Home Automation with Smart Electric Power Supply using IOT

Deepa B G
Assistant Professor
School of CSA
REVA University
deepabg03@gmail.com

Dr. Senthil S
Professor
School of CSA
REVA University
senthil.s@reva.edu.in

KabeerAhamed P.
MCA Student
School of CSA
REVA University
kabeer.clk@gmail.com

ABSTRACT

In the 21st century, where the automation is acquiring every platform and every place with reducing lots of human efforts, all by replacing the human hands into the machinery, including the industry, the automation is also occupying the personal and economic life which in simple can call as the home automation, smart cities and much more.

The Smart Electric Power Supply is project using IOT technology for the smart monitoring, tracking, analysis and prediction of the flow of electricity in the city, this also useful for the easy detection of the electricity loss like electricity theft and illegal connections and this also gives the information of the status of the devices, their conditions and also the regular usage and smart electricity monitoring of individual building, area, city and others as we maintain the electricity.

This including the smart electricity supply system this also provide the built-in support to the smart automation of the devices with auto programmable solution as the open source, this makes the machine to dynamically update and operate the devices with all the facilities which are required for the users.

This technology is built on the IOT technology, specifically in the Raspberry Pi board, and also built on the REST API technology.

Keywords:
Internet, automation, layer, process, rest api, crontab, storage, devices, Home User, Centralized Authority.

Introduction:

The IOT (Internet of Things) is the interconnection of different electronic equipment on which the devices are interconnected in the new intelligent form [1]. The IOT technology arise during the revolution in the fields of the robotics and artificial intelligence, this has merged the sensor technology with the computation making the embedded systems, the invention of Arduino and Raspberry Pi given a new face for the development of the IOT, hence IOT is also known as the playing with the sensors, actuators and other machinery elements.

The IOT is the combination of Hardware, software and the mechanics, hence this is the best technology for the automation ranging from small bulb to driverless cars in the world, which can cause a huge change with the modern implementation of technologies.

Raspberry Pi:

The Raspberry Pi is the single chip computer which is resembles to the size of a credit card. This is the most powerful tool to automate the devices with compact space. This is very simple to use and implement. The raspberry pi use the simple ARM processors which can provide the speed up to 700MHz, which is sufficient for any of the devices to implement and run any kinds of electronics by using sensors and actuators.

It also provides the GPIO ports which enables the developer to automate any devices using the actuators and can access any kind of analog or digital information as input and it can pass the information in same manner.

This works only for the power supply of 5 volts to 02 volts of inputs (depending on connected devices and performance).

REST API:

The REST API is the shortened form for Representational State Transfer, this defines the many of the rules and constraints for defining the web services, the one of the major and important feature of the REST API is, this can be developed in any platforms and can be processed anywhere and this is one of most secured way for the web services and very easy to use and maintain. The REST API gives many of the features to insert, retrieve, modify the data in an easy manner, hence this is the most useful and easy to learn and understand.

Python:

The Python is the best dynamic programming language which is used for automation in the enterprises. The python is the dynamic and easy to learn and understand. It provides the best way to automate the things up. This provide the English like syntax for the script which makes nay one to understand easily and use easily. Now the python is the most needed programming language for any kind of automation.

Literature Survey:

In the current technology, most of the Home automation is done using Mobile, via internet or the Bluetooth internet [1]. In which the programs and

instruction clearly has been defined the usage and this is static and can only upgrade the features, but we cannot upgrade the technology, and if any one access the mobile then this cause the simple misuse of the devices. They can work like as shown below:

Fig – 1: The existing home automation structure.

As shown in above diagram, all the mobile controlled devices are only dependent on a mobile using a Bluetooth or internet and they are just the replacement of the mechanical switching system into the mobile switching system.

The IOT has also upgraded its place in the field of direct internet connectivity [2]. The embedded system is connected to internet directly through the IP address in Internet. They are also working as same manner, just replacing the mechanical switching system by the mobile switching system, they also won't give any kind of centralized operation and they also lack the proper update.

