Portfolio of Kim Jin-hee as Android developers who specialize in data structure and program design through various majors and long university life

Contents

I. List of participated projects

- IDEA Dot. Participation in entrepreneurship (online competition system design) – 2019/01 ~ 2019/04
- Development of LoRa based wireless energy meter system (firmware, tree structure, Linux, C, Python) – 2017/09 ~ 2018/08
- Development of ICT based fusion polar environment monitoring system (Data analysis, GUI, Java) - 2017/03 ~ 2018/08
- Dept of Electronics and Communication Engineering Capstone Design (**Android**, SQLite, SQL, Java, C++) – 2015/09 ~ 2016/05
- Cource - Microprocessor application, character code generator (Java, C) – 2015/03 ~ 2015/06
- Dx-Ball game imitation (Windows API, Linked List, C/C++) – 2012/07 ~ 2012/08

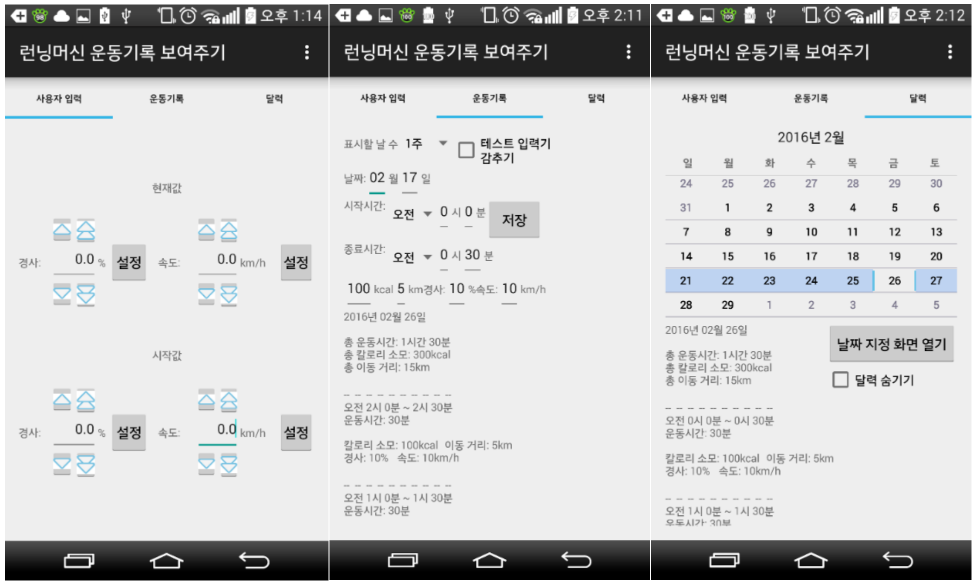
I have been working on a LoRa-based wireless energy meter development project that I have been doing since graduate student, and I have been using data structures in tree form or using dictionary structures.

I once had a project that developed Android apps (electronic communication engineering and capstone design).

II. Android development project

Dept of Electronics and Communication Engineering Capstone Design – 2015/09 ~ 2016/05

- An application that calculates the Bluetooth signal strength to check whether a user is on a fitness equipment and automatically sets the information input to the fitness equipment



2016년 02월 26일

총 운동시간: 1시간 30분
총 칼로리 소모: 300kcal
총 이동 거리: 15km

- - - - - - - - - - -
오전 2시 0분 ~ 2시 30분
운동시간: 30분

칼로리 소모: 100kcal 이동 거리: 5km
경사: 10% 속도: 10km/h

- - - - - - - - - - -
오전 1시 0분 ~ 1시 30분
운동시간: 30분

일	월	화	수	목	금	토
24	25	26	27	28	29	30
31	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	1	2	3	4	5

2016년 02월 26일

총 운동시간: 1시간 30분
총 칼로리 소모: 300kcal
총 이동 거리: 15km

- - - - - - - - - - -
오전 0시 0분 ~ 0시 30분
운동시간: 30분

칼로리 소모: 100kcal 이동 거리: 5km
경사: 10% 속도: 10km/h

- - - - - - - - - - -
오전 1시 0분 ~ 1시 30분
운동시간: 30분

Implemented features

- Bluetooth signal strength calculation and user location recognition
- Bluetooth signal strength calculation and user location recognition
- Save to SQLite database
- Reading from an SQLite database
- Scroll tabs with left and right swipes, scroll
- All UI component functions

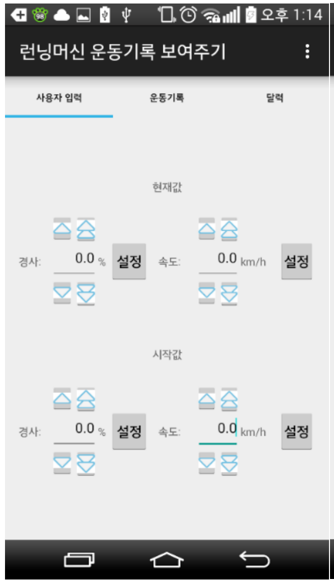
The source code that produces the same results as the screenshot is lost, and the current source code is in a state where bug fixes are required.

https://github.com/kjh0311/kim_jin_hee-projects/tree/master/2016%20-%20Android%2C%20Data%20Science

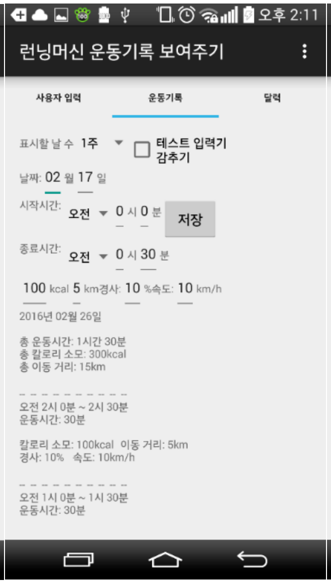
II. Android development project

Dept of Electronics and Communication Engineering Capstone Design – 2015/09 ~ 2016/05

