인하공업전문대학
INHA TECHNICAL COLLEGE

무선 센서 네트워크
5주차

인하공업전문대학 컴퓨터 정보과

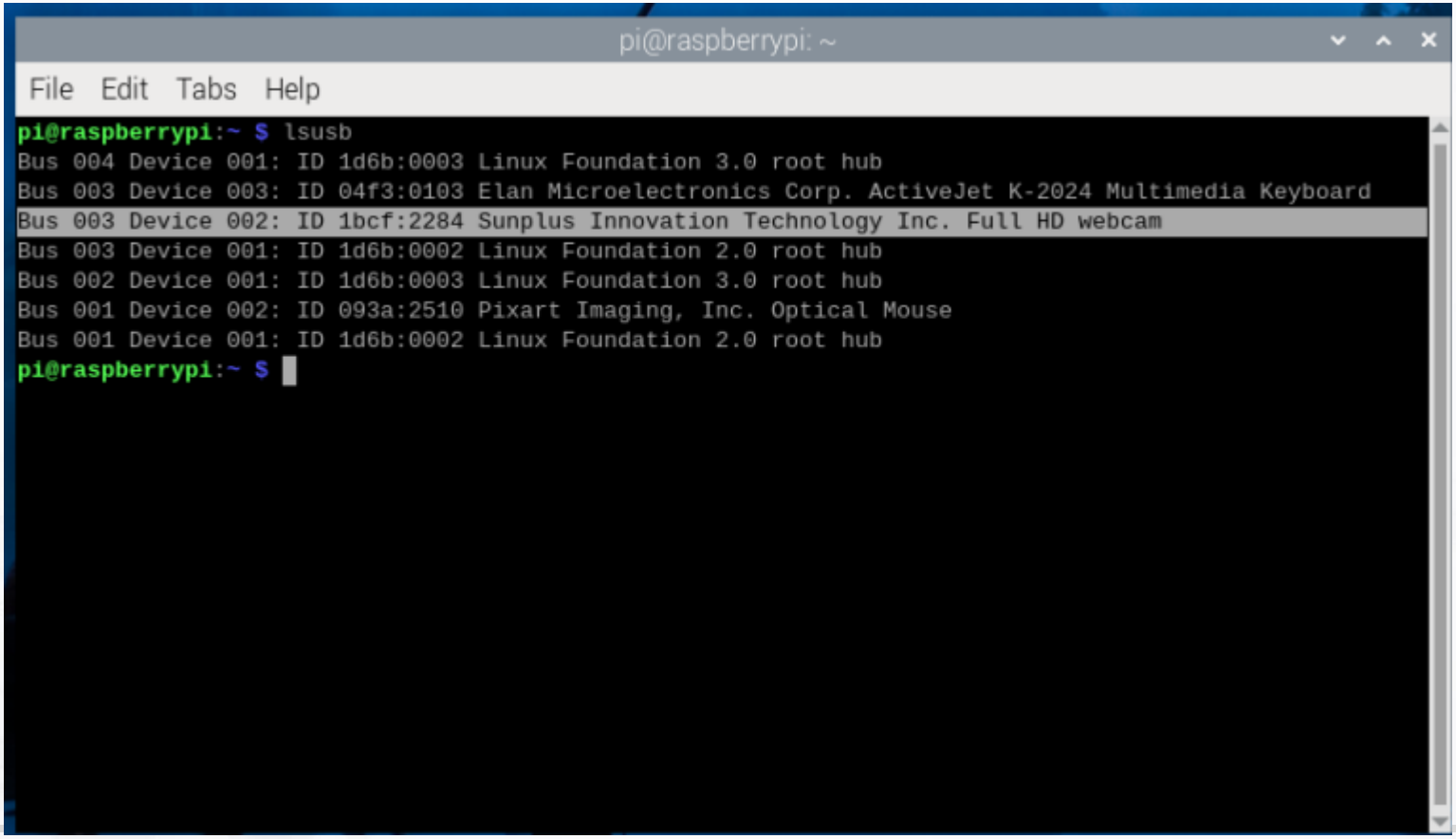
김한결 겸임교수

목차

- 요약
  I. TelegramBot Camera 제어
  II. Zigbee TinyOS 기반 온습도 센서
  III. NesC기반 미들웨어

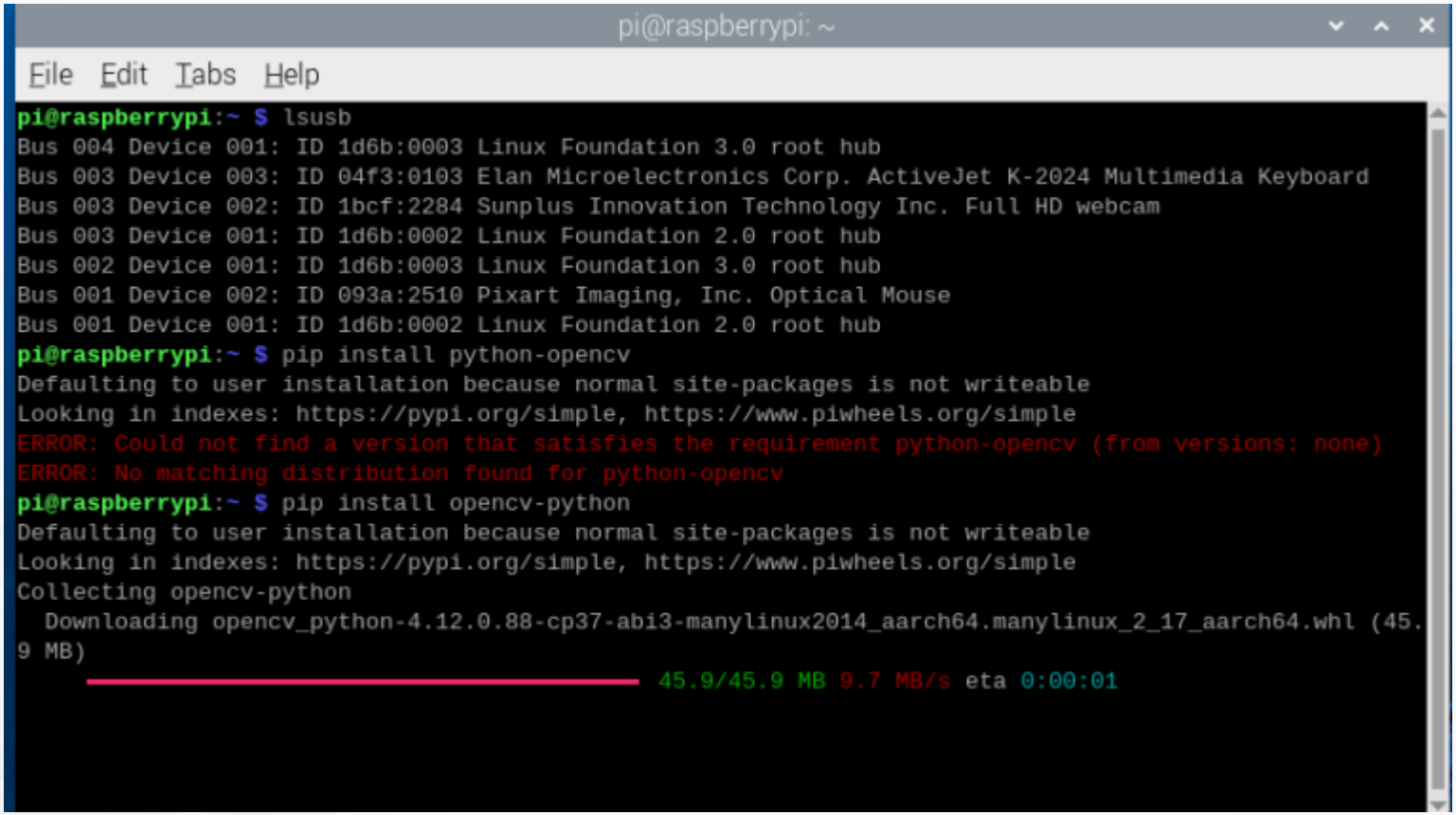
라즈베리파이 + Telegram 연동 하기

- Camera USB 연결 확인
- $lsusb



라즈베리파이 + Telegram 연동 하기

- Python – opencv 라이브러리 설치
- $pip install opencv-python



라즈베리파이 + Telegram 연동 하기

- Python – camera example code
- Test_camera.py

test_camera.py *

```python
import cv2

cam = cv2.VideoCapture(0)

if not cam.isOpened():
    print("Error: Could not open video device.")
    exit()

while True:
        # Read a frame from the camera
    ret, image = cam.read()

    if not ret:
        print("Error: Failed to grab frame.")
        break

        # Display the image
    cv2.imshow('USB Camera Feed', image)

        # Wait for a key press (1ms delay), if 'q' is pressed, break the loop
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

    # Save the last captured image (optional)
    cv2.imwrite('/home/pi/captured_image.jpg', image)

    # Release the camera and destroy all OpenCV windows
cam.release()
cv2.destroyAllWindows()
```

