

# GARDENS Information Page

**Greenhouse Automation and Relay Device for Expansionary Nodal Systems (G.A.R.D.E.N.S.)** is being developed by [Andrew Baker](#) and [Willy Duffle](#). \*\*EDIT: Willy Duffle helped me start this project, and because of personal reasons, he was unable to continue in the class and lab, therefore I (Andrew) did most of the work seen here, so if you have any questions and want to email us, please email me, as Willy will probably tell you to do the same :) \*\* The goal of this project is to develop a system which can monitor greenhouse conditions and act accordingly. This, used in conjunction with typical greenhouse upkeep by the user, will help improve plant performance and encourage users to produce their own consumable food products. This system is also designed to aid the handicap in caring for their plants. There are several "key" parts of the system:

## G.R.O.W. (Greenhouse Robustness and Operations Workstation)

- Main interface device for the user to program the "zones" (more below)
- Graphic LCD screen also displays current conditions along with any troubleshooting information
- Speaker lets the users know if a function was successful or a failure
- Front panel of the system is smooth, to allow easy cleaning of the keypad and screen
- Provides power to the rest of the "zones" via ~~regular Cat5e cabling with standard male RJ-45 connectors~~ Cat 5e was the original plan, but it was quickly discovered that more current was needed in the interconnect wire than initially planned, and so heavier gauge wire was needed, mainly for concern of safety.
- Can be removed and taken into the house to be plugged into the computer for easier programming

## S.E.C.A.N.T (Sensor Enabled Controller and Automation Node Transceiver)

- Utilizes different sensors to observe the conditions within the greenhouse and activate appropriate controllers
- Controlled via I2C communications protocol ~~via Cat5e cabling~~
- Acts as an independent unit along the chain of communication
- Can be set to any address (via software) within the valid I2C parameters (0 to 128)
- Has the ability to be taken out of the communication chain and troubleshooted ~~via custom USB to RJ-45 cable~~
- Receives power from interconnect wire
- Can only talk with the G.R.O.W. module

## R.A.C. (Relay Activation Controller)

- Houses all relays, keeping mains 120V power away from lower voltage system and limiting the possibility of dangerous electrical wiring faults
- Used to turn on and off water valves with the possibility of expansion to HVAC and Humidifier/Dehumidifier controller.

The code for the GROW and SECANT can be found on the [GARDENS Code](#) page

The development of this system was inspired by friends who grew their own vegetables in a garden, who are also techno savvy and wanted something they could integrate into their current garden that would aid them in their plant's production of consumable goods. The current system is not 100% complete but is 100% functional. With the limited budget and man hours only one zone was created with its water supply relay attached directly to the SECANT unit. The system was designed with the intentions of being expandable up to 127 zones (I2C supports 7-bit addressing). The current code that is written for the SECANT units can handle this expansion, but the code for the GROW unit will need to be slightly modified to be able to handle more than one zone.

Figure 5 shows the connection schematic for the system. The power connections were left out because it will clutter the schematic, and the boards are marked with their power pins. One important note to mention, is that the two microcontrollers MUST have at least ONE common ground for the I2C link to work properly. Also, the Vcc on the microcontrollers, unlike other units used for other projects (such as BS2SX) is NOT for unregulated voltage! It must be 3.3V! If you need to power the board from a higher potential, you must use the RAW voltage input, which is then fed into a 3.3V regulator on the Pro Mini. I used this method and ran 5V rail to the unit so that I could power the relay and servo from the 5V rail but run at 3.3V logic (which is compatible for transmitting from a 3.3V to a 5V system, but the servo and relay need 5 V to their power sources to operate).

The GROW unit is connected to the LCD screen via the TX serial connection on the Pro Mini, and the keypad IRQ line, which is used to indicate when a pad has changed state, is connected to Digital Input 2. The SDA and SCL lines are used for I2C communications. SDA stands for Serial Data, and SCL stands for Serial CLock. These are the lifeline of the system and connect everything together. Although the keypad is connected to the bus lines, only the GROW unit utilizes its functions. R1, R2, and R3 are acting as pull up resistors on the bus lines, since I2C is defined as an open drain protocol (and the IRQ line on the keypad operates in the same manner). The next thing we see, reading the schematic from left to right, is S1. This switch is used to put the SECANT mode into independent monitor mode. If this switch is closed and the chip is powered up, the chip will enter this mode and allow a USB connection to the FT232RL chip which converts the USB data into TTL data, for the chip to understand.

