requires human intervention to control and monitor the irrigation process. Therefore, monitoring the irrigation with these controllers is a big challenge. The lack of monitoring may result reduced irrigation efficiency.

The purpose of this thesis is to design and develop an embedded system for surge irrigation that resolves the drawback associated with the commercial surge valves. In this thesis, a “Surge Controller” is designed, implemented, and tested on the farm. The surge controller is a microcontroller-based embedded system, which runs the real-time operating system FreeRTOS on a single core ARM Cortex M3 microcontroller for multitasking. The important feature, which makes the surge controller “Smart”, is the Internet of Things (IoT) that enables the controller to send irrigation data over the Internet to a remote station.

During the Spring and Summer, 2017, the surge controller was developed and tested in the field at Rice Research and Extension Center of the University of Arkansas. Five irrigation events were run in a 20 acres soybean field. The controller was tested for the durability of the components in the environment and field conditions, performance and overall feasibility of the

device to achieve successful results from an irrigation event. After the successful testing, the IoT feature was added in Fall 2017 and Spring 2018 and tested for its functionality by running a few irrigation events in the Laboratory. The surge controller worked as expected continuously without interruption.

Acknowledgement

The development, implementation, and testing of this thesis was a lengthy process. There are the number of people who helped me in finishing the prototyping of this thesis. I want to take this opportunity to thanks all of them for their contribution.

First of all, I would like to thank Dr. Christophe Bobda and Dr. Christopher G Henry for providing me this opportunity. This project would not have been possible without their support and motivation. Their constant observation on my work and timely decisions have led this project to the correct direction. As a result, I did not waste time in working on a wrong path that would not have delayed the project. Their passion, dedication, and the way of managing the work-life balance have taught me a lot that will surely help me in my future.

Secondly, I would like to thank to the laboratory technical experts James Andress and Leland Schrader for taking care of the technical part of the project, and Dustin Pickelmann and Casey Seiber for helping me in setting up the surge system in the field. James took care of the electrical aspect of the system. His understanding of electrical and electronic devices helped me. He designed the power supply unit for the system. Leland took care of the fabrication of the mechanical valve. His precise work on the mechanical part of the valve is noticeable. Dustin and Casey helped me not only in setting up the surge irrigation system in the test field at the Rice Research Center, but also throughout out the irrigation season in calibrating the flow sensor, running irrigation events, and performing the necessary tasks at the end of the season. I want to thank all for providing me the support, because it would have been difficult to finish this thesis without them.

I specially want to thank my thesis committee members, Dr. David Andrews and Dr. Dale Thompson who have acknowledged my work and provided the approval on this thesis.

Their research area and the work have provided me a great motivation that has helped me in this thesis. I also want to thank the Computer Science and Computer Engineering department for providing me support.

Finally, I would like to thank my family and god for providing me the life, and the support. This thesis is important to me as it has opened the doors to the technologies where I want to build my career.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Overview

Agriculture plays a significant role in the process of economic development of a country. Increase in agricultural production leads to rise in per-capita income, and together with the urbanization and industrialization expedite the economic growth of the country [1]. The population of the world rose rapidly from around 3 billion in the 1960s to around 7 billion in 2010 and is expected to reach 9.3 billion by 2050 [2]. The worlds population growth is outpacing the world food production capacity. Thus, researchers and scientists are leaned towards using innovative techniques and modern technologies in agriculture to increase agricultural production. Irrigation, which is an important aspect of agriculture, is also undergoing these technological advancements. Today, various new irrigation techniques are found to be more effective than old techniques. This thesis study discusses the optimization of one of such techniques called surge irrigation and how the technology has implemented to automate the process of surge irrigation. This chapter introduces briefly to the idea behind the thesis and the technologies that have been used to optimize the surge irrigation. The following section discusses the need of management in irrigation and how does it help in increasing agricultural production. It also provides a brief introduction to the Internet of Things (IoT) technology and how it will assist in management of irrigation. The later section provides a brief overview of the thesis that includes how the system is designed and developed, which technologies are used in the project, and how the system was tested for its functioning. The final section of this chapter organizes and briefly explains the rest of the chapters in the thesis report.

1.2 Surge Irrigation

Irrigation in agriculture is the artificial application of water when cultivating lands to obtain higher production. Irrigation is the one that can influence the crop's growth process from seedbed preparation, germination, root growth, nutrient utilization, plant growth, yield and quality while using irrigation water efficiently. Irrigation uses the largest fraction of global fresh water, accounting for 70% of fresh water withdrawals [3]. In the United States, approximately 80% of available water goes to irrigation [4]. In China, about 65% and in India 90% of the fresh water is used for irrigation [5]. The increasing population requires more food and shelter, which can be fulfilled by expanding crop production with available resources. Consequently, the need for technology to improve irrigation management is useful.

The key to effective irrigation is to minimize the deep percolation of water (infiltrated water) past the crop root zone. Deep percolation losses depend on irrigation performance. Surface runoff is another issue which affects irrigation performance. Deep percolation, tail water and uneven application of water make irrigation inefficient [6], which also causes waste of not only water but also energy in terms of more energy required to pump water.

Properly managed irrigation results in high crop productivity and conservation of water. The management of irrigation can be characterized by factors, such as appropriate application depth of water suitable to the crop, irrigating the land for correct duration of time, minimizing the water loss at the end of the field, measuring the rate of flow of the water in the field, and referring to previous irrigation events to estimate the current event [7]. These factors can result in better irrigation performance. There are different types of irrigation systems such as surface irrigation, drip irrigation, sprinkler irrigation, have been invented and adopted to improve irrigation efficiency. However, the setup and the equipment required for implementing these irrigation techniques is expensive.

One of the more efficient irrigation techniques is surge irrigation. Surge irrigation is a type of surface furrow irrigation technique which replaces conventional continuous irrigation. In surge irrigation, water is applied intermittently by creating a series of ON and OFF cycles of water of variable time duration. This reduces the infiltration of water into the soil at the top end of the field, and the water advances faster to the bottom end of the field. As a result, surge irrigation requires less water than continuous flow irrigation and reduces total irrigation time [8] [9].

Surge irrigation uses a surge valve, which is an electro-mechanical device, for automating the process of surge irrigation. The commercial surge valves available on the market are designed to run a surge irrigation program of advance and soak phase by switching the direction of the water either to the left or to the right side of the valve. These devices are successful in automating surge irrigation and irrigators have been using them for a long time. However, their functionality is limited. The most important limitation is that these controllers do not provide a provision to save the irrigation flow data into the surge program. This data may be useful in increasing the irrigation efficiency. Another limitation is that these valves do not provide any provisioning for a remote monitoring. The irrigator must be physically present in the field for monitoring the irrigation event. Also, these valves are not automatic and need human intervention for starting and stopping the irrigation.

1.3 Internet of Things in Irrigation

A high percentage of this irrigation water may be wasted due to lack of supervision and a real-time monitoring [10]. This problem might be improved using Internet of Things (IoT). IoT provides Internet-based services and applications supported by electronic devices that are attached to physical objects for acquiring data from them [11]. This data can be analyzed and processed to provide a useful information or controlling the process. Internet connectivity began to grow in the 1990s. This growth was significant in consumer and enterprise market. In the 2000s, the Internet became the standard and today almost every industry uses Internet to access information through various applications, web sites, and software [12]. These application and websites can be considered the things on the Internet as they provide information; but they require more human interventions. The concept of IoT arose from the idea when some invisible technology operates from behind and dynamically responding to how the "thing" should act. This idea behind IoT was so successful that today, over 5 billion devices have been deployed, and according to predictions over 50 billion devices will be connected to the Internet by 2020 [12].

The IoT technology can be implemented in surge irrigation for remote monitoring irrigation events. While irrigating, the real-time data such as number of irrigation cycles completed, current number of irrigation cycle, total time for irrigation is running, time remaining for irrigation, number of gallons water applied, and water flow rate are an important information in order to monitor the irrigation. Using the IoT technology, an IoT device connected to a physical surge valve can send this data to a remote machine over the Internet. Using data visualization techniques, this data can be seen in presentable format through a mobile application or a website. As a result, the irrigator can access this real-time information from anywhere. Moreover, the collected information over the time can be analyzed to understand irrigation pattern for a particular area or for various crops.

1.4 Overview of Thesis

This thesis focuses on optimizing surge irrigation using an embedded system and the IoT technology. It consists of two parts. First is the development of an embedded surge controller system which automates the process of surge irrigation. The surge controller performs many tasks during an irrigation event. A real time operating system FreeRTOS installed on a single core microcontroller can provide the environment that enables multiple tasks to run simultaneously. In this thesis, a successful attempt was made that implements this hardware-software combination to achieve multitasking in surge irrigation. The second part of the thesis is the implementation of the Internet of Things for achieveing better monitoring in irrigation and Advance Encryption Standard (AES) technique for data communication security.

The embedded surge controller is a microcontroller-based embedded system that has been designed and implemented on a newly designed mechanical valve and tested by running a few irrigation events in one of the fields at the Rice Research and Extension Center of the University of Arkansas. The proposed surge controller is a semi-automatic device which can stop the irrigation depending on the amount of water that has been applied. It can also run in a normal mode for which the irrigator has to shut down the controller to stop the irrigation.

The surge controller is a “Smart” device, as it uses the latest technology, “Internet of Things” (IoT), to log the irrigation data over the Internet in order to achieve better monitoring in irrigation. It uses a General Packet Radio Service (GPRS) wireless communication protocol to transfer the data from the controller to a remote server. The IoT feature of the surge controller enables the controller to send the irrigation data, such as the mode of irrigation, the irrigation cycle number, the amount of water applied, the time remaining for irrigation, and the water flow rate to a remote server. A simple web application has been developed using Python Flask environment to display the irrigation data on the web. The wireless communication between the surge controller and the remote server has been secured with the Advance Encryption Standard (AES) technique. The IoT feature of the surge controller has been tested by running the controller in a laboratory. The IoT feature may be helpful in monitoring irrigation from a remote location, and therefore removing the dependency on the irrigator to be physically present.

1.5 Organization of Thesis

The remainder of the thesis is organized as follows:

- Chapter 2 - Surge Irrigation: This chapter discusses the concept of surge irrigation in detail. It also discusses the terminology and equipment used in surge irrigation. Finally, it discusses the limitations and the current methods of surge irrigation.
- Chapter 3 - Embedded Surge Controller: This chapter discusses the development process of the surge controller. The first section discusses the architecture and hardware and software implementation of the surge controller. The later sections discuss the operation, features, and limitations of the surge controller.
- Chapter 4: Internet of Things for Surge Irrigation: This chapter discusses the Internet of Things for surge irrigation. It first discusses the architecture and the implementation of IoT in the smart surge controller. It also discusses the AES technique applied for the secure data transfer to the remote server. Finally, it discusses the advantages and limitations of the IoT in surge irrigation.
- Chapter 5 – Results: This chapter discusses the results obtained from the irrigation events conducted using the new surge controller. The surge controller was installed in a field and few irrigation events were executed to see the performance of the surge controller. This chapter discusses each event in detail.

- Chapter 6 - Conclusion and Future Work: This chapter concludes the thesis by stating its importance and summarizing the implementation of the surge controller. Finally, this chapter discusses the future work that could help the surge controller to evolve into more advance system.

2 Surge Irrigation

2.1 Definition and Overview

The concept of surge irrigation was introduced by Stringham and Keller at the 1979 Irrigation and Drainage Specialty Conference of the American Society of Civil Engineers [13]. The surge irrigation is a type of furrow irrigation. In the surge irrigation, water is applied intermittently to the furrows in a field, creating a series of ON and OFF cycles. In surge irrigation, the field is divided into two halves, so while the first half is undergoing the ON cycle, the second half will undergo the OFF cycle and vice versa. Surge irrigation uses a surge valve placed at the center of a field and gated pipes are placed across the field which carry water to the left or the right side of a field. Figure 2.1 shows the pictorial view of the surge irrigation technique.

