**PROGRESS REPORT**

**for PROJECT PART-I (EC 782)**

**ON**

**IoT Based Drainage Monitoring System**

**(In partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Electronics & Communication Engineering**

**Maulana Abul Kalam Azad University of Technology)**

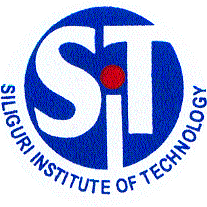
**Submitted by**

|  |  |  |
| --- | --- | --- |
| **Sl no.** | **Name** | **Roll No.** |
| **1** | **NEELESH BISWAS** | **11900315027** |
| **2** | **DURBA SARKAR** | **11900315016** |
| **3** | **AGRAPRIYA DAS** | **11900315004** |

**Under the guidance of**

**Mr. Debajyoti Misra**

**Assistant Professor, ECE Dept., SIT**



**DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING**

**SILIGURI INSTITUTE OF TECHNOLOGY**

**PO: SUKNA, SILIGURI, PIN: 734 009, WEST BENGAL**

**2018-2019**

**Abstract:**

For making a smart city one needs to consider many parameters such as smart water, smart electricity, smart transportation etc. There will be a need of smart underground infrastructure which includes underground water pipelines, communication cables, gas pipelines, electric flow, etc. As most of the cities in India have adopted underground drainage system, it is very important that this system should work in a proper manner to keep the city clean, safe and healthy. If they fail to maintain the drainage system the pure water may get contaminated with drainage water and can spread infectious diseases. So different kind of work has been done to detect, maintain and manage these underground systems. This project represents the implementation and design functions for monitoring and managing underground drainage system with different approaches using Internet of Things.

**Goals:**

In this project our main focus is to maintain the drainage system so that any kind of clog or water blockage related problems can be prevented. Doing so, there can be:

* Relaxation for manpower.
* Within an instance; we’ll realize there is any kind of clog in the system (if any).
* Monitor the water level by showing relevant data.

By this we can also maintain proper flow of the water level of the drainage system. Through appropriate IoT modules, sensors, actuators and with the help of Internet, we’ll try to achieve the goal and make this project a success.

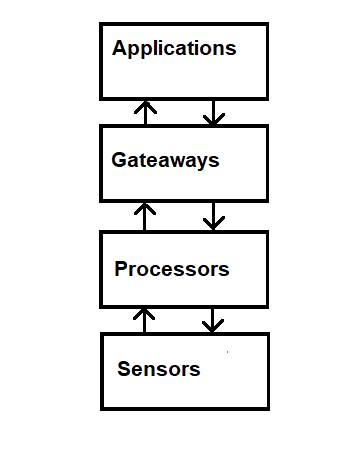
**Synopsis:**

1. **Introduction:**

The Internet of Things (IoT) is the network of physical objects or "things“embedded with electronics, software, sensors, and network connectivity, which enables these objects to collect and exchange data. In simple words, Internet of Things (IOT) is an ecosystem of connected physical objects that are accessible through the internet.

The term “IoT” cannot be clarified without its working principle. IoT works with the help of four basic components: Application, Gateaways, Processors and Sensors.

