# HOME AUTOMATION SYSTEM PROJECT IOT - CS 437

## Team Mates:

Below are the names and net id for each member of our teammates.

| Name | netid |
|---|---|
| Sahib Singh | sahibs2 |
| Saniya Wanhar | swanhar2 |
| Shashivrat Pandey | spandey6 |

Video Link: https://drive.google.com/file/d/1ARjkw817jib4INf-I7Oab7RYCCiVqiBN/view?usp=sharing

Github Link: https://github.com/sdsahib/IOTFinalProject

## Motivation:

Home Automation simply means making your Home Smart .
In actual sense it becomes smarter only when IoT ,i.e. Internet of Things comes into the role .We have countless benefits as we can Save our energy and connect “n” number of appliances with its help.
With help of IoT, everyone can have access to their home appliances through google clouds as it helps in passing the information .Technology is changing and today’s world demands “Smart Living” , using IoT it can be turned into reality .Smart Homes results into a Smart City & a Smart City results into a Smart world

The motivation for developing smart home systems comes from many reasons, but most prominent are convenience, security, energy management, connectivity and luxury. Smart Home systems are one of the newer areas of research that have not been fully integrated into our society. This is because the research requires many other disciplines of research and engineering to produce a functional smart home. The cost of installing a smart home is also a large hindrance to the emergence of smart home systems into the market. The extra

cost of the install is from the fact that even though a majority of homes were built in the near past, technology has been growing exponentially. This means that most homes were built before this technology was available, and this creates a barrier for the development and sales of smart home systems. However the technology is becoming better and cheaper, and this will help to make smart home systems an expense worth having when new homes are being built.
The biggest motivation behind smart home systems is the convenience. Convenience is really another way of saying “time saver”, and into day’s world where everything is moving faster, every second has value. Most of the technology we use today is based of convenience, for example cars get us where we need to go faster, phones get us information from other people faster, and computer’s get work done faster. Smaller conveniences in the home will be desirable because they allow the home to save the user time as well. There are already many convenient technologies in the home like the dishwasher, washing machine, and microwave ovens.

