

마이크로프로세서 실습 보고서

스마트 만보기



학 번	2011136004
학 과	
팀 명	1분반 5조
팀 원	강인창

목 차

서론	1
1. 작품 개요	1
2. 작품 사양	1
3. 개발 방법	2
4. 작품 재료	3
5. 개발 계획	5
본론	6
6. 진행 과정	6
6-1. 웹 애플리케이션 개발	6
6-2. 6-Digit 7-Segment 실습	16
6-3. GPS Shield 실습	31
6-4. Wi-Fi Shield 실습(+WiFi Shield firmware upgrade)	37
6-5. HTTP통신 이란?	49
6-6. 초기 작품 완성(코드 포함)	50
6-7. 최종 작품 완성(코드 포함)	56
6-8. 작품 모델링	61
6-9. 웹 애플리케이션 기능 추가	62
3. 결론	66
7. 작품 결과	66
4. 부록	69
8. 진행 과정 영상 모음	69
9. 참고 자료 모음	70

1. 개요

가. 작품 설명

마이크로프로세서 프로젝트 주제인 '스마트 만보기'는 만보기의 목적 그대로 사용자로 하여금 운동의 의지를 심어줄 수 있다. 그리고 웹 서버를 활용하여 웹서비스를 통해 일정 시간 간격 동안 사용자가 이동했던 곳의 위치를 확인 할 수 있어 범죄가 발생하였을 경우 범죄의 단서가 될 수 있다.

따라서, '스마트 만보기'를 통하여 건강과 방범문제를 해결할 수 있다.

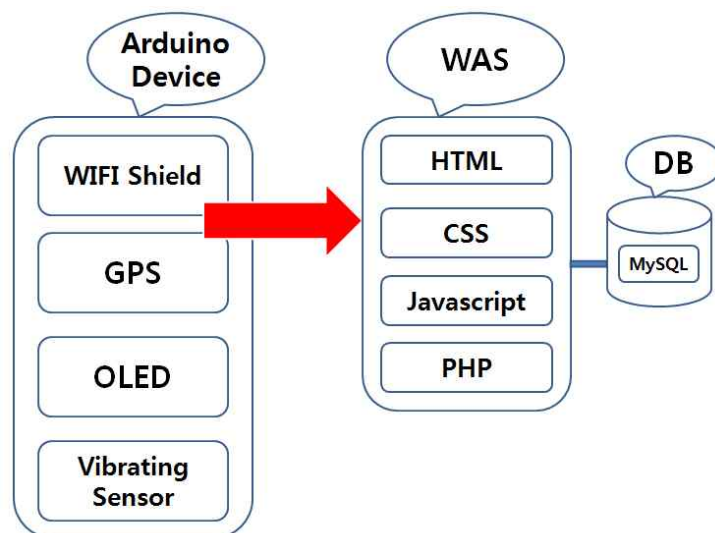


나. 상세 작동과정

1. 진동 센서가 진동을 느끼면 숫자를 카운트한다.
2. 카운트한 숫자를 OLED monitor에 출력한다.
3. 디바이스로부터 일정시간 간격동안의 GPS값 정보를 웹페이지에 출력한다.
4. GPS정보를 받은 웹페이지는 지도를 통하여 이를 마커로 표시 한다.

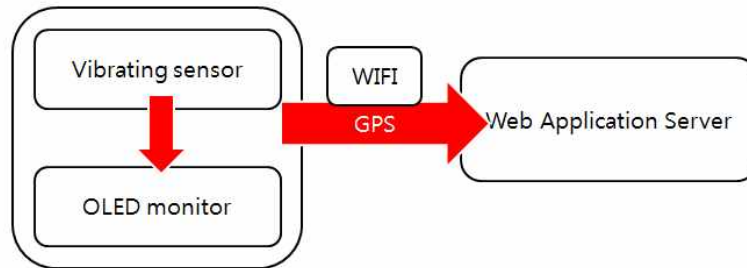
2. 작품 사양

가. 작품 구성



3. 개발 방법

가. H/W



1) 만보기

Arduino board에 Vibrating sensor와 OLED를 연결한다. 그리고 Vibrating sensor의 스위치가 켜졌을 때 카운트해서 그 값을 OLED monitor에 출력한다.

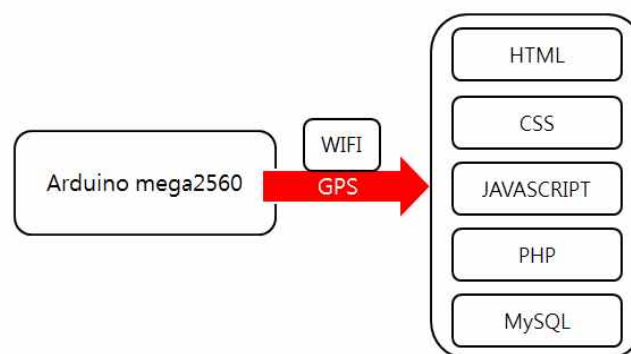
2) GPS

Arduino board에 GPS module 연결 후 GPS값을 입력받는다.

3) WIFI 통신

웹페이지로 GPS정보를 전송할 수 있도록 WIFI shield를 Arduino board와 연결하고 통신한다.






나. S/W








닷컴 사이트 가입 후 웹호스팅을 받는다. 그리고 HTML, CSS, JAVASCRIPT 사용하여 지도를 제작하고 PHP와 MySQL을 사용하여 일정 시간 간격의 GPS값을 DB에 저장하여 지도에 마커로 표시한다.

4. 필요 재료

(원본)

번호	품명	기능	수량	가격	총계	사이트
1	SW18010P 진동센서 	진동 감지	2	7700	15400	http://item2.gmarket.co.kr/item/detailview/Item.aspx?goodscode=670917599&pos_class_cd=111111111&pos_class_kind=T&pos_shop_cd=SH&keyword_order=SW18010P+%C1%F8%B5%BF%BC%BE%BC%AD&keyword_seqno=8038688212&search_keyword=SW18010P+%C1%F8%B5%BF%BC%BE%BC%AD
2	아두이노 0.96inch 128X64 White OLED I2C TWI 	숫자 출력	1	7900	7900	http://item2.gmarket.co.kr/item/detailview/Item.aspx?goodscode=698803821&pos_class_cd=111111111&pos_class_kind=T&pos_shop_cd=SH&keyword_order=%BE%C6%B5%CE%C0%CC%B3%EB+0%2E96inch+128X64+White+OLED+I2C+TWI&keyword_seqno=8038686238&search_keyword=%BE%C6%B5%CE%C0%CC%B3%EB+0%2E96inch+128X64+White+OLED+I2C+TWI
3	u-blox NEO-6M GPS 	GPS	1	23500	23500	http://killdoslab.com/product/detail.html?product_no=61&cate_no=59&display_group=1#
4	CC3000 WIFI shield 	WiFi 통신	1	46500	46500	http://igameplussshop.com/shop/goods/goods_view.php?goodsno=832&inflow=naver&NaPm=ct%3Die7cw3tk%7Cci%3D3d6fb0125ff6c395fa391091444d41e268e0152c%7Ctr%3Dslsl%7Cs%3D226306%7Chk%3Da23a4c766047ca75219b33db749ce65ee413d2c5
7	9V 1000mA SMPS 	어댑터	1	8000	8000	http://artrobot.co.kr/front/php/product.php?product_no=245
총 가격 :						101,300

(최종본)

번호	품명	기능	수량	가격	총계	사이트
1	아두이노 0.96inch 128X64 White OLED 		1	7900	7900	http://item2.gmarket.co.kr/item/detailview/Item.aspx?goodscode=698803821&pos_class_cd=111111111&pos_class_kind=T&pos_shop_cd=SH&keyword_order=%BE%C6%B5%CE%C0%CC%B3%EB+0%2E96inch+128X64+White+OLED+I2C+TWI&keyword_seqno=8038686238&search_keyword=%BE%C6%B5%CE%C0%CC%B3%EB+0%2E96inch+128X64+White+OLED+I2C+TWI
2	아두이노 디지털 기울기 센서 	기울기 감지	1	2400	2400	http://roboholic1.godo.co.kr/shop/goods/goods_view.php?goodsno=2794&inflow=naver&NaPm=ct%3Dihp2qn1s%7Cci%3Dca832db52c3b39fc7f57f55feb238a37ebf7b964%7Ctr%3Dslsl%7Csn%3D188145%7Chk%3D0537c7a238b1a481acda47bd183c798f2f41afca
3	아두이노 GPS 쉴드 	GPS 정보 수신	1	.	.	.
4	아두이노 WiFi 쉴드 	WiFi 통신	1	.	.	.
5	아두이노 호환 배터리 홀더 	휴대 기능	1	3300	3300	http://robotpia.net/goods_detail.php?goodsIdx=2564&NaPm=ct%3Dihp30btc%7Cci%3D0bbea4008dc6425aa8b695ad93e48da0a9684678%7Ctr%3Dslsl%7Csn%3D237929%7Chk%3Db89b9d6e7b402b6d358181c2cc4bce30d125c718
총 가격 :					13600	

5. 단계별 개발 계획

(원본)

	9월	10월	11월	12월
Vibrating sensor 활용 및 테스트				
OLED monitor 활용 및 테스트				
만보기 제작(OLED에 count값 출력)				
만보기 테스트				
GPS module 활용 및 테스트				
WIFI 통신				
GPS정보 웹페이지 연동				
웹 애플리케이션 제작				
웹페이지 GPS정보 지도에 출력				
통합테스트				

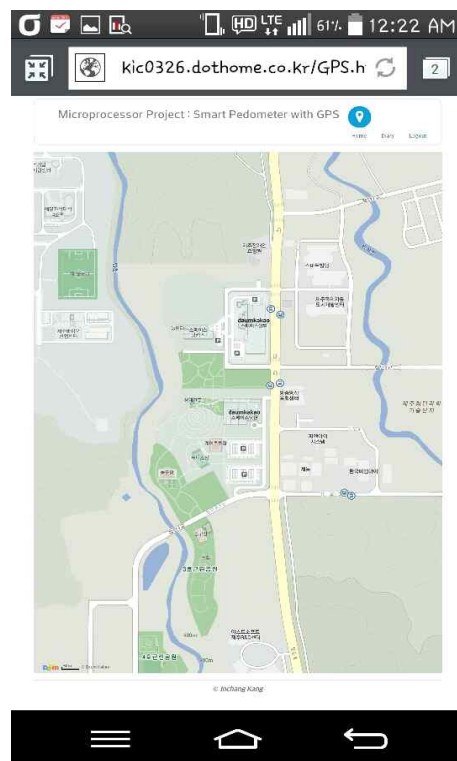
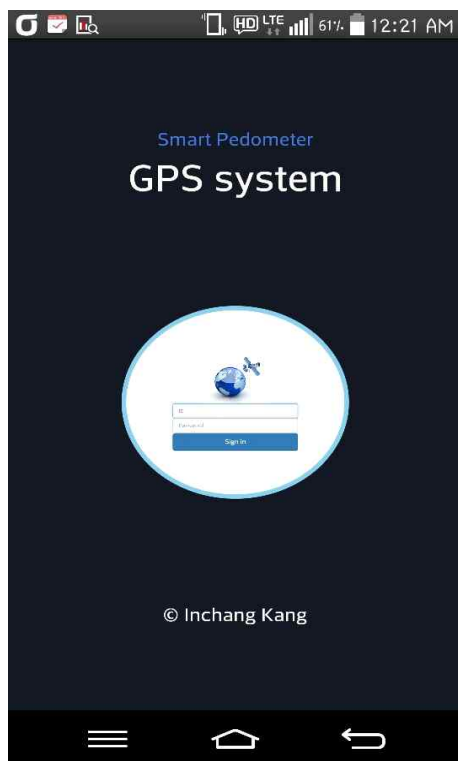
(최종본)

	9월	10월	11월	12월
Vibrating sensor 활용 및 테스트				
6-Digit 7-Segment 활용 및 테스트				
만보기 제작				
만보기 테스트				
GPS module 활용 및 테스트				
WIFI 통신 활용 및 테스트				
웹 애플리케이션 제작				
아두이노와 웹페이지 연동				
외관 모델링 및 제작				
OLED 런닝머신 디스플레이				
통합테스트				

6. 진행 과정

1주차(9월 7일 ~ 9월 13일)

- 교수님께서 우리가 처음 아두이노와 센서, 모듈들을 써보기 때문에 연결을 잘못하여 부품들이 상할 수 있다는 것을 감안해 부품을 2개씩 주문하라고 하여 수정하였다. 그리고 재료 조사 중 아두이노보드에 실드로 확장할 경우 많은 전력소모가 일어나므로 PC에 연결된 USB케이블로부터의 전류만으로는 제대로 작동하지 않을 수 있을 경우를 대비해 9V 어댑터를 주문하기로 결정하였고 개인적으로 실험하기 위해 브레드 보드도 주문하기로 하였다.
- 주문한 재료가 오기 까지 시간이 있기 때문에 웹 애플리케이션 제작을 선행 하였다. 우선, 웹 페이지를 구현해놓기 위해 닷홈에서 웹호스팅을 받았고 메인 페이지에 ID와 PW가 있어야 접속하도록 하였다. 접속 후 GPS에 대한 Map을 볼 수 있도록 구현 하였고 현재 개선 중에 있다.
- 애플리케이션 주소 : kic0326.dothome.co.kr



2주차(9월 14일 ~ 9월 20일)

- 홈페이지를 개선하여 어느 디바이스에서의 해상도에서나 맞도록 사이즈 조정을 하였다. 그리고 웹페이지와 연동할 수 있는 방법으로 처음엔 html5에서 제공하는 geolocation API를 생각했었지만, 이것은 웹에 접속한 사용자의 IP를 받아 그 위치를 표시해주므로 연동하는데 어려움이 있을 거라 생각하고 다른 방법을 모색하던 중에 HTTP통신을 발견했다. 다행히, 아두이노 스튜디오에 HTTP통신 예제가 있기 때문에 따라하면 원하던 것을 할 수 있을 것 같고 HTTP통신에 대하여 조사해보아야겠다.