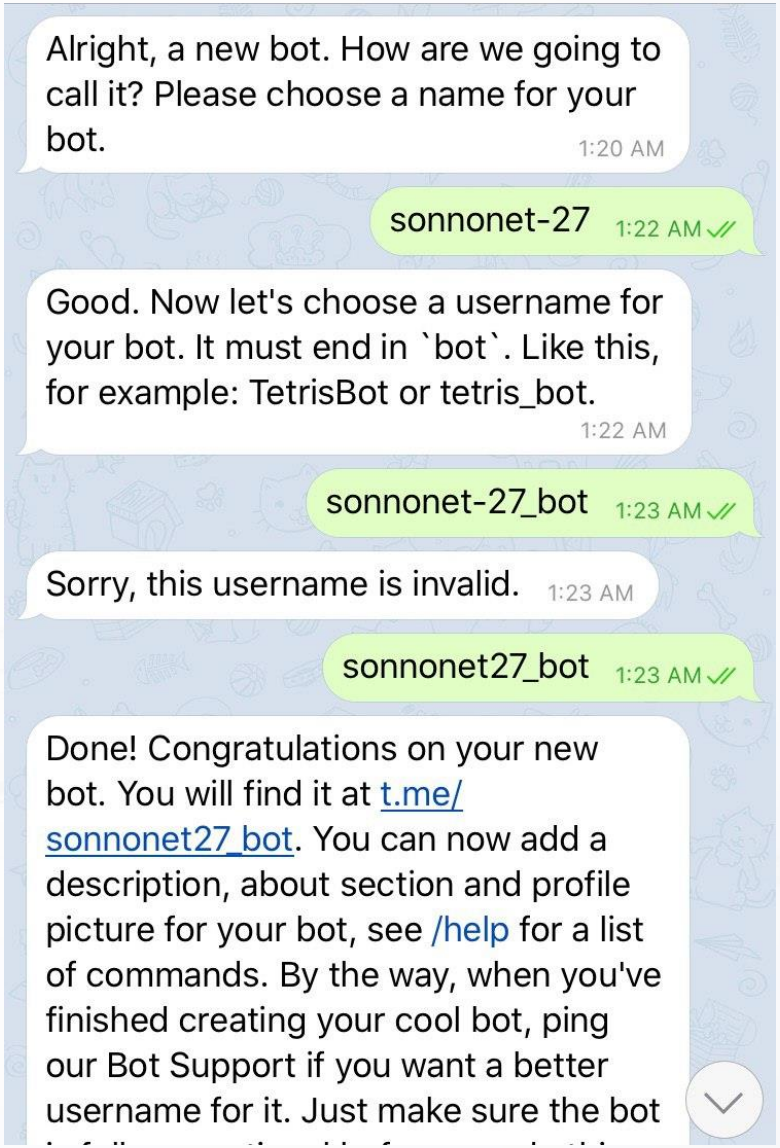
라즈베리파이 + Telegram 연동 하기

- 텔레그램 봇 만들기
  - /start

- 텔레그램 봇 만들기
  - /newbot

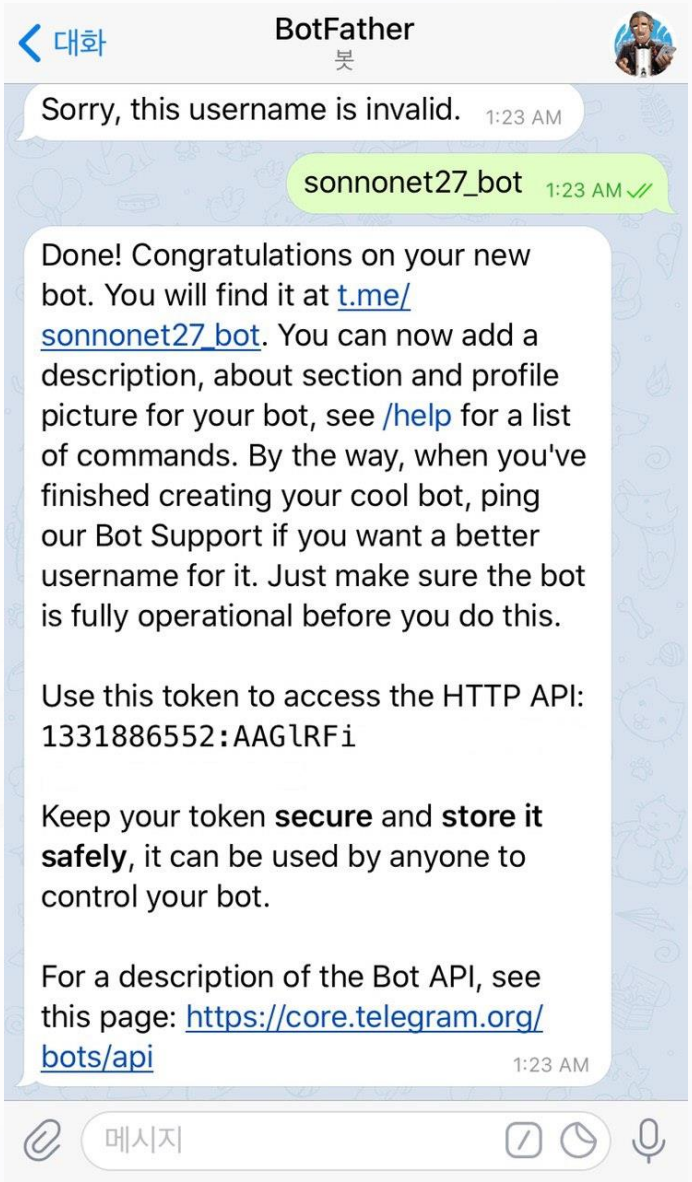
라즈베리파이 + Telegram 연동 하기

- 텔레그램 봇 이름 짓기
  - <원하는 이름>_bot



라즈베리파이 + Telegram 연동 하기

- 사용자 텔레그램 봇 token 부여 확인



라즈베리파이 + Telegram 연동 하기

라즈베리파이 Telegram API 설치

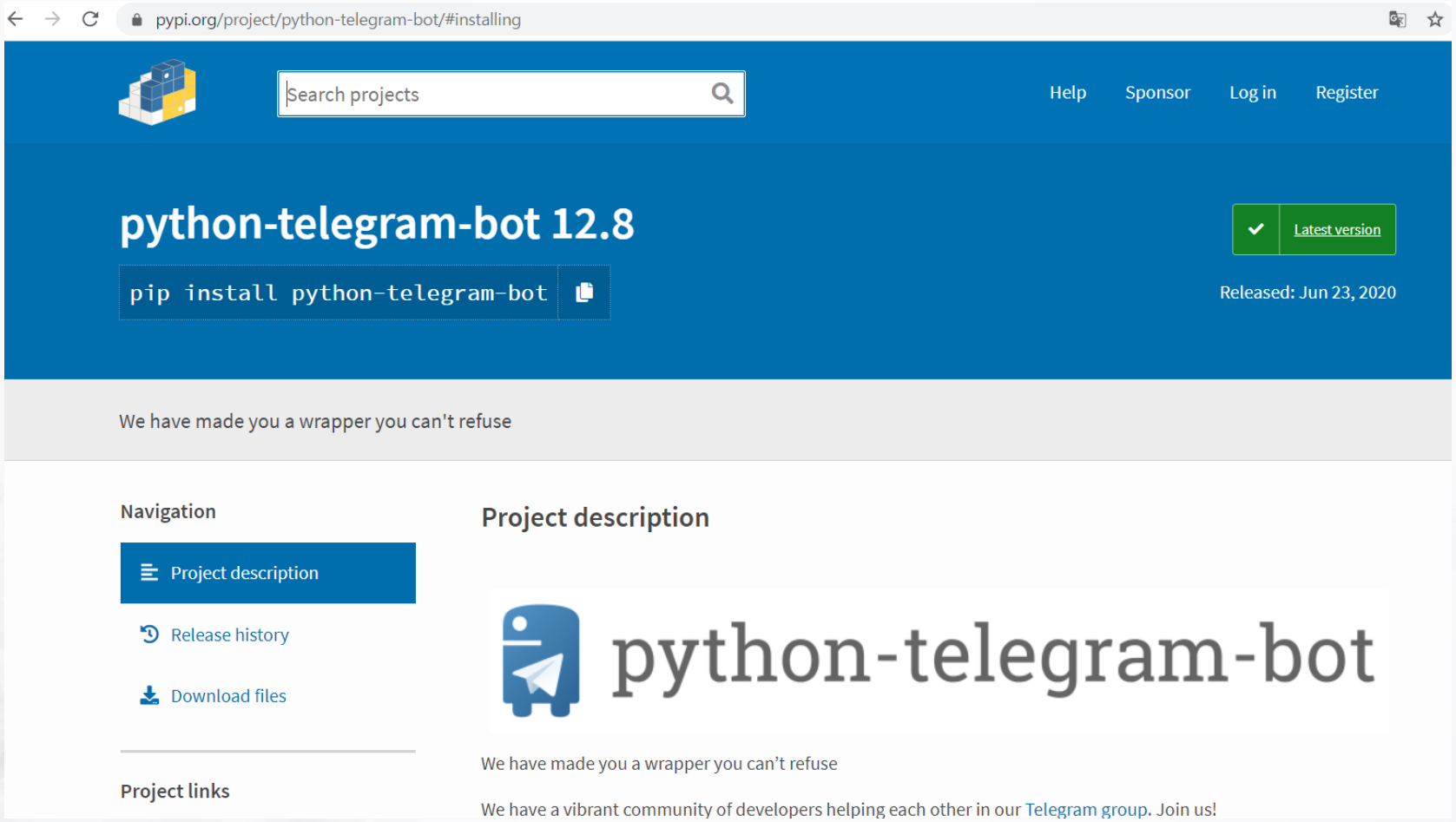
```
pi@raspberrypi: ~ $ pip install python-telegram-bot[job-queue]
```

```
pi@raspberrypi:~/work/jjvision $ pip3 install python-telegram-bot --upgrade
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Collecting python-telegram-bot
  Using cached https://files.pythonhosted.org/packages/a6/2d/c72fc9a28144277f617
0f2fcbfd3bd9427943497522b2689846596eb86cf/python_telegram_bot-12.8-py2.py3-none-
any.whl
Requirement already satisfied, skipping upgrade: certifi in /usr/lib/python3/dis
t-packages (from python-telegram-bot) (2018.8.24)
Requirement already satisfied, skipping upgrade: cryptography in /usr/lib/python
3/dist-packages (from python-telegram-bot) (2.6.1)
Requirement already satisfied, skipping upgrade: tornado>=5.1 in /usr/lib/python
3/dist-packages (from python-telegram-bot) (5.1.1)
Collecting decorator>=4.4.0 (from python-telegram-bot)
  Using cached https://files.pythonhosted.org/packages/ed/1b/72a1821152d07cf1d8b
6fce298aeb06a7eb90f4d6d41acec9861e7cc6df0/decorator-4.4.2-py2.py3-none-any.whl
