

Raspberry Pi

```
#Copyright KeonWoo PARK 2020, All right reserved.
#2020-11-12 Stepper motor control Library for Raspberry pi.
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
class stepper:
    def __init__(self,step,dir,speed):
        self.step_port=int(step)
        self.dir_port=int(dir)
        self.speed=int(speed)
        GPIO.setup(self.step_port,GPIO.output)
        GPIO.setup(self.dir_port,GPIO.output)
        GPIO.output(self.step_port,False)
        GPIO.output(self.dir_port,False)

    def setspeed(self,speed):
        self.speed=speed

    def forward(self,round):
        GPIO.output(self.dir_port,False)
        for i in range(200*round):
            GPIO.output(self.step_port,True)
            time.sleep(self.speed/1000000)
            GPIO.output(self.step_port,False)
            time.sleep(self.speed/1000000)

    def backward(self,round):
        GPIO.output(self.dir_port,True)
        for i in range(200*round):
            GPIO.output(self.step_port,True)
            time.sleep(self.speed/1000000)
            GPIO.output(self.step_port,False)
            time.sleep(self.speed/1000000)

from huskylib import HuskyLensLibrary #Huskylens lib
import json
import time
import serial #Pyserial lib for connect to Arduino
from stepper import stepper
from pyfingerprint.pyfingerprint import PyFingerprint
import wave
import pyaudio
import GPIO.RPi as GPIO
import pygame
human=["박건우","손태일","전하은","이준우"]
Husky = HuskyLensLibrary('SERIAL', '/dev/ttyUSB0', 3000000)
f = PyFingerprint('/dev/ttyUSB1', 57600, 0xFFFFFFFF, 0x00000000)
Arduino=serial.Serial(port='/dev/ttyACM0', baudrate=9600, timeout=.5)
Arduino.flush()
Arduino.flushInput()
Arduino.flushOutput()
stepl = stepper(9,10,700)
loc = '/home/pi/Desktop/HUSKYLENS/Voice/'
GPIO.setup(11,GPIO.output) ##마스크
GPIO.setup(12,GPIO.output)
GPIO.setup(13,GPIO.output)
GPIO.setup(14,GPIO.output)
GPIO.output(11,False)
GPIO.output(12,False)
GPIO.output(13,False)
GPIO.output(14,False)
def read():
    if Arduino.readable():
        res = Arduino.readline()
        res=res.decode()[:-2]
        return res
    else:
        return False
```

Raspberry Pi- 스텝모터

Arduino에서 스텝모터 운용

Raspberry Pi- main(발체)

Raspberry Pi에서 프로그램 총괄(지문인식, Husky lens 등)

Arduino Code

```
/*
Title: 아두이노 손소독제 자동 펌핑키트
Arduino Sanitizer Auto Pumping Kit
Youtube Link:
echo = 3
trig = 4
servo = 5
*/
#include <Servo.h>
Servo myservo;
int first_angle = 35; //you can change the value.
int second_angle = 1; //you can change the value.
int motion = 3;
void setup() {
    pinMode(motion, INPUT);
    myservo.attach(9);
    myservo.write(first_angle);
    Serial.begin(9600);
}
void loop() {
    int sensor = digitalRead(motion);
    Serial.println(sensor);
    if (sensor == HIGH) {
        myservo.write(second_angle);
        delay(400);
        myservo.write(first_angle);
        delay(800);
    }
    else{
        myservo.write(first_angle);
    }
}
```

Arduino-자동손소독제

인체감지센서를 이용해 사람 손 인식 후 손소독제 자동으로 나눔

```
#include <Wire.h>
#include <Adafruit_MLX90614.h>
#include <NewPing.h>
#define TRIGGER_PIN 6
#define ECHO_PIN 7
#define MAX_DISTANCE 200
NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE);
Adafruit_MLX90614 mlx = Adafruit_MLX90614();
void setup() {
    Serial.begin(9600);
    // start serial communication at 9600 bauds
}
void loop() {
    // Read and execute commands from serial port
    if (Serial.available()) {
        char commands = Serial.read();
        Serial.println(commands);
        switch(commands){
            case 't':
                Serial.println(mlx.readAmbientTempC());
                break;
            case 'c':
                Serial.println(sonar.ping_cm());
                break;
        }
    }
}
```

Arduino-비접촉 온도 센서 , 초음파센서

비접촉 온도 센서와 초음파 센서 시리얼 통신으로 라즈베리파이에 전달

COVID-19의 확산 방지를 위한 아이디어

위 문제를 해결하기 위해 출입구 예 설치가 가능한 무인 검사 프로토타입을 만들게 되었습니다.

이 프로토타입은 지문인식 센서를 이용하여 출입자 신원을 우선 확인한후,

인공지능 AI 카메라를 사용하여 출입자가 마스크를 착용하였을때 문이 열리게 되는 시스템으로

어느 한 조건을 만족하지 않았을 경우 문이 열리지 않게 되어 있으며

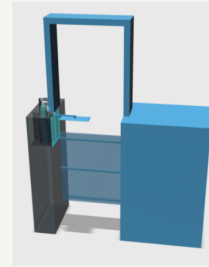
마스크 미착용자에 대해서는 마스크를 제공해주는 기능이 있습니다.

또한 출입시 손소독제 사용을 의무화 하도록 설계하였으며

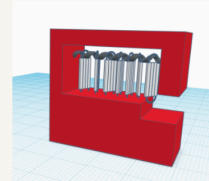
코로나 19의 방역에 도움을 주는 프로토타입을 구성 하였습니다.

프로토타입 작동 과정

1. 사용자가 입구 근처로 가까이 올 경우 안내음성 출력
2. 생체정보 인식(지문)으로 사용자 구별 (출입기록 작성)
3. AI 카메라를 통해 마스크 착용 여부 구별
 1. 마스크 미착용시 마스크 지급, 착용후 3번 과정 다시 반복
 2. 마스크 착용시 패스
4. 자동으로 손소독제 제공
5. 소독 과정 완료후 자동문 열림



전체 모형 렌더링 이미지



마스크 지급 모듈 내부 구조