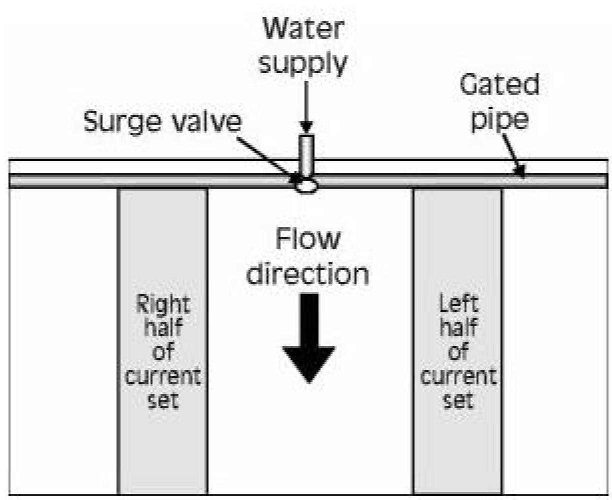

Figure 2-1. Typical Surge Irrigation System Installation. Source: *[14]*

During the OFF cycle, while draining the standing water in a furrow, the soil particles arrange themselves in a plate like structure and consolidate at the surface of the furrow. As a result, in a subsequent irrigation cycle, less water infiltrates into the soil and the water advances faster throughout the previously wetted furrow area. The reduction in infiltration occurs during the dewatering period which follows the initial wetting [15] [16]. It has been seen that surge irrigation requires less time to produce an equivalent reduction in infiltration as compared to the continuous irrigation. Also, the infiltration rate decreases to a steady state after waiting front has advanced. It also has been seen that, though the infiltration rate decreases both in surge irrigation and continuous irrigation during the season, the reduction is less in the surge irrigation than the continuous irrigation. Properly managed surge irrigation can reduce deep percolation losses and the water runoff at the end of furrows which in turn spreads water uniformly across the field. Figure 2.2 shows the infiltration profile for surge irrigation and continuous flow irrigation.

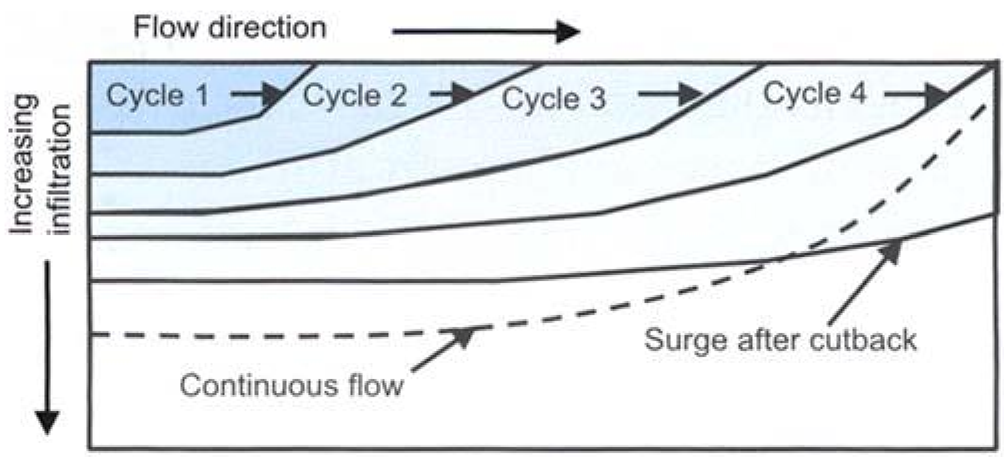

Figure 2-2. Infiltration Pattern for Surge Irrigation & Continuous Flow Irrigation. Source: *[17]*

Overall, surge irrigation can increase the irrigation efficiency and can offer improved water management in irrigation. However, this varies by soil type.

2.2 Surge Irrigation Terminology

To achieve greater efficiency and better management in irrigation using surge irrigation, it is required to understand the terminologies used in surge irrigation. The definitions used in surge irrigation have been taken from [18] with the author’s permission.

1. Advance Time: Time that is required for wetting front to “advance” from the crown to the end of the furrow.
2. Continuous flow: Irrigation flow in a furrow that does not stop from start to finish, before the required application depth is applied.
3. Recession Time: Time for the wave front to recede from the furrow. Essentially this is when the majority of the tail water has stopped draining from the field.
4. Opportunity Time: Time for water to infiltrate into the soil. The more opportunity time water has contact with the soil, the more volume is infiltrated.
5. Soak Time: Time after advance has completed where the remainder of the set time is used to meet the required application depth.
6. Application Depth: The depth of irrigation applied during a surge irrigation. This depth should be between 2.5 and 3.0 ac-in.
7. Number of cycles: The number of advance cycles (water on/water off) used to complete a surge advance program. Generally, surge advance times increase during the surge program, although some surge programs have a longer first advance than the second before increasing.
8. On-time: The time water is applied to one side.
9. Off-time: The time water is not applied to one side.
10. Cycle-time: The time required to complete an on/off cycle (sum of on-time and off-time).
11. Irrigation set time: The total irrigation time, this includes advance and soak times. The set time for row crops should always be less than 40 hours. If using a CHS plan, you must add the time for each set together to calculate the irrigation set time. For

example, if a surge is being used on two 24 hour set, the total time is 48 hours and the sets should be divided into three sets.

2.3 Surge Valve

2.3.1 Mechanical Valve

Surge irrigation uses a butterfly type of mechanical valve [19]. This valve is designed such that it diverts the water flow either to the left or to the right side of the valve. This valve is powered by batteries, air pumps, or solar cells. Figure 2.4 shows the general mechanical valve used in surge irrigation.

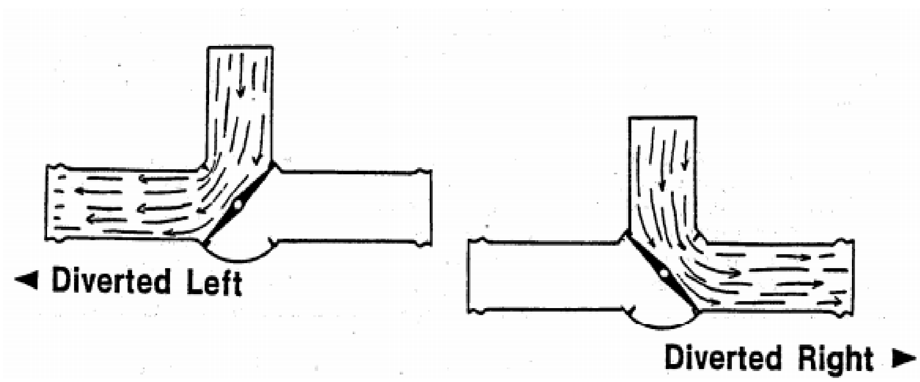

Figure 2-3. Butterfly mechanical valve for Surge Irrigation Source: *[19]*

2.3.2 Controller

Surge controller is an electronic device that monitors the time required for irrigation and switches the mechanical valve. The surge controller uses a microcontroller that implements surge program. The microcontroller is programmed to automatically decide the on and the off time for surge cycles. The surge controller is powered by a battery, solar cells, or both.

2.4 Present-day Surge Irrigation:

P&R Surge System Inc has been manufacturing a surge valve for 30 years. Their valve is T-shaped, cast aluminum that takes water at the inlet and delivers it either to the left or to the right port of the valve. This butterfly type valve activates through a motor fixed in the controller. This company makes 6 inch that delivers 2.65 m^3/min (700 gpm), 8 inch that delivers 4.54 m^3/min (1200 gpm), 10 inch that delivers 7.57 m^3/min (2000 gpm), and 12 inch that delivers 9.84 m^3/min (2600 gpm). They build three different controllers which can work with any surge valve they manufacture [20].

DamGates is another company that builds a system for surge irrigation called DamSurge valve. This valve is similar to the P&R surge valve; however, it is made up of a high strength impact resistant polymer material and has a resistive touch screen for user interfacing [21].

ACE/Eaton is another company that manufactures aluminum pipes, corrugated culvert pipe and accessories, also manufactures surge irrigation valve [22]. Their surge valve is made of aluminum and a single motor that routes the water either to left or right side of the valve.

2.5 Limitations of Surge Irrigation:

Though surge irrigation is a more efficient irrigation technique than others and can conserve water, it has some limitations. Surge irrigation may not always reduce the advance time of water down the furrow [23]. However, it is still beneficial due to its cost saving nature for labor and irrigation equipment, and it can ensure reduction in tail water. As infiltration of water is lower in surge irrigation, another concern may arise with the net water application. If the infiltration rate is reduced, less water may be saved at the root zone of crops during the irrigation set, and this can reduce the productivity of crops whose roots grow deeper into the soil [23]. To compensate, frequent irrigation may be required for such crops as well as increased irrigation set time, to avoid under watering. In this case, irrigation scheduling may need more attention. Field leveling could be another important aspect that irrigators must be cautious about. Many fields have low spots which create water ponding, and therefore water infiltrates more in such areas compare to other areas. It also causes the advancing water to slow down in such areas, therefore increasing the advance time [23].

3 Embedded Surge Controller

3.1 Overview

The hardware of the surge controller is based on the Arduino which is an open-source microcontroller platform and provides easy-to-use fast prototyping hardware boards. The software of the surge controller is based on the real time operating system FreeRTOS. The FreeRTOS running on the single core microcontroller provides a multitasking environment. As a result, multiple tasks involved in the irrigation process can be run using one core. The tasks involved in the surge irrigation process are discussed in the later section. This hardware-software implementation of the surge controller is unique and useful, as it has resulted into some valuable features which are explain in the later section of this chapter. Finally, this chapter discusses the detail operation and the limitations of the surge controller.

3.2 Introduction

Properly managed surge irrigation can increase crop productivity. Improper surge irrigation can result in deep percolation losses, insufficient water at the root zone, or water runoff at the end of furrows. If the actual advance time for a field is different than the programmed time, then either the furrows will not advance, or it will result into excessive tail water, resulting a loss of irrigation. Therefore, an irrigator may adjust the surge program on the surge controller to properly implement surge irrigation.

The surge valve has two parts. First is the mechanical valve that routes the water flowing through it, either to the left or to the right side of the valve. Second is the surge controller mounted on the top of the mechanical valve. The surge controller is an important part of the surge valve, as it provides an interface and control the surge valve. The controller accept the inputs from an irrigator such as the number of cycles for the advance phase (number of advance cycles can be preset or set to a default) , the time require for each cycle, calculate the on-time and off-time in each cycle as well as soak cycle times, change the direction of water either to the left or to the right side of the valve, and monitor the elapsed time.

3.3 Embedded Surge Controller

The surge controller is a microcontroller-based embedded system, designed to automate the process of surge irrigation. The design and implementation of the controller can be divided broadly into two parts: the hardware implementation and the software implementation.

3.3.1 Architecture

Figure 3.1 shows the architecture of the embedded surge controller. It consists of the microcontroller as a central processing unit to which the other peripherals are connected. The display unit communicates to the microcontroller over universal asynchronous receiver-transmitter (UART1). The motor driver is connected to the pins 4, 5, 7, 8 of the microcontroller. The flowmeter is connected to the pin 3 of the microcontroller. The microcontroller uses the timer3 interrupt to count the pulses from the flowmeter. The UART1, Vin pin, and GND pin of the cellular IoT device are connected to the UART2, 5 Vout pin, and GND pin respectively of the microcontroller. The secure digital (SD) card shield communicates to the microcontroller over serial peripheral interface (SPI). The microcontroller receives power from the combination of the battery and the solar panel unit. The custom power supply board is designed to step down the voltage of the battery from 12 volts to 5 volts. All the peripherals in the system receive power from this board.

