

# Remote Controlled Modularised System

---

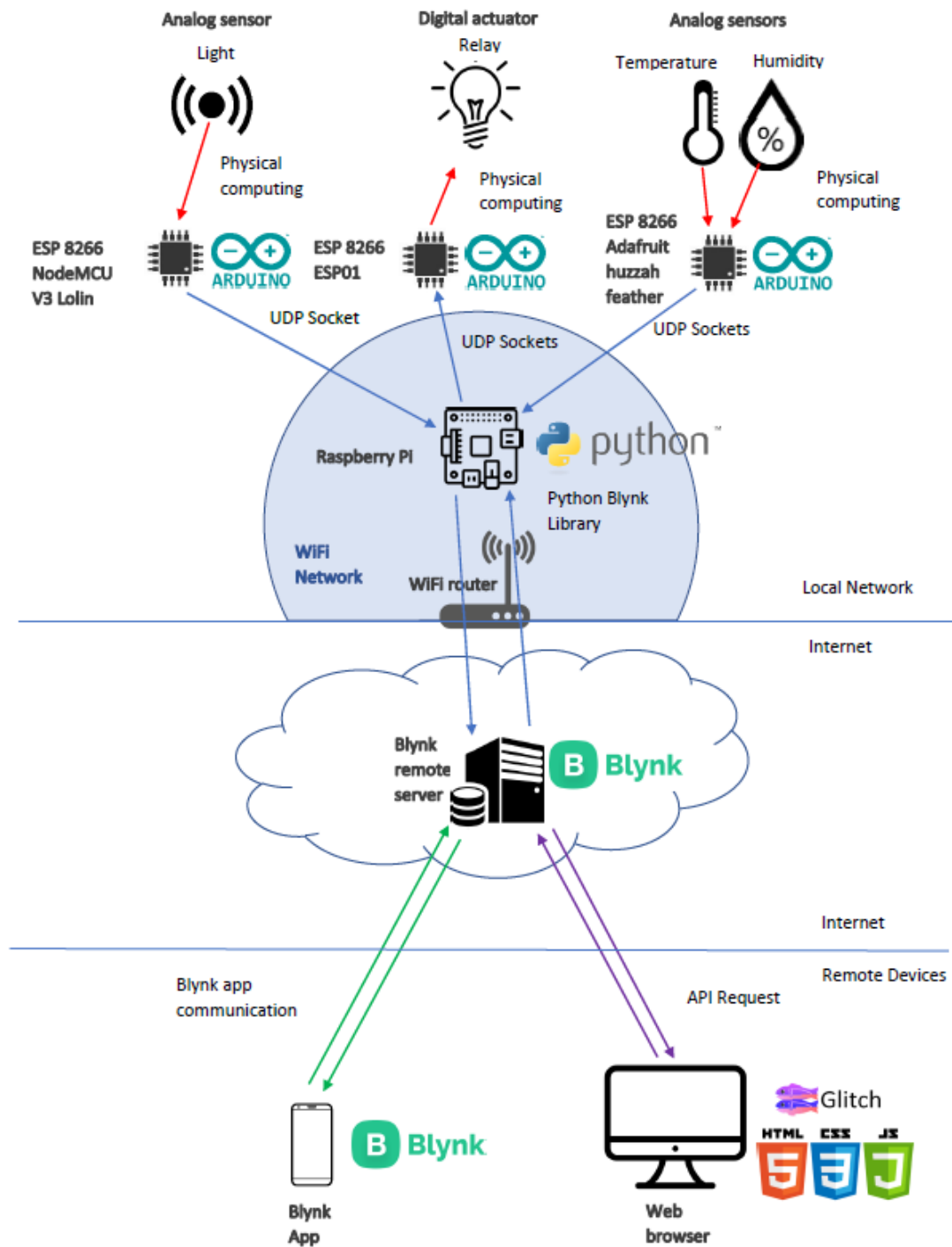
## Project description

In 2015/16 I studied a Post-Graduate Diploma in Business and Management at Queen's University of Belfast. As a final project I designed a business project in which I had to determine a possible way of building a home automation start up business. The report was focussed more on the business point of view rather than the technical aspect. The idea of business was to develop a home automation system. This mentioned previous business report describes a system that would be divided on a local network and a remote accessibility. The local network would be composed by a main hub which would be connected to other modules around the house using power line communications or wireless communication. These modules would be sensors and actuators that should be able to be added or removed from the system easily. The local system would have internet access to allow remote accessibility and monitoring from smart devices such as phone, tables or laptops. The full report can be found in the git repository:

<https://github.com/AlvaroSanchezDomingo/Computer-Systems-IoT-Project/blob/master/Documents/Business%20project.pdf>

This time I have focused on the technical aspect of this idea of business. I want to be able to prototype this idea of home modularised home automation system with my current available resources. I have created a main hub (Raspberry Pi) controller that handles the communication between the local modules (esp8266 chips) and the remote platform (Blynk). I have created three modules connected with the main hub via Wifi. These modules capture ambient data using analogue and digital sensors but also interact with physical world by using actuators. A remote platform manages the data from the local hub and also would allow monitoring and controlling the local system from anywhere in the world using a smart device or an internet browser.