All the above process is only can be considered as the remote switching system and they cannot produce the security against the electricity theft, misuse and much more and they also cannot work without the internet, hence they need to be upgraded with the self-learning process even when they are not connected to the internet. All these technologies are can only limited to a home or building. In this research, we are going to develop a centralized connectivity with the complete control and multiway update system with centralized monitoring with all the security against the electricity problems like electricity theft against the homes, industries and much more.

Implementation:

The Smart power supply is the one of the IOT implementation which consists of the five layers: Application layer, Network Layer, Process Layer, OS layer and finally the hardware layer as shown in the fig-2. In the figure, the five layers has their own unique functionality and working principle.

The very first is the application layer: The application layer can be the phone app, internet controlling or any kind of artificial intelligence bots. The work of the application layer is to accept the requests or commands from the users, process it and analyze whether they are the operational commands or the developmental commands and then redirect it into the perfect places. The application layer is basically built upon the REST API's. The network layer just encrypts and sends the information commands through the REST API's network. The process layer divides the working and performance into the two types: client-side type (Home User Type) and Server Type (Centralized Authority) and process them according to it and it also do another work like maintaining the history data for learning and much more. The OS layer and the hardware layer just perform work as a computer which will be the platform to run every command and much more.

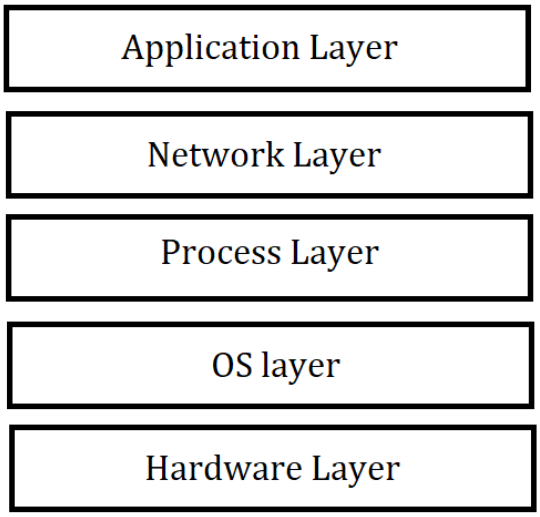

Fig – 2: The Layers of HWSEPS.

Application Layer:

The application layer is the dynamic layer of this project, where the application can be decided by the user based on which type the user needs to interact with the devices and this is developed based on REST API's. Hence the w can develop applications like web apps, android applications, System applications, Automation tools, Remote access tools and much more, hence the application layer is based on the specific user and also based on the specific interaction which user wishes. The structure for the *Application Layer* is shown in fig-3, in this figure, we can see that the system can access the information using the REST API's hence this can run on any of the platforms.

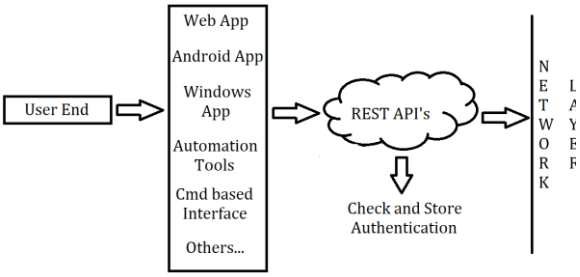

Fig-3: Application Layer Structure.

In this application layer, during the processing of rest API, it checks whether the client machine is authorized or not, by checking the IP which the machine is logged in, this recognizes and then processes the request, if not the person have to authenticate by logging in in the private Wi-Fi of the PI. This recognizes two types of the application requests: centralized authority, Home User. The centralized authority is nothing but the city electricity supplier. The centralized authority will have the authentication to access the electric usage log file which represents the duration of usage, quantity of usage and the time of usage. The Home user will have complete control over the machine and the home user even have the

capability to update the mechanism according to his convenience using REST API. The Home user can develop his own automation tools and he can upload it to the Machine using REST API.

The Home user will have the authorization and rights to modify the mode of control and everything which ever he needed. But the Centralized authorization can access the internal data of power consumption and the centralized authority can also has the capability to check the internal modifications of the user which is defined for the interface change.

