

2022年度 組込みシステム実践基礎 発表会資料
チーム1

赤外線センサを用いた夜間の警備システム

19T3072L 椿敬太 18T3038Y 桐原朋由 19T3056A 佐々木晴務

担当教員 武田茂樹教授

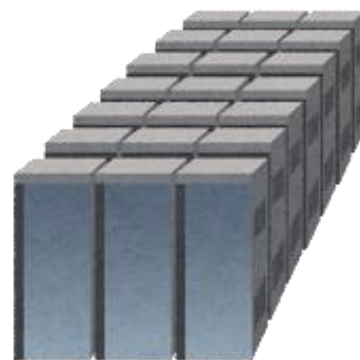
2022年8月8日

茨城大学 工学部 電気電子システム工学科

- I. 研究背景
- II. システムの概要
- III. 使用機器
- IV. システム動作の流れ
- V. 回路図
- VI. ソフトウェアについて
- VII. 作成したシステム
- VIII. システムの実行
- IX. まとめと今後の展望
- X. 参考文献

I . 研究背景

精密機器や危険物等を取り扱う大学の研究室では夜間の警備を強化することが必要。



夜間の人のない時間に研究室に忍び込む不審者を撮影し、関係者に送信するシステムを構築する。

Ⅱ. システムの概要

- Raspberry Pi4を用いて、夜間に焦電赤外線センサの前を人が通ると赤外線カメラ起動し撮影をする。
- 撮影した画像は日時と共にスマホ等へLINEで送信して離れた場所からでも確認できるようにする。

利点

- 赤外線を使用するので犯人に気づかれにくい。
- 赤外線カメラを用いることで暗闇での撮影が可能。夜間の警備に役立つ。
- 撮った写真はすぐにWi-Fiを通し送信されるため、いち早く被害に対応可能。
- Raspberry Pi 4を使用するため幅広い拡張性がある。
- カメラの起動設定を詳細に設定可能で様々な場面に対応できる。

Ⅲ. 使用機器

装置名	会社名	型番	シリアル番号
Raspberry Pi 4 Model B 4GB	Raspberry Pi Foundation	RASPI44SC0194#	100000001c9e4419
焦電赤外線センサ	Nanyang Senba Optical&Electronic	SB612A	210310D0356
赤外線カメラ	KUMAN	SC15-JP	X000HF3401

参考文献[1][2][3]



