# **INTERNEt Of THINGs : Raspberry Pi Home automation System based on IoT**

Home automation system will control home appliances in manual and automatic mode as per user choice. It may also include home security such as access control and alarm systems. What we have done is we have placed sensors like temperature, light and PIR to record conditions of surroundings. Algorithm can take decision based on that data, optimally in automatic mode without user interaction. User will always get to know the status of home appliances.

We can control home appliances via phone from anywhere in the world or let it get controlled automatically.

## **Architecture**

This Project can be divided into 3 modules as shown :

* End User(Phone)
* Raspberry Pi End (Home appliances)
* Server/Cloud (via which both part can communicate)

## **User End**

The interface for user end is made using python's kivy framework. User can control appliances manually using the mobile/desktop application in manual mode or can let them get controlled automatically in automatic mode.

Some glimpses of our user end :

main.kv,main.py and also the client\_secret.json file that you will get while setting up your server with Google Spreadsheets API are the required at user end.

### **Installation**

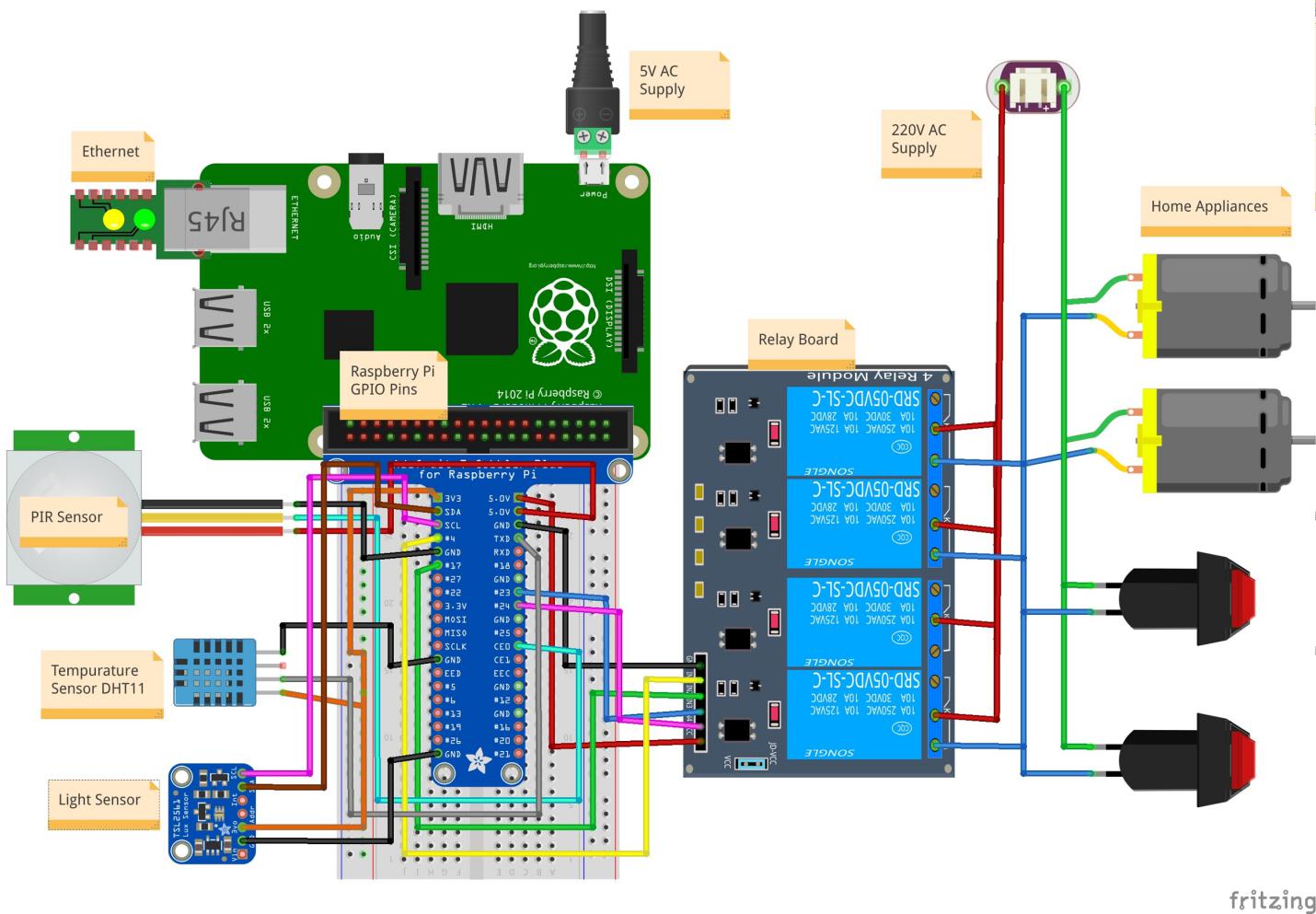
python -m pip install --upgrade pip wheel setuptools  
python -m pip install docutils pygments pypiwin32 kivy.deps.sdl2 kivy.deps.glew  
python -m pip install kivy.deps.gstreamer  
python -m pip install kivy  
pip install pygame  
pip install oauth2client  
pip install gspread

