Design Plan

Modular Garden Monitoring System
EECS Senior Design 2021

October 20, 2020

Team CE12
Sadie Gladden, Eric Krenz, Zuguang Liu, Alan Trester

Technical Advisor
Dr. Zachariah Fuchs

University of Cincinnati
College of Engineering and Applied Science
EECS 5031 CompE Senior Design I - 001

Contents

1 Problem Statement

Lawns and gardens are one of most essential elements for the typical American home. A survey conducted by National Association of Landscape Professionals in 2019 shows that 79 percent of American families value lawns when renting or buying a home, and about one in three Americans garden in their yards multiple times a week[1].

Consequently, there is a constantly high demand of water for use in lawns and garden. Per the United States Environmental Protection Agency, about 48 gallons of water is devoted for this use per family per day. Across America, nearly 1/3 of all residential water is used for landscaping irrigation totaling an estimated 9 billion gallons per day[2]. In a world undergoing climate change with consistent annual water shortages and wildfires in many parts of the world, wasteful water usage is simply unacceptable.

A 21st-century solution is needed to help new homeowners care for their lawns and gardens in a more informed and effective way while reducing the amount of wasteful water usage that is accounted for by residential lawn care and irrigation.

Originated from the Internet of Things (IoT) concept, the Modular Garden Monitoring System (MGMS) is a solution that will be able to provide real-time and historical information about environmental conditions. Simply having detailed information on-hand will allow homeowners to make more informed decisions on the types of plants to keep in their gardens as well as when and how much to water them. Internet connectivity can take decision making to the next level by being able to crowd-source gardening recommendations and consider local weather predictions for watering. Further system expansions can introduce features such as automatic watering to take work from homeowner's shoulders while reducing human error in the garden care process. Finally, a smart design will allow the system to be flexible and applicable in a variety of scenarios varying with garden size and irrigation needs and even between residential and industrial settings.

2 Objectives and Constraints

A garden monitoring system such as the one we are proposing is not a novel idea: several products already exist within the consumer and industrial farming markets with similar approaches towards data collection. The Onset HOBOnet system is a web-enabled data-collection solution for industrial farmers. While these systems are very popular and provide good results, with accessible user interfaces and informative data visualization, they are too expensive for consideration by homeowners and don't have the necessary features such as garden suggestions to be applicable in that market [3]. The Edyn Garden Sensor was a consumer-targeted system that aimed to tackle the same problems as the MGMS, unfortunately the product was burdened with limited modularity and expandability as well as a poorly designed app interface [4]. Characteristics of both products are analyzed and, along with interviews and the team's own expectations, are used to set a reasonable objectives baseline for the new system.

2.1 Attribute Table

Figure 1 shows a completed attribute table for the MGMS. Attributes were decided by the project team with insight from research into the products mentioned above as well as an interview of an agricultural engineering professional which helped gained insight into helpful functions and realistic expectations for system functionality. Other constraints and functions were chosen to support the proposed system goals. The most significant attributes that determines the success of the project are:

a. High importance placed on accuracy and accessible UIs, which are the most characteristic feature of the HOBOnet industrial system.

b. Expandability and Modularity, which was identified as a shortcoming of the Edyn garden system.

c. Low Cost to match the Edyn's reasonable price tag between 75-150 per system module.

Attribute Table				
Characteristic	Objective	Constraint	Function	Means
Hardware				
Measures Environmental Conditions			✓	
Accurate	✓			
Expandable and Modular	✓		✓	
Waterproof / Weatherproof		✓		
Inexpensive	✓	✓		
Wireless Communication & Power		✓		✓
Easy to Set Up Outside	✓	✓		
Low Power Consumption		✓		✓
Software				
Easy to Use, Intuitive UI	✓			
Saves Historical Data			✓	
Shows Real-Time Conditions			✓	
Links to Recommended Growing Conditions			✓	
Predicts Weather			✓	
User Configurable	✓		✓	

Figure 1: MGMS Attribute Table. Attributes were chosen with insight from a variety of sources including interviews and product market research.