図1 Raspberry Pi 4

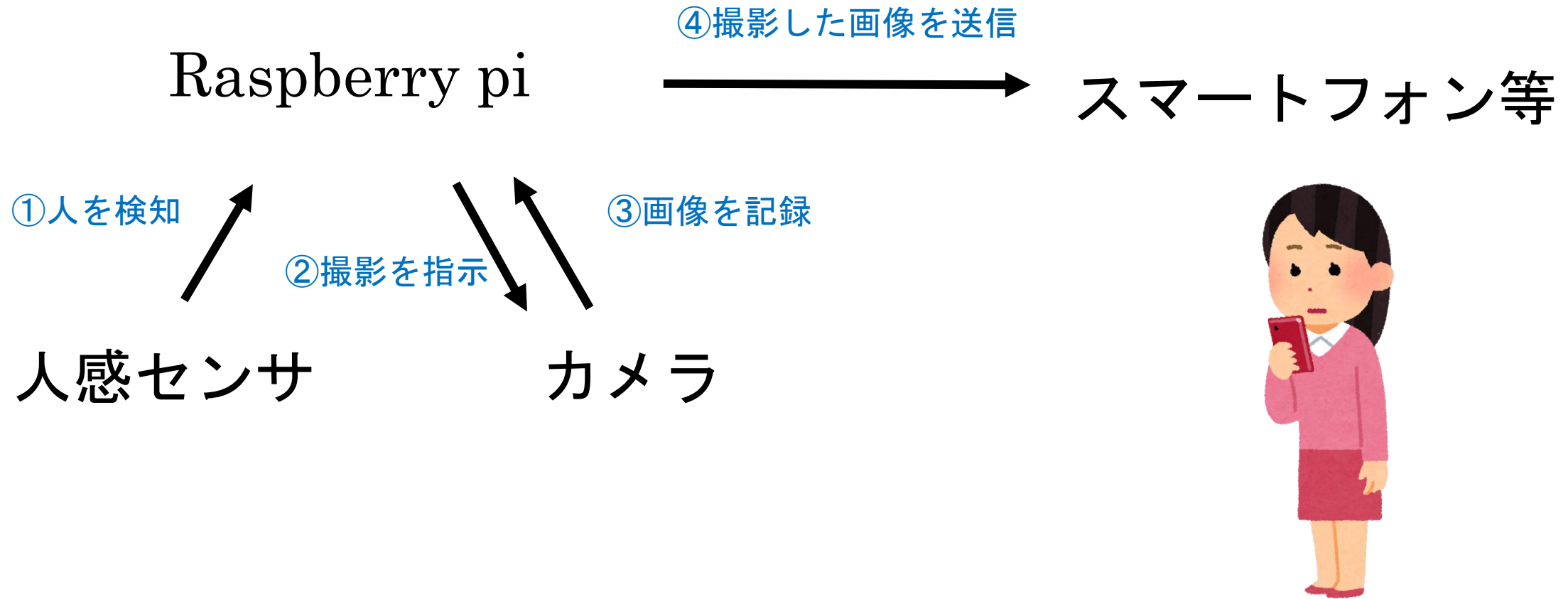


図2 焦電赤外線センサ



図3 赤外線カメラ

