

Smart Garden Monitoring System

Design Review

Team 74

Yen-Lin Liu, Chun-Lin Chao

TA: Benjamin Eng

March 8, 2016

ECE 445

Content

1.	Introduction.....	1
1.1.	Statement of purpose	1
1.2.	Objectives	1
1.2.1.	Goals.....	1
1.2.2.	Functions	1
2.	Design.....	2
2.1.	Block Diagram	2
2.2.	Block Description	3
2.2.1.	Node	3
2.2.2.	Gateway.....	5
2.3.	Flow Chart	6
2.4.	User Interface overview	7
2.5.	Packaging.....	8
2.6.		9
2.7.	Schematic	9
2.7.1.	Voltage Converter.....	9
2.7.2.	Light Sensor	10
2.7.3.	MCU.....	12
2.7.4.	USB Charging Circuit.....	13
2.8.	Solar Tracking System.....	13
3.	Requirements and Verification.....	14
3.1.	Table of Requirements and Verification	14
3.2.	Tolerance Analysis.....	16
4.	Cost and Schedule:.....	16
4.1.	Cost	16
4.1.1.	Labor	16
4.1.2.	Parts	17
4.1.3.	Total Cost.....	18
4.2.	Schedule	18

4.3.	Work Assignment.....	18
5.	Ethics & Safety issues.....	19
6.	References	19

1. Introduction

1.1. Statement of purpose

People often grow unsuitable plants in their gardens so that they usually die from the lack of proper nutrients, enough water, or even too much sunlight.

We want to solve this problem, so we decided to build a system that can monitor the environment and give user recommendations on what plants to grow.

Currently there are some solutions like “Edyn”, which is also a kind of agricultural sensing product. However, it only provides **single-point** sensing and it requires **Wi-Fi** access on all nodes. Single-point sensing is not capable of representing the whole garden status. Moreover, it is hard for Wi-Fi signal to be reachable in the whole garden since Wi-Fi network needs support from the power grid. We plan to make a system that provides **multi-point** sensing and builds its own **ZigBee network** instead of totally relying on existing Wi-Fi network.

1.2. Objectives

1.2.1. Goals

- Build a sensor network to detect environment parameters in a medium size garden (25m x 25m at most).
- Give advices on what to plant in each section of the garden according to the sensors.
- Reduce plants death from improper environment.

1.2.2. Functions

- Obtain ambient **temperature**, ambient **humidity**, soil **moisture**, and **illuminance** from a set of sensors.
- For each section of the garden, the values retrieved from each sensor will be displayed on the UI.
- Give **recommendations** on planting the garden; such as how much water one should give to their plant, how long should one expose he’s plant to the sun, and

so on.

- Use a self-built **ZigBee** network instead of relying on existing Wi-Fi network.

2. Design

2.1. Block Diagram

There are two sets of hardware in our system. One of them is the **gateway** as **Figure 1** shows, the other is the sensor **node**, as **Figure 2** shows.

Gateway is a box containing only communication modules, and is used to retrieve all data from the self-built network and upload them to a backend cloud database on the Internet.

Nodes are sensor set boxes placed in the garden, and are responsible for uploading all sensor values to the self-built ZigBee network.