2.2 Objective Tree

Figures 2 and 3 show expanded objective trees for the hardware and software components of the MGMS respectively. Objectives were chosen from the attribute table in figure 1. Then, main categories towards which these objectives are contributing were identified for greater organization of the project goals. These goals are:

Hardware:

- Marketable
- Useful
- Reliable

Software:

- Accessible
- Customizable
- Helpful

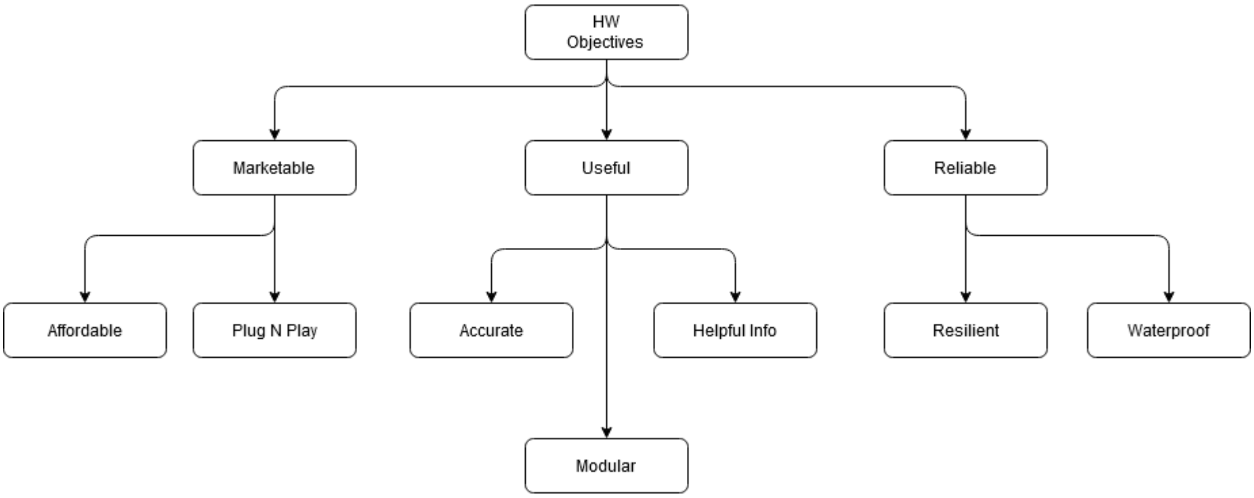

Figure 2: MGMS Hardware objectives tree grouped into three component goals: Marketable, Useful, and Reliable.

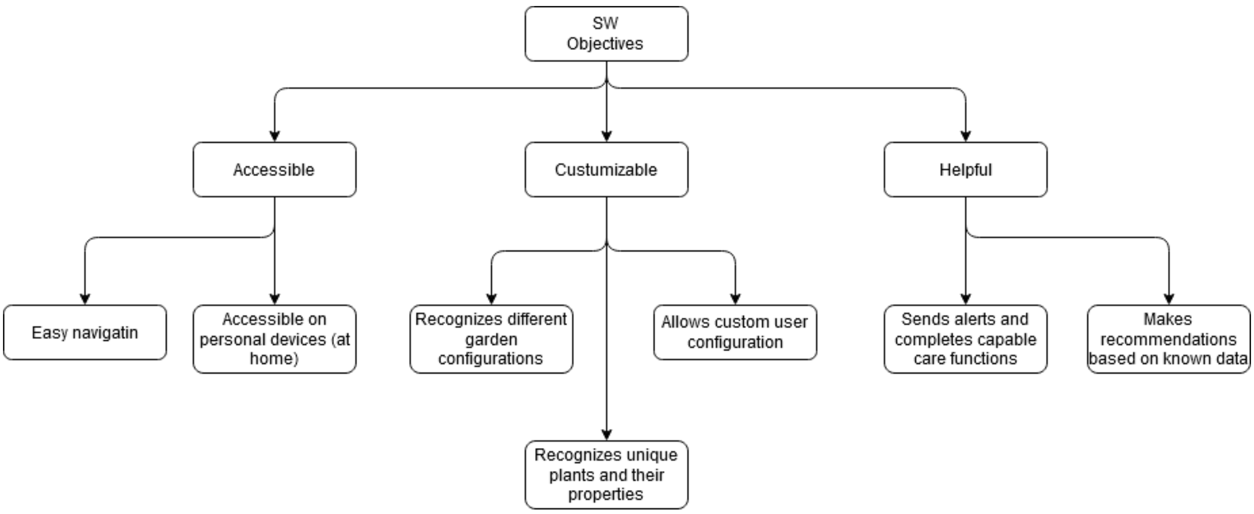

Figure 3: MGMS Software objectives tree grouped into three component goals: Accessible,Customizable,Helpful.

