Vertical Farm - Tasks

Patrick Schaffner
patrick.schaffner@unisg.ch

February 26, 2018

1 Frame

1.1 Construction

1.2 Electronics Integration

2 Automation

These tasks require programming a microcontroller, such as the Arduino or LoPy, and wiring up electrical components. Only rudimentary knowledge of programming and electrical engineering is required as step-by-step instructions are readily available on the internet.

Some facts about writing code that cannot be stressed enough:

- Programming is an iterative task. Try to start out as simple as possible, test your code thoroughly to identify errors early on, and only then add more functionality (which again should be tested immediately). Writing the final code in one go and then trying to figure out where the errors are is rarely successful, especially when electrical components are involved. If you have troubles finding an error, try simplifying your code until you have a minimal working example. The most promising strategy for getting help on programming websites is to post a minimal working example.
- Using Google is essential to software development. If you use Google once for every line of code, you're probably doing it right. There are too many details in programming to remember and there's no need to reinvent the wheel. Every task here has already been done by someone else, and every error you'll encounter has already been asked about somewhere. The difficult thing is using Google efficiently.

2.1 Sensors

Several sensors need to be wired up to a microcontroller for reading the state of the vertical farm. These tasks require following instructions to build an electrical circuit (using a breadboard, no soldering) and using the microcontrollers built-in functions to read voltage signals. Only little computation is necessary.

Additionally usage and necessary maintenance task shall be documented.

2.1.1 pH-Meter

Required items and resources:

- Arduino or LoPy (advanced)
- Analog pH sensor kit for Arduino
- Breadboard & jumper wires
- pH calibration solution
- pH meter manual
- *Advanced:* External 5V DC power supply.

Instead of an Arduino a LoPy microcontroller may be used. This will be more difficult and requires additional circuit elements (not documented here) as the LoPy runs on 3.3V while the sensor needs and returns 5V. In that case the external 5V power supply must be used and a voltage divider consisting of 2 resistors must be used to rescale the sensor output from 5V to 3.3V.

Instructions:

- Connect the sensor to the microcontroller's analog input pin.
- Read voltage values from the input pin.
- Convert voltage to pH value.
- Write function reading and returning pH values from the sensor.
- Calibrate sensor and add corrections to the code (get *correct* pH values).
- *Advanced:* Increase accuracy by powering the sensor with an external power supply instead of the microcontroller.

The manual includes all necessary circuit schematics, program code, and instructions to complete this task. Note that the given program code is more complicated/sophisticated than necessary. Breaking it down to the important bits would be a plus.

Advanced: Insert an external power supply by

- connecting the power supply's negative terminal to the microcontroller's ground,
- disconnect the sensor's positive terminal from the microcontroller and connect it to the power supply's positive terminal.

2.1.2 Temperature / Humidity Sensor

Required items and resources:

- Arduino or LoPy (advanced)
- DHT11 digital temperature & humidity sensor
- 5kΩ resistor
- Breadboard & jumper wires
- Adafruit DHT tutorial
- DHT11 manual
- idDHT11
- DHTLib
- Example code for LoPy

Instead of an arduino a LoPy microcontroller may be used. However, there might not already exist software libraries and you'll have to implement the highly time-critical protocol for yourself (see example code).

Instructions:

- Connect the sensor to the arduino according to the circuit schematics given in the tutorial/manual.
- Include a DHT software library to your arduino project.
- Try reading temperature and humidity values through the functions provided by the included library.
- Write a function that calls the library and returns temperature/humidity data.

The tutorial and manual show how to wire up the sensor. The manual also outlines the protocol used to communicate with it, but there already exist readily available arduino software libraries for that. Check the tutorial to see how to include software libraries to your project.

Two alternative libraries are listed above in case the tutorial doesn't work. See this on how to manually include libraries. Libraries always include example code in the "examples" folder.

2.1.3 Light Sensor

Required items and resources:

- Arduino or LoPy
- Photocell (light sensor)
- 10kΩ resistor (or higher)
- Breadboard & jumper wires
- Adafruit photocell tutorial

Instructions:

- Wire up photocell as outlined in the tutorial. Connect it to an analog input pin.
- Read analog voltage values from input pin.
- Convert it to a luminosity metric (lux if possible, or just "bright", "dim", and "dark").
- Write a function that reads values from the sensor and returns the luminosity metric.

2.1.4 Water Level Meter

Required items and resources:

- Arduino or LoPy
- Water level sensor
- Breadboard & jumper wires
- Water level sensor tutorial

Instructions:

- Wire positive and negative pins to 5V and GND.
- Wire the output pin to an analog input pin on the Arduino.
- Read analog values from the input pin. Higher values mean more water.
- Measure water level. Create a graph with water level on the X axis and input pin voltage on the Y axis.
- Write a function with no arguments that reads voltage values from the

sensor and returns the water level in cm.

2.1.5 Electrical Conductivity Meter

2.2 Pumps

2.2.1 Dosage Pumps

2.2.2 Main Pump

2.3 Lights

2.4 LoRa Communication

2.5 Control

3 Server

3.1 LoRa Communication

3.2 Frontend

3.2.1 Status Report

3.2.2 Command Panel

3.2.3 Device Management

3.3 Alerts