Network Layer:

The Network layer here comes with the different perspective. All the client devices should relate to the GSM which in turn provide the internet facility to the machine and this helps the device directly connected to the internet which can cause the secure connection by providing cryptography. The structure for the network layer is defined as shown below:

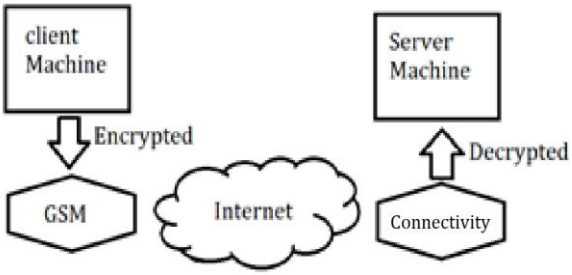

Fig-4: Network Layer Structure.

Consider the above diagram, the client machines are connected with the GSM module which in turn provide the direct connectivity to the destination database. Here the connected devices encrypt the data before sending of the information which in turn again decrypt at the stage of destination. The encrypted data is again encrypted basically in GSM. This can cause the direct connectivity and without the human intervention. The encrypted data can't be accessed to any of the client users and this directly moves into the server of the Centralized authority database and all these data are not storing in the client's database hence this step and process is more secure and stable for the data transmission and since this is stores inside of the server, damages of the machines inside of clients doesn't affect any of the process running or data in the centralized repository but the damage inside of the client causes to stop transmission of the client side application. For such process also the centralized server can consider by following algorithm:

1. If data transmission stops more than hour:
   a. Send authentication msg to that client.
   b. If response == False:
      i. Invoke alert ticket to controller.

Algorithm-1: Info Transmission Exception

In this way we can consider and detect the quick damage and can manage efficiently. This is the secure point of the information transformation between two small devices. Due to its regular checkup in crontab and regular mode of transmission makes the devices to get alert their masters when the device stops working more than an hour. Even this is the simplest, the regular check up makes the device more stable and secure.

Process Layer:

The Process layer is the layer which connects all the above layers (Application layer and Network layers). Since every layer performs its own processes but the actual main processes are done in the process layers. The above layers like Network and applications layers work processes like just as the interface and the basic functionalities. The process layer is the layer which bounds the all the above layers with the OS layer which is the platform for all those layers. The Process layer performs the functionalities like: Storing the status, which helps for the prediction of the future electricity consumption and much more, the data processing and the prediction upgrading and much more will be taken care by the process layer. The process layer consists of the crontab which is responsible for the continuous execution of the application or a program inside of an operating system. This is very simple and efficient process for functioning any form of data. Using crontab we are only adding or replacing the newly created functions instead of replacing complete software for any changes in the devices. The Process layer again helps for the creating the report for sending to the network layer for transmission.

The Process layer has separate running functionality for the interface management with the client side. This manages all the Input devices like sensors and the indicators whereas the Application layer manage the interface. The Process layer also helpful for taking care of the security issues of the device. In simple words, the Process layer is the simple and most secure layer among all the Layers. Which consists all the main software process activities which needed for the device. All the features are defined based on the crontab technology.

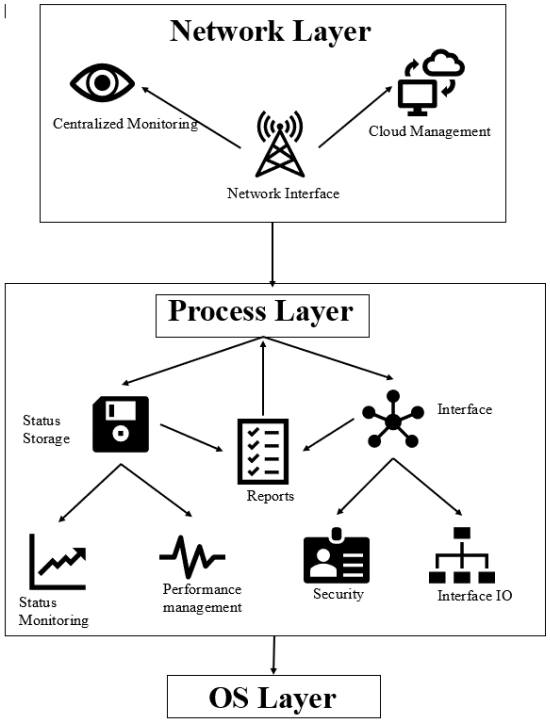

Fig-5: Process Layer Structure and Interaction.

