# Egg Incubator Project

Carl Chandler, PhD
Sun Aquasystems and Chandler Family Farm

January 21, 2025

## 1 Introduction

This is a project to document all the necessary parts for building and running an incubator at the scale of a small farm. This is arbitrarily determined as a roughly 200 egg capacity and must be reliable enough to run constantly. It also needs an sms alarm system so any errors can be quickly fixed. Commercial incubators currently on the market are too expensive for small farms, and hobby farm machines are also often very expensive for the level of reliability that they offer. [1]. DIY incubators are reasonably common but reliability and alarm systems are not typically documented.

This project is divided into 3 sections. First, the physical incubator CAD design, construction methods. Second, programming the control system and making the electronics work, Third, using a sms service and server service to host the alarm system remotely.

This project uses off the shelf components that are readily available, but does require certain skill and tools that are less common. The builder needs to be familiar with python programming, some electronics soldering, some 120v electrical work. Some parts are 3d printed in PLA or ASA and a table saw is useful for cutting wood and foam. Undoubtedly, making a custom circuit board for controls with a micro controller would be more cost effective at scale, perhaps this project could be useful for such a further project. For the time being, however, we concentrate on documenting a version that can be built by a handyman with some programming experience.

## 2 Design

### 2.1 Rationale

An incubator needs to do a few key things. It needs to maintain the proper temperature and humidity through out the whole egg area, it needs to recover quickly from changes in temperature and humidity such as when the door is opened. It needs to gently turn the eggs through at least 90 degrees every hour. It needs to let in a small amount of outside air to avoid CO2 buildup, and it needs to record the temperature, humidity, turning status so that the operator can verify that it is working correctly.

The temperature requirements ( 37c) mean that an insulated enclosure is needed, so the main body is built from foam panels. Eggs can be turned by tilting the egg trays or by placing them on rollers. Rollers can turn the egg through 360 degrees, but it's also possible for the eggs to slip and not move as much as expected. Thus we will go with tilting trays since if we verify that the tray has tilted, we can be sure that the eggs have been turned. We use two limit switches to verify that the eggs have been turned.

The heating is provided by incandescent light bulbs and air flow provided by two 12v computer cooling fans. Air exchange is provided by a 12v centrifugal blower that can open a weighted flap door when on, but allow the flap to seal with gravity when off.

Controls are provided through a dedicated PC or laptop using phidgets USB VINT. This allows for python control code, data logging and plotting, and sms alarms.

### 2.2 CAD

Detailed CAD file for cabinet and mechanical parts is found in this git directory, or at `https://cad.onshape.com/documents/54e720e4c56f1f3ff93b5703/w/6fa07bb4474c0df67681418a/e/e38df37c81685f3c348806ba?renderMode=0&uiState=678fe72e8840fe73946f891e` This is an as built file, but there are some oddities where I screwed up. I would make the cabinet 2" deeper in the future and avoid the strange cutout in the door. Door likely only needs to be 1" xps over the whole area. Door uses externally mounted spring latches, not in CAD file.

### 2.3 Electronics

Here is a crude drawing of the wiring for 12v, 120v, and logic level.1 Note the the VINT hub cannot directly power the logic of an SSR, needs to use the Digital Ouput Phidget which includes a voltage booster to 5v. Limit switches should be wired normally open, but it's easy enough to change in the code if they get wired normally closed instead.

### 2.4 Control code

The python control code is found in the code directory. The `Main_loop_V2.py` runs the control loop. `Python_server_alarms.py` is designed to run on a server and look at the public git repository for your incubator data. Then it uses Twillio to send text messages. The user can reply with Stop 1 to stop sms messages for 1 hour, or Stop n where n is any floating point number to stop messages for that number of hours. Text Reset to restart messages, or they will automatically restart after the time is up.

1. How to setup?
2. Make your own public git archive on github for your incubator.
3. Start the `main_loop_v2.py` inside this git archive to start your incubator. This will update the `today_dataV2.csv` file.
4. Use python anywhere or similar server to run the `Python_server_alarms.py`
5. Sign up for Twillio, there is a free trial of a few text messages, update the key in `Python_server_alarms.py` so it can send text messages.
6. It should now send an alarm if the data is out of the set range on the incubator, of if the incubator doesn't update for 10 min.