2.3 Impact-Effort Matrix

Impact-effort matrix is a tool that helps organize priorities so that practically limited time and resource can be used effectively. Tasks in the early stage of the project are placed in such matrix in Figure 4, where it becomes obvious that meeting the basic design requirements and goals should be the first priority. It is also important to keep good documents and communications, but since it does not require much effort, a focused time slot per week should be enough investment. As far as the low-impact items, implementing "good-to-have" features is the last priority provided that there is still time and resource to spare, whereas features out of scope of the project will not be considered in the development at all.

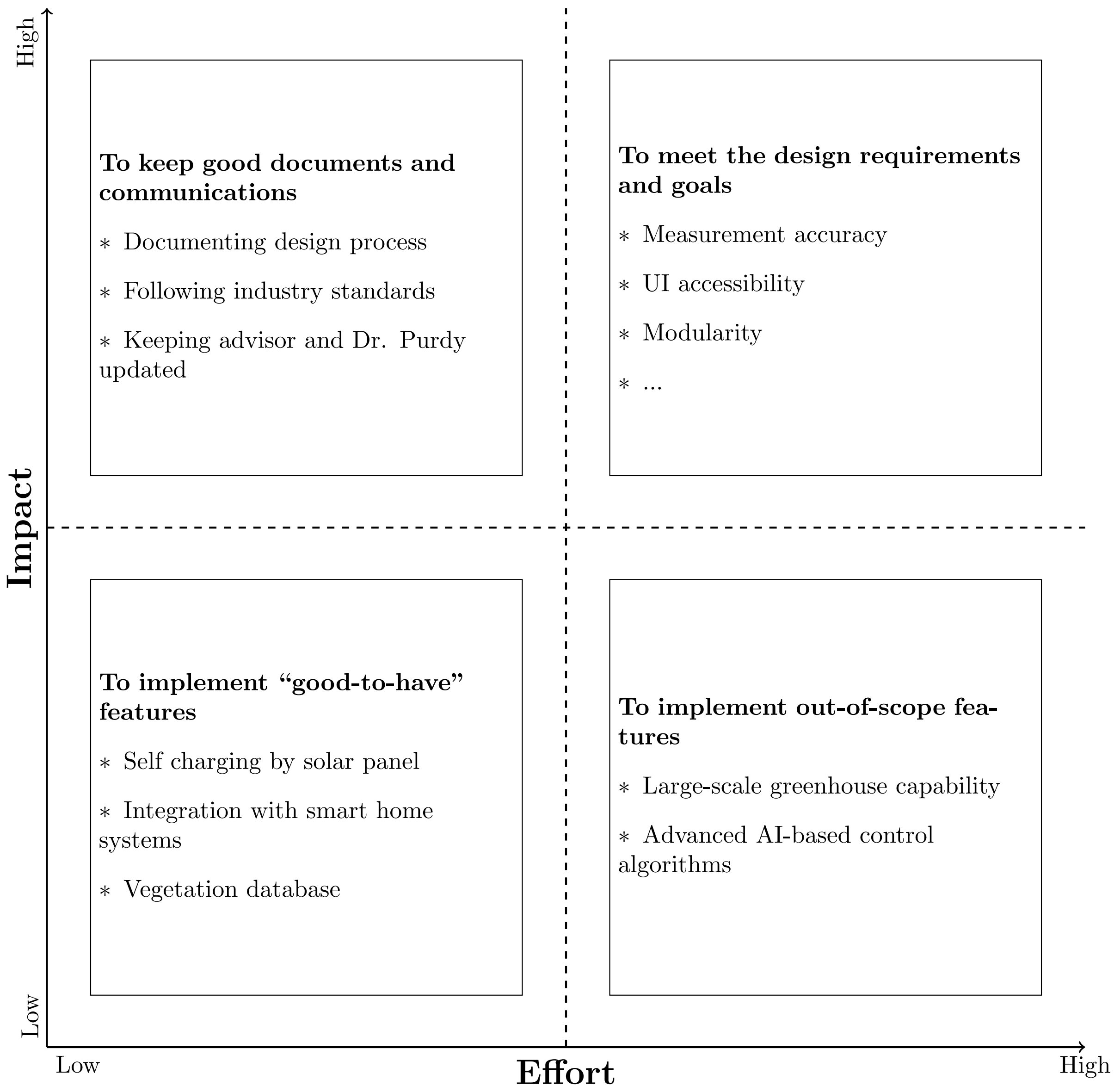

Figure 4: Impact-effort matrix on project-related tasks.

