CALIFORNIA STATE UNIVERSITY, NORTHRIDGE

Networked Irrigation Controller using UDOO

A graduate thesis project submitted in partial fulfillment of the requirements

For the degree of Masters of Science in

Computer Science.

By

Prerak Diwan

May 2014

The graduate project of Prerak Diwan is approved:

_____ _____

Dr. John Noga Date

_____ _____

Dr. Adam Kaplan Date

_____ _____

Dr. Jeff Wiegley, Chair Date

California State University, Northridge

ACKNOWLEDGEMENT

I would like to thank my professor Dr. Jeffrey Wiegley for being my project guide and advisor. I am grateful to him for his continuous support and invaluable inputs he has been providing me through the development of the project. This work would not have been possible without his support and encouragement. I would like to thank him for introducing me to topic related to embedded applications and providing the best hardware possible to work with.

I would like to express my gratitude to all those who gave me the possibility to complete this Project. I want to thank Dr. Adam Kaplan and Dr. John Noga for their stimulating support.

Besides, I would like to thank the Department Chair Dr. Richard Covington of Computer Science for providing me with a good environment and facilities to complete this project. It gave me an opportunity to participate and learn about the technologies that were part of this project.

Table of Contents

List of Tables

List of Figures

ABSTRACT

IRRIGATION CONTROL SYSTEM USING UDOO WITH NETWORKING CAPABILITITES

By

Prerak Diwan

Masters of Science in Computer Science

Automated landscape irrigation systems have become quite common nowadays. Electronic controllers are used to control these systems. In this project, a residential irrigation controller system has been created and presented that uses a UDOO board to control and monitor the system. The system hosts a web-based user interface where a user has access to the entire functionality of the system. Two boards "ARDUINO DUE" and "UDOO" that were used during this project are discussed and compared here. The system allows a user to create multiple schedules for the water sprinklers and also monitor the current state of the controller.

CHAPTER 1

1. Introduction:

It is estimated that about 50-75% of the residential water usage is for irrigation. A residential landscape is divided into different zones. A zone is an area that is watered at the same time. Zones are created based on the irrigation needs such as lawns, flowerbeds, etc. All these zones combined make an irrigation system. An irrigation controller is a device that controls these irrigation systems [13, 14, 15]. These controllers, in general, are mechanical, electromechanical, electronic or computer-based [15]. These allow flexible scheduling of irrigation systems. The general operation of a typical residential irrigation controller involves scheduling options like days of week, run time, percent, and program. A typical residential irrigation controller is shown in Figure 1. A controller can be set for irrigation for a particular day, every other day, etc. [2].

A typical controller may have a custom option for setting frequency of irrigation. This custom option is useful during water restrictions, when irrigation is to be set for once or twice a week [2]. The amount of time that a certain irrigation zone is active may be set from few minutes to several hours. The amount of water to be released in a particular zone may be set in percentage [1]. This setting is available in most controllers and may be used to get the relative run time. Controllers have the capacity of storing multiple programs. For example “Program A” can be created to water six zones every other day for sixty minutes, while “Program B” can be used to water two zones every day for twenty minutes[2]. The lower-end controllers support simple mechanisms for users to specify one or two watering

programs supporting fewer zones. The more costly controllers can be used to schedule complex watering programs and support many more zones.

Figure 1. A typical residential irrigation controller [2]

1.1 Current residential solutions

There are a lot of residential irrigation controllers in the market from various manufacturers. Some of the top brands are Hunter Industries and Rain Bird. Hunter provides two types of residential irrigation controllers depending on the requirements of a household. One is the X-Core series, which is the low-end starting product that supports up to two, four, six and eight zones [4]. PCC series is the other type that supports up to fifteen zones at a time but it costs a little more. PCC series is considered to be lightly commercial for small office settings. Both the controllers can store three programs and have four start times per program. A typical Hunter X-Core controller costs around $125 approximately [4]. A full comparison chart of specifications is shown in Appendix A.

Rain Bird is another big name in the field of irrigation systems. It also has two solution series for a household. First is the ESP series, where ESP stands for "Extra Simple Programming" that can control up to six stations [5]. The Rain Bird SST ("Simple to Set") series controllers are another commonly used in residential irrigation systems. It has both indoor and outdoor models. The Rain Bird models are cheap compared to Hunter Models but then Hunter Models support more stations control. The low-end residential Rain Bird systems cost around $60 to below $100 for higher models [5].

Besides Hunter and Rain Bird, there are other products in the market from vendors like Orbit, Toro, Irritrol, Weathermatic, Cyber Rain, and many others. In general, all these products provide a variety of add-on features like weather sensors that increases the overall cost of the irrigation system. Besides this there are many limitations to these controllers. Most controllers support up to 12 zones at most and these are the higher end models that cost more than the average controllers. Besides this, most controllers allow very few watering programs with limited start times for each program [4, 5].

1.2 Proposed solution

In this project, an irrigation controller system is built that tries to address some of the issues with the current solutions available in the market. This system allows users to access a web user-interface that controls the functionality of the system. Following the features of the system:

- Supports up to 16 sprinkler zones.
- Secure Login to the system.
- Get current status of the sprinkler zones.
- Modify the current state of a sprinkler zone at any time.

- User can create day/time-scheduling programs that can later be assigned to any sprinkler zone available.
- Finally the irrigation controller system supports networking.

The irrigation controller system is built on an embedded open hardware system UDOO [10]. The board is connected to a passive irrigation shield, which in turn is connected to the sprinklers. Figure 2 shows the block diagram of the components of the irrigation system.

