Greenhouse project

Embedded Operating Systems VIA UC

November 2017

Silvija Krupaviciute

Dmitry Rachkovsky

Table of Contents

[Introduction 2](#_Toc498971358)

[Greenhouse Model 3](#_Toc498971359)

[Light Intensity Sensor 5](#_Toc498971360)

[Circuit 5](#_Toc498971361)

[Pins 5](#_Toc498971362)

[Code 5](#_Toc498971363)

[Usage 5](#_Toc498971364)

[Light Intensity Control 6](#_Toc498971365)

[Circuit 6](#_Toc498971366)

[Pins 6](#_Toc498971367)

[Code 6](#_Toc498971368)

[Usage 6](#_Toc498971369)

[Temperature and Humidity Sensor 7](#_Toc498971370)

[Circuit 7](#_Toc498971371)

[Pins 7](#_Toc498971372)

[Code 7](#_Toc498971373)

[Usage 7](#_Toc498971374)

[Servo Motor Control 8](#_Toc498971375)

[Circuit 8](#_Toc498971376)

[Pins 8](#_Toc498971377)

[Code 8](#_Toc498971378)

[Usage 8](#_Toc498971379)

[Heater Control 9](#_Toc498971380)

[Circuit 9](#_Toc498971381)

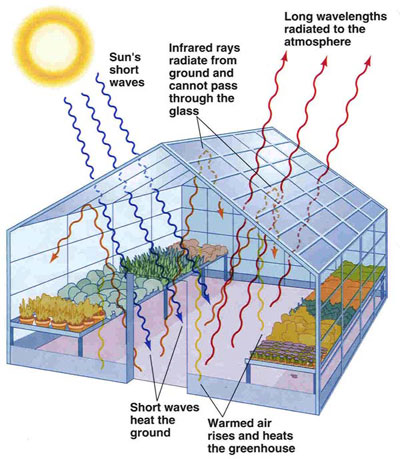
[Pins 9](#_Toc498971382)

[Code 9](#_Toc498971383)

[Usage 9](#_Toc498971384)

# Introduction

There are several aspects that need to be taken in consideration while creating a greenhouse control system. The greenhouse flora need optimal levels and duration of **daylight** for photosynthesis and plant respiration. **Ventilation** is also one of the most important components in a successful greenhouse in order to prevent bacteria and infections due to humid environment, regulate the temperature when it gets too hot and ensure air movement to prevent buildup of necrotic fungus. Finally, the **heating** is important for the greenhouse to operate properly in colder climates. Therefore, the main aspects that need to be considered are:

Monitoring:

* Humidity
* Temperature
* Daylight

Environment Control:

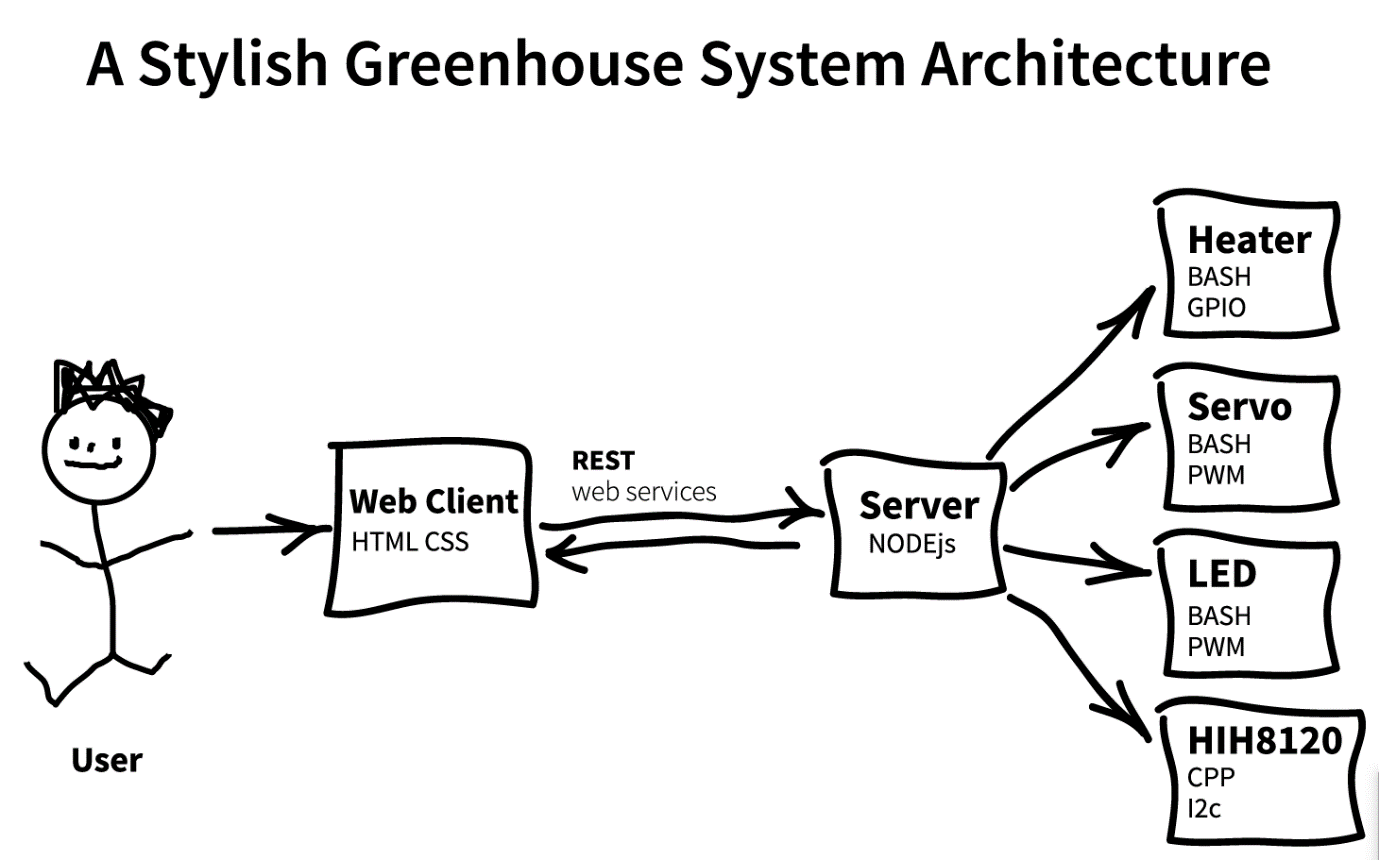
* Window Opening
* Heater
* Artificial Daylight

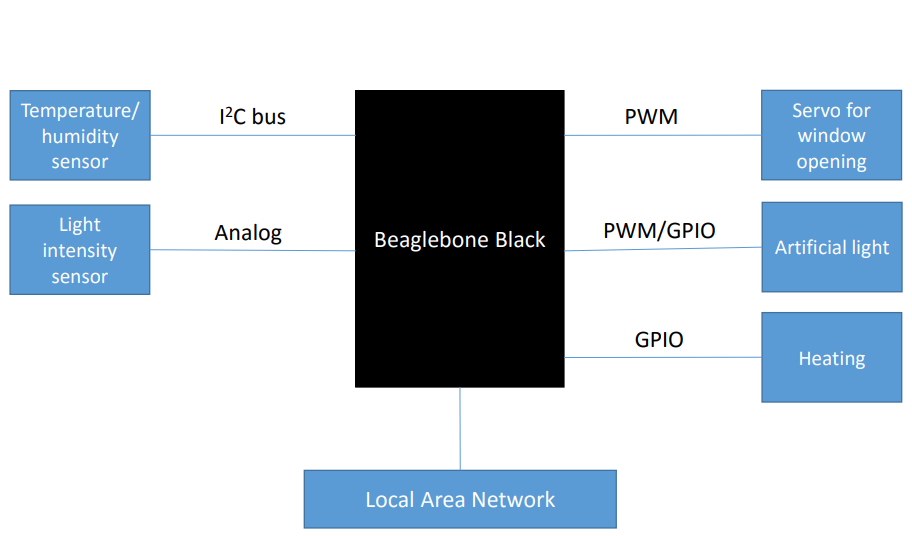
User Interface:

