OPTIMIZATION OF WIRELESS SENSOR NETWORK

Contents

1 Introduction to WSN **2**

2 WSN Technology **2**

3 Software Used **2**

3.1 Arduino IDE 2

3.1.1 RF24Network library 2

3.1.2 Wifi.h library 2

3.1.3 PubSubClient Library 2

3.2 Node-RED 3

4 Hardware Used **3**

4.1 PIR Sensor 3

4.2 Power Source 4

4.3 Communication module 4

4.4 Arduino Pro mini 4

5 Theory of Operation **4**

5.1 Working of nrf24l01 5

5.2 MQTT Protocol 6

5.2.1 Working of MQTT Protocol 6

6 Block diagram: **7**

7 Output **8**

8 Conclusion **8**

1 Introduction to WSN

A wireless sensor network (WSN) consists of spatially distributed autonomous sensors to monitor physical or environmental conditions, such as temperature,sound,pressure, etc. and to cooperatively pass their data through the network to a main location. The more modern networks are bi-directional, also enabling control of sensor activity. The development of wireless sensor networks was motivated by military applications such as battlefield surveillance; today such networks are used in many industrial and consumer applications, such as industrial process monitoring and control, machine health monitoring, and so on

2 WSN Technology

The WSN is built of "nodes" –from a few to several hundreds or even thousands, where each node is connected to a sensor. Each such sensor network node has typically several parts: a radio transceiver with an antenna, a microcontroller, an electronic circuit for interfacing with the sensors and an energy source, usually a battery or an embedded form of energy harvesting. The cost of sensor nodes is similarly variable, ranging from a few to hundreds of dollars, depending on the complexity of the individual sensor nodes. Size and cost constraints on sensor nodes result in corresponding constraints on resources such as energy, memory, computational speed and communications bandwidth. The topology of the WSNs can vary from a simple star network to an advanced multi-hop wireless mesh network.

3 Software Used

3.1 Arduino IDE

The Arduino integrated development environment (IDE) is a cross-platform application that is used to write and upload programs to Arduino compatible boards and other vendor development boards ,such as ESP32.
In the Arduino IDE,libraries used are:

3.1.1 RF24Network library

This library enables an easy way to build a wireless network with many Arduino boards(with nrf's) communicating to each other.

3.1.2 Wifi.h library

This library allows an Arduino board to connect to the internet using Wi-Fi. It can serve as either a server accepting incoming connections or a client making outgoing ones.

3.1.3 PubSubClient Library

This library allows you to send and receive MQTT messages. It supports the latest MQTT 3.1.1 protocol and can be configured to use the older MQTT 3.1 if needed.