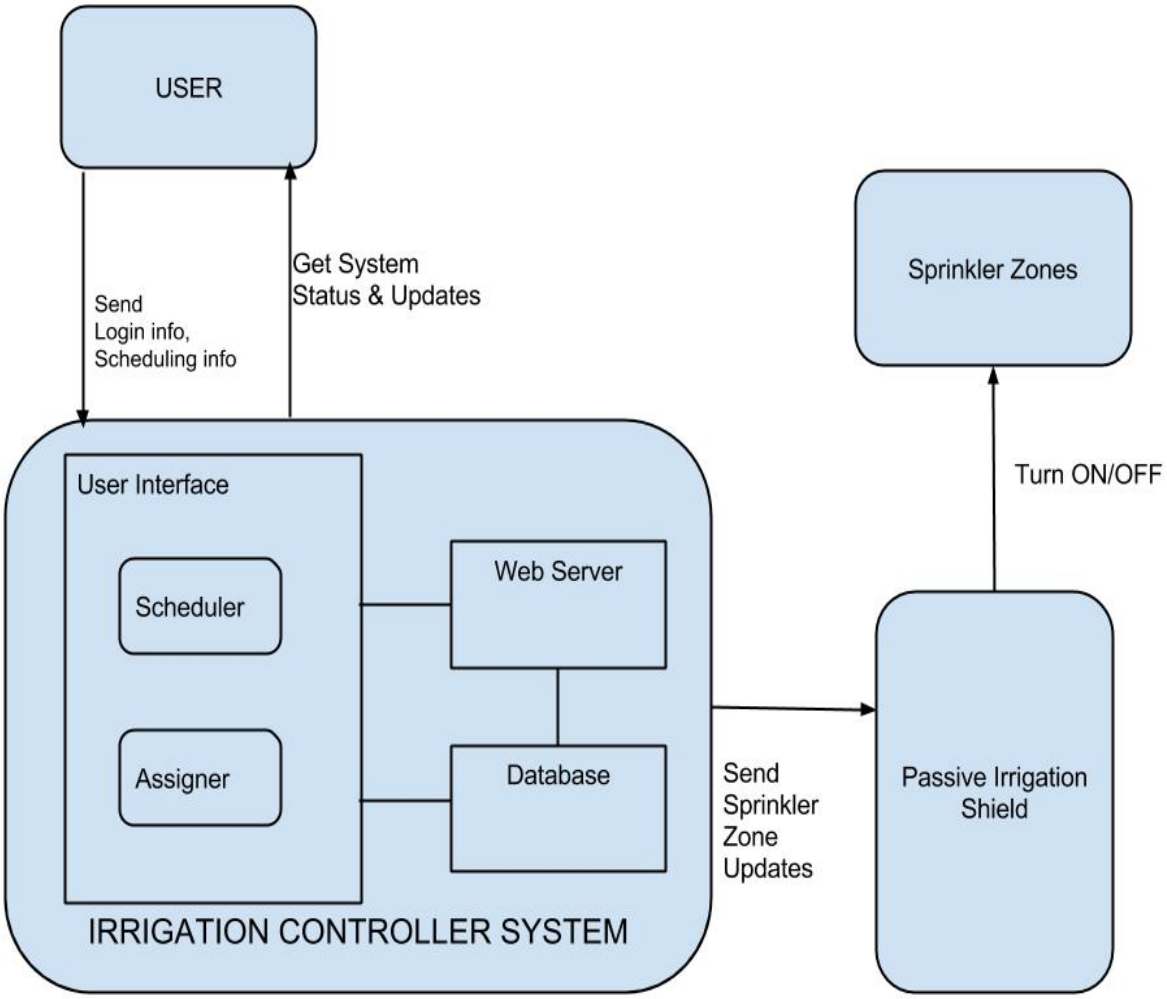


Figure 2. Block Diagram of major components of the Irrigation system

A validated user can login to the system to get access to the entire irrigation controller system. User gets system status and updates as and when requested. Scheduler is a list of schedules set up by the user. The list can be modified and/or deleted. Assigner is tool used to assign time-scheduled jobs created before to the sprinkler zones. Both these files I stored as tables in the database. The user can in anytime change the state of the sprinklers directly besides the scheduling program.

1.3 Thesis Organization

The thesis is organized as follows:

Chapter 2 discusses the technologies required and used during this project. This includes the software technologies as well as the hardware technologies. This will be a guideline to a person who wants to take this work and improvise for other related projects. Chapter 3 provides depth analysis of two ARM Microcontroller boards: ARDUINO DUE and UDOO. It discusses points taken into consideration using one over the other. Chapter 4 discusses the user interface in detail. Chapter 5 is the conclusion where the overview of the project and providing details of future work that can be done to further improve this project are discussed.

CHAPTER 2

2. Technologies

This section will include details about all the tools, whether software or hardware, that were used during this project. This may be a good guide for someone who wants to do this project from scratch or use this as a guide to develop a related similar kind of project.

2.1 Hardware requirements

2.1.1 UDOO

UDOO is a new project launched in a KickStarter campaign on April 9, 2013 and was fully subscribed within the first 60 days launch period [10]. UDOO is a single-board computer platform that combines dual or quad core ARM Freescale Cortex A9 iMX CPU and an ARDUINO DUE compatible board with a dedicated ARM processor. A typical UDOO is shown in Figure 3.

Figure 3. UDOO Quad supporting vast range of input and outputs.

2.1.2 ARDUINO DUE

Arduino Due is microcontroller board based on the Atmel SAM3X8E ARM Cortex-M3 CPU. It is the first Arduino board that runs a 32-bit ARM processor [9]. Table 1 shows the complete specifications of Arduino Due.

Microcontroller	AT91SAM3X8E
Operating Voltage	3.3V
Input Voltage	7-12V
Digital I/O pins	54 (of which 12 provide PWM output)
Analog Input pins	12
Analog Output pins	2
Total DC Output Current on all I/O lines	130 mA
DC Current for 3.3V Pin	800 mA
DC Current for 5V Pin	800 mA
Flash Memory	512KB all available for the user applications
SRAM	96 KB (two banks: 64KB and 32KB)
Clock Speed	84 MHz

Table 1. Summary of Arduino Due

2.1.3 Passive Irrigation Shield

A Passive irrigation shield is designed and developed by Dr. Jeffrey Wiegley that turns ON/OFF based on the output set on the Arduino Due board. The shield is currently a prototype to demonstrate the functionality of the controller and currently supports configuration of up to 8 sprinkler controllers. The shield has built-in LEDs to test the functionality of the system given input from the user.

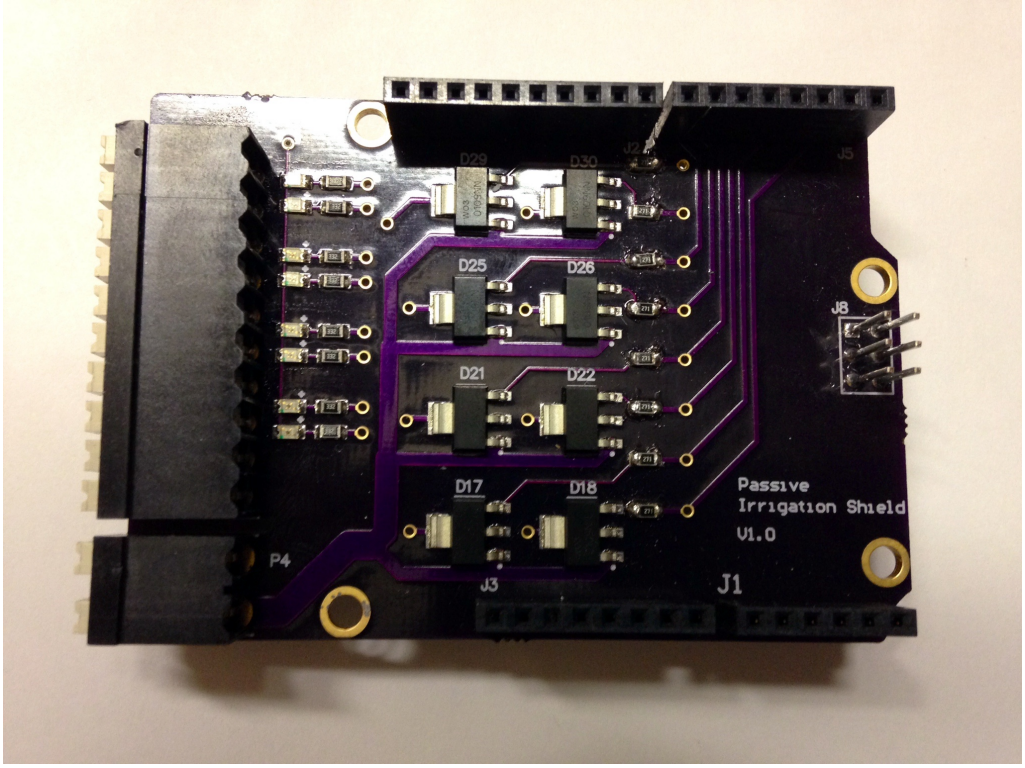


Figure 4. Passive Irrigation Shield V1 with built-in LEDs.

2.2 Software Requirements

2.2.1 Arduino Integrated Development Environment (IDE)

Arduino IDE is cross-platform application written in JAVA designed for newbie programmers. It is freely available for download under GPL license. It includes a code editor that is capable of automatic indentation of code, brace matching and syntax highlighting. Compiling and uploading code to the board can be done using a simple one-click. Arduino programs are either created in C or C++ [9]. It comes with a built-in software library called “Wiring” that makes most of the input/output operations easy to program. User needs to set two methods namely setup() and loop() to create a cyclic runnable program [9]. Link to get Arduino IDE can be found in Appendix B.