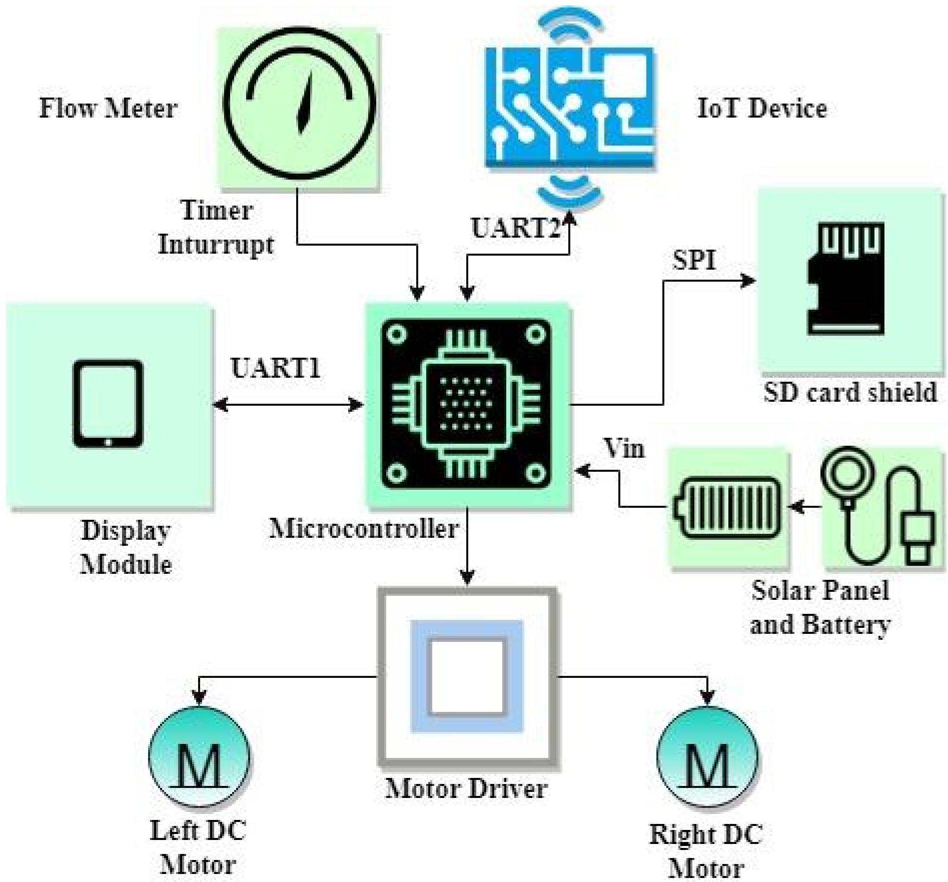

Figure 3-1. Architecture of Embedded Surge Controller.

3.3.2 Hardware Implementation

The hardware of the controller has the following components.

1. Micro-controller: The surge controller is based on the Arduino Due board for prototype that uses the single core ATSAM3X8E microcontroller, which is based on the ARM® Cortex®-M3 RISC processor. It is a 32-bit microcontroller that runs on 84 MHz, has 512 KB of flash memory, 96 KB of SRAM in 64KB+32KB banks, 54 digital input/output pins, 12 analog inputs, 4 UARTs, and 2 ADCs [24]. The Arduino Due board operates on 3.3V logic and has everything that is needed to support the microcontroller. Its features, such as a high operating frequency, a large RAM, and a large flash memory, have made it

suitable for larger scale applications. The microcontroller is the heart of the system. It receives the inputs from the display module and calculates the irrigation parameters and accordingly instructs other peripherals to achieve a successful irrigation.

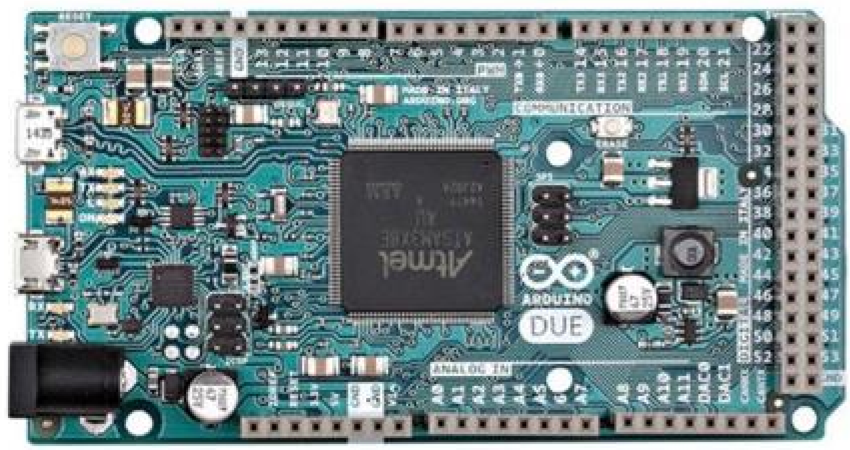

Figure 3-2. Arduino Due Board with ATSAM3X8E microcontroller. Source: *[25]*.

2. Power Supply: The power to the system is provided by a 12 volts 7.5 ampere-hours battery. Twelve volt is required to actuate the DC motors for switching the side of the valve, and to run the system for a long time without interruption due to the power loss. Though, the Arduino due board can run on a 12volt input, the on-board voltage regulator becomes very hot due to its inefficiency during the voltage convergence from 12 volts to 3.3 volts, which can damage the controller board. Therefore, a custom power supply circuit was made which reduces the input voltage down from 12 volts to 5 volts. The power supply board provides the power to other devices such as the touch screen, the direct current (DC) motor driver shield, the flow sensor, and the Secure Digital (SD) card shield.
3. Motor Driver: The surge controller uses Pololu Dual VNH5019 Motor Driver Shield. This board is easy to use and has two VNH5019 motor drivers to control two

bidirectional DC motors. Its features include, a wide operating voltage range of 5.5 - 24V, a continuous high output current of 12A, an input compatible with 5.5 or 3.5V system, a PWM operation up to 20kHz that is ultrasonic and allows for quieter motor operations, and an ability to control the current drawn and therefore the speed of the motors, which have made it highly robust and useful in the motor driver applications. The motor driver along with corresponding libraries for Arduino controls the two DC motors attached to the mechanical valve in order to change the direction of the water through the pipe either to the left or to the right side of the valve [26] [27].

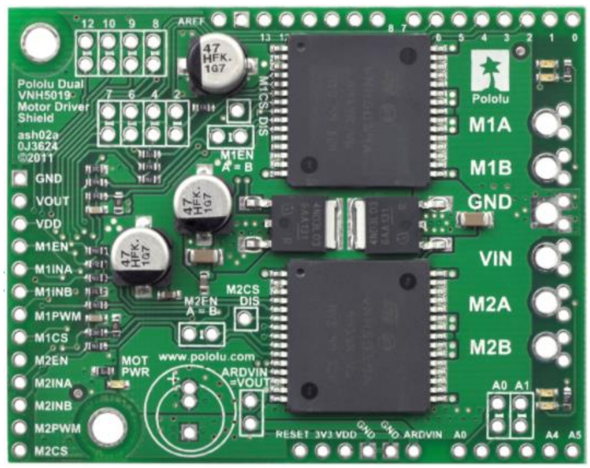

Figure 3-3. Pololu Dual VNH5019 Motor Driver Shield for Arduino. Source: *[26]*

4. Touch Screen Display: The controller uses 4D system's LCD TFT 5.0 inches capacitive touch Gen4 intelligent display module embedded with a DIABLO16 processor [28]. This display module has an 800 * 480 resolution, RGB 65K true to life colors, 6 banks of 32kB flash memory, and a 4.0–5.5V input voltage range. The microcontroller communicates with the display module over serial communication. The display module serves as the user interface. The microcontroller receives the irrigation parameter from an irrigator through the display module. Once the irrigation is started, the display module shows the irrigation status on the screen.

5. Solar Panel and Battery: The surge controller uses a 12 Volt 7.5 Amp Hour lithium iron phosphate (LiFeP) battery. This battery maintains higher voltage during discharge, has built in over current protection, and can replace 12V 17Ah Lead Acid batteries. The solar panel is used to change the battery. The solar panel provides 14V 180 milliamps output in full sun which provides current to the system as well as charges the battery. During night time or when the solar panel does not provide the required current, the system receives the current from the battery.
6. Customized Polycase box: The surge controller uses Polycase's YH-080804 series polycarbonate resin customized box that has solar panel mounted over it. This box is IP68 rated [29].

3.3.3 Software Implementation

The software of the surge controller is written in the Arduino Programming Language on the Arduino IDE. It is very similar to C/C++ language. The software code is written in the Arduino IDE, compiled on a local computer using the Arduino compiler, and the Arduino Due board is programmed using Universal Serial Bus (USB). The surge controller performs many operations while running irrigation, such as the system check which makes sure the battery voltage is enough to run irrigation, it calculates the irrigation parameters, monitors the time for irrigation, switches the side of the valve, shows the irrigation data on the display module, runs the Internet of Things task, and stops the irrigation once it is completed. The surge controller performs all these tasks using a single core microcontroller with the help of a real-time operating system FreeRTOS.

3.3.3.1 FreeRTOS

The FreeRTOS is one of the market's leading real-time operating systems that supports more than thirty architectures. It is professionally developed, quality controlled, robust, supported, and free to embed in commercial products without any licensing requirement. On a single core processor, multitasking is possible by switching between tasks very fast which creates an illusion of parallel processing. In FreeRTOS, different tasks are created depending on the functional requirements. The tasks are independent; however, they can relate to each other. The execution sequence of the tasks is defined by assigning priorities to them. The task which has the highest priority will be executed first. Once the task is completed, the control moves to the lower priority task. The program then runs in an infinite loop executing each task if it is in a ready state. If the task does not complete in assigned time frame, the controller saves the data related to that task into the stack memory and reloads it once the control comes to that task again. This means that the task resumes from where the control last switched to another task. The tasks can be created or deleted during the run time [30] [31]. The FreeRTOS therefore provides the core real-time scheduling functionality, inter-task communication, timing and synchronization primitives only [32]. Figure 3-4 shows the difference between concurrent programming and multitasking on a single core processor. Figure 3-5 shows the general lifecycle of the FreeRTOS Task. The structure of a program with FreeRTOS is as shown in Figure 3-6.

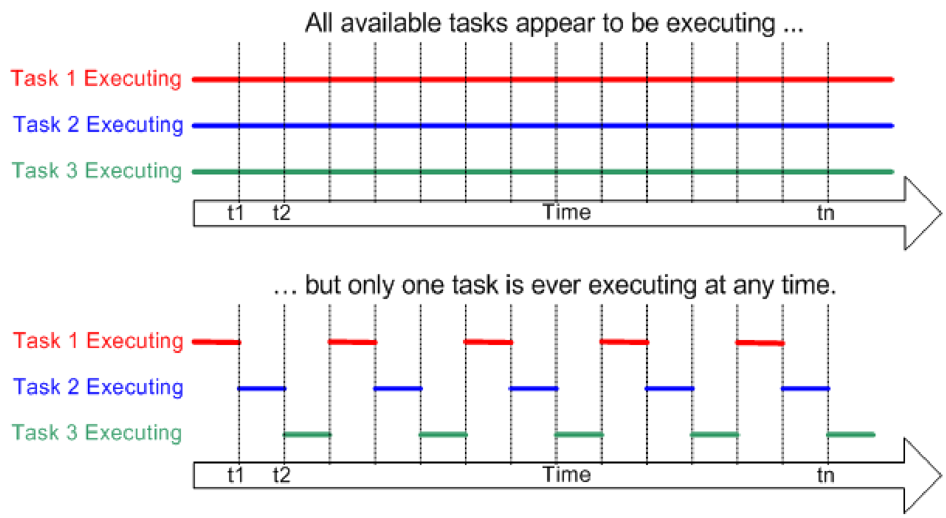

Figure 3-4. The difference between concurrent and multitasking. Source: *[33]*

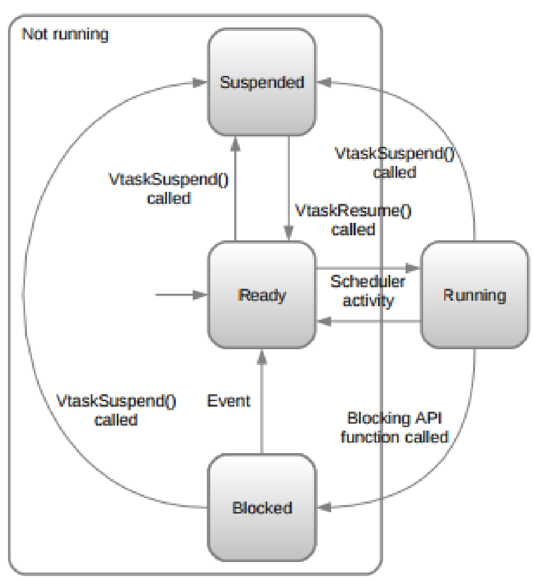

Figure 3-5. General life cycle of a task in FreeRTOS. Source: *[34]*

```
#include <FreeRTOS_ARM.h>
TaskHandle_t xTaskOne;
TaskHandle_t xTaskTwo;
// define two tasks for Blink & AnalogRead
static void TaskOne( void *pvParameters );
static void TaskTwo( void *pvParameters );
void setup(){
    //Initialize tasks
    xTaskCreate(TaskOne,
        "Task1",
        configMINIMAL_STACK_SIZE,
        NULL,
        tskIDLE_PRIORITY,
        &xTaskDummy
    );
    //Initialize tasks
    xTaskCreate(TaskTwo,
        "Task1",
        configMINIMAL_STACK_SIZE,
        NULL,
        tskIDLE_PRIORITY,
        &xTaskDummy
    );

    vTaskStartScheduler();
    Serial.println(F("Die"));
    while(1);
}
void loop(){
    // Do nothing...
}
//Function Definitions
static void TaskOne (void *pvParameters){
    /*code for task one...*/
    //The task will sleep for 10 seconds.
    vTaskDelay((10000L * configTICK_RATE_HZ) / 1000L);
}
static void TaskTwo (void *pvParameters){
    /*code for task two...*/
    //The task will sleep for 10 seconds.
    vTaskDelay((10000L * configTICK_RATE_HZ) / 1000L);
}
```

Figure 3-6. Structure of FreeRTOS for ARM microcontroller.

