

Individual Contribution

Wen Yuen Chao (004772951)

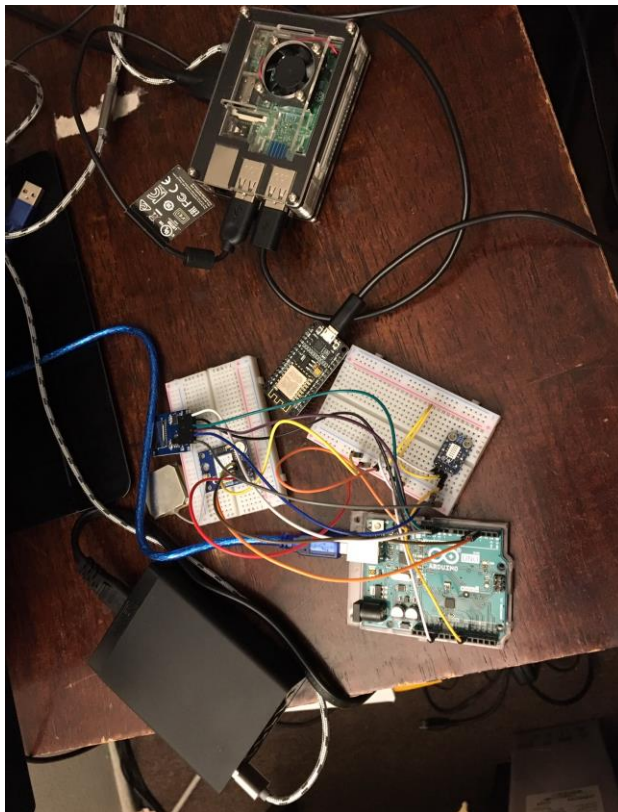
● Project Contribution

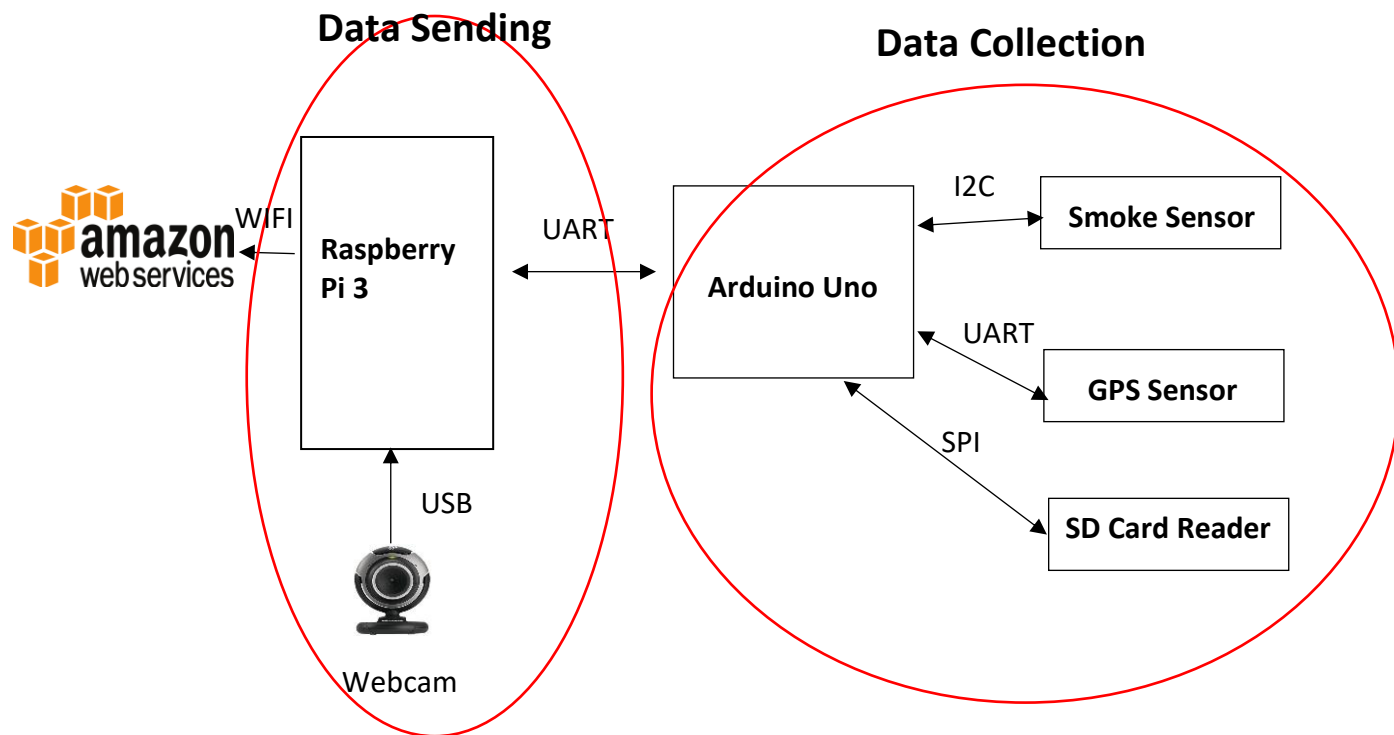
My main responsibility is in the hardware part. First, I built an fire alert platform, which not only can collect data, buy also monitor and detect the fire status. Second, after collecting the enough data, I will send it to server for analyzing via HTTP. Third, I set up a webcam, which can monitor the situation of fire every 30 seconds by using bash scripting. Once, the webcam detects the fire, it will send an alert to server via HTTP.