IV. システム動作の流れ



V. 回路図

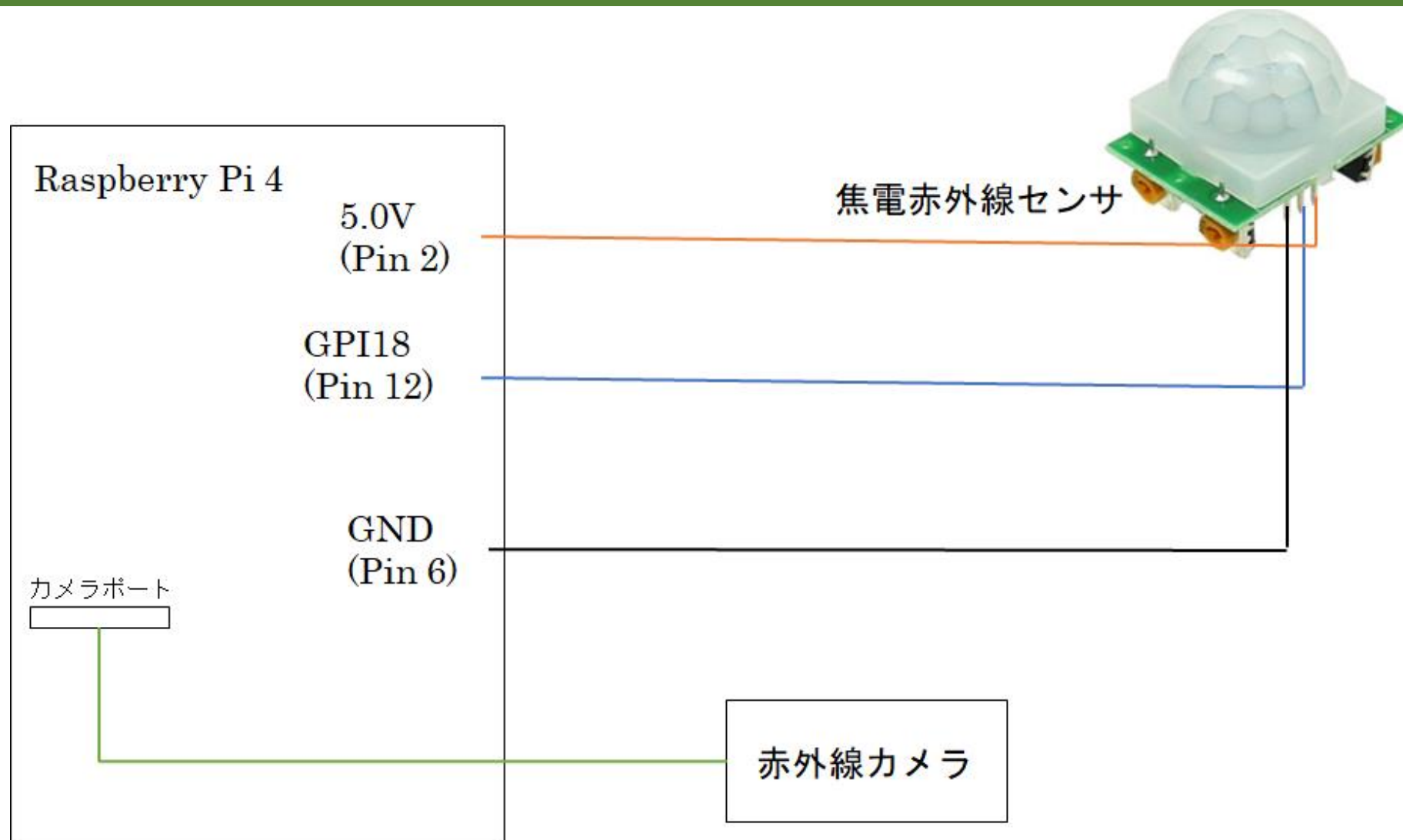


図4 作成した回路

参考文献[7]