2.2.2 PHP

PHP stands for Hypertext Preprocessor is a widely used Open Source general-purpose scripting language that is especially suited for Web development and can be embedded into HTML[6]. Its syntax draws upon C, Java, and Perl, and is easy to learn. The main goal of the language is to allow web developers to write dynamically generated web pages quickly, but you can do much more with PHP [6]. PHP 5 is installed to develop the form-based user interface of the system. Instructions on how to install PHP can be found in Appendix B.

2.2.3 Linaro Ubuntu 12.0.4 LTS

Ubuntu is a Debian-based Linux operating system with Unity as its default desktop environment. The irrigation system is built on an ARM processor, so it requires specialized version of Ubuntu [12]. Linaro is a non-profit engineering organization that consolidates and optimizes the open source Linux software and tools for the ARM architecture [11]. UDOO comes with a Micro SD card readers and pre-installed Micro SD card. Linaro Ubuntu 12.0.4 LTS SD image is used to have bootable Ubuntu Linux environment to work with UDOO. Instructions on how to create a bootable SD image and link to Linaro Ubuntu 12.0.4 LTS can be found in Appendix B.

2.2.4 Apache Web Server

The Apache HTTP Server Project is an effort to develop and maintain an open-source HTTP server for modern operating systems including UNIX and Windows NT. The goal of this project is to provide a secure, efficient and extensible server that provides HTTP services in sync with the current HTTP standards [7]. Apache 2 is installed to host

the web based user interface for the user. Description on current versions of Apache server and how to install Apache 2 on Ubuntu can be found in Appendix B.

2.2.5 MySQL – Server

MySQL is the one of the most widely used open-source relational database management system. SQL stands for Structured Query Language. It is named after co-founder Michael Widenius's daughter, “My” [8]. It is a popular choice for use in web applications and is the important component of the widely used open-source software stack LAMP. LAMP is the acronym for Linux, Apache, MySQL, and PHP/Perl/Python. Free software projects that require a full functional database use MySQL. Instructions on how to install MySQL on Ubuntu are included in Appendix B.

2.3 Other Necessary Hardware

Besides the list of hardware and software tools mentioned above, there are certain necessary tools that are not mandatory but are recommended. LEDs are very important for testing some piece of new code on the board related to I/O functions. Secondly, a simple Volt/Amp meter is useful to check voltage to be passed as Input or Output to the boards. Every board has voltage range for its operation, so anything above the operating voltage can burn the processor on the board.

CHAPTER 3

3. Hardware

This chapter is focused on the two ARM processor boards: Arduino Due and UDOO in detail that were used during this project. Arduino Due has been out in the market for good amount of time while UDOO is quite new, launched in 2013.

3.1 Arduino Due

Arduino Due is built with 32-bit ARM core microcontroller. The summary of specifications of Arduino Due is seen in Table 1 in the previous chapter. Figure 5 shows the front and backside of the Arduino Due [9]. It is the only Arduino board that uses ARM processing. The ARM processor has several benefits over a traditional 8-bit microcontroller.

- A 32-bit core, that allows operations on 4 bytes wide data within a single CPU clock.
- CPU Clock at 84Mhz.
- 96 KBytes of SRAM.
- 512 KBytes of Flash memory for code.
- DMA controller that can relieve the CPU from doing memory intensive tasks.

It is important to note that Arduino Due operates on 3.3V unlike other Arduino boards that work on 5V. Connecting more voltages to the Arduino Due will damage the

Due. The board can take power from the USB connectors or the DC plug [9]. If using the DC connector, supply a voltage between 7V and 12V. The Arduino Due has an efficient switching voltage regulator, compliant with the USB host specification. If the *Native* USB port is used as host by attaching a USB device to the mirco-A usb connector, the board will provide the power to the device. External power is required whenever the board is used as a usb host.

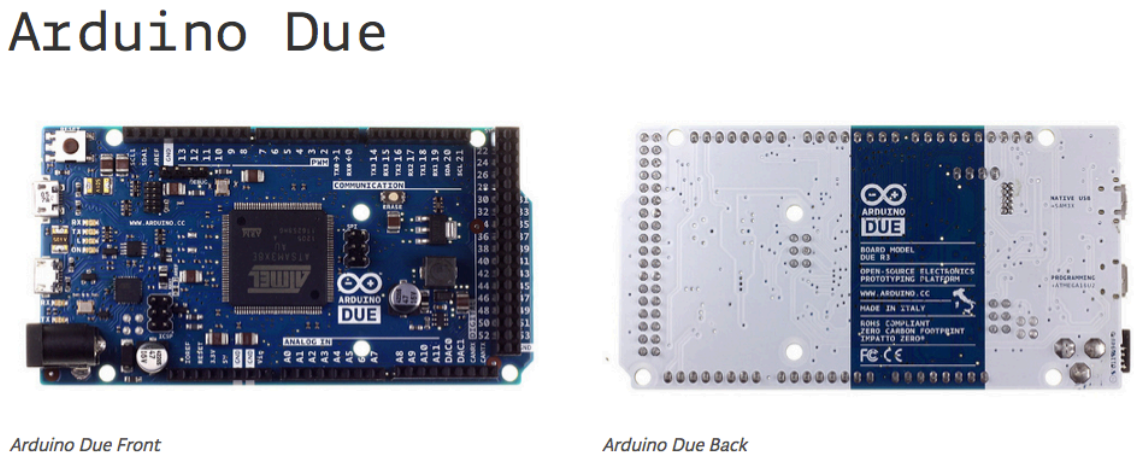


Figure 5. Front and Back view of Arduino Due

To begin programming the board one simply needs to connect it to a computer using the Programming port on the board and get started. For Mac and Linux users there are no drivers that are required to install. For users working on Windows environment a set of drivers come along when you download the Arduino IDE for Windows. Arduino IDE is the same for all Arduino boards and so it necessary to select the correct board before uploading the program sketches to the board. This can be done using the Tools menu in IDE [9].

The Arduino IDE is used to create programs called sketches. Each sketch can consist of C or C++ code written mainly under two functions named setup() and loop(). The setup function is used to initialize all the object settings that are going to be used in the entire program [9]. This functions runs only once at the start of the program execution. The loop function as the name suggests runs forever until the board is powered off. Figure 6 shows a simple "Hello World" kind of program to blink an LED continuously.

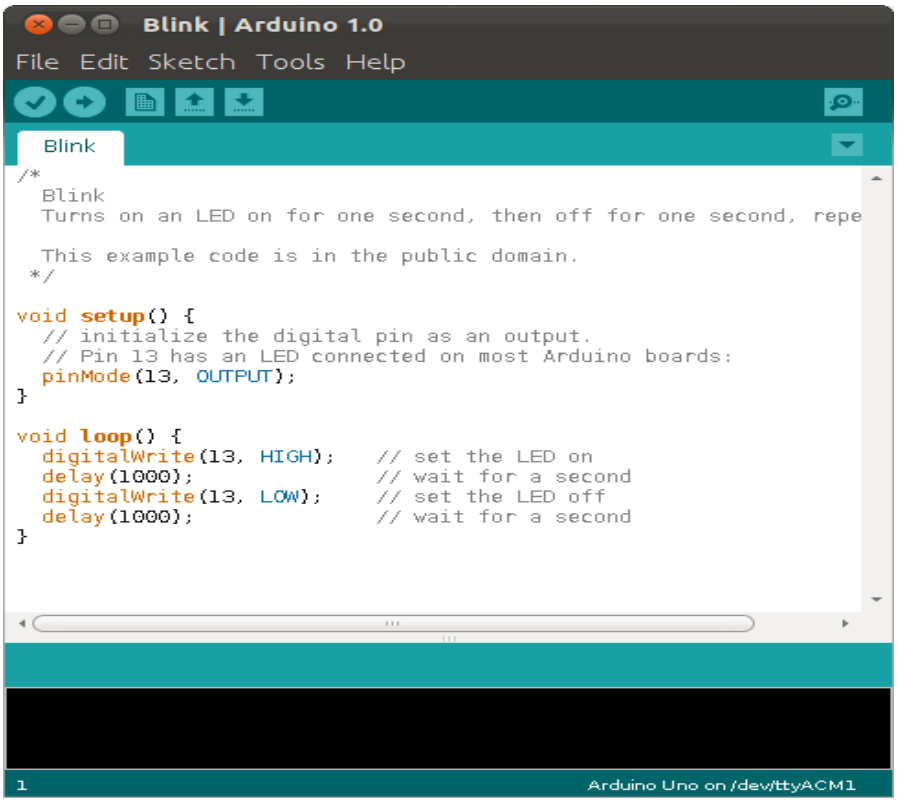


Figure 6. Simple LED blinking program written in Arduino IDE.