### 2.5 Parts list and tools

In this file is a spreadsheet for all the parts and cost. Some of the 3D printed parts are not listed here, but they are not expensive if one already owns a 3D printer. The cad file here are made to work with a 0.8mm nozzel printing ASA on p1s bambu labs printer. Other materials and settings might need adjustment of the fitting tolerances on the sliding parts and around the motor.

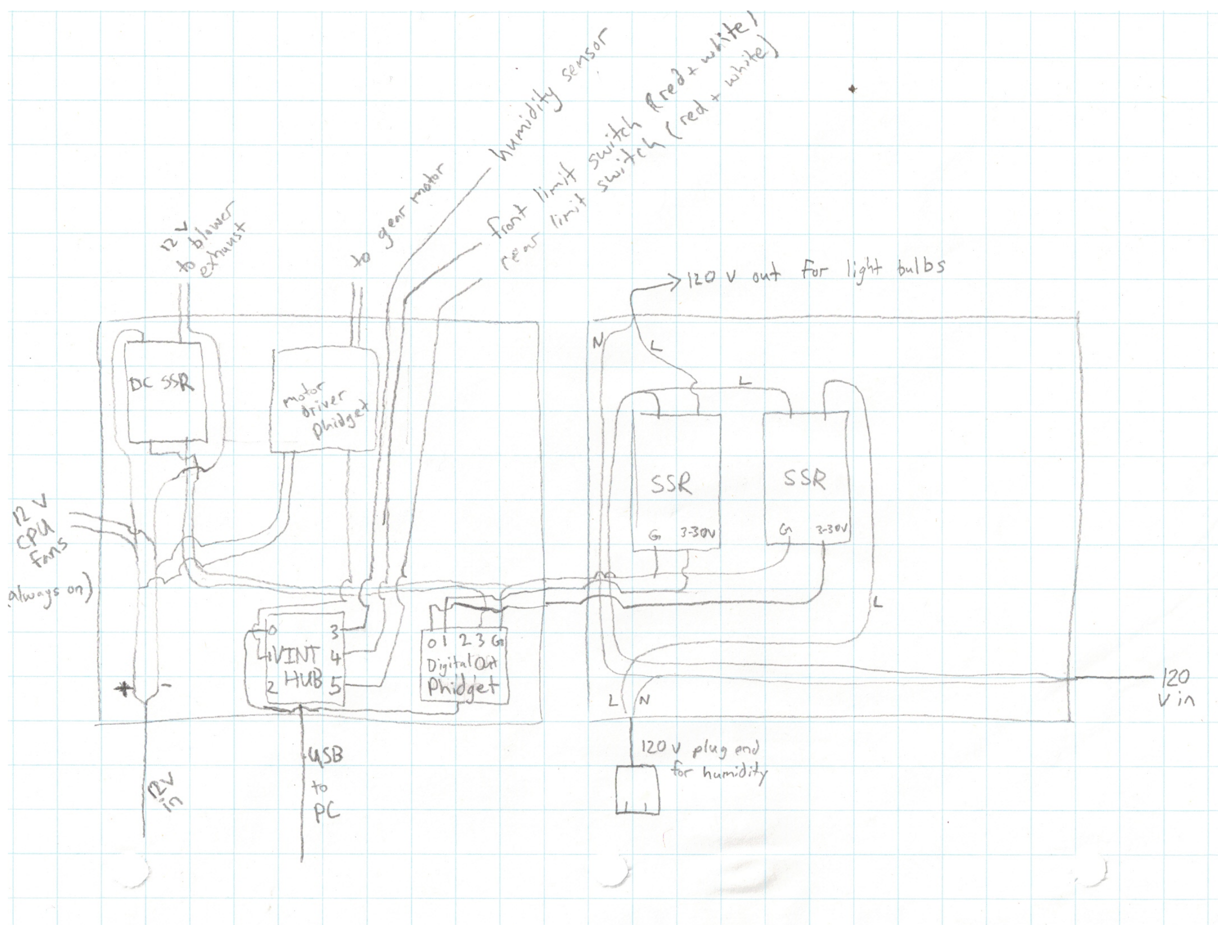


Figure 1: Wiring for electronics in incubator

## References

[1] B. Hill. (2025) sportsman professional incubator, 1449.99 cad. [Online]. Available: https://berryhill.ca/products/digital-gqf-sportsman-professional-incubator