## Technical Approach:

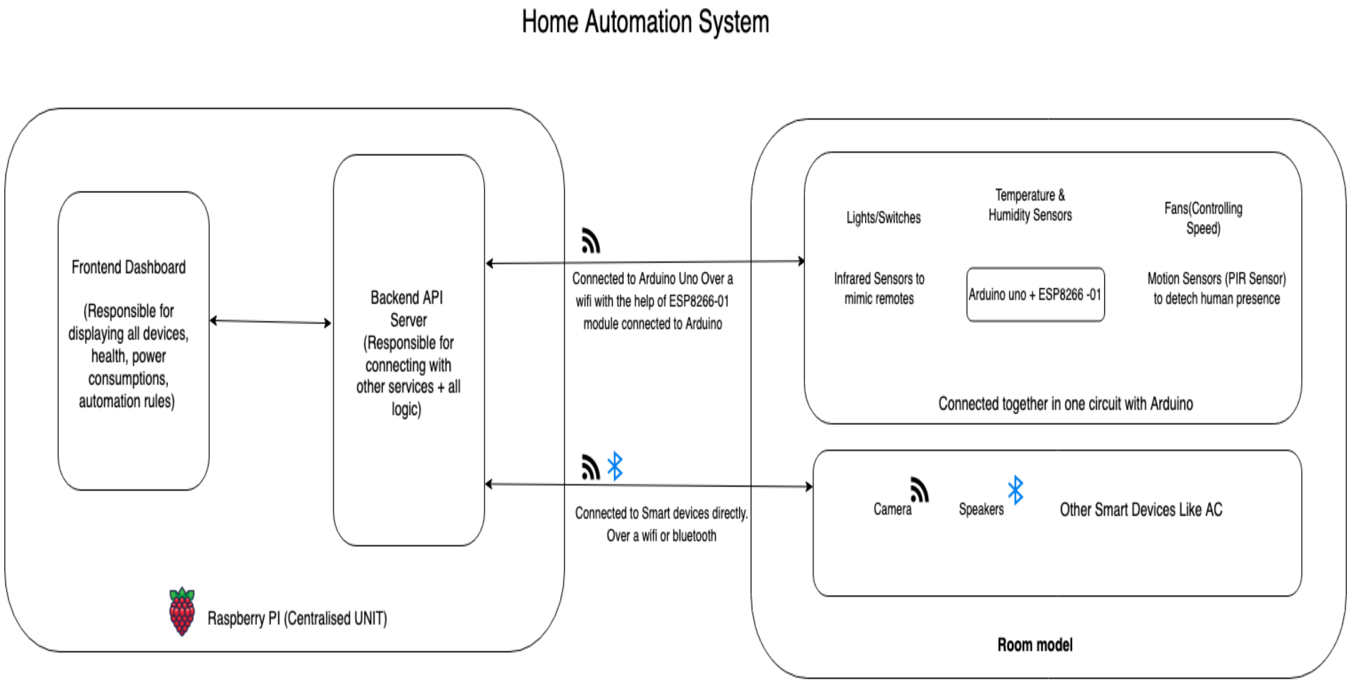

Home Automation System
Frontend Dashboard
(Responsible for displaying all devices, health, power consumptions, automation rules)
Backend API Server
(Responsible for connecting with other services + all logic)
Raspberry PI (Centralised UNIT)
Connected to Arduino Uno Over a wifi with the help of ESP8266-01 module connected to Arduino
Connected to Smart devices directly. Over a wifi or bluetooth
Lights/Switches
Temperature & Humidity Sensors
Fans(Controlling Speed)
Infrared Sensors to mimic remotes
Arduino uno + ESP8266 -01
Motion Sensors (PIR Sensor) to detech human presence
Connected together in one circuit with Arduino
Camera
Speakers
Other Smart Devices Like AC
Room model

Figure 1

Above diagram shows our project overview, which we plan to do.
We had our Raspberry PI as our Main server, and then Arduino Uno as a controlling unit per room.

We designed this architecture keeping in mind that if we need to convert normal home to smart home, we will just add our Arduino module(shippable product) per room and then one raspberry pi(zero) to act as the main processor. Which will be both cost efficient and easy to use DIY for the customer if they want to do it by them self.

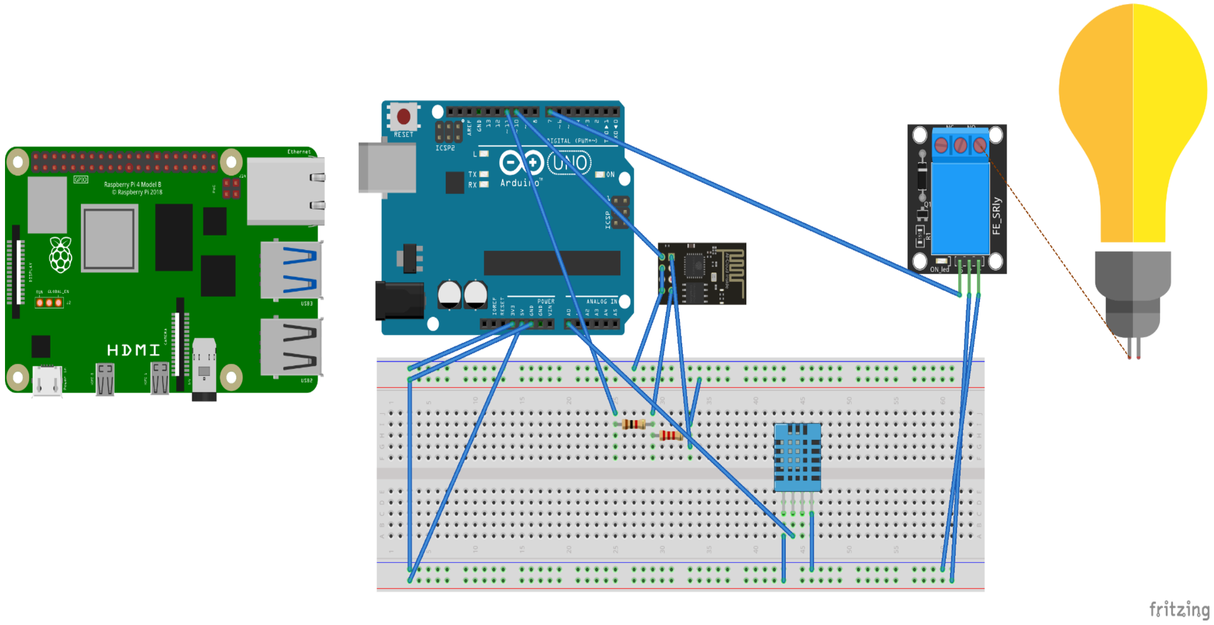
- Steps they need to follow will be attach all the connection wires to the relay of arduino PCB ( In first version we were planning to support up to 10 relays).
- Connect the Raspberry pi to the home wifi, and after the first time intiallization, which will be configuring room -> and relay pins in the raspberry PI.
- Voila you home is now a smart home.