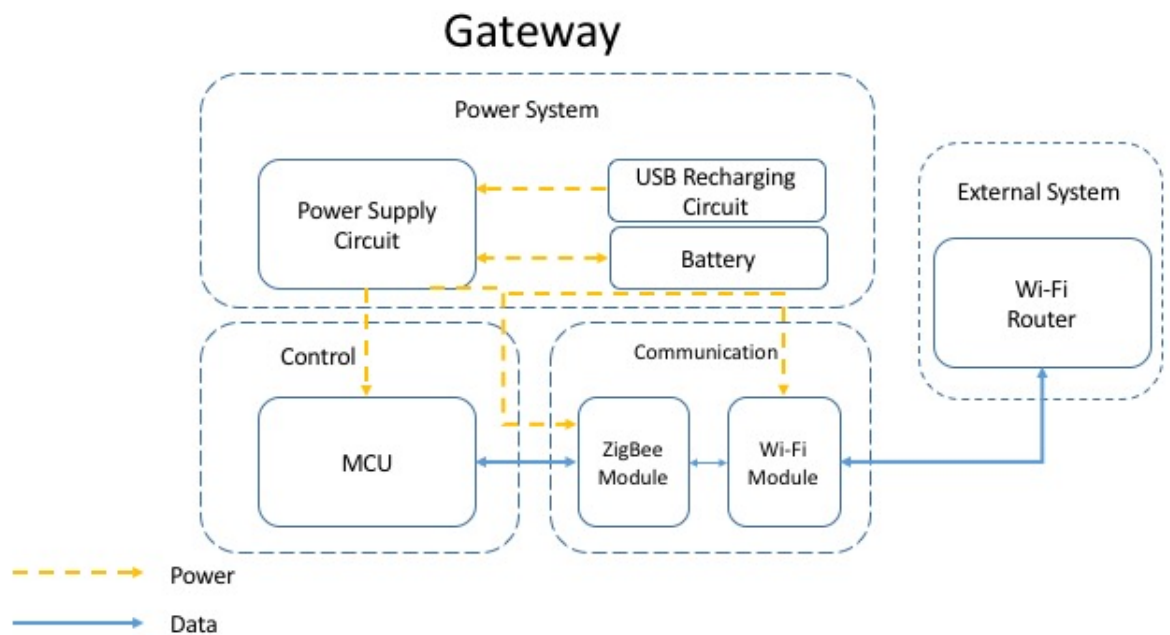


Figure 1

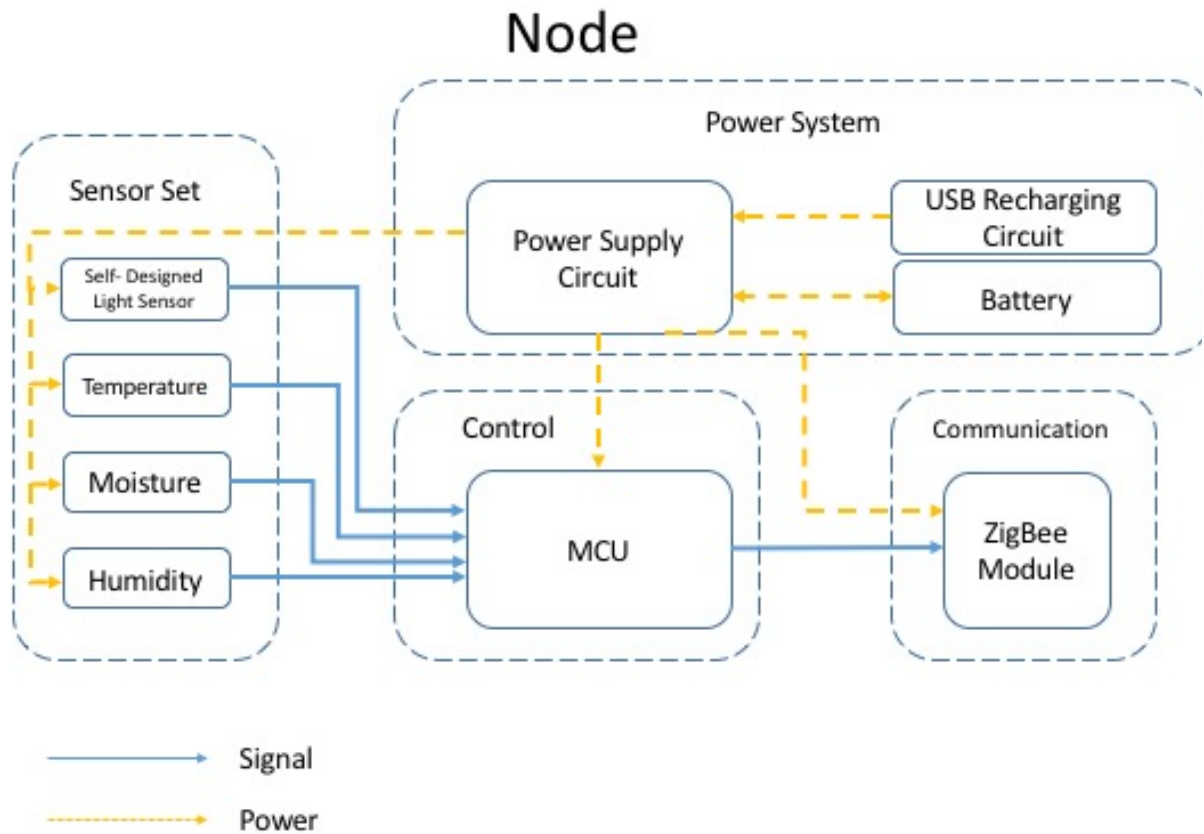


Figure 2

2.2. Block Description

2.2.1. Node

2.2.1.1. Sensor Set

For each set, we will put in ambient **light**, ambient **temperature**, soil **moisture**, and ambient **humidity** sensor. Each of them will output either analog or digital data to the MCU. Except for the moisture sensor that is put in the soil, the rest of them will be sensing data from air.

We will design an illuminance sensor by ourselves, because modern low-cost illuminance sensors we found are not accurate enough since it contains only a Light Dependent Resistor (LDR) and an amplifier, and it outputs analog signal according to the resistance of the LDR. However, it cannot measure it accurately since it uses only a voltage divider, so we decide to design a more complex one.

2.2.1.2. MCU

We will use **ATMega328P** as our microcontroller in our nodes. It will receive raw data from the sensors and pass them onto our self-built ZigBee network. We will have **Arduino bootloader** on the chip for implementation of software since it can provide an easier API than AVR programming. It will be placed on a PCB and wired up with all other sensors, power system and communication modules.

2.2.1.3. Power System

We will use rechargeable button cells (**LIR2450**) as batteries and construct our own circuit to supply all the requiring voltage of the circuit. We will also construct a **USB recharging circuit** to charge the batteries

As for the MCU, it requires 1.8V-5.5V to operate; the sensors mostly require 5V or 3.3V to operate. So we will need a step-up converter to provide the voltage we needed. We choose **MC34063A** as our step-up converter, which allows a wide range of input voltage and outputs 5V.

We choose MCP73831 as our recharging chip, which is a charge management chip for li-ion battery, to construct the circuit. It provides the capability of USB charging to our batteries.

For batteries(**LIR2450**), its battery capacity is 140mAh (for continuous usage). The whole system will consume roughly 20mAh.($140/20=7$ hours)We'll take samples once an hour and each process takes about 30 seconds.

As a result, we have to recharge batteries after $7*60*60/30=840$ hours(roughly 35 days).

2.2.1.4. ZigBee Module

We are mainly using **ZigBee** as our communication protocol and we choose **XBee** module to be the chip. We choose ZigBee for reasons as follows:

First, ZigBee is more suitable for constructing a light-weight network rather than Bluetooth or Wi-Fi. Bluetooth transmissions requires pairing, but only up to 7 devices are allowed to be active at the same time. Wi-Fi requires extra hardware like routers, but it is not always possible to have Wi-Fi routers placed around the whole garden.

ZigBee allows up to 255 devices active on the network, so it best out all three.

Second, the cost of constructing a network using ZigBee protocol is the lowest among all three methods. Bluetooth and Wi-Fi require much more power than ZigBee does. Wi-Fi also requires extra hardware like we mentioned above. ZigBee only requires an antenna on each chip to construct the network, so is then cheaper than the others.