- An application that calculates the Bluetooth signal strength to check whether a user is on a fitness equipment and automatically sets the information input to the fitness equipment



경사: 0.0 % 속도: 0.0 km/h
경사: 0.0 % 속도: 0.0 km/h



표시할 날 수 1주

날짜: 02 월 17 일

시작시간: 오전 0 시 0 분

종료시간: 오전 0 시 30 분

100 kcal 5 km경사: 10 %속도: 10 km/h

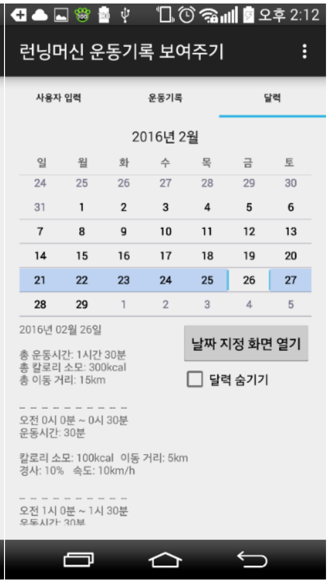
2016년 02월 26일

총 운동시간: 1시간 30분
총 칼로리 소모: 300kcal
총 이동 거리: 15km

- - - - - - - - - - -
오전 2시 0분 ~ 2시 30분
운동시간: 30분

칼로리 소모: 100kcal 이동 거리: 5km
경사: 10% 속도: 10km/h

- - - - - - - - - - -
오전 1시 0분 ~ 1시 30분
운동시간: 30분



2016년 2월

일	월	화	수	목	금	토
24	25	26	27	28	29	30
31	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	1	2	3	4	5

2016년 02월 26일

총 운동시간: 1시간 30분
총 칼로리 소모: 300kcal
총 이동 거리: 15km

- - - - - - - - - - -
오전 0시 0분 ~ 0시 30분
운동시간: 30분

칼로리 소모: 100kcal 이동 거리: 5km
경사: 10% 속도: 10km/h

- - - - - - - - - - -
오전 1시 0분 ~ 1시 30분
운동시간: 30분

Development tools

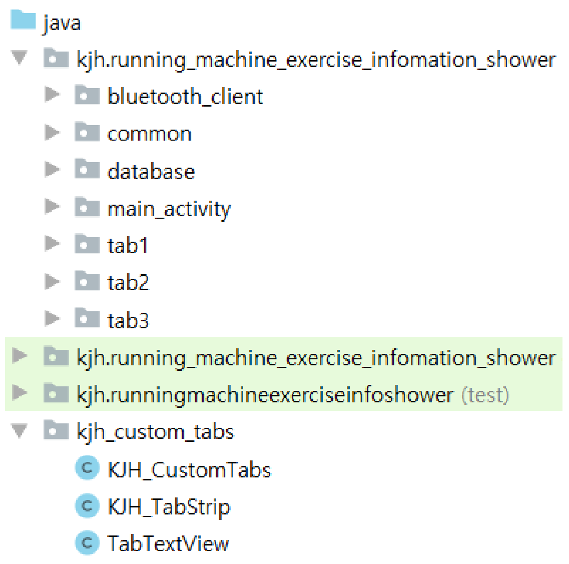
- Android Studio
- Android SDK Platform 21 (revision: 2)
- Build Tools: 23.0.2
- Gradle: 3.2.1

II. Android development project

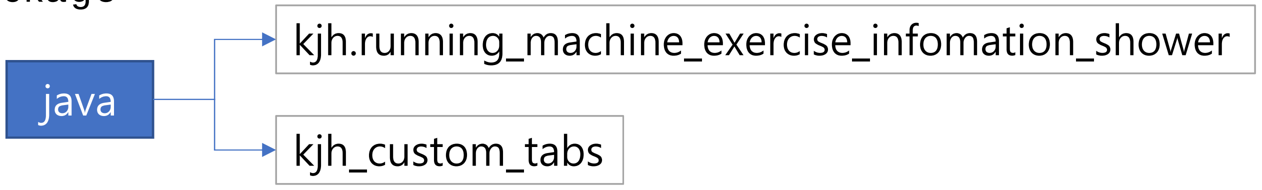
Dept of Electronics and Communication Engineering Capstone Design – 2015/09 ~ 2016/05

- An application that calculates the Bluetooth signal strength to check whether a user is on a fitness equipment and automatically sets the information input to the fitness equipment

Code Structure – Java top level package



```
java
▼ kjh.running_machine_exercise_infomation_shower
    ▶ bluetooth_client
    ▶ common
    ▶ database
    ▶ main_activity
    ▶ tab1
    ▶ tab2
    ▶ tab3
▶ kjh.running_machine_exercise_infomation_shower
▶ kjh.runningmachineexerciseinfoshower (test)
▼ kjh_custom_tabs
    KJH_CustomTabs
    KJH_TabStrip
    TabTextView
```



The java file contains a total of 7 packages in the kjh.running_machine_exercise_infomation_shower package, and a kjh_custom_tabs package outside of that package.

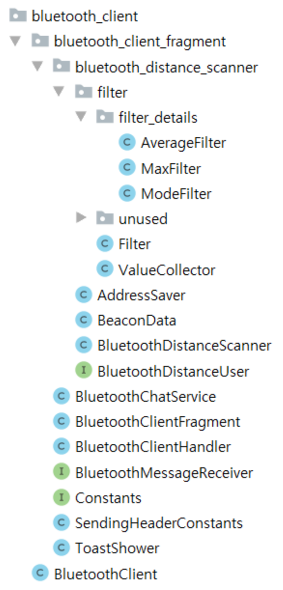
The class in the kjh_custom_tabs package is a class created by adding and modifying SlidingTabLayout.java, the source file provided by Google.
It is a module that allows you to drag and use from different places,
I explained how to use it as a Korean comment at the top of the source file.