## Project graphic description



## Main Hub

A Raspberry Pi is used as a local hub to handle the communications with the local wireless modules and the remote Blynk platform. The Raspberry Pi Hub hosts a program written in Python that will manage the Blynk communication and will run a UDP server-client communication to receive and transmit data with the local wireless modules.

Controller: Raspberry Pi 3

Program: Python

```
Module 1 -> 1
Reply: OK from: ('192.168.137.2', 8888)
Module 3 -> light
Reply: 70.02 from: ('192.168.137.4', 8888)
Module 2 -> temperature
Reply: 22.00 from: ('192.168.137.3', 8888)
Module 2 -> humidity
Reply: 52.00 from: ('192.168.137.3', 9999)
█
```

<https://github.com/blynkkk/lib-python>

## IO Modules

Several ESP8266 chips are used to control and monitor the actuators and sensors. One module is going to be an ESP8266 huzzah feather from Adafruit Wifi module. Other module is going to be developed on an ESP8266 ESP01 Wifi chip. Last module is going to be developed on a ESP8266 NodeMCU WiFi chip. These modules can be programmed using Arduino IDE including the "WiFiUdp.h". This library allows creating UDP client-server communication. These modules have GPIO built in the board, therefore it allows implementing the physical computing needed to get information from the outside world and to control physical actuators. Two of these modules will be sensor module, they will send ambient information to the hub (Raspberry Pi). The other module will be an actuator module, it will receive commands from the hub.

### Module 1, RELAY

A relay is used as a digital actuator connected to the GPIO of ESP01 chip. This relay is connected to a lamp.

Controller: esp8266 ESP-01

Program: Arduino IDE

Hardware: esp-01/01s relay v4.0



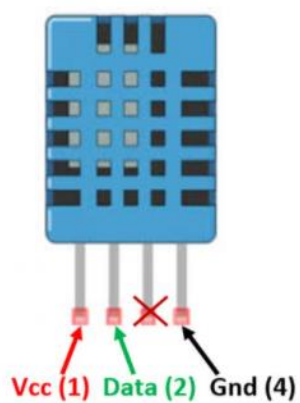
## Module 2, TEMPERATURE AND HUMIDITY

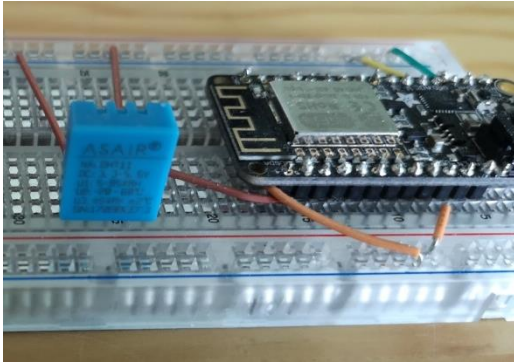
A DHT sensor is connected to the GPIO of the Adafruit huzzah feather chip. This sensor detects the ambient temperature and humidity.