● Modules

- I. Raspberry Pi 3 model B
- II. Arduino Uno
- III. Smoke Sensor (Adafruit MiCS5524)
- IV. GPS (NEO-6M)
- V. SD card (SDHC Mini TF Card Adapter)
- VI. WIFI module (ESP8266)

● Implements

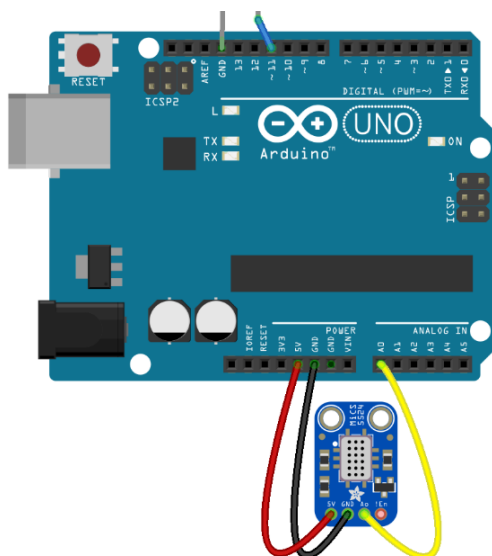




About the architecture of the system, I separate it as two parts, data collection and data sending. In the collection part, it basically is finished by Arduino. To explain, smoke sensor helps us to detect the concentration of CO and Methane. For GPS sensor, once we detect the fire, GPS can help us locate the location. Last, SD card can help us save the data, so we can analyze these data in the backend, which is very convenient. Moreover, webcam can help us take pictures and do fire detection test. Second part is Data Sending. Raspberry Pi 3 provides us WIFI, so we can send the data to server easily. Also, Raspberry Pi provides a IDE for Arduino, so I can write my code in this platform. For reducing the load of processor, we receive the data every 30 seconds and send to the server. We have different kind of data, like image and number data, which is a very big data stream.

● Tutorials

I. Smoke Sensor (Adafruit MiCS5524)



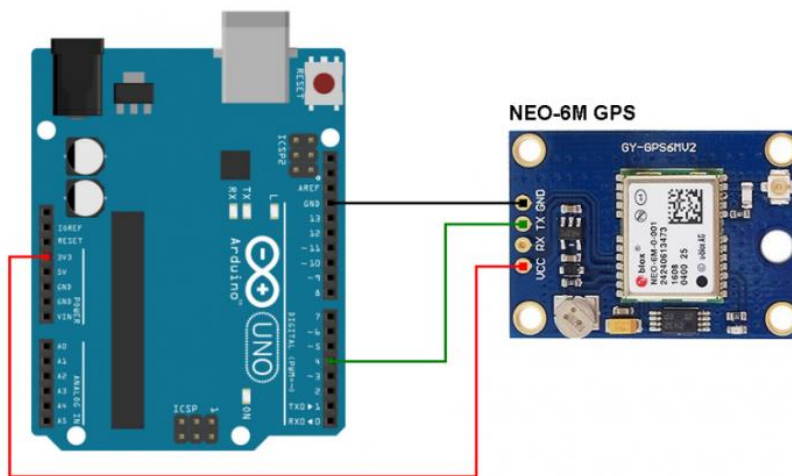
<Source Code>

```
#define LED 11
```

```
void setup() {  
  Serial.begin(115200);  
  Serial.println("MiCS-5524 demo!");  
  pinMode(LED, OUTPUT);  
}
```

```
void loop() {  
  int reading = analogRead(A0);  
  Serial.println(reading);  
  
  analogWrite(LED, reading);  
  delay(10);  
}
```

II. GPS sensor



NEO-6M GPS board GND pin goes to Arduino GND
TX pin goes to Arduino digital pin 4
RX pin is not connected
VCC pin goes to Arduino 5V pin (can be connected to 3.3V pin)