II. Android 개발 프로젝트

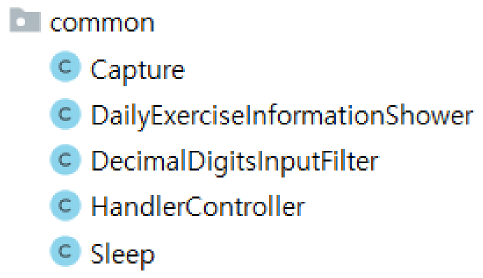
Dept of Electronics and Communication Engineering Capstone Design – 2015/09 ~ 2016/05

- An application that calculates the Bluetooth signal strength to check whether a user is on a fitness equipment and automatically sets the information input to the fitness equipment

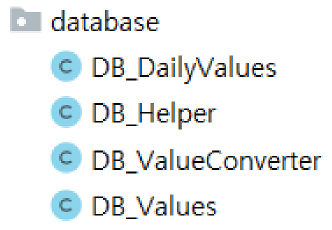
Code Structure – Internal of packages



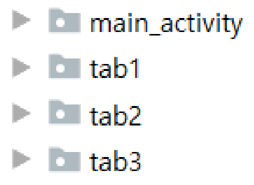
```
bluetooth_client
▼ bluetooth_client_fragment
  ▼ bluetooth_distance_scanner
    ▼ filter
      ▼ filter_details
          AverageFilter
          MaxFilter
          ModeFilter
      ▶ unused
        Filter
        ValueCollector
      AddressSaver
      BeaconData
      BluetoothDistanceScanner
      BluetoothDistanceUser
    BluetoothChatService
    BluetoothClientFragment
    BluetoothClientHandler
    BluetoothMessageReceiver
    Constants
    SendingHeaderConstants
    ToastShower
  BluetoothClient
```



```
common
  Capture
  DailyExerciseInformationShower
  DecimalDigitsInputFilter
  HandlerController
  Sleep
```



```
database
  DB_DailyValues
  DB_Helper
  DB_ValueConverter
  DB_Values
```



```
▶ main_activity
▶ tab1
▶ tab2
▶ tab3
```

bluetooth_client: Location calculation algorithm and message transmission / reception function using Bluetooth

common: Capture screens, show content imported from database, etc. Features that will be common across multiple tabs

database: Database reads and writes

main_activity: the starting point of this app

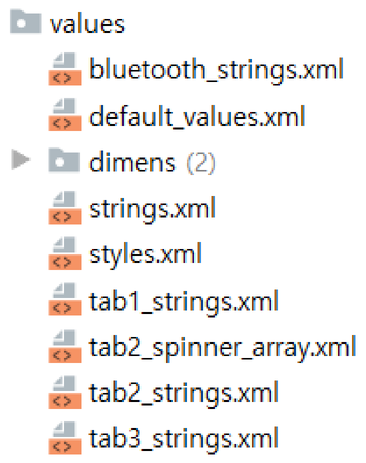
tab1,2,3: the part that connects UI and function of each tab

II. Android development project

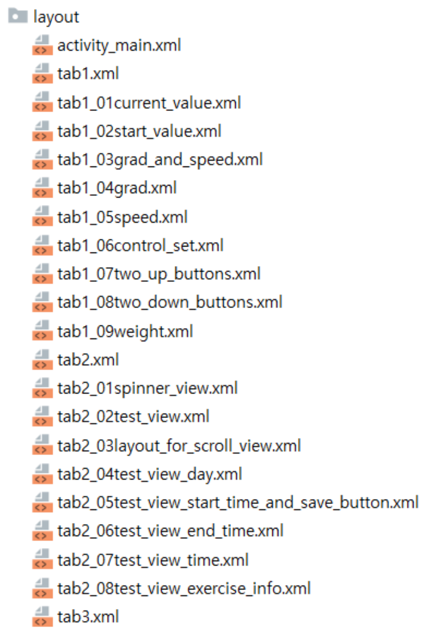
Dept of Electronics and Communication Engineering Capstone Design – 2015/09 ~ 2016/05

- An application that calculates the Bluetooth signal strength to check whether a user is on a fitness equipment and automatically sets the information input to the fitness equipment

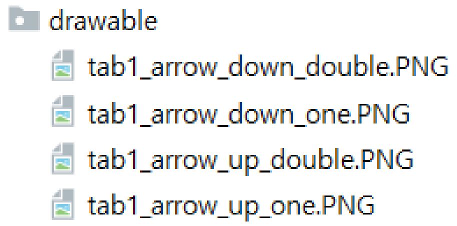
Code Structure – UI



```
values
    bluetooth_strings.xml
    default_values.xml
  ▶ dimens (2)
    strings.xml
    styles.xml
    tab1_strings.xml
    tab2_spinner_array.xml
    tab2_strings.xml
    tab3_strings.xml
```



```
layout
    activity_main.xml
    tab1.xml
    tab1_01current_value.xml
    tab1_02start_value.xml
    tab1_03grad_and_speed.xml
    tab1_04grad.xml
    tab1_05speed.xml
    tab1_06control_set.xml
    tab1_07two_up_buttons.xml
    tab1_08two_down_buttons.xml
    tab1_09weight.xml
    tab2.xml
    tab2_01spinner_view.xml
    tab2_02test_view.xml
    tab2_03layout_for_scroll_view.xml
    tab2_04test_view_day.xml
    tab2_05test_view_start_time_and_save_button.xml
    tab2_06test_view_end_time.xml
    tab2_07test_view_time.xml
    tab2_08test_view_exercise_info.xml
    tab3.xml
```



```
drawable
    tab1_arrow_down_double.PNG
    tab1_arrow_down_one.PNG
    tab1_arrow_up_double.PNG
    tab1_arrow_up_one.PNG
```