- WiFiWebClient 코드- <https://www.arduino.cc/en/Tutorial/WiFiWebClient>
- HttpClient 코드- <https://www.arduino.cc/en/Tutorial/HttpClient>
- Http통신 이란?- <http://takeiteazy.tistory.com/298>

- 웹 애플리케이션 코드

index.html (첫 화면)

```
<?php
    header("Content-Type: text/html; charset=UTF-8");
    session_start();
    //error_reporting(E_ALL);
    //ini_set("display_errors", 1);
?>
<!DOCTYPE html>
<html lang="ko">
    <head>
        <meta charset="UTF-8">
        <title>GPS Maps</title>
        <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.2/css/bootstrap.min.css">
        <style type="text/css">
            .form-signin{
                background-color: white;
                width:50%;
                height:500px;
                margin-top:25%;
                margin-bottom:25%;
                margin-left:auto;
                margin-right:auto;
```

```

padding-left:10%;
padding-right:10%;
border-radius: 100%;
border: 10px solid skyblue;
}
header h3{
margin-top:250px;
font-size:50px;
color:#4374D9;
text-align: center;
}
header h1{
font-size:100px;
color:white;
text-align: center;
}
p.alert{
margin:0;
padding:5px;
text-align:center;
font-size: 17px;
}
#foot{
color:white;
font-size: 50px;
margin-bottom:250px;
}
img{
margin-top:40px;
border-top-right-radius: 50%;
border-top-left-radius: 50%;
}
</style>
</head>
<body style="background-color:rgb(20, 24, 35);">
<header>
<h3>Smart Pedometer</h3>
<h1>GPS system</h1>
</header>
<form class="form-signin" method="POST" action="login_ok.php">

    <?php if($_GET['err']): ?>
        <p class="alert alert-danger" role="alert">Check ID or Password again</p>
    <?php endif; ?>

    <label for="inputId" class="sr-only">Identification</label>
    <input name="id" id="inputId" class="form-control" placeholder="ID" required
autofocus>
    <label for="inputPassword" class="sr-only">Password</label>
    <input name="pw" type="password" id="inputPassword" class="form-control"
placeholder="Password" required>
    <button class="btn btn-lg btn-primary btn-block" type="submit">Sign in</button>
</form>
</body>
<footer>
    <center><h4 id="foot">&copy; Inchang Kang</h4></center>
</footer>
</html>

```

header.php (GPS.html의 헤더)

```

<?php
    session_start();
?>

<!DOCTYPE html>
<html lang="ko">
    <head>
        <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.2/css/bootstrap.min.css">
        <style>
            .container{
                margin:5px auto;
                width:90%;
                border: 3px solid #ebebeb;
                border-radius:10px;
                padding-top:15px;
                padding-left:50px;
            }
            h2{
                display: inline;
            }
            img{

```

```

        margin-left:12px;
        margin-right:12px;
    }
</style>
<meta charset="UTF-8">
</head>
<body>
    <div class="container">
        <header>
            <h2 class="text-muted">Microprocessor Project : Smart Pedometer with GPS </h2>
            
        </header>
        <nav>
            <ul class="nav nav-pills pull-right">
                <li role="presentation"><a href="GPS.html">Home</a></li>
                <li role="presentation"><a href="record.php">Diary</a></li>
                <?php
                    if($_SESSION['login_id']){
                        echo ' <li><a href="logout.php">Logout</a></li>';
                    }
                ?>
            </ul>
        </nav>
    </div>
</body>
</html>

```

footer.php (GPS.html의 푸터)

```

<!DOCTYPE html>
<html lang="ko">
    <head>
        <style>
            footer{
                width:90%;
                margin:0 auto;
                border-top: 3px solid #ebebeb;
            }
        </style>
        <meta charset="UTF-8">
    </head>

```

```

<footer>
  <center><h4 class="text-muted">&copy  <em><strong>Inchang Kang</strong></em></h4></center>
</footer>
</html>

```

login_ok.php (로그인)

```

<?php
header("Content-Type: text/html; charset=UTF-8");
$mysql = new mysqli('localhost', 'DB아이디', 'DB비밀번호', 'DB아이디');
$result = $mysql -> query("SELECT * FROM user WHERE `id` =
"".addslashes($_POST['id'])."");
$row = $result -> fetch_assoc();
if($row['pw'] == $_POST['pw']) {
    session_start();
    $_SESSION['login_id'] = $_POST['id'];
    header('location:/GPS.html');
} else {
    header('location:/index.html?err=1');
}
?>

```

GPS.html (현재 위치 지도)

```

<?php include "header.php" ?>
<?php
if(!$_SESSION['login_id']):
    header('location:/index.html?err=1');
?>
<?php else:
?>
<?php endif: ?>

<?php
header("Content-Type: text/html; charset=UTF-8");
$host = "mysql:host=localhost;dbname=DB아이디";
$user = "DB아이디";
$password= "DB비밀번호";
$conn = new PDO($host, $user, $password, array(PDO::MYSQL_ATTR_INIT_COMMAND
=> "SET NAMES utf8"));
$stmt = $conn->prepare('SELECT * FROM gps');
$stmt->execute();
$list = $stmt->fetchAll();

```

```

foreach($list as $gps){
    $latitude = $gps['latitude'];
    $longitude = $gps['longitude'];
}

?>
<!DOCTYPE html>
<html lang="ko">
    <head>
        <style>
            #map{
                margin:15px auto;
            }
        </style>
        <meta charset="UTF-8">
        <meta http-equiv=refresh content="1,/GPS.html">
        <title>GPS Maps</title>
    </head>
    <body>
        <div id="map" style="width:90%;height:1000px;"></div>
        <script type="text/javascript" src="//apis.daum.net/maps/maps3.js?apikey=a08b894cef34ee836139835bb7e2e3b7"></script>
        <script language="JavaScript">
            setTimeout("history.go(0);",5000);
            var latitude = <?=$latitude?>;
            var longitude = <?=$longitude?>;
            //console.log(latitude);
            //console.log(longitude);
            var container = document.getElementById('map');
            var options = {
                center: new daum.maps.LatLng(latitude, longitude),
                level: 1
            };
            var map = new daum.maps.Map(container, options);
            // 마커가 표시될 위치입니다
            var markerPosition = new daum.maps.LatLng(latitude, longitude);
            // 마커를 생성합니다
            var marker = new daum.maps.Marker({
                position: markerPosition
            });
            // 마커가 지도 위에 표시되도록 설정합니다

```

```

        marker.setMap(map);
    </script>
</body>
</html>
<?php include "footer.php" ?>

```

logout.php (로그아웃)

```

<?php
    session_start();
    unset($_SESSION['login_id']);
    header("location:/index.html");
?>

```

dbconfig.php (HTTP통신)

```

<?php
    echo $_GET['x'].$_GET['y'];
    $mysqli = new mysqli('localhost', 'DB아이디', 'DB비밀번호', 'DB아이디');
    $mysqli -> query('INSERT INTO gps(latitude, longitude) VALUE('.$_GET['latitude'].',
    '.$_GET['longitude'].')');
?>

```

record.php (기록된 위치 지도)

```

<?php include "header.php" ?>
<?php
    if(!$_SESSION['login_id']):
        header('location:/index.html?err=1');
    ?>
<?php else:
    ?>
<?php endif: ?>
<?php
    $connect = mysql_connect("localhost", "DB아이디", "DB비밀번호");
    $db_con = mysql_select_db("DB아이디", $connect);
    $sql = "select * from gps";
    $result = mysql_query($sql, $connect);

    $size = 0;                                // 크기
    $latitude = array();    // gps테이블에 있는 위도 값들을 넣을 php배열
    $longitude = array(); // gps테이블에 있는 경도 값들을 넣을 php배열
?>

```

```

<?php
while($row = mysql_fetch_array($result)){
    $latitude[$size] = $row['latitude'];    // 위도 값 저장
    $longitude[$size] = $row['longitude']; // 경도 값 저장
    $size++;                                // 크기
}
?>
<!DOCTYPE html>
<html lang="ko">
    <head>
        <style>
            #map{
                margin:15px auto;
            }
        </style>
        <meta charset="UTF-8">
        <title>Recorded information</title>
    </head>
    <body>
        <div id="map" style="width:90%;height:1000px;"></div>
        <script type="text/javascript" src="//apis.daum.net/maps/maps3.js?apikey=82efa9caba9af3640b436747196453bc"></script>
        <script language="JavaScript">
            // php의 위도값 배열을 자바스크립트 배열에 저장
            var latitude = <?php echo json_encode($latitude); ?>;
            // php의 경도값 배열을 자바스크립트 배열에 저장
            var longitude = <?php echo json_encode($longitude); ?>;
            var size = <?=$size?>;// 크기 저장
            for(var i=0; i<size; i++){
                console.log(latitude[i]);
                console.log(longitude[i]);
            }
            // 지도를 표시할 div
            var mapContainer = document.getElementById('map'),
            mapOption = {
                //지도 중심 좌표
                center: new daum.maps.LatLng(latitude[size-1], longitude[size-1]),
                level: 2
            };
            //지도 생성
            var map = new daum.maps.Map(mapContainer, mapOption);

```



```

    for(var i = 0; i<size; i++) {
        // 다중 마커 생성
        var marker = new daum.maps.Marker({
            position: new daum.maps.LatLng(latitude[i], longitude[i])
        });
        marker.setMap(map);
    }
    </script>
</body>
</html>
<?php include "footer.php" ?>

```

- Arduino.cc 홈페이지에 들어가 getting started 파트를 학습하여 아두이노 소프트웨어를 설치하였다. 그리고 아두이노 보드를 연결하고 드라이버를 설치한 후 아두이노 스튜디오를 실행 하였다.
- Proto shield를 연동한 후 예제로 blink를 학습하였다.

```

void setup() {
    // initialize digital pin 13 as an output.
    pinMode(13, OUTPUT);
}

void loop() {
    digitalWrite(13, HIGH);    // turn the LED on (HIGH is the voltage level)
    delay(1000);               // wait for a second
    digitalWrite(13, LOW);     // turn the LED off by making the voltage LOW
    delay(1000);               // wait for a second
}

```

- 교수님께서 6Digit 7-Segment 모듈에 숫자 출력을 과제로 내주셨다. 그래서 CNDI 홈페이지에 방문해 관련 자료를 다운 받았고 라이브러리를 아두이노 스튜디오에 삽입했다. 우선 라이브러리를 학습해야 겠다.