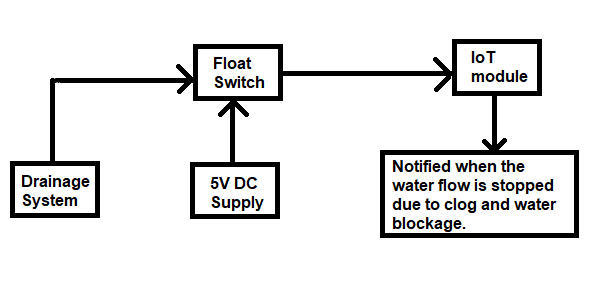
1. Application: The applications for internet connected devices are extensive. Multiple categorizations have been suggested, most of which agree on a separation between consumer, enterprise (business), and infrastructure applications. The ability to network embedded devices with limited CPU, memory and power resources means that IoT finds applications in nearly every field.
2. Getaways: An Internet of Things (IoT) Gateway provides the means to bridge the gap between devices in the field (factory floor, home, etc.), the Cloud, where data is collected, stored and manipulated by enterprise applications, and the user equipment (smart phones, tablets etc.). The IoT Gateway, provides a communication link between the field and the Cloud and can also offer local processing and storage capabilities to provide offline services and if required real time control over the devices in the field. The protocols that support the information exchange between interoperability domains can be classified to message-centric (AMQP, MQTT, [JMS](https://en.wikipedia.org/wiki/Java_Message_Service), REST) and data-centric (DDS, CoAP, XMPP). C, Java, Python and some scripting languages are the preferable choices used by IoT applications.
3. Processor: A Processor performs basic operations in the data provided by the storage or memory. In IoT, the work of a processor is to process the code written in C or Java or in any programming language to a machine language which is uploaded to microcontroller boards like Arduino, ESP8266, etc.
4. Sensors: IoT loses its distinction without sensors. In the broadest definition, a sensor is a device, module, or subsystem whose purpose is to detect events or changes in its environment and send the information to other electronics, frequently a [computer processor](https://en.wikipedia.org/wiki/Computer_processor). A sensor is always used with other electronics, whether as simple as a light or as complex as a computer. Sensors are used in everyday objects which dim or brighten by touching the base, besides innumerable applications of which most people are never aware.

****

**Fig:** Block diagram of IoT

1. **Functional Block Diagram:**

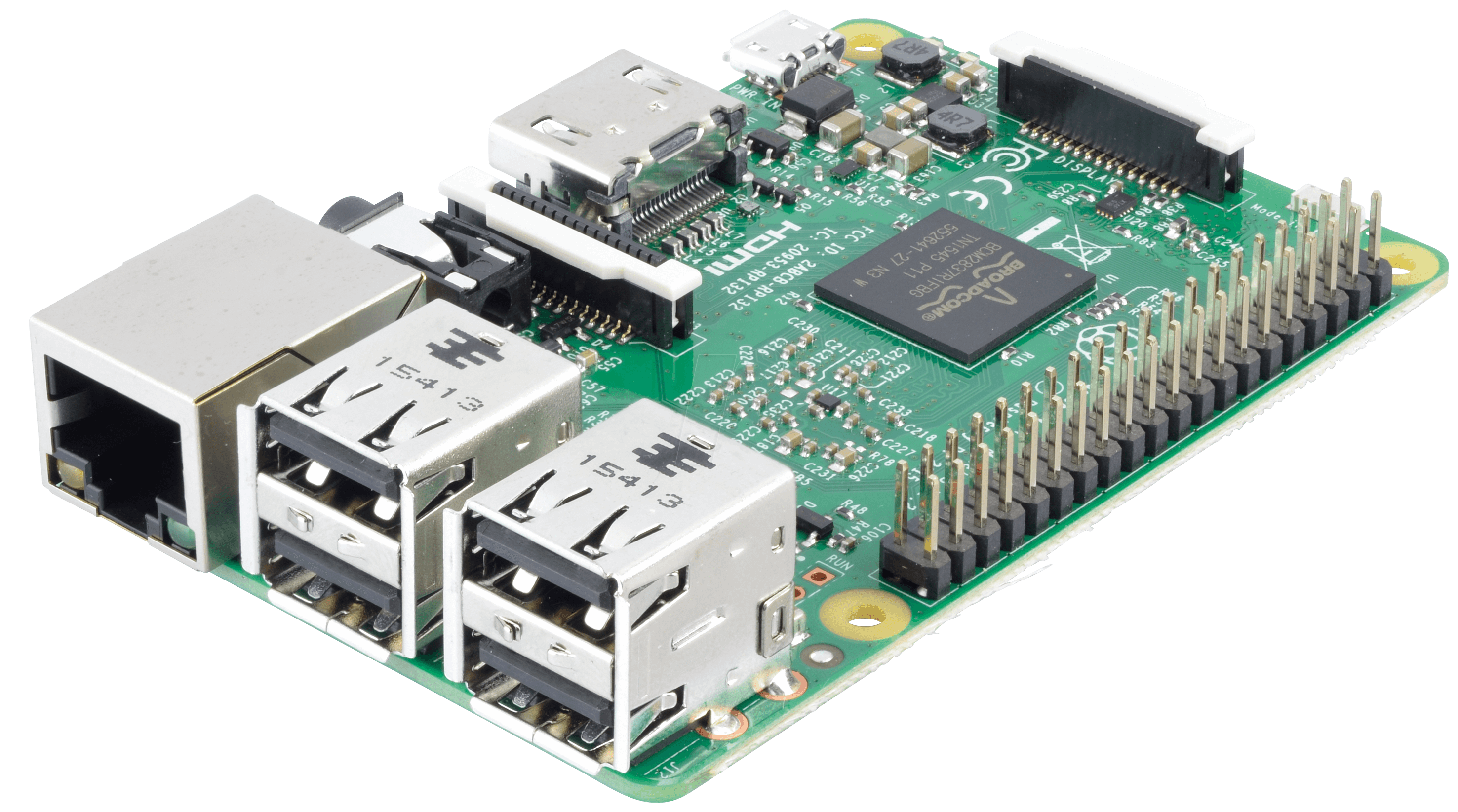
The block diagram describes the monitoring of water level in the drainage system. During the cases of any blockage in the pipe lines or during the affect of storm water in the system, level of water increases. This continues increase in sewage level causes overflow which is to be controlled. This increase in level is monitored using a float switch (level sensor). The signal form the level sensor is fed to the micro controller where the controller commands the IoT module to alarm complaints via mails and/or SMS. The controller is programmed such that the complaint alarms would be triggered repeatedly unless the level reaches the minimum set.

