**Firefighter Automation**

by

Tasaporn Kittipongpysan 5911867

Simon Fehrmann 5911329

Benjarat Soysangtong 5925402

A report submitted in partial fulfillment of the requirements for

the degree of Bachelor of Engineering in

**Mechatronics Engineering**

Project Advisor:

**Asst. Prof. Dr. Narong Aphiratsakun**

Examination Committee:

Dr. Jerapong Rojanarowan, Dr. Wisuwat Plodpradista,

Assoc. Prof. Dr. Jiradech Kongthon, Mr. Sunchanan Charanyananda,

Mr. Amulya Bhattarai, Mr. Ehsan Ali

Assumption University (12)

Vincent Mary School of Engineering

Thailand

October 2020

Approved by Project Advisor:

Name: Asst.Prof.Dr.Narong Aphiratsakun

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Plagiarism verified by:

Name: Mr. Ehsan Ali

Signature:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Date:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Table of Contents

**Abstract**

Nowadays, when there is a fire incident, the firemen and rescue service members will be sent as a team to extinguish fire and to rescue and protect people trapped in the calamity. The most vulnerable team is the fire-rescue team that have to get into the fire area to find and rescue victims in buildings or in emergency situations.

This project was designed to help the fire-rescue team (within the area) to automatically spray water to the burning site. This robot will capture and use sensors to detect fires, then command a water spray to extinguish the fire efficiently. They can monitor the fire extinguishing process of the machine and to send real-time image to a mobile application with the capability to be controlled by the user in a vary convenient manner.

**1.Introduction**

**1.1 Introduction of Topic**

Nowadays, automation in the industrial workplace plays a significant role in the world and it provides the advantages as increasing safety, reducing error, better product quality, and profitability.

In our project, we are going to do firefighter automation in which we apply some automation technology to automatically detect fire and the user will be able to monitor through the mobile application.

**1.2 Project Idea**

As we see from news in the last few months, the increase of climate change creating more intense and more frequent wildfires where massive burns from Australia, the Amazon and California in 2019. Many people and animals were injured and killed. The wildlife is extended every year and getting more dangerous to living things and human.

Wildfire firefighters/ firefighters have to face dangerous situations and get into the fire area. They were required to work long hours in challenging and changing conditions; high temperature and steep terrain that increase their risk of on-the-job death and injury.

This project “firefighter automation” was designed to help firefighters to detect and suppress the fire, where it is not able to extinguish the wildfire or burning.

**2.Project Overview**

**2.1 Initial Study** & **Background Study**

In this project, we choose the Raspberry Pi which is a minicomputer to be our controller for receiving the data from camera and sensor and controlling servo and pump. We use Python to be our programming language.

We must set the machine to be in the proper distance between the burning object and machine where it can precisely spray the water.

**2.2 Theory**

Raspberry Pi is a minicomputer that it can run multiple programs at a time. There is a huge range of "HATs" (Hardware Attached on-Top) and other accessories which you can connect to the Raspberry Pi and add specific features such as: Cameras, LCD displays, Motor drivers, Sensors, GPS, Mobile data connection, Digital TV decoders, etc. So, we use this controller because we use many devices to communicate together such as camera, temperature sensor, distance sensor, servo, and pump. Moreover, we want to show the real time data and real time video on python platform.

Arduino is a microcontroller which is a part of the computer that it run only one program. So, it’s not suitable to use in our project and Raspberry Pi is faster than Arduino by 40 times in clock speed.

To detect the fire, we use the knowledge of image processing that there are 2 concepts to determine the fire which are “Fire Detection” and “Smoke Detection”.

Then, we compare and analyze these 2 data that is it fire or not.

**3.** **Methodology**

**3.1 Method of Implementation**

This project will use the robot that is able to detect the fire and respond to the fire (capable of extinguishing buildings) and also able to use smartphone applications to receive the information with real time imaging.

