

## Smart Door

Oğuzhan Topal - 190503001 - Group:35

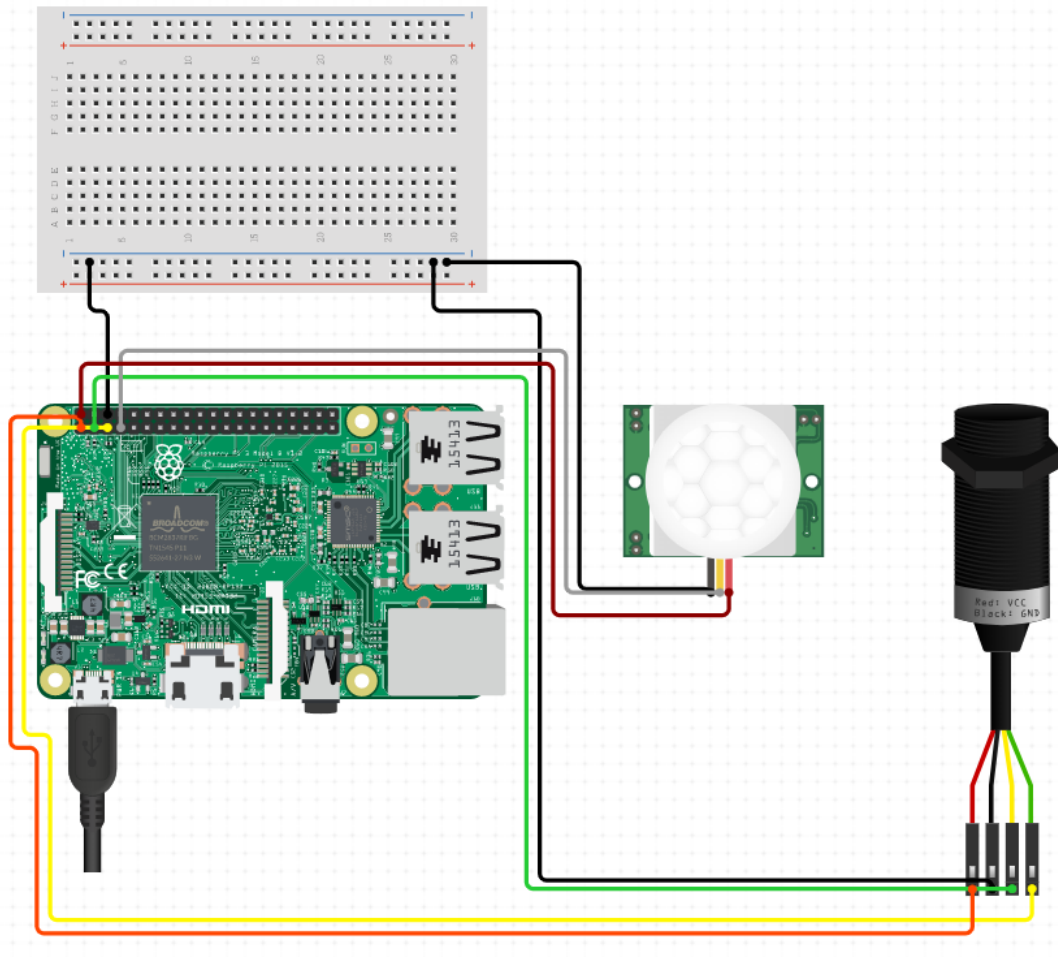
## 1. Beschreibung des Problem

Das Hauptproblem, das das Projekt zu lösen versucht, besteht darin, die Sicherheit der Haustür zu gewährleisten. Um dieses Problem zu lösen, wird bei Bewegung vor dem Haus ein Foto aufgenommen und an den Benutzer gesendet.

## 2. Komponenten und Circuit Diagram

Das System besteht aus drei Komponenten:

- Raspberry pi: Code funktioniert in dieser Komponente. Wenn eine Feedback vom Sensor oder der Knopf kommt, es nimmt ein Bild von der Kamera auf und sendet es dem Benutzer per E-Mail
- Bewegungssensor: Es gibt eine Feedback an pi, wenn eine Bewegung erkannt wird
- Camera: Es nimmt ein Foto auf und sendet es bei Bedarf an Pi.