- 7-Segment 6-Digit Module libraries

FndDisplay.ino

```
// 헤더 파일 인클루드
#include "FndControl.h"

// 출력할 위치를 찾는다. 오른쪽부터 Digit의 sel의 순서를 의미한다.
const int Fndtable[6]={
    // 16진수, 아두이노 <- 세그먼트
    FND_1_SEL, // 첫번째
    FND_2_SEL, // 두번째
    FND_3_SEL, // 세번째
    FND_4_SEL, // 네번째
    FND_5_SEL, // 다섯번째
    FND_6_SEL // 여섯번째
};

// 출력할 텍스트의 모양을 의미한다.
char u_cText[10] = {'0','1','2','3','4','5','6','7','8','9'};

void setup(){
    // 출력할 위치와, 각 세그먼트의 핀을 설정한다.
    FndInit();
}

void loop(){
    // 원하는 위치에 원하는 텍스트 모양을 출력한다.
    DrawTextFnd(Fndtable[],u_cText[]);
}
```

FndControl.cpp

```
// 출력할 위치에 대한 아두이노 디지털 핀을 의미한다.
const int FndSelectTable[8] =
{
    // 10진수, 아두이노 -> 세그먼트
    FND_SEL1 , // 첫번째
    FND_SEL2 , // 두번째
```

```

FND_SEL3 , // 세번째
FND_SEL4 , // 네번째
FND_SEL5 , // 다섯번째
FND_SEL6 , // 여섯번째
};

// 출력할 세그먼트에 대한 아두이노 디지털 핀을 의미한다.
const int FndPinTable[8] =
{
    FND_A , // 0 - 위
    FND_B , // 1 - 위 오른쪽
    FND_C , // 2 - 아래 오른쪽
    FND_D , // 3 - 아래
    FND_E , // 4 - 아래 왼쪽
    FND_F , // 5 - 위 왼쪽
    FND_G , // 6 - 중앙
    FND_DP // 7 - 점
};

// 화면에 띄울 0~9까지 숫자의 비트 연산, 원하는 텍스트를 편집할 수 있는 부분이다.
const int FndNumberTable[10] =
{
    A_BIT | B_BIT | C_BIT | D_BIT | E_BIT | F_BIT , // '0'
    B_BIT | C_BIT , // '1'
    A_BIT | B_BIT | D_BIT | E_BIT | G_BIT, // '2'
    A_BIT | B_BIT | C_BIT | D_BIT | G_BIT, // '3'
    B_BIT | C_BIT | F_BIT | G_BIT, // 4
    A_BIT | C_BIT | D_BIT | F_BIT | G_BIT, // '5'
    C_BIT | D_BIT | E_BIT | F_BIT | G_BIT, // '6'
    A_BIT | B_BIT | C_BIT , // '7'
    A_BIT | B_BIT | C_BIT | D_BIT | E_BIT | F_BIT | G_BIT, // '8'
    A_BIT | B_BIT | C_BIT | F_BIT | G_BIT, // 9
};

// 출력하기 위해 모든 핀을 OUTPUT모드로 설정
void FndInit()
{
    int i;
    for(i = 0; i<MAX_FND; i++){
        pinMode(FndSelectTable[i], OUTPUT); //Fnd Sel Pin OUTPUT Set
        pinMode(FndPinTable[i], OUTPUT);    //Fnd Data Pin OUTPUT Set
    }
}

```

```

    }
}

// 표현 해주려는 데이터를 판단하고 이를 화면에 출력한다. FndData(text)의 모양
void FndData(char text);

// 출력할 테이블(S0 ~ S5)을 찾는다. FndSelect(Position)에서 들어오는 위치
void FndSelect(int Position)
{
    int i;
    for( i = 0; i<MAX_FND; i++){
        // A & B = 두개의 비트 모두 1일 때 TRUE를 반환한다.
        // A << B = A비트를 B만큼 왼쪽으로 이동한 값
        if(Position & (1<<i)){
            digitalWrite(FndSelectTable[i], LOW); // 테이블 OFF ( LOW = 0 )
        }
        else {
            digitalWrite(FndSelectTable[i], HIGH); // 테이블 ON ( HIGH = 1 )
        }
    }
}

// 출력할 테이블(S0 ~ S5)를 찾고 데이터('0' ~ '9')를 7-SEG화면에 출력한다.
void DrawTextFnd(int Position, char text){
    FndSelect(Position);
    FndData(text);
}

```

FndControl.h

```

#define FND_SEL1      8
#define FND_SEL2      9
#define FND_SEL3     10
#define FND_SEL4     11
#define FND_SEL5     12
#define FND_SEL6     13

#define FND_A 0 //위
#define FND_B 1 //위 오른쪽
#define FND_C 2 //아래 오른쪽
#define FND_D 3 //아래
#define FND_E 4 //아래 왼쪽

```

```
#define FND_F 5 //위 왼쪽
#define FND_G 6 //중앙
#define FND_DP 7 //점

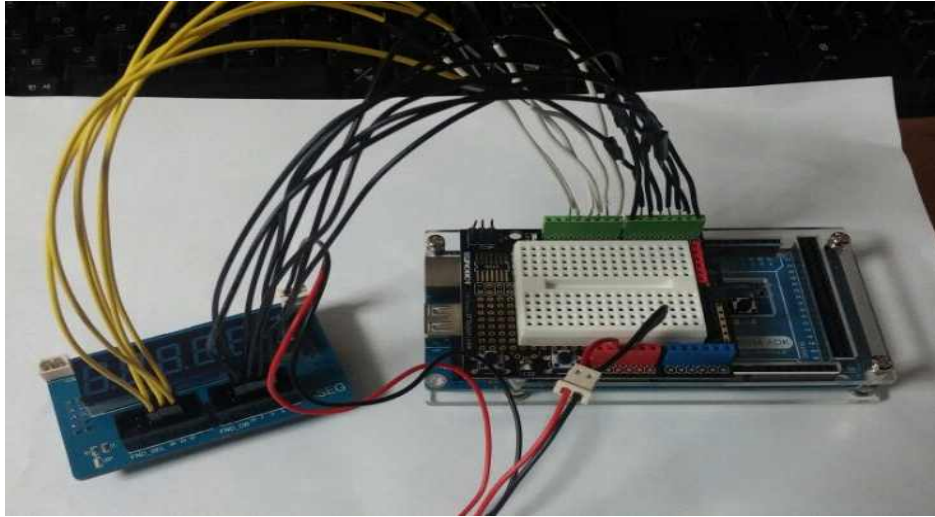
#define A_BIT 0x01 // 0000 0001
#define B_BIT 0x02 // 0000 0010
#define C_BIT 0x04 // 0000 0100
#define D_BIT 0x08 // 0000 1000
#define E_BIT 0x10 // 0001 0000
#define F_BIT 0x20 // 0010 0000
#define G_BIT 0x40 // 0100 0000
#define DP_BIT 0x80 // 1000 0000

#define FND_1_SEL 0x01 // 0000 0001
#define FND_2_SEL 0x02 // 0000 0010
#define FND_3_SEL 0x04 // 0000 0100
#define FND_4_SEL 0x08 // 0000 1000
#define FND_5_SEL 0x10 // 0001 0000
#define FND_6_SEL 0x20 // 0010 0000

#define MAX_FND 8

void FndInit();//화면에 출력할 세그먼트와 출력할 위치의 핀모드 설정
void FndSelect(int Position); //화면에 각 세그먼트를 출력할 위치
void DrawTextFnd(int Position, char text);//출력
void FndOff();//멈춤
```

- 7-Segment 6-digit Module test



Mega 2560 Pin	7-SEG Pin
0	FND_DB A
1	FND_DB B
2	FND_DB C
3	FND_DB D
4	FND_DB E
5	FND_DB F
6	FND_DB G
7	FND_DB DP
8	FND_SEL S0
9	FND_SEL S1
10	FND_SEL S2
11	FND_SEL S3
12	FND_SEL S4
13	FND_SEL S5

우선, 테스트를 위해 위와 같이 결선 하였다. Arduino Mega 2560의 핀 번호와 7-Segment 6-digit module의 결선 방법은 위의 표와 같다.

※ 문제점 < 참고 URL - <https://www.baldengineer.com/millis-tutorial.html> >
- delay() 메소드 사용으로 문제 발생 -> millis() 메소드로 해결

다섯째 자릿수 까지 숫자 자동 증가(.ino 파일)



<https://www.youtube.com/watch?v=-a4NhTsUG30&feature=youtu.be>

```
#include "FndControl.h"

const int Fndtable[6]={
    FND_1_SEL,
    FND_2_SEL,
    FND_3_SEL,
    FND_4_SEL,
    FND_5_SEL,
    FND_6_SEL
};

char u_cText[10] = {'0','1','2','3','4','5','6','7','8','9'};

unsigned long time = 0;
int one = 0, ten = 0, hundred = 0, thousand = 0, ten_hundred = 0;

void setup(){
    FndInit();
    time = millis();
}
```

```

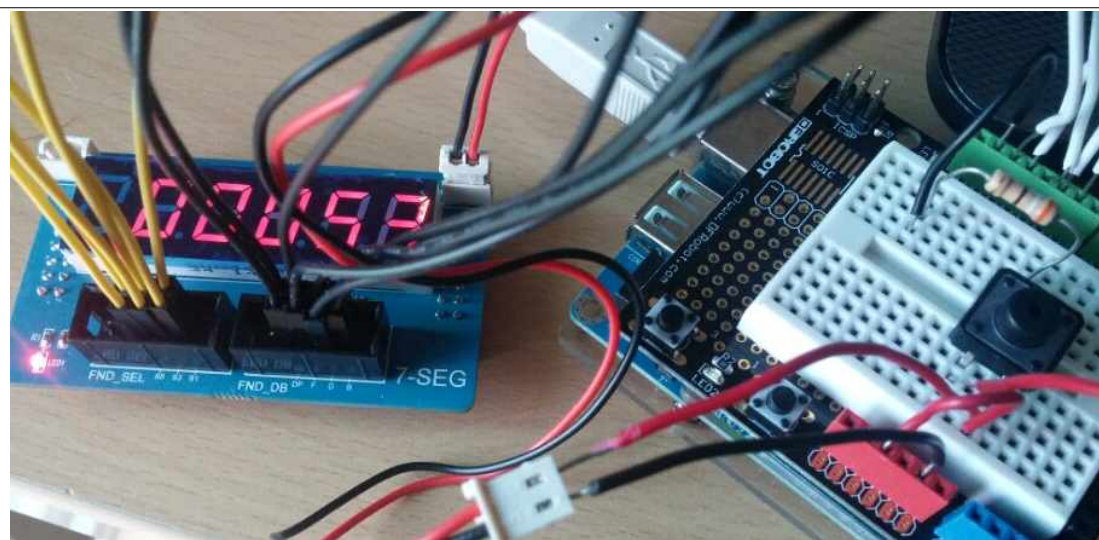
void loop()
{
    unsigned long time2 = millis();
    if(time2 - time >= 1000){
        one++;
        ten++;
        hundred++;
        thousand++;
        ten_hundred++;

        time = time2;
    }
    if(ten%10 == 0) one = 0;
    if(hundred%100 == 0) ten = 0;
    if(thousand%1000 == 0) hundred = 0;
    if(ten_hundred%10000 == 0) thousand = 0;

    DrawTextFnd(Fndtable[0],u_cText[one]);
    DrawTextFnd(Fndtable[1],u_cText[ten/10]);
    DrawTextFnd(Fndtable[2],u_cText[hundred/100]);
    DrawTextFnd(Fndtable[3],u_cText[thousand/1000]);
    DrawTextFnd(Fndtable[4],u_cText[ten_hundred/10000]);
}

```

다섯째 자릿수 까지 숫자 버튼 클릭 시 증가(.ino 파일)



<https://www.youtube.com/watch?v=XbomhwUJSXI&feature=youtu.be>

#include "FndControl.h"


```

const int Fndtable[6]={
    FND_1_SEL,
    FND_2_SEL,
    FND_3_SEL,
    FND_4_SEL,
    FND_5_SEL,
    FND_6_SEL
};

char u_cText[10] = {'0','1','2','3','4','5','6','7','8','9'};

const int BUTTON = 14;
unsigned long time = 0;
int one = 0, ten = 0, hundred = 0, thousand = 0, ten_hundred = 0;

void setup(){
    FndInit();
    time = millis();
    pinMode(BUTTON, INPUT);
}

void loop()
{
    unsigned long time2 = millis();
    if(time2 - time >= 120){
        if(digitalRead(BUTTON) == HIGH){
            one++;
            ten++;
            hundred++;
            thousand++;
            ten_hundred++;
        }
        time = time2;
    }
    if(ten%10 == 0) one = 0;
    if(hundred%100 == 0) ten = 0;
    if(thousand%1000 == 0) hundred = 0;
    if(ten_hundred%10000 == 0) thousand = 0;

    DrawTextFnd(Fndtable[0],u_cText[one]);
}

```

```

DrawTextFnd(Fndtable[1],u_cText[ten/10]);
DrawTextFnd(Fndtable[2],u_cText[hundred/100]);
DrawTextFnd(Fndtable[3],u_cText[thousand/1000]);
DrawTextFnd(Fndtable[4],u_cText[ten_hundred/10000]);
}

```

TEST 3) 7-Segment 6-digit module에 HELLO문자열 띄우기

< 실행 화면 >



FndControl.cpp

```

#include <Arduino.h>
#include "FndControl.h"

const int FndSelectTable[8] =
{
    FND_SEL1 , //첫번째
    FND_SEL2 , //두번째
    FND_SEL3 , //세번째
    FND_SEL4 , //네번째
    FND_SEL5 , //다섯번째
    FND_SEL6 , //여섯번째
};

const int FndPinTable[8] =
{
    FND_A , //위

