IoT Based Information Collection System of Victims From Crash Sites

Submitted By

Md. Iqbal Hossain

ID: 2013-2-60-020

Rajib Ghose

ID: 2013-2-60-062

Supervised By

Dr. Ahmed Wasif Reza

Associate Professor,
Department of CSE, EWU.

A Thesis

Submitted in partial fulfillment of the requirements

for the degree of Bachelor of Science to

Computer Science and Engineering

Dhaka, Bangladesh December, 2017

DECLARATION

This thesis has been submitted to the department of Computer Science and Engineering, East West University in the partial fulfillment of the requirement for the degree of Bachelor of Science in Computer Science and Engineering by us under the supervision of Dr. Ahmed Wasif Reza, Associate Professor at Department of CSE at East West University under the course 'CSE 497'. We also declare that this thesis has not been submitted elsewhere for the requirement of any degree or any other purposes. This thesis complies with the regulations of this University and meets the accepted standards with respect to originality and quality. We hereby release this thesis to the public. We also authorize the University or other individuals to make copies of this thesis as needed for scholarly research.

____

Md. Iqbal Hossain
ID: 2013-2-60-020
Department of Computer Science and Engineering
East West University.

____

Rajib Ghose
ID: 2013-2-60-062
Department of Computer Science and Engineering
East West University.

LETTER OF ACCEPTANCE

The thesis entitled "IoT Base information collection system of victims from crash sites" Submitted by Md. Iqbal Hossain, ID 2013-2-60-020 & Rajib Ghose, ID 2013-2-60-062 to the department of Computer Science & Engineering, East West University, Dhaka 1212, Bangladesh is accepted as satisfactory for partial fulfillments for the degree of Bachelor of Science in Computer Science & Engineering in December 2017.

Board of Examiners

1________________________________

Dr. Ahmed Wasif Reza
Associate Professor Supervisor
Department of Computer Science and Engineering Supervisor
East West University, Dhaka, Bangladesh

2______________________________

Dr. Ahmed Wasif Reza
Associate Professor Supervisor
Department of Computer Science and Engineering Chairperson
East West University, Dhaka, Bangladesh

ABSTRACT

In this project, an IoT based information collection system of victims from crash sites is developed in order to minimize the death toll of the victims who are still trapped alive in the crash site through capturing images and sending the images continuously to cloud storage and also taking necessary information through various sensors and send the information to web server. The communication between the web server and the hardware is established via motion, sound, humidity and temperature sensors and another connection is established between hardware and cloud storage via camera. The project is developed for real time data fetching from hardware by sensors and camera and storing the information in web server and cloud storage which are used by the rescuers to have the exact information of victims in order to save their lives in time.

ACKNOWLEDGEMENTS

First of all, we are grateful to the Almighty Allah for establishing us to complete this research. We wish to express our sincere thanks and gratitude to our advisor Dr. Ahmed Wasif Reza, Associate Professor at Dept. of CSE for the continuous support during our thesis study and related research, for his patience, motivation, and immense knowledge. His guidance helped us in all the time of research and writing of the thesis. We will always be grateful for having the opportunity to study under him.

We are thankful to all of our teachers, Department of CSE, East West University. We also grateful to all of our primary and secondary school teachers who are our first teachers in our life and initiator of our basic knowledge.

We would like to express our thanks to our parents and siblings for supporting us spiritually throughout writing this thesis. And we are thankful to all my friends and colleagues. And at last we again thanks to the creator Allah for everything.

TABLE OF CONTENTS

Table of contents

LIST OF FIGURES

LIST OF ABBREVIATION

IOT	-	Internet of Thing
LAN	-	Local Area Networks
API	-	Application Programming Interface
HTTP	-	Hypertext Transfer Protocol
MQTT	-	Message Queue Telemetry Transport
CoAP	-	Constrained Application Protocol
SCADA	-	Supervisory control and data acquisition
GSM	-	Global System for Mobile communication
GPRS	-	General Packet Radio Services

LIST OF TABLES

CHAPTER 1
INTRODUCTION

1.1 Background

We are going through an enormous evolution of modern science and technology and science has evolved so much that vast number of previously unsolved problems has been brought under valid and logical explanation. IoT(Internet Of Things) is generally a huge network of devices or equipment connected to the internet including smartphones, wireless devices, tablets and anything with a sensor in it. When the Internet and networks expand to places such as manufacturing floors, energy grids, healthcare facilities, and transportation, cars, machines in production plants, jet engines, wearable devices and more to collect and exchange data, then that is called IOT. This technology can be fully automated without the interaction between human to human or human to machine. It has made its way into virtually every industry, agriculture, aviation, mining, healthcare, energy, transportation, smart cities, and on and on.

IoT is not only the future promising phase of internet, it’s just radically reshaping and strengthening the internet as we know of it.

With the advancement of modern technology, the numbers of high raised buildings, structures are being constructed. With the growing rate of constructing buildings, the number of accidents have increased like structural collapse, fire disaster, burst due to gas leakage etc. Natural catastrophic disasters like earthquake, cyclone, storm can also lead to tragic accidents, building collapse and hence serious injury and tragic loss of lives.

With The help of IoT, we propose a drone based human detection and rescue system can be developed which can't help in the tragic loss of death but can help finding the people who gets trapped under or inside the crash more significantly and fast.

1.2 Motivation

In a developing country like Bangladesh, where a large number of population is living in a relatively small area, where high raised building are being constructed radically day by day to accommodate the fast growing population, occurrence of accidents are more frequent. To meet with the necessity, more weakly-built buildings, apartments and structures are being built. As a result, collapse of buildings are becoming common. Also with the lack of proper knowledge and alertness, people are being the victim of frequent gas leakage and fire occurrence. We remember the tragic loss of countless lives in the accidents of Rana Plaza 24th April 2013 in Savar, where a five-story commercial building named Rana Plaza collapsed and considered as the most tragic and deadliest garment factory accident of the history. Approximately 1,134 people were dead and 2500 were injured. The rescue mission lasted for nearly a month because of the lack of proper rescue system and more people lost their lives due to not being rescued in time.

So, to minimize the losses of lives in accidents such like that, we came forward with the idea of drone based human detection and rescue system which will be more effective as the drone is much faster and can easily go to any place where it is much difficult to reach for a rescuer. The information collected by the drone will be immediately served to the authority handling the system and rescuers will have a proper lead to their mission.

1.3 Problem Statement

1. We are looking forward to establish an effective system where we can find an optimized solution to human detection and rescue from the crash sites in shortest possible time and more efficiently.
2. We tend to build a system that will help the rescue authority or firefighters who will be directly involved in the rescue mission of the people to make their work more easy and significantly fast.
3. We observe that there are no related works like that are done in our country so far and we want to contribute on that part. As we find a little help from any helpful sources because of the lack of proper information and data, though we want to build an effective system with all we have.

1.4 Objectives

1. To design an effective drone based victim detection and rescue system with the help of necessary equipment and sensors.

2. To develop a cloud based system where the information about the real time detection of the people will be stored immediately.

1.5 Thesis Organization

The followings are overviews of the contents of the chapters that are presented throughout in this research:

Chapter 1: This chapter describes the background, motivation, problem statements and objectives of this thesis paper. Why we have selected this topic, what encouraged us to go with the topic are discussed here.

Chapter 2: Here we provide an overview of the literature survey, introduction, revolution and architecture of IoT. It also provides the technique of building an IoT strategy,security of IoT and hardware specification of the system.

Chapter 3: In this part we have discussed our proposed methodology and design, block diagram and collection of information. We described the whole proposed system, how it works and also gave the algorithm related to this research. The step by step implementation of the hardware section of this research are also given.

Chapter 4: In this section, we have provided the experimental result and discussion of our research using web site and our developed hardware.

