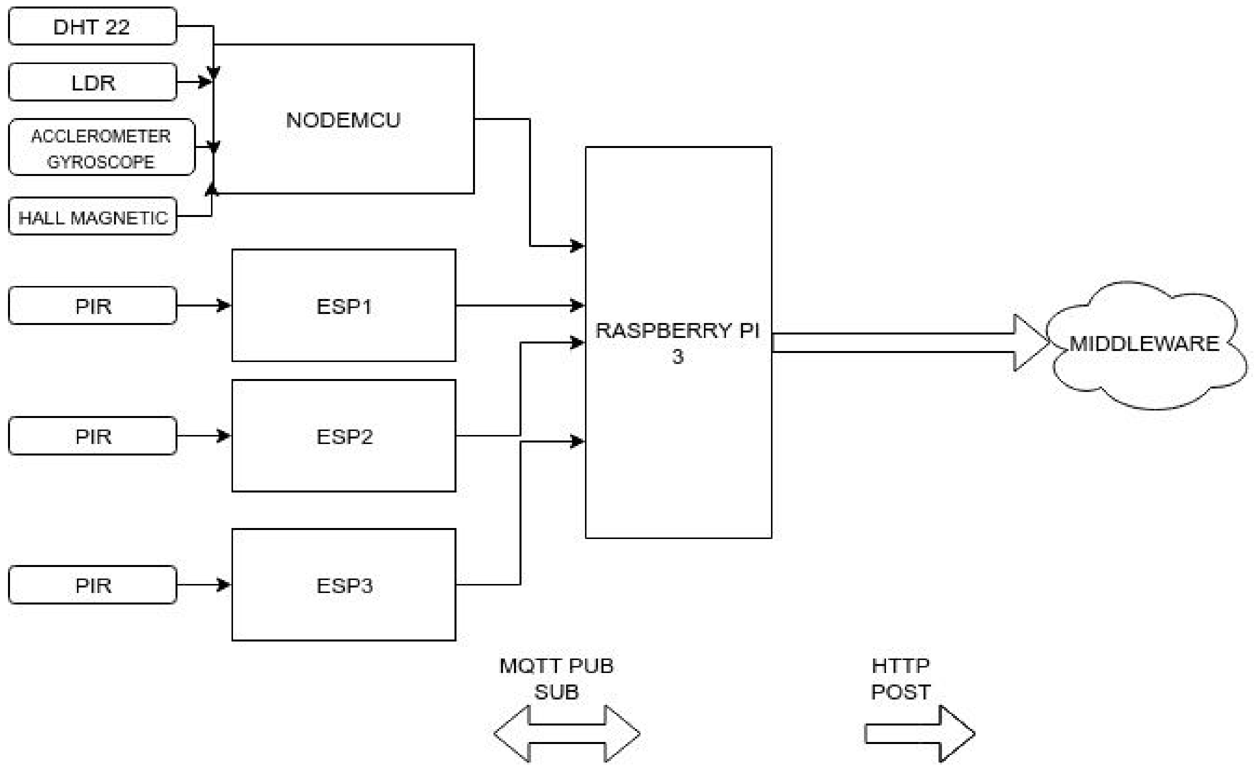
RASPBERRY PI INTERFACED WITH NODEMCU



INTRODUCTION:

STEP 1:
Raspberry pi as access point:

a.

```
sudo apt-get update
sudo apt-get upgrade
```

```
B.
```

to install hostapd and dnsmasq.

```
sudo apt-get install hostapd
sudo apt-get install dnsmasq
```

```
C.
```

modify the dhpcd.conf file by running the following command on our Raspberry Pi.

```
sudo nano /etc/dhcpcd.conf

Add this to the bottom of the file:
interface wlan0
static ip_address=192.168.220.1/24
static routers=192.168.220.0
```

restart our dhcpd service so it will load in all our configuration:
sudo service dhcpcd start

D.

```
sudo nano /etc/hostapd/hostapd.conf



EDIT LAST TWO LINES:

interface=wlan0
driver=nl80211

hw_mode=g
channel=6
ieee80211n=1
wmm_enabled=1
ht_capab=[HT40][SHORT-GI-20][DSSS_CCK-40]
macaddr_acl=0
ignore_broadcast_ssid=0

# Use WPA2
auth_algs=1
wpa=2
wpa_key_mgmt=WPA-PSK
rsn_pairwise=CCMP

# This is the name of the network
ssid=Pi3-AP
# The network passphrase
```

```
wpa_passphrase=raspberry

E.
sudo nano /etc/default/hostapd
```
find the following line and replace it:
Find:

```
#DAEMON_CONF=""
```
Replace with:

```
DAEMON_CONF="/etc/hostapd/hostapd.conf"

F.
sudo nano /etc/init.d/hostapd
```
find the following line and replace it:
Find:

```
DAEMON_CONF=
```
Replace with:

```
DAEMON_CONF=/etc/hostapd/hostapd.conf

G.
sudo mv /etc/dnsmasq.conf /etc/dnsmasq.conf.orig

sudo nano /etc/dnsmasq.conf
```
To this file add the following lines, these lines basically tell the dnsmasq service how to handle all the connections coming through.

```
interface=wlan0       # Use interface wlan0
listen-address=192.168.220.1   # Specify the address to listen
on
bind-interfaces      # Bind to the interface
server=8.8.8.8       # Use Google DNS
domain-needed        # Don't forward short names
bogus-priv           # Drop the non-routed address spaces.
dhcp-range=192.168.220.50,192.168.220.150,12h # IP range and
lease time

H.
```

configure your Raspberry Pi so that it will forward all traffic from our wlan0 connection over to our ethernet connection

```
sudo nano /etc/sysctl.conf
```