```

```

FND_B , //위 오른쪽
FND_C , //아래 오른쪽
FND_D , //아래
FND_E , //아래 왼쪽
FND_F , //위 왼쪽
FND_G , //중앙
FND_DP //점
};

const int FndNumberTable[5] =
{
    B_BIT | C_BIT | E_BIT | F_BIT | G_BIT , // 'H'
    A_BIT | D_BIT | E_BIT | F_BIT | G_BIT, // 'E'
    D_BIT | E_BIT | F_BIT , // 'L'
    D_BIT | E_BIT | F_BIT , // 'L'
    A_BIT | B_BIT | C_BIT | D_BIT | E_BIT | F_BIT, // 'O'
};

void FndInit()
{
    int i;
    for(i = 0; i<MAX_FND; i++){
        pinMode(FndSelectTable[i], OUTPUT);    //Fnd Sel Pin OUTPUT Set
        pinMode(FndPinTable[i], OUTPUT);       //Fnd Data Pin OUTPUT Set
    }
}

void FndSelect(int Position)
{
    int i;
    for( i = 0; i<MAX_FND; i++){
        if(Position & (1<<i)){
            digitalWrite(FndSelectTable[i], LOW);
        } else {
            digitalWrite(FndSelectTable[i], HIGH);
        }
    }
}

void FndData(char text)
{
    int i;

```

```

if(text == 'H')
{
    for( i = 0; i<MAX_FND; i++){
        if(FndNumberTable[0] & (1<<i)){
            digitalWrite(FndPinTable[i], HIGH);
        }
        else{
            digitalWrite(FndPinTable[i], LOW);
        }
    }
    delay(1);
}
if(text == 'E')
{
    for( i = 0; i<MAX_FND; i++){
        if(FndNumberTable[1] & (1<<i)){
            digitalWrite(FndPinTable[i], HIGH);
        }
        else{
            digitalWrite(FndPinTable[i], LOW);
        }
    }
    delay(1);
}
if(text == 'L')
{
    for( i = 0; i<MAX_FND; i++){
        if(FndNumberTable[2] & (1<<i)){
            digitalWrite(FndPinTable[i], HIGH);
        }
        else{
            digitalWrite(FndPinTable[i], LOW);
        }
    }
    delay(1);
}
if(text == 'L')
{
    for( i = 0; i<MAX_FND; i++){
        if(FndNumberTable[3] & (1<<i)){
            digitalWrite(FndPinTable[i], HIGH);

```

```

    }
    else{
        digitalWrite(FndPinTable[i], LOW);
    }
}
delay(1);
}
if(text == 'O')
{
    for( i = 0; i<MAX_FND; i++){
        if(FndNumberTable[4] & (1<<i)){
            digitalWrite(FndPinTable[i], HIGH);
        }
        else{
            digitalWrite(FndPinTable[i], LOW);
        }
    }
    delay(1);
}
if(text == '.'){
    for( i = 0; i<MAX_FND; i++){
        if(FndNumberTable[5] & (1<<i)){
            digitalWrite(FndPinTable[i], HIGH);
        }
        else{
            digitalWrite(FndPinTable[i], LOW);
        }
    }
    delay(1);
}
delay(1);
}

void DrawTextFnd(int Position, char text)
{
    FndSelect(Position);
    FndData(text);
}

```

FndDisplay.ino

```
char u_cText[8] = {'H','E','L','L','O'};
```

```

void setup()
{
    FndInit();
}

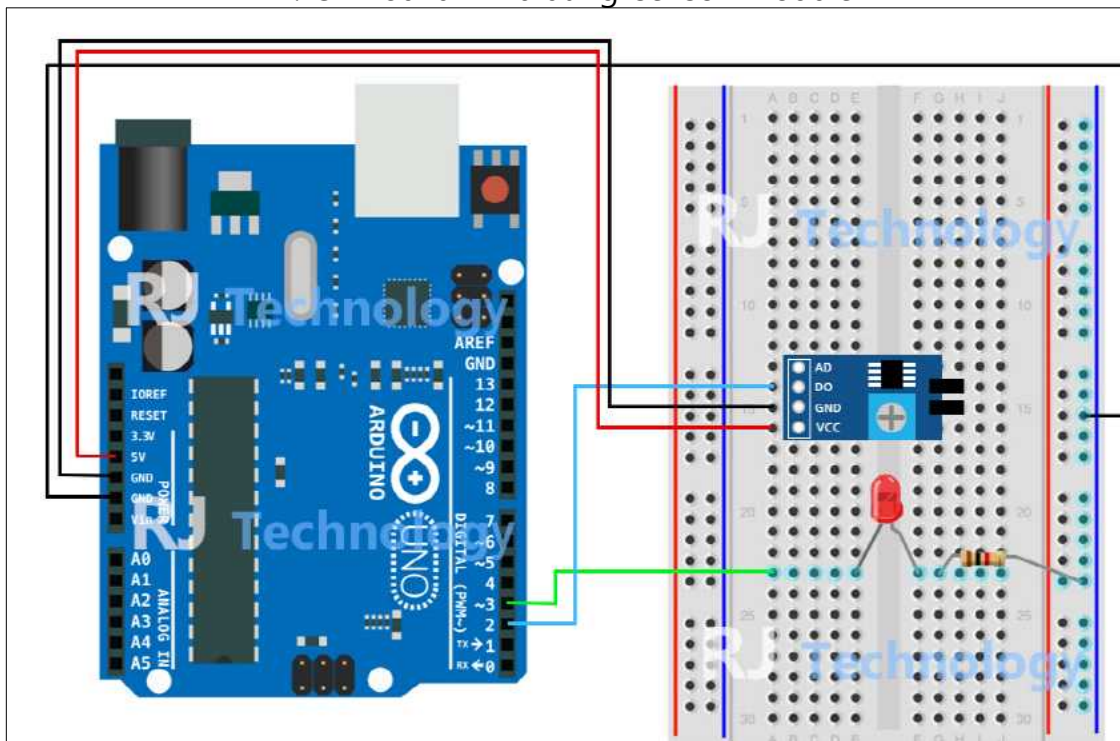
void loop()
{
    for(int i=0; i<5; i++)
    {
        DrawTextFnd(Fndtable[5-i],u_cText[i]);
    }
}

```

3주차, 4주차 (9월 21일 ~ 10월 1일)

- 재료가 도착하기 전까지 각 센서와 모듈, 쉴드에 대해 이해하기 위해 라이브러리와 결선 방법 등을 찾기 위해 구글링을 하였다.
- 원래 만보기 숫자 출력을 OLED Display 모듈로 출력해주기로 했는데 실험에 성공한 7-Segment를 사용해 출력해주기로 하여 재료신청을 수정 하였다.

1. SW18010P Vibrating sensor module



< http://blog.daum.net/_blog/BlogTypeView.do?blogid=0ZPt7&artcleno=1006&_bloghome_menu=recenttext >

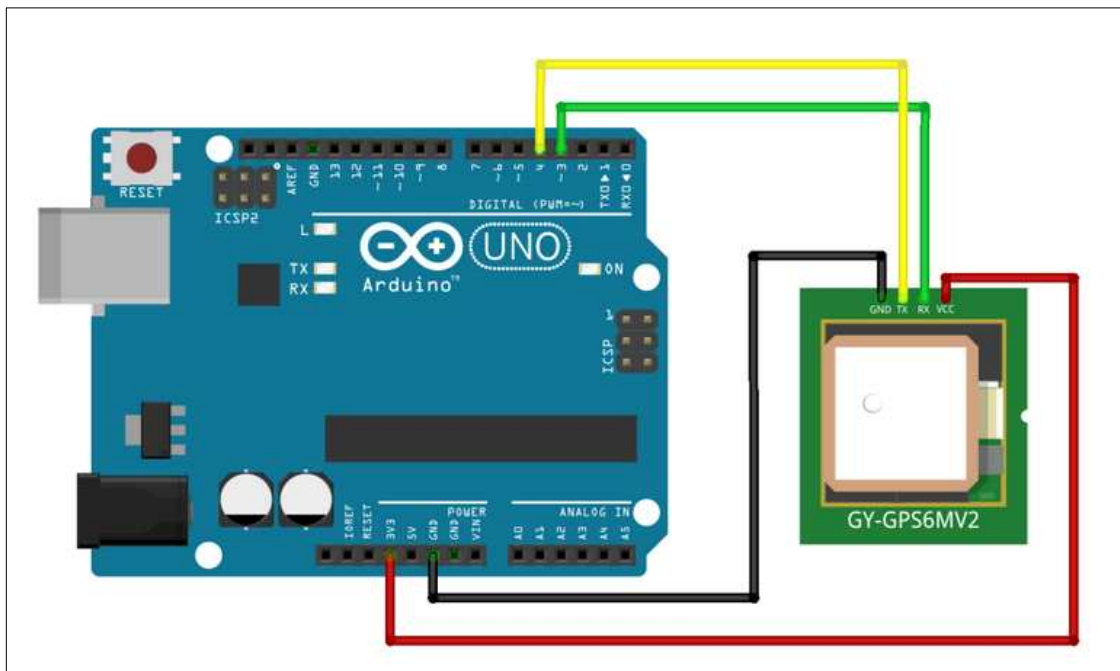
```
int led = 3;
int sensor = 2;

void setup()
{
  pinMode(led, OUTPUT);
  pinMode(sensor, INPUT);
}

void loop()
{
  if( digitalRead(sensor) == LOW )
    digitalWrite(led, HIGH);
  else
    digitalWrite(led, LOW);

  delay(100);
}
```

2. NEO-6M GPS module



· 라이브러리 링크 제공 및 예제 코드 설명

>>> <http://www.hardcopyworld.com/ngine/arduino/index.php/archives/755> >

>>> <http://bbangpan.tistory.com/42>

>>> <http://www.openmakerlab.co.kr/#!Arduino-Intermediated-GPS-%ED%99%9C%EC%9A%A9%ED%95%98%EA%B8%B0/c2413/55cae8440cf2b432d2491328>

· 결선 방법 및 간단한 예제 코드

>>> <http://thefortuneplanet.com/interfacing-arduino-uno-with-gps-module/>

>>> <http://jules.dourlens.com/gps-ublox-neo-6m-and-arduino/>

2. CC3000 WiFi shield



· 라이브러리 링크 제공 및 결선방법, 예제 코드 설명

>>> <http://bbangpan.tistory.com/20>

· 라이브러리 링크 제공 및 결선방법, 예제 설명 및 HTTP통신 예제소스 제공

>>> <http://www.hardcopyworld.com/ngine/arduino/index.php/archives/1958>

5주차(10월 1일 ~ 10월 12일)

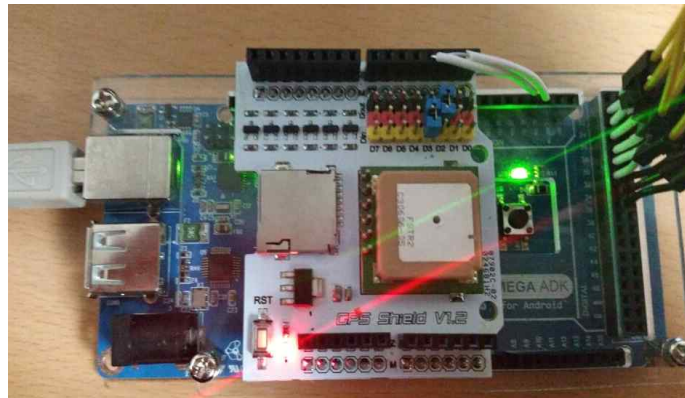
- CNDI GPS Shield를 받아 실험하였다.

TEST 1) Serial monitor에 GPS정보 출력하기

※ 문제점

- UserGuide GPS Shield와 Mega 2560간 핀이맞지 않음 -> 재설정하여 해결

<https://www.youtube.com/watch?v=JYFZ9fvcsdc>



Mega 2560 Pin	GPS Shield
18(TX)	3
19(RX)	2

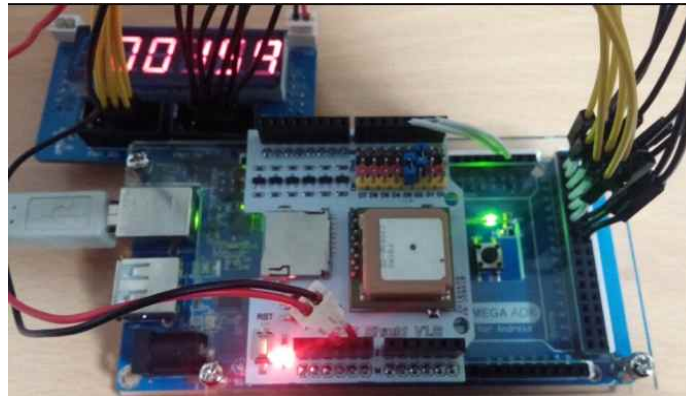


GPS정보값 위치 확인 결과 GPS Shield가 제대로 작동하는 것을 알 수 있다.