Operating System Layer:

The Operating system layer is nothing but the operating system, here, we have used the Raspbian operating system which is very light and also very light weight operating system recognized for Raspberry Pi platform. The operating system provides the platform for the all the activities which are done above. The Raspbian is chosen for its simplified work and also the inbuilt functionalities. The ubuntu also the best option for defining any kind of the AI or IOT devices.

The Operating System here won't do any much functionalities rather than the basic activities. This just defined only to create the dynamic platform instead of just making automation.

Hardware Layer:

The hardware is the last and important layer of all the process because the hardware is the basic layer for all the layers. The main controller hardware is the Raspberry Pi. The Raspberry Pi is chosen because of its compactness, simplicity ease of use. The other factors also need to verify like the heat produced during continuous use, support to various of the sensors and output devices and much. The Pi cannot stand more than 180 minutes (3 hours) due to its heat production with the extreme processing, hence there is the need to implement the cooling system. Without the cooling system, the device may get damaged due to excessive heat, which is one of the main drawbacks for the device for implementation. However, the Raspberry pi B+ has made the device which is more compatible due to reduced heat production and less energy consumption processes.

The Sensors are directly connected to the Raspberry pi. If we need to implement more number of sensors than the limit of Arduino then we need to define the Arduino interface as the middleware which can act as the input device and using this can cause less pressure in the electricity and can make the device perform work more efficiently.

The sensors may be directly connected to the raspberry pi or through the Interface like Arduino, we need to define the interface via the program on what we are doing.

Algorithm:

Most of the working functionality in the algorithm part has been defined in the Process layer and the limited process are defined in other layers. The brief algorithm for the working of the machine with both client side and server side is showing as shown below:

Client Side:

1. Start.
2. Check for all hardware parts.
3. If hardware parts.found = defective:
   a. Report hardware issue critical.
   b. Limit the energy consumption.
4. start crontab with all running program.
5. for(inifnity):
   a. start reporting processes.
      i. Encrypt transmitting data.
      ii. Access acknowledgement.
      iii. If acknowledgement not found:
         A. Report network issue.
   b. Start user defined processes.
   c. Check for any new updates.
   d. If new programs are present:
      i. Upload program.
      ii. Start from process '3'.
   e. Track management and security processes.
6. Repeat from process 3.

Algorithm-2: Client Side/ Home User Side Algorithm.

This is the one of the simple algorithms which defines the overall work process of the device at client side.

In the very beginning, when the device starts, it will check all the hardware parts and also the hardware devices which is related to it. This technology will check the hardware in two steps: The first is: connection and disconnection status of the hardware devices can be accessed by the /dev/ directory where the Linux stores all the details or drivers of the devices, which guarantees that the device is connected. When we connect the devices then it shows the driver file which is connected to it. By checking that file name, we can easily predict whether the device is properly communicating with the central mother board or not. The simple example for the driver is shown as below.

Fig-6: Device driver and device file when a device is connected

The method of seeing and accessing from the /dev/ is very easiest thing which ever we can, and this is also the best option which won't slow down the performance of the machine, but we need to manually predefine which are all the devices we are using. One of the example codes is shown as below:

```
for file in os.path.listdir('/dev/'):
        if file == 'ttyACM0':
                Status = 'Active'
                return Status
return Decomissioned
```

Code -1: Device Checking

In the above source code, the function checks if the driver is present or not, if found immediately returns saying that the device is in active condition or else it returns saying that the device is in decommissioned, it means the processor not using the device for some reason, then this throws the exception to the user saying that this particular device is not accessible and we need to troubleshoot it. Sends with the exception signal.

The Second method for checking the device status is to send the instruction to the device and access the information from the input devices and compares the data with the previous records. If the result found in the record not matching the data and also the device giving the error data, then it throws the critical exception defining that we need to replace that particular device immediately:

```
Temp_sensor = measure_temperature()
With open('sample.csv','r') as file:
    f = csv.DictReader()
count = 0
for key in f:
avg_temp = temperature + avg_temp
    count = count + 1
sample_test = avg_temp/count
c = datetime.time.now()
if now – time < 2:00
if abs(sample_test – Temp_sensor) > 10:
 raise exc
logger.critical('Temp sensor not supporing')
    else:
        pass
```

Code-2: Checking devices using prediction.