Under this mode a user can troubleshoot the unit, change variables, or view the analog data from the temperature and moisture sensors. If this pin is low (which by default it is) then the chip, on power up, will act normally. Finally we get to the SECANT, where all of the processing really takes place! The SECANT is connected to a TMP36 (datasheet can be found online) analog temperature sensor, a Vegetronix moisture sensor, a relay which controls a 120V 15A socket, and a small RC servo to open and close the roof hatch. The temperature sensor and moisture sensor are connected to the Analog ports 0 and 1. The Arduino pro mini has a built in analog to digital converter which can take this analog voltage input and convert it, with 10-bit resolution, to a digital signal for the microcontroller to process.

The part of this project that took the most amount of time was the code. This project could have been done on a passive level with voltage comparators to turn on and off values, but the trade off is that it would not be programmable and each unit would have to be fine tuned to its specific area and application. The GROW and SECANT are running somewhat similar codes (for the I2C bus communications) but after that point are totally different. Added together there is roughly 830 lines of code! Although this is quite a lot for this process, most of the code on the GROW unit is for formatting the customized menu system and being able to handle the input from the keypad. On the SECANT unit most of the code is for the serial terminal when the unit is put into independent mode. The purpose of putting so much work into the code along is to provide for a simple user interface that would allow any kind of user, no matter what technical skill they have, to be able to operate this system. Even if the end user needed to recompile the code and reprogram the controllers, it is very simple. To program the microcontrollers all that has to be done is copying the code into the Arduino IDE software, which is free, and simply hit the program button after an FTDI cable has been attached to the headers onboard of the PCB. The code for these units is quite extensive, and is mostly code that was custom written for this application. On the Arduino Forums website, bits of code were found about and pieced together, mainly to act as a setting or configuration process for the MPR121 chip. The original code for using the MPR121 keypad was provided by SparkFun Electronics, but I ended up using another code set (link can be found in the code comments which can be found on the GARDENS code page).

Even though this project was initially developed from a tech savvy friend, we quickly discovered its many uses, such as someone who has special needs yet wants to be able to garden and produce food for themselves. This unit would help them accomplish this goal by assisting them in day to day upkeep of their garden. Even though this design is a closed system, there was some thought into a system which was wireless and could link several greenhouses together, making a giant mesh network of GROW and SECANTs. The GROW's main purpose is to provide an easy interface to someone who wants to use the system. This is also why the SECANT can operate in independent mode, where a computer can directly talk to the unit, if that is deemed necessary.

## Conclusions

This project started out as a fun learning experience, and I believe it has now evolved into a process that could be mass marketed to companies (which is why key parts of this project are not posted on this website, including the code for each of the units) EDIT: I originally wasn't planning on posting the code, but as seen above I now have, and you can find it [HERE](#). As you will find on the GARDENS Code page, this project was intended to originally be open source, and used open source resources (code examples) to be built.

## Code of Ethics

This project was produced under the IEEE code of ethics. It does not pose a safety hazard to the public, their property, reputation, or employment by any means. It also was not built under any form of bribery. No discrimination took place while building this project and I also assisted my colleagues in building and developing their projects, while still maintaining honor to the IEEE code of ethics. There was no conflict of interest while developing this process. Safety was and always will be a top priority in any design I am responsible for or work on. My education of certain technological advances grew as I learned how to write code for custom menu systems, which is something I've wanted to learn how to do for awhile. All considerations for this project were given thought and no idea was immediately tabled from the design platform. Although there wasn't any point during this project that I needed outside assistance, from the beginning I would have asked for help in the event that a problem arose that needed assistance. The IEEE code of ethics is extremely important to follow, not only because it helps ensure safety of the public, but also provides a positive work environment to the professional.