TEST 2) GPS Shield + 7-Segment TEST

우선, 7-SEG와 GPS Shield간 핀이 겹치므로 설정을 바꾸어 주었다.

>>> https://www.youtube.com/watch?v=4iM_GxX3rmE&feature=youtu.be



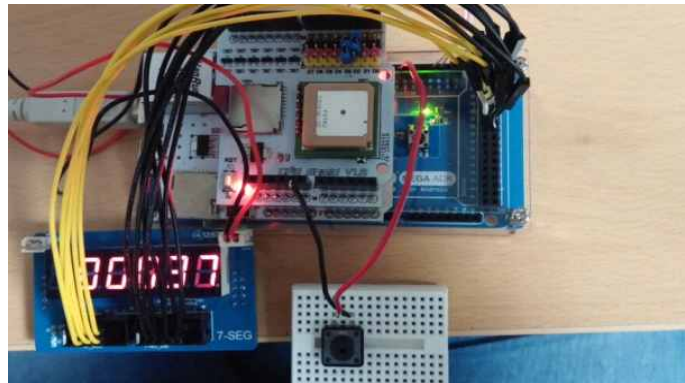
Mega 2560 Pin	GPS Shield
18	3
19	2
Mega 2560 Pin	7-SEG Pin
23	FND_DB A
25	FND_DB B
27	FND_DB C
29	FND_DB D
31	FND_DB E
33	FND_DB F
35	FND_DB G
37	FND_DB DP
22	FND_SEL S0
24	FND_SEL S1
26	FND_SEL S2
28	FND_SEL S3
30	FND_SEL S4
32	FND_SEL S5

TEST 3) GPS Shield + 7-Segment TEST + Button TEST

※ 문제점 <참고 URL - <http://studymake.tistory.com/166>>

- 버튼을 누르는 순간에 바운싱 현상이 발생한다. 바운싱이란, 접점이 붙거나 떨어지는 짧은 순간에 접점이 고속으로 여러번 on/off되는 현상을 말하며 기계적인 스위치라면 반드시 발생하는 현상 -> 디바운싱코드를 구글링 하여 찾았지만 최대한 사용하지 않고 바운싱이 조금 발생하도록 시간차를 두었다.

>>> https://youtu.be/jHNI_qPpDCE



Mega 2560 Pin	GPS Shield
18	3
19	2
Mega 2560 Pin	7-SEG Pin
23	FND_DB A
25	FND_DB B
27	FND_DB C
29	FND_DB D
31	FND_DB E
33	FND_DB F
35	FND_DB G
37	FND_DB DP
22	FND_SEL S0
24	FND_SEL S1
26	FND_SEL S2
28	FND_SEL S3
30	FND_SEL S4
32	FND_SEL S5
Mega 2560 Pin	BUTTON
14	
GND	

GPS + 7SEG + BUTTON(.ino)

```
#include <TinyGPS.h>
#include "FndControl.h"

// TinyGPS 객체 생성
TinyGPS gps;

// 각 digit의 위치
const int Fndtable[6]={
    FND_1_SEL,
    FND_2_SEL,
    FND_3_SEL,
    FND_4_SEL,
    FND_5_SEL,
    FND_6_SEL
};

// 7Segment를 사용해 표현할 숫자
char u_cText[10] = {'0','1','2','3','4','5','6','7','8','9'};
// 각 digit의 숫자를 참조하기 위한 변수
int one = 0, ten = 0, hundred = 0, thousand = 0, ten_hundred = 0;
// 버튼의 포트 번호
const int BUTTON = 14;

void setup()
{
    // 7Segment의 각 digit의 포트와 각 segment의 포트 설정
    FndInit();
    // 버튼의 포트 설정
    pinMode(BUTTON, INPUT_PULLUP);
    // 시리얼 통신 전송속도 115200bps (bit per second)
    Serial.begin(115200);
    // 시리얼 통신 전송속도 9600bps (bit per second)
    Serial1.begin(9600);
    //GPS라이브러리의 버전 출력, TinyGPS내에 있는 정적함수인 버전을 호출한다.
    Serial.print("Simple TinyGPS library v. ");
    Serial.println(TinyGPS::library_version());
    Serial.println();
}

void loop()
```

```

{
    // 버튼이 on상태에서 off가 될때 실행
    if(digitalRead(BUTTON) == LOW){
        one++;
        ten++;
        hundred++;
        thousand++;
        ten_hundred++;
    }

    bool newData = false;
    unsigned long chars;
    unsigned short sentences, failed;
    // 버튼의 바운싱문제를 해결하기 위해 시간을 0.12초로 설정하였다.
    /* start가 millis()함수를 통해 그때의 시간을 받는다. 그리고 milli()함수는 계속 증가하
        고 millis() - start < 120일 경우 for문을 빠져나온다.
    */
    for (unsigned long start = millis(); millis() - start < 120;)
    {
        while (Serial1.available())
        {
            char c = Serial1.read();
            // Serial.write(c); // uncomment this line if you want to see the GPS data flowing
            // gps객체의 encode()함수가 true를 반환했을 경우 newData = true
            if (gps.encode(c))
                newData = true;
        }
        // 지속적으로 7Segment에 숫자 디스플레이
        if(ten%10 == 0) one = 0;
        if(hundred%100 == 0) ten = 0;
        if(thousand%1000 == 0) hundred = 0;
        if(ten_hundred%10000 == 0) thousand = 0;
        DrawTextFnd(Fndtable[0],u_cText[one]);
        DrawTextFnd(Fndtable[1],u_cText[ten/10]);
        DrawTextFnd(Fndtable[2],u_cText[hundred/100]);
        DrawTextFnd(Fndtable[3],u_cText[thousand/1000]);
        DrawTextFnd(Fndtable[4],u_cText[ten_hundred/10000]);
    }
    // newData가 true일 경우
    if(newData)
    {

```

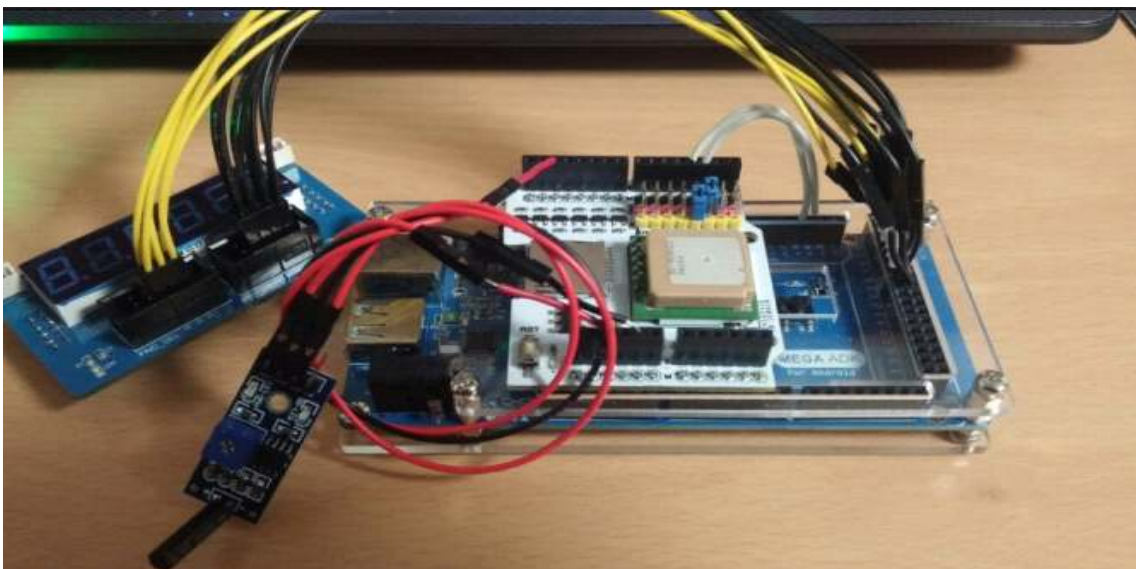
```

float flat, flon;
unsigned long age;
// gps객체 내에 있는 f_get_position()함수를 사용해 gps정보를 받는다.
gps.f_get_position(&flat, &flon, &age);
//Serial.print의 첫번째 인자가 실수일 경우 두번째 인자는 소수점 자릿수를 말한다.
Serial.print("LAT=");
Serial.print(flat == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : flat, 6);//위도
Serial.print(" LON=");
Serial.print(flon == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : flon, 6);//경도
}
// gps의 객체 stats를 사용해 gps의 상태를 확인한다.
gps.stats(&chars, &sentences, &failed);
Serial.print(" CHARS=");
Serial.print(chars);
Serial.print(" SENTENCES=");
Serial.print(sentences);
Serial.print(" CSUM ERR=");
Serial.println(failed);
if (chars == 0)
    Serial.println("** No characters received from GPS: check wiring **");
}

```

6주차(10월 13일 ~ 10월 19일)

- 우선 만보기의 기능을 구현하기 위해 스위치를 진동센서로 대신하였고 이를 실험하였다. 코드 부분은 핀 설정을 14번에서 13번만 바꿔준 것을 제외하면 동일하다. < URL - <https://youtu.be/1GNuhFUPGxw> >

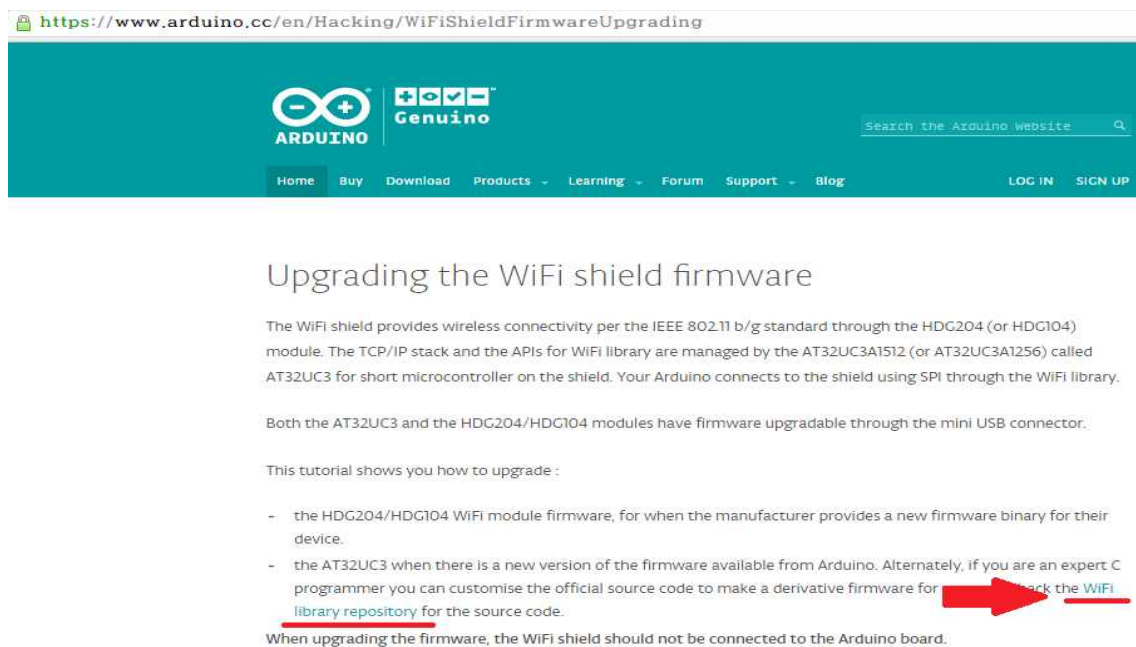


- WiFi Shield를 활용해 예제프로그램을 실행하였다.