The sub-main objective as following

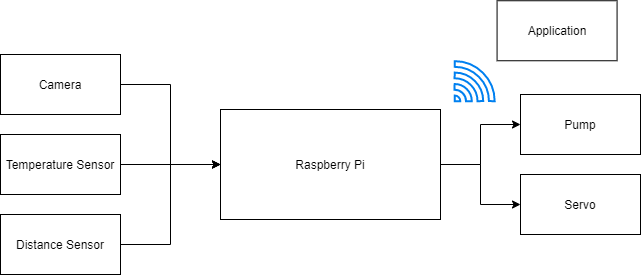
-To reduce the risk of dangerous work in dangerous situations.

-To reduce human labor.

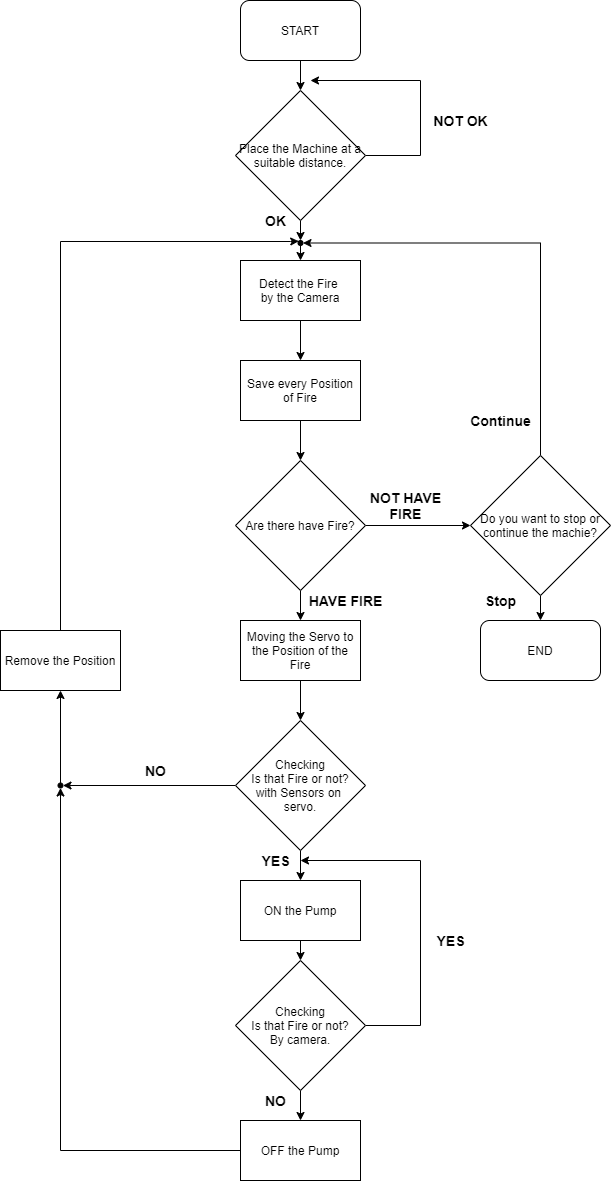
-To increase safety to human.