pinMode() is a function that takes two arguments: Pin number and Pin type i.e. either OUTPUT or INPUT. digitalWrite() specifies whether the output pin should be on or off. To turn the pin ON it sets the pin to HIGH that sends out 3.3V. Setting the pin to

LOW will withdraw that voltage. If no external LEDs are connected the board has a built-in LED that turns ON for the user to confirm that this piece of code worked. The LED will blink forever till the board is disconnected from power. Delay() is used to delay the program execution for the amount of milliseconds specified with it. In the above program delay() is set for 1 second every time it turns ON/OFF the output pin. This creates the blinking effect that we see on the LED.

3.1.1 Arduino Libraries

Arduino Due supports a lot of standard libraries that comes with the Arduino IDE installation package. Any time you want to use one of these libraries, SELECT Sketch→Import Library. Following the libraries that can be used by any Arduino board:

- EEPROM – reading and writing to permanent storage.
- Ethernet – connecting Arduino to Internet using an Arduino Ethernet shield.
- Firmata - for communicating with applications on the computer using a standard serial protocol.
- GSM - for connecting to a GSM/GRPS network with the GSM shield.
- LiquidCrystal - for controlling liquid crystal displays (LCDs)
- SD - for reading and writing SD cards
- Servo - for controlling servo motors
- SPI - for communicating with devices using the Serial Peripheral Interface (SPI) Bus

- SoftwareSerial - for serial communication on any digital pins. Version 1.0 and later of Arduino incorporate Mikal Hart's NewSoftSerial library as SoftwareSerial.
- Stepper - for controlling stepper motors
- TFT - for drawing text, images, and shapes on the Arduino TFT screen
- WiFi - for connecting to the internet using the Arduino WiFi shield
- Wire - Two Wire Interface (TWI/I2C) for sending and receiving data over a net of devices or sensors.

Besides the libraries mentioned above there are libraries that specially work only with the Arduino Due:

- Audio – It allows play audio files from a SD card.
- Scheduler – This help manage multiple non-blocking tasks.
- USBHost – Helps communicate with USB peripherals like mice and keyboards.

A large number of developers have contributed custom libraries to the Arduino Community that can be downloaded separately as they are not included with the Arduino IDE package. Arduino Community even guides programmers on how to create a library and make it available to all community users. These libraries support communication (networks and protocols), sensing, displays and LEDs, audio and waveforms, motors and PWM, timing and other utilities.

3.2 UDOO

UDOO was launched as a KICKSTARTER program to raise $500,000. The response was amazing from the engineering community that the funds were raised before the ending 60-day period. UDOO is mini computer that can run either Linux or Android on it. The board is designed to provide a flexible environment that allows exploring the new frontiers of the Internet of Things. UDOO is promoted as the single board capable of doing 4 times the Raspberry Pi and Arduino Due [10]. UDOO allows you to switch between Linux and Android in a few seconds, simply by replacing the Micro SD card and rebooting the system. To get started with UDOO a user needs a keyboard and mouse, HDMI display and an Ethernet connection. Figure 7 shows a typical UDOO board ready to use.

Figure 7. UDOO Quad connected with Keyboard, Mice, Ethernet connection and a HDMI display.

3.2.1 Running Linux

UDOO is a standalone computer based on a Freescale i.MX 6 CPU. It runs an optimized version of Linux Ubuntu called Linaro that is capable of running everything a user needs for programming. The user can use the Arduino IDE to build & upload your

sketches to the Arduino-compatible embedded board, without additional/external cable connections [10].

3.2.2 Running Android

Android 4.2.2 Jelly Bean runs smootly on UDOO giving you all the features of an Android device (further Android releases will be available at the time of distribution). Apps interface with Arduino-compatible embedded board through Accessory Development Kit (ADK) connection for building accessories and smart devices based on Android [10].

3.2.3 Full Support for Arduino

UDOO has a Arduino Due ARM Processor built on it that functions exactly the same as Arduino Due. UDOO is compatible with all the sketches, tutorials and resources available on the Arduino community as well as all the shields, sensors and actuators for Arduino Due available on the market. UDOO's I/O pins are 3.3V compliant so providing shields with higher voltage say 5V will burn the board [10].

3.2.4 Specifications

UDOO gives the power and flexibility to users for any kind of projects. The major specifications for UDOO are as follows:

- Freescale i.MX 6 ARM Cortex-A9 CPU Dual/Quad core 1GHz
- Integrated graphics, each processor provides 3 separated accelerators for 2D, OpenGL® ES2.0 3D and OpenVG™
- Atmel SAM3X8E ARM Cortex-M3 CPU (same as Arduino Due)

- RAM DDR3 1GB
- 76 fully available GPIO
- Arduino-compatible R3 1.0 pinout
- HDMI and LVDS + Touch (I2C signals)
- Ethernet RJ45 (10/100/1000 MBit)
- WiFi Module
- Mini USB and Mini USB OTG
- USB type A (x2) and USB connector (requires a specific wire)
- Analog Audio and Mic
- SATA (Only Quad-Core version)
- Camera connection
- Micro SD (boot device)
- Power Supply 12V and External Battery connector

3.3 Arduino Due VS UDOO – Board selection.

UDOO supports all the functionalities of the Arduino Due plus it is a mini computer running Linux. This means a user can install LAMP on the board itself without the need to keep the server on any another computer. Also every time the user makes changes to the sketch program it has be recompiled and pushed back to the Arduino processor that takes a significant amount of time. On the other side, UDOO quad core processor has full control over the GPIOs on the board. Every time a programmer needs to make changes to I/O pins it just has to make changes to the Linux system class files that pushes the changes to the board immediately.

CHAPTER 4

4. SOFTWARE

This chapter describes the user-interface made available to the user from the UDOO board. The main components of the user interface includes logging in to the system, creating irrigation schedules, assigning schedules to a particular sprinkler zone and storing this information in a database.

4.1 USER INTERFACE

4.1.1 User Authentication

It is important that only authorized users can access the irrigation controller system, as you don't want anybody else to control your residential sprinklers. New users can register to the system by providing username, password, and email. To assure security to the user, a password hash and salt are generated that are random and unique. These added hashes and salts are useful for secured login experience. Figure 8(a) and Figure 8(b) respectively, shows the registration form and the login form available to the user. Every time a user logs in a user session is created that gets destroyed only when user logs out.

Figure 8(a). User Registration Form.

