

System Programming

Final Project Report
Team-1

Submission date : December 12, 2016 Monday

Professor : Ko Jeong Gil

T/A : Kim Nam Kyung

Team : 1

Team member : Software 201423050 Jeong Jin Hwan

Software 20142304 Nam HwaSu

Software 201220962 Lee InTae

Mechanical Engineering 201021531 Lee Byung Jae

Contents

1. Project Motivation

2. Algorithm's outline

3. Explanation each equipments & tools

4. Scenario

5. Not implementation factor

6. Member's Role

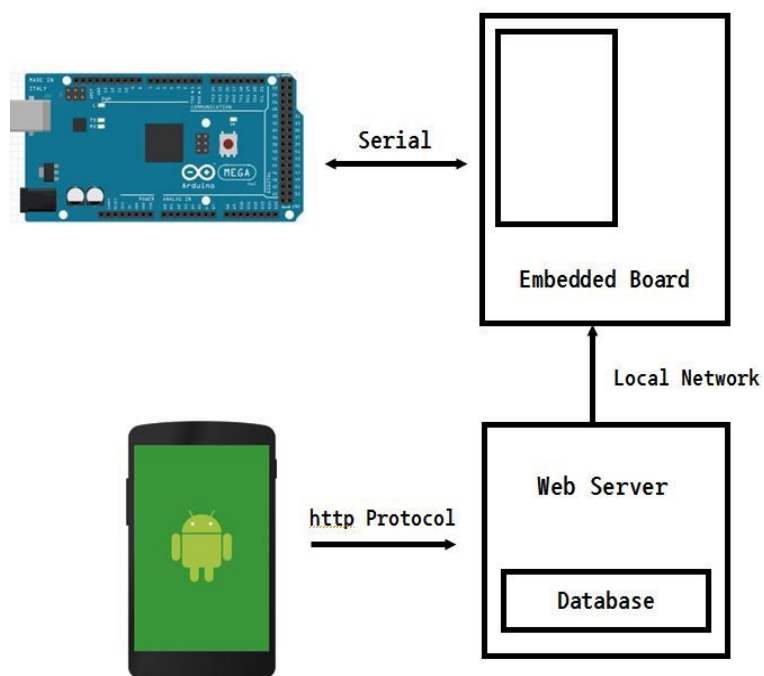
1. Project Motivation

Many people find it difficult to get up. Conventional alarms wake we up, but it does not get we out of bed. So we thought that we have to make a practical alarm. The function of the alarm should focus on making us move. It can be a body or a head. So our team devised this project. The main goal of this project is to develop alarm machines using games. A user must be able to select the type of game that is right for him / her.

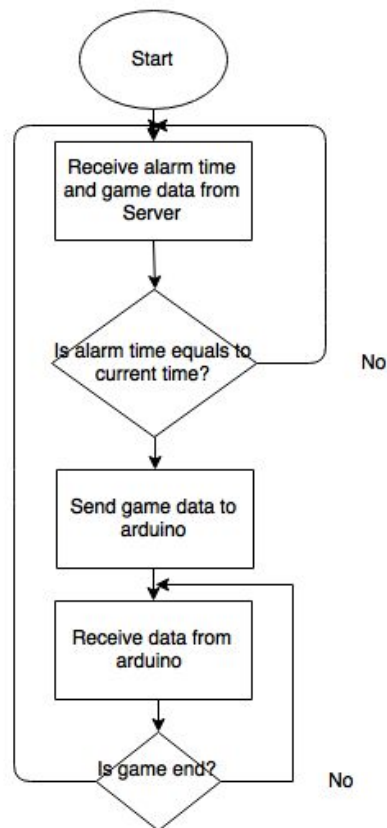
This project machine's name is "The One Alarm". We got the idea from the 'Lord of the Rings'. We look forward to making the morning a little easier with this machine.