VI. ソフトウェアについて

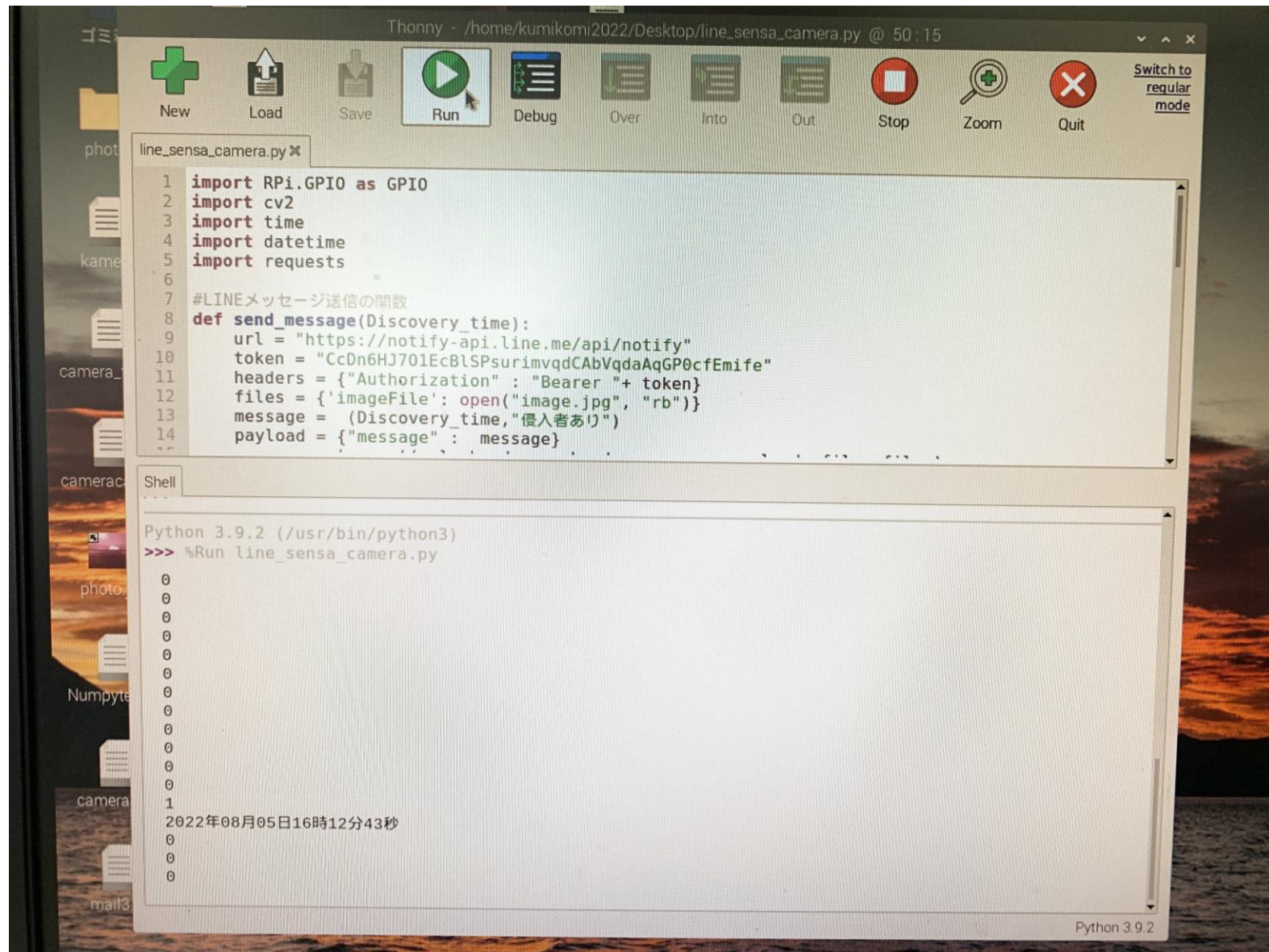
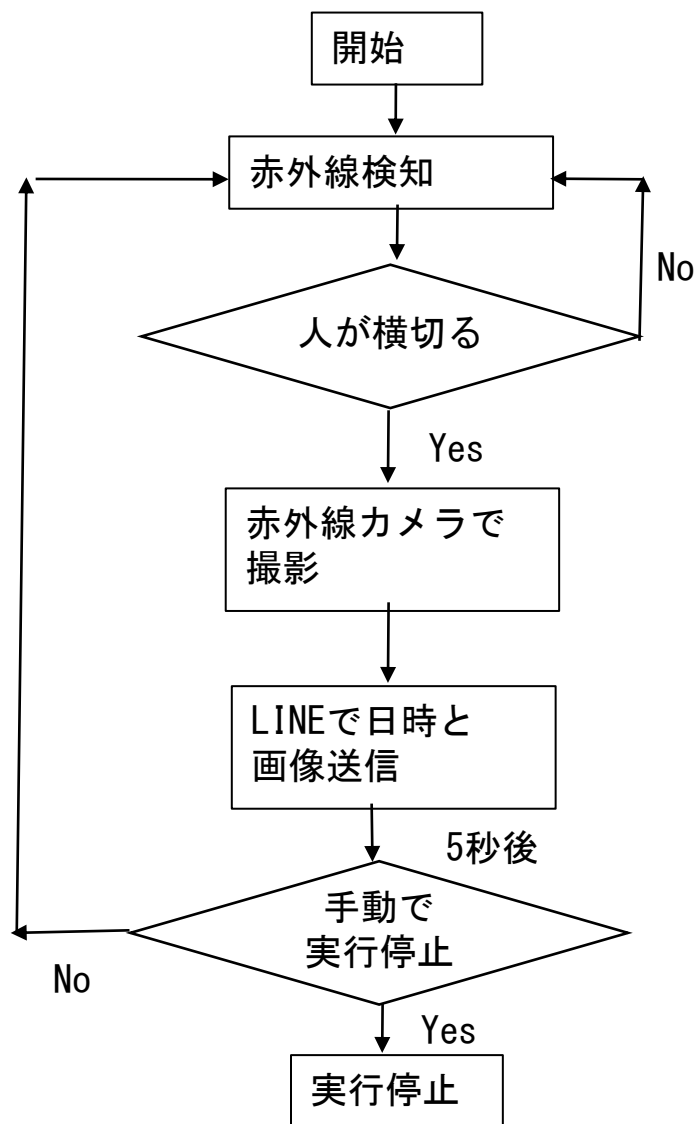