The surge controller uses FreeRTOS to achieve multitasking. Table 3.1 compares the priority assigned to them, name of the tasks, and the delay in milliseconds that the task will wait for the next run. Below the table, the tasks are explained in detail in the order of execution priority.

Table 3-1. Concurrent Tasks in Surge Controller.

Priority	**Task Name**	**Delay in milliseconds**
1	TaskTimer	2 000
2	TaskFlowMeter	100
3	TaskSysCheck	900 000
3	TaskSDCard	900 000
3	TaskIoT	600 000
4	TaskValve	5
5	TaskDummy	1800 000

1. Timer Task: The timer task is created to maintain the time duration for any operation during irrigation. The timer task initiates its operation for the first time during the irrigation event from the parameters calculated by the valve task. In subsequent cycles or modes of irrigation, the timer task calculates the time required and then counts down the time to zero. Once all the modes are executed, the timer task goes into the suspended state. As the timer tasks executes every 2 seconds, the display module shows the time that counts down by 2 seconds.
2. Flow Meter Task: This task is created to measure the flow rate of the water through the valve. The flow sensor attached to the surge controller counts the pulses depending on the flow of water through the valve. Depending on the number of the pulses, this task calculates the flow rate in gallons per minute.
3. System Check Task: This function is the second function that the controller executes when it is powered up or reset. This function checks the battery voltage every 15 minutes and displays voltage on the screen. If the voltage goes below the threshold value, this

function displays an alert message on the screen. The threshold value is 11 volts because the motors do not function below this voltage. It also checks whether the controller has resumed, or the irrigation has started with a new event. If the controller is resumed from the previously interrupted state, the controller resumes its operation by fetching the irrigation parameters from the flash, which has been saved by the SD card task before the interruption.

4. SD card Task: This task is created to save the irrigation data on the SD card installed on the system. It also saves the irrigation parameters that are required to resume the irrigation in the flash memory of the microcontroller in the event of an interruption.
5. IoT Task: This task in created to send the irrigation data to the IoT device in the surge controller. The IoT device and the microcontroller communicates with each other over serial communication. This task provides the run time data to the IoT device which sends this data to a central server over a GPRS cellular network protocol.
6. Valve Task: The valve task function is created to support user interfacing through the touch screen. This task executes the screen event-handler function through which the microcontroller communicates with the display module. This task runs every 5 milliseconds to ensure there is no delay in the display operation. Therefore, this task has the second lowest priority among all other tasks.
7. Dummy Task: The dummy task function is the first task that executes when the controller is powered up. This task does nothing but start the system check task. There should be at least one task running in the FreeRTOS operating system when all the tasks are in a suspended state. This task is the lowest priority task and runs every 30 minutes. This task

can provide a platform to run any other functionality that would be required in the future while the controller is in a standby state.

3.3.3.2 4DGL Program for Display Module

The 4D System's display module uses 4DGL programming language for programming a display module using the 4D system's WorkShop IDE. The 4DGL is a graphic oriented language that has similar syntax structure of languages such as C, Basic, and Pascal. The display graphics was developed in the Visi-Genie environment provided by the WorkShop IDE and couple of functions were coded in 4DGL language to put the display in a deep sleep mode for power conservation and display the amount of water used by the controller on the screen.

3.4 Working of Surge Controller

An irrigator powers on the controller by switching on the button on the surge controller. Once the controller is powered on, the software running on microcontroller starts the dummy task. The dummy task then starts the system check task where the system checks for the battery voltage and if the irrigation event was interrupted previously. If the voltage is inadequate for running an irrigation event, the system shuts down itself. If the system check passes, the system check task starts the valve task. The valve task enables the display module. The user can access the display module to enter the irrigation parameters. These parameters include the advance time for a field, the soak cycle time, the application depth, number of the surge cycles, the furrow length, the area of the field, the ratio of the left side division of a field in percentage, the mode of operation, and the automatic or the manual mode of irrigation. At the end, the system confirms with the irrigator for the entered parameters, starts the SD card task and the flow meter task, and begins the surge program. However, the irrigation time does not start until the flow meter senses the water flow through the valve. The irrigator can change the advance time during the surge mode of operation. However, parameters cannot be changed in the soak and the cutback mode of operation. The irrigator has been given an authority to start the IoT task though the display module. If the IoT task is running, the run time irrigation parameters are automatically sent to the remote server. However, the same parameters can be seen on the home page of the display unit.

3.5 Features of Smart Surge Valve

Though the smart surge controller resembles a simple microcontroller based embedded system, some new features that have been implemented in the software of the controller that some of which have never been implemented in surge controllers. These features are as follows:

1. Flow Sensor: The surge valve is equipped with a flow sensor. The surge controller can work as a flowmeter while running an irrigation. This is one of the unique features of the surge valve. None of the surge valves available in the market provide this feature.
2. Saves Irrigation Data: The surge valve has an embedded SD card shield for saving the irrigation data while irrigation is running. The irrigation data is saved in SD card in Excel format. The irrigator can access this data for any further analysis.
3. Dynamic Change in Irrigation parameters: The surge controller accepts dynamic changes in the input parameters. However, the irrigator can change the input parameters while the irrigation is in the surge mode. This is the most useful feature as it adjusts the subsequent advance times when the current advance time is modified while irrigating and help in order to achieve a successful irrigation.
4. Motor Drivers: The surge controller uses the motor driver shield that can limit the current flowing through the DC motors. Therefore, the speed of the motors can be controlled.

5. New Modes of Operation: The surge controller can run the three modes of irrigation: surge-soak, surge-cutback, and surge-soak-cutback. However, the commercial surge controllers run only the surge-soak mode.
6. Automatic Mode of Operation: The surge controller can run in automatic mode or manual mode. In the automatic mode, the controller automatically stops the irrigation once the desired application depth is achieved. In the manual mode, the controller runs the last mode of irrigation forever and the irrigator has to stop the irrigation manually.
7. Uneven Side Division: The surge irrigation requires a field division into two halves. The surge controller is independent of the proportion of the field division. The surge controller changes time per set per side according to the field division. This feature is already present in other surge controllers.
8. Internet of Things: This is a novel feature for surge irrigation. The surge controller uses telemetry to send the irrigation data over the Internet. As a result, the irrigator does not need to be present in the field during irrigation to monitor progress.
9. One Click Irrigation: The surge controller saves the input irrigation parameters onto the flash memory before starting the irrigation event. This data can be used to start an irrigation event for the next time on one click on the display module.
10. LCD Display: The surge controller uses a capacitive touch graphics display screen which provides a appealing user interface.

3.6 Limitations of Smart Surge Valve

Though the surge controller has some novel features and can achieve higher irrigation efficiency, improper management of it could lead to undesired results. Improper surge irrigation can affect crop productivity. Therefore, the irrigator should understand the limitations of surge controller and know how to use it. It is the irrigator's responsibility to input the correct irrigation parameters. If the surge controller receives the incorrect parameters, it could fail to run the event successfully. While an irrigation event is running, the screen must be put in a lock mode by pressing the lock button on the screen. This puts the screen processor in a deep sleep mode which reduces the power consumption of the system. If the screen is not put in the lock mode, the device could lose power and the event can fail. The screen can come out of the deep sleep mode if it is touched for couple of seconds.

4 Internet of Things in Surge Irrigation

4.1 Overview

Considering the impact of the Internet on human life through various domains, the Internet of Things (IoT), which is a recent and a significant evolution of the Internet, has become extremely important. Its ability to gather, analyze, and distribute data can be turned into valuable information and knowledge [35]. IoT has facilitated and improved human life and work efficiency in various domains. Figure 3.2 shows the area of IoT market and its some of the stakeholders.

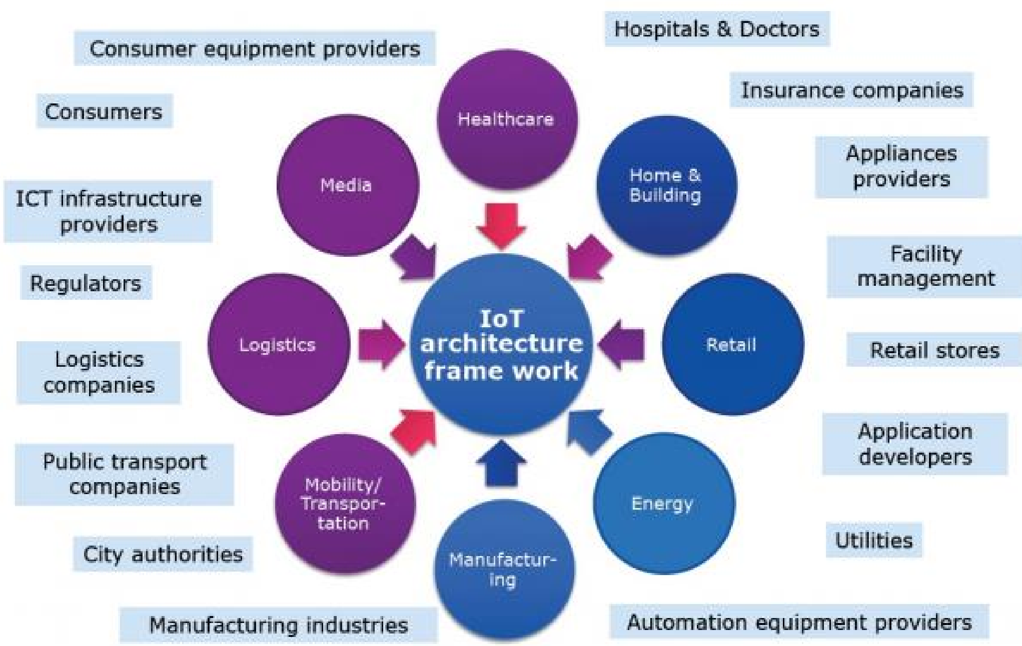

Figure 4-1. IoT market and stakeholders. Source: *[36]*

IoT is the system in which mechanical or digital machines, objects, sensors provided with unique identity and ability to transfer their data through computing device over the network without requiring human-to-human or human-to-computer interaction [37]. IoT has opened numerous ways of farming and raising livestock with the use of low cost, easy-to-install sensors and monitors that gather data, analyze it, and can act upon farmer's decisions to make farmers' lives better. The global population is expected to reach 9.6 billion by 2050 [4]. Therefore, the food production is expected to increase. However, extreme weather, rising climate change, and environmental conditions can impact the food production. With the help of IoT, the farming industry can achieve better monitoring and enhance productivity in farming.

This chapter describes the use of IoT technology to achieve better monitoring in surge irrigation. The embedded surge controller extends its functionality to make use of IoT technology for sending irrigation data over the internet to a remote server. An irrigator can access this information through a web application or a mobile application running on the remote server. This removes the restriction on the irrigator to be present at the field for monitoring the irrigation. The irrigator can see the irrigation status from anywhere on the mobile application or on the website. The IoT feature of the surge controller was tested in the laboratory environment by running few dummy irrigations without water.

4.2 IoT for Surge Irrigation

The surge controller makes use of the GPRS to make the device Internet enabled. The GPRS is a packet-based wireless communication service that allows computers and mobile phones to connect to the Internet. The IoT feature of the surge controller sends the irrigation data to a remote server. The remote server accepts the Transmission Control Protocol (TCP) connection, receives the data, and retains this data into the database. A simple web application is developed that runs on the same machine to show this data on the website. The Figure 4-2 shows the architecture of the IoT for surge irrigation.

