

## Smart Plant Controller



by Joren vanGoethem

I had an idea of creating a fully automatic plant controller. So essentially it makes sure the plant has water at all times, has enough light and even heats the soil if it's too cold.

Seems fun right? Creating something that can grow plants and food for you!

I will also incorporate a camera that takes pictures and these will be visible on the website so you can monitor the growth.

It will also run a website on the Raspberry Pi with a database to keep all the sensordata and images and so you can make changes to the settings and see the sensordata in realtime.

let's get to it, starting with the supplies.

depending on where you buy them and how long you are willing to wait for materials it will cost about 200 euro's (this is including the raspberry pi itself)

Do mind that a 1GB pi is a lot cheaper than a 4GB pi and will be more than enough for this project.

### **Supplies:**

### **Components:**

- Raspberry Pi 3 or 4
- Raspberry Pi + Power Supply
- SD Card, minimum 16GB
- T-cobbler for Pi
- Pi Camera (optional)
- 16x4 LCD (optional)
- DS18B20 Temperature Sensor
- Light sensor(s)
- soil moisture sensor
- small water pump
- Heating Element
- LED's or led strips
- 3x 5 Volt relay
- PCF8574 I2C chip (only required when you use an LCD)
- MCP3008 chip
- 10kOhm potentiometer
- 2x 10kOhm resistors
- 1x 4.7kOhm resistor
- 1x 220Ohm resistor
- 1x 330Ohm resistor
- Jumper Cables

#### Tools:

- Soldering Iron (not required)
- Multi Meter (not required but usefull for troubleshooting and testing)

in the rar file below is a complete list with pricing and links on where to buy components.



<https://www.instructabl...>

Download

## Step 1: Putting the Hardware Together

I made a fritzing schema so it's easy to recreate, I uploaded the scheme below.

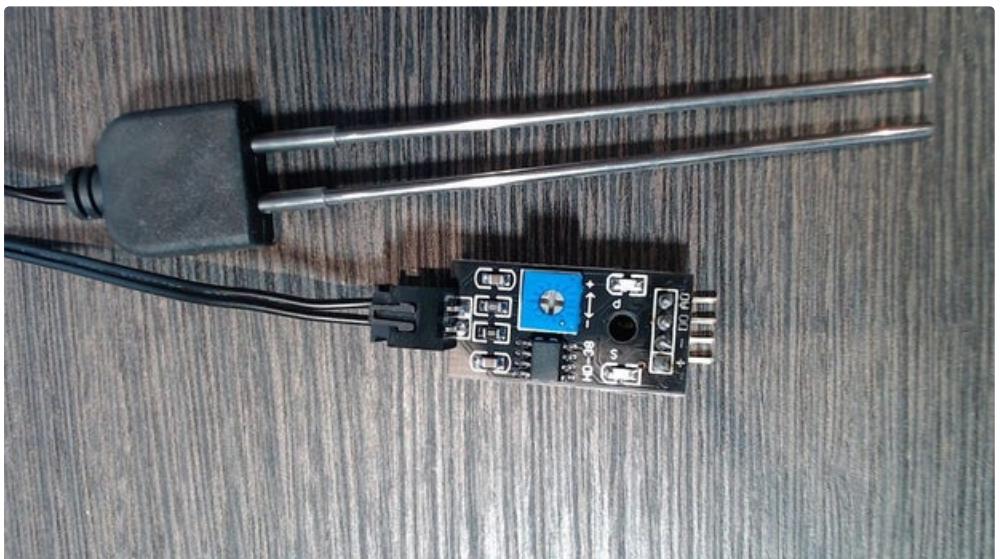
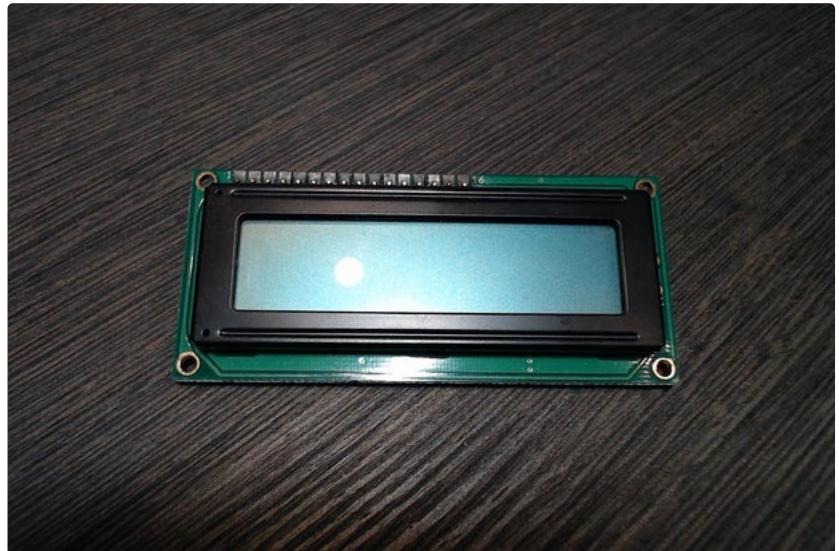
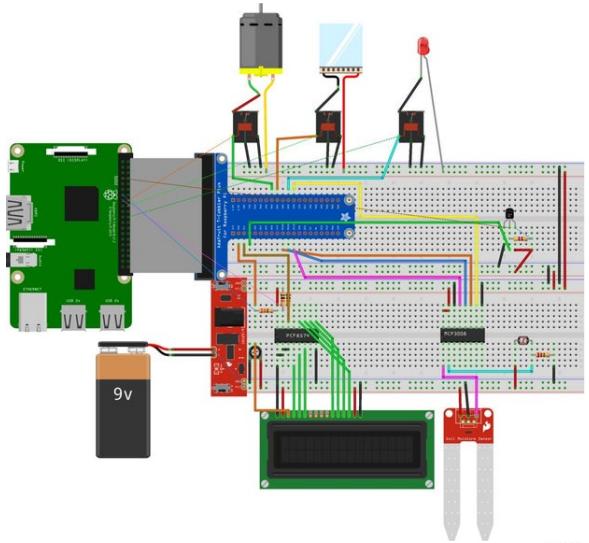
The circuit has a **Light Sensor**, a **Temperature Sensor** and a **Soil Moisture Sensor**.