### 3. Coden und Algorithmendesign

In diesem Abschnitt erkläre ich den Algorithmus des Programms, das auf dem Raspberry Pi läuft, und zeige die Flussdiagramme. Das System verfügt über zwei Unterfunktionen und eine Hauptfunktion (wie sie genannt werden), um das Foto von der Kamera zu empfangen und an den Benutzer zu senden. In diesem Abschnitt werde ich erklären, wie diese drei Funktionen mit den Codes.

#### 3.1 main Function

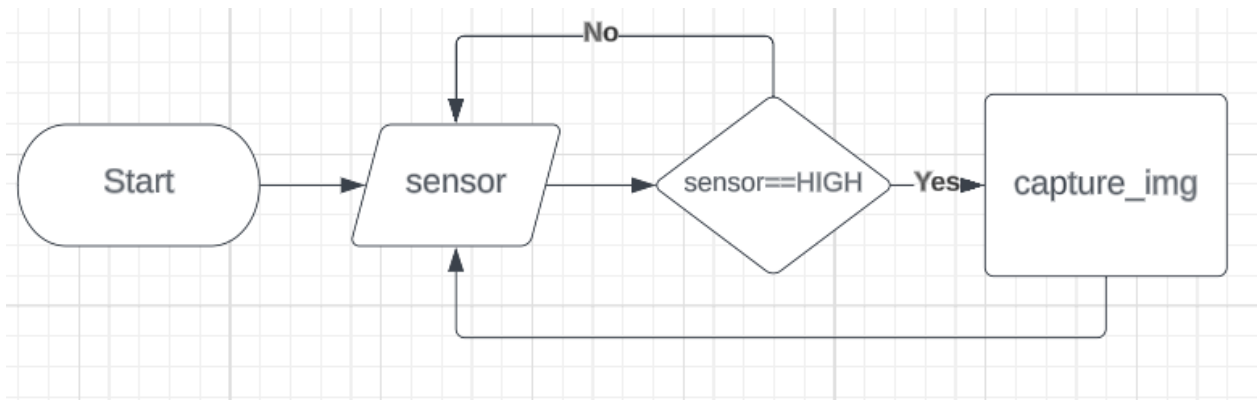
Die Hauptfunktion wird als Programm Eintrittspunkt definiert.

- Es richtet die wiringPi-Bibliothek ein und konfiguriert die Pin-Modi für den Bewegungssensor.
- Tritt in eine Endlosschleife ein, um den Sensor-Status kontinuierlich zu überwachen.

Innerhalb der Schleife:

- Liest die Zustände des Bewegungssensors mithilfe der digitalRead-Funktion.
- Wenn eine Bewegung erkannt wird (Bewegungssensor Status ist HIGH), wird „Motion Detected“ gedruckt und die Funktion capture\_img aufgerufen.
- Fügt mit Hilfe der Verzögerungsfunktion vor der nächsten Iteration eine kleine Verzögerung von 100 Millisekunden hinzu.

```
int main(void) {  
    wiringPiSetup();  
    pinMode(11, INPUT);  
    while (1) {  
        int motion = digitalRead(11);  
  
        if (motion == HIGH) {  
            printf("Motion Detected\n");  
            capture_img();  
        }  
        delay(100);  
    }  
    return 0;  
}
```



### 3.2 capture\_img Function

Die Funktion `capture_img` ist definiert, die für die Aufnahme eines Bildes und den Aufruf der Funktion `send_mail` verantwortlich ist. Es führt die folgenden Schritte aus:

- Prüft, ob das Verzeichnis zum Speichern von Bildern existiert und erstellt es, wenn nicht.
- Konstruiert einen Befehl, um mithilfe der Befehle `ls` und `grep` die aktuelle Bilddatei im Verzeichnis zu finden.
- Ruft die Anzahl der neuesten Bilddatei ab und erhöht sie, um den neuen Bilddateinamen zu ermitteln.
- Erfasst ein Bild mit dem Befehl `raspistill` (oder dem entsprechenden Befehl für Ihre Kamera) und speichert es unter dem generierten Dateinamen.
- Ruft die Funktion `send_mail` mit dem Dateinamen als Argument auf, um die E-Mail mit dem Bildanhang zu senden.

```

void capture_img() {
    printf("Capturing\n");
    if (system("test -d ./Visitors/") != 0) {
        system("mkdir ./Visitors/");
    }

    char command[200];
    sprintf(command, "ls -lt %s | grep \"%s[0-9][0-9][0-9].jpg\" | head -n 1", DIR,
prefix);

    FILE* pipe = popen(command, "r");
    if (pipe == NULL) {
        printf("Error executing command.\n");
        return;
    }

    char output[100];
    fgets(output, sizeof(output), pipe);
    pclose(pipe);

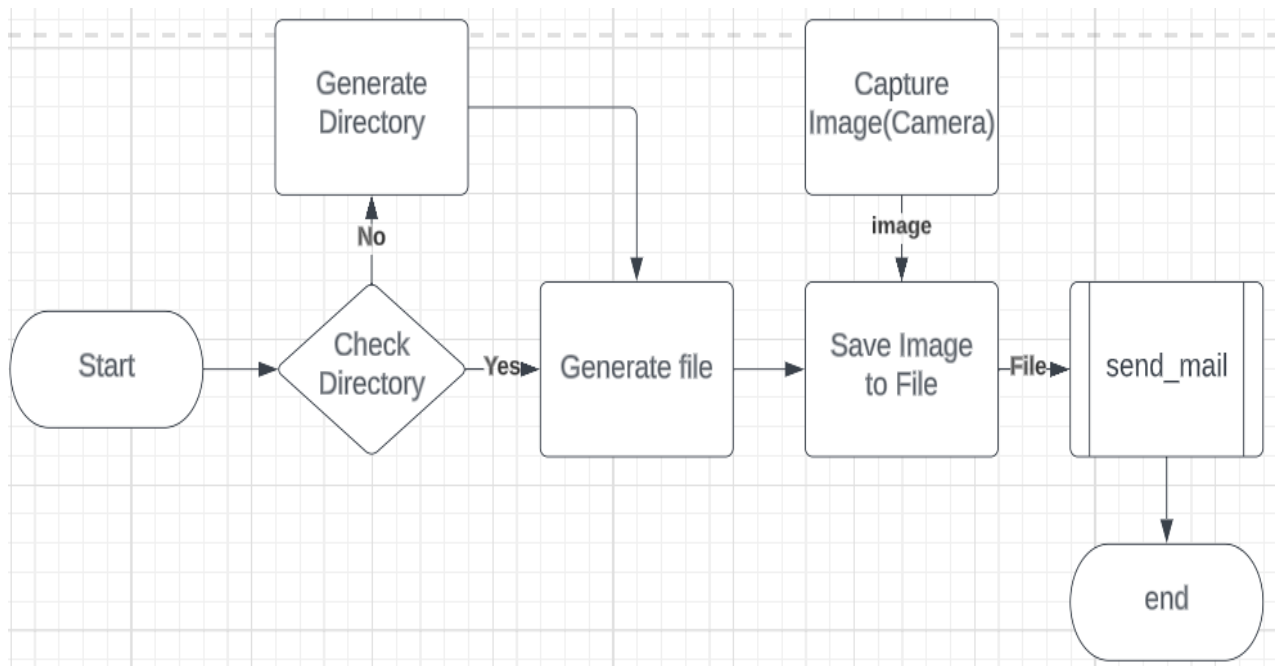
    int count = 0;
    if (output[0] != '\0') {
        sscanf(output, "%*^[0-9]%d", &count);
        count++;
    }

    char filename[100];
    sprintf(filename, "%s%s%03d.jpg", DIR, prefix, count);

    sprintf(command, "raspistill -o %s", filename);
    system(command);

    send_mail(filename);
}

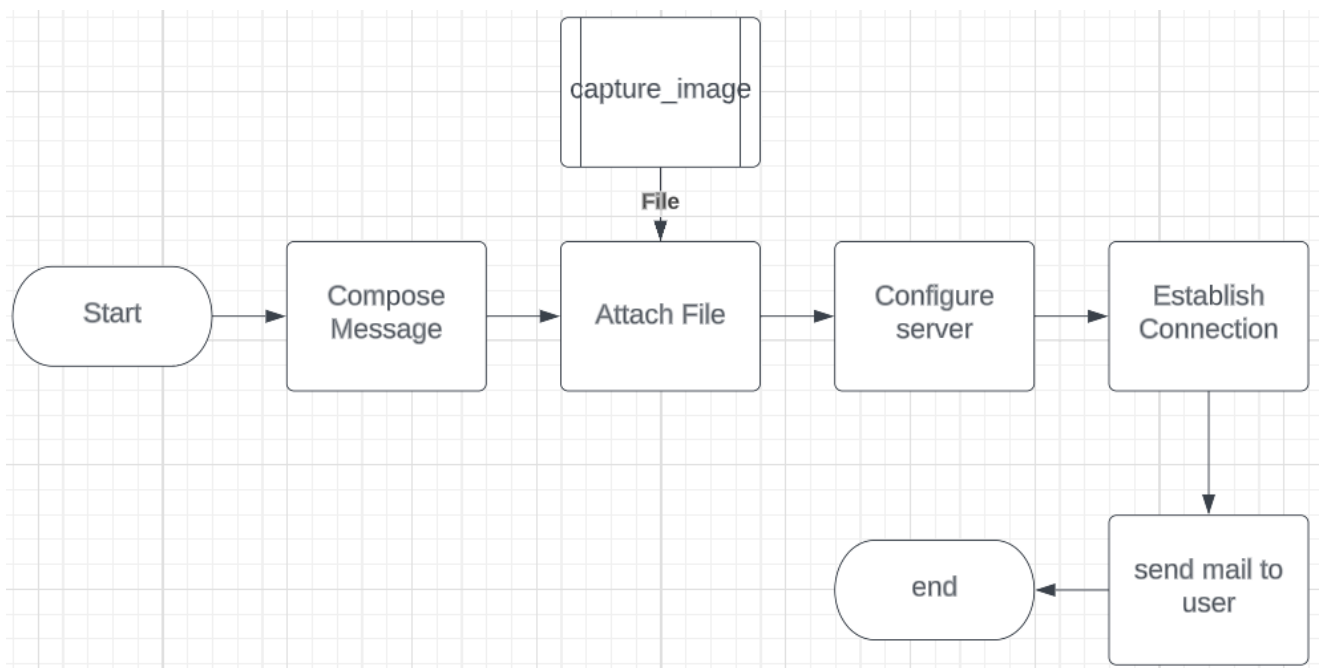
```