**3.2 Project System**

**3.2.1. Block Diagram**



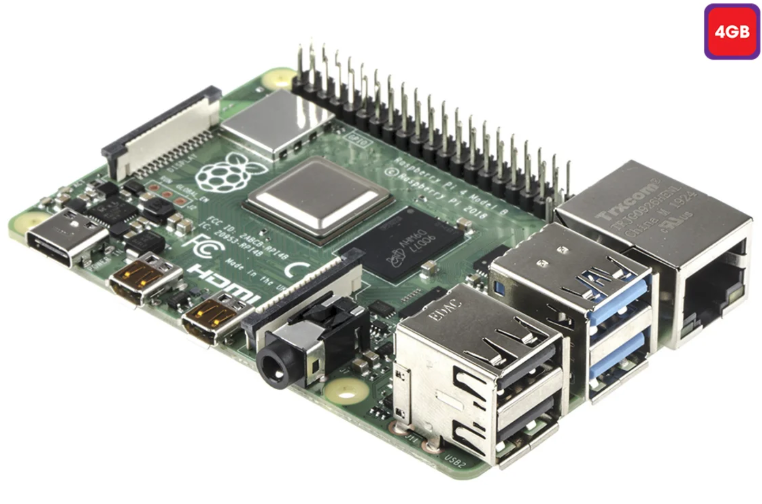
**3.2.2. Flow Chart**



**4.Application:**

**4.1 Controller**

Raspberry Pi 4 4G Model B.



### **Raspberry Pi 4 Model B**

### The Raspberry Pi 4 Model B is the latest computer board from Raspberry Pi. This latest version offers more processing power, more memory and more connectivity while being backward compatible with the projects created on previous models.

**Processor:** Broadcom BCM2711 Processor Quad core A72 (ARM v8) 64-bit SoC

**Memory (depending on model):** 2GB LPDDR4 SDRAM or 4GB LPDDR4 SDRAM

**Bluetooth:** Bluetooth 5.0

**WiFi:** 2.4GHz/5.0GHz IEEE 802.11ac

**Ethernet:** Gigabit Ethernet

**USB:** 2 x USB 2.0 ports, 2 x USB 3.0 ports

**Connection:** 40-pin GPIO Header

**HDMI:** 2 x micro HDMI ports (up to 4Kp60 supported)

**Video:** 2-lane MIPI DSI display port, 2-lane MIPI CSI camera port

**Audio:** 4 Pole stereo output and composite video port

**Multimedia:** H.265 (4Kp60 decode), H.264 (1080p60 decode, 1080p30 encode. OpenGL ES 3.0 graphics.

**Storage:** microSD card slot for loading operating system and data storage

**Input power:** 5v DC via USB-C Connector (min 3A), 5v DC via GPIO header. POE enabled (PoE HAT required)

**Operating Temperature:** 0 to 50°C

4.2 Sensors

4.2.1 Raspberry Pi Camera Module V2



The Raspberry Pi Camera v2 is a high quality 8 megapixel Sony IMX219 image sensor custom designed add-on board for Raspberry Pi, featuring a fixed focus lens. It's capable of 3280 x 2464 pixel static images, and also supports 1080p30, 720p60 and 640x480p60/90 video. It attaches to Pi by way of one of the small sockets on the board upper surface and uses the dedicated CSi interface, designed especially for interfacing to cameras. The board itself is tiny, at around 25mm x 23mm x 9mm. It also weighs just over 3g, making it perfect for mobile or other applications where size and weight are important. It connects to Raspberry Pi by way of a short ribbon cable. The high quality Sony IMX219 image sensor itself has a native resolution of 8 megapixel, and has a fixed focus lens on-board. In terms of still images, the camera is capable of 3280 x 2464 pixel static images, and also supports 1080p30, 720p60 and 640x480p90 video.

### **Product Information**

* 8 megapixel camera capable of taking photographs of 3280 x 2464 pixels
* Capture video at 1080p30, 720p60 and 640x480p90 resolutions
* All software is supported within the latest version of Raspbian Operating System
* Applications: CCTV security camera, motion detection, time lapse photography
* For Use With: Compatible with all Models of Raspberry Pi 1, 2 & 3
* Kit Contents: Raspberry Pi Camera Board, Ribbon Cable