Installing collected packages: decorator, python-telegram-bot
Successfully installed decorator-4.4.2 python-telegram-bot-12.8
```

라즈베리파이 + Telegram 연동 하기

라즈베리파이 Telegram API 사이트

https://pypi.org/project/python-telegram-bot/



라즈베리파이 + Telegram 연동 하기

라즈베리파이 Telegram API 사용

pi@raspberrypi: ~ $	git clone https://github.com/python-telegram-bot/python-telegram-bot --recursive

```
pi@raspberrypi:~/work $ git clone https://github.com/python-telegram-bot/python-telegram-bot --recursive
Cloning into 'python-telegram-bot'...
remote: Enumerating objects: 17, done.
remote: Counting objects: 100% (17/17), done.
remote: Compressing objects: 100% (16/16), done.
remote: Total 15876 (delta 7), reused 4 (delta 1), pack-reused 15859
Receiving objects: 100% (15876/15876), 6.22 MiB | 3.74 MiB/s, done.
Resolving deltas: 100% (12519/12519), done.
Submodule 'telegram/vendor/urllib3' (https://github.com/python-telegram-bot/urllib3.git) registered for path 'te
Cloning into '/home/pi/work/python-telegram-bot/telegram/vendor/ptb_urllib3'...
remote: Enumerating objects: 12388, done.
remote: Total 12388 (delta 0), reused 0 (delta 0), pack-reused 12388
Receiving objects: 100% (12388/12388), 3.07 MiB | 2.24 MiB/s, done.
Resolving deltas: 100% (8699/8699), done.
Submodule path 'telegram/vendor/ptb_urllib3': checked out '1954df03958b164483282330b3a58092c070bc7a'
```