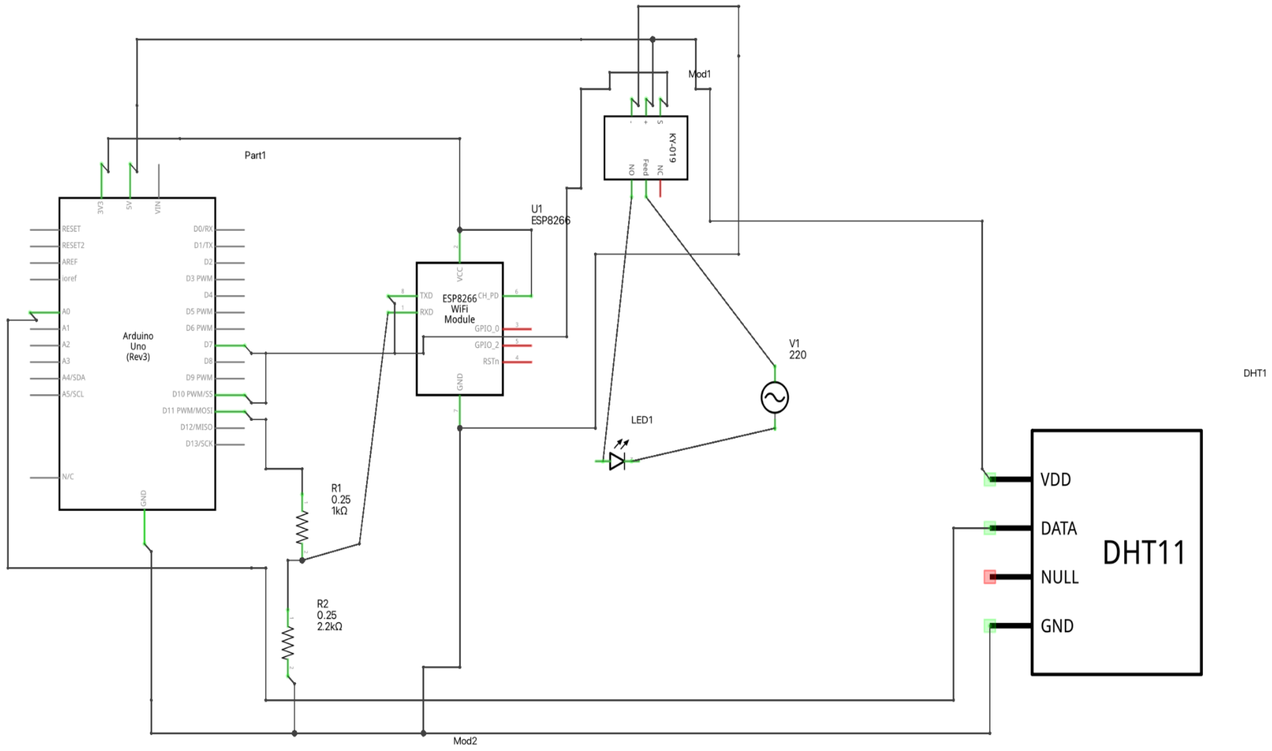
**Raspberry PI:**
We planned to host backend & frontend Service in our Raspberry PI.
Backend is Node.Js express App, so that we can design an control using the REST API's for UI & socket connections for regular interval data from sensors.
It contains all the logic for connecting to the Arduino Uno + handling the incoming data and processing it into meaning ful detail, preparing data for Frontend Data visualization, API's for creating the automation(certain task to happen automatically, in our case, may be turn off all the appliances for a particular room, or change the light color or play a particular song).
Backend has capability to connect with Arduibo Uno over a Wifi or to work separately with smart appliance like color changing light, bluetooth speakers or wifi cameras.

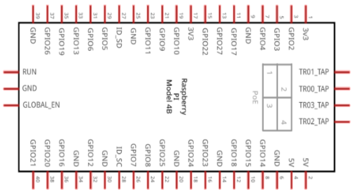
**Arduino Uno + ESP 8266 -01:**
We used Arduino-Uno to act as our connector/CPU unit for a particular room.
So that we can connect

- smart switches,( or Relays to existing switches),
- sensors like PIR( for detecting motion which detects on the basis of heat emitted by the body),
- temperature or humidity sensors like DHT11 or 22.
- Flame detection sensors
- Infrared (IR) light sensors to control devices.