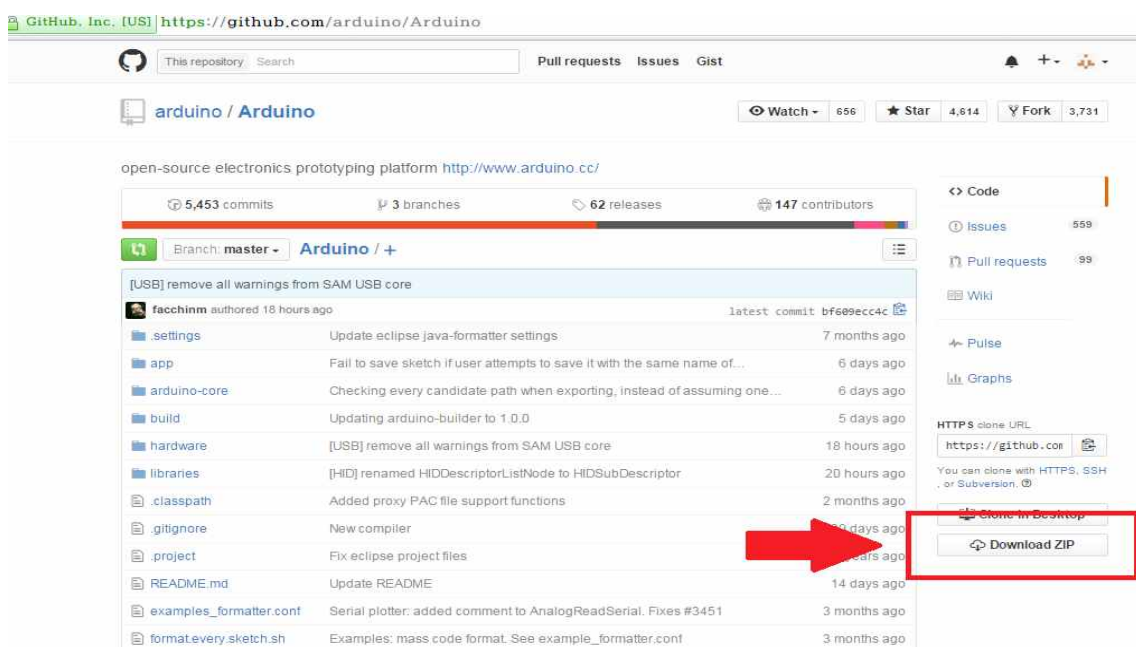
※ 문제점

- 낮은 버전 firmware이므로 업그레이드 하여야 함

① <https://www.arduino.cc/en/Hacking/WiFiShieldFirmwareUpgrading>로 접속 후 WiFi library repository 클릭



② github에서 최신 firmware 오픈소스를 다운로드한다.



③ <http://www.atmel.com/tools/FLIP.aspx>로 접속 후 아래의 파일을 다운

www.atmel.com/tools/FLIP.aspx

Ordering information

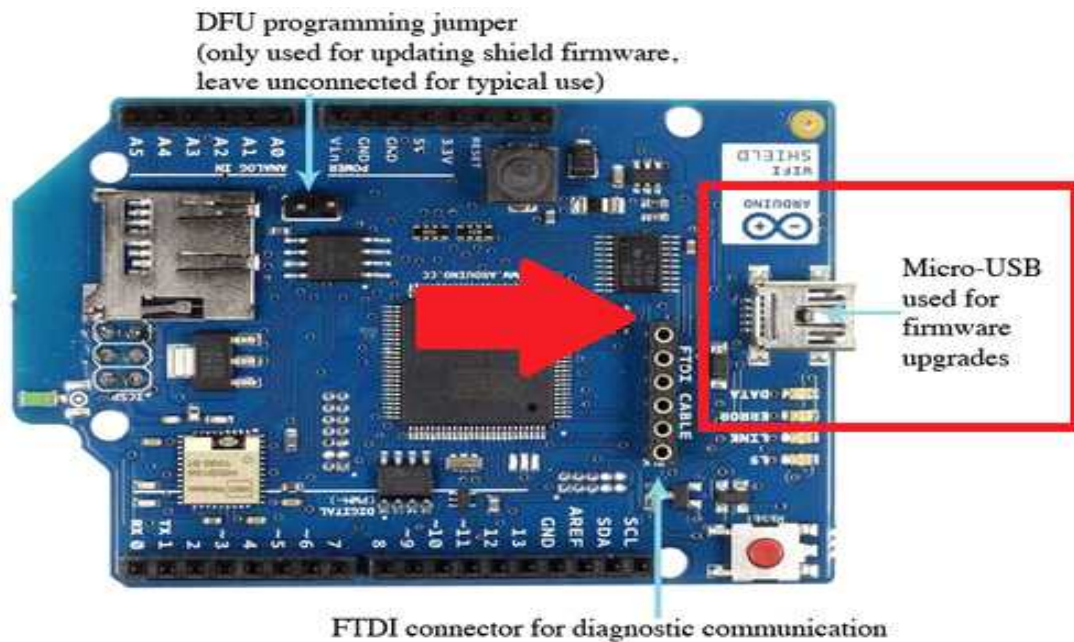
Ordering Code	Atmel Store Availability ¹	Unit Price (USD) ²	Buy Online
FLIP	0	Varies	Check Distributor

¹Backlog orders can be placed for items currently not available.
²Suggested retail price per unit for budgetary use only.

Software Description

- FLIP 3.4.7 for Windows (requires Java Runtime Environment)
 (5.7MB, updated August 2012)
 Runs on Windows 98/Me/NT/2000/XP/Vista/7/8
- FLIP 3.4.7 for Windows (Java Runtime Environment included)**
 (21MB, updated August 2012)
 Runs on Windows 98/Me/NT/2000/XP/Vista/7/8
- FLIP 3.2.1 for Linux x86 (require Java Runtime Environment)
 (1.3MB, updated October 2007)
 Runs on Linux x86
- FLIP 2.4.6 for Windows
 (4.6MB, updated February 2010)
 Runs on Windows 98/Me/NT/2000/XP/Vista/7.
 This version is kept available because FLIP 3 does not support Smart card readers ICs.

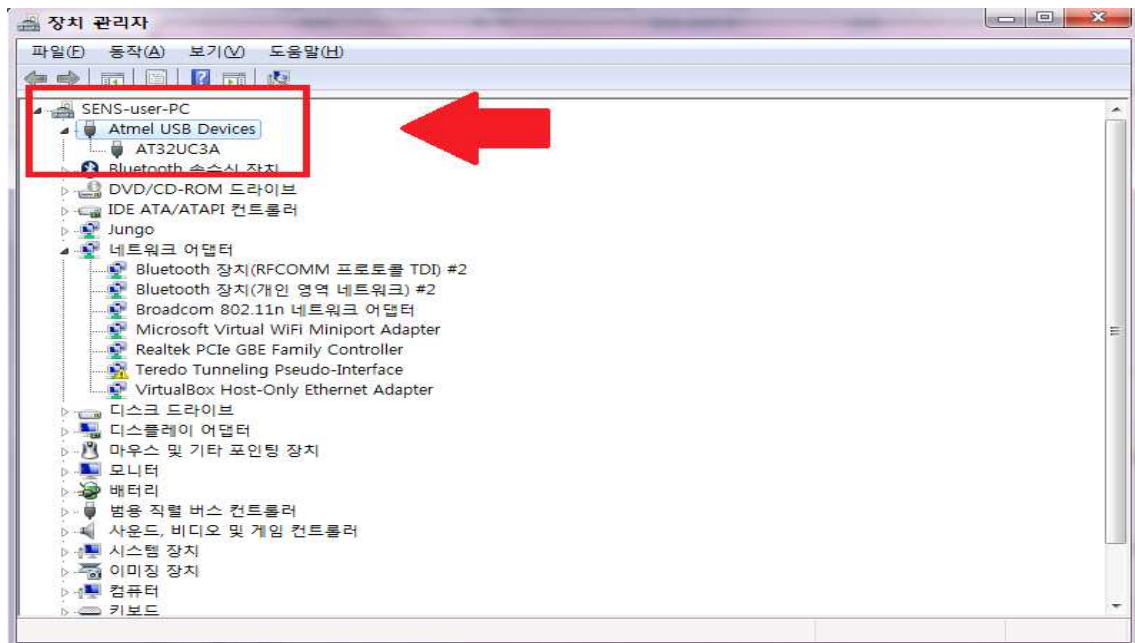
④ 와이파이 실드의 Micro-USB부분에 포트 연결, DFU porgramming jumper에 있는 점퍼선 결함





⑤ 아두이노 스튜디오의 버전이 최신일 경우 포트를 연결하면 드라이버 소프트웨어가 자동 설치된다.

그렇지 않을 경우 내 컴퓨터 -> 속성 -> 장치관리자 -> AT32UC3A 마우스 오른쪽 클릭 -> 드라이버 소프트웨어 업데이트 클릭



⑥ 아래의 버튼 클릭 -> C:\Program Files\Atmel\Flip 3.4.7\usb 검색 후 다음 클릭, 그러면 드라이버 소프트웨어 업그레이드가 진행된다.



⑦ cmd창을 열고 아래의 명령어 순서에 맞게 실행한다.

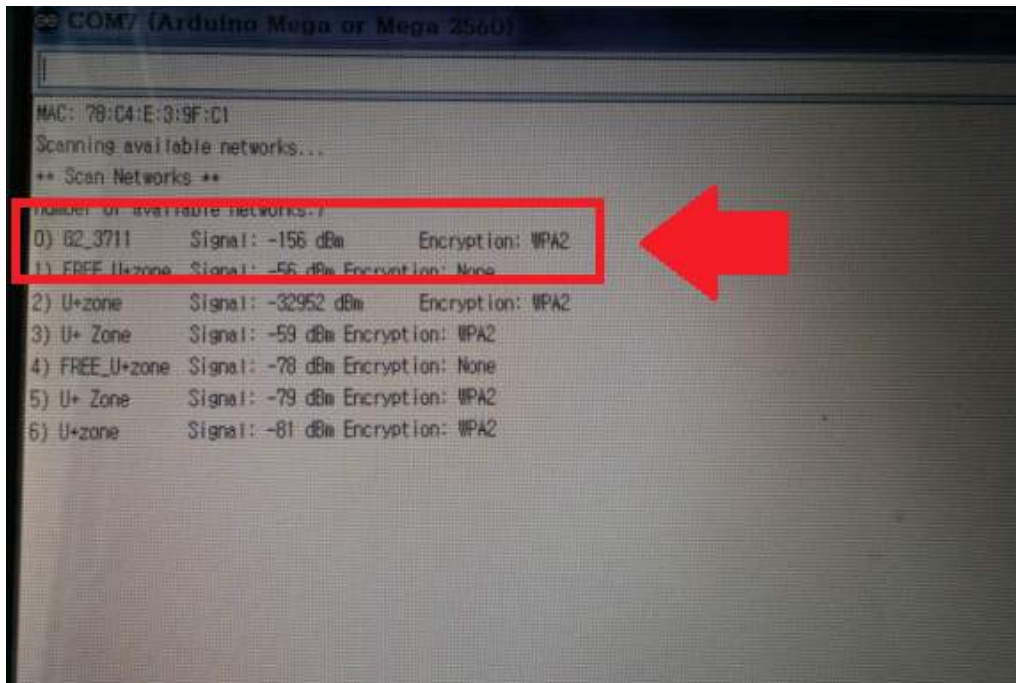
※ wifi_dnld.elf, wifiHD.elf 두 파일의 위치 주의한다.

1. cd C:\Program Files\Atmel\Flip 3.4.7\bin
2. batchisp.exe -device AT32UC3A1256 -hardware usb -operation erase f memory flash blankcheck loadbuffer "C:/Program Files (x86)/Arduino/hardware/arduino/firmwares/wifishield/binary/wifi_dnld.elf" program verify start reset 0
3. WiFi Shield의 RESET버튼 클릭
4. batchisp.exe -device AT32UC3A1256 -hardware usb -operation erase f memory flash blankcheck loadbuffer "C:/Program Files (x86)/Arduino/hardware/arduino/firmwares/wifishield/binary/wifiHD.elf" program verify start reset 0

⑧ 업그레이드가 완료 후 DFU programming jumper에 있는 점퍼선 결함을 해제한다.