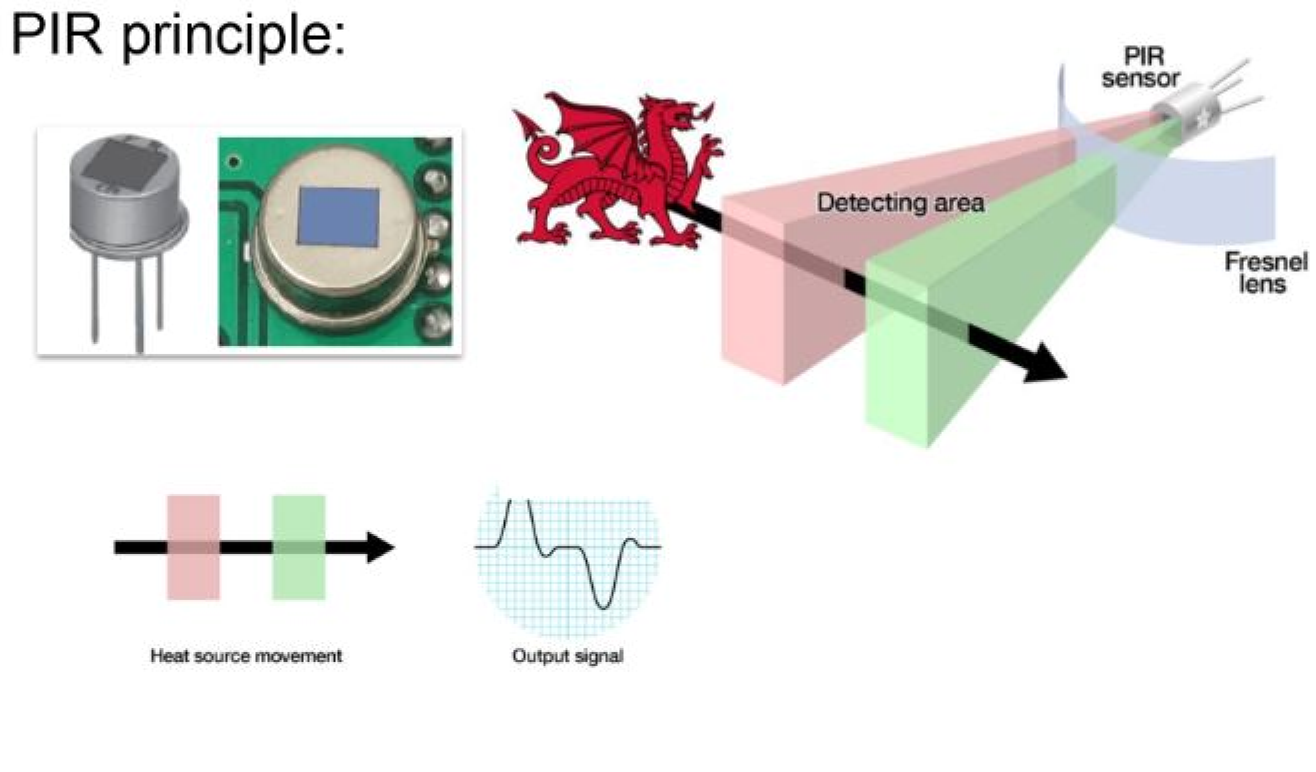
3.2 Node-RED

Node-RED is a flow-based development tool for wiring together hardware devices such as Arduino boards,Raspberry Pi and ESP32. MQTT nodes can make properly configured using Node-RED.The flows created in Node-RED are stored and sent using JSON.

4 Hardware Used

4.1 PIR Sensor

Passive Infrared (PIR) sensor's functionality is based on infrared radiation emitted from the human body. It is a useful tool for detection of human movement as it detects a change in infrared radiation as a result of moving warm-blooded objects within their range. All objects, including human beings, produce electromagnetic radiations. The wavelengths of these radiations are dependent on the temperature of objects. Human beings emit infrared radiation with wavelengths ranging between 0.7 and 300 micrometres [16, 29]. On the other hand, normal body temperature of human beings radiates IR at wavelengths of 10 microme-tres to 12 micrometres.PIR sensors detect motion by sensing the fluctuations of infrared radiation.



4.2 Power Source

The power source for the node also depends on the node's intended use.Since we need power efficiency along with rechargeable feature,Li-ion 18650 batteries are used.Their features include light weight,low cost and provides a 3.3V output which is enough to power the arduino pro mini.The power source is connected to the node and provides energy required to run the sensors, arduino board, and the nrf.

4.3 Communication module

nRF24L01+ is used as a communication module which is a radio transceiver for the 2.4-2.5 GHz ISM band. We can select the output power channel and protocol by setting through the SPI port. The current consumption for the nRF24L01+ is extremely low - under the transmitter mode, when the transmitting power is 0dBm, the current consumption is only 11.3mA; under the receiving mode, it is 13.5mA.Due its low power consumption,we used this module instead of WiFi module.

4.4 Arduino Pro mini

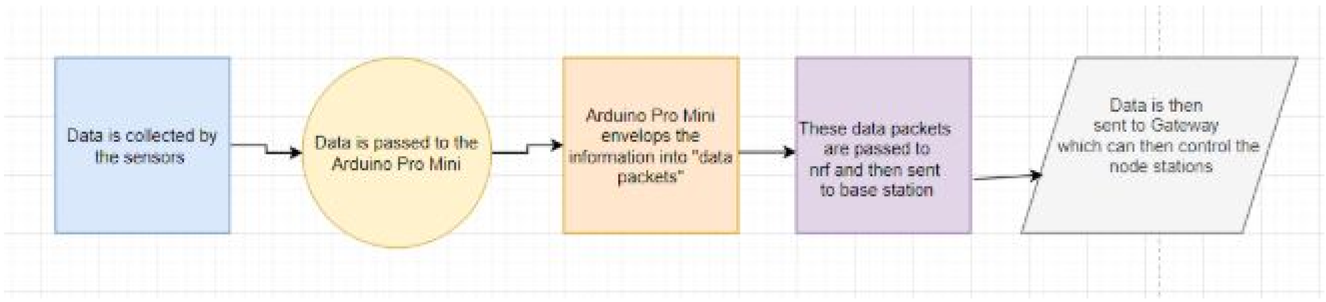
The Arduino Pro Mini is a microcontroller board based on the ATmega328. It has 14 digital input/output pins, 6 analog inputs, an on-board resonator, a reset button, and holes for mounting pin headers. A six pin header can be connected to an FTDI cable to provide USB power and communication to the board.