drawable: arrow icon files visible in the first tab

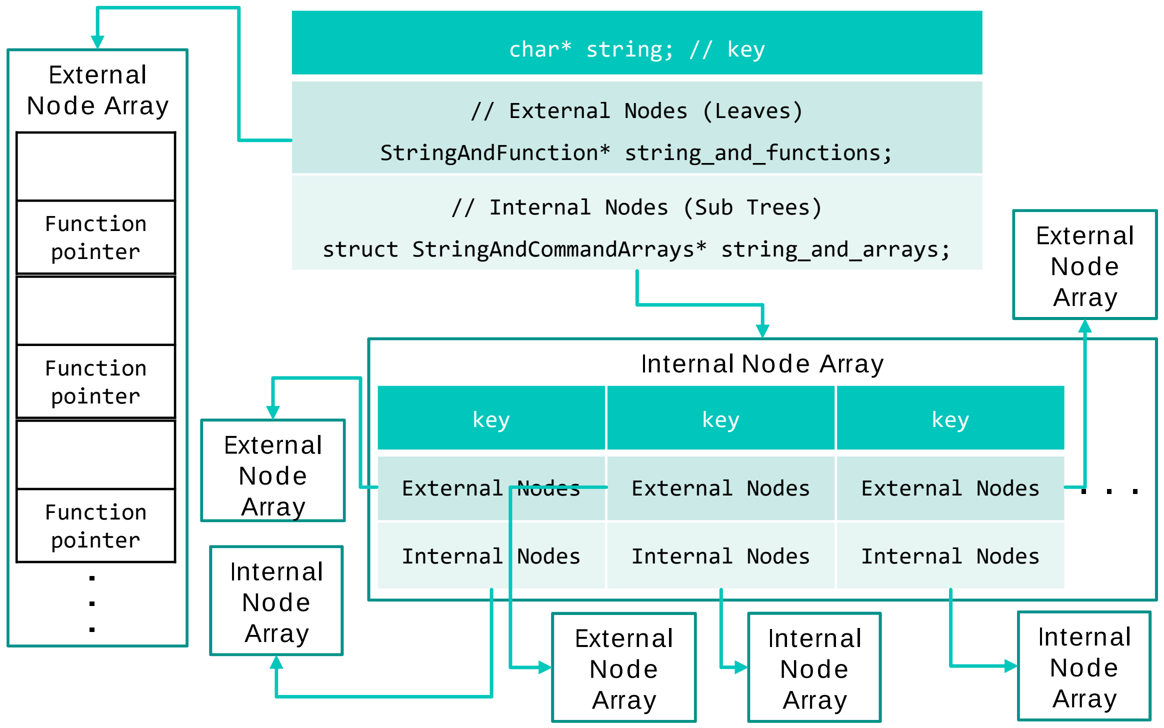
values: Characters to display on screen

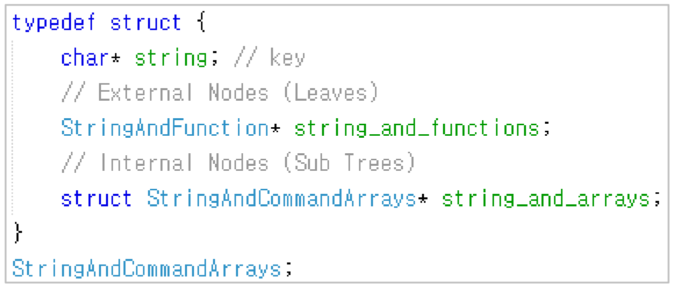
layout: Defines the component to be displayed on the screen

III. Design and application of data structure

Development of LoRa based wireless energy meter – 2017/09 ~ 2018/08

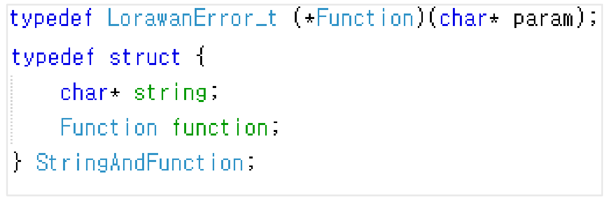
○ Command Interpreter Using Tree Structure - Design (C)





```
typedef struct {
    char* string; // key
    // External Nodes (Leaves)
    StringAndFunction* string_and_functions;
    // Internal Nodes (Sub Trees)
    struct StringAndCommandArrays* string_and_arrays;
}
StringAndCommandArrays;
```

Implementation of internal node



```
typedef LorawanError_t (*Function)(char* param);
typedef struct {
    char* string;
    Function function;
} StringAndFunction;
```

Implementation of external node

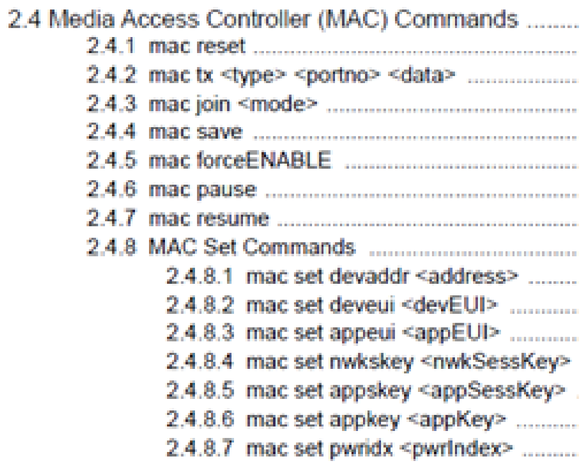
The data structure was designed in tree form to develop the command interpreter, and the tree was implemented using a pointer to the structure, function pointer, and arrangement.

https://github.com/kjh0311/kim_jin_hee-projects/tree/master/2017~2018%20-%20Graduate