4.3 Architecture of IoT

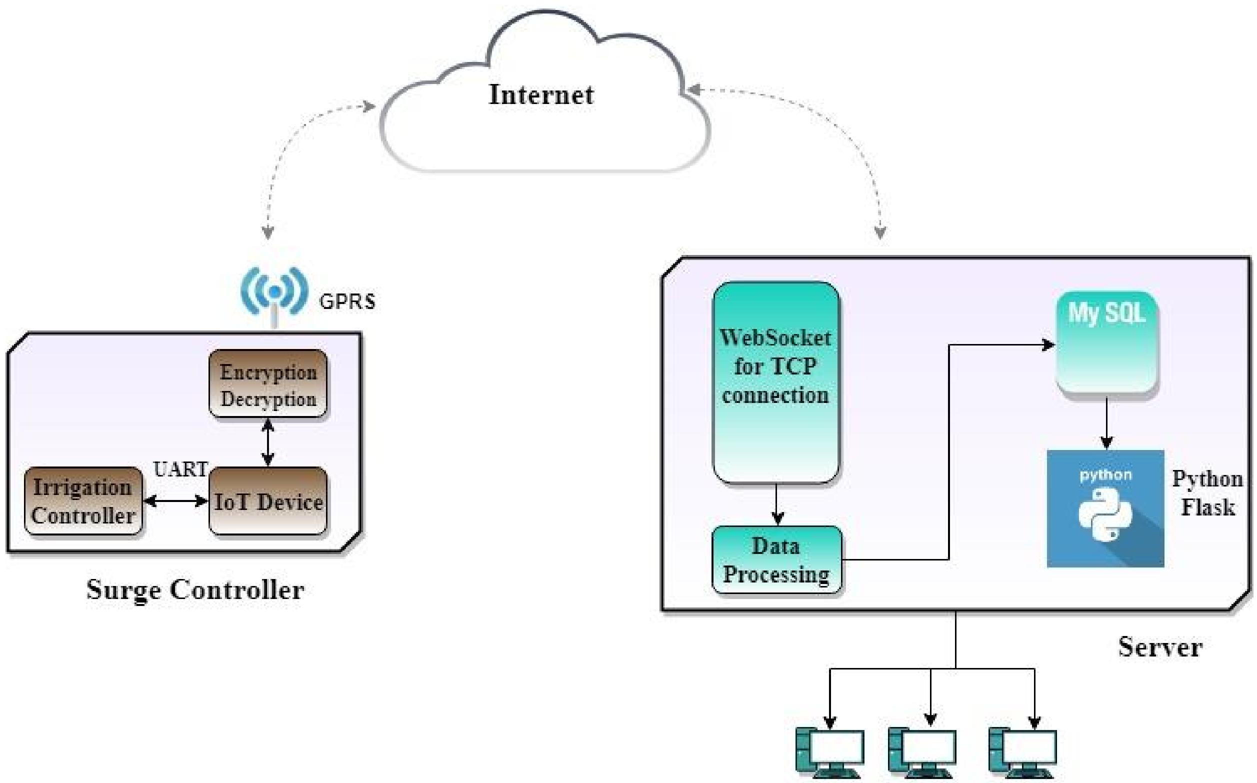

Figure 4-2. IoT Architecture for Surge Irrigation

The architecture of the IoT system for surge irrigation consists of Seeeduino GPRS IoT device embedded in the surge controller and the set-up for accepting and processing the irrigation data at the server-side. The microcontroller client running the irrigation sends the irrigation data to the IoT device over the UART. The IoT device encrypts the data and sends it over the Internet. The server IP is hardcoded in the IoT device. At the server-side the WinSock program is designed to accept the TCP connection from any client. The server waits until it receives the connection request from the client. Once the connection is established, the server accepts the data, process it, saves into a MySQL database, and returns to a listening state. The program logic is designed in such a way that it ensures the data is received from the correct client and the false data will not be saved into the database. The data can be retained on the database forever or as per the defined retention policy. The website running on the server shows the ten data points if irrigation is running or displays a message that irrigation is not running.

4.4 Hardware Implementation of IoT

The surge controller uses Seeeduino General Packet Radio Service (GPRS) IoT shield [38]. This shield connects to the internet through GPRS. It can make or answer calls and can send SMS messages. This shield also supports Bluetooth communication. It is based on Atmage32U4 and SIM800H. Atmage32U4 is a microcontroller compatible with the Arduino. However, the resources provided by this microcontroller are not enough to run the surge irrigation application. SIM800H support Quad-band 850/900/1800/1900MHz. This has an on-board voltage logic converter; therefore, it can work for 3.3 volts as well as 5 volts logic. Figure 4.2 shows the Seeeduino GPRS shield.

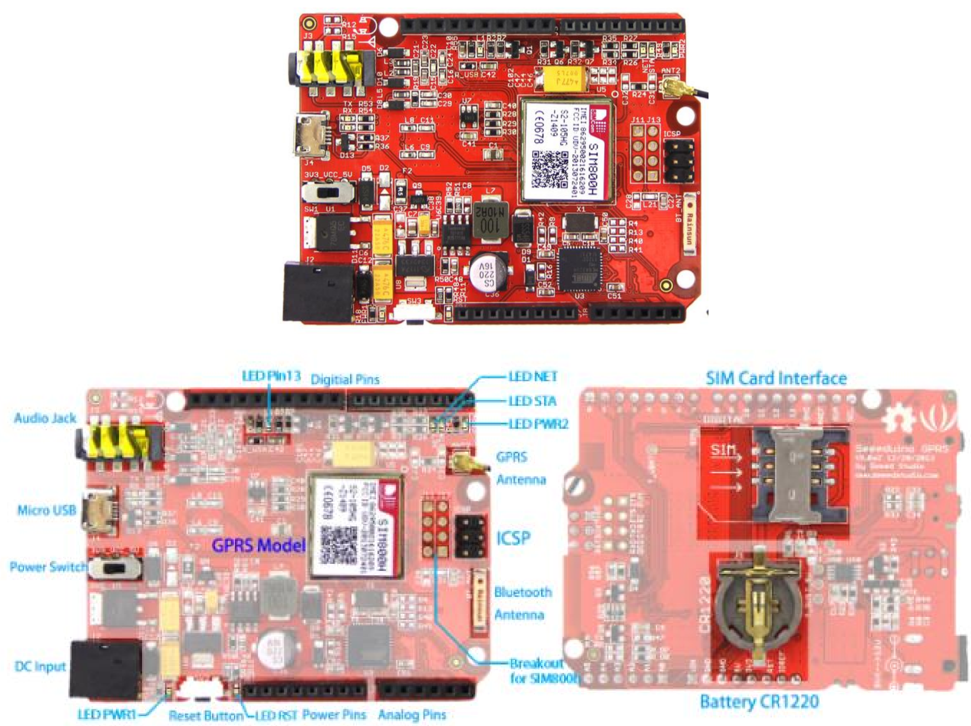

Figure 4-3. Seeeduino GPRS shield for Arduino. Source: *[38]*

4.5 Software Implementation of IoT

The software implementation of IoT for surge irrigation has two parts: software at the surge controller side and software at the server side.

4.5.1 Controller-Side Software Implementation

The Seeeduino GPRS board, at the controller side, has on-board Arduino compatible microcontroller. Therefore, the software for IoT is developed in C++ on the Arduino platform. The software code is written in C++, compiled on a local computer using Arduino compiler, and programmed the board using serial interfacing. The software code runs continuously on the board and performs its action only when it receives the data from the surge controller.

The GPRS shield requires the data enabled SIM card to connect to the internet. The GPRS shield along with the GPRS library can communicate to any internet enabled device over the internet. The second UART of the Arduino Due board is connected to the first UART of the GPRS shield for data communication. The surge controller executes the IoT task created in FreeRTOS that sends the irrigation data in the C data structure format to the GPRS shield every 10 minutes. The software running on the GPRS shield will wait for the data to be sent by the surge controller. Once the data is available at the GPRS shield, it adds its identification number to the received data and encrypts it using AES-128 bits encryption technique. The encrypted cipher text is then converted to a C-style character array to make it suitable for data transmission that is supported by GPRS library. For the prototyping, the AES encryption key is hardcoded in the GPRS code itself. The generated characters' array is then sent to the remote server over the TCP protocol. Figure 4.3 shows the program flow diagram of the IoT at the controller side.

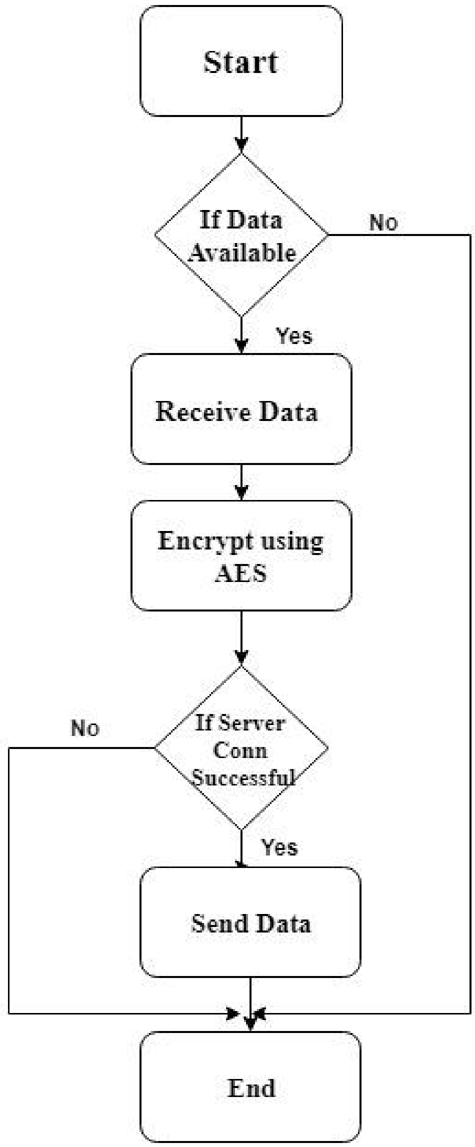

Figure 4-4. Flowchart of Controller-side IoT.

4.5.2 Server-Side Software Implementation

4.5.2.1 WebSocket Transmission Control Protocol (TCP) Connection

The server-side software implementation is designed to receive the data transmitted by the surge controller over the TCP connection, process the received data, and save it into the standardized query language (SQL) MySQL database installed on the server. At the server side, a WebSocket program is coded in the C++ language to accept the TCP connection from clients.

The WebSocket protocol opens asynchronous, bi-directional communication channel over TCP that works across the existing network. That means, the WebSocket program running on any machine does not require any extra piece of hardware other than an Internet connection. The surge controller sends the data to the remote server through the channel established by the WebSocket protocol. The server receives the AES encrypted data, the cipher transmitted by the controller, in the C style characters' array format. Once the data reception is completed, the server-side software decrypts the cipher and converts it into the C style data structure. This data is then passed to another function that saves it into the MySQL database. The AES decryption method requires the same key that the controller uses for the encryption. Therefore, for the prototyping, the key is hardcoded in the software itself. Figure 4.4 shows the program flow diagram at the server-side.

4.5.2.2 MySQL Database and the Website for Surge Irrigation

The surge controller sends the irrigation run time parameters to a remote server using IoT while irrigating a field. These parameters include the battery voltage, the current mode of surge, the irrigation cycle number, the elapse time for the cycle, the total time since when the irrigation is running, the flow rate in gallons per minute, the total amount of water in gallons used for an irrigation event, the total amount of water since the surge controller installed in a field for first time, and the status of the open side of the valve. All these parameters are saved in to the MySQL database table along with the device identification number to know from which device the data is received. As of now, the database structure consists of two tables for holding the information about surge irrigation. The first table is "devices_tb" which holds the use information about the user and the devices subscribed by the user. The other table is “deviceirrdata_tb” which holds the actual information of the irrigation parameters. Figure 4.5 shows the tables’ structure used for the IoT surge application.