2.2.2. Gateway

2.2.2.1. MCU

The MCU obtains sensor data from the XBee module and post them on the Internet through the Wi-Fi module. For faster implementation, we choose ***LinkIt Smart 7688 Duo*** as our MCU since it has a built-in and well integrated Wi-Fi module. It is also solderable and cost much lower than buying them separately.

Moreover, it also contains a MPU. We can perform an online update of the firmware on the MCU through the MPU. It can also provide a local monitoring UI incase the network failed and thus inaccessible from the mobile app.

2.2.2.2. Wi-Fi Router

An external Wi-Fi router will be needed here in order to gain access to the Internet. This can be an existing one or a new-installed one as long as it has access to the Internet and runs basic DHCP and NAT service.

2.3. Flow Chart

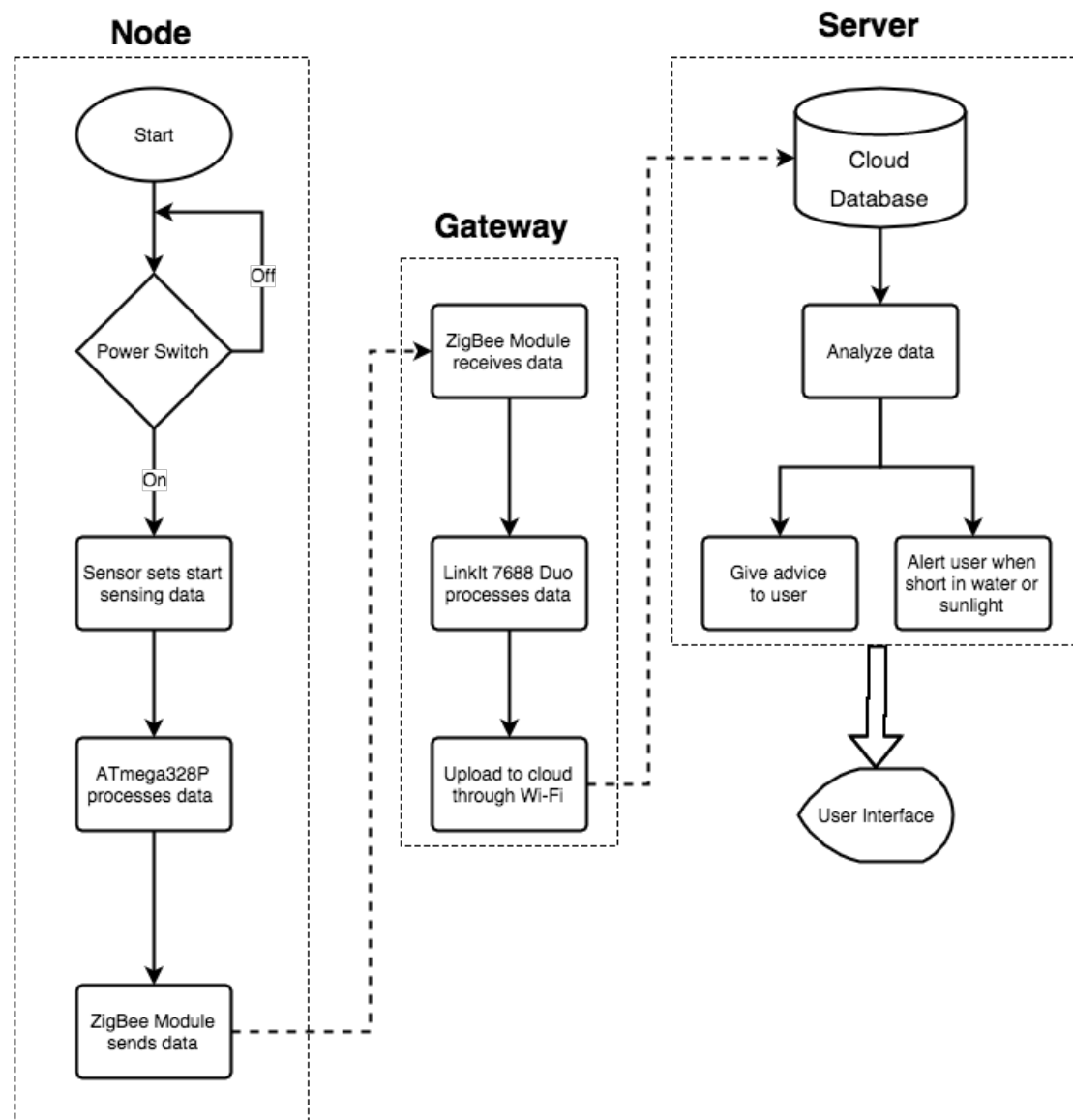


Figure 3

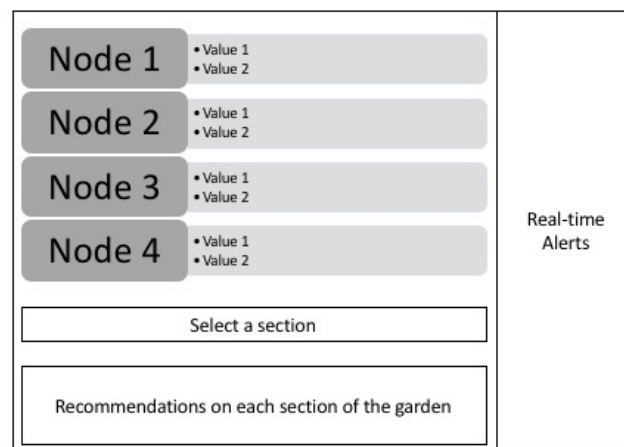
This is a more detailed flow to the whole system. As mentioned above, the sensor nodes collect data from surrounding environment, convert them into digital signal and transmit them to the gateway. The gateway retrieves data from the sensor nodes and upload them to the backend cloud database.