2. Algorithm's outline

- System structure

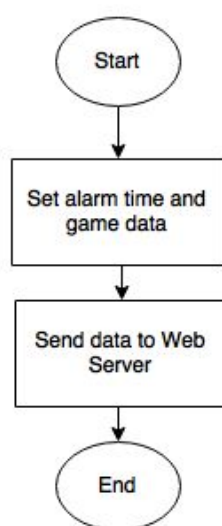


- Embedded Board

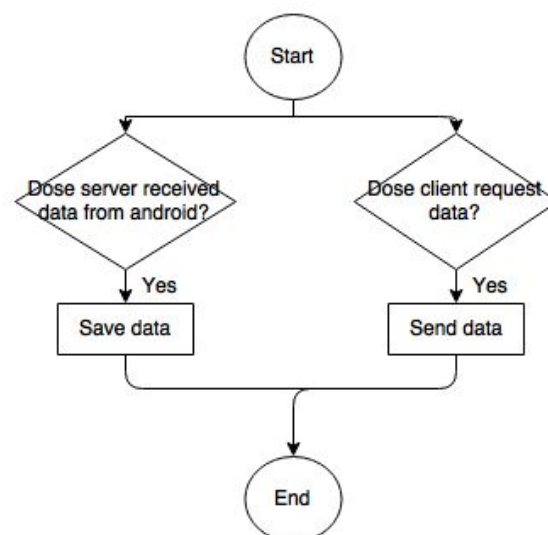


- Android & Database

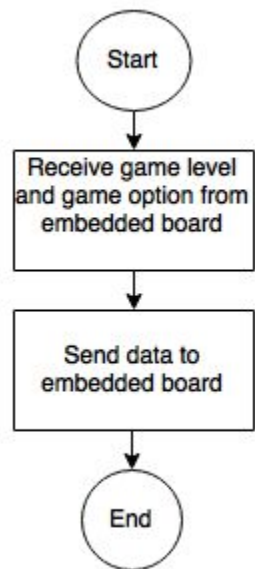
[1] Android



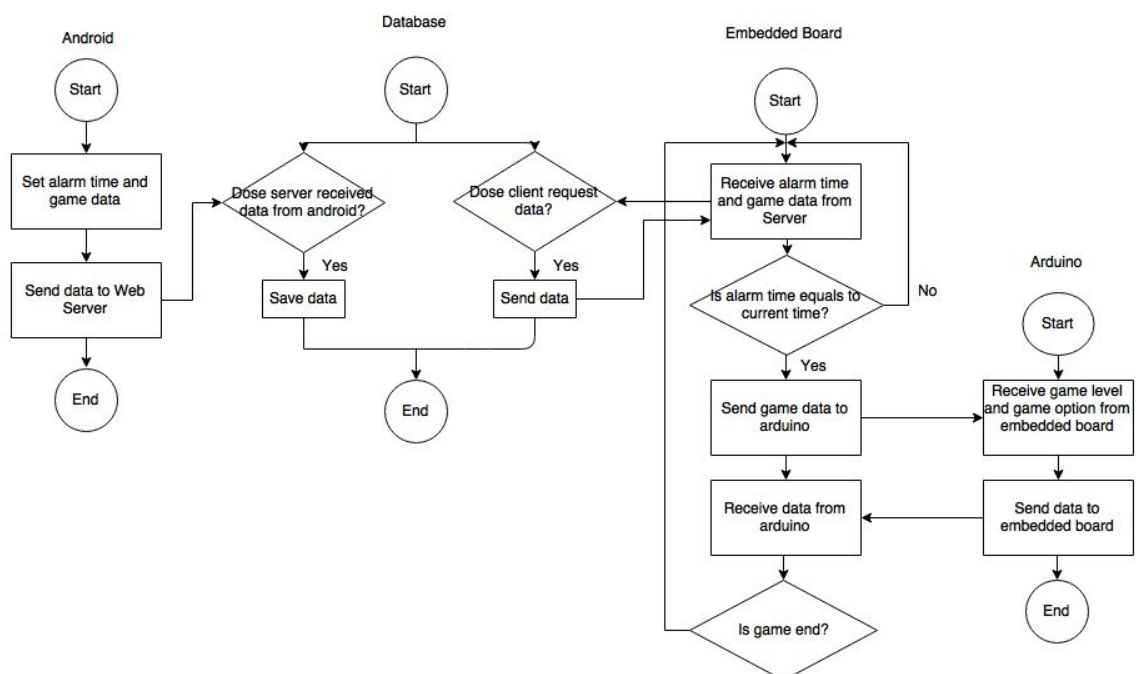
[2] Database



- Arduino

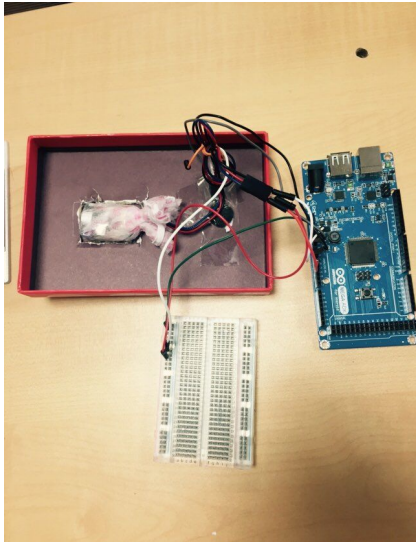


- Entire outline



3. Explanation each equipments & tools

- Arduino



[1] Analog Sound Sensor

- This sensor converts analog sound value to digital sound value. If there is no sound, sound sensor generates 0 value. So, we ignored 0 value.

[2] Pulse Sensor

- This sensor measures the heart pulse rate. Measuring principle is checking stream of light's up and down. So, when light is directly checked by pulse sensor, measured value is maximum. And our purpose is lead user to exercising. So, we ignore values under 100 and over 150.

- Embedded Board

[1] OLED

- Show images for each game level.