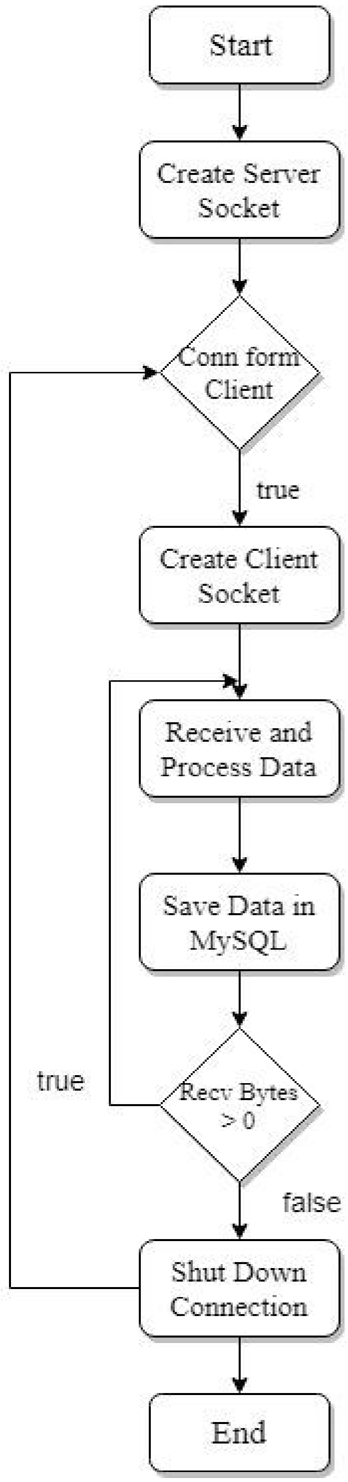

Figure 4-5. Flowchart of server side IoT

```
mysql> desc devices_tb;
+---------------+--------------+------+-----+-------------------+-------+
| Field         | Type         | Null | Key | Default           | Extra |
+---------------+--------------+------+-----+-------------------+-------+
| user_id       | int(6)       | YES  | UNI | NULL              |       |
| name          | varchar(100) | YES  |     | NULL              |       |
| email         | varchar(100) | YES  |     | NULL              |       |
| username      | varchar(50)  | YES  |     | NULL              |       |
| password      | varchar(100) | YES  |     | NULL              |       |
| register_date | timestamp    | NO   |     | CURRENT_TIMESTAMP |       |
+---------------+--------------+------+-----+-------------------+-------+

mysql> desc deviceirrdata_tb;
+----------------+-------------+------+-----+-------------------+-------+
| Field          | Type        | Null | Key | Default           | Extra |
+----------------+-------------+------+-----+-------------------+-------+
| bvolt          | varchar(20) | YES  |     | NULL              |       |
| cycle_time     | varchar(50) | YES  |     | NULL              |       |
| elaps_time     | varchar(50) | YES  |     | NULL              |       |
| total_time     | varchar(50) | YES  |     | NULL              |       |
| total_gals     | varchar(20) | YES  |     | NULL              |       |
| current_gals   | varchar(20) | YES  |     | NULL              |       |
| flow_rate      | varchar(20) | YES  |     | NULL              |       |
| side           | varchar(10) | YES  |     | NULL              |       |
| cycle_number   | varchar(10) | YES  |     | NULL              |       |
| surge_mode     | varchar(20) | YES  |     | NULL              |       |
| device_id      | varchar(50) | YES  |     | NULL              |       |
| recv_timestamp | timestamp   | NO   |     | CURRENT_TIMESTAMP |       |
+----------------+-------------+------+-----+-------------------+-------+
```

Figure 4-6. MySQL tables' structure used for IoT Surge Application.

A simple web application was developed in the Python Flask environment to display the irrigation data on the website. The Flask is a micro web development framework written in Python and based on the Werkzeug toolkit and Jinja2 template engine [39]. In the web application for surge irrigation, the irrigator who has access on the website can see his subscribed devices by login using the user identification and password. The use can access his subscribed devices only. Figure 4-7 shows the device dashboard created for a dummy user. By clicking the device_id link on the device dash board page, the irrigator enters in the device information web page where the information about the irrigation status logged by the device into the database can be seen. Figure 4-8 shows the device information page. The device information page shows the latest data logged into the database by the device and the past 10 entries. However, the web page can be designed as per the need and more functionalities can be added.

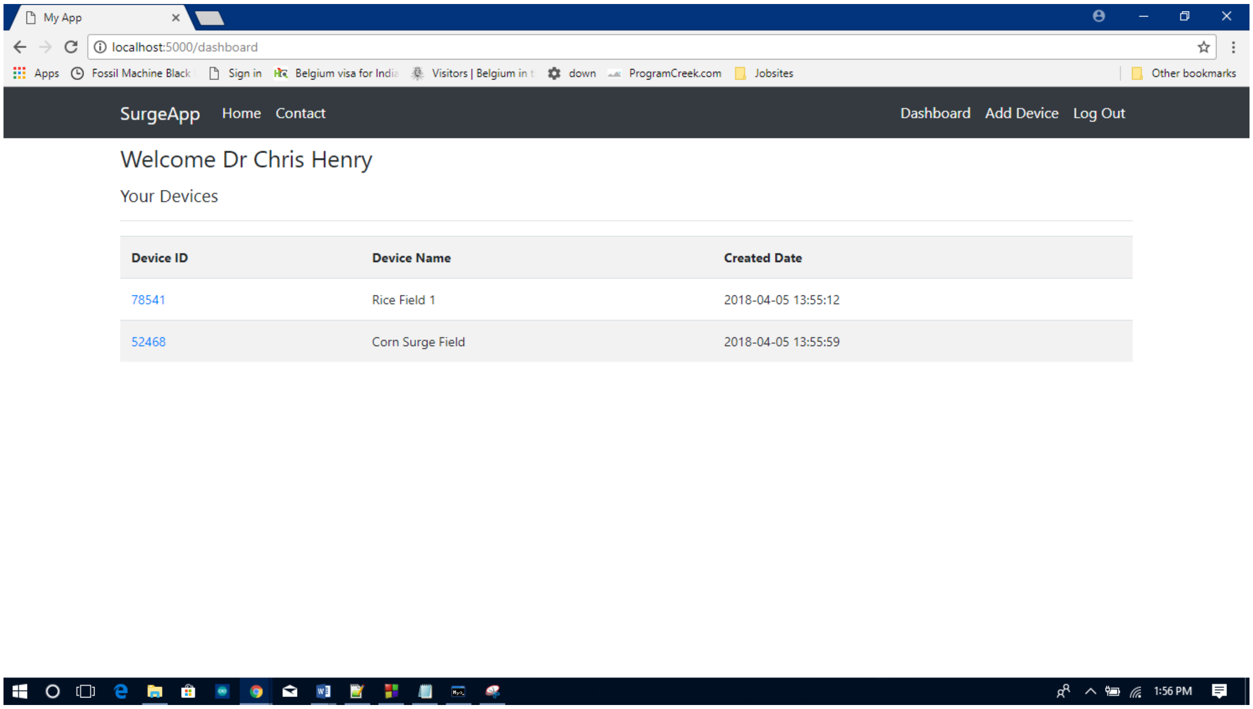

Figure 4-7. The device dashboard webpage for the user.

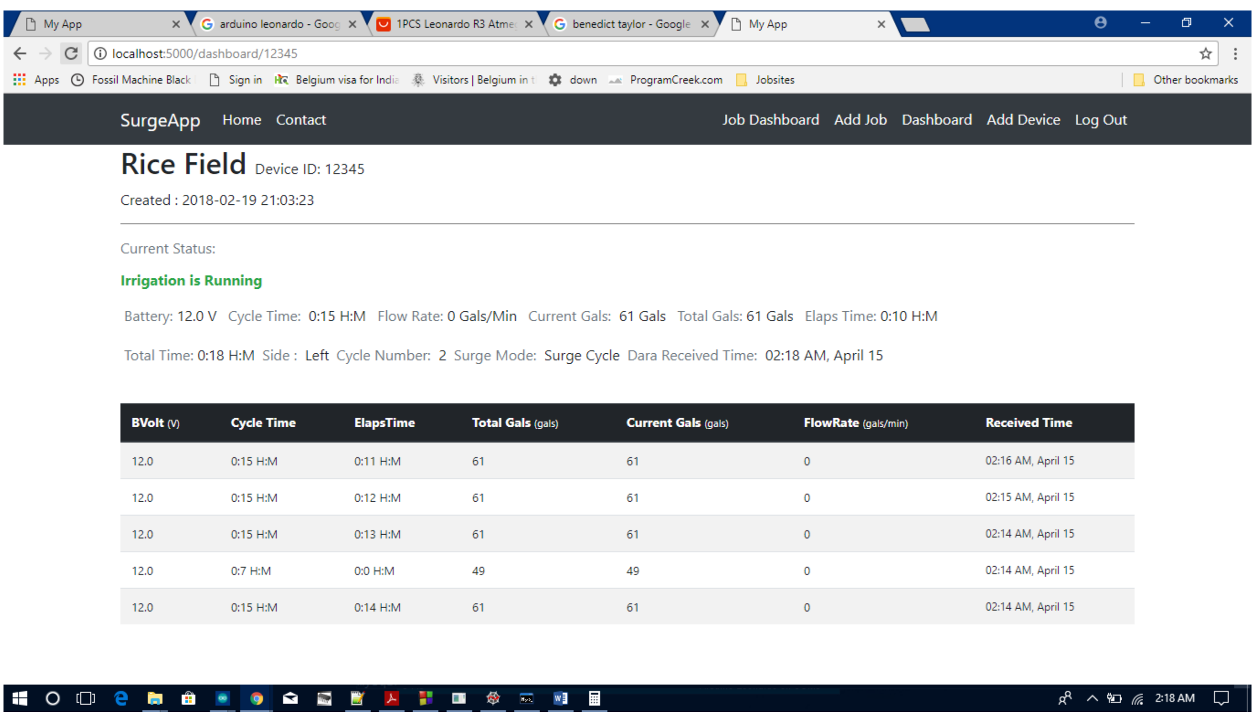

Figure 4-8. The device information web page to check irrigation status.

4.6 Advance Encryption Standard

The IoT feature of the surge controller uses GPRS wireless communication to send the data to the remote server. This data communication can easily be intercepted by a malicious user. This could make the IoT feature unsecured and vulnerable. In order to provide the privacy and the security to the data communication process, the surge controller is embedded with the encryption algorithm Advance Encryption Standard (AES). The AES technique transforms the irrigation data into the encrypted information called cipher text using a secret key embedded in the system before attempting the data transfer.

The AES algorithm is a type of symmetric encryption algorithm that was developed by two Belgian cryptographers Joan Daemen and Vincent Rijmen [40]. It uses a single key to encrypt and decrypt the data. It has been proven that AES is more effective than the other symmetric algorithms such as Data Encryption Standard (DES) [41]. The AES encryption is a block cipher that encrypts data on a per-block basis. The AES supports data block of 128 bits, 196 bits, and 256 bits. The data block can be encrypted using a key of size 128 bits, 196 bits, and 256 bits. The AES encryption method undergo a series of linked operations. Some of them involve replacing an input with different substitutions and others involve shuffling the bits around. These operations repeat themselves in number of rounds which are depended on the key length. The AES for 128 bits key undergoes 10 rounds, 196 bits key undergo 12 rounds, and 256 bits key undergo 14 rounds. The decryption process is exactly opposite of the encryption process. Figure 4.6 shows the process involved in AES for encryption and decryption [42]. AES method is faster and more secure than the IDES and 3DES encryption algorithms [43]. It is widely used international standard encryption method [44]. The table 4.1 shows the comparison between DES, AES, and RSA algorithms [45].

Table 4-1. Comparison between DES, AES, and RSA algorithm. Source: [45]

Features	DES	AES	RSA
Developed	1977	2000	1977
Key Length	56 bits	128, 192, 256 bits	More than 1024 bits
Algorithm	Symmetric	Symmetric	Asymmetric
Block Size	64 bits	128 bits	Minimum 512 bits
Security	Not secure enough	Excellent secured	Least secure
Hardware & Software Implementation	Better in hardware than software	Better in both	Not efficient
Encryption and Decryption	Moderate	Faster	Slower

The surge controller encrypts the irrigation data using AES-128 bits encryption. The program that runs on the microcontroller embedded in the Seeeduino IoT device performs the encryption process. The irrigation data received over the UART in a C-style data structure is converted to C style characters' array. This will be the plain text for encryption operation. If the plain text is not of the size of the plain text supported by the AES method, it undergoes the padding operation that adds zeros at the end of the plain text. This plain text then undergoes a series of operations for 10 rounds. The generated cipher is then transmitted to the server. At the server side, the WebSocket program receives the cipher text into the receive buffer. The cipher text then undergoes the decryption process to retrieve the irrigation data. However, this data is still in the C style characters' array, which is then converted to C-style data structure by the program. Now the data in the structure can be accessed to retrieve the individual irrigation parameters. These parameters are then processed to save the data in suitable format on the MySQL database.