The function matches with the matching dataset with the defined average time hour of the day. If the constraints are matched, then the device will predict whether the device is working properly or not. If this is working properly then the device will add that data to the csv table to once again. In this way, the analysis of the device will get smarter and efficient. This is simple linear regression like algorithm which will predict the condition of the device. The sample dataset which the device would refer is shown ass below.

Time	Temperature (Celcius)	Humidity(%)	Light (%)	Motion
23	35	56	33	yes
0	32	45	33	yes
1	31	89	0	no
2	35	55	0	no
3	36	50	0	no
4	25	42	0	no
5	40	78	10	no
6	23	95	22	no
7	22	62	38	yes
8	21	63	40	yes
9	28	80	60	yes
10	29	80	60	yes
11	31	93	60	yes
12	33	90	70	yes
13	30	75	95	yes
14	30	85	90	yes
15	30	90	82	yes
16	29	72	80	yes

Table-1: Smart reference table to check condition of devices.

After checking and establishing all the hardware devices, the third and main part of the device is to initiate the crontab. In simple words, the crontab is used to run the processes or the programs in the regular time stamps or the regular time intervals. The crontab is normally used in the server level. In the crontab there are six fields are present, they are as follows:

1. Minute.
2. Hour.
3. Day of Month
4. Month of year
5. Day of the week
6. The command

Here we can define the crontab from minute to the year level. To insert or update the command, we use the following command:

$ crontab -e

Here by using this command, the crontab editor will get open where we will enter the crontab entries. Instead of creating the crontab entries by manual process, we use the automated file operation process. The crontab is located on the below location:

/var/spool/cron/crontabs/

Using this path, instead of going on manual process of opening crontab editor and entering the manual

process, we are creating the automated process which will get added to the site, the simple shell help view of crontab editor is as shown in below figure:

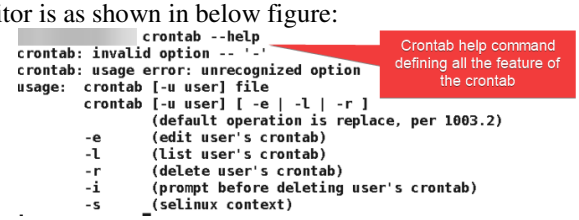

Fig-7: Crontab help option in Linux.

We can directly add the crontab with altering the file. But I just chosen the easier option, instead of directly disturbing the crontab file directly, we can use the crontab module of python. The code to schedule the job using the crontab using the python is shown in below:

```
Cron = CronTab(user = True)
Cron = Crontab(tabfile = 'newfile.tab')
Cron = CronTab(tab = " " " * * * * * * command " " ")
```

Code – 3: Scheculing in crontab using Python.

Using the above statements, we are scheduling the jobs as well as the programs when ever needed, hence it will run all the jobs continuously as well as refreshing itself.

In the next step it checks whether any of the programs are posted to the device which is stored in a separate user defined directory.

/home/scheduledprograms/

All the posted paths are directly stored in this path and when the device turns on, if the device crontab doesnot has the entry of the program which is present in the above directory path, then immediately it adds the function or command to the crontab, and we also can schedule the device.

Finally, the task management and security functions are works same as the user defined functions which can be uploaded on time and deleted. The security function can only upload by the centralized authority. The best security for the privacy is to create the tokens which can provide the read access or write access to the particular fields of the device. The main purpose of creating security access only to the main server side is due to the threat of hacking.

The more important than all these stuffs, the crontab's main work is to send the energy consumption report every hour. If the reporting has stopped then within two hours, the client device sends an email alert and stops functioning. This will cause threat to hackers. But this threat can be solved more clearly.

All the processes defined above is called one cycle, when the device once complete the cycle then again, this start from third step.

Server Side:

In the server side or we can also call it as the centralized authority, there is nothing but the tracking system. In the centralized authority, instead of the monitoring and security management, there is nothing much. But for internal stuff, we need all those processes which the normal server processes. The basic processes are managing the discs, back failure tracker and much more.