라즈베리파이 카메라 이용

TimerBot 수정

pi@raspberrypi: ~/python-telegram-bot/examples

File Edit Tabs Help

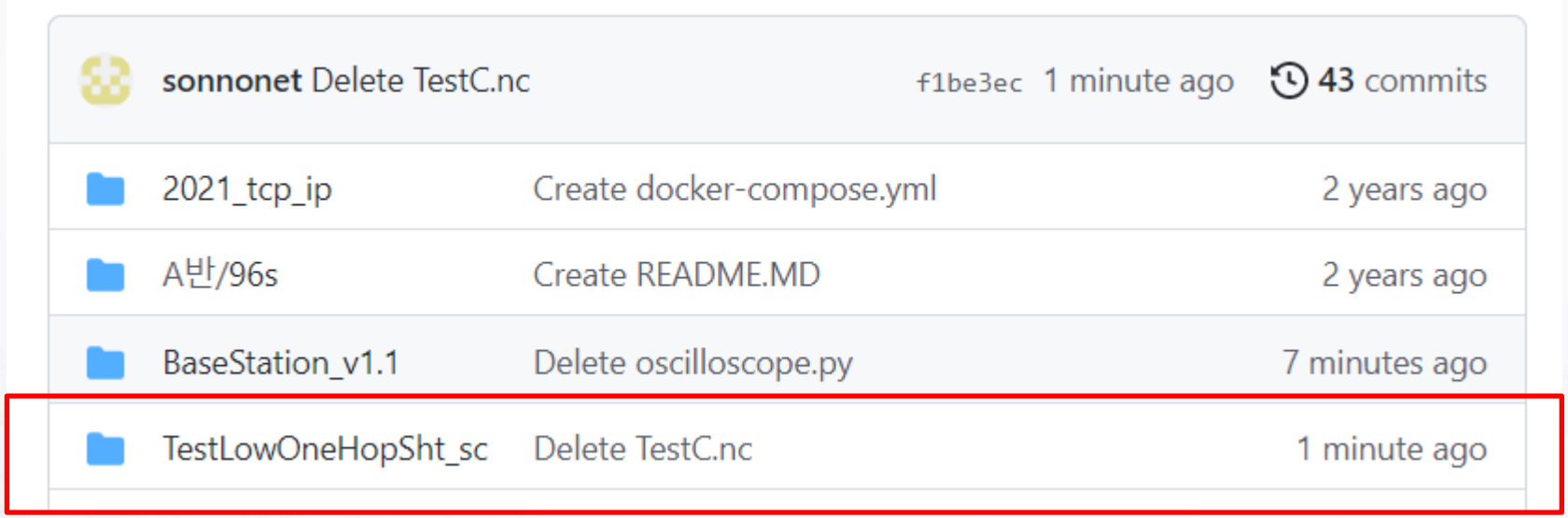
```
import logging

from telegram import Update
from telegram.ext import Application, CommandHandler, ContextTypes

# Enable logging
logging.basicConfig(
    format="%(asctime)s - %(name)s - %(levelname)s - %(message)s", level=logging.INFO
)


# Define a few command handlers. These usually take the two arguments update and
# context.
# Best practice would be to replace context with an underscore,
# since context is an unused local variable.
# This being an example and not having context present confusing beginners,
# we decided to have it present as context.
async def start(update: Update, context: ContextTypes.DEFAULT_TYPE) -> None:
    """Sends explanation on how to use the bot."""
    await update.message.reply_text("Hi! Use /set <seconds> to set a timer")


async def alarm(context: ContextTypes.DEFAULT_TYPE) -> None:
    """Send the alarm message."""
    job = context.job
    await context.bot.send_message(job.chat_id, text=f"Beep! {job.data} seconds are over!")


def remove_job_if_exists(name: str, context: ContextTypes.DEFAULT_TYPE) -> bool:
    """Remove job with given name. Returns whether job was removed."""
    current_jobs = context.job_queue.get_jobs_by_name(name)
    if not current_jobs:
        return False
    for job in current_jobs:
        job.schedule_removal()
    return True


async def set_timer(update: Update, context: ContextTypes.DEFAULT_TYPE) -> None:
    """Add a job to the queue."""
    chat_id = update.effective_message.chat_id
    try:
```

55,59 35%

TinyOS 기반 NesC 프로그래밍 (온습도센서)

- Github 소스코드 다운

- https://github.com/sonnonet/inhatc



TinyOS 기반 NesC 프로그래밍 (온습도센서)