****

**Fig:** Block diagram of our project

1. **About Raspberry Pi 3B:**

Raspberry Pi is a series of single-board computers developed in the UK. Raspberry Pi is highly preferred for programming (especially Python) and electronics projects. But for high processing power than other IoT modules, we use Raspberry Pi 3 B as the main component. It is a computer having an HDMI port, 4 USB Ports, an Ethernet Port, a 3.5mm audio input jack and a micro SD card slot as the SD card being the system storage of the Raspberry Pi 3. And on the other hand, it works like a microcontroller by having 40 GPIO pins for implementing our projects. Unlike Windows, we use Raspbian, a Debian-Linux based OS provided by Raspberry Pi Foundation of UK. Other OSes include Ubuntu Mate, Windows IoT Core, RISC OS, etc.



**Fig:** Raspberry Pi model 3B

1. **Internet Protocols:**

A protocol is a system of rules that allow two or more entities of a [communications system](https://en.wikipedia.org/wiki/Communications_system) to transmit [information](https://en.wikipedia.org/wiki/Information) via any kind of variation of a [physical quantity](https://en.wikipedia.org/wiki/Physical_quantity). The protocol defines the rules, [syntax](https://en.wikipedia.org/wiki/Syntax) and [synchronization](https://en.wikipedia.org/wiki/Synchronization) of [communication](https://en.wikipedia.org/wiki/Communication) and possible [error recovery methods](https://en.wikipedia.org/wiki/Error_detection_and_correction). Protocols may be implemented by [hardware](https://en.wikipedia.org/wiki/Computer_hardware), [software](https://en.wikipedia.org/wiki/Software), or a combination of both. Protocols should therefore specify rules governing the transmission. In general, much of the following should be addressed:

1. Data formats for data exchange.
2. Address formats for data exchange.
3. Address mapping.
4. Routing.
5. Detection of transmission errors.
6. Acknowledgements.
7. Loss of information - timeouts and retries.
8. Direction of information flow.
9. Sequence control.
10. Flow control.