## **RaspberryPi End**

This module contains the code for fetching data from server and changing state of appliances accordingly. In automatic mode it will utilise the output of sensors to control state of devices.

Raspend.py is required at RaspberryPi end.

### **Connections**

****

### **Installation**

pip install oauth2client  
pip install gspread  
git clone https://github.com/szazo/DHT11\_Python.git

go inside DHT11\_Python and copy dht11.py and **init**.py inside your project folder.

## **Google Spreadsheet setup(Server)**

For communication between mobile application and Raspberry Pi via internet there should be a server, here we have used Google Cloud as our database server.

The following steps are required to be followed to setup the database server :-

1. Go to the Google APIs Console.
2. Create a new project.
3. Click Enable API. Search for and enable the Google Drive API.
4. Create credentials for a Web Server to access Application Data.
5. Name the service account and grant it a Project Role of Editor.
6. Download the JSON file.
7. Copy the JSON file to the code directory and rename it to client\_secret.json.
8. Find the client\_email inside client\_secret.json. Back in spreadsheet, click the Share button in the top right, and paste the client email into the People field to give it edit rights. Hit Send.

## **What is Internet Of Things?**

IoT is the most trending and important topic nowadays. The concept is, devices of a surrounding get the capabilities to communicate with each other, with the Internet and can take decision smartly based on that communication. Isn’t it exciting? For example: How about devices of your home can communicate with each other via a black box technology and can take the decision when they have to switch on and off devices smartly. Like you entered the room and lights, fans get automatically switch on / off depending on the intensity of light and temperature of the room. You don’t even have to touch switches. “SMART”. Even more, you can check the status of your home devices from anywhere in the world and can control them also via cell phone, web etc.

**Power Of IoT :**

From the previous paragraph its easy to measure the power of IoT technology. Let's dive into it more in terms of big data, finance, and smart machines. As all of the devices begin to store data about our activities, they will begin to understand our lives. Data from all the connected devices from the bigger organization can get collected and analyzed by data analysts and then can apply advanced machine learning algorithms to make smart decisions by those connected things.

A great example is Google’s application of machine learning and IoT to its data centers last year. Data centers need to remain cool, so they require vast amounts of energy for their cooling systems to function properly (or you could just dunk them in the ocean). This represents a significant cost to Google, so the goal was to increase efficiency with machine learning.

With 120 variables affecting the cooling system (i.e. fans, pumps speeds, windows, etc.), building a model with classic approaches would be a huge undertaking. Instead, Google applied machine learning and cut its’ overall energy consumption by 15%. That represents hundreds of millions of dollars in savings for Google in the coming years.

**How does Internet based Raspberry Pi Home automation works ?**

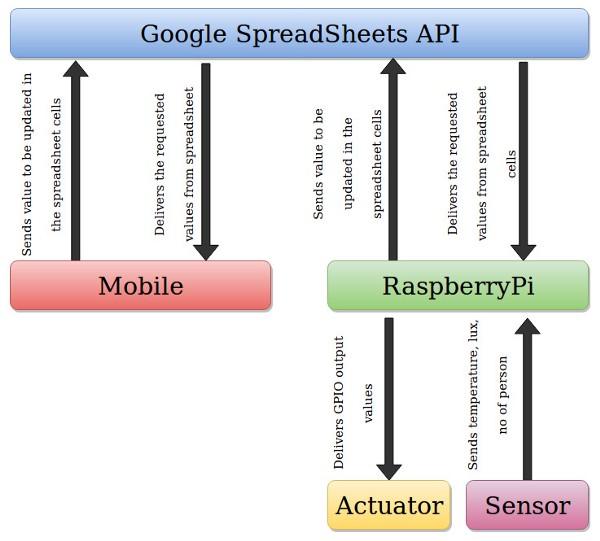
**FLOW DIAGRAM**

Home automation systems you can use to control appliances in your homes like lights, door locks, and air conditioning through a web interface or smartphone application. A lot of technologies are being developed around this concept, such as independent lightweight IoT networks and protocols for passing data.