3 Project Requirements

3.1 Project Goals

The end-goal of this project is to develop a marketable product that functionally addresses the issues previously discussed in the problem statement: Poor landscaping practices and Water conservation.

As a consumer-based product, we define a qualitative functionality as the end goal - supporting users to care for their gardens and reasonably complete automated garden care when enabled. Aside from hardware sensor specifications, quantitative requirements are not realistic for this project, especially considering the time and resource limitations associated with the senior capstone framework. For example, assessing the effectiveness of the MGMS in improving garden yield would require long-term testing in a dedicated space, which would not be possible to complete following a semester timeline.

Initial project requirements will be wholly qualitative outside of hardware requirements and will follow previously identified objectives. For the intended system demonstration in April 2020, system effectiveness will not be tested in lieu of testing for intended system functionality and correct hardware performance. With correct hardware performance, system functionality can be more easily tweaked in software to improve overall system performance once that testing occurs. Because of this, the described level of testing is acceptable for a proof-of-concept demonstration in April.

Qualitative system attributes such as "ease of use" will be evaluated during the testing process using virtual surveys. This testing along with functionality testing will be defined during the development process as the system is more specifically defined.

The intended system functionality is as follows:

a. Promotes green spaces by lowering the learning curve of home lawn or garden care.
   - Real-time vital statistics
   - User configurable setup
   - Modular to mold to a variety of use-cases

b. Solves the common problem of garden over-watering to conserves water
   - Control system to keep garden soil moisture at healthy levels
   - Predicts weather patterns and only automatically waters when needed

3.2 Design Strategy

The definition of the product inherently makes the design an embedded system that requires multi-disciplinary knowledge and skills. Thus, we use the **strategy of design decomposition** to reduce the complexity of the problem to match each team member's expertise. Each hardware device (including sensors, controllers and actuators) will be set-up and tested individually during design prototyping, then combined and tested afterwards. The UI software does not depend on hardware as much, so the front-end development is performed separately, while having tasks and deadlines in the same pace as the hardware development, such that the whole system can be defined and prototyped synchronously.

Using a strategy of design decomposition provides several benefits to the project development efforts, the biggest of which is acting as a "cushion" for possible issues that may arise during development and prototyping. Design decomposition means that each system component is evaluated separately, removing dependency on any one component for the final system function. This way, if an issue arises during development, a component or design can be adjusted without affecting the major development of the project as a whole. This is especially important because of the many different sensors and communication technologies being considered for the project. It is likely that sensor accuracy or communication performance may arise as an issue for individual components. Thanks to a design decomposition strategy, these issues will be able to be solved without much consequence.

3.3 Standards

The development of the project conforms to various kinds of professional standards in the embedded system and IoT industry for the sake of security, readability and compatibility.

The product uses I2C bus and protocol for intra-board communication between devices, and uses Zigbee as inter-module wireless protocol. I2C (Inter-Integrated Circuit) is a synchronous serial communication bus invented by Philips Semiconductor (now NXP Semiconductors) [5] and widely used by current IC's in the market. Zigbee is a protocol developed by Zigbee Alliance based on IEEE-802.15.4 standard. IEEE-802.15.4 defines a two-layer architecture for low-data-rate wireless personal area networks (WPAN) [6], while Zigbee enhances it with two software layers [7]. Together they form a mature model to implement IoT concepts.

Additionally, electrical diagrams such as circuit schematic and PCB (printed circuit board) layout will be documented digitally in CAD (computer-aided design) software with standard rules and symbols built in. Common circuit diagram and PCB standards are specified in [8] and [9].

Finally, standards used in the software development, such as syntax and architecture, are based on specific dependencies, and they must be obeyed in order for the source code to successfully build or run. These standards are flexible in the development phase and will be documented in the final project delivery.

4 Functional Description

The MGMS system will utilize a modular design consisting of a central hub which will wirelessly connect to multiple sensing and watering modules that can be placed around a garden or house. The hub will host the central user interface and allow for customizing different garden setups. The hub software will make decisions based on the user configuration to control when to utilize the connected field modules in order to continuously monitor and water the garden. The user interface will be able to alert the user to garden events and make suggestions based on information available on the internet.