After uploading sensor data to the cloud, there will be a periodic task on the server which monitors the garden status; i.e., sunlight, moisture or humidity level, in real-time. If any of the values exceed or drop below some threshold, users will be alerted through a mobile app.

Moreover, there will also be another periodic task on the server that analyzes

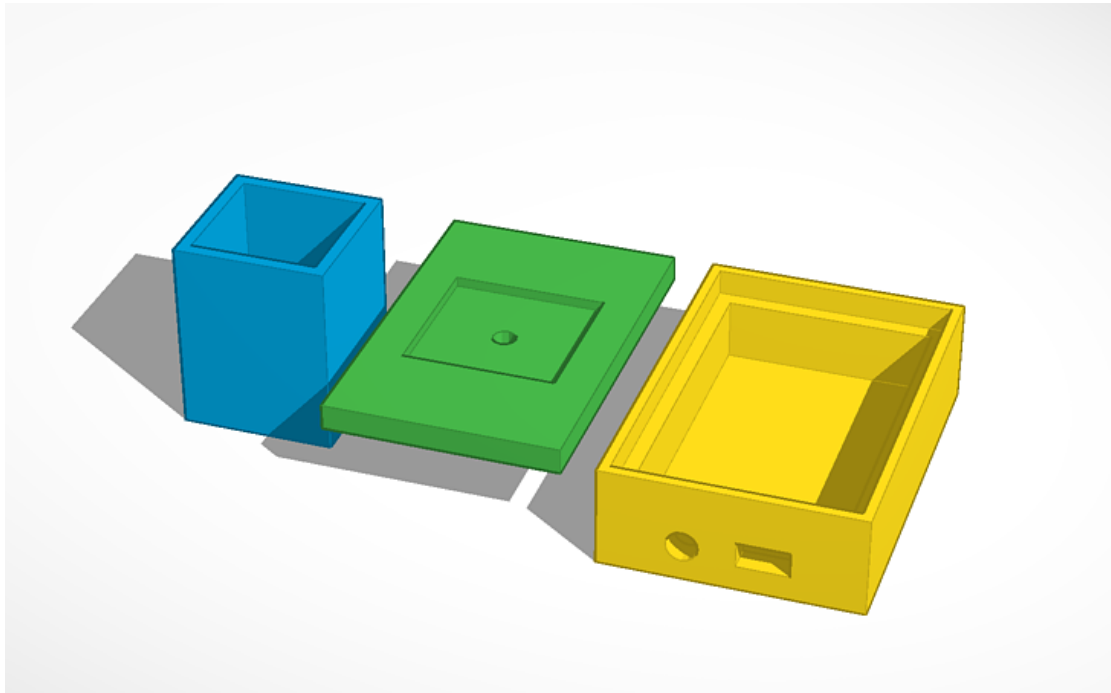
the data, and give users recommendations to plant their gardens. As for the analyze, we will calculate the average illuminance, humidity level temperature and moisture level and feed it into a machine learning algorithm. The algorithm based on a plant database which labels each plant with its growing conditions including light, temperature, moisture and humidity. Also, each kind of plant may require different conditions in different stages, our mobile app will also give advices according to what we have in our database.

2.4. User Interface overview

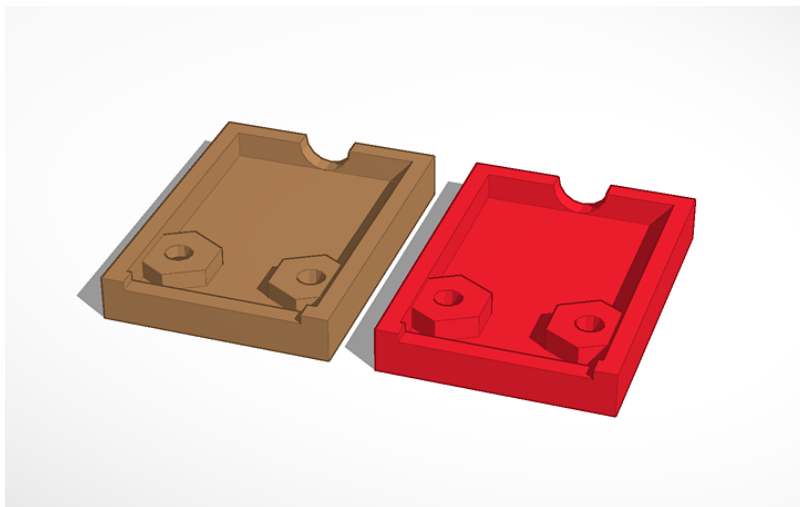


- The top left part shows real-time sensor data of all nodes.
- User clicks on a node to select a section, and the corresponding advice will show up in the section below.
- Real-time alerts will be shown on the right side

2.5. Packaging



Box for sensor node (the blue one is transparent)