図5 フローチャートと実行画面

VI. ソフトウェアについて

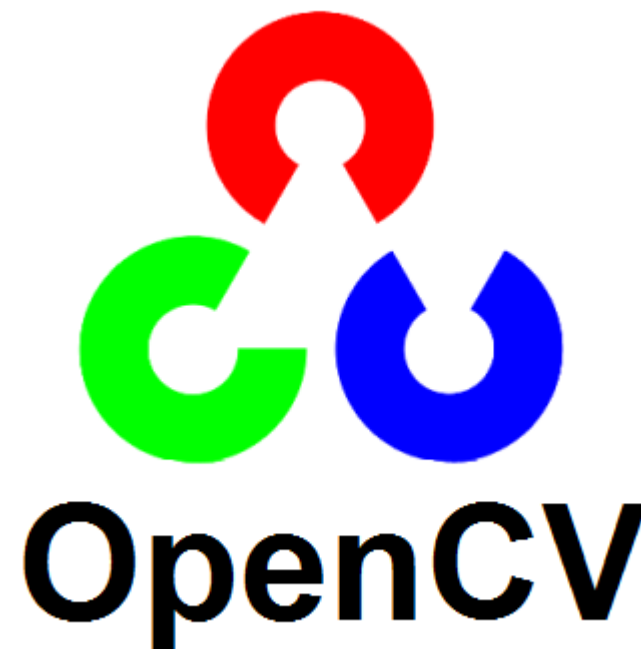
使用したアプリ

- ・ line notify

LINEが出しているサービスの一つ
トークンを発行することで、画像を含め
メッセージを簡単にLINEで送信すること
ができる。

- ・ OpenCV

コンピュータ内の画像・動画に関する処理機
能が入ったオープンソースのライブラリ
今回は画像を保存および管理するために導
入した。



VI. ソフトウェアについて

ソースコード

参考文献 [4] [5] [6]

```
import RPi.GPIO as GPIO
import cv2
import time
import datetime
import requests
```

```
#LINEメッセージ送信の関数
def send_message(Discovery_time):
    url = "https://notify-api.line.me/api/notify"
    token = "トークン"
    headers = {"Authorization": "Bearer "+ token}
    files = {'imageFile': open("image.jpg", "rb")}
    message = (Discovery_time, "侵入者あり")
    payload = {"message": message}
    r = requests.post(url, headers = headers,
params=payload, files=files)
```

```
#センサーを使う準備
GPIO_PIN = 18 GPIO.setmode(GPIO.BCM)
GPIO.setup(GPIO_PIN, GPIO.IN)
```

```
while True:
    if(GPIO.input(GPIO_PIN) == GPIO.HIGH):
        #センサー検出時の処理
        print("1")
```

```
#検出時間の取得
    dt_now = datetime.datetime.now()
    Discovery_time = dt_now.strftime('%Y
年%m月%d日%H時%M分%S秒')
    print(Discovery_time)
```

```
#カメラ画像を保存する
    cap = cv2.VideoCapture(0)
    ret, frame = cap.read()
    cv2.imwrite("image.jpg", frame)
    cap.release()
    #LINEメッセージ送信
    send_message(Discovery_time)
```

```
#5秒待機
    time.sleep( 5 )
else:
    #センサー未検出時の処理
    print("0") time.sleep( 1 )
```

```
GPIO.cleanup()
```