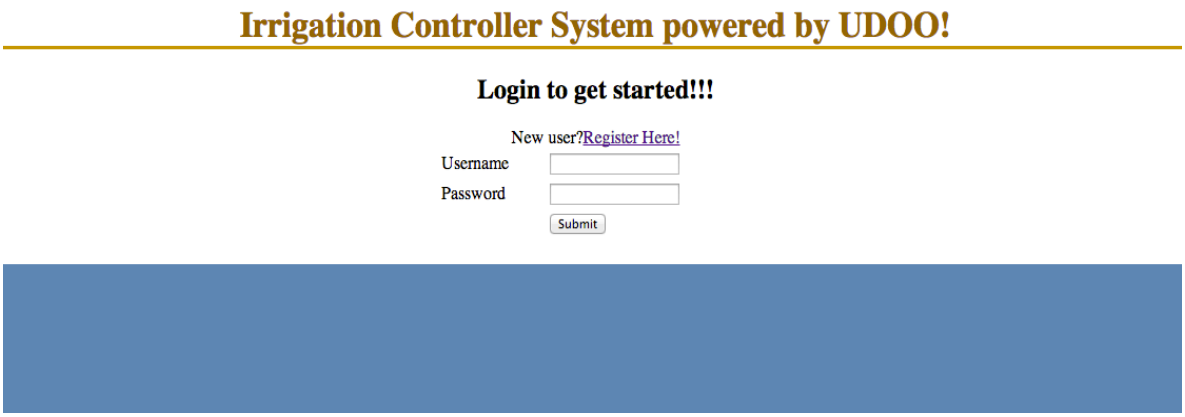


Figure 8(b). Login form.

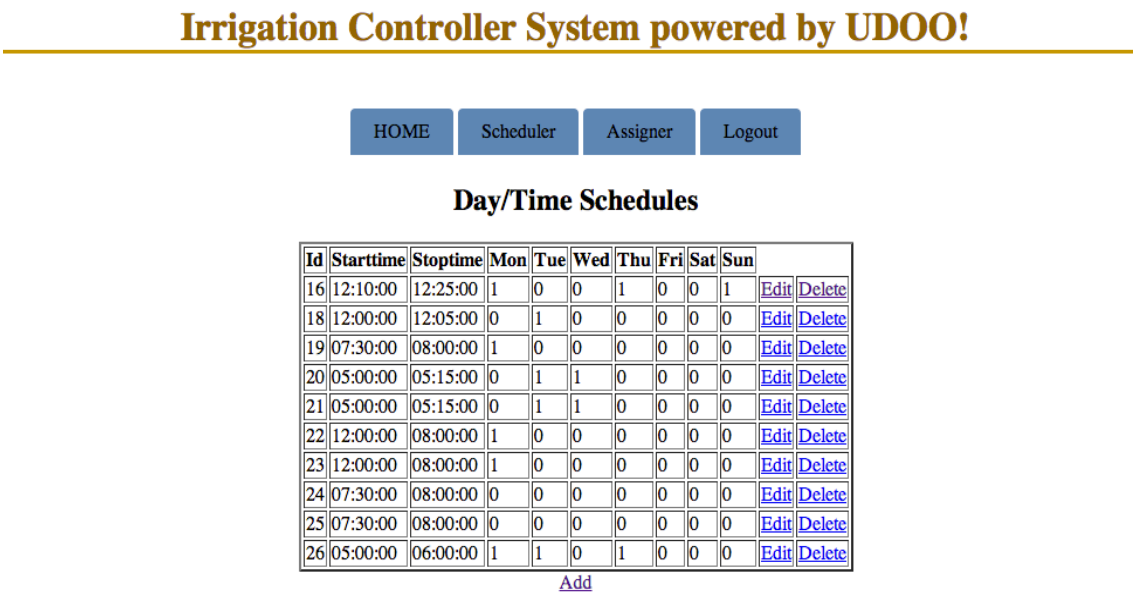
4.1.2 Scheduler Tool

Once the user is authenticated, user can create scheduling programs for the irrigation system. User can select days of a week, along with time of a day and save it. User can create as many schedules as desired or required for the residential setting. Here, The user just creates the schedules but they still are required to assign to any of the available sprinklers connected to the system. This information is stored in a separate table "daystime" with days of week and time of the day where changes can be made very easily. User can remove or edit the already created schedules as and when required. Figure 9(a) shows the page where user views the current schedules and figure 9(b) shows the create new/edit page.

4.1.3 Assigner Tool

User can use any of the schedules previously created using the scheduler program. User gets the option to select any of the schedules and assign it to any sprinkler motors. To do this user has to first select a sprinkler from the drop down list and then select the

list of schedules that are required to run on a sprinkler. Figure 10 displays the assigner tool.



Id	Starttime	Stoptime	Mon	Tue	Wed	Thu	Fri	Sat	Sun		
16	12:10:00	12:25:00	1	0	0	1	0	0	1	Edit	Delete
18	12:00:00	12:05:00	0	1	0	0	0	0	0	Edit	Delete
19	07:30:00	08:00:00	1	0	0	0	0	0	0	Edit	Delete
20	05:00:00	05:15:00	0	1	1	0	0	0	0	Edit	Delete
21	05:00:00	05:15:00	0	1	1	0	0	0	0	Edit	Delete
22	12:00:00	08:00:00	1	0	0	0	0	0	0	Edit	Delete
23	12:00:00	08:00:00	1	0	0	0	0	0	0	Edit	Delete
24	07:30:00	08:00:00	0	0	0	0	0	0	0	Edit	Delete
25	07:30:00	08:00:00	0	0	0	0	0	0	0	Edit	Delete
26	05:00:00	06:00:00	1	1	0	1	0	0	0	Edit	Delete

Figure 9(a). Scheduler page displaying the days of week with start and stop times.

