

```

from datetime import datetime
import time
import RPi.GPIO as GPIO
import picamera
from slack import slack

```

```

# インターバル
INTERVAL = 3
# スリープタイム
SLEEPTIME = 10
# 使用するGPIO
GPIO_PIN = 18

GPIO.setmode(GPIO.BOARD)
GPIO.setup(GPIO_PIN, GPIO.IN)

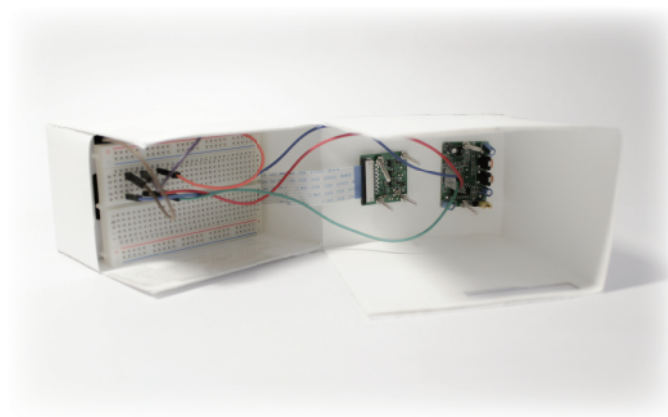
#slackのtoken
token = " slackのtoken"
slacker = Slacker(token)
# #toiletに送信
channel_name = "#"+
# メッセージ内容
message = 'トイレした'

```

```

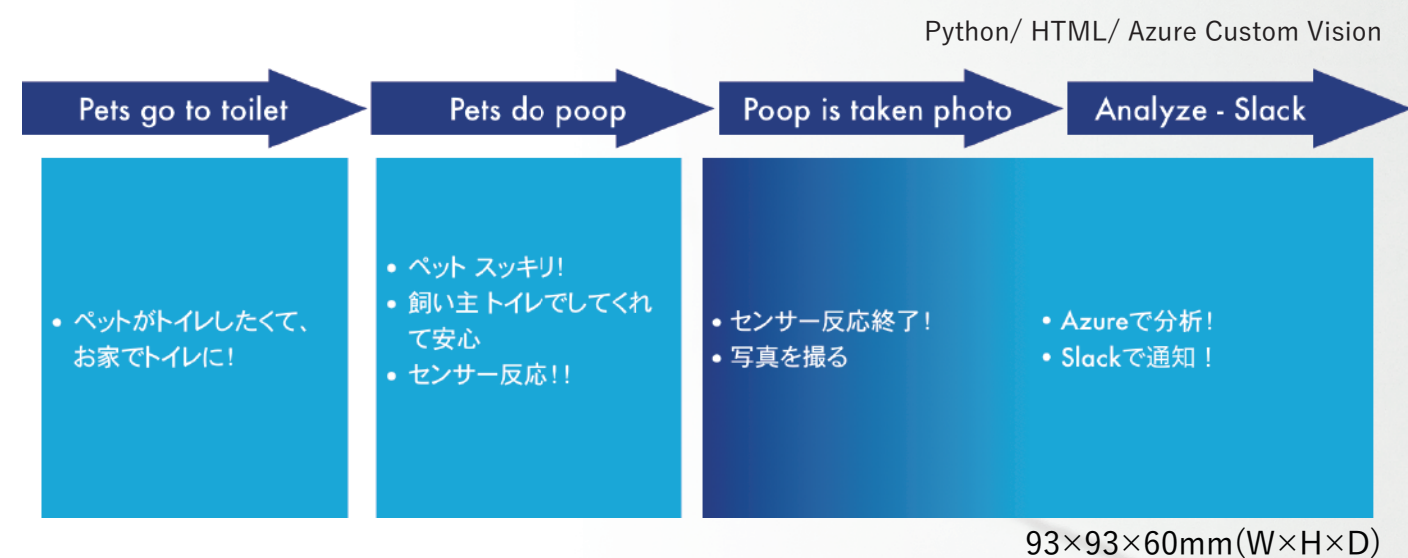
if __name__ == '__main__':
    try:
        print ("処理開始")
        cnt = 1
        while True:
            # センサー
            if(GPIO_PIN):
                print ("トイレした")
            else:
                print ("トイレして")
            cnt = cnt + 1
            time.sleep(SLEEPTIME)
        with picamera.PiCamera() as camera:
            camera.resolution = (640, 480)
            camera.start_preview()
            # Camera view
            time.sleep(INTERVAL)
            camera.capture_image(image)
            camera.close()
            # camera view
            # このとき、センサーが反応
            slacker.post(channel_name, message)
    except KeyboardInterrupt:
        print("終了処理中...")
    finally:
        GPIO.cleanup()
        print("GPIO clean完了")

```



人間の健康に関心が寄せられているのと同様に、昨今ではペットの健康にも大きな関心が寄せられている。しかし、ペットと人間の共通言語が少ない現在、体調を把握する方法は人間に比べて非常に少ない。そこでペットから発せられる視覚的情報の微々たる変化からを継続的にログを収集することで健康管理ができるのではないかと思い至った。センサーやコンピュータの小型化により、連続的な観察を行うことが昔に比べて容易になったからである。

そこで、小型コンピュータである「Raspberry Pi」と Microsoft 社が提供しているクラウドサービスの Azure で提供される画像認識サービスである「Custom Vision」を活用してペットの健康管理をするシステム及びプロダクトを製作した。



https://github.com/ryo-simon-mf/submit-Practice_of_Open_Design
Ryo Nishikado