[2] Text LCD

- When the game level is 2, shows the question on 1 line
- Displays the hexadecimal value entered with the keypad or message on 2 line.

[3] BUS LED

- When you play Level 1 and Level 3 games, you can visually check the degree of gauging.

[4] 7-segment

- Collected numerical value from arduino is displayed on the 7-segment.

[5] Dot-Matrix

- When the game level is 2, if the answer you entered using the keypad is the correct, it shows 'O O' message. However, if the answer is an incorrect, it displays 'L L' message

[6] Buzzer

- Ring the alarm.

[7] Full Color LED

- While the alarm is ringing, turn on green light.
- When alarm is off, turn on red light.

[8] Key-Matrix

- When you play a Level 2 game, use it to write down the answer to the problem.

[9] Touch LCD

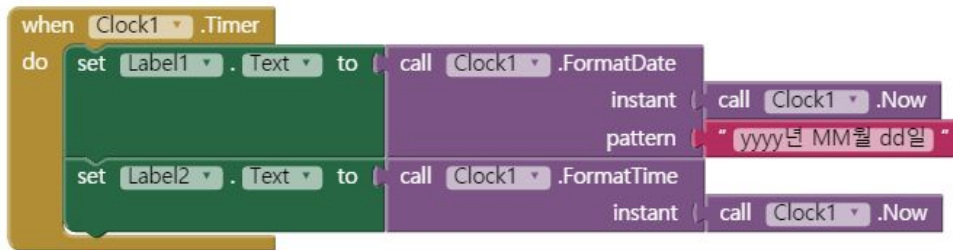
- When the game starts, you can see a description of the game. You can start each game from the main screen.

