

1. Background and Purpose of research

As technology in this world advances, more and more things are being automated, humans are trying to simplify their lives by adding remote control systems as well as sensors to everyday appliances for ease of use. These technological advancements follow a new trend, the Internet of Things.

The idea is relatively untouched and in an attempt to better understand the mechanics behind this new trend as well as try our hands on creating and automating a system that could help make a building's safety system more efficient in times of need. We have set out to build our own set-up for the Internet of Things.

In order to do so we have created a template for the internet of things with our own server, client as well as set-up. By implementing different sensors and programs we can modify this template to do a variety of things. For example, if adopted, the buildings safety system will be able to inform everyone in the building to evacuate in a matter of few seconds in case of an emergency. It will also inform the firemen and the police providing them with the location of the same.

The reason we chose to improve the smoke detectors through the Internet of Things is to reduce loss of life when there is a fire in a building. We will like to contribute to the society in making this world a better place to live in.

Through our project we have learnt much about this new trend as well as found an affordable way to access this design of the Internet of Things.

2. Hypothesis

Our hypothesis is that Smoke Detectors can be made better and more effective using Arduino and Programming. The present smoke detectors technology are quite slow in the circulation of the news and using programming will speed up this process.

3. Research Method and Materials (Methodology)

Materials Used

- **Arduino Board** : The Arduino is a small, programmable microcontroller board that accepts and stores code from your computer capable of various outcomes
- **Jumper Wires** : You use wires to connect the Arduino to the breadboard and create the circuits that make everything communicate.
- **Breadboard** : The breadboard makes connections between different electronics without soldering them together. It's made up of a grid of tiny holes where you can connect different components. These holes are all linked like a small Lite-Brite. On the outside of

the board, they're linked horizontally so that one end of the board is connected to the other. On the inside they're linked vertically. This makes it so one wire can send information to another component without having to physically connect them. The one in the picture to the right is a mini breadboard we experimented with, but this project is easier with a regular sized breadboard.

- 12V DC Power Supply : This is standard 12V power supply. We can find the voltage on the back of a power supply brick or beneath the tongs under "output." We are using it to directly power the Ethernet Shields, which is why you won't be connecting it into the Arduino itself.

Research Method

Step One: Install Arduino, Processing, and the Arduino Drivers on Your Computer

We first Download and install the Processing and Arduino software. We also needed to download the Arduino compatible USB Drivers so that our machine can properly interface with the Arduino.

Step Two: Copy, Paste, and Run the Code in Arduino

We Open the Arduino software and paste the Arduino code into a new sketch. Then we Click Sketch > Verify / Compile to make sure the code is properly in there. After Saving the file and attaching the Arduino to the computer with the USB cable we Click File > Upload. Now everything is ready on the software side.

```
import time
import serial
import smtplib
TO = 'putyour@email.here'
GMAIL_USER = 'putyour@email.here'
GMAIL_PASS = 'putyourpasswordhere'
SUBJECT = 'Intrusion!!'
TEXT = 'Your smoke detector detected movement'
ser = serial.Serial('COM4', 9600)
def send_email():
    print("Sending Email")
    smtpserver = smtplib.SMTP("smtp.gmail.com",587)
    smtpserver.ehlo()
    smtpserver.starttls()
    smtpserver.ehlo
    smtpserver.login(GMAIL_USER, GMAIL_PASS)
    header = 'To:' + TO + '\n' + 'From: ' + GMAIL_USER
```

```

header = header + '\n' + 'Subject:' + SUBJECT + '\n'
print header
msg = header + '\n' + TEXT + '\n\n'
smtpserver.sendmail(GMAIL_USER, TO, msg)
smtpserver.close()
while True:
    message = ser.readline()
    print(message)
    if message[0] == 'M' :
        send_email()
        time.sleep(0.5)

```

Step Three: Arduino Code

The Arduino will send a message over USB Serial connection whenever smoke is detected. However, this could have the potential to generate a lot of emails. For this reason the Arduino sends a different message if its too soon to send another email.

```

int pirPin = 7;
int minSecsBetweenEmails = 60; // 1 min
long lastSend = -minSecsBetweenEmails * 1000l;
void setup()
{
    pinMode(pirPin, INPUT);
    Serial.begin(9600);
}
void loop()
{
    long now = millis();
    if (digitalRead(pirPin) == HIGH)
    {
        if (now > (lastSend + minSecsBetweenEmails * 1000l))
        {
            Serial.println("SMOKE");
            lastSend = now;
        }
        else
        {
            Serial.println("Too soon");
        }
    }
    delay(500);
}

```

Step Four: Connect the Smoke Detector to the Breadboard

The only thing that we are connecting to the Arduino is the smoke Detector, so we can just simply push the wires attached to the Smoke Detector directly into the Arduino board.