Chapter 5: In this portion we have given the overall conclusion and the summary of this thesis by visualizing and analyzing the whole paper and we have also provided some recommendations for further research and future works

CHAPTER 2
LITERATURE REVIEW

1.1 IoT : The revolution of Technology

The Internet of things (IoT) is the between systems administration of physical gadgets, vehicles, structures, and different things installed with hardware, programming, sensors, actuators, and system availability which empower these items to gather and exchange data. The IoT enables items to be detected or controlled remotely across existing system infrastructure creating open doors for more straightforward of the physical world into computer-based systems and resulting in improved efficiency, accuracy and economic benefit in addition to reduced human intervention. Only IoT can connect physical world to the web. The figure 2.1 shows how IoT have emerged over the years. Figure 2.2 shows the interaction between human and devices in the field of IoT.

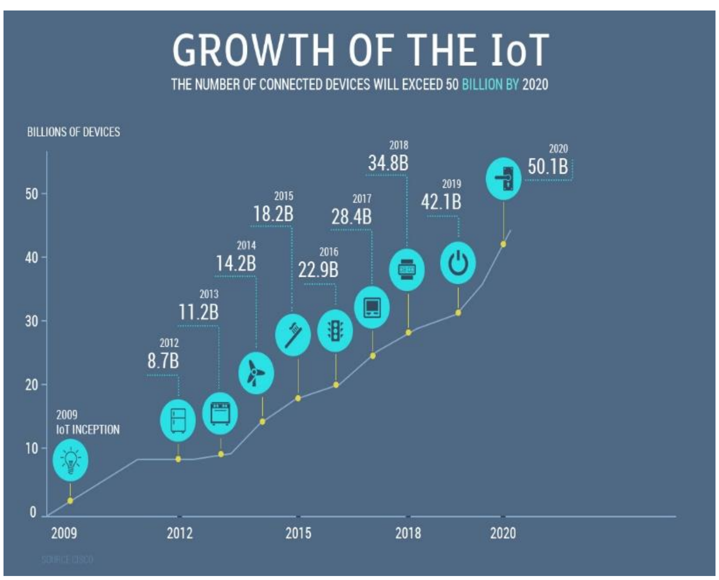

Figure 2.1: The growth of IoT

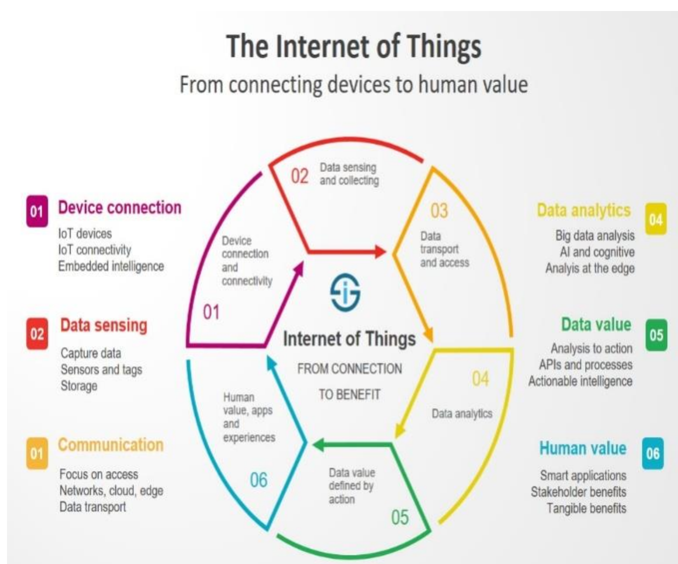

Figure 2.2: connecting devices to human value

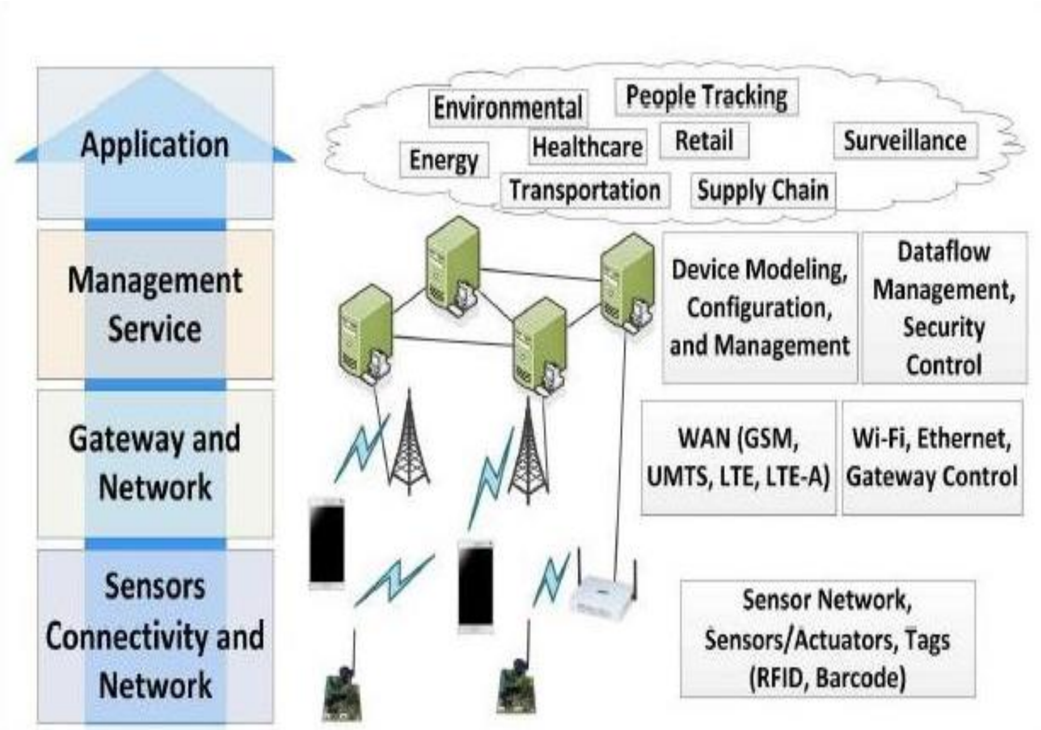

Figure 2.3: IoT Basic layer architecture

The IoT is more than internet connected consumer gadgets. Sooner or later every IT organization will need to create a framework to support it. Energy companies already use networked sensors to measure vibrations in turbines. They feed that data through the network to computing systems that analyze it to predict when machines will need maintenance and when they will fail. Jet engine manufacturers embed sensors that measure temperature, pressure, and other conditions to improve their products. Even a gift basket business can deploy sensors to constantly monitor the temperature of perishable products. Figure 2.3 shows the different layers of IoT architecture.

1.2 Architecture of IoT

To do any project or research about IoT having knowledge of architecture of IoT is required. There are four layers of IOT architecture. They are Interface layer, Service layer, Networking layer and Sensing layer.

1.2.1 **Application and Interface Layer**

The first layer of IoT is application and interface layer. This layer provides the interaction methods between users and application. This section looks how user can easily be used the system. This includes three main approaches. Firstly, we need the ability to create web-based front-ends and portals that interact with devices and with the event-processing layer. Secondly, we need the ability to create dashboards that offer views into analytics and 11 event processing. Finally, we need to be able to interact with systems outside this network using machine-to-machine communications (APIs).

The recommended approach to building the web front end is to utilize a modular front-end architecture, Web server-side technology, such as Java Servlets/, JSP, PHP, Python, Ruby, etc.

1.2.2 **Management and Service layer**

This layer is used to create and manage services to satisfy user needs. To do so, it process data deep processing. To make more user friendly application, it provides database with different data and divides work.