TestAppC.nc

~/sonnonet_tinyos/Tinyos/TestLowOneHopSht_sc

```
1  includes Test;
2  configuration TestAppC
3  {
4  }
5  implementation
6  {
7    components TestC, MainC;
8    components LedsC, new TimerMilliC();
9
10   components ActiveMessageC as AMC;
11   components new AMSenderC(AM_TEST_DATA_MSG) as AMSC;
12
13   TestC.Boot -> MainC;
14   TestC.Leds -> LedsC;
15   TestC.MilliTimer -> TimerMilliC;
16
17   TestC.RadioControl -> AMC;
18   TestC.RadioSend -> AMSC;
19
20   components new SensirionSht11C() as Sht11ChOC;
21   TestC.Temp -> Sht11ChOC.Temperature;
22   TestC.Humi -> Sht11ChOC.Humidity;
23
24   components new IIluAdcC() as IIlu;
25   TestC.IIlu -> IIlu;
26
27   components BatteryC;
28   TestC.Battery -> BatteryC;
29 }
                                        16,0-1        All
```

TinyOS 기반 NesC 프로그래밍 (온습도센서)

Test.h

~/sonnonet_tinyos/Tinyos/TestLowOneHopSht_sc

```
#ifndef TEST_H
#define TEST_H
#include "message.h"
enum {
  TEST_PERIOD = 10240LU,
};
enum {
  DFLT_VAL = 0x11,
};
enum {
  TEST_DATA_LENGTH = TOSH_DATA_LENGTH - 6,
};
enum {
  AM_TEST_DATA_MSG = 0xA4,
};

typedef nx_struct test_data_msg {
  nx_am_addr_t srcID;
  nx_uint32_t seqNo;
  nx_uint16_t type;
  nx_uint16_t Temp;
  nx_uint16_t Humi;
  nx_uint16_t lllu;
  nx_uint16_t battery;
  //nx_uint8_t testData[TEST_DATA_LENGTH];
} test_data_msg_t;

#endif // TEST_H
```

TestC.nc

~/sonnonet_tinyos/Tinyos/TestLowOneHopSht_sc

```
1  module TestC
2  {
3    uses {
4
5        ?
6
7
8      interface SplitControl as RadioControl;
9      interface AMSend as RadioSend;
10
11     interface Read<uint16_t> as Temp;
12     interface Read<uint16_t> as Humi;
13     interface Read<uint16_t> as Illu;
14
15     interface Battery;
16   }
17 }
18
```

TinyOS 기반 NesC 프로그래밍 (온습도센서)

TestC.nc

~/sonnonet_tinyos/Tinyos/TestLowOneHopSht_sc

```
19 implementation
20 {
21   message_t testMsgBffr;
22   test_data_msg_t *testMsg;
23
24   uint32_t seqNo;
25   uint8_t step;
26
27
28   task void startTimer();
29   event void Boot.booted() {
30     testMsg = (test_data_msg_t *)call RadioSend.getPayload(
31       &testMsgBffr, sizeof(test_data_msg_t));
32     testMsg->srcID = TOS_NODE_ID;
33
34     seqNo = 0;
35
36     post startTimer();
37   }
38
39   task void startTimer() {
40     call MilliTimer.startPeriodic(TEST_PERIOD);
41   }
42
43   task void radioOn();
44   event void MilliTimer.fired() {
45     post radioOn();
46   }
```

TestC.nc

```
48   void startDone();
49   task void radioOn() {
50     if (call RadioControl.start() != SUCCESS) startDone();
51   }
52
53   event void RadioControl.startDone(error_t error) {
54     startDone();
55   }
56
57   task void readTask();
58   void startDone() {
59     step = 0;
60     post readTask();
61     call Leds.led0Toggle();
62   }
63
64   void sendDone();
65   task void sendTask() {
66     testMsg->seqNo = seqNo++;
67     testMsg->type = 2; //THL type 2
68
69     if (call RadioSend.send(AM_BROADCAST_ADDR, &testMsgBffr,
70       sizeof(test_data_msg_t)) != SUCCESS) sendDone();
71     call Leds.led2Toggle();
72   }
73
```

TestC.nc

```
74   event void RadioSend.sendDone(message_t* msg, error_t error) {
75     sendDone();
76   }
77
78   task void radioOff();
79   void sendDone() {
80     call Leds.led0Off();
81     call Leds.led1Off();
82     call Leds.led2Off();
83     post radioOff();
84   }
85
86   void stopDone();
87   task void radioOff() {
88     if (call RadioControl.stop() != SUCCESS) stopDone();
89   }
90
91   event void RadioControl.stopDone(error_t error) {
92     stopDone();
93   }
94
95   void stopDone() {
96   }
97
```

TestC.nc

```
97   task void readTask() {
98     switch(step) {
99       case 0:
100        call Temp.read(); break;
101      case 1:
102        call Humi.read(); break;
103      case 2:
104        call Illu.read(); break;
105      default:
106        testMsg->battery = call Battery.getVoltage();
107        post sendTask();
108        break;
109    }
110    step += 1;
111  }
112
113  event void Temp.readDone(error_t error, uint16_t val) {
114    //if (error != SUCCESS) call Leds.led0On();
115    testMsg->Temp = error == SUCCESS ? val : 0xFFFA;
116    post readTask();
117  }
118  event void Humi.readDone(error_t error, uint16_t val) {
119    //if (error != SUCCESS) call Leds.led1On();
120    testMsg->Humi = error == SUCCESS ? val : 0xFFFB;
121    post readTask();
122  }
123  event void Illu.readDone(error_t error, uint16_t val){
```

TestC.nc

```
123   event void Illu.readDone(error_t error, uint16_t val){
124     testMsg->Illu = error == SUCCESS ? val : 0xFFFC;
125     post readTask();
126   }
127 }
```