- **Android**

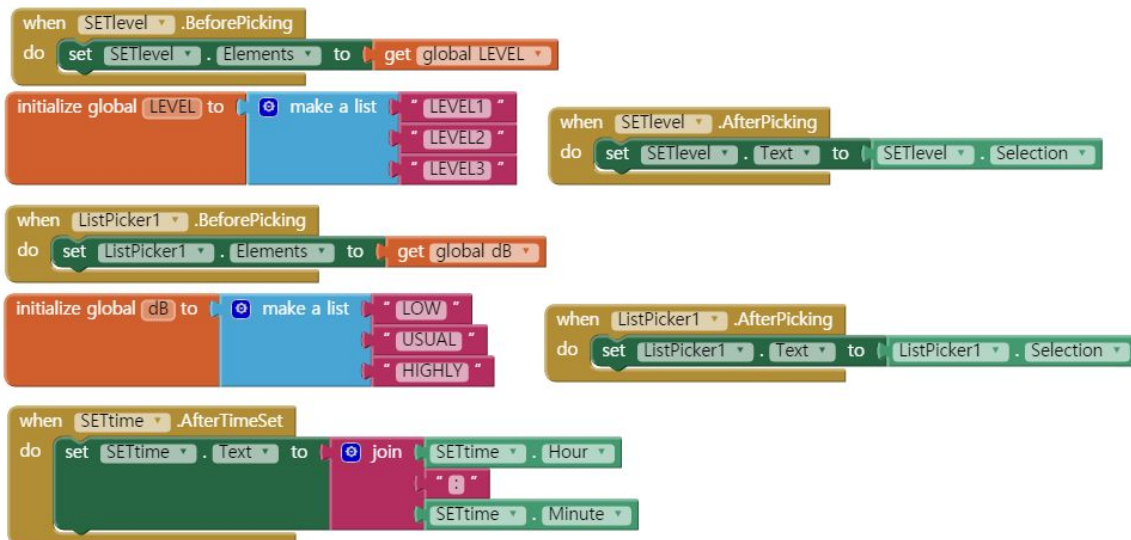
Android was implemented using 'app inventor' made by mit. Below image is the initial screen when user launch the app. I will explain below how each component was created.



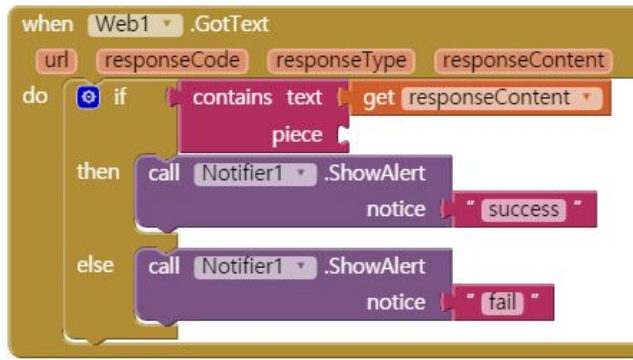
[1] When the page starts, use the clock component to display the current date and time on the screen.



[2] The game level and game option use the list picker component to select among three 'LOW', 'USUAL', and 'HIGH'. And Allows the user to set the desired time.



[3] If user click the 'SAVE' button, it connects the preset URL and sends the above values to the server in database. To prevent garbage values from being entered without selecting a game level and game option, When the selection value is 0, an error message is output without transmitting the value. If the value is successfully transferred, 'success' is returned, otherwise 'none'.



- Server

We configured the server using Java. Create a socket and bind to port 8988. The server socket waits until the client tries to connect. When a client tries to connect, it fetches the most recently saved alarm information stored in the database through the web server.