We also connected the Arduino with ESP8266 esp-01 module, to provide wifi functionality, so that it can receive input commands and share the data collected to Raspberry PI server.







The above diagram shows the details about the arduino and it's connected component. In this POC (proof of concept) we have an Arduino Uno connected to

ESP8266 - 01 module, and then relay is connected to bulb, which is connected to the AC power supply. And the signal pin of Relay is connected to Arduino digital Pin 7.
We also have one DHT11 for collecting the temperature and humidity information.

**Relay Flow:**

| **ESP8266 Pin** | **Arduino pin** | **ESP8266 Pin** | **Arduino pin** |
|---|---|---|---|
| TX | 10 | GND | GND |
| CH_PD | 3.3V | GPIO1 | - |
| RESET | - | GPIO2 | - |
| VCC | 3.3V | RX | 11 |

| **Relay module** | **Arduino Uno** |
|---|---|
| Signal pin | Pin 7 |
| GND | GND |
| Vcc | 5V |
| AC(Normal Open) | Positive terminal of bulb |
| AC(Feed) | Negative or Earth(AC) |
| Bulb negative | AC positive |

We used bulb to showcase our POC, we connected the positive end of bulb to the NO pin of the relay and negative to the AC and then FEED pin of the relay to AC. (refer the above diagram).

1. Raspberry Pi send the signal to ESP8266-01 module, which ESP8266-01 passes serially to the arduino uno board.
   Signal example: using TCP-> socket data packet contains "**<bulb,on>**" or "**<bulb,off>**".
   Currently in this POC we have hardcoded the command names, in future release we are planning to make it customizable from raspberry PI itself.
2. Then arduino process the <bulb,on> or off command, it checks it's internal Relay struct array to fetch the pin allocated to that.
3. And accordingly it digitalWrite(HIGH) or LOW to that pin.

**DHT11 Flow:**

- DHT 11 is connected to the Analog pin A0, Vcc pin is connect to 5V output from arduino and GND is connected to GND
- We used DHT11.h library[1], and then created a DHT struct. Then using the read11 function provided in the library. We fetched both temperature and humidity.

Arduino acts as our main controller per room and main two function are to receive the command from the Raspberry pi server to the action and, to send the sensors data to raspberry pi.

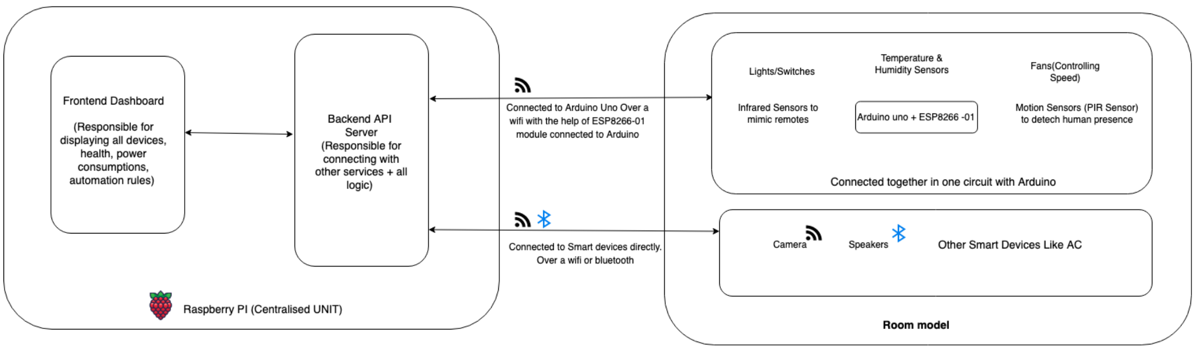
## Implementation Details:

Lets now talk about implementation steps we followed.

**Step 1:**

We decided on the architecture of out project. To keep one unit(arduino uno) for controlling sensors and one main unit(Raspberry pi) to act as the main server. Since each rooms will have one arduino unit we added the wifi functionality to our arduino uno, such that it can communicate with the main server and act accordingly.