III. Design and application of data structure

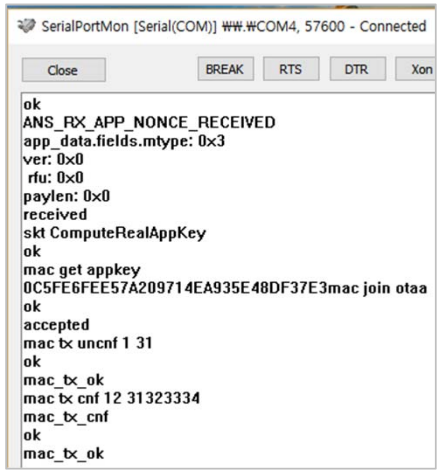
Development of LoRa based wireless energy meter – 2017/09 ~ 2018/08

Command Interpreter Using Tree Structure - Application (C)



2.4 Media Access Controller (MAC) Commands ....
    2.4.1 mac reset ....
    2.4.2 mac tx <type> <portno> <data> ....
    2.4.3 mac join <mode> ....
    2.4.4 mac save ....
    2.4.5 mac forceENABLE ....
    2.4.6 mac pause ....
    2.4.7 mac resume ....
    2.4.8 MAC Set Commands ....
        2.4.8.1 mac set devaddr <address> ....
        2.4.8.2 mac set deveui <devEUI> ....
        2.4.8.3 mac set appeui <appEUI> ....
        2.4.8.4 mac set nwkskey <nwkSessKey>
        2.4.8.5 mac set appskey <appSessKey>
        2.4.8.6 mac set appkey <appKey> ....
        2.4.8.7 mac set pwridx <pwrIndex> ....

Collection of commands to implement (partial)



```
ok
ANS_RX_APP_NONCE_RECEIVED
app_data.fields.mtype: 0x3
ver: 0x0
 rfu: 0x0
paylen: 0x0
received
skt ComputeRealAppKey
ok
mac get appkey
0C5FE6FEE57A209714EA935E48DF37E3mac join otaa
ok
accepted
mac tx uncnf 1 31
ok
mac_tx_ok
mac tx cnf 12 31323334
mac_tx_cnf
ok
mac_tx_ok
```

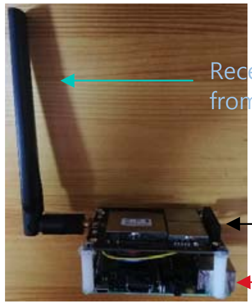
Test results from SKT test bed

- The picture on the left is part of Microchip's LoRa Development Kit manual.
- For that information, I designed and developed a command interpreter myself, and I used a command interpreter that I developed in SKT's test bed to make sure that LoRaWAN specification is implemented according to the KR region parameter.
- **'skt ComputeRealAppKey'** command is a command created for use in the SKT test bed.

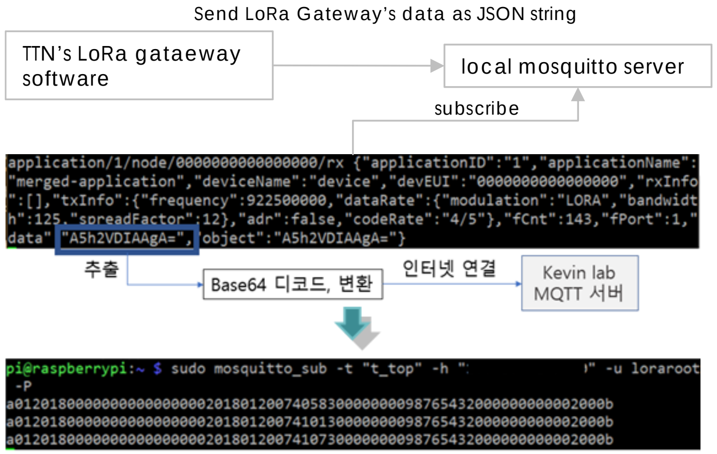
III. Design and application of data structure

Development of LoRa based wireless energy meter – 2017/09 ~ 2018/08

JSON <-> Dictionary transform (Python)



Hardware Connection



```
application/1/node/0000000000000000/rx {"applicationID":"1","applicationName":
"merged-application","deviceName":"device","devEUI":"0000000000000000","rxInfo
":[],"txInfo":{"frequency":922500000,"dataRate":{"modulation":"LORA","bandwidt
h":125,"spreadFactor":12},"adr":false,"codeRate":"4/5"},"fCnt":143,"fPort":1,"
data":"A5h2VDIAAgA=","object":"A5h2VDIAAgA="}
```

```
pi@raspberrypi:~ $ sudo mosquitto_sub -t "t_top" -h "[redacted]" -u loraroot
 -P
a01201800000000000000000201801200740583000000000987654320000000000000002000b
a01201800000000000000000201801200741013000000000987654320000000000000002000b
a01201800000000000000000201801200741073000000000987654320000000000000002000b
```

Result of subscription of external Mosquitto server which data received

- TTN's gateway software outputs a JSON string of packets received by LoRa Gateway on the local Mosquitto server, and this program decodes data encoded with BASE64 from the JSON string, converts the data into strings in the form printed in the following figure, and sends it to an external Mosquitto server on the Internet.

  The following python code converts the data passed to JSON into Dictionary to decode base64 data corresponding to the key 'data'.

```python
json_string = get_json_string(mqtt_line)
dictionary = json.loads(json_string)
```

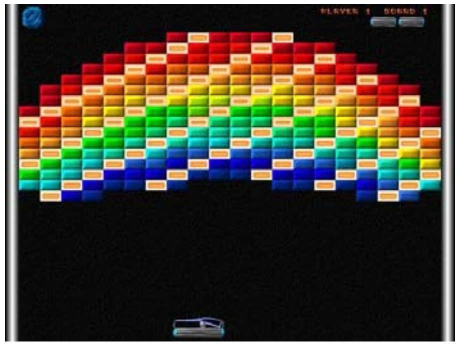
```python
base64_data = dictionary['data']
hex_received_data = base64.b64decode(base64_data)
```

- By developing this program, I learned that communication between systems is an effective way to communicate in JSON, that HEX data is encoded in Base64 and contained within JSON strings, and that JSON strings are converted into a Dictionary structure to be accessed and treated as key values.