However, the wires from the Detector, are a bit loose in the Arduino sockets so we used the breadboard layout below.

Step Five: Plug the Arduino Into the Computer, Run Processing and Test

Next, run the Processing code we compiled earlier. For a test run, we used a lighter to burn a paper and brought the paper near to this setup.

4. Results

After our set- up was ready, we practically tried out the experiment. We just used a lighter and burnt a small piece of paper near the smoke detector. The smoke detector detected high levels of smoke and as hypothesised, it sent an email to us saying “Fire, evacuate the building and do not use the lift”. The email was received within 1 minute.

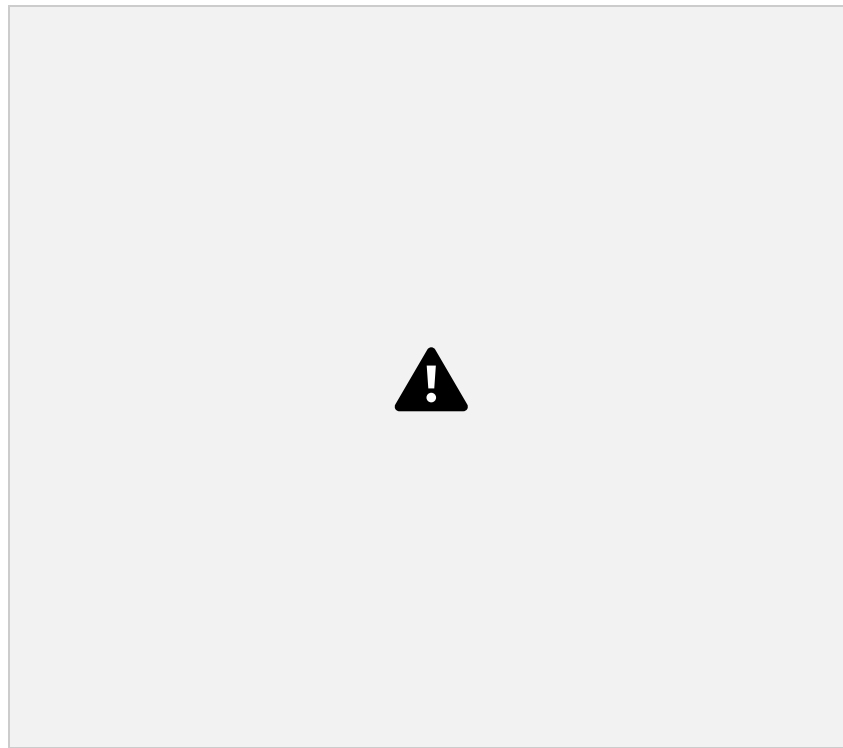


Fig 1:- Image showing the email sent by the program to us
image)

(self-created

Our methodology has been effective as we were able to inform the “culprits” about the situation. So, in a real emergency case, the buildings faculty will be informed about the situation and they will evacuate the building. In the meantime, the police and the firemen are on their way towards the fire.

5. Conclusions

Obtaining the results as hypothesised proves that our methodology is correct. We have improved the effectiveness of the smoke detectors in the Internet of Things, currently, tested only at a small scale. We will like to conclude that this smoke detector, with its new features can be used at a large scale in buildings to improve the building’s safety system.

6. Future Work

Firstly, we would like to test our experiment at a large scale. Mainly, to test whether the arduino can send an email to everyone in the building, informing them about the threat.

Secondly, we would like to do further research and send a text message or an emergency call.

Thirdly, we would like experiment in order to inform the firemen about the exact location of the fire in the building beforehand so that they are already prepared for the same.

7. References

<http://forum.arduino.cc/index.php?topic=200789.0>

<http://www.instructables.com/id/Modify-a-6-EUR-smoke-detector-for-use-with-microco/>

<http://jobs.arduinoexperts.com/jobs/arduino-smoke-detector-with-ethernet-and-notification-functionality/>

<https://learn.adafruit.com/rgb-led-strips>

<http://playground.arduino.cc/Main/ManualsAndCurriculum>

<http://arduino.cc/en/Main/arduinoBoardLilyPad>

<https://learn.sparkfun.com/resources?page=all>

<http://www.instructables.com/id/Smoke-Detector/>

<http://as3breeze.com/arduino-smoke-detector-2/>

<http://node-ardx.org/>

http://www.sgbotic.com/index.php?dispatch=categories.view&category_id=7