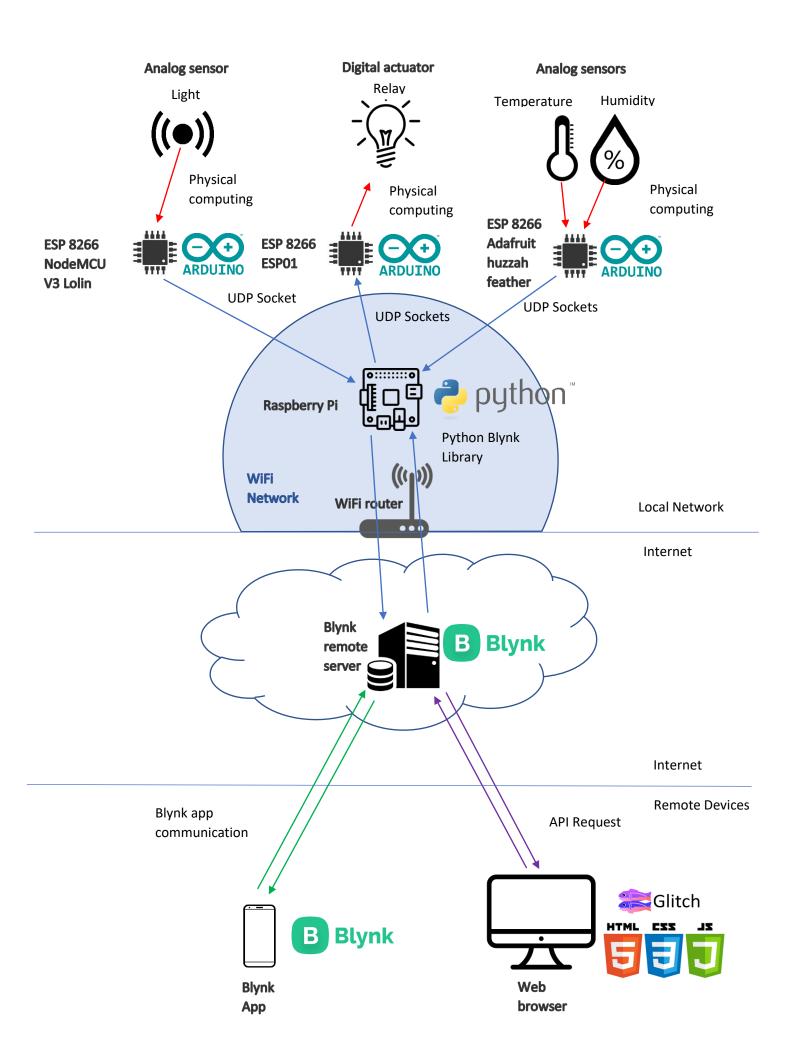# Remote Controlled Modularised System

Alvaro Sanchez Domingo 20091387
22/12/2020

Repository:
https://github.com/AlvaroSanchezDomingo/Computer-Systems-IoT-Project

Analog sensor
Light
Physical computing
ESP 8266 NodeMCU V3 Lolin
ARDUINO
UDP Socket

Digital actuator
Relay
Physical computing
ESP 8266 ESP01
ARDUINO
UDP Sockets

Analog sensors
Temperature   Humidity
%
Physical computing
ESP 8266 Adafruit huzzah feather
ARDUINO
UDP Sockets

Raspberry Pi
python™
Python Blynk Library

WiFi Network
WiFi router

Local Network

Internet

Blynk remote server
B Blynk

Internet

Remote Devices

Blynk app communication

API Request

B Blynk
Blynk App

Web browser
Glitch
HTML CSS JS

Sensor and actuator level

A relay is going to be used as a digital actuator connected to the GPIO of one of the modules. A presence sensor is going to be used as a digital sensor connected to the GPIO of one of the modules. A DHT sensor is going to be connected to the GPIO of one of the modules. This sensor detects the ambient temperature and humidity.

Processing modules

Several ESP8266 chips are going to be 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.

Communications HUB

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

Remote server

For this project I am going to use 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.

Mobile application

Blynk app its going to be 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. I will need to set it up in such a way that can request information using JavaScript.