Here in this project, its gonna explain how to make it possible as a real deployable project for your home/room by using Raspberry Pi and Google cloud. i.e. control your home appliances via phone from anywhere in the world or let it get controlled automatically.

This Project can be divided into 3 modules as shown:- 1) End User(Phone), 2) Raspberry Pi End (Home appliances), 3) Server/Cloud (via which both parts can communicate). Let’s do our hands dirty with IoT now ;)

#### **Server/Cloud :**

****

Data Flow Diagram

You can definitely choose any server or cloud kind of thing for communication between both cell phone and appliances. It can be your own server too, since we have to access devices from anywhere in the world there should be one . In this Project, we will use Google cloud (gspreadsheet) database for the same.

To use Google sheet database for the project you first need to create a project on Google developer console and get the credentials for using Google Sheets API.

Follow the guidelines as written :

1. go to [Google developer console](https://console.developers.google.com/projectselector/apis/dashboard) .
2. click by clicking on “*Create*” , create new project .
3. Give name of your project as you want and next .
4. Add or *enable Google Sheets API* to the same project .
5. After enabling the sheet API , *create credentials* by clicking on the same and then choose settings as *server* , then select to access *application data.*
6. then after that in create service account , give name as per your choice and select role as Project -> editor .
7. Done , Download json file and rename it as ‘client\_secret.json’ . In that json file there should be a client\_email field , copy that email address .
8. Create sheets on the Google drive and share them withe same client\_email as we got in previous steps . sheets to be created -> mode, client, status . These three sheets we will be using for the project .

Cheers !!!!

Now we are done with setting up of the database on Internet(cloud) and fully ready to use it also. As since it’s Google ‘s database it would be safe also.

#### **End User (Cell Phone or Desktop Application) :**

Now Its time to create an application for your phone/ desktop.

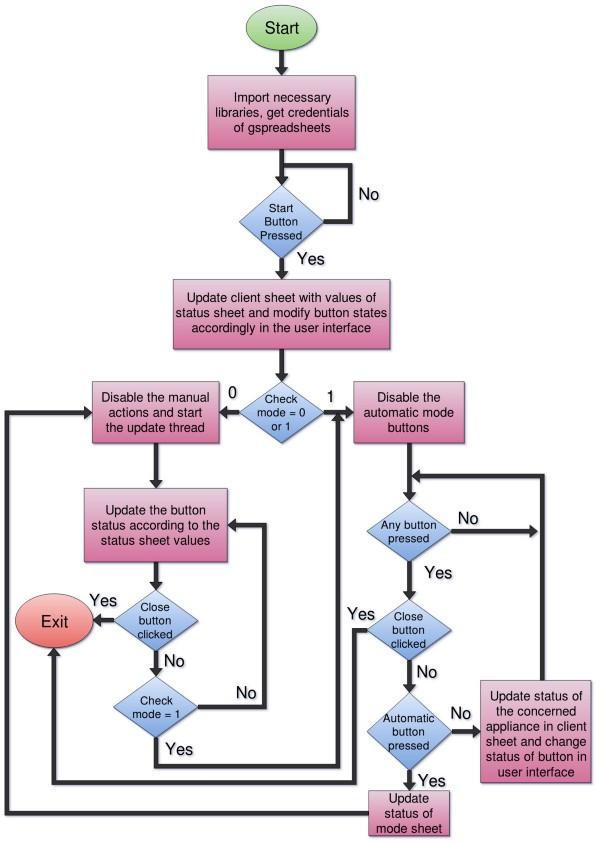
Language: Python, Kivy framework for UI design and build.

Requirement and instruction for the code to work properly are clearly specified in

Use “Buildozer tool” to convert desktop application made up in python to android application.

*[How to make .apk file from .py and .kv code using buildozer / python-for-android](https://kivy.org/docs/guide/packaging-android.html)*

What this code is gonna do ??