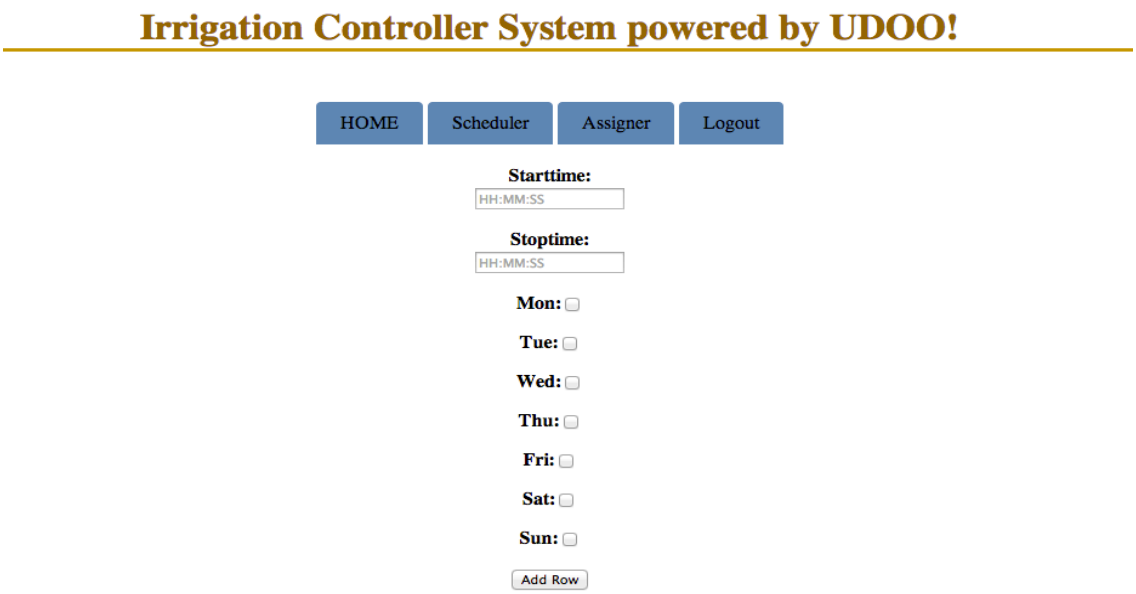
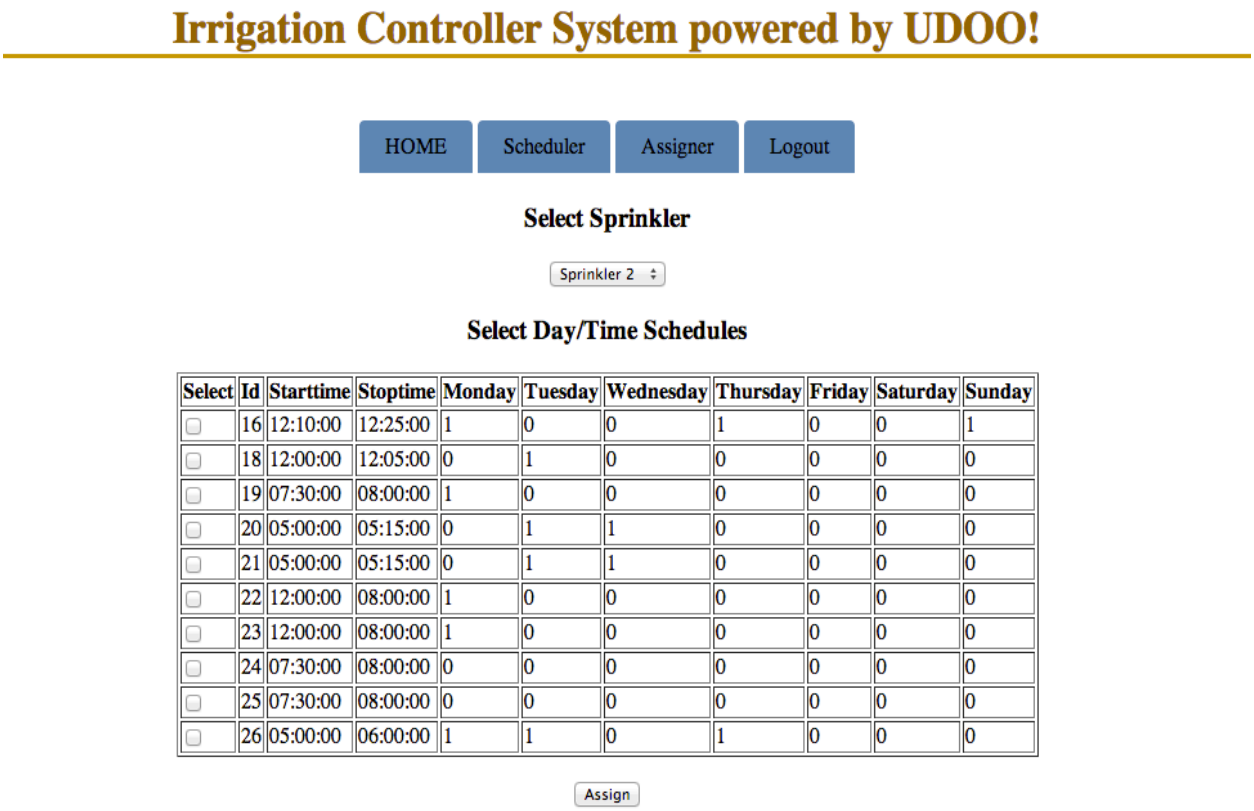


Figure 9(b). Add new schedules and/or edit current schedules.



Select	Id	Starttime	Stoptime	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
☐	16	12:10:00	12:25:00	1	0	0	1	0	0	1
☐	18	12:00:00	12:05:00	0	1	0	0	0	0	0
☐	19	07:30:00	08:00:00	1	0	0	0	0	0	0
☐	20	05:00:00	05:15:00	0	1	1	0	0	0	0
☐	21	05:00:00	05:15:00	0	1	1	0	0	0	0
☐	22	12:00:00	08:00:00	1	0	0	0	0	0	0
☐	23	12:00:00	08:00:00	1	0	0	0	0	0	0
☐	24	07:30:00	08:00:00	0	0	0	0	0	0	0
☐	25	07:30:00	08:00:00	0	0	0	0	0	0	0
☐	26	05:00:00	06:00:00	1	1	0	1	0	0	0

Figure 10. Assigner tool to assign schedules to sprinklers.

4.1.4 System Current Status

User gets access to the current state of the irrigation system as soon he is authenticated. User can modify the current state of a sprinkler as and when the user wishes without being required to create any schedules. Figure 11 gives access to the user a set of toggle buttons corresponding to the sprinklers. User can turn on/off a sprinkler from this page. To turn on/off a sprinkler, the page script manipulates the GPIO pins of the UDOO board. The on/off buttons creates hidden beacon objects that opens the GPIO pin files, modifies it and then closes it. Since the beacons are hidden, it does not affect the toggle page appearance.

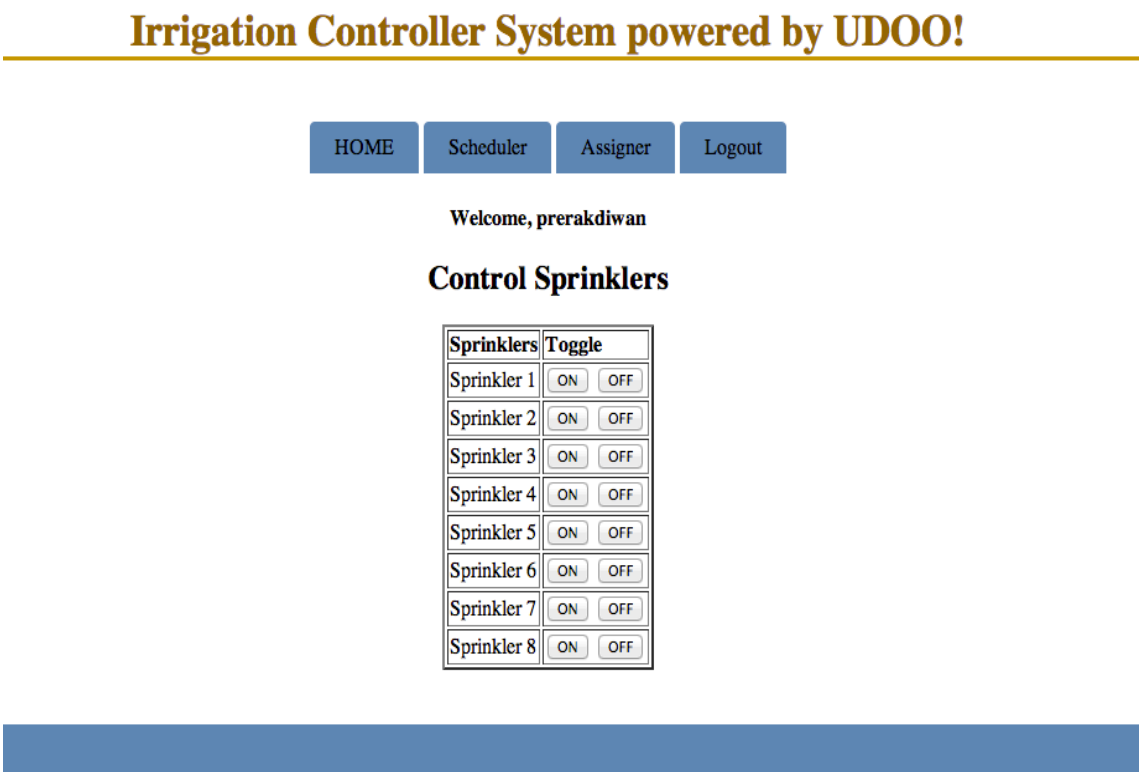


Figure 11. Home/Control Sprinklers page.