* **TCP/IP:**

The Transmission Control Protocol (TCP) is one of the main [protocols](https://en.wikipedia.org/wiki/Communications_protocol) of the Internet protocol suite. It originated in the initial network implementation in which it complemented the [Internet Protocol](https://en.wikipedia.org/wiki/Internet_Protocol) (IP). Therefore, the entire suite is commonly referred to as [TCP/IP](https://en.wikipedia.org/wiki/TCP/IP). Major Internet applications such as the [World Wide Web](https://en.wikipedia.org/wiki/World_Wide_Web), [email](https://en.wikipedia.org/wiki/Email), [remote administration](https://en.wikipedia.org/wiki/Remote_administration), and [file transfer](https://en.wikipedia.org/wiki/File_transfer) rely on TCP.

The Transmission Control Protocol provides a communication service at an intermediate level between an application program and the Internet Protocol. It provides host-to-host connectivity at the [Transport Layer](https://en.wikipedia.org/wiki/Transport_Layer) of the Internet model.

* **UDP:**

The User Datagram Protocol (UDP) is one of the core members of the [Internet protocol suite](https://en.wikipedia.org/wiki/Internet_protocol_suite). With UDP, computer applications can send messages, in this case referred to as [datagrams](https://en.wikipedia.org/wiki/Datagram" \o "Datagram), to other hosts on an [Internet Protocol](https://en.wikipedia.org/wiki/Internet_Protocol) (IP) network. Prior communications are not required in order to set up [communication channels](https://en.wikipedia.org/wiki/Communication_channel) or data paths.

UDP is suitable for purposes where error checking and correction are either not necessary or are performed in the application; UDP avoids the overhead of such processing in the [protocol stack](https://en.wikipedia.org/wiki/Protocol_stack). Time-sensitive applications often use UDP because dropping packets is preferable to waiting for packets delayed due to [retransmission](https://en.wikipedia.org/wiki/Retransmission_(data_networks)), which may not be an option in a [real-time system](https://en.wikipedia.org/wiki/Real-time_system).

* **Difference between TCP/IP and UDP:**

|  |  |  |
| --- | --- | --- |
| **Parameters** | **TCP/IP** | **UDP** |
| Acronym | Transmission Control Protocol/Internet Protocol | User Datagram Protocol |
| Connection | TCP is a connection-oriented protocol. | UDP is a connectionless protocol. |
| Usage | TCP is suited for applications that require high reliability, and transmission time is relatively less critical. | UDP is suitable for applications that need fast, efficient transmission, such as games. UDP's stateless nature is also useful for servers that answer small queries from huge numbers of clients. |
| Weight | TCP is heavy-weight | UDP is light-weight. |
| Data Flow Control | TCP does Flow Control. | UDP does not have an option for flow control. |
| Error Checking | TCP does error checking and error recovery. Erroneous packets are retransmitted from the source to the destination. | UDP does error checking but simply discards erroneous packets. Error recovery is not attempted. |

1. **Hardware requirements:**
2. 1Raspberry Pi 3B.
3. 1 Flow Sensor.
4. 1 Water Level Sensor.
5. 1 Pi Camera Module.
6. 1 Breadboard.
7. 1 Power Supply 5V 2A (Micro USB Type-B Charger used for Mobiles).
8. Jumper cables (for connection).
9. **Review of Tasks:**

The current status is On-going. After doing some necessary study and research, we have arranged the main components and started working on the project. However, one of the components, Pi Camera Module, is not available with us at the moment which may lead to the delaying of the time taken for the project but will be completed on time.

1. **Short term plans:**
2. Collect the remaining components which will help in completing the project as early as possible.
3. Choose an internet protocol suitable for the project and write the code according to that.