NesC 미들웨어 프로그래밍 (온습도센서 파싱)

oscilloscope.py

~/sono_raspberrypi/BaseStation_v1.1

```
# Data Format
# CO2  Data0
# THL Temperature Data0, Humidity Data1, Illumination Data2, Battery Data3

import sys
import tos
import datetime
import threading

AM_OSCILLOSCOPE = 0x93

class OscilloscopeMsg(tos.Packet):
    def __init__(self, packet = None):
        tos.Packet.__init__(self,
                            [('srcID',  'int', 2),
                             ('seqNo', 'int', 4),
                             ('type', 'int', 2),
                             ('Data0', 'int', 2),
                             ('Data1', 'int', 2),
                             ('Data2', 'int', 1),
                             ('Data3', 'int', 1),
                             ('Data4', 'int', 2),
                             ],
                            packet)
if '-h' in sys.argv:
    print "Usage:", sys.argv[0], "serial@/dev/ttyUSB0:57600"
    sys.exit()

```

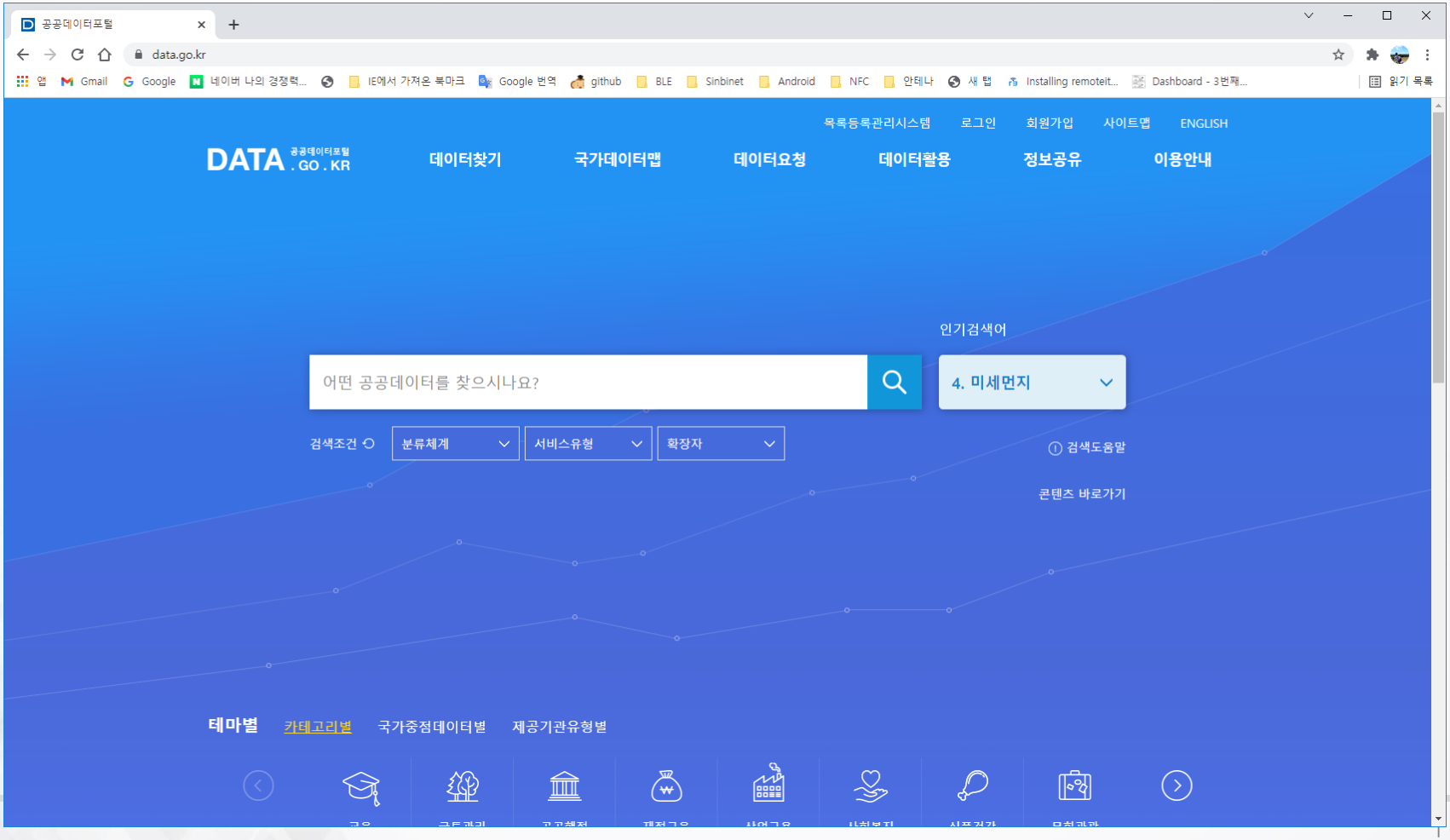
oscilloscope.py

```
29 am = tos.AM()
30
31 while True:
32     p = am.read()
33     msg = OscilloscopeMsg(p.data)
34     print p
35
36 ####### THL Logic ############
37     if msg.type == 2:
38         battery = msg.Data4
39
40         lllumi = int(msg.Data2)+ int(msg.Data3*256)
41         lllumi = lllumi
42         humi = -2.0468 + (0.0367*msg.Data1) + (-1.5955*0.000001)*msg.Data1*msg.Data1
43         temp = -(39.6) + (msg.Data0 * 0.01)
44         try:
45             with conn.cursor() as curs:
46                                 Now = datetime.datetime.now()
47                                 sql = """insert into JB_Sensor_THL(NODE_ID,SEQ,TEMPERATURE,HUMIDITY,ILLUMINATION,REGDATE)
   ₩
48                                     values(%s, %s, %s, %s, %s, %s)"""
49                                 curs.execute(sql,(msg.srcID,msg.seqNo,temp,humi,lllumi,Now))
50                                 conn.commit()
51         except all,e:
52             print e.args
53             conn.close()
54         print "id:" , msg.srcID, " Count : ", msg.seqNo, ₩
55                 "Temperature: ",temp, "Humidity: ",humi, "lllumination: ",lllumi, "Battery : ", battery
```

