RESEARCH ARTICLE OPEN ACCESS

Design IoT House installation system

1. Yousef B. Almutairi, 2. Hamad M Alfouzan
Public Authority For Appleid Ducation And Training
Public Authority For Appleid Ducation And Training
Corresponding Author: Yousef H Almutairi

ABSTRACT :
In this paper, an efficient implementation will be designed for an IoT based automation evolving around a house to monitor and control home appliances through a local area network with an internet option. This design concept can be aptly incorporated to make a smarter and safer solution. This project aims at controlling home appliances through Wifi using smartphones and/or a web interface. Hardware to be implemented is raspberry pi as a server, with more focus on building a smart wireless home system, whereas home appliances like lights, fan and door lock are remotely controlled through a simple website. The designed project create a price effective Home Automation system.

Date of Submission: 30-03-2019 Date of acceptance: 13-04-2019

I. OBJECTIVES:

The objective is to develop a design, prototype, and software for multi-purpose control devices and supported interfaces in the home. This project shall have a quick setup, low power consumption, and automated discovery of smart devices over network interfaces.

II. INTRODUCTION:

IoT is developing amazingly because of the continuous efforts of a large community, stretching from hobbyists to researchers[1]. The IoT tends to have unlimited applications, as there are seemingly limitless needs in every circle of life. Consumers have found that the available IoT-enabled products are not able to cater to the universe of people's requirements or there is no such product for a specific need. Everything in the home will have a unique Internet address and will be able to be controlled or monitored over the network. This will be possible only if the whole IoT community starts developing and deploying things[2].

Basic Design Requirements:

It is mandatory to ensure comfort around living in design around users. In addition to automating repetitive tasks where possible. Provides a User-friendly interface to simplify user interaction including inexperienced users. Have Network connectivity including the ability to control devices over an internet web-based interface or a mobile application.

Hardware Requirements:

As a hosting server, Raspberry pi3 is an excellent choice for performance versus price; it includes an Ethernet 10/100/1000 Mbit network interface, a Wi-Fi 802.11b/g/n standard network interface [3]. As an IoT device ESP8266 is a well-known Wi-Fi capable development board for IOT solutions, it comes in multiple forms and versions. It hosts a Deep-Sleep mode mechanism, can work in different modes according to software selection including Access point or as a Station hosting a web server [4].

In addition to these devices, we can implement a range of motion sensors, Light control, integrate home security with a possibility of streaming audio/video, integration of Loudspeaker as an audio output, The ability to interface with voice recognition platform to add artificial intelligence. It is possible also to integrate an uninterrupted power supply with the possibility for Solar panel integration.

Software Requirements:

There is multiple smart home software solution, out of the open source group Home Assistant shines above the rest, it has a very friendly user interface that can be used over web or mobile.

Home Assistant (Also called Hassio) configuration is based on YAML (Yet Another Markup Language) it is a form of a human-friendly language for programming.

Hassio interface is loaded with tools that make automation simpler, on top of these tools is Configurator, it is a YAML editor with a YAML language error corrector. Automation also is a

YAML based tool used for automating tasks. Groups, which is a YAML based tool to create groups of devices.

Moreover, there are much community and third party add ons that can be integrated into Hassio, which they can simplify the automation process. Users can also create their own add ons and share it with others in the open source community [5].

IoT Protocols:

Some of the common IoT protocols(but not limited to):

Infrastructure (ex: 6LowPAN, IPv4/IPv6, RPL)
Identification (ex: EPC, uCode, IPv6, URIs)
Comms / Transport (ex: Wifi, Bluetooth, LPWAN)
Discovery (ex: Physical Web, mDNS, DNS-SD)
Data Protocols (ex: MQTT, CoAP, AMQP, WebSocket, Node)
Device Management (ex: TR-069, OMA-DM)
Semantic (ex: JSON-LD, Web Thing Model)
Multi-layer Frameworks (ex: Alljoyn, IoTivity, Weave, Homekit) [6]

` From the above mentioned IoT protocols, MQTT (Message Queue Telemetry Transport) is the most common among them. MQTT is a lightweight data-centric application layer protocol which runs on top of the transport layer of the 7 layers OSI TCP/IP network model, MQTT is lightweight hence it extremely suitable to short on memory IoT devices. MQTT designed to ensure reliable message delivery in constrained environments such as low network bandwidth and unreliable network resource-constrained devices. While using MQTT devices does not need to know the existence of other devices, and this is because all devices publish and subscribe messages to a central MQTT broker. The broker will handle the delivery of messages to other subscribers, hence using this protocol between IoT devices work on local area networks as well as networks which use Network Address Translation. MQTT also provides a mechanism for quality of service for message delivery, QoS 0, QoS 1, QoS 2 are the available flags which give a message reliable delivery. MQTT specification also provides an Authentication feature. [7]

IoT Devices

IoT devices with the improvement of technology in software and hardware became more powerful and lower in cost. IoT devices will be programmable, host applications. And it will be possible to manage these applications remotely. Remote management refers to access the application programming interfaces of the device to install, update, delete and manipulate the device's applications remotely.