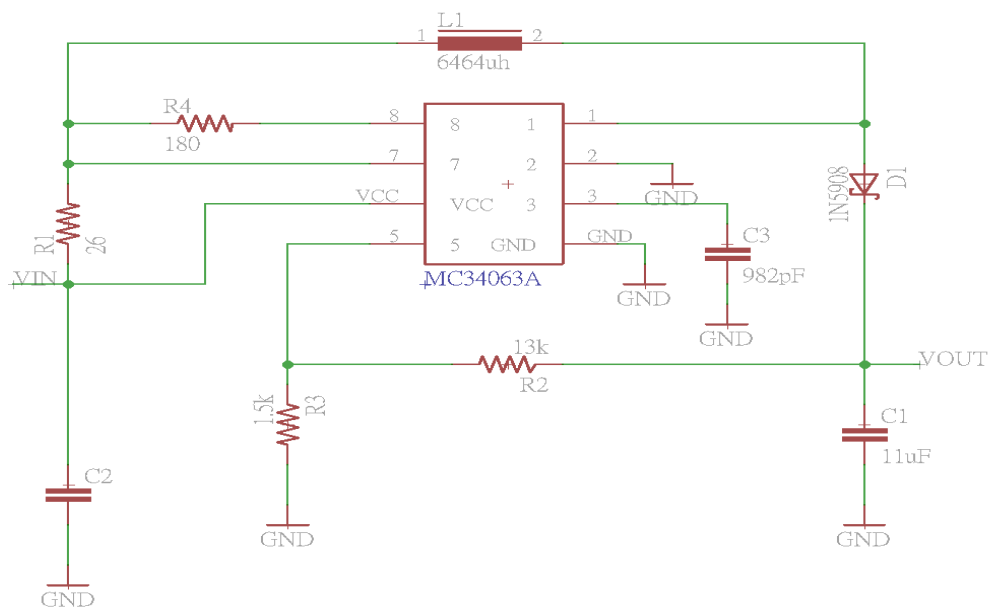
Case for moisture sensor

2.6.

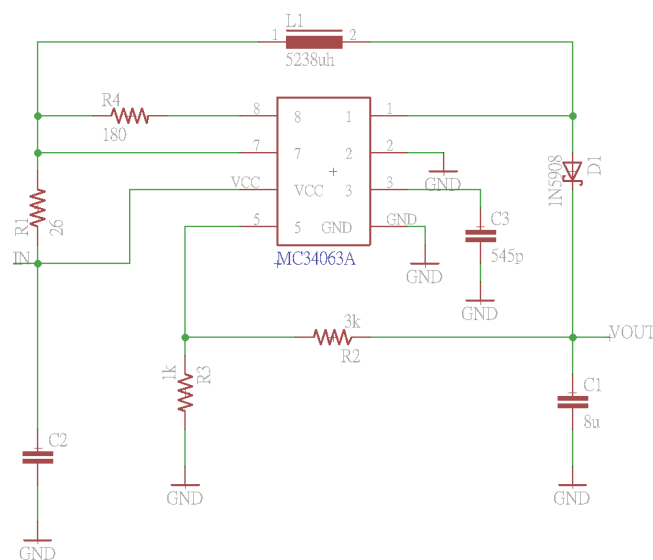
2.7. Schematic

2.7.1. Voltage Converter

(3.6V to 12V)



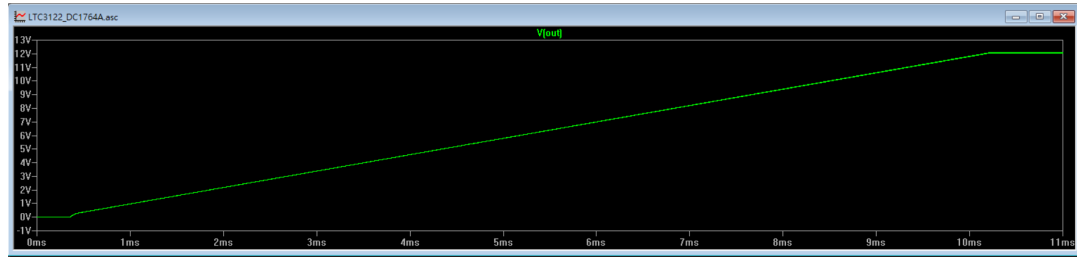
(3.6V to 5V)



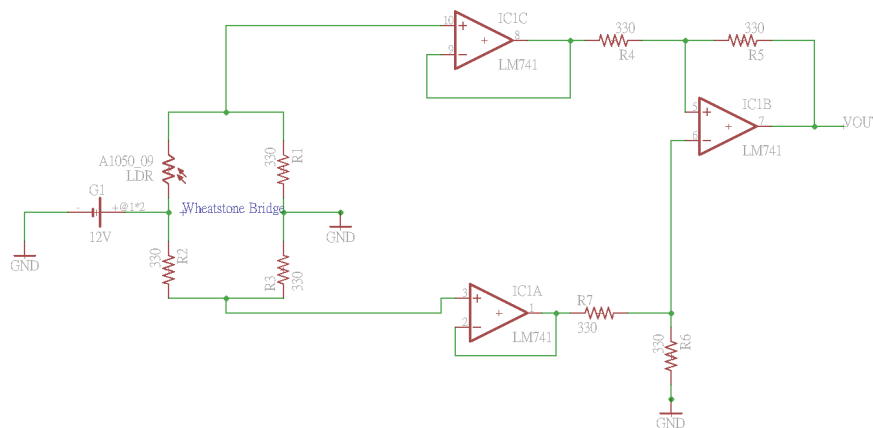
MC34063A is a high efficiency boost converter which accept the input voltage

range from 0.6 to 5.5V. We want to use **LIR2450** battery which provide 3V input voltage. To fit the operating voltage of microcontroller and ZigBee module, we use this converter to make 12V/5V output voltage and 1.5mA output current.

2.6.1.1 Simulation



2.7.2. Light Sensor



We use Wheatstone Bridge to measure the resistance of a light dependent resistor (LDR). Then, we use two voltage followers as a first stage amplifier to isolate signal source from output. Finally, we use a differential amplifier to amplify the output voltage for better resolution.