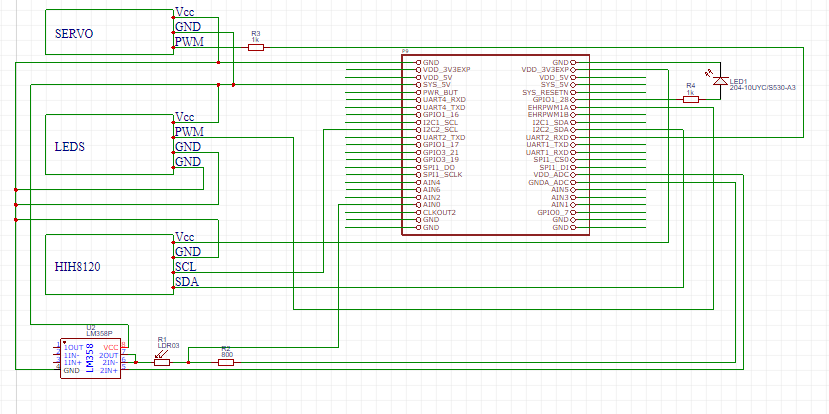
* Web Interface for Monitoring and Controlling remotely

# Greenhouse Model

|  |  |
| --- | --- |
| Feature | Technology |
| Measure temperature | I2C bus |
| Measure Humidity | I2C bus |
| Measure Light Intensity | Analog |
| Control servo motor for window | PWM |
| Control heater | Digital on/off |
| Control light intensity | PWM |
| Monitor and control remotely | Node.js server, RESTful API |

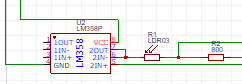






# Light Intensity Sensor

## Circuit



## Pins

P9\_07 – voltage

P9\_01 – ground

P9\_32 – ADC voltage

P9\_34 – ADC ground

P9\_39 – raw value

## Code

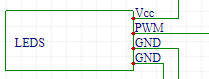
Controlled via a bash script ‘get\_light\_intensity.sh’

## Usage

Script used to get light intensity in percentage.

# Light Intensity Control

## Circuit



## Pins

P9\_07 – voltage

P9\_01 – ground

P9\_14 – PWM

## Code

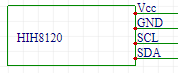
PWM controlled via bash scripts ‘set\_led.sh’ and ‘get\_led.sh’.

## Usage

Setting and getting current intensity of the LED stripe in percentage.

# Temperature and Humidity Sensor

## Circuit



## Pins

P9\_04 – voltage

P9\_01 – ground

P9\_19 – SCL

P9\_20 – SDA

## Code

I2C controlled via C++ compiled binary ‘hih8120.bin’

## Usage

Binary allows to get current value of temperature and/or humidity read by sensor.

# Servo Motor Control

## Circuit



## Pins

P9\_07 – voltage

P9\_01 – ground

P9\_22 – PWM

## Code

PWM controlled via bash scripts ‘set\_lid.sh’ and ‘get\_lid.sh’. PWM is not enabled all the time – it only enables to move the motor and then disables again.

## Usage

Setting and getting current position of the box lid in percentage – 0 for completely closed and 100 for completely opened.

# Heater Control

No heater was given to us so we just used an LED.

## Circuit



## Pins

P9\_02 – ground

P9\_12 – output

## Code

Controlled via bash scripts ‘get\_heater.sh’ and ‘set\_heater.sh’

## Usage

Setting ang getting current “heater” status – 0 for off, 1 for on.