This is an important layer for three reasons:

1. The ability to support an HTTP server and/or an MQTT broker to talk to the devices;
2. The ability to aggregate and combine communications from different sensing devices and to route communications to a specific device (possibly via GSM/GPRS).
3. The ability to bridge and transform between different protocols that is to offer HTTP based APIs that are mediated into an MQTT message going to the device.

1.2.3 **Networking or Communication Layer**

The Networking or Communication layer supports the connectivity of the devices. There are multiple potential protocols for communication between the devices and the cloud. The most well-known three potential protocols are:

1. HTTP/HTTPS (and restful approaches on those)

2. MQTT 3.1/3.1.1

3. Constrained application protocol (CoAP)

Let's take a quick look at each of these protocols in turn **HTTP** is well known, and there are many libraries that support it. Because it is a simple text based protocol, many small devices such as 8-bit controllers can only partially support the protocol – for example enough code to POST or GET a resource. The larger 32-bit based devices can utilize full HTTP client libraries that properly implement the whole protocol.

There are several protocols optimized for IOT use. The two best known are MQTT6 and CoAP7.

MQTT was invented in 1999 to solve issues in embedded systems and SCADA. It has been through some iterations and the current version (3.1.1) is undergoing standardization in the OASIS MQTT Technical Committee8. MQTT is a publish-subscribe messaging system based on a broker model. The protocol has a very small overhead (as little as 2 bytes per message), and was designed to support loss and intermittently connected networks. MQTT was designed to flow over TCP. In addition there is an associated specification designed for ZigBee-style networks called MQTT-SN (Sensor Networks).

CoAPis a protocol from the IETF that is designed to provide a restful application protocol modeled on HTTP semantics, but with a much smaller footprint and a binary rather than a text-based approach. CoAP is a more traditional client-server approach rather than a brokered approach. CoAP is designed to be used over UDP [4].

1.2.4 **Sensor connectivity Layer**

Sensors collect data from the environment or object under measurement and turn it into useful data. This layer covers everything from legacy industrial devices to robotic camera systems, water-level detectors, air quality sensors, accelerometers, and heart rate monitors. And the scope of the IOT is expanding rapidly, thanks in part to low-power wireless sensor network technologies and Power over Ethernet, which enable devices on a wired LAN to operate without the need for an A/C power source.

1.3 IoT Platforms and Security

Even with the recent attention given to security for IoT devices, it can be easy to overlook the need for end-to-end security for an IoT platform. Every part of a platform should be analyzed for security prospects. From internet connections to the applications and devices to the transmitted and stored data, there is a potential for an attack vector. Without question, the single most important non-functional requirement of an IoT platform is that it offers robust security.[4]

2.4 Building an IoT strategy

Along with the unique new opportunities, new challenges also comes in notice. To maximize the possibilities that IoT presents some things should be checked for building IoT strategy. These are:

2.4.1 IoT Security

As companies collect data beyond traditional IT boundaries, IoT security measures will be critical. Some key considerations include being able to secure and monitor devices, encrypt sensitive data, and build risk mitigation into systems.

2.4.2 Streaming Data

IoT applications accumulate more data than traditional batch processing can manage. Having capabilities for streaming data continually is key to reliably feeding real-time business processes and extracting timely insights.

2.4.3 IoT Platform

An IoT platform makes it possible to develop, deploy, and manage IoT and M2M applications. Automate processes and network connections, store and manage sensor data, connect and control your devices, and analyze your data.

2.4.4 IoT Applications

"Next generation" IoT applications must be able to capture, collect, interpret, and act on vast amounts of information – detecting connectivity gaps, handling interruptions, and meeting specific business and industry requirements.

2.4.5 IoT Cloud

IoT cloud solutions provide affordable access to high-speed data networks – to significantly extend the reach and usability of your IoT applications. They can also offer data storage, processing, analysis, and remote device management.

2.4.6 IoT Data

IoT data management technologies ensure that you can collect the right data at the right time, even when connectivity is interrupted. Rely on in-memory systems to process massive data volumes generated by thousands of devices.

2.5 Hardware specification

2.5.1 Arduino UNO REV3 [5]

Figure 2.4 shows a typical Arduino Uno REV3.

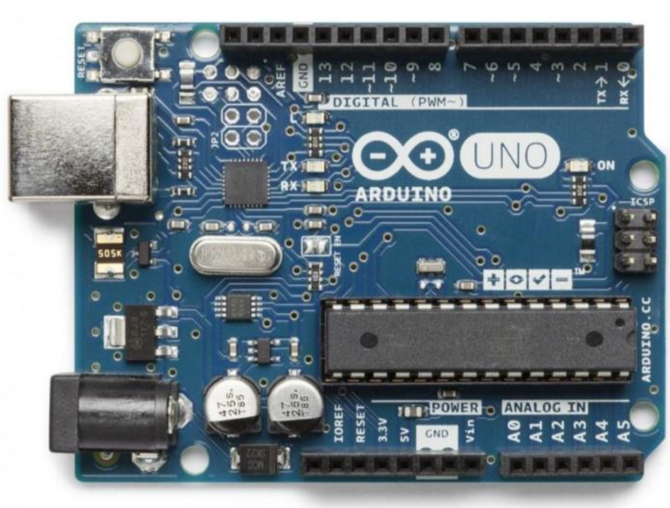

Figure 2.4: Arduino Uno

Microcontroller	ATmega328P
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
LED_BUILTIN	13
Length	68.6 mm
Width	53.4 mm

Table 1: **Arduino Uno Hardware specification** [5].

2.5.2 Sound Sensor

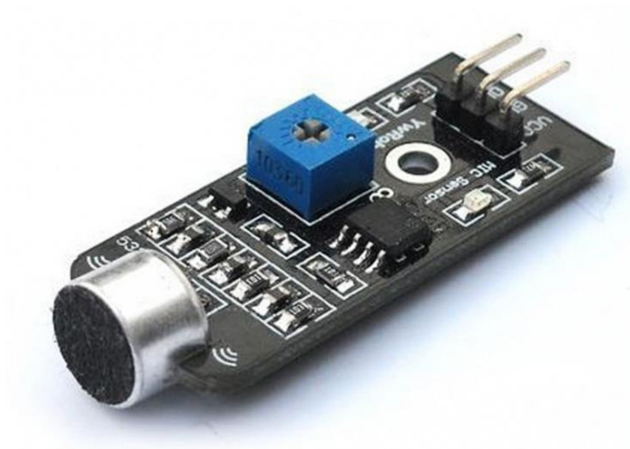

Figure 2.5: Sound Sensor

Figure 2.5 shows a typical Sound Sensor which senses sound from our surroundings. [6]

Overview

Onboard audio power amplifier LM386

Audio signal gain up to 200

Adjustable accuracy

Signal output indicator

Specifications

Mic sensitivity	52dB
Frequency range	50Hz ~ 20KHz
Mounting holes size	2.0mm

Table 2: **Hardware specifications of the sound sensor**

2.5.3 Motion sensor [7]

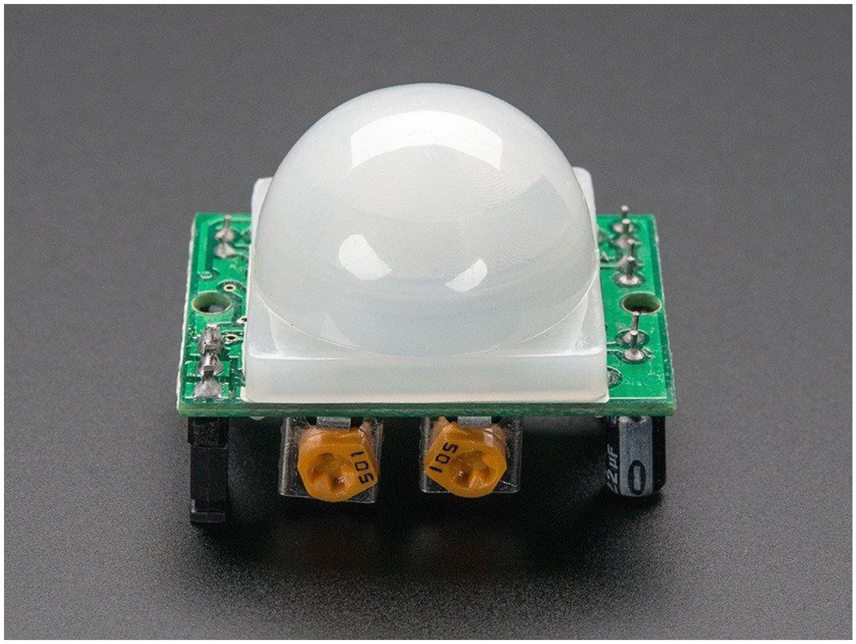

Figure 2.6: Motion Sensor

Description

Figure 2.6 represents a PIR motion sensor which detects motions of human or other living beings.