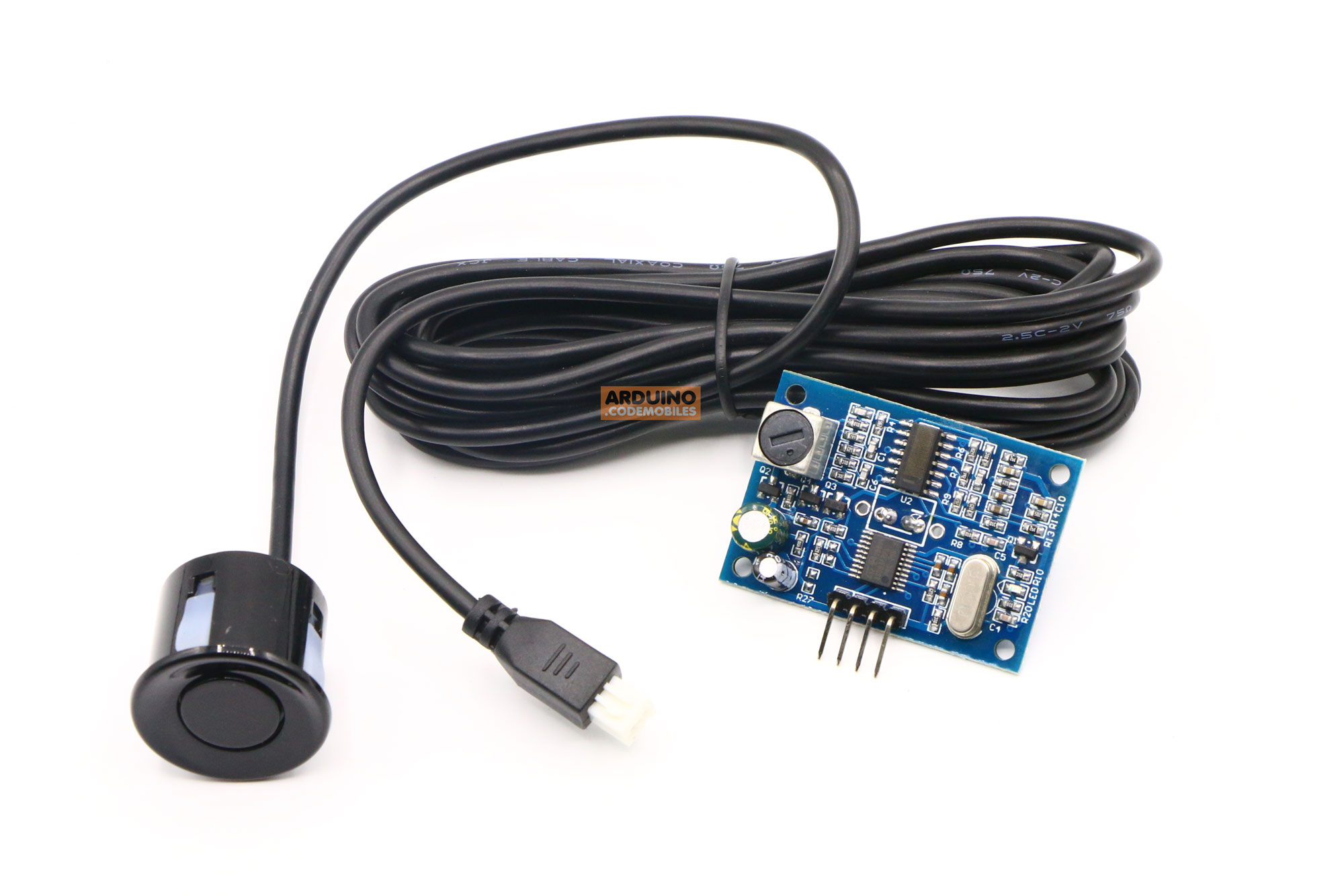
**High Quality**

* Capture video at 1080p30, 720p60 and 640x480p90 resolutions
* All software is supported within the latest version of Raspbian Operating System
* 1.12 µm X 1.12 µm pixel with OmniBSI technology for high performance (high sensitivity, low crosstalk, low noise)
* Optical size of 1/4"