VII. 作成したシステム

システムの外形

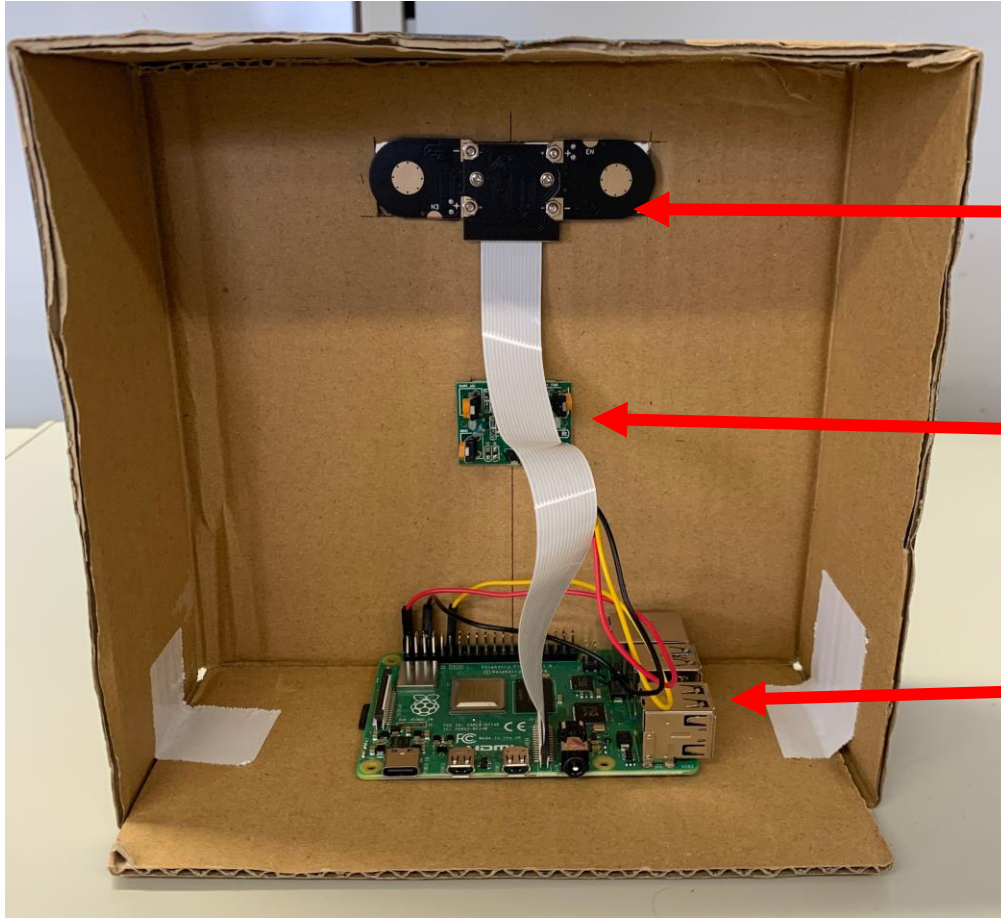


図6 システム裏面

赤外線カメラ

焦電赤外線センサ

Raspberry Pi 4



図7 システム表面

VIII. システムの実行

実験環境

図の様にシステムを設置し実行した。



背面から見たシステム



図8 出入口から見たシステム

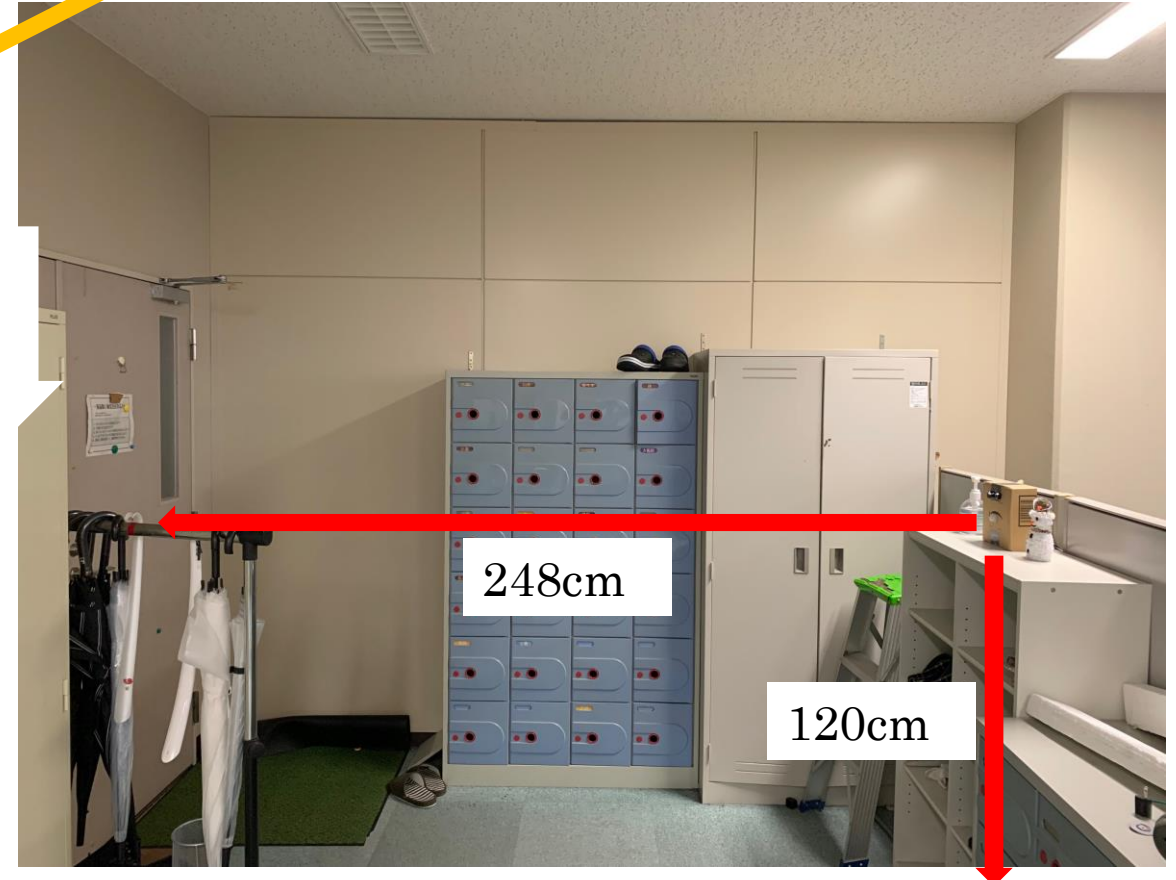


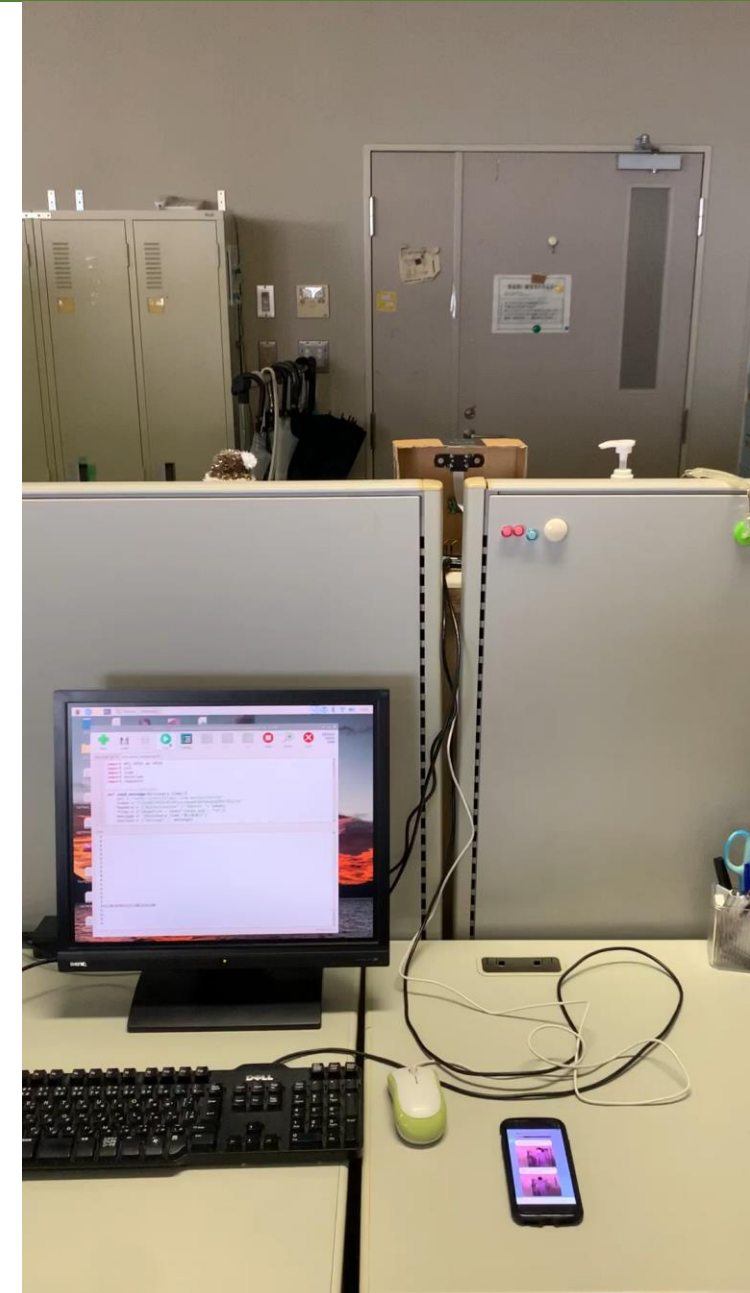
図9 横から見たシステム

VIII. システムの実行

実験の様子（※動画）



図10 実行している様子



VIII システムの実行

実験結果

明所で撮った写真とメッセージ