Basically, there are three Google spreadsheets named as “mode”, “client” and “status” have been created with working as follows :

1)Mode: this sheet will contain on which mode user wants to operate the appliances of home

Mode 0 i.e. Automatic Mode: User will tell raspberry pi to control the home as per data from sensors.

Mode 1 i.e. Manual Mode: User will control according to his/her customized needs.

2) Client : If in manual mode then the client will click on particular lights/fans icon to make them on/off as per his/her choice. And that information of which appliances need to be on/off will set into “Client” sheet on the cloud. Client (User who is using the app) will have the power to modify this sheet only. Now Raspberry Pi end will fetch this “Client” sheet and can read from there what it has to do and control the appliances accordingly.

3) Status : If in Automatic mode then User at-least should get to know what is the status of appliances at home :P. Which devices are on and which are off. Hence Raspberry Pi end will update values/status in “status” sheet and user end will access that sheet and in the application, it will show the status accordingly. Raspberry Pi will get data from sensors:- Temperature sensor, PIR sensor, the Light sensor (Digital) . and according to that data, the algorithm will find out which devices should be in which state, without human interaction. i.e. Appliances will get controlled automatically.

SIMPLE ! and after that this is how it will look and work like. Classy ;)

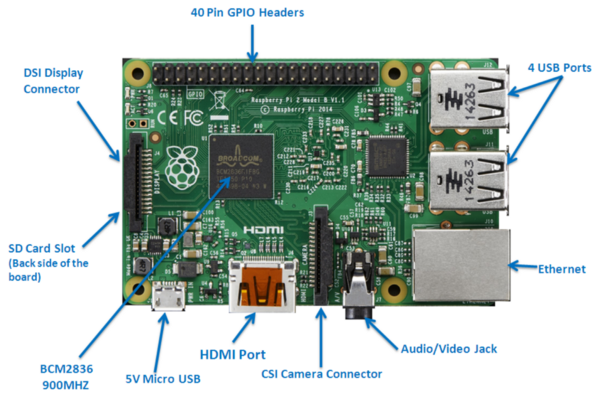
**This is how app Looks like .**

#### **Raspberry Pi End (*Home appliances*)**

Okay now its time to play with Raspberry Pi(Yummy XD), Sensors, Relay, Wires.

*[What Raspberry Pi is??](https://www.raspberrypi.org/help/what-%20is-a-raspberry-pi/)* **Raspberry Pi**, a credit card-sized, low-cost but fully functional and programmable computer with modern high-definition multimedia capabilities. It may be the device that gets us back to computing basics.

*How Raspberry Pi works ?*

**

The Raspberry Pi device looks like a [motherboard](https://computer.howstuffworks.com/motherboard.htm), with the mounted chips and ports exposed .

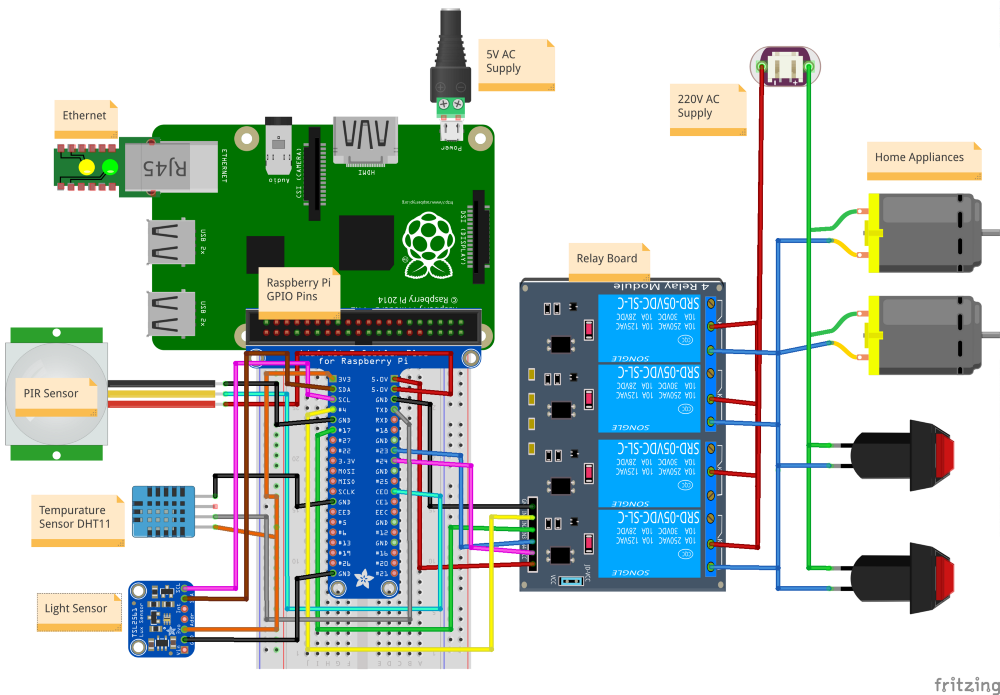
Here are the various components on the Raspberry Pi board and they literally explaining how Raspberry Works :