4.1.5 Cronjob Scheduling.

The entire system is running on Ubuntu OS installed on the UDOO board. Two cronjobs are created on the system. The first job is to change access permissions of the files that are linked to the GPIO pins of the board. This is necessary as these files are virtual links to the board pins that resets every time the board is rebooted or turned off. The second cronjob is to run the scheduling PHP script that is connected to the database. This script checks the server time and compares it to the schedules created by the user. If the server time is in range then based of it the board turns on/off a sprinkler. This cronjob is set to run every minute to see whether the system needs to turn on/off a particular sprinkler.

4.2 UDOO I/O pins handling

This section covers the GPIO pin management done in UDOO. The naming of the GPIO is different from the naming printed on the actual board. Table 2 just display values of pin number and corresponding GPIO for the pins used by the passive irrigation shield.

PIN NUMBER	GPIO NUMBER
1	112
3	16
4	17
5	18
6	41
7	42
8	21
9	19

Table 2. Pin number and corresponding GPIO number for the irrigation shield connected to UDOO

These GPIO pins can be managed from the /sys/class/gpio/ folder which contains subfolder for all 54 I/O pins. Each subfolder has two files “direction (in/out)” and “value (0/1)”. These files can be modified using UBUNTU’s echo and cat commands [10].

CHAPTER 5

5. Conclusion, Achievements and Work for the future

The irrigation controller setup can control up to 8 sprinkler settings that can be turned on or off manually as and when desired by the user. Since UDOO is running its own Apache web server, the user from anywhere having access to Internet can access the controller and modify the current system settings. User can create, as many schedules as desired, as there are no space limitations. These schedules can be assigned to an available sprinkler controller that is not previously assigned a schedule overlapping its current assignment. Currently, the system can turn on a total maximum of three sprinkler motors at a single given time. The prototype irrigation shield just messes up when the program tries to turn more than three sprinkler motors.

Irrigation Controller systems play a vital role today both in commercial setting and residential setting [15]. About 70% of water usage of an average household is for irrigation. Irrigation controllers have developed starting from being mechanical to electronic and finally computerized that can be managed from a central location [13]. With the development of products like UDOO and other open source microcontrollers, there is no limit to the vast range of applications that can be created. In this project both UDOO and Arduino DUE were explored and used. It is amazing how UDOO makes things easier and faster compared to its competition in the market.

REFERNCES

[1] Muhammad Umer Tariq, Hasan Arshad Nasir, Abubakr Muhammad, and Marilyn Wolf. 2012. Model-Driven Performance Analysis of Large Scale Irrigation Networks. In *Proceedings of the 2012 IEEE/ACM Third International Conference on Cyber-Physical Systems* (ICCPS '12). IEEE Computer Society, Washington, DC, USA, 151-160. DOI=10.1109/ICCPS.2012.23 http://dx.doi.org/10.1109/ICCPS.2012.23

[2] Michael D. Dukes and Dorota Z. Haman. "Operation of Residential Irrigation Controllers". http://edis.ifas.ufl.edu/pdffiles/AE/AE22000.pdf . Last retrieved May 6, 2014

[3] Jishun Jiang; Lina Liu; Tian Jiang, "Notice of Retraction
Study of a New Type Field Irrigation Intelligent Controller," *Information Engineering and Computer Science (ICIECS), 2010 2nd International Conference on* , vol., no., pp.1,4, 25-26 Dec. 2010
doi: 10.1109/ICIECS.2010.5678226

[4] About Hunter – Commercial and Residential Irrigation Systems. http://www.hunterindustries.com -Last retrieved May 6, 2014

[5] About Rain Bird Irrigation Systems. http://www.rainbird.com/homeowner/index.htm

- Last retrieved May 6, 2014

[6] About PHP - PHP Manual. http://www.php.net/manual/en/preface.php - Last retrieved May 6, 2014

[7] About Apache HTTP Server Project. http://httpd.apache.org/ - Last retrieved May 6, 2014

[8] About MySQL Server.
http://www.oracle.com/us/products/mysql/overview/index.html - Last retrieved May 6, 2014

[9] About Arduino Due. http://arduino.cc/en/Main/arduinoBoardDue - Last retrieved May 6, 2014

[10] About UDOO http://www.udoo.org/ - Last retrieved May 6, 2014

[11] About Linaro http://www.linaro.org/ - Last retrieved May 6, 2014

[12] About UBUNTU http://www.ubuntu.com/about/about-ubuntu - Last retrieved May 6, 2014

[13] Fazackerley, S.; Lawrence, R., "Reducing turfgrass water consumption using sensor nodes and an adaptive irrigation controller," *Sensors Applications Symposium (SAS), 2010 IEEE* , vol., no., pp.90,94, 23-25 Feb. 2010
doi: 10.1109/SAS.2010.5439386

[14] M. B. Haley, M. D. Dukes, and G. L. Miller, "Residential Irrigation Water Use in Central Florida," Journal of Irrigation and Drainage Engineering, vol. 133, no. 5, pp. 427-434, 2007. [Online].Available: http://link.aip.org/link/?QIR/133/427/1 - Last retrieved May 6, 2014

[15] T. V. der Gulik, "Landscape Sprinkler Irrigation Scheduling Calculator Users Guide," Irrigation Industry Association of British Columbia, Tech. Rep., February 2009. - Last retrieved May 6, 2014

Appendix A

1. Rain Bird SST current residential models comparison chart

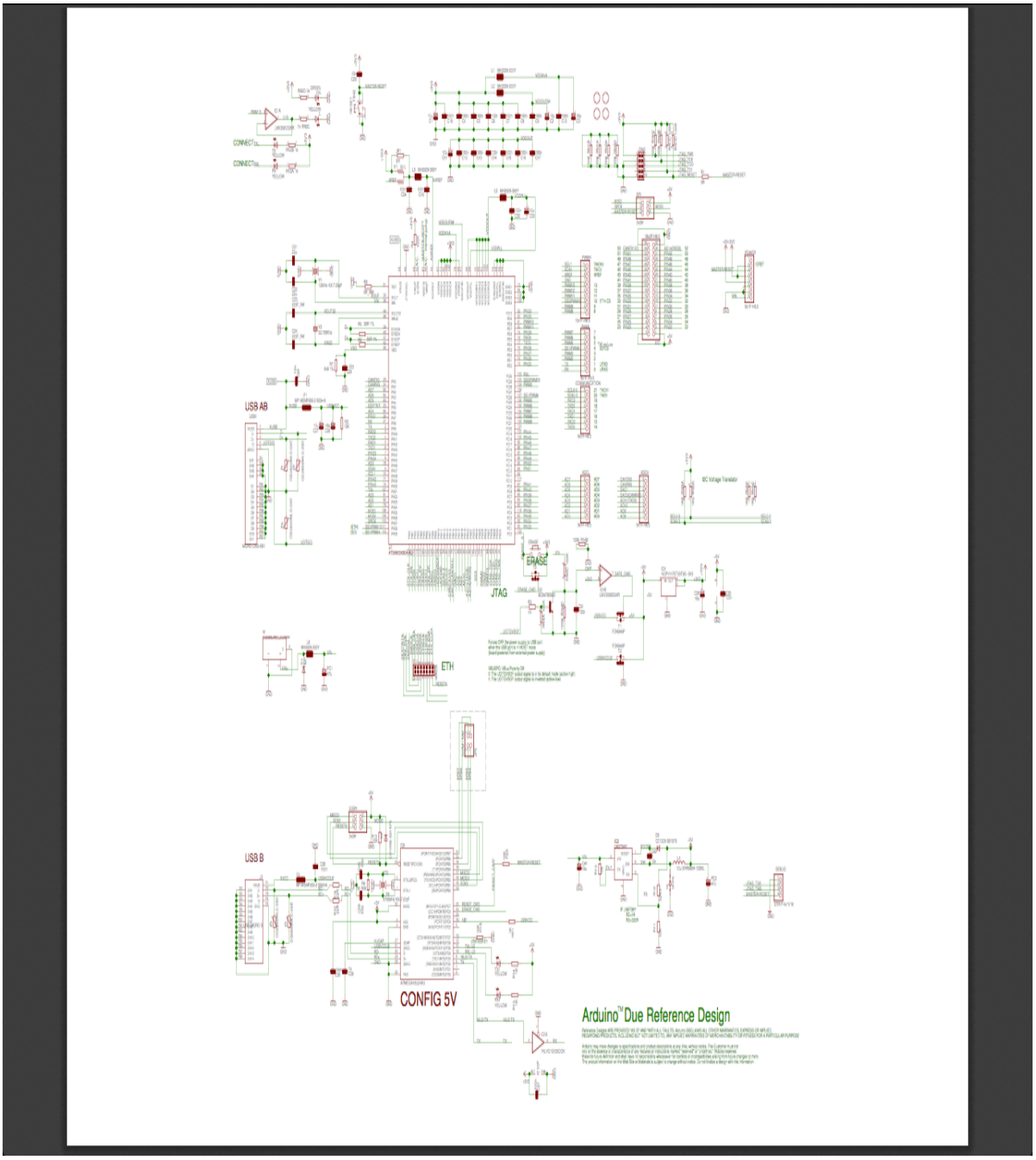
Timers					
Primary Applications	SST-400i	SST-600i	SST-900i	SST-600o	SST-1200o
4 Stations	●				
6 Stations		●		●	
9 Stations			●		
12 Stations					●
Water Budgeting	●	●	●	●	●
Drip Compatible	●	●	●	●	●
Pump Control Relay	●	●	●	●	●
Master Valve	●	●	●	●	●
Outdoor installation				●	●