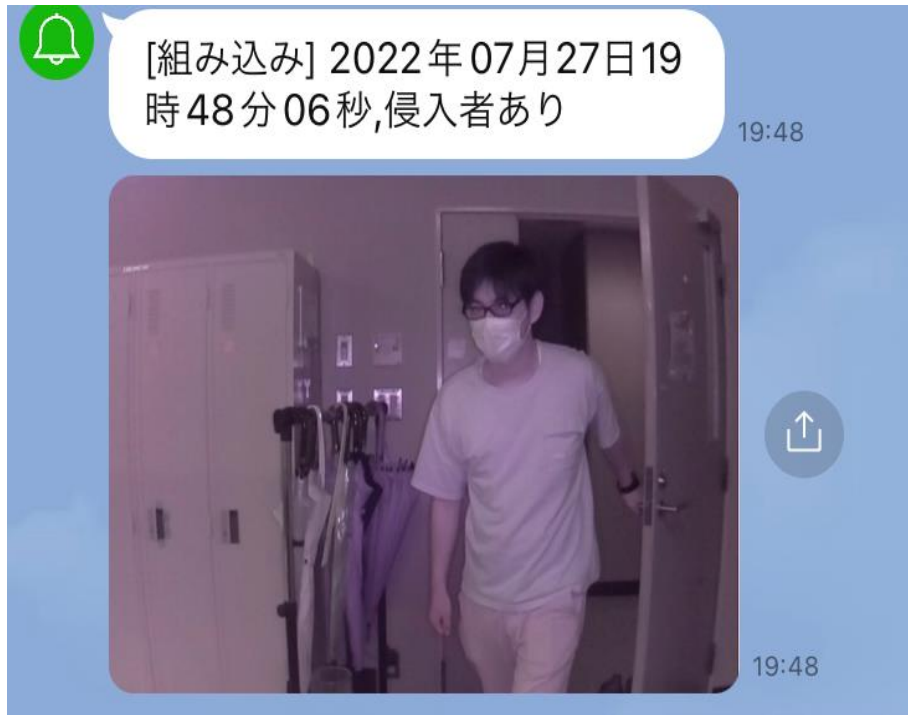


図11 実際に送られたメッセージ



図12 iPhoneXSのカメラで撮った写真

VIII システムの実行

実験結果

暗所で撮った写真とメッセージ



図13 実際に送られたメッセージ

IX. まとめと今後の展望

まとめ

- 焦電赤外線センサで人を検知し、写真を撮影、LINEに送信するシステムを実現することができた。

今後の展望

- 外部から無線でシステムを制御できるようにする。
- 画像認識で研究室の人と部外者を判別できるようにする。

IX. 参考文献

[1] Raspberry pi4

<https://rasberry-pi.ksyic.com/main/index/pdp.id/498/pdp.open/498>

[2] 焦電人感センサ

<https://akizukidenshi.com/catalog/g/gM-14064/>

[3] 赤外線カメラ

<https://www.amazon.co.jp/dp/B01IC05CFI>

[4] カメラ撮影で使用

[Raspberry Piカメラモジュールのセットアップと使い方 \(V2、HQ対応\) - Indoor Corgi \(indoorcorgielec.com\)](#)

[5] OpenCV のインストールで使用

<http://independence-sys.net/main/?p=6858>

[6] コード作成で使用

[【ラズベリーパイ】監視カメラの作り方 | Pythonでカメラモジュールを操作 | sozorablog](#)

[7] ピン配列の確認

[Raspberry PiのGPIOピン配置 - TECHHOBBY](#)