There are two versions of the Pro Mini. One runs at 3.3V and 8 MHz, the other at 5V and 16 MHz.We are using 3.3V 8MHz version.

The basic functions of the arduino is to make decisions and deal with collected data. The arduino pro mini stores collected data in its memory until enough information has been collected. Once this point is reached, the microprocessor portion of the electronic brain then puts the data in "envelopes," or packages of data formatted for greatest transferring efficiency. These envelopes are then sent to the radio for broadcast.The node is connected to the base station and interacts with the sensors and radio.

5 Theory of Operation

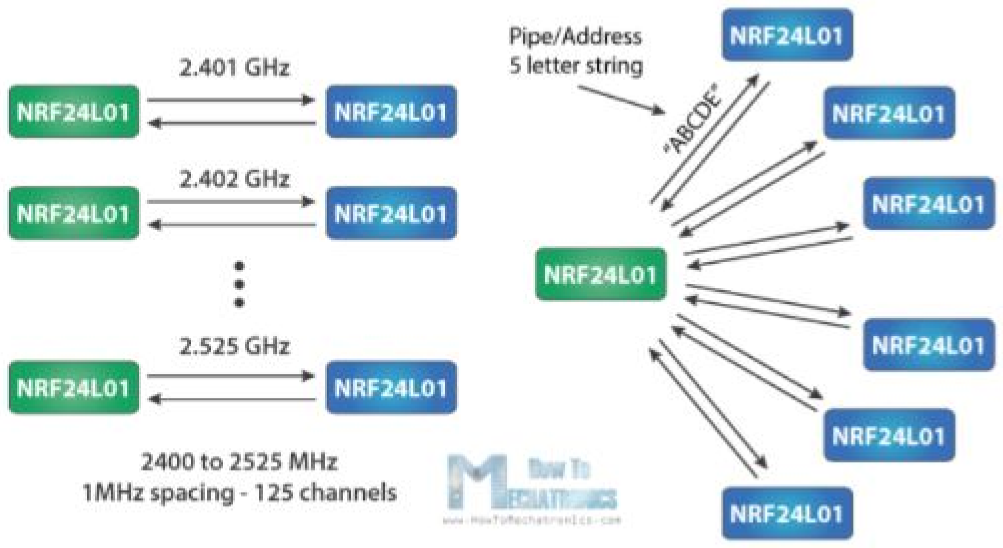
Nodes collect and transfer data using four stages: collecting the data, processing the data, packaging the data, and communicate the data. Each node collects data using its PIR sensor. After collecting the data, the node then sends the data to ESP32(base station) via nrf24l01.


Data is collected by the sensors
Data is passed to the Arduino Pro Mini
Arduino Pro Mini envelops the information into "data packets"
These data packets are passed to nrf and then sent to base station
Data is then sent to Gateway which can then control the node stations

5.1 Working of nrf24l01

Working of nrf24l01: The nRF24L01,used for RF communication is having up to 6 channels (pipes) of radio communication open in a receiving or read mode simultaneously. This takes the form of a hub receiver and up to six transmitter nodes .The addresses or pipes must have a distinct pattern of bytes, only the fifth byte is entirely unique among all the pipes and is known as the Least Significant Byte (LSB). Three pins of nrf are for the SPI communication and they need to be connected to the SPI pins of the Arduino.The pins CSN and CE can be connected to any digital pin of the Arduino board and they are used for setting the module in standby or active mode, as well as for switching between transmit or command mode. The last pin is an interrupt pin which doesn't have to be used.