Controller: esp8266 Adafruit huzzah feather

Program: Arduino IDE

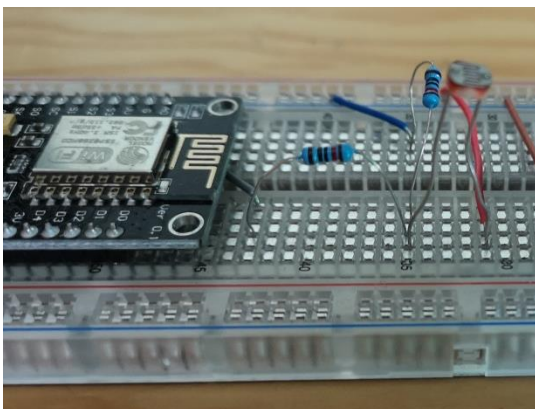
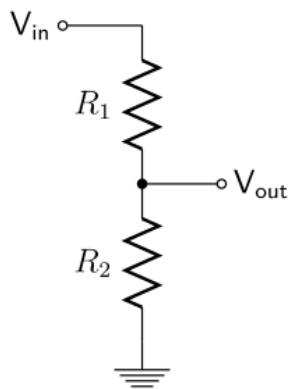
Hardware: DHT11 sensor





### Module 3, LIGHT SENSOR

A photocell sensor is connected to the GPIO of the NodeMCU chip. The photocell is used to sense the ambient light. The operation of the photocell is done using a voltage divider ( $V_{out} = V_{in} * R_2 / (R_1 + R_2)$ ), we can then sense the ambient light level on the output.



Controller: esp8266 NodeMCU

Program: Arduino IDE

Hardware: Photocell, 2x 10k Ohms resistors.

## BLYNK

For this project I am using Blynk as a remote platform to handle the information with the local IoT system and allow control and monitoring functions to mobile devices and laptops. I have decided to use Blynk for few reasons.

- Blynk requires very little code on the raspberry pi to establish communications with the servers.
- Blynk allow third party applications to interact with its API.
- Blynk allow to create mobile app very easily and user friendly.
- It is very easy to use and configure which makes prototyping quicker and easier.

Blynk app is used to develop a simple mobile app to monitor and control the local IoT system. Blynk can be downloaded for Android and IOS and allows creating an easy interaction with the hub and it is free.



## WEB APP

The idea is to create a Web App accessible from any browser, thus it will be accessible from any device with a web browser. I want to use my knowledge acquired during the summer with the module ICT skills in which we used glitch, JavaScript, HTML and CSS to develop a Web app. The web app will need to interact with Blynk platform using its API.

<https://blynkapi.docs.apiary.io/#>

## Landing page

IoT Dashboard	Signup	Login
---------------	--------	-------

GitHub Repository

## Signup

IoT Dashboard	Signup	Login
---------------	--------	-------

**Register**  

First Name

Last Name

Email

Password

Submit

## Login

IoT Dashboard	Signup	Login
---------------	--------	-------

**Log-in**  

Email

Password

Login

## Dashboard

IoT Dashboard	Dashboard	Logout
---------------	-----------	--------

**Dashboard**

Relay	Temperature	Humidity	Light
<input type="checkbox"/>	23.00 C	53.00 %	50.20 %

## Web App access

Access the web app with the following link:

<https://iot-dashboard-alvaro-sanchez.glitch.me>

View code:

<https://glitch.com/edit/#!/iot-dashboard-alvaro-sanchez>