2. Hunter Residential Irrigation Solutions Comparison Chart

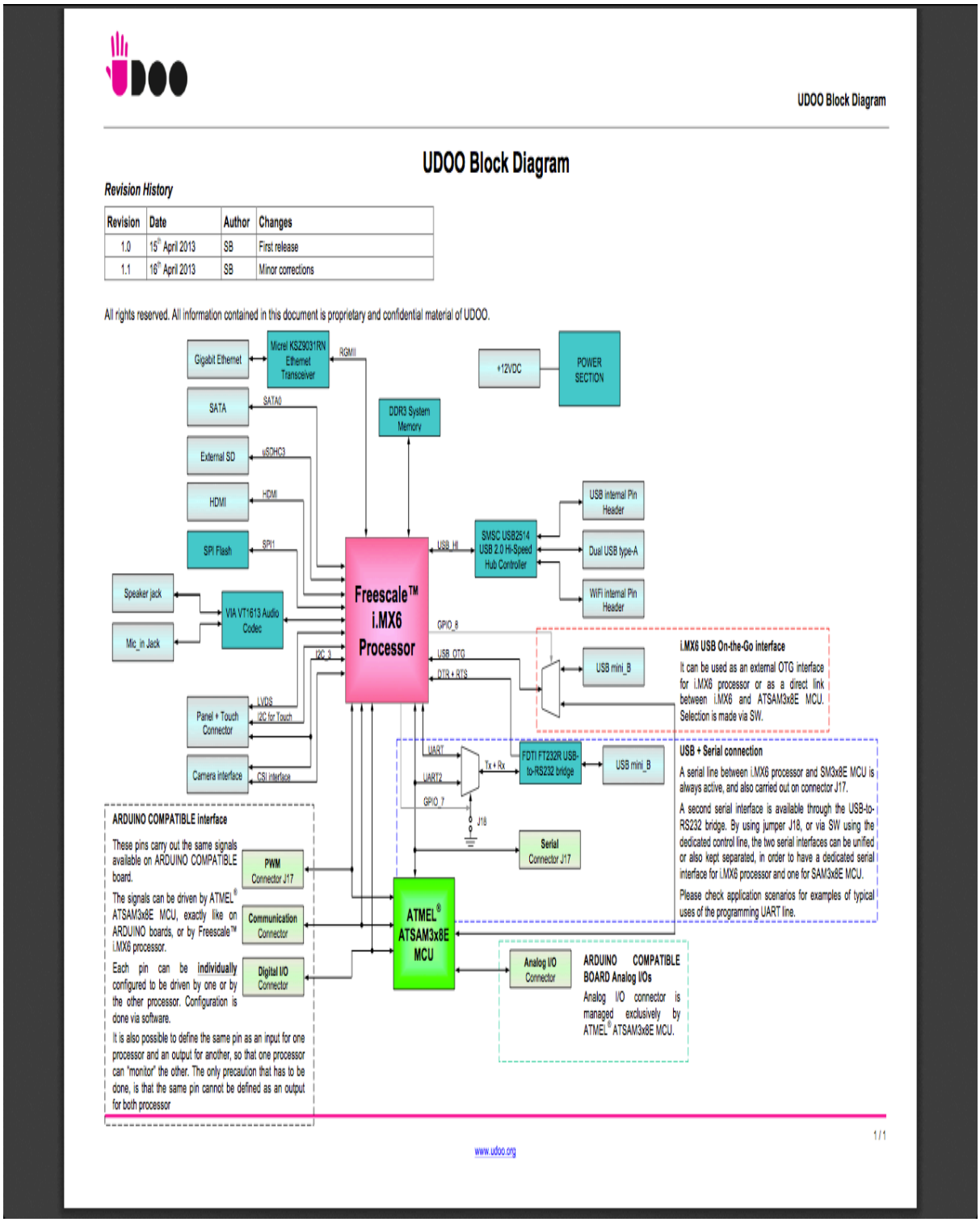
	X-CORE	PRO-C	PCC
QUICK SPECS			
Number of Stations	2, 4, 6, 8	3 to 15	6, 9, 12, 15
Type*	Fixed	Modular	Fixed
Number of Programs	3	3	3
Start Times Per Program	4	4	4
Warranty	2 Years	2 Years	2 Years
FEATURES			
Enclosure Type	Plastic Indoor Plastic Outdoor	Plastic Indoor Plastic Outdoor	Plastic Indoor Plastic Outdoor
Transformer Types	Plug In Outdoor	Plug In Outdoor	Plug In Outdoor
Hunter Solar Sync ET Sensor Compatible	Yes	Yes	Yes
Central Control Compatible	No	No	No
Remote Control Compatible	Yes	Yes	Yes
Flow Sensor Compatible	No	No	No
Rain/Freeze Sensor Compatible	Yes	Yes	Yes
Number of Sensor Inputs	1	1	1
Max Station Run Time (Hrs)	4	6	6
APPLICATION			
Residential	✔	✔	✔
Light Commercial		✔	✔
Commercial			

* Fixed or modular indicates the controllers ability to expand the number of stations from a base count.

1. Arduino Due Full Schematic



2. UDOO Block Diagram



UDOO Block Diagram

UDOO Block Diagram

Revision History

Revision	Date	Author	Changes
1.0	15th April 2013	SB	First release
1.1	16th April 2013	SB	Minor corrections

Appendix B

Some useful Download Links

1. Linaro Ubuntu 12.04 LTS, **Select the version for UDOO Quad.**

   http://www.udoo.org/downloads/

2. Instructions on how to install PHP, MySQL, Apache on Linaro Ubuntu can be found here.

   http://www.udoo.org/ProjectsAndTutorials/udoo-web-server/

3. Arduino IDE

   http://arduino.cc/en/Main/Software