Within this file you need to find the following line, and remove the # from the beginning of it.
Find:

```
#net.ipv4.ip_forward=1
```

Replace with:

```
net.ipv4.ip_forward=1
```

run the following command to activate it immediately:

```
sudo sh -c "echo 1 > /proc/sys/net/ipv4/ip_forward"
```

I.
 the following commands to add our new rules to the iptable:

```
sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
sudo iptables -A FORWARD -i eth0 -o wlan0 -m state --state
RELATED,ESTABLISHED -j ACCEPT
sudo iptables -A FORWARD -i wlan0 -o eth0 -j ACCEPT
```

To save our new set of rules run the following command:

```
sudo sh -c "iptables-save > /etc/iptables.ipv4.nat"
```

J.

```
sudo nano /etc/rc.local
```

Find:

```
exit 0
```

Add above “exit 0”:

```
iptables-restore < /etc/iptables.ipv4.nat
```

```
K.
sudo service hostapd start
sudo service dnsmasq start
sudo reboot
```

Step 2:
Mqtt publish data from the three pir esp’s and the one node mcu:
NODEMCU 1:
This NODEMCU module is interfaced with three sensors, Accelerometer & Gyroscope GY 521, Hall Magnetic Sensor, DHT 22, LDR. The purpose of this module is to publish all sensor data to the Raspberry Pi which hosts it's mqtt broker and subscribes to this topic and publishes it to the middleware .

NODEMCU 2/3/4:
Each of these modules are interfaced with a pir sensor. This pir sensor is triggered every time there is a motion. If you want to increase the sensitivity, decrease the delay. The pir publishes to the Raspberry Pi every time it is triggered. If there isn’t motion continuously for more than 5 minutes then it publishes saying no motion detected.

Step 3:
Raspberry pi: set up mqtt broker:

Mosquitto

```
cd ~
wget http://repo.mosquitto.org/debian/mosquitto-repo.gpg.key
sudo apt-key add mosquitto-repo.gpg.key
cd /etc/apt/sources.list.d/
sudo wget http://repo.mosquitto.org/debian/mosquitto-stretch.list
sudo apt-get update

cd ~
sudo wget
http://security.debian.org/debian-security/pool/updates/main/o/openssl/libssl1.0.0_1.
0.1t-1+deb8u8_armhf.deb
sudo dpkg -i libssl1.0.0_1.0.1t-1+deb8u8_armhf.deb
sudo wget
http://ftp.nz.debian.org/debian/pool/main/libw/libwebsockets/libwebsockets3_1.2.2-1
_armhf.deb
```

```
sudo dpkg -i libwebsockets3_1.2.2-1_armhf.deb
sudo apt-get install mosquitto mosquitto-clients
```

Test:

SUBSCRIBE:

```
$ mosquitto_sub -d -u Raspberry_Pi -P Raspberry -t test
```

PUBLISH

```
$ mosquitto_sub -d -u Raspberry_Pi -P Raspberry -t test -m “ello”
```

Setting up mqtt broker on Pi:

```
$ sudo nano /etc/mosquitto/mosquitto.conf
```

Remove *include_dir /etc/mosquitto/conf.d*

And add:

```
allow_anonymous false

password_file /etc/mosquitto/pwfile

listener 1883
```

```
$ sudo mosquitto_passwd -c /etc/mosquitto/pwfile username
```

Step 4:
Python code on raspberry pi to be able to subscribe to the NODEMCU topics (ESP2/ESP3/ESP4/ESP1) and make a http post to the middleware

Step 5:
Registration of each device:

1. Register with

```
curl -k -i -X GET
"https://smartcity.rbccps.org/api/0.1.0/register" -H
'apikey: <provider_api_key>' -H 'resourceID:
<Your-Resource-ID>' -H 'serviceType:
publish,subscribe,historicData'
```

Provider API key - `beee69bb9d024fbf97800be726f85a57`
Resource ID - Unique to the device, say MAC address
'histroicData' to be included if provided for in the schema

2. API key is granted by the above command which is it to be noted

3. Schema is to be validated by the validator
   `python jsonschema_validator.py <schema.json>`

   Find the validator at
   https://github.com/rbccps-iisc/smart_cities_schemas/tree/v0.1.0

4. Change directory to the folder that has the onboarding scripts

5. Run
   ```
   ./onboard.sh <Your-Resource-ID> <schema.json>
   <provider_api_key>
   ```

6. The above generates a tmp.sh which is to be executed
   `./tmp.sh`
   The curl command in tmp.sh may need -k attribute to skip certificate check

7. Check if onboarded at https://smartcity.rbccps.org/api/0.1.0/cat/

8. Entries by the registered device into the database can be viewed by

```
curl -XGET -i -k 'https://smartcity.rbccps.org/api/0.1.0/historicData?pretty=true&size=10' -H
'apikey: beee69bb9d024fbf97800be726f85a57' -H 'Content-Type: application/json' -d '{
"query": { "match": { "key": "<Resource-ID>" } } }'
```

9. Subscription by ThingsBoard

```
curl -ik -X GET http://10.156.14.6:8989/queue?name=thingsboard_application_2 -H 'apikey:
1e8282f04f2f4590975876912e5dd338'
```