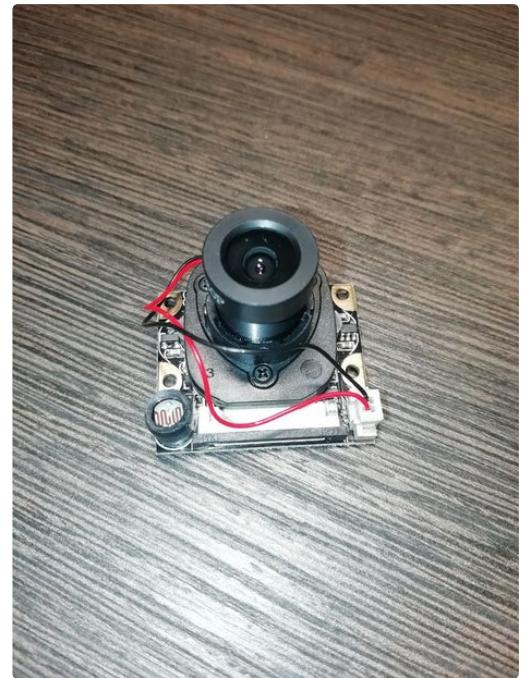
The data they output will then be used to activate **Lights** (I used LED strips, you can use whatever you prefer and maybe even a real grow light for plants), a

**Heating Element** and a **Pump**.

There is also a **PiCamera** that will take pictures every set period to monitor plant growth.

the **LCD display** is used for showing the IP addresses of the pi.