PIR sensors are used to detect motion from pets/humanoids from about 20 feet away. This one has an adjustable delay before firing (approx 2-4 seconds), adjustable sensitivity and we include a 1 foot (30 cm) cable with a socket so you can easily reposition the sensor or mount it using the two drills on either side runs on 5V-12V power (if you need to run it off of 3V you can do that by bypassing the regulator, but that means doing a bit of soldering). Digital signal output is 3.3V high/low. Sensing range is about 7 meters (120 degree cone).

2.5.4 Temperature and Humidity Sensor [8]

Figure 2.7: DHT-11 Sensor

Description

The figure 2.7 shows a DHT-11 sensor which senses both temperature and humidity.

The DHT11 temperature and humidity sensor features a calibrated digital signal output with the temperature and humidity sensor complex. Its technology ensures the high reliability and excellent long-term stability. A high-performance 8-bit microcontroller is connected. This sensor includes a resistive element and a sense of wet NTC temperature measuring devices. It has excellent quality, fast response, anti-interference ability and high cost performance advantages.

Each DHT11 sensors features extremely accurate calibration of humidity calibration chamber. The calibration coefficients stored in the OTP program memory, internal sensors detect signals in the process, we should call these calibration coefficients. The single-wire serial interface system is integrated to become quick and easy. Small size, low power, signal transmission distance up to 20 meters, making it a variety of applications and even the most demanding applications. The product is 4-pin single row pin package. Convenient connection, special packages can be provided according to users need. [8]

Specification

Supply Voltage	+5 V
Temperature range	0-50 °C error of ± 2 °C
Humidity	20-90% RH ± 5% RH error
Interface	Digital

Table 3: **Hardware specifications of the Temperature and Humidity sensor**

2.5.5 Raspberry Pi 3 [9]

Figure 2.8: Raspberry Pi 3

The above figure 2.8 features a Raspberry Pi 3 for which, specifications are given below:

The Raspberry Pi 3 Model Bis the latest version of the Raspberry Pi, which is a credit card size computer. It is operated on LINUX operating system. Just adding a keyboard, mouse, display, power supply, micro SD card with installed Linux distribution, a fully-fledged computer will be ready that can run applications from word processors and spreadsheets to games. This mini computer is very ideal for any kind of hardware related projects even robotics. [9]

Specifications

SoC	Broadcom BCM2837
CPU	4× ARM Cortex-A53, 1.2GHz
GPU	Broadcom VideoCore IV
RAM	1GB LPDDR2 (900 MHz)
Networking	10/100 Ethernet, 2.4GHz 802.11n wireless
Bluetooth	Bluetooth 4.1 Classic, Bluetooth Low Energy
Storage	microSD
GPIO	40-pin header, populated
Ports	HDMI, 3.5mm analogue audio-video jack, 4× USB 2.0, Ethernet, Camera Serial Interface (CSI), Display Serial Interface (DSI)

Table 4: **Hardware specifications of Raspberry Pi 3**

2.4.6 Raspberry Pi 3 Camera Module [10]

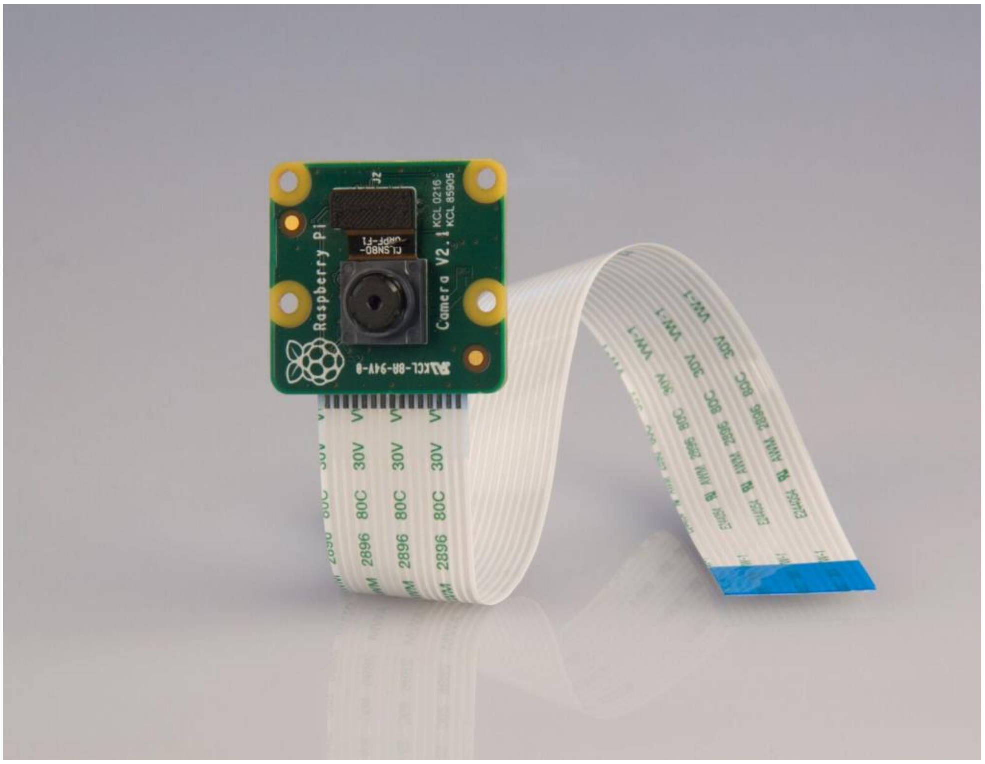

Figure 2.9: Raspberry Pi 3 Camera Module

The figure 2.9 represents a typical camera module specially made for Raspberry Pi

The Raspberry Pi Camera Module is an official product from the Raspberry Pi Foundation. The original 5-megapixel model was released in 2013, and an 8-megapixel Camera Module v2 was released in 2016. For both iterations, there are visible light and infrared versions. [10]

Hardware features

Available	Implemented
Frame rate up to 120 fps	Max 90fps. Limitations on frame size for the higher frame rates (VGA only for above 47fps)
AEC/AGC 16-zone size/position/weight control	No - done by ISP instead
Mirror and flip	Yes
Cropping	No - done by ISP instead (except 1080p mode)
Lens correction	No - done by ISP instead
Defective pixel cancelling	No - done by ISP instead
10-bit RAW RGB data	Yes - format conversions available via GPU
Support for LED and flash strobe mode	LED flash
Support for internal and external frame synchronisation for frame exposure mode	No
Support for 2 × 2 binning for better SNR in low light conditions	Anything output res below 1296 x 976 will use the 2 x 2 binned mode
Support for horizontal and vertical sub-sampling	Yes, via binning and skipping
On-chip phase lock loop (PLL)	Yes
Standard serial SCCB interface	Yes
Digital video port (DVP) parallel output interface	No
MIPI interface (two lanes)	Yes