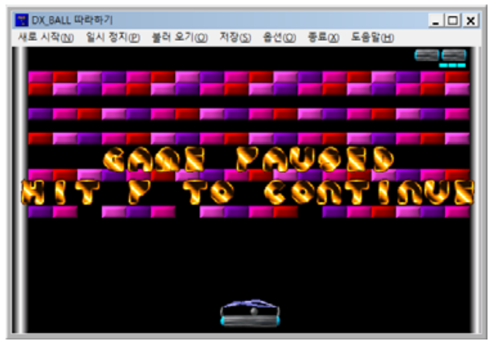
III. Design and application of data structure

Dx-Ball game imitation – 2012/07 ~ 2012/08

- Linked List (C/C++)

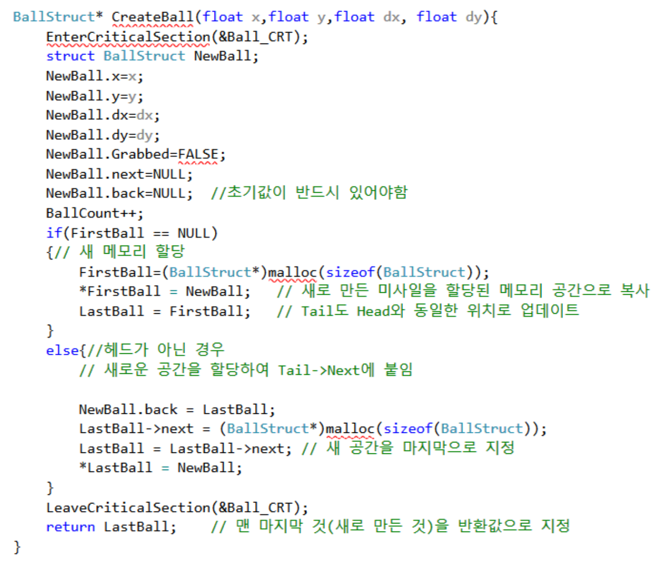


Original


DX_BALL 따라하기
새로 시작(N) 일시 정지(P) 불러 오기(O) 저장(S) 옵션(O) 종료(X) 도움말(H)
GAME PAUSED
HIT P TO CONTINUE

Imitation

The code on the right is the code that adds the number of balls when an item is acquired in the game. This code is implemented using the Linked List.



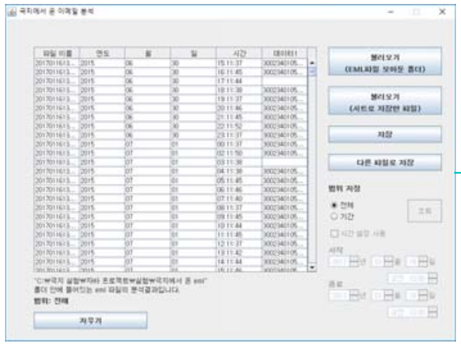
```
BallStruct* CreateBall(float x,float y,float dx, float dy){
    EnterCriticalSection(&Ball_CRT);
    struct BallStruct NewBall;
    NewBall.x=x;
    NewBall.y=y;
    NewBall.dx=dx;
    NewBall.dy=dy;
    NewBall.Grabbed=FALSE;
    NewBall.next=NULL;
    NewBall.back=NULL;  //초기값이 반드시 있어야함
    BallCount++;
    if(FirstBall == NULL)
    {// 새 메모리 할당
        FirstBall=(BallStruct*)malloc(sizeof(BallStruct));
        *FirstBall = NewBall;   // 새로 만든 미사일을 할당된 메모리 공간으로 복사
        LastBall = FirstBall;   // Tail도 Head와 동일한 위치로 업데이트
    }
    else{//헤드가 아닌 경우
        // 새로운 공간을 할당하여 Tail->Next에 붙임

        NewBall.back = LastBall;
        LastBall->next = (BallStruct*)malloc(sizeof(BallStruct));
        LastBall = LastBall->next; // 새 공간을 마지막으로 지정
        *LastBall = NewBall;
    }
    LeaveCriticalSection(&Ball_CRT);
    return LastBall;    // 맨 마지막 것(새로 만든 것)을 반환값으로 지정
}
```

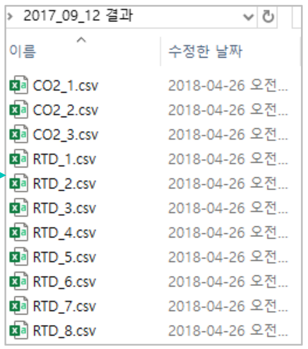
https://github.com/kjh0311/kim_jin_hee-projects/tree/master/2012%20(Beginner)%20-%20Imitation%20of%20DX-Ball%20Game

IV. Design program

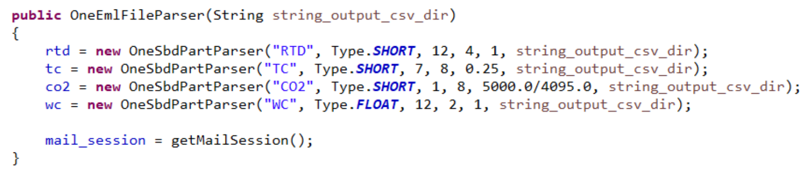
Development of ICT based fusion polar environment monitoring system - 2017/03 ~ 2018/08

- **Develop a program that reads a large number of email files (EML) and automatically create spreadsheet files (CSV) with the required data, GUI development (Java)**





```
2017_09_12 결과
이름            수정한 날짜
CO2_1.csv       2018-04-26 오전...
CO2_2.csv       2018-04-26 오전...
CO2_3.csv       2018-04-26 오전...
RTD_1.csv       2018-04-26 오전...
RTD_2.csv       2018-04-26 오전...
RTD_3.csv       2018-04-26 오전...
RTD_4.csv       2018-04-26 오전...
RTD_5.csv       2018-04-26 오전...
RTD_6.csv       2018-04-26 오전...
RTD_7.csv       2018-04-26 오전...
RTD_8.csv       2018-04-26 오전...
```



```java
public OneEmlFileParser(String string_output_csv_dir)
{
    rtd = new OneSbdPartParser("RTD", Type.SHORT, 12, 4, 1, string_output_csv_dir);
    tc = new OneSbdPartParser("TC", Type.SHORT, 7, 8, 0.25, string_output_csv_dir);
    co2 = new OneSbdPartParser("CO2", Type.SHORT, 1, 8, 5000.0/4095.0, string_output_csv_dir);
    wc = new OneSbdPartParser("WC", Type.FLOAT, 12, 2, 1, string_output_csv_dir);

    mail_session = getMailSession();
}
```

Simplify logic with classes and constructors

The OneSbdPartParser class is used to extract various types of sensor data and save it as a file.