### 3.3 send\_mail Function

Die Funktion send\_mail ist definiert, die den Dateinamen eines Bildes als Eingabe verwendet. Sie erstellt eine E-Mail mit dem angehängten Bild und versendet diese über die Systemfunktion. Das System geht davon aus, dass senders gmail bereits am System angemeldet ist und ruft es mit der Systemfunktion auf

```
void send_mail(const char* filename) {  
    char command[500];  
    sprintf(command, "echo \"Picture is Attached.\" | mail -s \"Motion Detected\" -A %s  
-r %s %s", filename, sender, receiver);  
    system(command);  
}
```



## 4. Anwendungsaufgaben

In diesem Abschnitt finden Sie Screenshots des auf dem Terminal angezeigten Bildes und der vom Programm an uns gesendeten E-Mail. Da die Anwendung unabhängig vom Benutzer arbeitet, sendet sie über das Terminal nur Rückmeldung, um zu zeigen, dass sie problemlos funktioniert.

### 4.1 Console Output

```
Motion Detected
Capturing
Picture is Attached.
Motion Detected
Capturing
Picture is Attached.
Motion Detected
Capturing
Picture is Attached.
```

### 4.2 Mail

Motion Detected Harici Gelen Kutusu x



sender sender <ssender973@gmail.com>

Alici: ben ▼

Picture is Attached

Bir ek • Gmail tarafından tarandı ⓘ



↩ Yanıtla

➡ Yönlendir