The nrf24L01 communication is showing in the figure below:



Once our base station(ESP-32) receives the data through nrf,it transmits the data to the gateway using Wi-Fi module.Data is sent to the gateway using MQTT Protocol.

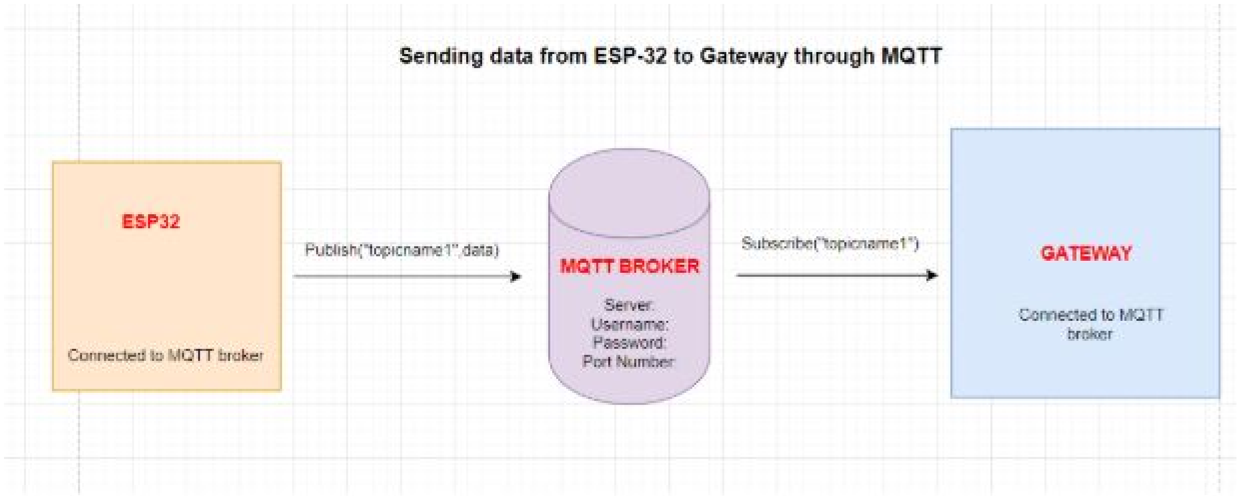
5.2 MQTT Protocol

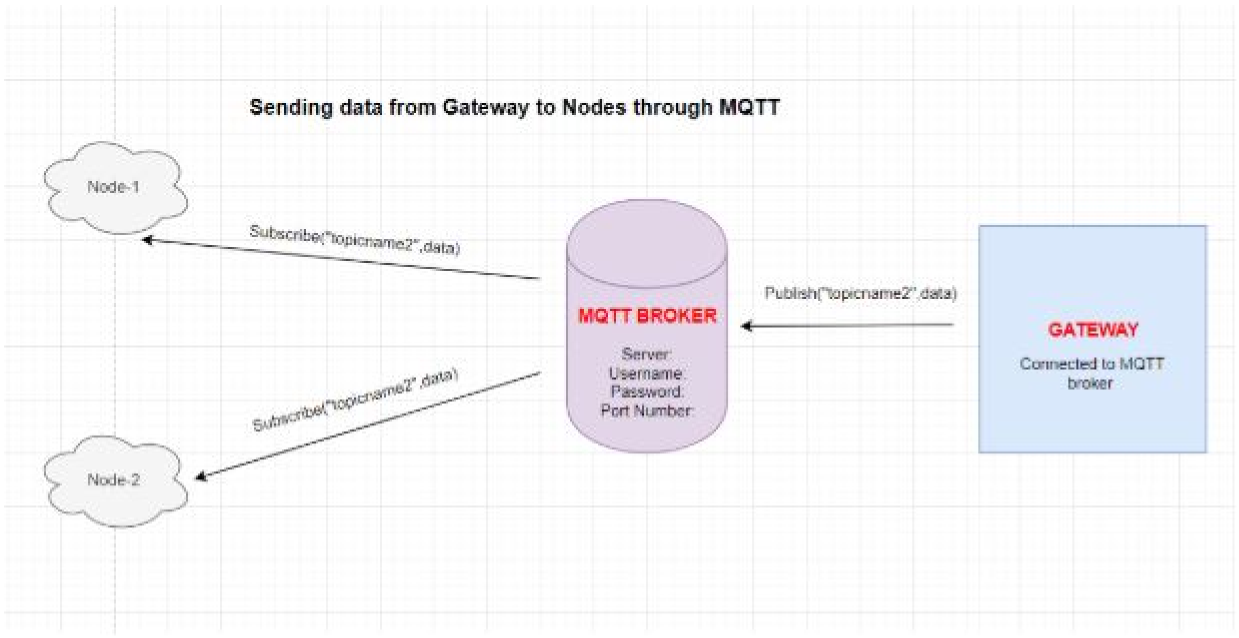
MQTT is a very light weight protocol designed for loT projects. It might be a question why we need MQTT protocol when HTTP is still exists and working fine. The clarification of that question is that HTTP is great for doing request and response like a client may ask for information to a server and server will respond according to that request but it doesn't really have a good solution when a source of information should push a change to many clients and there is no built in support for quality of service. The text based format of HTTP requires more bandwidth and any device that act as host must have a server installed and keep the server live to answer incoming request consumes a lot of battery life. First of all it introduce a publish and subscribe messaging pattern which means that any source of information such as a sensor can publish its data and then any client can subscribe that data All these is happening in a broker like keep track of all subscription and publications so when a publisher sense and update with new data it publishes a message and the broker takes care of sending the new data to all subscribers. The fixed header of MQTT is actually only 2 bytes and it also has a small implementation footprint that requires less battery. Each MQTT control packet consist of three parts. a fixed header, variable header and payload. MQTT has a small header overload that makes it appropriate for loT by lowering the amount of data transmitted over week network.