* **ARM CPU/GPU** — This is a Broad-com BCM2836 System on a Chip (SoC) that’s made up of an ARM central processing unit (CPU) and a Videocore 4 graphics processing unit (GPU). The CPU handles all the computations that make a computer work (taking input, doing calculations and producing output), and the GPU handles graphics output.
* **GPIO** — These are exposed general-purpose input/output connection points that will allow the real hardware hobbyists the opportunity to tinker.
* **RCA** — An RCA jack allows connection of analog TVs and other similar output devices.
* **Audio out** — This is a standard 3.55-millimeter jack for connection of audio output devices such as headphones or speakers. There is no audio in.
* **USB** — This is a common connection port for peripheral devices of all types (including your mouse and keyboard). Model A has one, and Model B has two. You can use a USB hub to expand the number of ports or plug your mouse into your keyboard if it has its own USB port.
* **HDMI** — This connector allows you to hook up a high-definition television or other compatible device using an HDMI cable.
* **Power** — This is a 5v Micro USB power connector into which you can plug your compatible power supply.
* **SD cardslot** — This is a full-sized SD card slot. An SD card with an operating system (OS) installed is required for booting the device. They are available for purchase from the manufacturers, but you can also download an OS and save it to the card yourself if you have a Linux machine and the wherewithal.
* **Ethernet** — This connector allows for wired network access and is only available on the Model B.

*Lets Setup Raspberry Pi :*

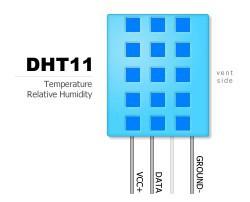
* First Of All boot a micro SD card with Raspbian OS(or any other as per compatibility) . [Here is the link!](https://www.raspberrypi.org/documentation/installation/installing-images/) , Now your SD card has OS inside it .
* PowerUp the Pi with 5V adapter .
* Connect Pi to desktop with HDMI cable . Now your Desktop will turn into Raspberry PC and we can access Raspbian OS .
* If you want to access Raspberry Pi from laptop then connect laptop and raspberry via Ethernet cable . Do ssh to access into Raspberry OS (Terminal) .

Now lets Connect Raspberry Pi to hardware (Home Devices) and after that we will write code in Raspberry OS to control Hardware Devices . Here is the connection Diagram :



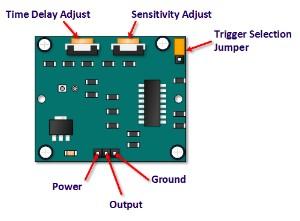
*Components Required :*

1. 4/8 channel relay board to control 220V from 5V (as GPIO pins will give o/p in 5V and then we need to control devices which usually runs on 220V .



2. Temperature Sensor : DHT11 , it will sense temperature of room . (we can’t use analog sensors as GPIO only accept digital i/p )





3. PIR sensor HC-Sr501 : To detect presence of human in room .

Sensitivity adjust :- Refers to from how far / with what sensitivity , sensor will detect presence of human. Rotate knob left (anti-clockwise) to decrease and reverse to increase .

Time delay adjust : after how much time sensor will again start detection . Adjustment same as sensitivity .