<Source Code>

```
void loop() {  
  
  while (SoftSerial.available() > 0) {  
  
    if (gps.encode(SoftSerial.read())) {
```

```

if (gps.location.isValid()) {
    Serial.print("Latitude = ");
    Serial.println(gps.location.lat(), 6);
    Serial.print("Longitude = ");
    Serial.println(gps.location.lng(), 6);
}
else
    Serial.println("Location Invalid");

if (gps.altitude.isValid()) {
    Serial.print("Altitude = ");
    Serial.print(gps.altitude.meters());
    Serial.println(" meters");
}
else
    Serial.println("Altitude Invalid");

if (gps.speed.isValid()) {
    Serial.print("Speed = ");
    Serial.print(gps.speed.kmph());
    Serial.println(" kmph");
}
else
    Serial.println("Speed Invalid");

if (gps.time.isValid()) {
    Serial.print("Time (GMT) : ");
    if(gps.time.hour() < 10) Serial.print("0");
    Serial.print(gps.time.hour());
    Serial.print(":");
    if(gps.time.minute() < 10) Serial.print("0");
    Serial.print(gps.time.minute());
    Serial.print(":");
    if(gps.time.second() < 10) Serial.print("0");
    Serial.println(gps.time.second());
}
else
    Serial.println("Time Invalid");

if (gps.date.isValid()) {
    Serial.print("Date : ");
    if(gps.date.day() < 10) Serial.print("0");
    Serial.print(gps.date.day());
    Serial.print("/");
    if(gps.date.month() < 10) Serial.print("0");
    Serial.print(gps.date.month());

```

```

        Serial.print("/");
        Serial.println(gps.date.year());
    }
    else
        Serial.println("Date Invalid");

    if (gps.satellites.isValid()) {
        Serial.print("Satellites = ");
        Serial.println(gps.satellites.value());
    }
    else
        Serial.println("Satellites Invalid");

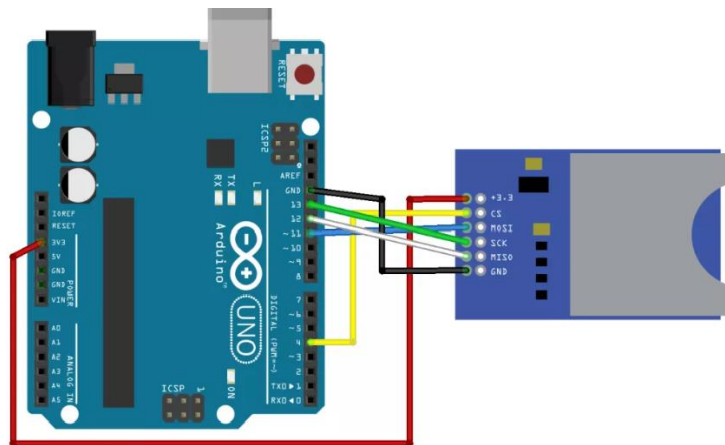
}

}

}

```

III. SD card reader



<Source Code>

```

#include <SPI.h>
#include <SD.h>

```

File myFile;

```

void setup() {
    // Open serial communications and wait for port to open:
    Serial.begin(9600);
    while (!Serial) {
        ; // wait for serial port to connect. Needed for native USB port only
    }
}

```

```
}
```

```
Serial.print("Initializing SD card...");
```

```
if (!SD.begin(4)) {  
  Serial.println("initialization failed!");  
  while (1);  
}  
Serial.println("initialization done.");
```

```
if (SD.exists("example.txt")) {  
  Serial.println("example.txt exists.");  
} else {  
  Serial.println("example.txt doesn't exist.");  
}
```

```
// open a new file and immediately close it:  
Serial.println("Creating example.txt...");  
myFile = SD.open("example.txt", FILE_WRITE);  
myFile.close();
```

```
// Check to see if the file exists:  
if (SD.exists("example.txt")) {  
  Serial.println("example.txt exists.");  
} else {  
  Serial.println("example.txt doesn't exist.");  
}
```

```
// delete the file:  
Serial.println("Removing example.txt...");  
SD.remove("example.txt");
```

```
if (SD.exists("example.txt")) {  
  Serial.println("example.txt exists.");  
} else {  
  Serial.println("example.txt doesn't exist.");  
}  
}
```

```
void loop() {  
  // nothing happens after setup finishes.  
}
```

● Reference

1. <https://learn.adafruit.com/adafruit-mics5524-gas-sensor-breakout/usage>
2. <https://simple-circuit.com/arduino-neo-6m-gps-module/>

3. <https://randomnerdtutorials.com/guide-to-sd-card-module-with-arduino/>
4. <https://github.com/arduino-libraries/SD/blob/master/examples/Files/Files.ino>