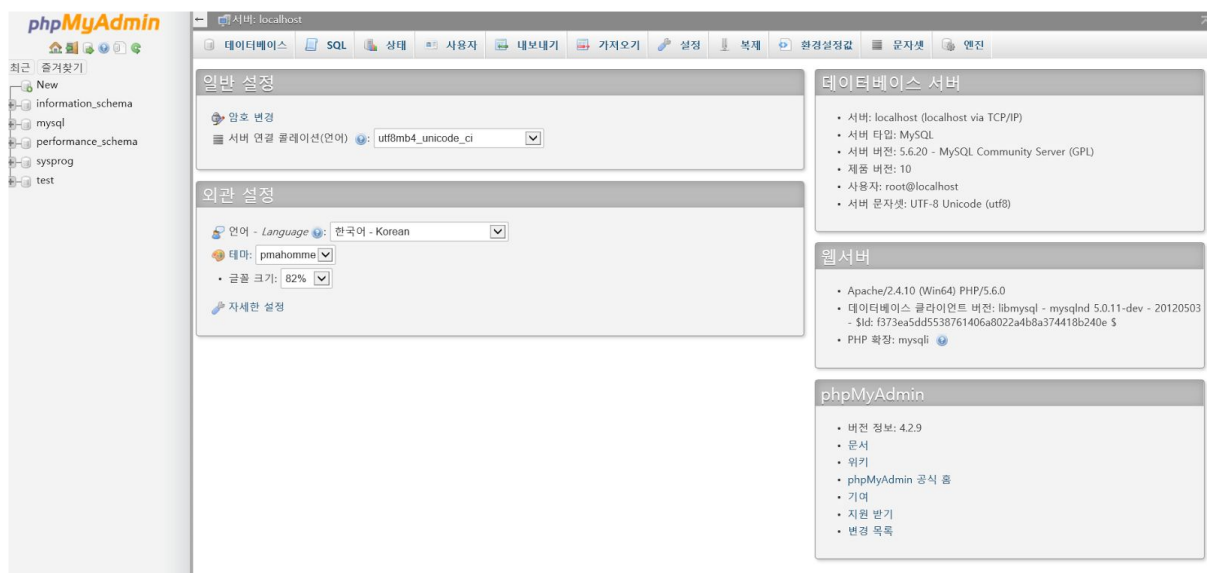
```

Properties Problems Console
Server [Java Application] C:\Program Files\Java\jre1.8.0_101\bin\javaw.exe (2016. 12. 12. 오후 8:14:04)
Server is ready.
Waiting Connect.
/192.168.43.43 is connected
Waiting Connect.
/192.168.43.43 is connected
Waiting Connect.
/192.168.43.43 is connected
Waiting Connect.
/192.168.43.43 is connected

```

- MySQL & PHP & APACHE

The server and database can be accessed via APM (Apache HTTP Server + php + mysql). We used phpMyadmin to manage MySQL on the web.



Create a database called 'sysprog' on your server for our project. Create a table called 'clock' in dB, and add three columns to store the input value from Android: 'set_time', 'set_game', 'set_option'.
Also, we added AUTO_INCREMENT to save the input order.

The screenshot shows a MySQL database interface with the following table structure:

#	이름	종류	데이터정렬방식	보기	Null	기본값	추가	실행
1	time_index	int(11)		아니오	없음	AUTO_INCREMENT		변경 삭제 기본 고유값 인덱스
2	set_time	varchar(20)	utf8_general_ci	예	NULL			변경 삭제 기본 고유값 인덱스
3	set_game	varchar(20)	utf8_general_ci	예	NULL			변경 삭제 기본 고유값 인덱스
4	set_option	varchar(20)	utf8_general_ci	예	NULL			변경 삭제 기본 고유값 인덱스

We wrote the following php to connect a specific database of mysql from outside and store the value.

```
<?php
$set_game = $_GET['set_game'];
$set_option = $_GET['set_option'];
$set_time = $_GET['set_time'];

$connect = mysqli_connect("localhost","root","autoset")
or die("fail.."); print"success<br>";

$flag = mysqli_select_db($connect,"sysprog");
if(!$flag) die("[DB selection error]");
else echo "데이터베이스 sysprog가 선택됨 <br>";

$sql = "insert into clock(set_game,set_option,set_time) values
('$set_game','$set_option','$set_time')";
$result = mysqli_query($connect,$sql) or die ("fail..".mysqli_error());

if($result)
    echo "OK";
else
    echo "0";

?>
```

tethering our cellphone to keep the same ip and set the server ip to the cellphone ip.

The screenshot shows a web browser with the URL <http://192.168.43.54/submit.php>. Below the browser window, the following PHP error messages are displayed:

```
Notice: Undefined index: set_game in C:\AutoSet9\public_html\submit.php on line 2
Notice: Undefined index: set_option in C:\AutoSet9\public_html\submit.php on line 3
Notice: Undefined index: set_time in C:\AutoSet9\public_html\submit.php on line 4
success
데이터베이스 test가 선택됨
OK
```

4. Scenario

1) Setting & Alarm

Before sleeping, use Android to set the time, game type, and difficulty level.

----Sleeping----

When the preset time is reached, the alarm sounds through the buzzer.

2) Play a game & Turn off the alarm

When the user wakes up, the game type set on the previous day can be checked through the touch panel.

case 1: Sound gauge game

Sounds to the sound sensor connected to the Arduino.