Tools such as a pairwise comparison chart, morphological chart, and decision tables were used to make design decisions in the definition and development phases of this project.

4.1 Pairwise Comparison Chart

Pairwise comparison chart (Figure 5) helps to evaluate the importance of a goal among others. Goals are listed in rows and columns, where if a row is more important than a column, the corresponding cell is marked a '1', otherwise '0'. Eventually the scores are summed up into the final column, representing the overall importance in an ascending order. This information helps make design decisions by highlighting the importance of features to be considered

Pairwise Comparison Chart						
Goals	Accurate	Easy to Use	Inexpensive	Weatherproof	Modular	Score
Accurate	-	0	1	0	1	2
Easy to Use	0	-	1	0	0	1
Inexpensive	0	0	-	0	0	0
Weatherproof	1	1	1	-	1	4
Modular	1	1	1	0	-	3

Figure 5: Pairwise Comparison Chart used to help determine priorities between specified goals of the MGMS

4.2 Morphological Chart

Attributes identified as most important in the pairwise chart above are used to make decisions on components and features during the development process. Some of the chosen features and technologies are displayed in a morph chart in figure 6) among the other possible considerations.

A morphological chart (or morph chart) lists possible means to implement defined functionality or requirements, among which the most suitable choice is selected by discussion.

4.3 System Diagram

A high-level system architecture (Figure 7) is constructed as per the objectives, constraints goals, and desired features identified above.

Morph Chart					
Func. | Means	1	2	3	4	5
Front-End Language	C	C++	Java	**Python**	-
Embedded Language	C	**C++**	Assembly	-	-
Wireless Protocol	Bluetooth	**Zigbee**	Wi-Fi	Cellular	-
Air Temp/Humid Sensor	SHT3X	**DHT11**	DHT12	-	-
Light Sensor	TSL2591	BH1750	TCS3472	**VEML7700**	**ALS-PT19**
User Interface	Touch Screen	**Local Website**	Windowed Application	-	-
Microcontroller Core	**AVR**	ARM	MSP	-	-

Figure 6: Morph Chart containing functional choices for elements of the MGMS. Bolded element is the selected choice for each function (showed by row).

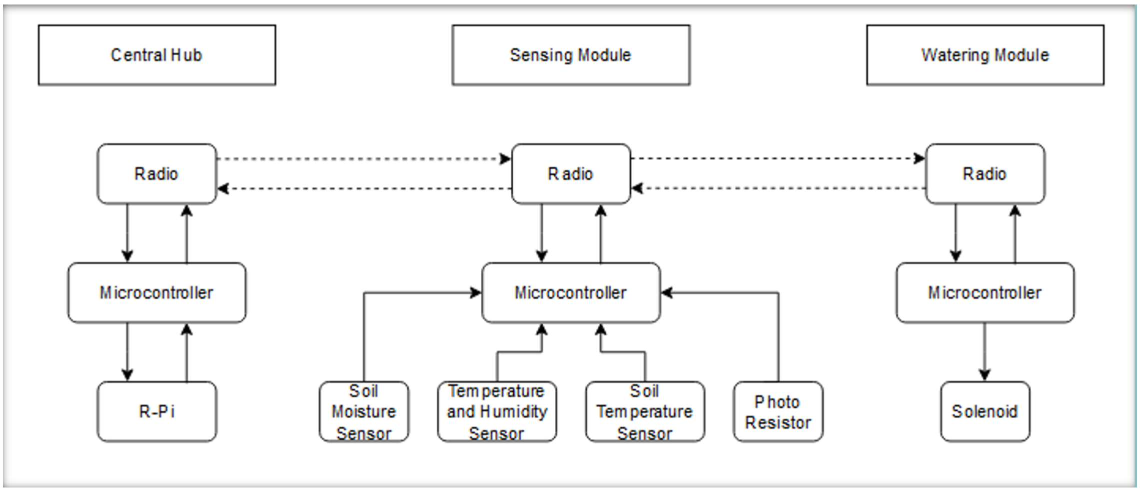

Figure 7: System Diagram for the proposed solution for the MGMS. This diagram shows the basics for the modularity of the system and the communication between modules.

4.4 MCU Decision Table

Finally, a detailed research on the microcontroller unit (MCU) is conducted since it very much affects the development efficiency and the performance of the final solution. All the considered choices are listed in Figure 8. It was eventually decided to use *ATMega328p* for an extensive community support, plentiful availability as well as a good balance between budget and performance.