## Steps needed to further complete the project

The GROW code needs to be modified so that when you enter the menu you are given an option to select the appropriate SECANT address to then modify values for. This extra step would not be hard to add into the system, just a matter of putting in the LCD formatting and inserting it before a call to the original menu function! Since I coded the software to be as general as possible, you can simply call to the functions that check for keypad entry! Also, different versions of the SECANT code would be nice to have for things like you see in Figure 7 where one of the units has a photo-sensor onboard and can turn on or off the lights. Ideally you would also have a Real Time Clock (RTC) inside the GROW unit so you could tell the system when to not activate during certain time periods. I would also want this project to be certified by ETL or some other major certification body before heading to market, mainly to provide assurance that this project can withstand conditions that might cause a failure and personal injury.

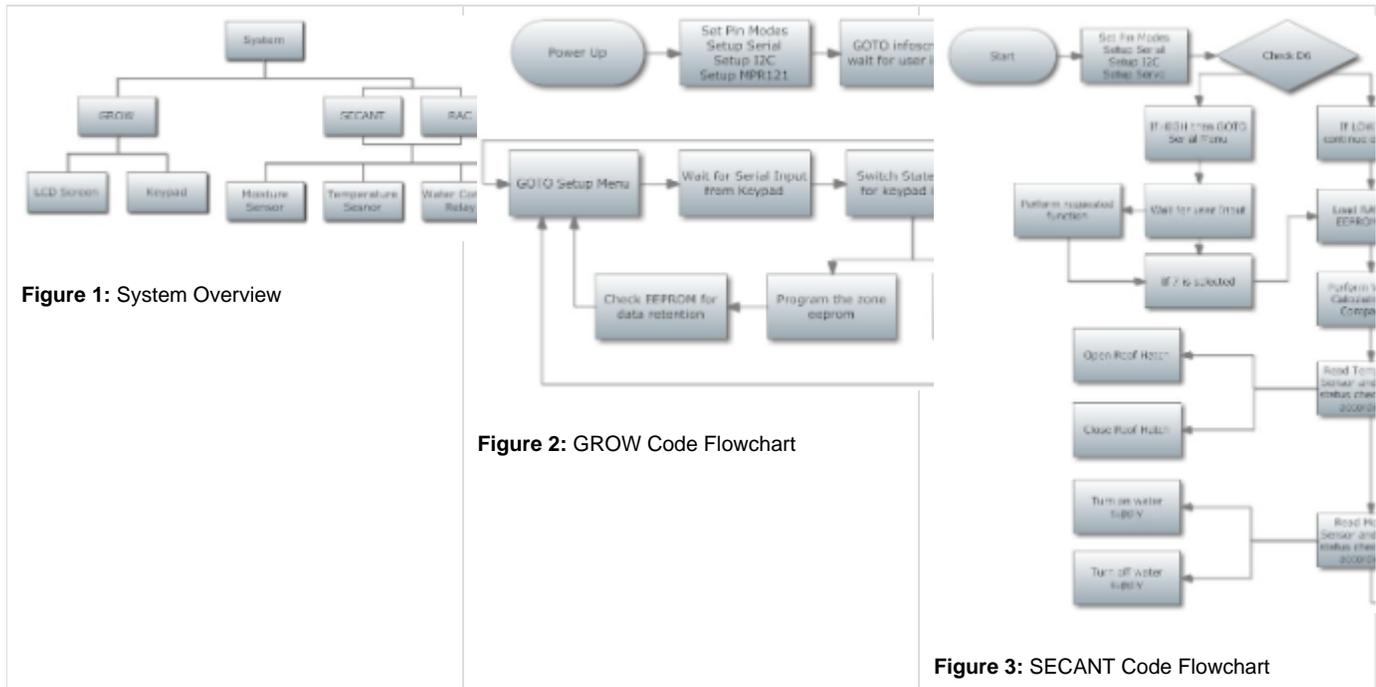
## So you want to replicate this project, huh?

I am happy that I got you interested! Make sure to read ALL of the information on the GARDENS Code page, along with taking a look at all the figures on this page. Figure 6 shows the budgets costs, which also can be used as the Bill Of Materials list. Here I will go over this to help clarify any questions you might have...