The alarm ends when the sound gauge reaches Max value according to the set difficulty level.

case 2: Convert binary to hexadecimal game

Convert binary to hexadecimal.

Enter the hexadecimal value on the text lcd using the keypad.

If the answer is correct, the alarm ends.

case 3: Heartbeat gauge game

Measure current heart rate to the pulse sensor connected to the Arduino.

The alarm ends when the heart rate gauge reaches Max value according to the set difficulty level.

Force quit

Alarm can be forcibly terminated by using dip switch.

5. Not implementation factor

1) Camera Component Use.

- Due to the nature of the alarm machine, there was no use for the camera.

2) Arduino DFPlayer Mini Mp3 Module Use.

```

#include <SoftwareSerial.h>
#include <DFPlayer_Mini_Mp3.h>

void setup()
{
  Serial.begin(9600);
  mp3_set_serial(Serial);
  mp3_set_volume(2);
}

void loop()
{
  mp3_play(1);
  delay(2000);
}

```

- It is Arduino sketch code for using “DFPlayer Mini Mp3 Module”. If we use this code with speaker module, we can listen to music. However, trash values are transmitted to Serial when this code is executed in Arduino. Below show this phenomenon.



- Trash values are transmitted when “mp3_set_serial()” or “mp3_set_volume()” called and every time “mp3_play()” called. It occurs because of “DFPlayer_Mini_Mp3.h” header’s write function.

```

void h_send_func () {
  for (int i=0; i<10; i++) {
    _hardware_serial->write (send_buf[i]);
  }
}

//
void s_send_func () {
  for (int i=0; i<10; i++) {
    _software_serial->write (send_buf[i]);
  }
}

```

- All the trash-related functions above are eventually reached to these two functions, h_send_func() and s_send_func(). If the write() in these function is not executed, trash values are not transmitted and also music does NOT played. Additional,

DFPlayer module is very ambiguous in functional perspective. Sound volume is arbitrary sometimes, and play command is sometimes ignored.

- By all these reasons, we did not use “DFPlayer Mini Mp3 Module” and used buzzer component of embedded board for replacement in sound function of our project.

3) Connect E.B to external time server to synchronize E.B’s time to real world’s time. To synchronize E.B’s time to real world’s time, we tried many methods.

- First, we tried to use RTC(Real Time Clock) in the E.B. If we turn off the power of E.B, set time of E.B is reset to initial time. But, time value of RTC does NOT reset. Thus, if we set real world’s time to RTC and RTC’s time value is set to system’s time when each the power of E.B is turned on, time of E.B can be synchronized to real world’s time. BUT, embedded board’s RTC is broken down. So this method has failed.
- Second, we tried to use ‘rdate’ instruction to synchronize time with external NTP server. To install ‘rdate’ in E.B, we have to use ‘apt install tool’. But E.B doesn’t have ‘rdate’ and ‘apt install tool’. So we had to bring it using ‘tftp’. But in these install ‘rdate’ process, we did few mistakes and something’s gone wrong. So we gave up to this method.
- Finally, we set the E.B system’s time passively by using “date -s” instruction. It is convenient method but every time the power of E.B is turned on, we have to use this instruction.

6. Member’s Role

InTae || 201220962

- ▶ Arduino
- Loudness sensor
- ▶ Embedded
- LCD module
- Text LCD
- Buzzer
- 7-segment
- Color-LED
- ▶ Arduino-Embedded communication

ByungJae || 201021531

- ▶ Arduino
- MP3 module

- ▶ Embedded
- Key matrix
- ▶ Database
- Communication between Embedded board and DB using Web Server

HwaSu || 201423047

- ▶ Arduino
- Sound sensor
- ▶ Embedded
- OLED
- ▶ Android & Database
- Android App making
- Database construction
- Communication between Android and DB
- ▶ Design
- Make image file & PPT

JinHwan || 201423050

- ▶ Arduino
- Pulse sensor
- ▶ Embedded
- Dot Matrix
- Dip Switch
- Bus LED