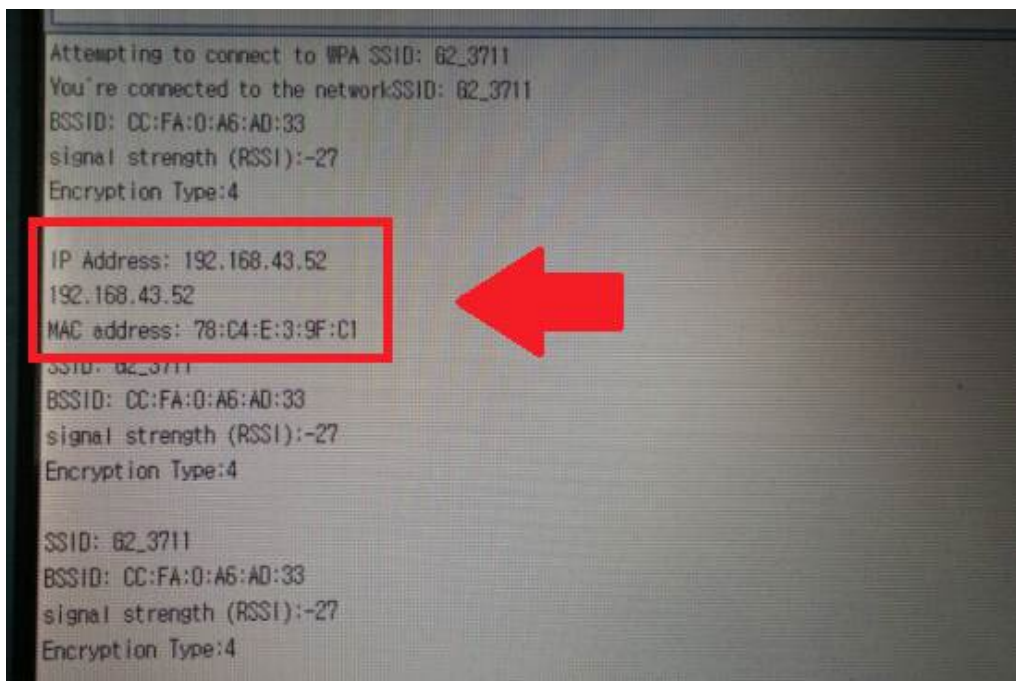
TEST 1) 연결 가능한 네트워크 검색하기

소스코드는 ScanNetworks 예제프로그램 파일과 동일하다.



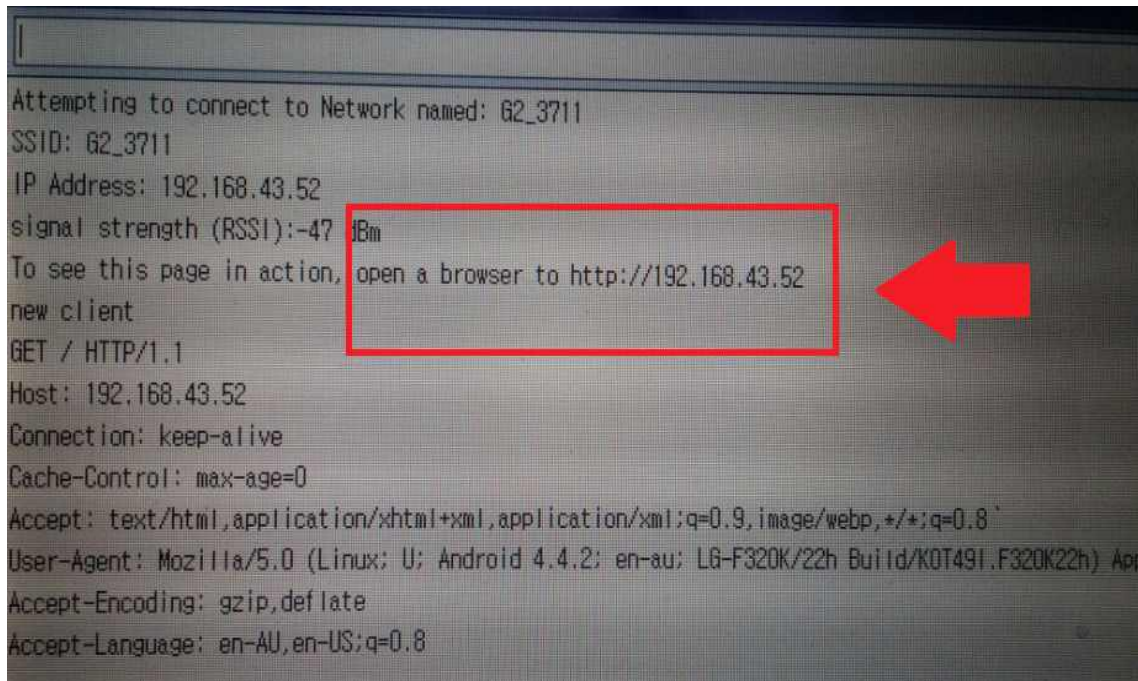
TEST 2) 와이파이 쉼트의 맥주소와 아이피주소 확인하기

소스코드는 ConnectWithWPA 예제프로그램 파일과 동일하다.



TEST 3) 와이파이로 할당 받은 로컬서버 접속하기

소스코드는 SimpleWebServerWiFi예제프로그램 파일과 동일하다.



< 와이파이실드의 로컬 서버 접속 화면 >

TEST 4) WiFi Shield + GPS Shield + 7Segment + 진동센서

모든 재료를 결선하고 실험하였다. 그리고 로컬서버에 접속해 만보기의 값을 초기화 할 수 있도록 제어하였다. 마지막으로, 감도가 좋지 않아 진동센서의 INPUT값이 0이 두 번 들어올 경우로 디바운싱 하였다.

실행 동영상 URL - <https://youtu.be/9E5iOXAKes0>

WiFi Shield + GPS + 7SEG + BUTTON(.ino)

```
#include <SPI.h>
#include <WiFi.h>
#include <TinyGPS.h>
#include "FndControl.h"

// TinyGPS 객체 생성
TinyGPS gps;

char ssid[] = "G2_3711";//내 네트워크 ID, 핫스팟 사용
char pass[] = "12345678";//내 네트워크 PW
int keyIndex = 0;//my network key Index number(needed only for WEP, usually it's 0)

int status = WL_IDLE_STATUS;
WiFiServer server(80);//로컬 웹서버 포트번호

// 각 digit의 위치
const int Fndtable[6]={
  FND_1_SEL,
  FND_2_SEL,
  FND_3_SEL,
  FND_4_SEL,
  FND_5_SEL,
  FND_6_SEL
};

// 7Segment를 사용해 표현할 숫자
char u_cText[10] = {'0','1','2','3','4','5','6','7','8','9'};
// 각 digit의 숫자를 참조하기 위한 변수
int one = 0, ten = 0, hundred = 0, thousand = 0, ten_hundred = 0,
ten_thousand = 0;
// 버튼의 포트 번호
const int sensor = 13;
```

```

float lat = 0.0, lon = 0.0; // 서버로 보내주기 위한 위도와 경도 변수
void setup()
{
    // 7Segment의 각 digit의 포트와 각 segment의 포트 설정
    FndInit();
    // 버튼의 포트 설정
    pinMode(sensor, INPUT);
    // 시리얼 통신 전송속도 115200bps (bit per second)
    Serial.begin(115200);
    // 시리얼 통신 전송속도 9600bps (bit per second)
    Serial1.begin(9600);
    //GPS라이브러리의 버전 출력, TinyGPS내에 있는 정적함수인 버전을 호출한다.
    Serial.print("Simple TinyGPS library v. ");
    Serial.println(TinyGPS::library_version());
    Serial.println();

    // 와이파이 코드
    if (WiFi.status() == WL_NO_SHIELD) {
        Serial.println("WiFi shield not present");
        // 와이파이 쉴드가 없다면 실행하지 않는다.
        while (true);
    }
    // 펌웨어 버전
    String fv = WiFi.firmwareVersion();
    if ( fv != "1.1.0" )
        Serial.println("Please upgrade the firmware");

    // 네트워크 연결 시도
    while ( status != WL_CONNECTED) {
        Serial.print("Attempting to connect to Network named: ");
        Serial.println(ssid);
        // Connect to WPA/WPA2 network. Change this line if using open or WEP network:
        status = WiFi.begin(ssid, pass);
        // 와이파이 접속을 위해 10초를 기다린다.
        delay(10000);
    }
    server.begin();// 웹 서버 접속
    printWifiStatus();// 웹 서버에 접속 후 와이파이 상태
}
void loop()
{

```

```

WiFiClient client = server.available(); // listen for incoming clients
// LOW값이 두 번 들어올 경우로 디바운싱처리
if(digitalRead(sensor) == LOW){
    if(digitalRead(sensor) == LOW){
        one++;
        ten++;
        hundred++;
        thousand++;
        ten_hundred++;
        ten_thousand++;
    }
    bool newData = false;
    unsigned long chars;
    unsigned short sentences, failed;
    // 버튼의 바운싱문제를 해결하기 위해 시간을 0.12초로 설정하였다.
    /* start가 millis()함수를 통해 그때의 시간을 받는다. 그리고 milli()함수는 계속 증가하
        고 millis() - start < 120일 경우 for문을 빠져나온다.
    */
    for (unsigned long start = millis(); millis() - start < 20;)
    {
        // 버튼이 on상태에서 off가 될때 실행

        while (Serial1.available())
        {
            char c = Serial1.read();
            // Serial.write(c); // uncomment this line if you want to see the GPS data flowing
            // gps객체의 encode()함수가 true를 반환했을 경우 newData = true
            if (gps.encode(c))
                newData = true;
        }
        // 지속적으로 7Segment에 숫자 디스플레이
        if(ten%10 == 0) one = 0;
        if(hundred%100 == 0) ten = 0;
        if(thousand%1000 == 0) hundred = 0;
        if(ten_hundred%10000 == 0) thousand = 0;
        DrawTextFnd(Fndtable[0],u_cText[one]);
        DrawTextFnd(Fndtable[1],u_cText[ten/10]);
        DrawTextFnd(Fndtable[2],u_cText[hundred/100]);
        DrawTextFnd(Fndtable[3],u_cText[thousand/1000]);
        DrawTextFnd(Fndtable[4],u_cText[ten_hundred/10000]);
        DrawTextFnd(Fndtable[5],u_cText[ten_thousand/100000]);
    }
}

```

```

}
// newData가 true일 경우
if(newData)
{
    float flat, flon; // GPS의 위도와 경도
    unsigned long age;
    // gps객체 내에 있는 f_get_position()함수를 사용해 gps정보를 받는다.
    gps.f_get_position(&flat, &flon, &age);
    //Serial.print의 첫번째 인자가 실수일 경우 두번째 인자는 소수점 자릿수를 말한다.
    Serial.print("LAT=");
    Serial.print(flat == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : flat, 6);//위도
    Serial.print(" LON=");
    Serial.print(flon == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : flon, 6);//경도
}
// gps의 객체 stats를 사용해 gps의 상태를 확인한다.
gps.stats(&chars, &sentences, &failed);
Serial.print(" CHARS=");
Serial.print(chars);
Serial.print(" SENTENCES=");
Serial.print(sentences);
Serial.print(" CSUM ERR=");
Serial.println(failed);
if (chars == 0){
    Serial.println("** No characters received from GPS: check wiring **");
}
if (client) {// 클라이언트로서 이벤트가 발생하면
    Serial.println("new client");//시리얼모니터에 클라이언트 접속 출력
    String currentLine = "";
    while (client.connected()){//클라이언트가 접속 중이라면
        if (client.available()) {//클라이언트에 입력 데이터가 있으면
            char c = client.read()
            Serial.write(c);
            if (c == '\n') {
                // if the current line is blank, you got two newline characters in a row.
                // that's the end of the client HTTP request, so send a response:
                if (currentLine.length() == 0) {
                    // HTTP headers always start with a response code (e.g. HTTP/1.1 200 OK)
                    // and a content-type so the client knows what's coming, then a blank line:
                    client.println("HTTP/1.1 200 OK");
                    client.println("Content-type:text/html");
                    client.println();
                }
            }
        }
    }
}