Table 5: **Hardware specifications of Raspberry Pi Camera Module**

CHAPTER 3
DESIGN AND METHEDOLOGY

3.1 Decisions of design specification and hardware

The project is showcasing victim information collection from crash sites and the used hardware components are:

1. Arduino UNO REV3
2. Raspberry Pi 3
3. Sound sensor
4. Motion sensor
5. Temperature and humidity sensor
6. Breadboard
7. Power supply
8. USB cable
9. Router
10. LAN cable
11. Micro SD card as Raspberry Pi hard disk drive
12. Male-to-female jumper wires
13. Female-to-male jumper wires
14. Female-to-Female jumper wires
15. Male-to-Male jumper wires

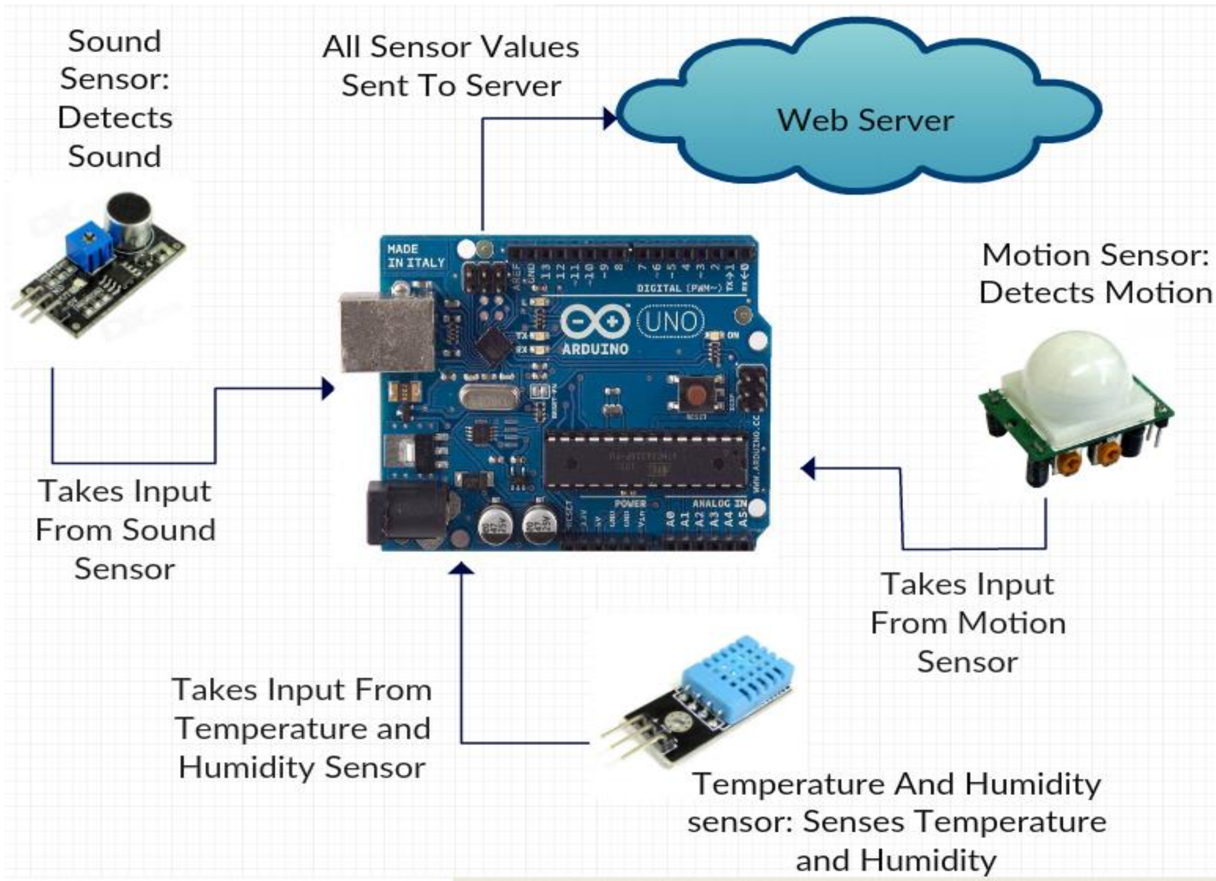

Figure 3.1: Arduino Uno block diagram of the project

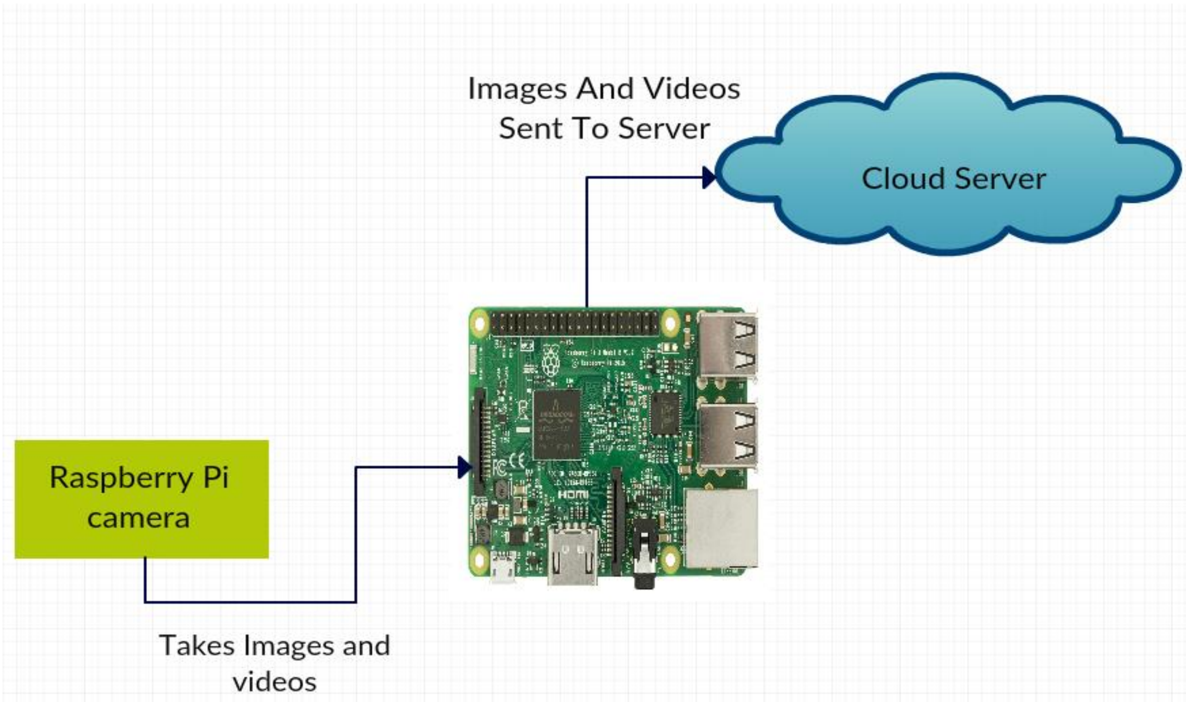

Figure 3.2: Raspberry Pi block diagram of the project

The figures 3.1 represents the block diagram of the overall working procedure for Arduino Uno, and figure 3.2 represents block diagram that explains the total working procedure for Raspberry Pi 3 of the project. An arduinouno, and a Raspberry Pi 3 modules are used to for the automation and controlling of the devices that's are: sound, motion, temperature and humidity sensors and a Raspberry Pi camera for capturing images and video recording the surrounding. The paper briefly explains how a victim can be detected and the information of the victim is collected through the respective sensors and sent immediately to the web servers which are being operated by any specialist of the rescue team and instantly take action on the basis of the collected information.