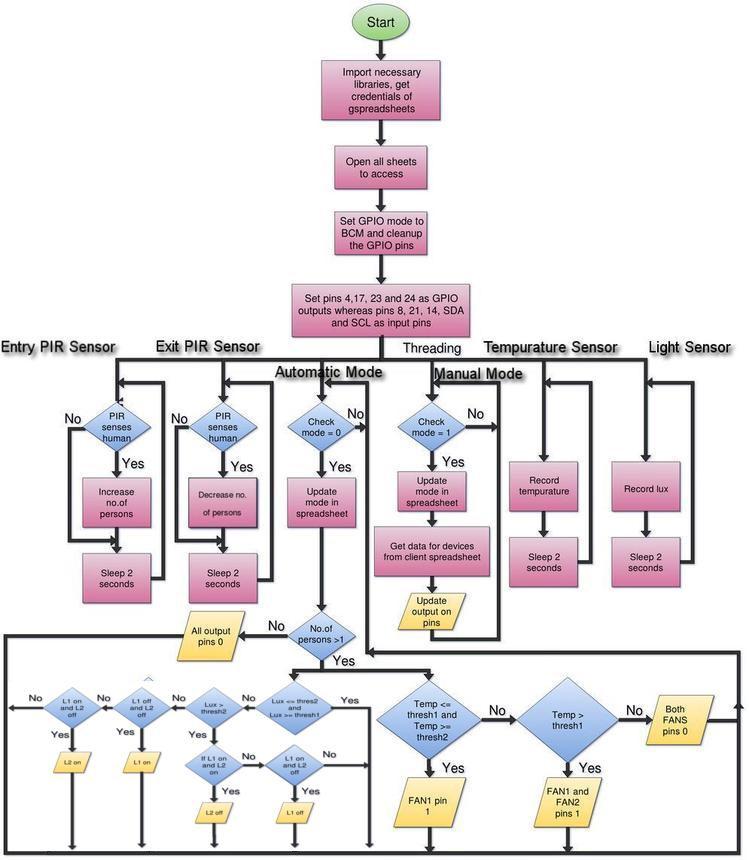
4. TSL2561 Lux Sensor : To measure intensity of light present in the room .

we can use other sensors also as per needs .

Now connect everything with raspberry pi according to connection diagram .

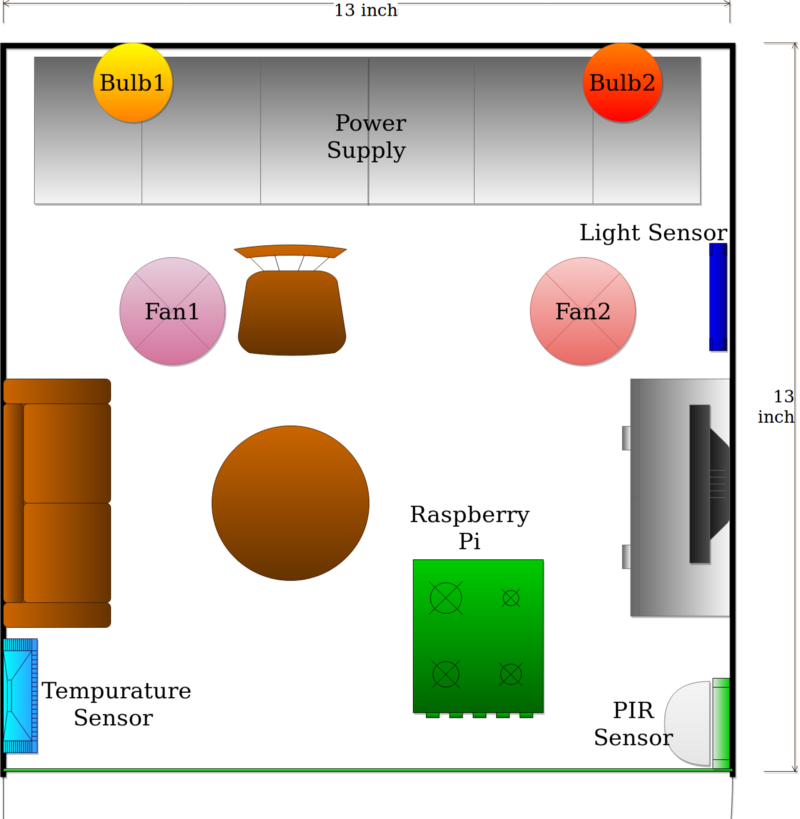
Okay, open raspberry terminal ans start writing code to communicate between sensors , appliances and end user .

What code will actually do :



In manual mode, raspberry Pi gives o/p to GPIO pins according to the Client requests. In automatic mode sensors data has been used by the algorithm to make decisions.

Some glimpses of the working project :



**Model Design For Home Automation**