<https://www.instructabl...>

Download

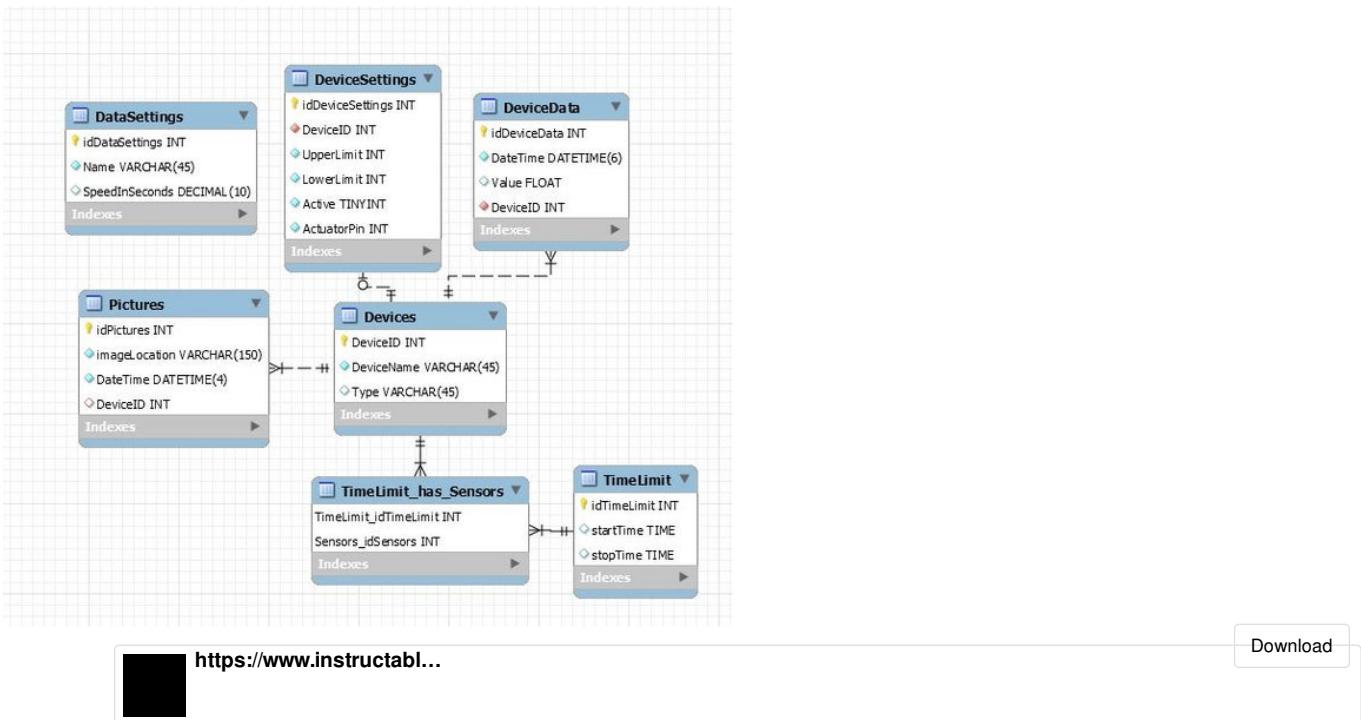
## Step 2: Creating the Database

you can see my **ERD diagram** above, below you will find the model file to import the database yourself.

i also included my SQL file with all data i have been collecting for the last few days.

the database stores information and settings for all the devices such as:

- sensor settings
- sensor data
- picture locations
- etc..



## Step 3: Setup a Connection With the Pi

you can use your development software of choice, I used **atom** and **Visual Studio Code**.

you can first test if you can connect using a command prompt, type "**ssh USERNAME@IPADDRESS**"

It will also prompt you for a password and then connect.

Both require you to get one of the extensions that

```
Command Prompt
Microsoft Windows [Version 10.0.18363.900]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\SKIKK>ssh pi@192.168.1.77
```

allow ssh connections. for VS Code you will then have to add a new connection and type "**SSH USERNAME@IPADDRESS -A**"

It will prompt you for a password and then it should connect.

For a complete guide you can follow [these](#) instructions on VS's website.

## Step 4: Setting Up the Software on the PI

first off start with **updating your Pi** and then extract the rar file I included below.

there will be 2 other folders in the rar file, one called Code and one called HTML and i will explain where to put these on your PI in the next steps.

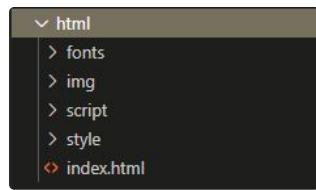
## Step 5: Setting Up the Webserver

We're going to install an **Apache webserver**. By doing so you will be able to open the website on any device connected to your network.

Type the following command and press enter: "**sudo apt-get install apache2**"

Now go to the folder: **/var/www/html/**

Here can you place all your files from the html folder you extracted in the previous step. do NOT put the html directory IN **/var/www/html/**  
you need to put the files from the html folder in



**/var/www/html/**

the index.html page will be opened whenever you browse to the IP of your Pi. Be careful not to type a capital I in index.html if you do not use my files otherwise it might not automatically open the index page.

if you get any issues about permissions later you might have to run this command: "**sudo chmod 644 /var/www/html/**"

## Step 6: Python Software

I made multiple python scripts. now you will need the 'Code' directory for all the python files

I coded use some classes to get the sensordata, activate actuators and to display my IP addresses on the LCD.

Eventually I use only one file to make the full project work, this is named app.py. In this file I coded some routes so I could read data from my database and sent in to a json object which I then used in my

javascript files.

I also used socketio to show the sensordata in 'realtime', the speed of this can be adjusted in the settings on the website.

This directory does not need to be in a specific place on the pi, but I recommend that you put it in the home directory of your user.

## Step 7: Creating a Service So It Runs on Boot

To make the project start up on boot we will make it a service, to do this make a new file with a name you like and the extension .service  
i'd recommend putting this in /lib/systemd/system/

in this file you need to put something like this:

#### [Unit]

**Description=choose a descriptions**

**After=multi-user.target**

#### [Service]

**Type=idle**

**ExecStart=/usr/bin/python3**

/path/to/your/app.py/file

#### [Install]

**WantedBy=multi-user.target**

you can change the Description and the ExecStart to the correct locations.

now run these commands on the new service file:

**"sudo chmod 644 /filelocation/file.service"**

"sudo systemctl daemon-reload"

"sudo systemctl enable NAME.service"

## Step 8: Final Touches

Everything should be good to go so now you can start by planting seeds or a small plant and watch it grow!

I will post pictures of my case here as an inspiration but I would like to see what you can come up with. let me know if you recreated my instructable and how your design looks!