Acknowledging the different nature of IoT devices, each kind of device may have its exclusive communication protocol, knowledge requirements of all corresponding communication protocol for the possibility of remote management of every IoT devices. Communication will be difficult to interact with all the IoT devices without having an understanding of the different protocols. The web platform provides a common interacting platform for these IoT devices. Besides, MQTT becomes the inter-IoT communication protocol so that such proprietary protocols can interact together.

IoT devices such as sensors, actuators, and embedded devices and are millions and increasing very fast. It will be impossible for these millions of devices which go on and off for their functioning, to allocate an IP address.

However until 5G networks become up and running and cost-effective, IPv4 address will not suffice for the addressing of all IoT devices. These millions of IoT devices to connect to the Internet a Virtual Private Network (VPN) or Network Address Translations (NAT) must be used. This may be a problem in REST, as it based on HTTP request-response between client and server and they must have an IP address to communicate with each other. Hence, there might be an issue when the server tries to communicate with the client which resides behind the NAT. Here, the server cannot connect to the client as it is unaware of the IP address of the client which is behind the NAT. The solution for communication between server and client in NAT case can be to use the protocol which does not depend on the IP address of the client-server endpoints but only focuses on the delivery of data between those endpoints.[8]

IoT Hardware:

IoT hardware covers a significant area of devices like devices for routing bridges, sensors, switches, microcontrollers, etc. Such IoT devices are used to manage specific tasks and functions like system activation, managing security, performing actions, handling communications, and detecting specific support objectives and actions.

The components of IoT Hardware can vary from low power single board processors such as Arduino Uno in which it can have smaller boards that can be plugged into the main boards to improve and increase its functionality like GPS, interactive displays, photo sensors, and temperature sensors. A programmer then determines the board's input and output ports, to create a circuit design explaining the interaction of these inputs and outputs.

An alternative IoT platform is Raspberry Pi, a tiny computer very affordable and can host an

entire web server. It can run Windows 10 on it in addition to IoT Core and relatively powerful processing capabilities especially while using Python programming language.

BeagleBoard single board computer is another alternative; it uses Linux based OS on an ARM processor, it has more processing power than Raspberry Pi. More alternatives also are available from Galileo and Edison boards which are suitable for large scale production, and last but not the least Qualcomm who has manufactured an array of enterprise-level IoT technologies for Automotive and Cameras to healthcare.

a. Hardware in IoT divided into four building blocks:

i. Thing

"Thing" building block is the asset that needs to be controlled or monitored or measured and observed closely. In many products such "thing" is included in smart devices.

ii. Data Acquisition Module

This module acquires physical signals from the "thing" being observed or monitored and then converts these signals into digital signals that can be easily manipulated or interpreted or encrypted and compressed by a computer.

iii. Data Processing Module

This third building block is the physical computer and central processor for local analytics, local storage, and performs other computing operations

iv. Communication Module

This last IoT Hardware Building block enables communications with the Cloud Platform, and with 3rd party systems either locally or abroad.

b. IoT Sensors

Sensors are essential IoT hardware. They include a variety of modules such as Energy, RF, power management, and sensing modules.

IOT SENSORS

IoT Software:

Since embedded IoT systems have less storage and processing power, it requires different software requirements. Most commonly used operating system for embedded systems are Linux or Android.IoT software includes a wide variety of software and programming languages like C++, Java, Google's Go, Python, or Parasail.The following illustration specifies each IoT programing language and its specific area of programming.

Moreover, a software that was developed by Theo Arends which is called Tasmota (Theo Arends Sonoff MQTT Over The Air) is designed to be hosted on esp8266 processor-based devices. This software removes the complications of programming to simple drop-down menu selections through a web interface. Once proper settings are done, there is no need for this web interface, and there is an option to change it from the configuration page into a user page that hosts only the switching or sensors reading, or it can also be shut down for additional security, and having automation handled by Hassio. [9]

IoT software platforms can be broadly classified into four types; the first mainly focuses on the connectivity of devices via subscriber identification module cards, that is, telecommunication networks, termed connectivity/M2M platforms. Next classification is intended to provide processing power and hosting space for IoT applications and is termed infrastructure as service platforms. Next category is hardware-specific software platforms that are built for specific IoT hardware items. And the last classification caters to the consumer/enterprise software extensions that offer software packages and integration of IoT devices.