So, the whole methodology can be divided into two sections:

1. Collecting information which can be done by the sensors and camera
2. Sending information to the web servers or cloud storage.

3.3 Collecting Information

Our methodology works on the basis that, the Raspberry Pi camera constantly takes picture of the crash site or takes video streaming and continuously uploads the images to our cloud server. Observing the images, the rescuer operating the whole hardware system can detect whether there is any victim moving or not. Based on the information, the rescue team takes necessary action. If the imaging process is somehow disturbed, than there is alternative ways that are done by the sensors. When the camera is capturing pictures, the other sensors continuously takes surrounding information. The motion detector searches for motion, if detects any, the temperature and humidity sensors immediately takes information too. Analyzing the information taken by the sensors, necessary steps are taken to rescue victims if found any.

3.4 Sending information to the web servers or cloud storage

The gathered information are continuously uploaded to the web server and captured images are also uploaded to the cloud storage. The rescuer or admin who has accesses to the web server and the cloud storage, analyzes the information and gives necessary instruction the rescue team.

3.5 Hardware Connection

3.5.1 Arduino Uno connection with sensors:

Figure 3.3 represents the total hardware connection for the Arduino Uno including the sensors, breadboard, wires and a power supply.

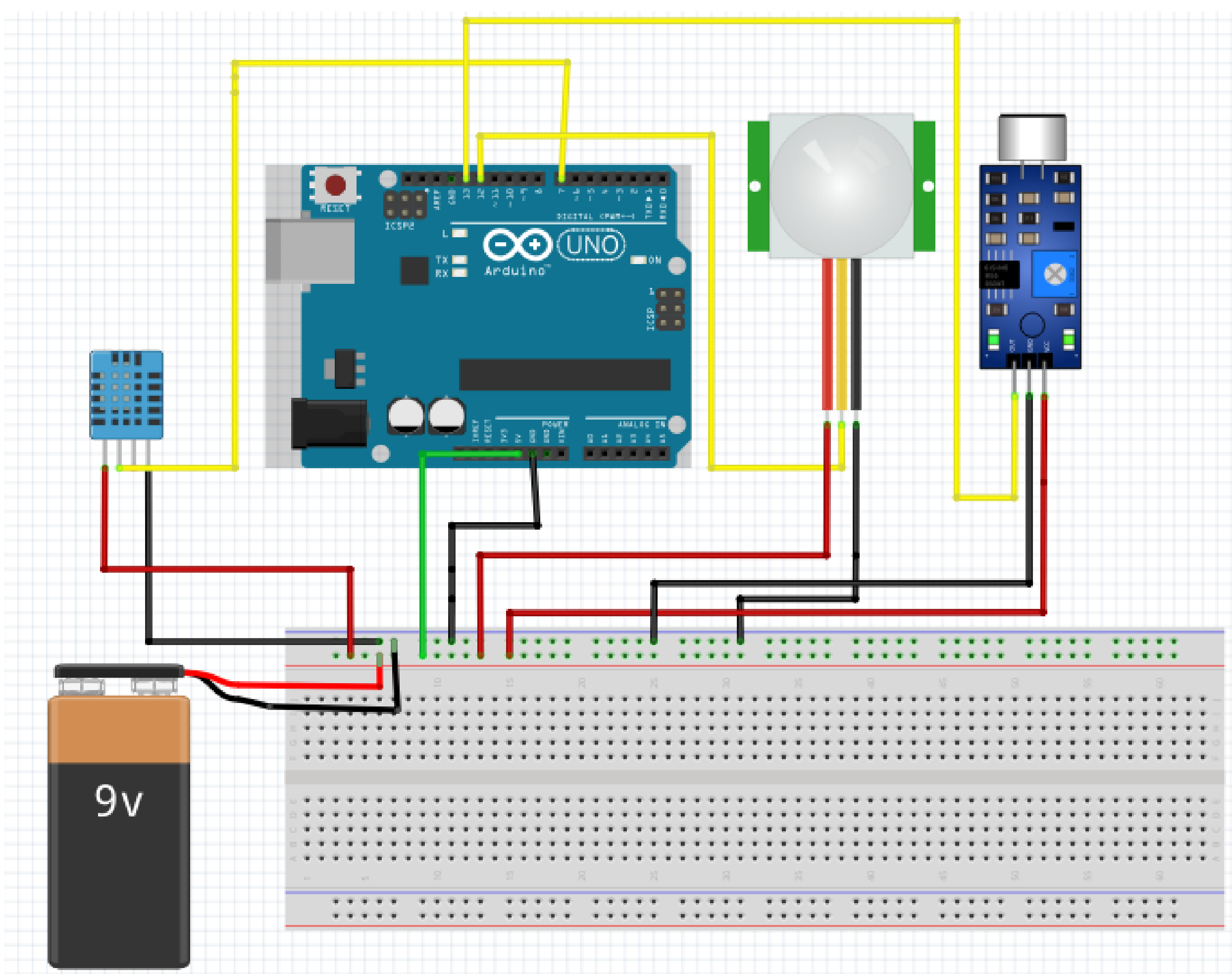

Figure 3.3: Arduino Uno with sensors

Schematic diagram

Figure 3.4 represents the schematic diagram that relates to the original hardware connection of the Arduino Uno.