5.2.1 Working of MQTT Protocol

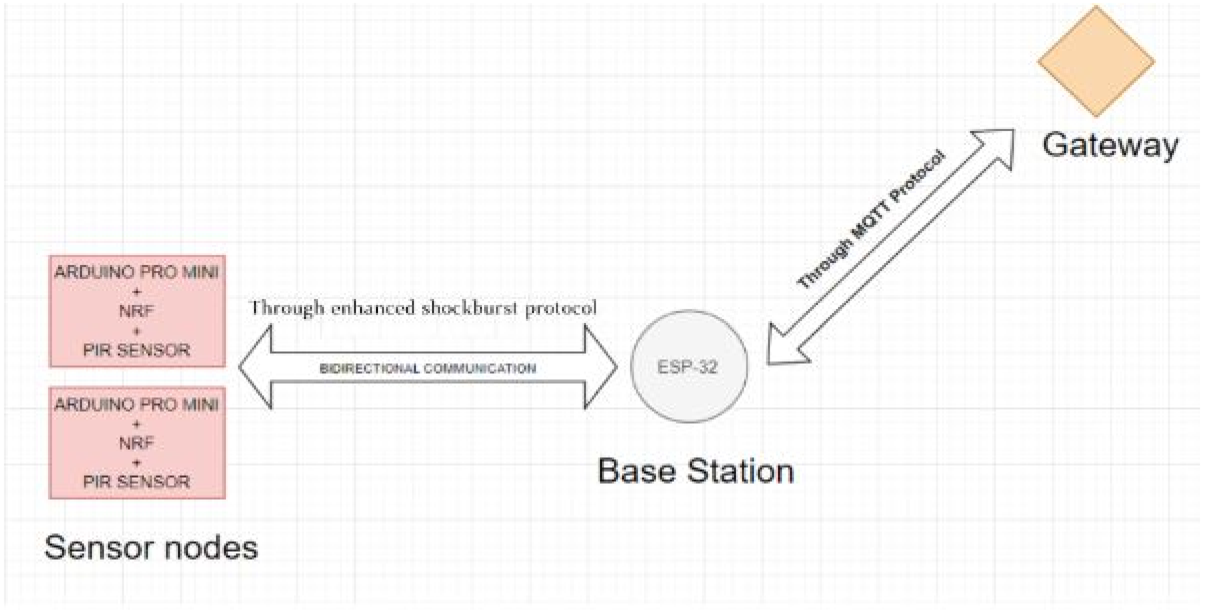
It follows publish/subscribe messaging protocol which allows a message to be published once and multiple subscribers to receive the message providing decoupling between the publisher and subscriber. A publisher publishes a message in the Topics.