Analysis:

For a Wheatstone bridge: (V_{dc} = input voltage)

$$V_a = V_{dc} * \frac{Ra}{Ra + Rt}$$

$$V_b = V_{dc} * \frac{Rb}{Rb + Rc}$$

In the balanced condition:

$$V_a = V_b$$

$$V_{dc} * \frac{Ra}{Ra + Rt} = V_{dc} * \frac{Rb}{Rb + Rc}$$

$$\frac{Ra}{Ra+Rt} = \frac{Rb}{Rb+Rc}$$

$$Ra \cdot R_b + Ra \cdot R_c = Ra \cdot R_b + R_b \cdot R_t$$

$$R_c/R_b = R_t/R_a$$

Now, when the physical condition changes R_t increases or decreases by ΔR and the bridge is no longer in the balanced condition

$$V_a \neq V_b$$

$$V_a = V_{dc} * \frac{Ra}{Ra+(Rt+\Delta R)}$$

$$V_b = V_{dc} * \frac{Rb}{Rb+Rc}$$

We assume all the resistors to be equal.

$$\text{i.e. } Ra = R_b = R_c = R_t = R$$

$$V_a = V_{dc} * \frac{Ra}{Ra+(Rt+\Delta R)}$$

$$V_a = V_{dc} * \frac{Ra}{R+(R+\Delta R)}$$

$$V_a = V_{dc} * \frac{Ra}{2R+\Delta R}$$

$$V_b = V_{dc} * \frac{R}{R+R}$$

$$V_b = V_{dc}/2$$

Now,

$$V_{ab} = V_a - V_b$$

$$= V_{dc} * \frac{Ra}{2R+\Delta R} - \frac{V_{dc}}{2}$$

$$= \frac{V_{dc}}{2} * \frac{2R-2R-\Delta R}{2R+\Delta R}$$

$$= \frac{V_{dc}}{2} * \frac{-\Delta R}{2R+\Delta R}$$

Output Voltage for a differential amplifier is given by

$$V_o = \frac{-R_2}{R_1} * V_{ab}$$

$$= \frac{-R2}{R1} * \frac{-\Delta R}{2R + \Delta R} * \frac{V_{dc}}{2}$$

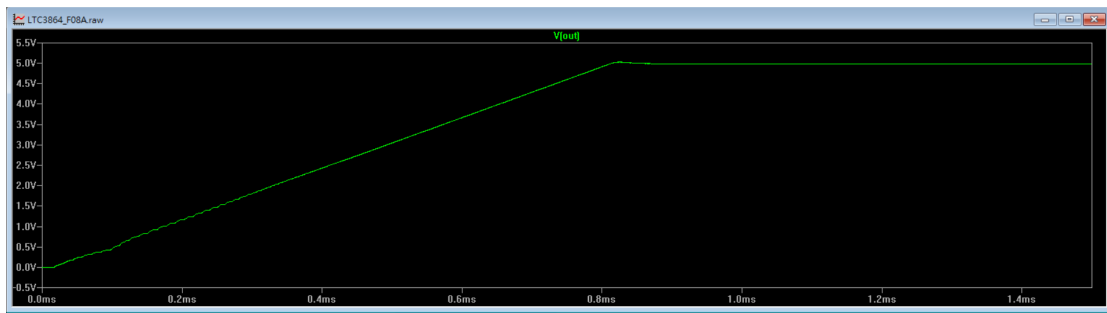
Hence,

$$V_o = \frac{R2}{R1} * \frac{\Delta R}{2R + \Delta R} * \frac{V_{dc}}{2}$$

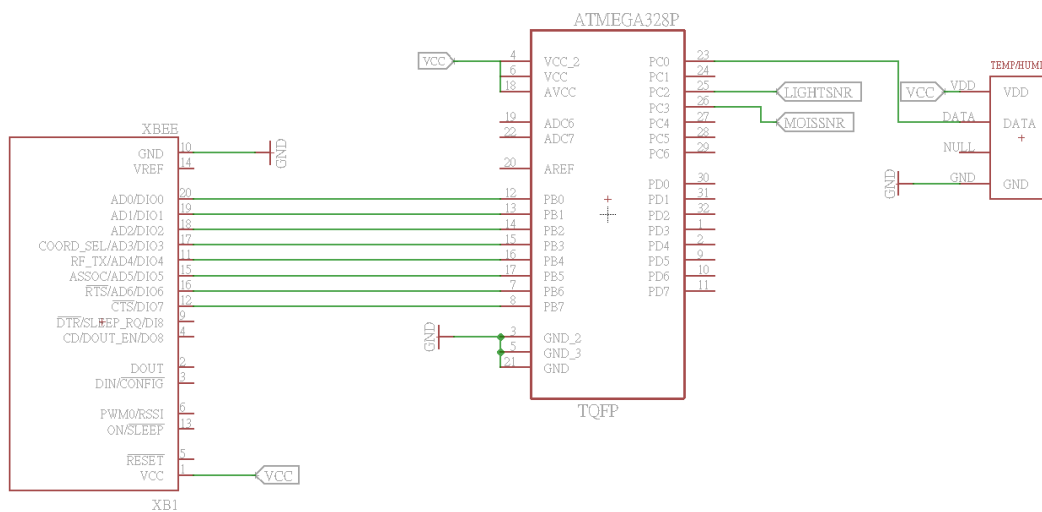
Where resistance of the LDR is $(R + \Delta R)$

2.6.2.1 Simulation

If we use $10k\Omega$ of LDR , then the output voltage is about 5.3V:



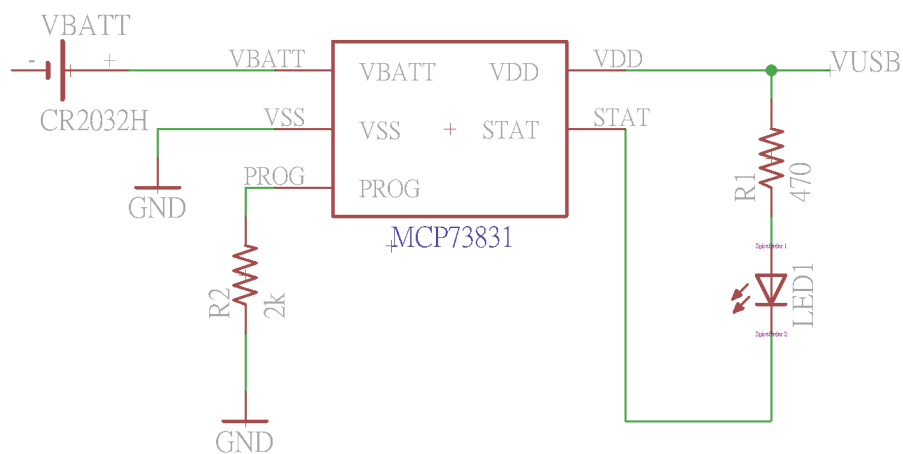
2.7.3. MCU



ATmega328P is a low-power CMOS 8-bit microcontroller combined with a high data rate transceiver for the 2.4 GHz ISM band. We will retrieve sensor values and transmit data to the ZigBee network using this chip with Arduino bootloader programmed inside. It accepts 1.8V-5.5V as operating voltage, and we are using 3.6V as its operating voltage.

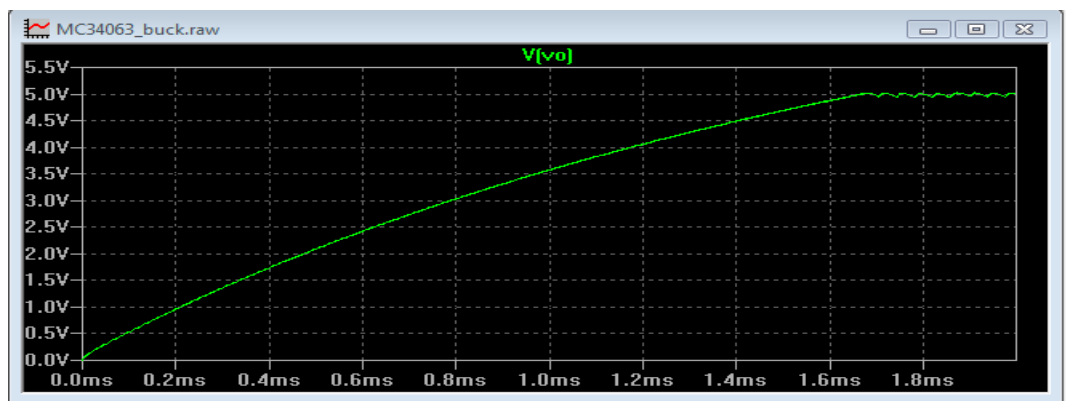
It will receive digital signal from temperature and humidity sensor, analog signal from moisture sensor and light sensor through GPIO pins. ZigBee module communication will be done by digital signal through SPI interface.

2.7.4. USB Charging Circuit



This circuit allow users to charge the battery by using the USB port on the computer or directly from power bank. The basic design is as same as the application circuit in MCP73831 datasheet.

2.6.4.1 Simulation



2.8. Solar Tracking System

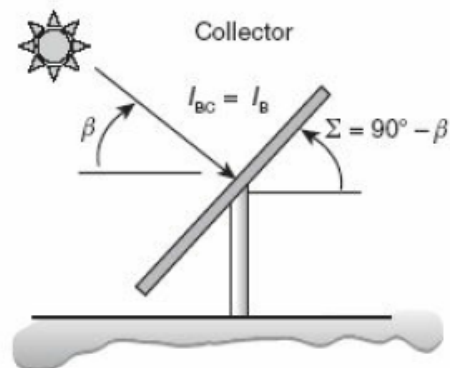
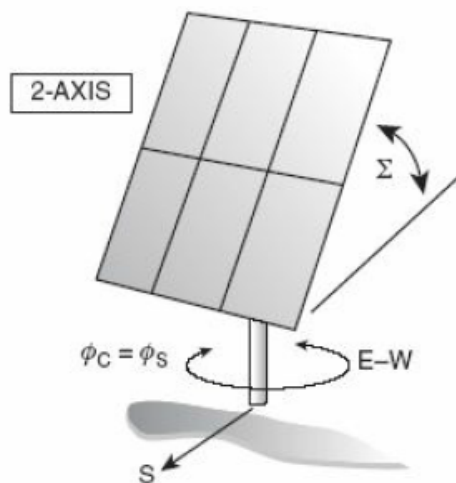
2.7.1 Two-Axis Servo

We'll combine two servos to make a two-axis servo motor tracking the sun to obtain maximum luminous. We use TowerPro SG90 as our servo and It operates under 5V.



2.7.2 Two-Axis Tracking

To collect the maximum luminous, we rotate the collector along the z-axis to make the angle match solar azimuth angle Φ_s (15° per hour rotation) and tilt the collector to make the angle normal to solar altitude angle. ($\Sigma = 90^\circ - \beta$)



3. Requirements and Verification

3.1. Table of Requirements and Verification

Requirement	Verification
Temperature sensor	Temperature sensor
1. The reading of the sensor should be at	(a) Hold a thermometer next to the sensor