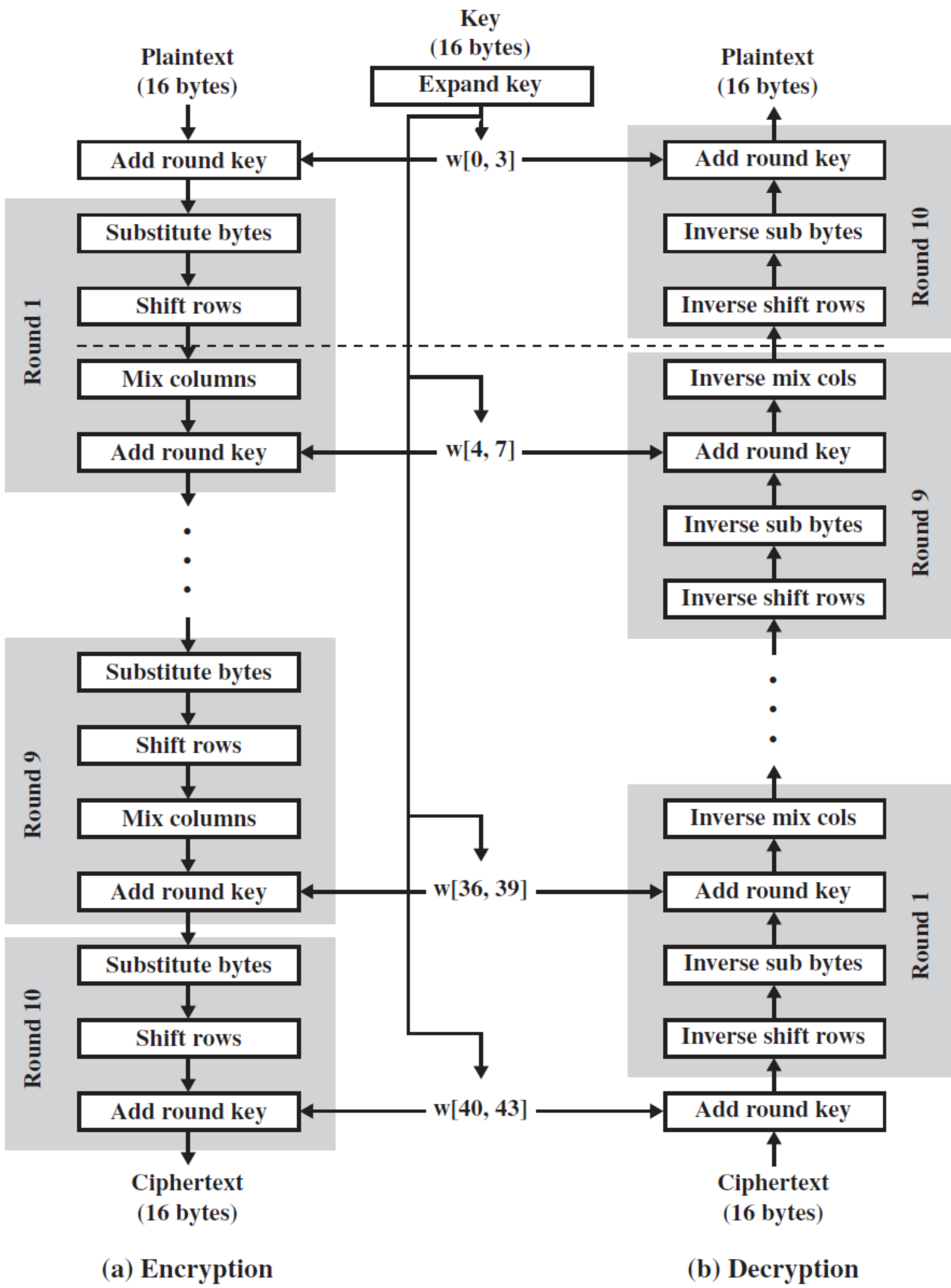

Figure 4-9. The process of encryption and decryption in AES. Source: *[42]*

4.7 Advantages of IoT in Surge Controller

Though, the IoT feature does not take a part in actual irrigation, it is found to be successful in providing a virtual assistance to the irrigator for monitoring irrigation. The irrigator can monitor the status of the irrigation event from anywhere provided he has the internet connection. The IoT feature of the surge controller logs the irrigation data into the MySQL database every 10 minutes. The server can retain this data for a long time. With the use of different data analytics methods and techniques, it is possible to understand the irrigation pattern of different fields and keep a track on amount of water required for different crops during a particular season or in the subsequent years. All this analysis can be used to understand the impact of irrigation on the crop productivity. As a result, changes or modifications can be done in order to achieve better crop production. Moreover, bi-directional data communication using IoT technique can remove the dependency on an irrigator for starting the irrigation event. In this way, the remote connection with the user using GPRS model to the surge controller hardware can help to automate the surge valve for irrigation.

4.8 Limitations of IoT in Surge Controller

The IoT components comprised of the GPRS modem and a remote server is useful in terms of monitoring the irrigation events. However, this thesis study limits the use of IoT in the surge controller to a one directional communication process. The surge controller opens a communication channel to communicate with a remote sever and sends irrigation data. However, the server does not communicate back to the surge controller. This means the surge controller's operations cannot be controlled remotely, and the irrigator has to be present physically at the field to start or change the irrigation event parameters. However, further software development in this area can remove this limitation. Another limitation of the IoT system designed for the surge controller is the AES encryption key and the IoT device ID are hardcoded in the device. There is no provision provided for changing the encryption key once it is programed in the device. However, this has been done for the prototyping purpose and is a feature that can be added in the future.

5 Results

5.1 Overview

The surge controller was fabricated and tested inside the Biological and Agricultural Engineering laboratory; however, it was briefly tested in the field. As discussed in previous chapters, the surge controller is divided into two parts: the standalone embedded system and the IoT for the surge controller. The IoT part of the surge controller was implemented and tested in the laboratory after implementing the embedded system by running some irrigation events on the controller without water. The functionality of the IoT system was examined by comparing the irrigation parameters on the controller display module and the parameters sent by the IoT to the remote server. The embedded system part of the surge controller was tested in a 20 acres soybean field during the summer 2017 irrigation season. Five irrigation events were performed with the controller to check its functionality, sustainability, reliability, and overall performance in the natural field conditions. The surge controller was expected to run day and night for 45-50 uninterrupted hours, because an irrigation could take this long. However, the duration could be different depending on the field length, size, and water flow rate. The following section discusses more about the irrigation events that were performed with the controller.

5.2 Results of Irrigation Events

Before installing the surge valve in a field, it was tested in the laboratory for proper functioning of the software and the other peripherals used in the surge controller. After this, the controller was installed in a field and operated for the surge irrigation events. At every event, the advance time for the field was set to different times than previous events to check the behavior of the valve and its capability of handling various inputs.

5.2.1 First Event

The first event was run with a 24 hour advance time, water flow rate of 500 gallons per minute, in surge-soak irrigation mode, and manual mode. Depending on the inputs, the controller was expected to run the advance phase for 24 hours and then the soak mode until the controller was shut down manually. However, the first event ran only for 10 hours. Though the first irrigation event did not run for the input time, it ensured that the peripherals used in the system were functioning as expected. Within this time duration, all the components in the system performed their task. The system did not run for the expected time duration due to power loss. The solar charger was found to be faulty, and the battery alone could not run the system beyond 10 hours. This irrigation event was successful in terms of the reliability of the peripherals used in the system and the software. The FreeRTOS code on the microcontroller ran successfully. It monitored the time correctly, measured the flow rate and provided the user interface while irrigating.

5.2.2 Second Event

The second event was created with the advance time of 22 hours, a flow rate of 700 gallons per minute, in surge-soak irrigation mode, and manual mode. It was expected from the controller to run the advance phase for 22 hours and the soak phase until the controller was manually shut down. However, this event ran for 18 hours and the controller was shut down due to power loss. The event was started in the afternoon at 1 PM and the system lost its power during the night around 5 AM. Further analysis of this issue came to the point that the system consumes more current than what is generated by the solar panel. Therefore, during the day, to fulfil the power need, the system used both the solar panel and the battery. During the day, the system received part of the required current from the solar panel. Therefore, the discharge rate of the battery during this time was less. After the sun set, the battery alone was supporting the system; therefore, it was discharged faster, and the system was shut down after the battery was discharged completely. The system was found to require the current of 215 milliamps when motors were not moving; however, the solar panel can deliver 180 milliamps during the full sun and around 140 milliamps or even less in an overcast sky.

5.2.3 Third Event

At the third event, the focus was more on reducing the current consumption of the system. It was found that the DC motors and the display module consumed more current than the other components. It was also found that while using the relay switches for moving the motors, the current consumption was reaching 5-8 Amps, and as voltage went down, the current consumptions was going higher than this number. This was quite obvious in the constant load circuit that as voltage goes down, the current consumption increases. This was causing the battery to drain out very quickly just after two or three advance cycles of the irrigation. The DC motor shield along with the corresponding Arduino library can used to control the speed of DC motors. As motor speed decreases, the motor current consumption decreases. In this way, the DC motors was made to consume a maximum of 2 amps at the cost of compromising their speed. Another change was made was that the display module was programed to use the deep sleep mode. The changes made in the system were found effective as they dropped the current consumption dramatically from 225 milliamps to 120 milliamps when motors were stationary and maximum of 2.3 amps when motors were moving. The third event was tested with the same settings that was used in the second event. The event ran successfully for 45 hours without interruption. Even after 45 hours, the battery was found fully charged. This event was the first event that ran successfully throughout entire event. However, later it was observed that the deep sleep mode of the display module had an issue. The display module was taking time to reestablish the connection with the microcontroller for synchronization of data transfer. The reason behind this was the user interfacing task that performs the connection reestablishment in the FreeRTOS based controller program was switching to the next task before the successful connection established. Though this issue took a long time to resolve, it has finally resolved by altering the code in the display program. The display code was programed in such a way that the display module was inaccessible to the user until the successful connection was reestablished between the display module and the microcontroller. Once the successful connection was reestablished, the display module showed the home page from where the irrigator was able to access the controller. In this way, whenever the irrigator wanted to check the irrigation status, he had to touch the screen for two seconds to wake up the display module from the deep sleep mode and then had to wait 1-2 seconds for appearing the home page.

5.2.4 Fourth Event

The fourth event was conducted to test the sustainability of the surge valve at a higher water flow rate. This event was tested with the advance time of 12 hours and the water flow rate of 1000 gallons per minute. This event ran with a partial success. Though the controller ran for 12 hours uninterrupted, it was not possible for the irrigator to wake up the display module from the deep sleep mode properly. As a result, the display module lost the communication with the controller and did not show the status of the irrigation. The controller had to restart to establish the communication between the display module and the microcontroller. However, this was not related to flow rate the valve was experiencing. The controller withstood the 1000 gallons per minute flow rate. The resume function of the valve was tested during this event and functioned as intended. After resetting the controller deliberately during the running event, the controller resumed the irrigation from the point when it was reset.

5.2.5 Fifth Event

After the fourth event, some changes were made on the surge program to resolve the synchronization issue of the display module due to the deep sleep mode. The user could wake the display up from the sleep mode, but the display did not properly show the home screen where all the irrigation data should have been updated by the microcontroller until the connection is established with the microcontroller. Though this worked in the lab, in the field during irrigation it did not work as expected. Initially it worked, but when the user attempted to wake up the screen after 5 hours, it did not wake up for a long time. Finally, the controller was reset, and the event was completed.

5.2.6 Laboratory Tests after Irrigation Season

After running the few irrigation events during the irrigation season, some of the listed features were added in the fall 2016. These features include dynamic changes in the input parameters, changes in the irrigation parameters according to the field division, and gather and bundle up the irrigation data to send it to the server over the internet. These features were tested in the lab by running the controller without water. Also, during this time, the issue with the display module was resolved to a greater extent than previously. The issue was resolved by changing the display behavior after waking up from the deep sleep mode. The display does not show the data on the screen until it establishes and synchronizes with the microcontroller. Therefore, the wake-up time for the display is not a fixed time. It varies between 2 to a maximum of 8 seconds. The IoT feature was developed and implemented in the fall 2017 and the spring

2018. It was tested by running the surge controller in the laboratory. The data sent to the remote server was compared with the data displayed on the display unit. Both the data points were found same. After the successful data communication between the client and the server, few irrigation events each of 10 to 15 hours were run on the controller to check if there was any loss of data points in data communication with the server. Only one or two data point loss were observed out of 20 to 30 data points. The reason was the receiving data buffer was not cleared properly in the last data point reception. As a result, when the next data point reached to the server, it exceeded the buffer limit and resulted into the disconnection. However, the program running on the server was able to reestablish the connection with the client to accept the next data point. Though the work was done on calculating and decreasing the power consumption of the system during the field testing, the total power consumption with the IoT device connected was not calculated.