- This program allows you to import e-mail files that record sensor-acquisition data for climate environments from the polar regions into folders and organize them into csv files by data.

https://github.com/kjh0311/kim_jin_hee-projects/tree/master/2017~2018%20-%20Graduate

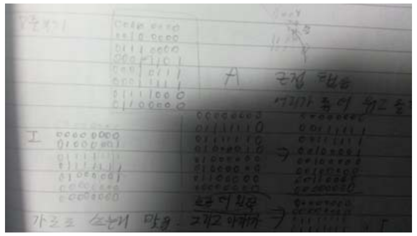
IV. Design program

Cource - Microprocessor application, character code generator – 2015/03 ~ 2015/06

○ **Process for creating a character code generator (Java)**

1. Write the letter A on the paper and analyze it.

   const char font_A[8] = { 0x00,0x40,0x70,0x1D,0x17,0x1F,0x78,0x60 }; // A

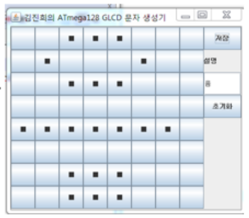


This is a picture taken at the time of development

2. Implement paper analysis to Java as follows:

- 만들어 져야 할 코드
{0X10, 0X12, 0XD5, 0XD5, 0XD5, 0X12, 0X10, 0X00}
- 위 배열은 각 행 별로 값이 있으며, 그 값은
그 행에 어떤 열에 점을 찍어야 할 지를 보여준다.

```java
int result[] = new int[Main.cols];
for(int i=0;i<Main.rows;i++){
  for(int j=0;j<Main.cols;j++){
    if (Main.record[i][j]){ // 점 찍은 경우
      result[j] |= 1 << i;
    }
  }
}
```



```java
private String getHex(int iv){
  String hex = "";
  hex = Integer.toHexString(iv).toUpperCase();
  if (hex.length()==1)
    hex = "0"+hex;
    hex = "0X"+hex;
  return hex;
}
```

```java
String s = "{";
String explainText;
s += getHex(result[0]);
for (int i=1; i<Main.rows;i++){
  s += ", ";
  s += getHex(result[i]);
}
s += "};";
```

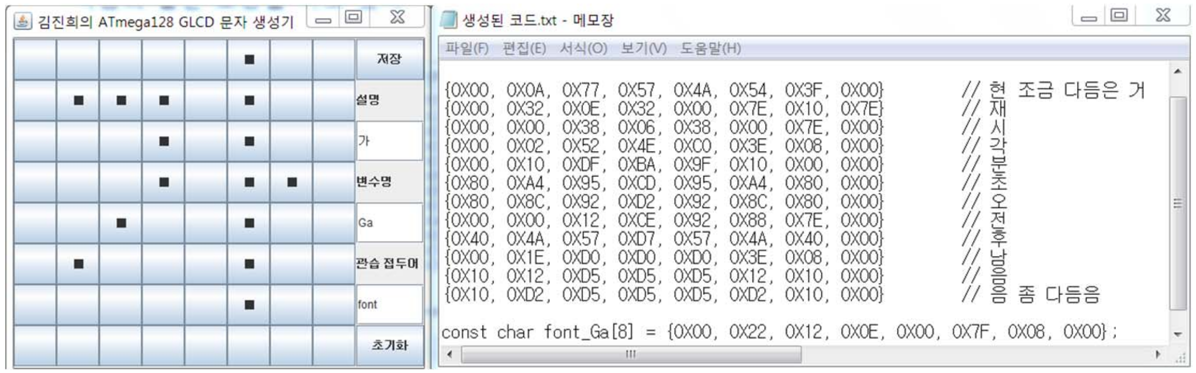
https://github.com/kjh0311/kim_jin_hee-projects/tree/master/2015%20-%20Microprocessor/Application%20of%20microprocessor

Ⅳ. Design program

Cource - Microprocessor application, character code generator – 2015/03 ~ 2015/06

- **character code generator (Java)**
  - To display Han-gul characters, we created a character code generator by analyzing the following code provided to express the font as well as alphabet A.

    const char font_A[8] = { 0x00,0x40,0x70,0x1D,0x17,0x1F,0x78,0x60 }; // A

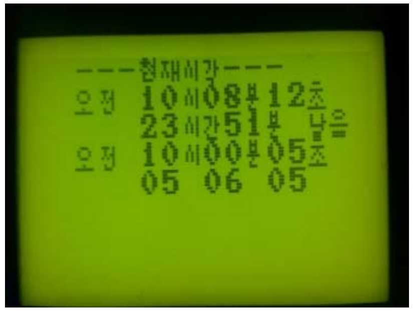

김진희의 ATmega128 GLCD 문자 생성기
저장
초기화
생성된 코드.txt - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