IOT SOFTWARE

Testing Tools (Node-Red):

Node-RED is a flow-based visual programming tool, which was originally developed by IBM Emerging Technology Service team who are now part of JS Foundation. Flow-based

Programming was invented by J. Paul Morisson in the 1970s, and it is designed to describe application behavior as a network of black-boxes or nodes. Hence they are called Node-RED. Each node has its own well-defined purpose, it is given data, does something with this data, passes the data on. The network is responsible for data flow between nodes. Having its interface visual makes it accessible to a wider range of users with more specific tasks concerning their area of expertise [10].

Case Study:

A Smart Home is an automated home; it is based on using leading-edge technologies to automate daily repetitive tasks. It is not a new idea; it started decades ago when fiction writers and their perspective view of the future and what would it look like with robotics and smart feature homes. In which it was a long journey starting with the invention of electricity through the introduction of home appliances to introducing the first computer in the kitchen, later on, the need to care for elderly people produced Gerontology which the care of elderly people through automation, next was the smart homes and smart automation. Today's smart homes are more about security and living greener[11]. Modern trends in home automation include remote mobile control, automated lights, automated in house climate adjustment, scheduling appliances, mobile/email/text notifications, and remote video surveillance, all centralized into one management system.

With the current cost-effective manufacturing, IoT devices have become relatively cheap, IoT devices depend on the futuristic 5G network to host the vast additions of automation in all areas around the world. But in this study, we are concentrating on introducing IoT devices to home automation. The model home design for our study consists of 2 floors, and the ground floor includes a living room, a guest room with a bathroom, kitchen, washing room. 1st floor includes a master bedroom, two bedrooms.Automation will be handled through the control of lights, shutters, power sockets, indoor climate control, a possibility of interfacing irrigation control in the future.

Exterior lighting shall be controlled to switch on, and shutters shall close 15 minuses to sunset using Hassio automation, groups, and the sun / MQTT platform.For IoT devices, we have chosen to have TASMOTA installed on them, and we have used MQTT as a communication protocol.Hardware used in the study is the cost-effective esp-01 for small applications due to its limitation in the number of I/O (inputs/outputs), Nodemcu is another common version of the same chip which has more features and I/Os.Some additional sensors such as PIR (Passive Infra-Red) were used for motion and detecting people presence, Temperature and humidity and barometric pressure sensors were used to monitor in house climate.LDR (Light Dependent Resistor) used as a light intensity detector for further redundancy of automation.A significant requirement that we have noticed in this study is a perfect WiFi network coverage and a proper power source for each IoT device.

To create Hassio automation, we start in grouping the same function devices into groups to simplify automation and isolate them in a separate group.YAML file to simplify the looks of the configuration, then we add the required configuration parameters to the configuration.YAML file, then we write the automation.YAML file which will consist of a declared platform, Trigger, Condition, and Action for each automated device/devices.It is also possible to create a Script for repetitive tasks. It is also possible to create a Scene in which it is an automation that controls multiple devices with one Scene command.

REFERENCES:

[1]. He, Jing, et al. "Integrating Internet of Things (IoT) into STEM undergraduate education: Case study of a modern technology infused courseware for embedded system course." 2016 IEEE Frontiers in Education Conference (FIE). IEEE, 2016.
[2]. Pavithra, D., and Ranjith Balakrishnan. "IoT based monitoring and control system for home automation." 2015 global conference on communication technologies (GCCT). IEEE, 2015.
[3]. https://en.wikipedia.org/wiki/Raspberry_Pi
[4]. https://www.espressif.com
[5]. https://www.home-assistant.io/
[6]. https://www.postscapes.com/internet-of-things-protocols/
[7]. http://mqtt.org/documentation
[8]. SRIJAN MANANDHAR-MQTT BASED COMMUNICATION IN IOT
[9]. https://data-flair.training/blogs/iot-hardware
[10]. https://nodered.org/
[11]. https://www.iotevolutionworld.com

Yousef B. Almutairi" Design IoT House installation system" International Journal of Engineering Research and Applications (IJERA), Vol. 09, No.03, 2019, pp. 90-93