1. First on the list is the temperature sensor. This sensor has an analog output which is fed directly into the Pro Mini analog input pin. I talk more about this sensor on the GARDENS Code page.
2. Serial Graphic LCD from Sparkfun is used to show the user a custom menu. I picked the serial LCD because it only requires a single data connection from the Pro Mini, and its much simpler to program for using the Serial Graphic Library I found (which you can also look into by looking at the comments on the SECANT code)
3. Obviously we have the Arduino Pro Minis \:)
4. The MPR121 capacitive touch keypad was chosen because we initially had an idea to make a front panel (which we never go to) that would be completely smooth, with the keypad and LCD recessed on the other side. The only way would would be able to get a keypad to work like this was it had to be capacitive touch.
5. The Relay control PCB and Assorted Parts came from Sparkfun as a kit. I built this up and then integrated it with the GFCI outlet that I purchased at Home Depot.
6. The extension cord was cut up about 5 minutes after I bought it so I could wire the GFCI outlet through the Relay board and finally to a 120V 15A wall socket (its always fun to play with wall voltage... SAFELY!)
7. The pond pump was just to throw into the buckets of water, as seen in Figure 7
8. Although I put a 9V 1A power supply I found this wasn't enough for the circuit, so last minute I found a 12V 2A power supply lying around and used it instead. (Didn't have enough time to update the budget, but I know I would still be under)
9. The FTDI Friend and USB cable, both from Adafruit, are used to program the Pro Minis. I really REALLY like the FTDI friend because it allowed me to change the Vcc pin to 3.3V instead of the FTDI cables that you can purchase that, as I found out the hard way, have 5V on the Vcc pin EVEN THOUGH its a 3.3V logic level cable (who had THAT smart idea!?)
10. The Micro Servo was put into the red box (one of Sparkfun's boxes) to simulate the roof hatch
11. Various other components, like the wires, pull up resistors, etc!
12. The Vegetronix Moisture probe I talk more about on the GARDENS Code page
13. The Adjustable breadboard power supply takes the 12V 2A down to a 5V that I can pass into the RAW port of the Pro Minis
14. Finally we have the full sized and tiny breadboards that everything was tuck onto.

Well I hope that helped clear any remaining questions up, if you still want to know more information, you can email me [HERE](#)

Below are all the figures and a few YouTube videos of the system (they also include some explanation) followed by a PowerPoint presentation that covers more about the project.



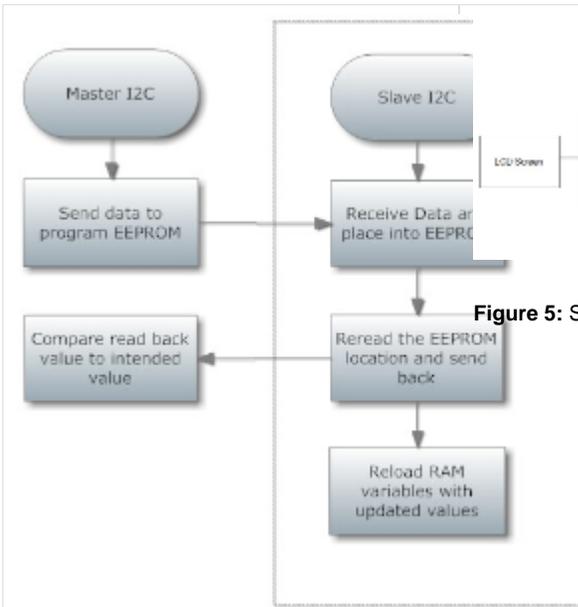
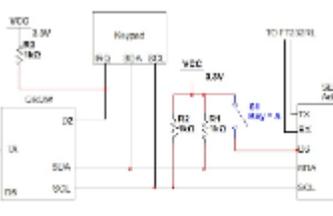