most $\pm 1^{\circ}\text{C}$ error within surrounding temperature.	(b) Ensure the error between thermometer and the sensor remains within $\pm 1^{\circ}\text{C}$ between $0 \sim 30^{\circ}\text{C}$
Humidity sensor 1.The reading of the sensor should at at most $\pm 5\%$ error within surrounding humidity.	Humidity sensor (a) Hold a humidity meter next to the sensor (b) Ensure the error between the meter and the sensor remains within $\pm 5\%$ both indoors and outdoors
Moisture sensor 1. The reading of the sensor should be at most $\pm 5\%$ error.	Moisture sensor (a) Put a moisture meter into the soil (b) Ensure the error between the meter and the sensor remains within $\pm 5\%$ when dry and just watered
Light sensor 1. The reading of the sensor should be at most ± 10 lux error within surrounding illuminance.	Light sensor (a) Hold an illuminance meter next to the sensor (b) Ensure the error between the meter and the sensor remains within ± 10 lux
MCU 1. Operate between 32°F to 120°F for at least a week.	MCU (a) Place the nodes in action for a week outdoors and indoors, check the database to see if there's any weird value or missing timestamps (b) Attach multimeter to the input and output pin (c) Ensure input and output voltage remains within 0 V and 5 V
Power Supply Li-ion Battery: Supply $+3.6\text{V} \pm 5\%$ power draw on the PCB. DC Converter: 1. Able to drive different sensors, XBee and MCU 2.Provide at least 12V, 5V, 3.3V output with $\pm 10\%$ error Solar Tracking:	Power Supply Li-ion Battery: (a) A multimeter will be used to check if the voltage outputs is equal to specified values. Use an oscilloscope to check if the voltage signal is steady. (b) Let the supply work for a week, and check the database if there's a weird or missing value (c)Ensure the error remains within $\pm 10\%$ DC Converter: (a) Use multimeter to measure the voltage of

1. The operating voltage of the servo is 5V. 2. Use DC converter to be able to provide 3.6V to the Battery	different output nodes, and they should show corresponding value Solar Tracking: (a) A multimeter will be used to check if the voltage outputs is equal to specified values.
ZigBee 1. ZigBee modules should transmit at least 2 sets of data through the network per minute, within 50 meters without any data lost.	ZigBee (a) Transmit data through different distance, from 0 – 25 meters (b) Place objects that may appear in garden between sensors such as flowers and verify the result

3.2. Tolerance Analysis

For the solar energy collector, to find solar altitude angle β :

$$\sin \beta = \cos L \cos \delta \cos H + \sin L \sin \delta$$

Where L is latitude, δ is solar declination($\pm 23.45^\circ$),

H is hour angle(e.g. 12PM= 0° , 11AM= $+15^\circ$, 10AM = $+30^\circ$, 1PM = -15°)

To find solar azimuth angle:

$$\sin \varphi_s = \frac{\cos \delta \sin H}{\beta}$$

4. Cost and Schedule:

4.1. Cost

4.1.1. Labor

Name	Hourly Rate	Total hour	Total = Rate * 2.5 * Total hour
Yen-Lin Liu	\$35	160	\$5600

Chun-Lin Chao	\$35	160	\$5600
Subtotal			\$11200

4.1.2. Parts

Name	Store	Quantity	Cost
Temp & Humidity Sensor DHT22	Amazon	4	\$16.80
Moisture Sensor SEN92355P	Digi-Key	4	\$19.60
ZigBee XB24CZ7PIS-004	Digi-Key	5	\$87.50
MCU ATmega328P	Digi-Key	4	\$14.80
Battery LIR2450	Amazon	6	\$12.44
Voltage Regulator MC34063A	Digi-Key	5	\$2.85
USB charging IC MCP73831T	Digi-Key	5	\$2.80
Diode 1N5908	Amazon	5	\$1.00
OP Amplifier LM741	Amazon	5	\$2.90
Resistor(26,180,330,470,1K,1.5K ,2K,3K,13Kohm)	ECE Store	20	\$3.00
Inductor (6464uH,5238uH)	ECE Store	20	\$10.00
Capacitor(8uF,11uF,545pF,982pF)	ECE Store	20	\$4.00
Total			\$177.69

4.1.3. Total Cost

Total = Labor + Parts = 11200.00 + 177.69 = \$11377.69

4.2. Schedule

Date	Yen-Lin Liu	Chun-Lin Chao
2/8	Prepare project proposal and research on components	
2/15	Design sensor circuit	Design power supply circuit
2/22	Finalize Design Review	Design MCU circuit
2/29	Order parts and design UI	Finalize Design Review
3/7	Design database schema & Test database API	Test ZigBee module communication & Test all sensors
3/14	Integrate API with MCU	Test power system & Design PCB
3/21	Work on algorithms	Assemble parts
3/28	Debug and test the device	
4/4	Prepare for mock demo	
4/11	Software debugging	Hardware debugging
4/18	Get feedback from users, seeking for final improvement	Prepare for the final paper/ Gathering and analyze data for final report
4/25	Finalize the presentation and demonstration	
5/2	Finalize the final paper	

4.3. Work Assignment

- Yen-Lin Liu:
 - Order parts
 - User interface, database design and implementation
 - Machine learning algorithm implementation
 - MCU integration with backend server

- MCU coding of control system
- Chun-Lin Chao:
 - PCB design
 - Circuit design
 - Sensor testing
 - Circuit testing
 - ZigBee module communication testing

5. Ethics & Safety issues

It is very important to follow the rules of the IEEE code of Ethics while we are designing anything. As an engineer, we can't just develop new products and then ignore the safety, health and welfare of the public. There are also various aspects that we must keep in mind all the time. The first one of those aspects is that we have to disclose promptly factors that might endanger the public or the environment. Meanwhile, we also must avoid injuring others, their property, reputation, or employment by false or malicious action. In other words, it's our responsibility as engineers to confirm our product is safe enough for people to use. A couple of safety issue of our product would be the circuit short and eventually leads to burn while user is using. Therefore, while designing, we have to make sure that the circuit would not be short.

6. References

- [1]*MC34063A datasheet*, 11th ed. Semiconductor Components Industries, 2003.
- [2]*MCP73831T datasheet*, 1st ed. Microchip Technology Inc., 2008.
- [3]*Lir2450 datasheet*, 1st ed. PowerStream, 2008.
- [4]*ATmega328P datasheet*, 1st ed. Atmel, 2015.
- [5]"Edyn", *Edyn*. [Online]. Available: <https://edyn.com/>. [Accessed: 08-Mar- 2016].