6 Conclusions and Future Work

6.1 Conclusions

Surge irrigation, which is an altered method of the continuous flow furrow irrigation, is an intermittent application of water that creates a series of ON and OFF cycles of variable time duration. This method ensures the reduction in the utilization of water in irrigation and the total irrigation time as compared to the continuous flow furrow irrigation. Properly managed surge irrigation can increase row crop irrigation efficiency.

In this thesis, the surge controller that automates the process of surge irrigation was designed, developed, and tested. The designed surge controller is an embedded control system that uses an ARM Cortex M3 microcontroller mounted on Arduino Due board, the Arduino compatible Pololu dual VNH5019 motor driver shield, customized power supply board, a flow sensor, an SD card shield, and capacitive touch display module for user interfacing. The firmware of the system was developed using the Arduino programming language that uses the real-time operating system FreeRTOS to achieve multitasking. The FreeRTOS-based software running on the microcontroller enabled modularity in the program, simplicity in application development, and improved application efficiency by providing the facility to divide different operations involved in surge irrigation into the different independent tasks. These tasks included the timer task that keeps the track of irrigation cycles' time and the total time for the irrigation event, the flow meter task that calculates the water flow rate depending on the pulses generated by the flow sensor, the system check task that checks the system upon the power-on and every 15 minutes to ensure proper functioning of the controller, the IoT task that takes care of sending the irrigation data to the IoT device which in turn sends the data over the Internet, the SD card task that ensures saving the irrigation data onto the SD card, the user interfacing task that communicates with the user through the touch screen, and the dummy task that runs forever until the system is powered off for proper functioning of the system.

Though all the components worked as expected and the surge controller was able to run the surge program successfully, the unexpected irregular power shutdown due to the high-power consumption was a significant issue that was faced during the field testing. However, this issue was resolved by replacing the relay switches used for actuating the motors by the motor drivers that can put the limit on the current flowing through it. Another power consuming source was the touch screen. However, the touch screen was programed to implement a deep-sleep mode to reduce the power consumption. These workarounds worked well up to the extent that the surge controller could run the surge irrigation event without the power loss.

In this thesis, IoT technology was successfully implemented along with the AES encryption technique for the data communication security. The IoT feature provided a virtual assistance for monitoring an irrigation event; thereby, removing the dependency on the irrigator to be physically present in the field for monitoring the irrigation event. The GPRS based IoT system was found useful as the data enabled SIM card did not require any other setup to connect to the Internet.

The surge controller was tested for its functionality in a field. Some of the tests were successfully run and some failed. The failed tests were analyzed and resolved in subsequent test runs. The newly implemented surge controller is equipped with more advance features such as ability to accept dynamic changes in the inputs, measure the water flow rate using a flow sensor, provided a touchscreen user interface, and use of IoT. These features may be useful in automating and monitoring irrigation.

6.2 Future Work

Though the new surge controller works good with the current hardware-software setup, there is still extensive scope in upgrading them for achieving the better performance. For an example, the resources available on the microcontroller used in the controller will not be sufficient if more functionality needs to be added in the system. In this case, other microcontroller with the greater resources can be used in the surge controller. Microprocessor is a good option for replacing the microcontroller as it provides more processing power and memory. However, it should be a low power device. Also, a customized dedicated processor can be made to replace the microcontroller using the latest technologies in the embedded systems field such as System on Chip (SoC) using Field Programmable Gate Arrays (FPGAs). Similarly, the other peripherals used in the surge controller can be replaced according to their performance, the price, and the power requirements.

Currently, the IoT feature of the surge controller is a one directional process that sends the data from the controller to the remote server. However, the remote server cannot send data to the controller. In the future work, the system can be made bi-directional so that the server can communicate with the controller. The bi-directional process will allow the remote server to send the irrigation input parameters to the controller and request to start an irrigation event, pause or cancel the irrigation. In this way, the irrigation process can be made completely automated. For demonstration of the IoT in this thesis, the Python website was developed to show the data on the web. The website shows the latest data entry from the device and the past 10 entries which provides more information about the irrigation event. The website can be made more informative by adding more functions including analyzing data and displaying more details on the website. In the future, the website can be replaced with a mobile application. Currently, the system uses a separate board for the IoT. However, the prototype could eventually be packed into a single Programmed Circuit Board.

7 References

[1] M. Macatta, "Importance of agricultural sector in a country's economic development," 2 December 2016. [Online]. Available: https://www.ippmedia.com/en/features/importance-agricultural-sector-country%E2%80%99s-economic-development.

[2] Department of Economic and Social Affairs, Population Division, "World Population Prospects: The 2015 Revision," United Nations, 2015.

[3] S. Kulkarni, "Innovative Technologies for Water Saving in Irrigated Agriculture," *International Journal of Water Resources and Arid Environments,* vol. 1, no. 3, pp. 226-231, 2011.

[4] G. Schaible and M. Aillery, "Irrigation & Water Use," United States Department of Agriculture, 28 April 2017. [Online]. Available: https://www.ers.usda.gov/topics/farm-practices-management/irrigation-water-use/.

[5] V. Dhawan, "Water and Agriculture in India," in *Global Forum for Food and Agriculture*, 2017.

[6] R. M. Waskom, "Best Management Practices For Irrigation Management," Colorado State University Cooperative Extension, Colorado, 1994.

[7] R. G. Evans, "Irrigation Technologies Comparisons," 2010.

[8] A. A. Bishop, G. J. Poole, N. L. Allen and W. R. Walker, "Furrow Advance Rates under Surge Flow Systems," *Journal of the Irrigation and Drainage Division,* vol. 107, no. 3, pp. 257-264, 1981.

[9] T. H. Podmore and H. R. Duke, "Field evaluation of surge irrigation," *ASAE Paper,* vol. 82, p. 2102, 1982.

[10] K. Padia, "How is IoT transforming irrigation and improving agriculture," 26 September 2016. [Online]. Available: https://opensourceforu.com/2016/09/iot-transforming-irrigation-improving-agriculture/.

[11] J. E. Ibarra-Esquer, F. F. González-Navarro, B. L. Flores-Rios, L. Burtseva and M. A. Astorga-Vargas, "Tracking the Evolution of the Internet of Things," *Sensor,* vol. 17, no. 1379, 2017.

[12] J. Chase, "The Evolution of the Internet of Things," Texas Instruments, Texas, 2013.

[13] G. E. Stringham and J. Keller, "Surge Flow for Automatic Irrigation," in *Irrigation and Drainage SpecialtyConference of the American Society of Civil Engineers*, 1979.

[14] Plant and Soil Sciences eLibrary, "Furrow Irrigation Management," Plant and Soil Sciences eLibrary, 2018. [Online]. Available: http://croptechnology.unl.edu/pages/informationmodule.php?idinformationmodule=1130447123&topicorder=1&maxto=13&minto=1.

[15] A. Solaimalai and A. Rajagopal, "Surge Irrigation: A New Surface Irrigation Method-A Review," *AGRICULTURAL REVIEWS-AGRICULTURAL RESEARCH COMMUNICATIONS CENTRE INDIA,* vol. 23, no. 3, pp. 214-218, 2002.

[16] C. D. Younts, "Impact of Surge Irrigation on Furrow Water Advance," *Transactions of the ASAE,* vol. 39, no. 3, pp. 973-979, 1996.

[17] K. K. Tanji and N. C. Kielen, Agricultural Drainage Water Management in Arid and Semi-Arid Areas, Rome: Food and Agriculture Organization of the United Nations, 2003.

[18] C. G. Henry and L. J. Krutz, "Surge Irrigation Information," 2017.

[19] G. Fipps, "Grower's Guide to Surge Flow Irrigation," Texas A&M Agrilife Extension, 2013.

[20] p&r surge systems, "P&R Valves," p&r surge systems, inc, 2017. [Online]. Available: http://www.prsurge.com/index.html.

[21] DamGates by Triad Fastener L.P., "Surge Irrigation," DamGates, 2016. [Online]. Available: http://www.damgates.com/DamSurgeValve/index.html.

[22] "ACE Irrigation / Main Page," ACE/Eaton Metals, [Online]. Available: http://www.acenebraska.com/index_files/ACEIRRIGATION.html.

[23] Nebraska Extension, "Fundamentals of Surge Irrigation," University of Nebraska-Lincoln, [Online]. Available: http://extensionpublications.unl.edu/assets/html/g1870/build/g1870.htm#target2.

[24] Atmel Corporation, "Atmel-11057C-ATARM-SAM3X-SAM3A-Datashee," 13 7 2012. [Online]. Available: http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-11057-32-bit-Cortex-M3-Microcontroller-SAM3X-SAM3A_Datasheet.pdf. [Accessed 23 3 2015].

[25] Arduino, "ARDUINO DUE," Arduino, 2018. [Online]. Available: https://store.arduino.cc/usa/arduino-due.

[26] Pololu Corporation, "Pololu Dual VNH5019 Motor Driver Shield for Arduino," Pololu Corporation, [Online]. Available: https://www.pololu.com/product/2507.

[27] STMicroelectronics, "VNH5019A-E,Automotive fully integrated H-bridge motor driver," STMicroelectronics, 2013.

[28] 4D Systems, "gen4-uLCD-50D Series," 4D Systems, [Online]. Available: https://www.4dsystems.com.au/product/gen4_uLCD_50D/.

[29] Polycase, "YH-080804 YH Series Hinged Waterproof NEMA Electrical Enclosures," Polycase, 2018. [Online]. Available: https://www.polycase.com/yh-080804.

[30] Amazon.com, Inc, The FreeRTOS™ Reference Manual, Amazon Web Services, 2017.

[31] R. Barry, Mastering the FreeRTOS™ Real Time Kernel, Real Time Engineers Ltd, 2016.

[32] Amazon Web Services, Inc., "What is An RTOS?," Amazon Web Services, Inc., [Online]. Available: https://www.freertos.org/about-RTOS.html.

[33] Amazon Web Services, Inc, "Multitasking," Amazon Web Services, Inc, [Online]. Available: https://www.freertos.org/implementation/a00004.html.

[34] Amazon Web Services, Inc, "Tasks," Amazon Web Services, Inc, [Online]. Available: https://www.freertos.org/RTOS-task-states.html.

[35] D. Evans, "The Internet of Things. How the Next Evolution of the Internet," Cisco Internet Business Solutions Group, 2011.

[36] IEEE, "Towards a definition of the Internet of Things (IoT)," IEEE, 2015.

[37] TechTarget, "Internet of Things (IoT)," TechTarget, [Online]. Available: https://internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT.

[38] Seeedstudio, "Seeeduino GPRS," Seeedstudio, [Online]. Available: http://wiki.seeedstudio.com/Seeeduino_GPRS/. [Accessed 2018].

[39] A. Ronacher, "Welcome to Flask," The Pallet Center, 2010. [Online]. Available: http://flask.pocoo.org/docs/1.0/.

[40] D. Joan and V. Rijmen, The design of Rijndael: AES-the advanced encryption standard, Springer Science & Business Media, 2013.

[41] A. A. Pammu, K.-S. Chong, K. Z. L. Ne and B.-H. Gwee, "High Secured Low Power Multiplexer-LUT Based AES S-Box Implementation," in *Information Systems Engineering (ICISE), 2016 International Conference on. IEEE, 2016*, 2016.

[42] I. Karabey and G. Akman, "A Cryptographic Approach for Secure Client - Server Chat Application using Public Key Infrastructure," in *International Conference for Internet Technology and Secured Transactions*, 2016.

[43] D. Selent, "Advance Encryption Standard," *Insight: rivier academic journal,* vol. 6, no. 2, 2010.

[44] P. Patil, P. Narayankar, N. D. G and M. S. M, "A Comprehensive Evaluation of Cryptographic Algorithms: DES, 3DES, AES, RSA, and Blowfish," in *International Conference on Information Security & Privacy (ICISP2015)*, Nagpur, India, 2015.

[45] D. P. Mahajan and A. Sachdeva, "A Study of Encryption Algorithms AES, DES and RSA for Security," *Global Journal of Computer Science and Technology,* vol. 13, no. 15, pp. 15-22, 2013.