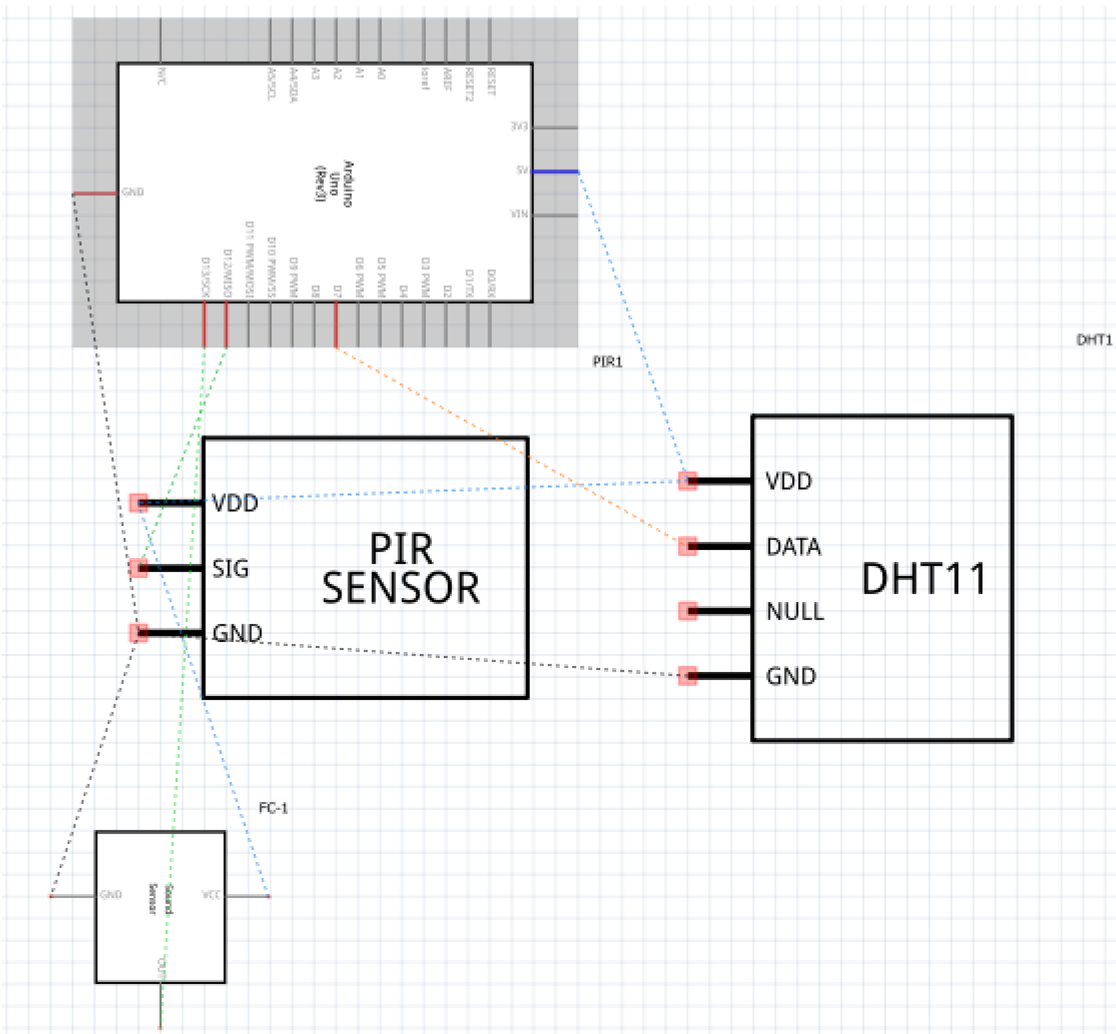

Figure 3.4: Arduino Uno schematic diagram

3.5.2 Raspberry Pi with Camera connection:

Figure 3.5: Raspberry Pi with camera connection

3.6 Algorithms for information collection

3.6.1 Arduino Uno Algorithms:

Algorithm sequence flow of Arduino Uno is given below:

step 1: Start Loop:

step 2: Read motion sensor data

step 3: If motion sensor value high set motion value = 1

step 4: Go to step 5 otherwise set motion, temperature and humidity value = 0 and go to step 2

step 5: Collect temperature and humidity value

step 6: Send information to web server

step 7: Read sound sensor data

step 8: If sound sensor value high set sound value = 1 and go to step 6

step 9: Otherwise set sound value = 0 and go to step 7

step 10: End of loop

3.6.2 Raspberry Pi algorithms

Algorithm sequence flow of Raspberry Pi is given below:

step 1: Set a variable1 as number of expected images

step 2: Set another variable2 and initialize it to zero

step 3: Start Loop:

step 4: While the variable2 value not equal to variable1 value

step 5: Start the camera and go to step 6

step 6: Capture images and send to cloud storage

step 7: Close camera

step 8: Increment the variable2 value to 1 and go to step 4

step 9: End of Loop

3.7 Implementation

3.7.1 Circuit Connection

3.7.1.1 For Arduino Uno

The project was developed by using motion, sound, temperature and humidity sensors directly connected to the different pins of Arduino. A common ground and a common voltage source is created in the breadboard. The DHT11 temperature and humidity sensor is connected to pin number 7 as output, common ground and common voltage source. The sound sensor is connected to pin number 2 for input and to number 3 as output. The motion sensor is connected to pin number 12 for input and to number 13 as output.

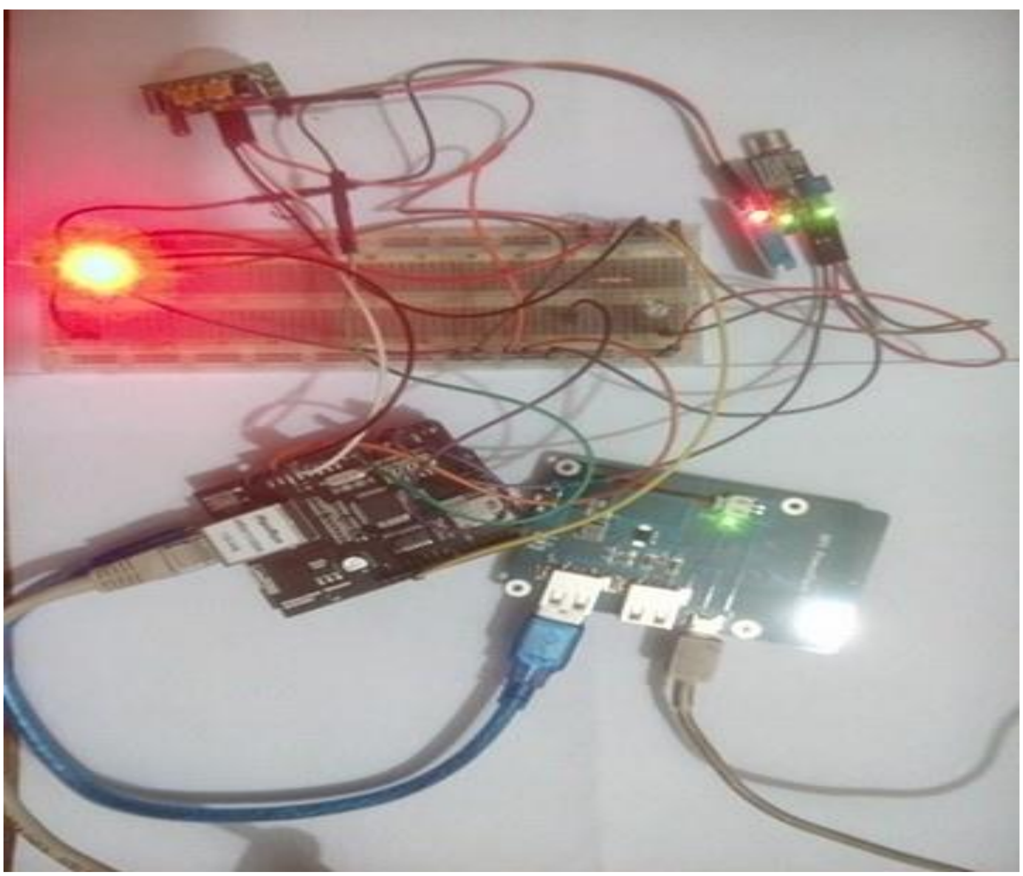

Figure 3.6: Connection of arduino Uno with sensors

3.7.1.2 Arduino Programming

Arduino IDE is used to write program for Arduino Uno Board and to upload the program to the board.

```
sound

int soundSensor = 2;
int LED = 3;

int ThermistorPin = 0;
int Vo;
float R1 = 10000;
float logR2, R2, T;
float c1 = 1.009249522e-03, c2 = 2.378405444e-04, c3 = 2.019202697e-07;

void setup()
{

  pinMode (12, INPUT);
  pinMode (13, OUTPUT);
  pinMode (soundSensor, INPUT);
  pinMode (LED, OUTPUT);
  Serial.begin(9600);
}

void loop()
{



  //for motion sensor
  if (digitalRead(12) == HIGH)
  {
    //Serial.println( "motion detected");
    digitalWrite(13, HIGH);

Compiling sketch...
```

Figure 3.7: Arduino Uno code implementation

3.7.1.3 For Raspberry Pi

The camera component was connected to the Raspberry Pi camera port. A monitor, 3.7 volt battery, mouse, keyboard and LAN cable for internet connection are directly connected to the respective ports of the Raspberry Pi.