As they are the common terms, we have some application level functions also. As the server we have used the Cent OS which is very lightest and very powerful web operating system which can be done in any platform like the netapp operating system, oracle zfs operating system and much more.

The basic algorithm as the server is shown below:

1. Start.
2. check for the data collection every hour.
3. List devices from which no signal received.
4. If no signal received > 2hrs:
   a. Set as decommissioned.

Algorithm-3: Server Work in brief.

The function of the server in the basic words is only to tracking all the devices and nothing else.

In the server also should be developed on the Python. In this process, instead of Database we are using the csv file which is more efficient and simpler than the mysql or sql lite.

Conclusion:

The Home Automation with Smart Electric Power Monitoring Supply using IOT is the technology is the technology for the future of electricity supply, this project helps the people to reduce the electricity consumption with the smart home automation system, using this process, the people can track their home electricity consumption as well as the centralized with recognizing the electric power supply level and mode of consumption. This also helps the government to understand the upcoming electric crisis in the future.

The project creates an interface for the home automation on which the people can implement their custom type of the interface which is useful for defining their own custom interface and also to change the custom interface with less resources. The updating and upgrading of the automation tool programs are easy and simple. The use of separate GSM Module gives the network security which can make the transmission of data private rather than transforming in the private.

This is the device technology which can be platform for all kinds of Home Automation. This also makes the easy prediction of energy consumption and much more. The advantages are dependent on the type of interface used.

The main limitation of the device is the cost, device implementation is highly expensive, and this cannot be maintained by everyone. But this will benefit in many ways for the homes who implemented the home automation systems. The second limitation is the incompatibility of the hardware, The Raspberry is capable of running all the sensors but it is not capable to run continuously due to its extreme heat productions. The third

limitation is the lack of graphical interface. The implementation of graphical interface can cause for easy illegal access inside the device, hence the graphical access is not provided.

References:

1. Rajeev Piyareand Seong Ro Lee: "Smart Home-Control and Monitoring System Using Smart Phone" ,July 2013, ResearchGate.

2. Deepali Jawlie, Mohd. Mohsin, Shreerang Nandanwar, Mayur Shingate: "Home Automation and Security System Using Android ADK", Volume – 3 Issue 2 (March 2013), IJECCT.

3. Abhay Kumar, Neha Tiwari: "Energy Efficient Smart Home Automation System", Volume 3 Issue 1, January 2018.

4. Mukesh Kumar, Shimi S. L.: "Voice recognition Based home Automation Ssytem for paralyzed people", Volume 4, Issue 10, October 2015.

5. Condoluci, M., Dohler, M., Araniti, G., Molinaro, A., Zheng, K.: Toward 5G densenets: architectural advances for effective machine-type communications over femtocells. IEEE, Commun. Mag. **53**(1), 134–141 (2015),doi: 10.1109/MCOM.2015.7010526

6. Masek, P., Hosek, J., Kovac, D., Kropfl, F.: M2M gateway: the centerpiece of future home. In: 2014 6th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT). St. Petersburg, Russia, pp. 286–293 (2014). ISBN: 978-1-4799-5290-8.

7. Di Fazio, A.R., Erseghe, T., Ghiani, E., Murroni, M., Siano, P., Silvestro, F.: Integration of renewable energy sources, energy storage systems, and electrical vehicles with smart power distribution networks. J. Ambient Intell. Hum. Comput. **4**(6), 663–671 (2013).

8. EN 13757-4:2005: Communication systems for meters and remote reading of meters- Part 4: Wireless meter readout (Radio meter reading for operation in the 868 MHz to 870 MHz SRD band)

9. Hosek, J., Masek, P., Ries, M., Kovac, D., Bartl, M., Kropfl, F.: Use case study on embedded systems serving as smart home gateways. In: Recent Advances in Circuits, Systems, Automatic Control. Budapest: EUROPMENT, pp. 310–315 (2013). ISBN: 978-960-474-349-0.

10. Hosek, J., Masek, P., Kovac, D., Ries, M., Kropfl, F.: Universal smart energy communication platform. In: 2014 International Conference on Intelligent Green Building, Smart Grid (IGBSG), pp. 1–4. IEEE, Taipei (2014). ISBN: 9781467361217