MCU#	Dev Board	Architecture	Power Draw	UART	I2C	ADC	Program Memory	Data Memory	Cost
ATtiny817	XMINI	AVR	Max 9.6 mA @ 5V	1	1	1*12ch*10b	8 KB	128 B	9.06
ATtiny817	XPRO	AVR	Max 9.6 mA @ 5V	1	1	1*12ch*10b	8 KB	128 B	36.72
ATtiny416	XNANO	AVR	Max 9.0 mA @ 5V	1	1	1*12ch*10b	4 KB	128 B	9.06
ATtiny1607	CURIOSITY NANO	AVR	Max 7.8 mA @ 5V	1	1	1*12ch*10b	16 KB	256 B	15.29
ATtiny3217	CURIOSITY NANO	AVR	Max 10 mA @ 5V	1	1	2*12ch*10b	32 KB	256 B	15.29
ATtiny3217	XPRO	AVR	Max 10 mA @ 5V	1	1	2*12ch*10b	32 KB	256 B	38.76
ATMega328p	Arduino Uno	AVR	12.2 mA @ 3.3V	1	1	1*8ch*10b	32 kb	1KB	12.98
MSP430FR2355	LAUNCHPAD	MSP	PM module ctrl	1	1	1*12ch*12b	32 KB	512 B	15.59
MSP430F5529	LAUNCHPAD	MSP	PM module ctrl	1	1	1*16ch*12b	128 KB	128 B	15.59
MSP430G2553	LAUNCHPAD	MSP	Max 4.5 mA @ 3V	1	1	1*8ch*10b	16 KB	256 B	11.99
MSP430FR2433	LAUNCHPAD	MSP	PM module ctrl	2	1	1*8ch*10b	15 KB	512 B	11.99
MSP430FR2476	LAUNCHPAD	MSP	PM module ctrl	2	2	1*12ch*12b	64 KB	512 B	17.99
MSP430FR5969	LAUNCHPAD	MSP	Max 2.5 mA @ 3V	2	1	1*16ch*12b	63 KB	128 B	19.19
ATSAMD21G18	RedBoard	ARM	Max 6 mA @ 3V	6 collectively		1*20ch*12b	256 KB	32 KB	24.95

Figure 8: Research into different Microcontrollers that could be used.

5 Budget and Cost

5.1 BOM

The budget of the project will change over time depending on decisions in design and development. An example is shown in a Bill of Material (BoM) for a prototype machine in Figure 9.

Prototyping BoM Rev. 10/13/2020				
Description	**Mfr. Part Number**	**Unit Price**	**Quantity**	**Notes**
Arduino ATMega328p Prototyping Board	A000066	$22	3	ATSAMD21G18 or MSP430 are also options
Temperature/Humidity Sensor	SHT35	$14.18	1	
Vegetronix Soil Moisture Sensor	VH400	$39.95	1	
Vegetronix Soil Temperature Sensor	THERM200	$33.95	1	
I2C Light Sensor	VEML7700	$10	1	Assess which light sensor will be used in final design
Analog Light Sensor	ALS-PT19	$7.95	1	
CO2 Sensor	CCS811	$20	1	For indoor use but will be assessed in prototyping
XBEE S1 PRO 802.15.4 Radio	XBP24-AWI-001J	~$20	2	Obsolete but Already Owned
Raspberry Pi 3 Model B+	N/A	$35	1	
	Total:	**$267**		

Figure 9: Working Bill of Materials in order to determine cost for parts for the prototyping and design phase.

6 Team Information

There are four members in the design team including two Computer Engineering and two Electrical Engineering students in Class of 2021. The advisor is Dr. Zachariah Fuchs in EECS Department.

Alan Trester is an Electrical Engineering student with co-op experience in software, hardware, and manufacturing engineering roles through GE Aviation Systems. He has a strong passion for technology, design, and "making". After graduating he will be pursuing full-time positions in embedded systems or other design engineering roles in the consumer-products industry.

Eric Krenz is a Computer Engineering student whose past co-op experience was in hardware, software development, and cyber-security. He has a passion for engineering, technology, and making the world a better place. Post graduation he will pursue a full-time career in Information Technology Consulting.

Sadie Gladden is a Computer Engineering student with co-op experience in software development, user interface creation, game engine development, computer graphics, and cloud solutions through Siemens PLM and Siemens Healthcare GmbH. She enjoys exploring the relationship between hardware and software and exploring the connection and overlap of technology and medicine.