Figure 5: System Schematic



QTY	PN	DESC	MANUFACT	PRICE	UNIT
1	TMP112C12-ND	Analog Devices	DigiKey	10.000000	IC SENSOR TEMP 1.2V 5.0 TO 55 C
1	LCD-0533L	SparkFun	SparkFun	10.000000	Serial Display LCD 128x64
1	DSV-00020	SparkFun	SparkFun	10.000000	Arduino Pro Mini 5V 16MHz
1	SNV-00006	SparkFun	SparkFun	10.000000	MCP9802 I2C 1.8V 10kPa
1	COM-00006	SparkFun	SparkFun	10.000000	Relay Control PCB
n/a	n/a	SparkFun	SparkFun	10.000000	Assorted Parts for Bread Control PI
1	WS-17000-00W	Adafruit	Adafruit	14.990000	16MHz 32.430MHz Real Time Clock
1	QWEM18-88-88	Adafruit	Adafruit	10.000000	Colorful Cable 23 Pin I2C Extension
1	MDL1300	Total Phase	Wawa Board	10.000000	Total Phase 40 HP Submersible Fountain
1	85	n/a	Adafruit	10.000000	5VDC 1A regulated output regulator
1	28A	Adafruit	Adafruit	10.000000	5VDC 1A regulated output regulator
1	200	n/a	Adafruit	10.000000	USB Cable - Nylon
1	200	n/a	Adafruit	10.000000	Micro SD card
n/a	n/a	n/a	n/a	10.000000	Various other components (Resistors, etc)
1	1V4000	Vegetronics	Vegetronics	10.000000	High-precision low-noise moisture sensor
1	20A	Adafruit	Adafruit	10.000000	Adafruit's breadboard power supply
1	65	Adafruit	Adafruit	10.000000	Tiny Breadboard
1	23A	Adafruit	Adafruit	10.000000	Clear Full Size Breadboard

Figure 6: Parts Cost and Budget

Figure 4: I2C communication code flowchart

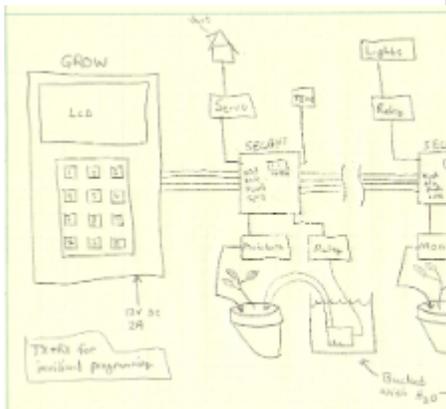


Figure 7: Full system layout

```

COM25 - PUTTY
UNIT SETUP MENU
1. Set Upper Temperature Threshold
2. Set Lower Temperature Threshold
3. Set Upper Moisture Threshold
4. Set Lower Moisture Threshold
5. Perform Servo Check
6. Perform Relay Check
7. Exit Setup Menu
Please Select a Function: 1
? <-- char read from buffer
? <-- Serial.available
? <-- after atol conversion

UNIT SETUP MENU
1. Set Upper Temperature Threshold
2. Set Lower Temperature Threshold
3. Set Upper Moisture Threshold
4. Set Lower Moisture Threshold
5. Perform Servo Check
6. Perform Relay Check
7. Exit Setup Menu
Please Select a Function: 1

```

Figure 8: Slave independent mode terminal monitor

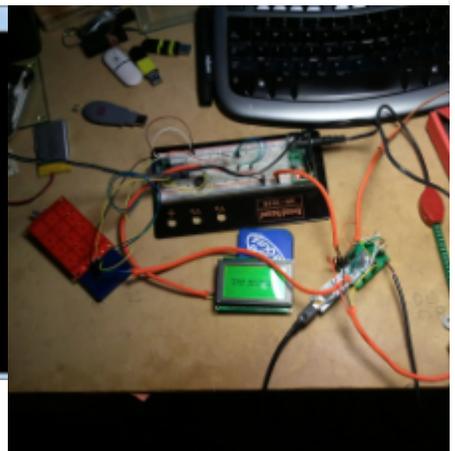


Figure 9: System Layout Picture moisture sensor

Figure 1: A flow chart of the system connections

Figure 2: GROW code flowchart showing its operation from power up

Figure 3: SECANT code flowchart showing its operation from power up

Figure 4: A flowchart showing how the units interact over the I2C protocol

Figure 5: General system schematic that shows interconnects to units and sensors

Figure 6: The Build of Materials and costs for this project

Figure 7: A hand drawing of the theory of operation for this project

Figure 8: A screen show from the serial terminal from a computer hooked up directly to a SECANT module

Figure 9: A picture of the system laid out on a desk showing the wiring and connections

Unknown macro: {you-tube}  
 Unknown macro: {you-tube}  
 Unknown macro: {you-tube}

Unknown macro: {viewppt}