Figure 3.8: Connection raspberry pi with camera

3.7.1.4 Raspberry Pi Programming

Python 3 IDE is used to write program for Raspberry Pi Board.

```
from subprocess import call
i=0

while i!=4:
    try:#We put everything inside a try to avoid running if the camera doesn't work
        import picamera
        import datetime



        timestamp=datetime.datetime.now().strftime("%Y-%m-%d_%H-%M-%S")
        camera = picamera.PiCamera()

        try:
            camera.resolution=(2592,1944)
            camera.rotation=360
            camera.start_preview()
            camera.capture(timestamp+".jpg")
        except:
            pass
        finally:
            camera.close()

        photofile = "/home/pi/Dropbox-Uploader/dropbox_uploader.sh upload /home/pi/"+timestamp+".jpg "+timestamp+".jpg "
        call([photofile], shell=True)


    except:
        pass
    i=i+1
```

Figure 3.9: Raspberry pi camera code implementation

CHAPTER 4

RESULT AND DESCUSSION

4.1 Result and Analysis for Arduino Uno

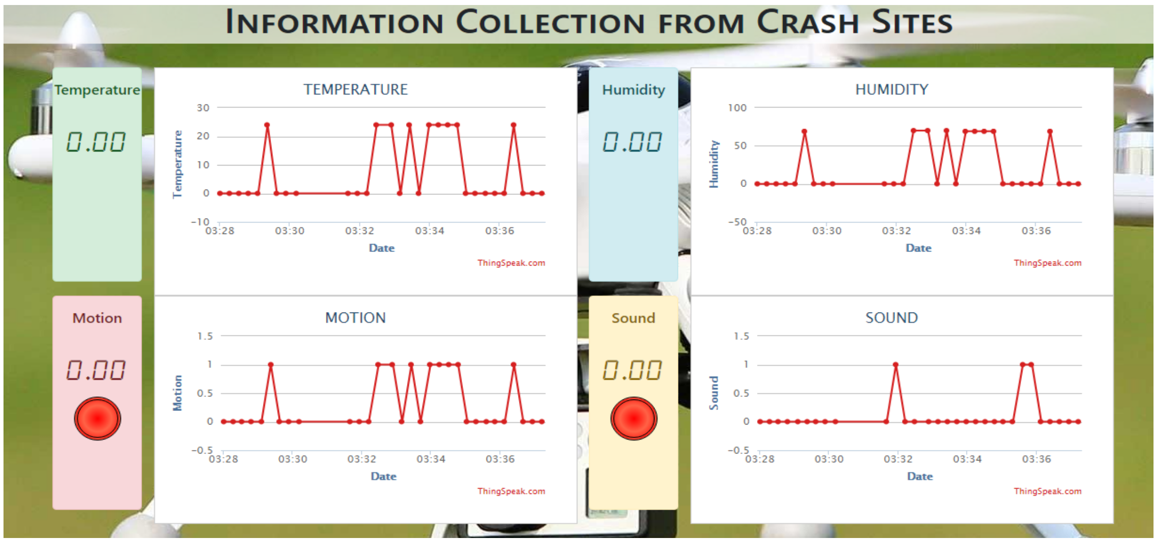

Figure 4.1: At Initial Stage

Figure 4.2: Sound Detected

Figure 4.3: Motion Detected

Figure 4.4: Sound and Motion Detected

4.2 Result and Analysis for Raspberry Pi

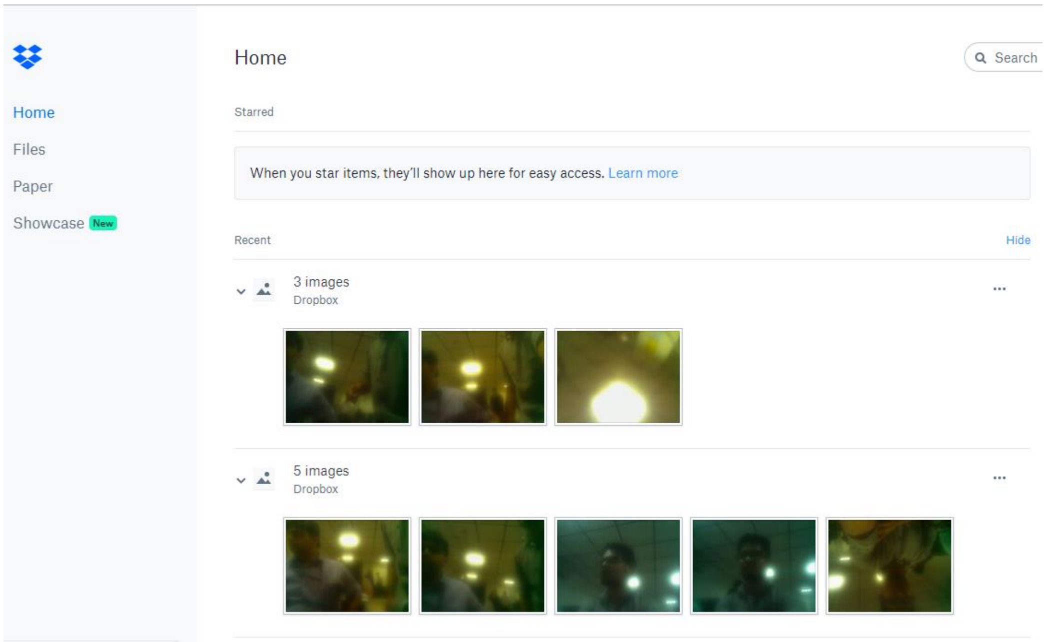

Figure 4.5: Captured images stored to Dropbox cloud storage

CHAPTER 5
CONCLUSION

5.1 Overall conclusion

The IoT based information collection of the victims from the crash sites system is successfully implemented using all the necessary sensors and camera to fetch real time data from the crash sites and continuously uploading them to the servers and cloud storage. Any authority can use this system to collect real time information of the alive victims who are trapped in any crash sites happened due to building collapse, fire break out or natural calamities. Any rescue team having this system can continuously have information about the trapped victims and can immediately response and send special support and help to rescue the victims in very short time. Web based real time data visualization and image analyzing makes this project more convenient and user friendly.

5.2 Future Work

It has been observed that this system can collect information successfully by sensors and camera and can communicate with web server and cloud storage so far. The system can be improved in case of widely use by using more powerful sensors and camera component. In case of Bangladesh, occurrence of such tragic accidents are frequent as we have more weak structural buildings and fire break down incidence. By upgrading the system, any fire brigade service or police authority can use this system for more accurate real time information collection to save countless number of innocent victims' lives.

REFERENCE

[1] Qiang Ye, Weihua Zhuang, " Distributed and Adaptive Medium Access Control for Internet-of-Things-Enabled Mobile Networks" IEEE Internet of Things Journal, Volume 4 Issue 2, April 2017

[2] J. Gubbi, R. Buyya, S. Marusic, M. Palaniswami, "Internet of Things (IoT): A vision architectural elements and future directions", *Future Gener. Comput. Syst.*, vol. 29, no. 7, pp. 1645-1660, 2013.

[3] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, M. Ayyash, "Internet of Things: A survey on enabling technologies protocols and applications", *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2347-2376, Dec. 2015.

[4] Internet of Things (IoT) - Part 2 (Building Blocks & Architecture)[online]. Available: http://www.c-sharpcorner.com/UploadFile/f88748/internet-of-things-part-2/.[Accessed: 30-Nov-2017]

[5] Arduino Uno Board."Cite a Website - Cite This For Me", Arduino.org, 2017. [Online]. Available:http://www.arduino.org/products/boards/arduino-uno. [Accessed: 30-Nov-2017]

[6] Sound Sensor,[online] .Available: https://www.waveshare.com/sound-sensor.htm.[Accessed: 30-Nov-2017]

[7] PIR (MOTION) SENSOR.[online] Available: https://www.adafruit.com/product/189. [Accessed: 30-Nov-2017]

[8] DHT11 Temperature and humidity sensor.[online] .Available: https://www.dfrobot.com/wiki/index.php/ DHT11_Temperature_and_Humidity_Sensor_(SKU:_DFR0067) [Accessed: 30-Nov-2017]

[9] Raspberry Pi 3.[online].Available: https://www.raspberrypi.org/magpi/raspberry-pi-3-specs-benchmarks/[Accessed: 30-Nov-2017]

[10] Raspberry Pi camera module.[online].Available: https://www.raspberrypi.org/documentation/hardware/camera/