설명: 가

변수명: Ga

관습 접두어: font

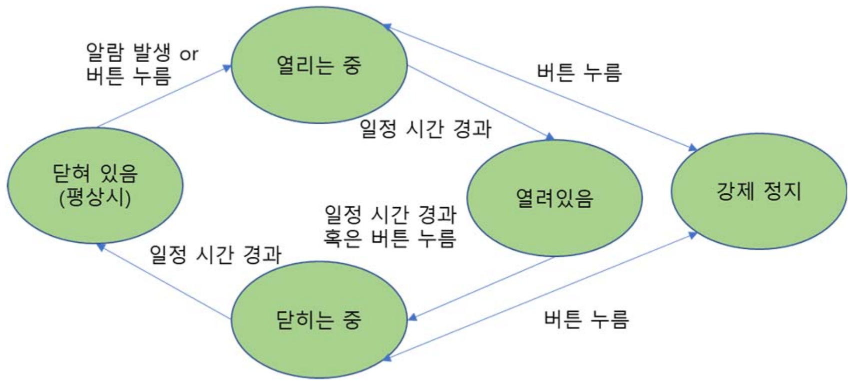
```
{0X00, 0X0A, 0X77, 0X57, 0X4A, 0X54, 0X3F, 0X00}          // 현 조금 다듬은 거
{0X00, 0X32, 0X0E, 0X32, 0X00, 0X7E, 0X10, 0X7E}          // 재
{0X00, 0X00, 0X38, 0X06, 0X38, 0X00, 0X7E, 0X00}          // 시
{0X00, 0X02, 0X52, 0X4E, 0XC0, 0X3E, 0X08, 0X00}          // 각
{0X00, 0X10, 0XDF, 0XBA, 0X9F, 0X10, 0X00, 0X00}          // 분
{0X80, 0XA4, 0X95, 0XCD, 0X95, 0XA4, 0X80, 0X00}          // 초
{0X80, 0X8C, 0X92, 0XD2, 0X92, 0X8C, 0X80, 0X00}          // 오
{0X00, 0X00, 0X12, 0XCE, 0X92, 0X88, 0X7E, 0X00}          // 전
{0X40, 0X4A, 0X57, 0XD7, 0X57, 0X4A, 0X40, 0X00}          // 후
{0X00, 0X1E, 0XD0, 0XD0, 0XD0, 0X3E, 0X08, 0X00}          // 남
{0X10, 0X12, 0XD5, 0XD5, 0XD5, 0X12, 0X10, 0X00}          // 음
{0X10, 0XD2, 0XD5, 0XD5, 0XD5, 0XD2, 0X10, 0X00}          // 음 좀 다듬음

const char font_Ga[8] = {0X00, 0X22, 0X12, 0X0E, 0X00, 0X7F, 0X08, 0X00};
```



IV. Design program

Cource - Microprocessor application, character code generator – 2015/03 ~ 2015/06

Introduction of FSM (Finite State Machine) graph into the design of step motor operating program - 1 (C)



```
#define      DOOR_CLOSED    0
#define      DOOR_OPENING 1
#define      DOOR_OPENED   2
#define      DOOR_CLOSING  3
// 버튼을 누를 경우 발생
#define      DOOR_STOPPED 4

typedef unsigned char State;
typedef struct
{
    State mode;
    SecondAndTenMili openingStartTime;
    SecondAndTenMili openedStartTime;
    SecondAndTenMili closingStartTime;
} DoorInfo;
```

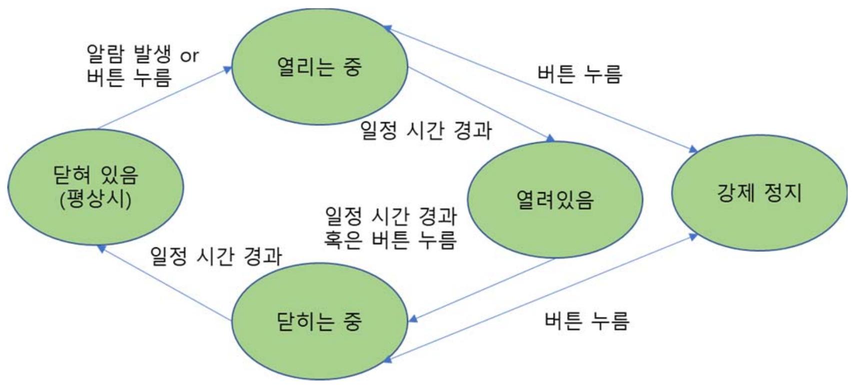
```
DoorInfo Door = {DOOR_CLOSED, 0, 0};
ShortInt DoorControl(bool keyPressed, ShortInt keyBuf)
{
        if ( (keyPressed && keyBuf == DOOR_BUTTON) ||
                    doorTimeExpired(&Door)      ) {
              Door.mode = ToNextState(&Door, keyPressed);
        }
        return DoorControlForState(&Door);
}
```

I drew a graph as on the left and implemented it on the right. (The implementation continues until the next page)

IV. Design program

Cource - Microprocessor application, character code generator – 2015/03 ~ 2015/06

- **Introduction of FSM (Finite State Machine) graph into the design of step motor operating program - 2 (C)**



```
bool doorTimeExpired(DoorInfo *pInfo)
{
    switch(pInfo->mode)
    {
    case DOOR_CLOSED:
        return checkAlarmTime();
    case DOOR_OPENING:
        return checkDoorOpeningTime
            (pInfo->openingStartTime);
    case DOOR_OPENED:
        return checkDoorOpenedTime
            (pInfo->openedStartTime);
    case DOOR_CLOSING:
        return checkDoorClosingTime
            (pInfo->closingStartTime);
    default:
        return false;
    }
}
```

```
State ToNextState(DoorInfo *pInfo, ShortInt keyPressed)
{
    switch(pInfo->mode){
    case DOOR_CLOSED:
        return openStart(pInfo);
    case DOOR_OPENING:
        return openedStart(pInfo, keyPressed);
    case DOOR_STOPPED: // 강제로 멈춘경우
    case DOOR_OPENED:
        return closeStart(pInfo, keyPressed);
    case DOOR_CLOSING:
        return closingToNext(keyPressed);
    default:
        return pInfo->mode;
    }
}
```

Thanks