Zuguang Liu is an Electrical Engineering student who has past Co-op experience in industrial system design, embedded system hardware design, and simple machine learning implementation. After finishing a Bachelor's Degree with a Embedded System minor, he continues to pursue a Master's Degree in Electrical Engineering.

Dr. Zachariah Fuchs is the professor for Introduction to Mechatronics. He has extensive knowledge on embedded system design, sensor fusion, robotics and control systems. We believe he could advise us on the overall system architecture as well as specific components in the system.

7 Project Timeline

7.1 Gantt Chart

The team plans to meet weekly using the Microsoft Teams video conferencing application. This weekly meeting will occur every Tuesday for approximately 30 minutes starting at 3:30pm, and the work itself will be documented and shared using Teams and GitHub. The scheduled timeline is illustrated in a Gantt chart shown below. Tasks regarding the Implementation, Testing and Delivery phases are not reduced in detail as they depend on the result of the Design phase.

Figure 10: Gantt chart that describes the projected timeline. Overall timeline is broken down into *Definition*, *Design*, *Implementation*, *Testing* and *Delivery* phases with tasks. The current stage is in design as of today (May 15, 2021), so the later phases are not described in detail. The expected delivery deadline is the date of senior design expo 2021.

7.2 Time Distribution

Throughout the course of the project, it is estimated that each individual person on the team will work approximately 7-10 hours per week. Some weeks will be lighter on work (waiting for parts to be shipped), while others will be more labor intensive (assembly and programming), but overall the estimate of 7-10 hours per week is a fair number. Below is a rough chart documenting the time spent working on the project, which supplements the information shown in the Gantt Chart in the previous section.

Time Spending	Alan	Eric	Liu	Sadie
Hours Worked per Week avg.	6-8 Hours	6-8 Hours	6-8 hours	6-8 Hours
Hours Worked Total	45 Hours	40 Hours	50 Hours	45 Hours
Hours Planned Total This Semester	120-150 Hours	120-150 Hours	120-150 Hours	120-150 Hours
Hours Planned Total	290-320 Hours	290-320 Hours	290-320 Hours	290-320 Hours
Hours Planned per Week	7-10 Hours	7-10 Hours	7-10 Hours	7-10 Hours
Hours Planned for Design Planning	14 Hours + 18-20 upcoming Hours	14 Hours + 30-40 upcoming Hours	14 Hours + 18-20 upcoming Hours	14 Hours + 30-40 upcoming Hours
Hours Planned for Dev and Prototyping	40-50 upcoming Hours	20 upcoming Hours	40-50 upcoming Hours	20 upcoming Hours
		Total Est.	305 Hours per Person	

Figure 11: Time Chart showing the allocation of past time spent and future time expected on the project.

Just like the Gantt Chart, this table will continue to be updated as the project progresses. However, this table is just an estimate, and actual time spent on the project may vary compared to what is included in the table and will be reflected in the final design history document.

References

[1] N. A. of Landscape Professionals, "New research confirms americans still value lawns and green spaces." Available at https://www.businesswire.com/news/home/20190401005679/en/New-Research-Confirms-Americans-Lawns-Green-Spaces.

[2] E. P. Agency, "Outdoor water use in the united states." Available at https://19january2017snapshot.epa.gov/www3/watersense/pubs/outdoor.html.

[3] Onset, "Field monitoring system onset data loggers." Available at https://www.onsetcomp.com/hobonet.

[4] Edyn, "Edyn: Welcome to the connected garden." Available at https://www.kickstarter.com/projects/edyn/edyn-welcome-to-the-connected-garden/description.

[5] "UM10204 i2c-bus specification and user manual," vol. 2014, p. 64.

[6] "IEEE standard for low-rate wireless networks," pp. 1–709. Conference Name: IEEE Std 802.15.4-2015 (Revision of IEEE Std 802.15.4-2011).

[7] Z. Alliance, "ZigBee specification."

[8] "IEEE standard american national standard canadian standard graphic symbols for electrical and electronics diagrams (including reference designation letters)," pp. i–244. Conference Name: IEEE Std 315-1975 (Reaffirmed 1993).

[9] "IEEE approved draft standard for electrical characterization of printed circuit board and related interconnects at frequencies up to 50ghz.," pp. 1–150. Conference Name: P370/D8, July 2020.