4.2.2 Distance Sensor.

Operating Voltage DC: 3.0-5.5v

Working current less than: 8mA

Probe frequency: 40kHz

Farthest range: 600cm

Recent range: 20cm

Distance accuracy: 1cm

Resolution: 1mm

Measurement angle: 75 degrees

Output echo signal: Output pulse width signal, or TTL

Wiring 3-5.5V ( positive power supply )  
 Trig ( control terminal ) RX  
 Echo (output) TX  
 GND (negative power supply

Product Size: L42 \* W29 \* H12 mm

Operating temperature: -20C +70C

**4.3 Driver and Motor**

 4.3.1 Servo.

SPT SPT5325LV-360 25KG Digital Servo 360°

Neutral Position: 1500&mu;s/50hz

Motor: Core Motor

500-2500&mu;s: 360 Continuous Rotation

Voltage Range: 4.8V-7.2V

Operating Speed(4.8V): 44rpm

Operating Speed(6.0V): 55rpm

Operating Speed(7.2V): 66rpm

Dimensions: 40.5X20X40.5mm

Weight: 57g

Connector Wire Length: JR 260 mm

**Coding:**

Extinguisher Sensor

Can measure 120 degree.

#include <Wire.h>  
#include <LiquidCrystal\_I2C.h>  
LiquidCrystal\_I2C lcd(0x27, 16, 2);  
  
int sensor[5] = {0, 0, 0, 0, 0};  
String my\_sensor ;  
int buzzer = 8;  
  
void setup ( ) {  
  
  lcd.begin();  
  lcd.backlight();  
  pinMode(3, INPUT);  
  pinMode(4, INPUT);  
  pinMode(5, INPUT);  
  pinMode(6, INPUT);  
  pinMode(7, INPUT);  
  pinMode(buzzer, OUTPUT);  
}  
  
  
  
void loop ( ) {  
  
  sensor[0] = digitalRead(7);  
  sensor[1] = digitalRead(6);  
  sensor[2] = digitalRead(5);  
  sensor[3] = digitalRead(4);  
  sensor[4] = digitalRead(3);  
  
  my\_sensor =  "S = " + String(sensor[0]) + "," + String(sensor[1]) + "," + String(sensor[2]) + "," + String(sensor[3]) + "," + String(sensor[4]) + ",";  
  
  if ((sensor[0] == 0) && (sensor[1] == 0) && (sensor[2] == 0) && (sensor[3] == 0) && (sensor[4] == 0)) {  
  
    lcd.setCursor(0, 0);  
    lcd.print(my\_sensor);  
    lcd.setCursor(0, 1);  
    lcd.print("NORMAL");  
  }  
  else {  
  
    lcd.setCursor(0, 0);  
    lcd.print(my\_sensor);  
    lcd.setCursor(0, 1);  
    lcd.print("FIRE ALARM");  
    buzzer\_on();  
  }  
  
  delay(10);  
  
}  
  
void buzzer\_on()  
  
{ unsigned char i, j;  
  
  { for (i = 0; i < 80; i++)  
  
    { digitalWrite(buzzer, HIGH);  
  
      delay(1);//delay1ms  
      digitalWrite(buzzer, LOW);  
      delay(1);//ms delay  
  
    }  
  
    for (i = 0; i < 100; i++)  
  
    {  
  
      digitalWrite(buzzer, HIGH);  
      digitalWrite(buzzer, LOW);  
      delay(2);  
  
    }  
  
  }  
  
}

SERVO Code:

There are 3 pin that there are Voltage 5V(+5V), Ground(GND) and pin to control(Pin9)

#include <Servo.h>

Servo myservo;

void setup()

{

myservo.attach(9); // Servo To declare pin 9 to be control

}

void loop()

{

myservo.write(0);

delay(1000);

myservo.write(90);

delay(1000);

myservo.write(180);

delay(1000);

}

**A picture containing tool, table, holding, wooden

Description automatically generated**