The subscribers subscribe to the Topics for message. A Broker,which is a message server matches publications to subscriptions.If one or more matches the message the broker will deliver the message to each subscriber.



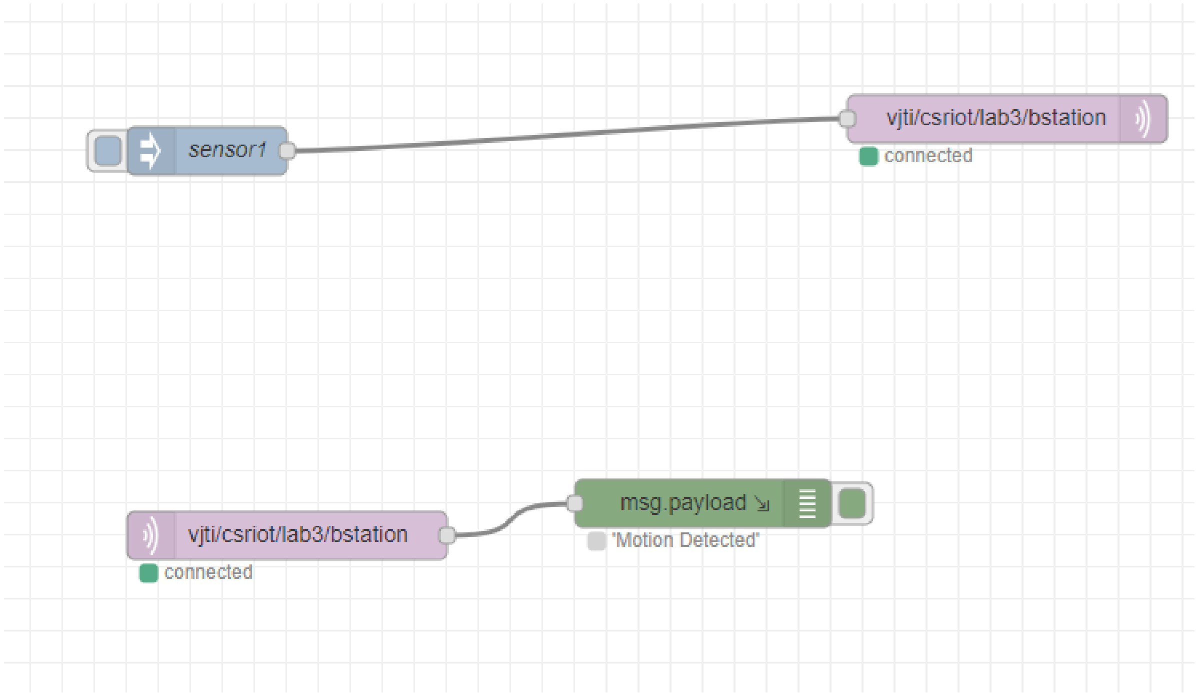

Sending data from Gateway to Nodes through MQTT
Node-1
Subscribe("topicname2",data)
MQTT BROKER
Server:
Username:
Password:
Port Number:
Publish("topicname2",data)
GATEWAY
Connected to MQTT broker
Subscribe("topicname2",data)
Node-2

6 Block diagram:

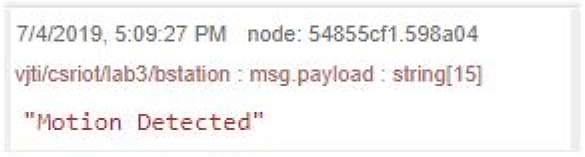


7 Output

On the Node-Red flow chart,the output sent by our nodes is fetched by the gateway as seen in the purple node. The message("Motion Detected") is obtained in message payload of the gateway.



On the Message Debug section,Message "Motion detected" is displayed with the other information like the topic on which it is published,node on which it is received along with the time-stamp.



```
7/4/2019, 5:09:27 PM   node: 54855cf1.598a04
vjti/csriot/lab3/bstation : msg.payload : string[15]
"Motion Detected"
```

8 Conclusion

WSNs are still uncommon.This is still a young technology, allowing WSNs great growth potential. The question that is constantly asked is "what new use can we come up with for this?" There have been many answers to this question, including: data collection, strategic mine placement, machinery monitoring, and much more. Another reason they are successful is the way that the nodes work together to form a network.The future of WSNs is bright, as increasing attention is brought to their uses.