공공데이터 API - 미세먼지

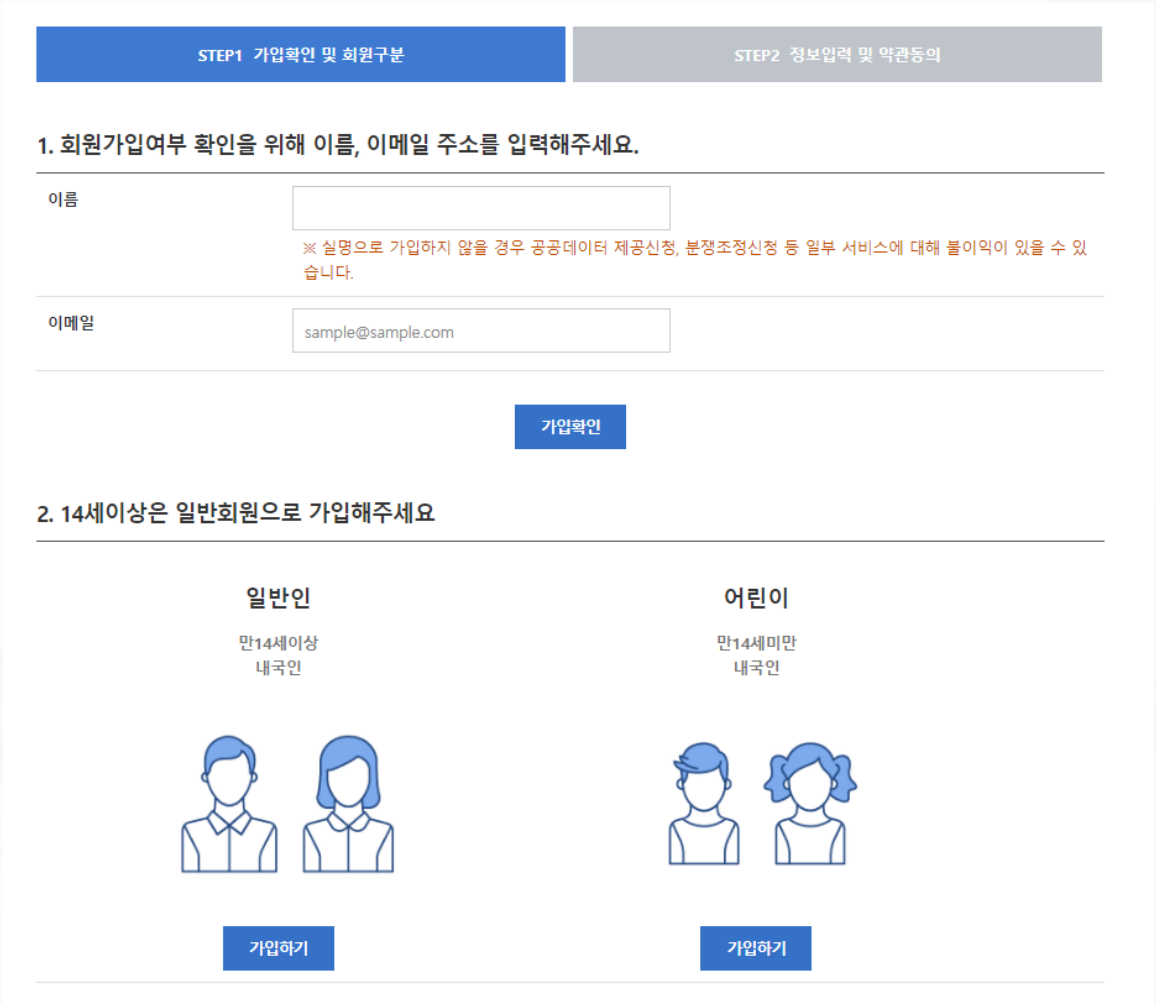
❖ 공공데이터 포털 URL

: https://www.data.go.kr



❖ 회원가입

: 이후 절차 생략


STEP1 가입확인 및 회원구분
STEP2 정보입력 및 약관동의
1. 회원가입여부 확인을 위해 이름, 이메일 주소를 입력해주세요.
이름
※ 실명으로 가입하지 않을 경우 공공데이터 제공신청, 분쟁조정신청 등 일부 서비스에 대해 불이익이 있을 수 있습니다.
이메일
sample@sample.com
가입확인
2. 14세이상은 일반회원으로 가입해주세요
일반인
만14세이상
내국인
어린이
만14세미만
내국인
가입하기
가입하기

5주차도 고생했습니다.