Home Automation System


Frontend Dashboard
(Responsible for displaying all devices, health, power consumptions, automation rules)
Backend API Server (Responsible for connecting with other services + all logic)
Raspberry PI (Centralised UNIT)
Connected to Arduino Uno Over a wifi with the help of ESP8266-01 module connected to Arduino
Connected to Smart devices directly. Over a wifi or bluetooth
Lights/Switches
Temperature & Humidity Sensors
Fans(Controlling Speed)
Infrared Sensors to mimic remotes
Arduino uno + ESP8266 -01
Motion Sensors (PIR Sensor) to detech human presence
Connected together in one circuit with Arduino
Camera
Speakers
Other Smart Devices Like AC
Room model

We have explained the basis about this above in the technical approach sensor.

**Step 2:**

We connected arduino uno to the ESP8266 -01 this part was very time consuming of our whole project.

It took lot of time as there was no proper library available to use with the module, plus all the reference only talk about data sending from ESP-01 to outside world, We spent give a time to figure out on how to send data to the ESP-01.

Then we tried REST approach to connec to the module but it was time consuming because of the extra headers, then we tried TCP socket approach to send the data. It was faster then the previous one and we used it to send the info to ESP-01.

We added “<” to our messages to indicate the start of our message and “>” to indicate the end of our message.
Then since we are allowing upto 10 relays we created the struct Relay which has pin, and name. We use name as the command.
So lets say we have 2 relay connected,
It will be like {1, “switch”},{2,”bulb”}.

So to turn on the bulb, Raspberry pi will send the command “<bulb,on>”. Which will be passed to the arduino and then arduino will search the above array of structure, and find the pin associated with it.
So in our case it’s 2, so arduino will digitalWrite(HIGH) to the pin, which inturn is connected to the relay and will turn on the relay and bulb circuit will complete and hence it will turn on.

**Step 3:**
Connecting relays to the Arduino uno, we connected the signal pin to the arduino pin according to the Relay configuration defined in the arduino -uno.
First time Raspberry pi while configuring uses the init call which will setup the configurations into the Arduino. Input data will consists of pair of pin numbers and relay identifier.
Ex: <init, <2,bulb>,<3,switch>>

And Arduino will process it and it will store the configuration in struct of relays which currently act as a hashmap of pin number and commond.

And when the Raspberry pi will send the command it checks this relay and fetches the pin number and perform the action specified in the command.

## Results:

This project and Intro IOT world have open a new world of opportunities for us. The solution we can create by combining CS+ electronics and mechanical are limitless. We can automate our daily life process like watering the garden, turning on the coffee machine, playing music etc.

After so many across the time zone meeting discussion meeting we were able to do our final project. Although it’s not even a MVP now, but soon will be. We learned to work on various sensors and to read the design manual for them to find the correct pins layouts, input voltage and what not. We learned how to debug faulty sensors, we learned how to flash ESP8286 -01 and reinstall it’s firmware which was fun and tiring.

Our POC for the first step “To convert normal appliance to smart appliance”(in sense you can control via phone, which turns on/off automatically sensing motions) which will save on power consumptions and in turn save cost. We were able to track temperature and humidity.

Our main goal was to make the product from scratch and not by using the customized component available in the market so that we can gain the raw knowledge and save on component cost as we are planning to make it safe and economical so that every house hold and afford it and have their dream smart home.

We were able to add the wifi to out arduino uno board and that was tough part. (we tried various ways of communication like REST and others) and compare them and finally we used normal TCP socket commands which proved out to be the fastest way.

We were able to detect the temperature and humidity and send the data to raspberry pi.

## **Next Steps:**

Our next target is to continue this project and create a markettable product and sell it on Amazon.

For that, we have certain action item for our arduino module:

1. To attach more sensors to the arduino like flame detector, smoke detector and various gas detector (like CNG or LPG).
2. Then to create the PCB for the arduino module and attach atleast 10 relays, and sensor mentioned above.
3. To create the 3D box design to fit in our PCB.

For Raspberry pi:

1. To find a way to connect the smart devies directly to our App. like camera, AC and other devices.
2. Create the UI for the Raspberry pi (Dashboard).
3. To attach a Screen to the Raspberry pi so that it can display the Dashboard for power consumtion and other functionality to control the appliances.
4. To create the PCB for the module.
5. To create the 3D box design.

## REFERENECES

1. https://github.com/adafruit/DHT-sensor-library
2. https://fritzing.org/
3. https://curtocircuito.com.br/datasheet/arduino_sensor_shield.pdf
4. https://docs.espressif.com/projects/esp-at/en/latest/esp32/AT_Command_Set/
5. https://www.microchip.ua/wireless/esp01.pdf