```



```

        // the content of the HTTP response follows the header:
        client.print("Click <a href=\"/L\">here</a> to reset the 7segment on pin 13 <br>");
        // The HTTP response ends with another blank line:
        //client.print(lat);
        //client.print(lon);
        // break out of the while loop:
        break;
    }
    else { // if you got a newline, then clear currentLine:
        currentLine = "";
    }
}
else if (c != '\r') { // if you got anything else but a carriage return character,
    currentLine += c; // add it to the end of the currentLine
}
// GET /L을 받으면 세그먼트의 값 리셋
if (currentLine.endsWith("GET /L")) {
    one = 0;
    ten = 0;
    hundred = 0;
    thousand = 0;
    ten_hundred = 0;
    ten_thousand = 0;
}
}
}

// 클라이언트 연결 종료
client.stop();
Serial.println("client disconnected");
}
}

// 와이파이의 상태를 시리얼 모니터에 출력
void printWifiStatus(){
    // print the SSID of the network you're attached to:
    Serial.print("SSID: ");
    Serial.println(WiFi.SSID());

    // 와이파이의 주소 출력
    IPAddress ip = WiFi.localIP();
    Serial.print("IP Address: ");

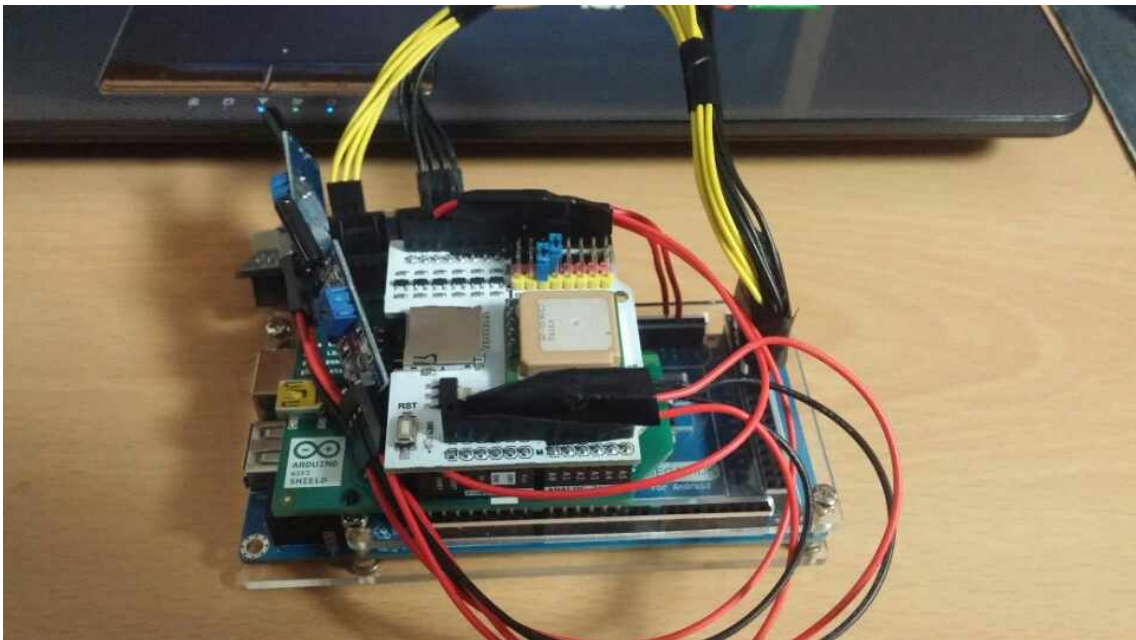
```

```

Serial.println(ip);
// 신호 강도 표시
long rssi = WiFi.RSSI();
Serial.print("signal strength (RSSI):");
Serial.print(rssi);
Serial.println(" dBm");
// 웹 서버 ip주소 출력
Serial.print("To see this page in action, open a browser to http://");
Serial.println(ip);
}

```

7주차(10월 19일 ~ 10월 28일)



- GPS 모듈의 핀을 재설정 하였고 진동센서의 민감도가 떨어져 만보기의 기능을 제대로 하지 못해 진동센서를 추가로 결선하였다.
- 6주차에서는 각각의 재료를 합쳐 각각의 기능을 하는지를 실험하였다면 7주차에서는 스마트만보기 기능 구현을 위해 WiFiWebClient 예제를 응용하여 WiFi 연결시 GPS정보를 웹에 연동할 수 있는지를 실험하였다. WiFiWebClient 예제프로그램은 와이파이шил드를 사용해서 해당 웹 서버로 HTTP통신을 요청하는 예제다.

· HTTP통신 이란?

웹상에서 클라이언트와 서버가 서로 대화하기 위한 통신 프로토콜(계약)이다. HTTP는 TCP/IP를 통해 연결이 이루어지며 기본 포트는 80포트이다.

네트워크 전송 프로토콜 TCP는 전송데이터를 일정 단위로 나누어 포장하는 것에 대한 계약이고 IP는 직접 데이터를 주고받는 계약이다. IP는 출발과 도착 주소를 가지는 헤더를 참조해 네트워크를 통해 데이터를 목적지에 전달한다. 그러나 이 과정에서 데이터가 잘 못 전달되거나 파손될 수 있고 보내진 순서대로 데이터를 전송하지 못할 수 있다. 이러한 단점을 보완하는 것이 TCP이다.

HTTP프로토콜도 TCP/IP, UDP등과 같은 socket 통신이며 TCP 프로토콜 위에 올려놓은 Application레벨의 protocol이다.

HTTP는 요청 / 응답 모델에 기반 한다. 클라이언트는 요청 자원의 URL과 헤더의 정보를 포함하여 웹 서버에 요청한다. 웹 서버는 클라이언트의 요청이 들어오면 자원과 헤더정보를 분석하여 요청한 자원의 데이터와 헤더를 포함하여 클라이언트에게 응답한다. HTTP 요청 방식으로는 GET방식과 POST 방식 등이 존재한다.

· HTTP 통신의 GET방식

- 클라이언트가 URL뒤에 직접적으로 데이터를 붙여서 요청한다.
- 데이터가 주소에 표시되므로 제한적이고 보안에 취약하다.
- 요청 자체를 위한 정보를 전송할 때 주로 사용한다.

· HTTP 통신의 POST방식

- POST방식은 GET방식과 달리 데이터가 URL뒤에 직접 붙어 있지 않고 바디 부분에 숨겨져 인코딩 후 서버로 전달된다.
- 인코딩으로 인해 GET방식보다 작업이 느리다는 단점이 있다.
- 서버 측에서의 정보 갱신 작업을 원할 때 주로 사용한다.

· 참고 URL

HTTP통신 이란? <http://najuung.tistory.com/entry/HTTP%ED%94%84%EB%A1%9C%ED%86%A0%EC%BD%9C%EC%9D%B4%EB%9E%80>

GET방식과과 POST방식 <http://soul0.tistory.com/185>
<http://najuung.tistory.com/entry/JSP-POST%EC%99%80-GET%EC%9D%98-%EC%B0%A8%EC%9D%B4%EC%A0%90>
http://www.w3schools.com/tags/ref_httpmethods.asp

※ 문제점

- HTTP요청 시 아두이노 딜레이 현상으로 세븐세그먼트에 숫자가 지속적으로 출력되지 않는다. -> 아두이노 라이브러리에서 WiFi.cpp파일에서 connect 함수에서 delay문을 주석처리 해서 해결하였다.

- 7Segment 동작 시 가장 위쪽 세그먼트의 불이 항상 켜져있다. -> 소프트웨어적으로는 문제가 없어보이므로 교체를 해야할 것 같다.

실행 동영상 URL - <https://youtu.be/v0zuqCOZOZQ>

WiFi로 GPS정보를 웹 지도에 출력하기(.ino)

```
// WiFi Shield 헤더
#include <SPI.h>
#include <WiFi.h>
// GPS Shield 헤더
#include <TinyGPS.h>
// 7Segment 헤더
#include "FndControl.h"

TinyGPS gps;      // TinyGPS 객체 생성
double flat, flon;// GPS칩드 예제에서 지역변수 였던 위도, 경도 변수를 전역변수 선언

char ssid[] = "G2_3711"; //사용자 네트워크 ID, 핫스팟 사용
char pass[] = "12345678"; //사용자 네트워크 PW
//my network key Index number(needed only for WEP, usually it's 0)
int keyIndex = 0;

int status = WL_IDLE_STATUS; //와이파이의 상태를 나타냄

char server[] = "DB아이디.dothome.co.kr"; //HTTP통신을 요청할 서버

WiFiClient client; // WiFiClient객체 선언

// http요청의 주기를 위한 카운트변수
unsigned int cnt;
String p = "GET /dbconfig.php?latitude=";
String p2 = "&longitude=";
String p3 = " HTTP/1.1";

// 각 digit의 위치
const int Fndtable[6]={
```

```

FND_1_SEL,
FND_2_SEL,
FND_3_SEL,
FND_4_SEL,
FND_5_SEL,
FND_6_SEL
};

// 7Segment를 사용해 표현할 숫자
char u_cText[10] = {'0','1','2','3','4','5','6','7','8','9'};
// 각 digit의 숫자를 참조하기 위한 변수
int one = 0, ten = 0, hundred = 0, thousand = 0, ten_hundred = 0,
ten_thousand = 0;

// 진동센서의 포트 번호
const int sensor = 13;
const int sensor2 = 12;

void setup()
{
    // 카운트를 0으로 초기화
    cnt = 0;
    // 7Segment의 각 digit의 포트와 각 segment의 포트 설정
    FndInit();

    // 진동센서의 포트 설정
    pinMode(sensor, INPUT);
    pinMode(sensor2, INPUT);

    // 시리얼 통신 전송속도 115200bps (bit per second)
    Serial.begin(115200);
    // 시리얼 통신 전송속도 9600bps (bit per second)
    Serial1.begin(9600);

    //GPS라이브러리의 버전 출력, TinyGPS내에 있는 정적함수인 버전을 호출한다.
    Serial.print("Simple TinyGPS library v. ");
    Serial.println(TinyGPS::library_version());
    Serial.println();

    // 와이파이 칩드가 없을경우
    if (WiFi.status() == WL_NO_SHIELD) {
        Serial.println("WiFi shield not present");
    }
}

```

```

    //while (true);// don't continue
}
// 와이파이의 펌웨어버전이 낮을경우
String fv = WiFi.firmwareVersion();
if ( fv != "1.1.0" )
    Serial.println("Please upgrade the firmware");

// 와이파이로 접속 시도
while ( status != WL_CONNECTED) {
    Serial.print("Attempting to connect to Network named: ");
    Serial.println(ssid);

    // Connect to WPA/WPA2 network. Change this line if using open or WEP
network:
    status = WiFi.begin(ssid, pass);
    // 와이파이 접속을 위해 10초
    delay(10000);
}
Serial.println("Connected to wifi");
Serial.println("\nStarting connection to server...");
}

void loop()
{
    /*진동센서의 민감도를 해결하기 위해 2개를 사용하였다.
    둘 중 하나라도 low값이 들어올 경우 ++연산수행
    */
    if( (digitalRead(sensor) == LOW) || (digitalRead(sensor2) == LOW)){
        one++;
        ten++;
        hundred++;
        thousand++;
        ten_hundred++;
        ten_thousand++;
    }

    bool newData = false;
    unsigned long chars;
    unsigned short sentences, failed;
    // 진동센서의 바운싱문제를 해결하기 위해 시간을 0.02초로 설정하였다.
    /* start가 millis()함수를 통해 그때의 시간을 받는다. 그리고 milli()함수는 계속 증가하

```

고 millis() - start < 20일 경우 for문을 빠져나온다.

```
*/
for (unsigned long start = millis(); millis() - start < 20;)
{
    while (Serial1.available())
    {
        char c = Serial1.read();
        // Serial.write(c); // uncomment this line if you want to see the GPS data
        flowing
        // gps객체의 encode()함수가 true를 반환했을 경우 newData = true
        if (gps.encode(c))
            newData = true;
    }
    // 지속적으로 7Segment에 숫자 디스플레이
    if(ten%10 == 0) one = 0;
    if(hundred%100 == 0) ten = 0;
    if(thousand%1000 == 0) hundred = 0;
    if(ten_hundred%10000 == 0) thousand = 0;
    DrawTextFnd(Fndtable[0],u_cText[one]);
    DrawTextFnd(Fndtable[1],u_cText[ten/10]);
    DrawTextFnd(Fndtable[2],u_cText[hundred/100]);
    DrawTextFnd(Fndtable[3],u_cText[thousand/1000]);
    DrawTextFnd(Fndtable[4],u_cText[ten_hundred/10000]);
    DrawTextFnd(Fndtable[5],u_cText[ten_thousand/100000]);
}
// newData가 true일 경우
if(newData)
{
    unsigned long age;
    // gps객체 내에 있는 f_get_position()함수를 사용해 gps정보를 받는다.
    // TinyGPS라이버러리 내에서 flat과 flon float에서 double형으로 타입변환
    gps.f_get_position(&flat, &flon, &age);
    //Serial.print의 첫번째 인자가 실수일 경우 두번째 인자는 소수점 자릿수를 말한다.
    Serial.print("LAT=");
    Serial.print(flat == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : flat, 6);//위도
    Serial.print(" LON=");
    Serial.print(flون == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : flon, 6);//경도
    Serial.println("Success");
    // 8 ~ 12초를 고려하여 GPS값을받았다면 http요청을한다.
    // 시간간격을 충분히 두지 않으면 소켓을 비울 시간이 없을 뿐더러 이로인해 멈춤현
    상이 발생한다.
```

```

    if(cnt % 400 > 375){
        httpRequest();
    }
}
// gps의 객체 stats를 사용해 gps의 상태를 확인한다.
gps.stats(&chars, &sentences, &failed);
Serial.print(" CHARS=");
Serial.print(chars);
Serial.print(" SENTENCES=");
Serial.print(sentences);
Serial.print(" CSUM ERR=");
Serial.println(failed);
if (chars == 0){
    Serial.println("** No characters received from GPS: check wiring **");
}
cnt++;
Serial.println(cnt);
}

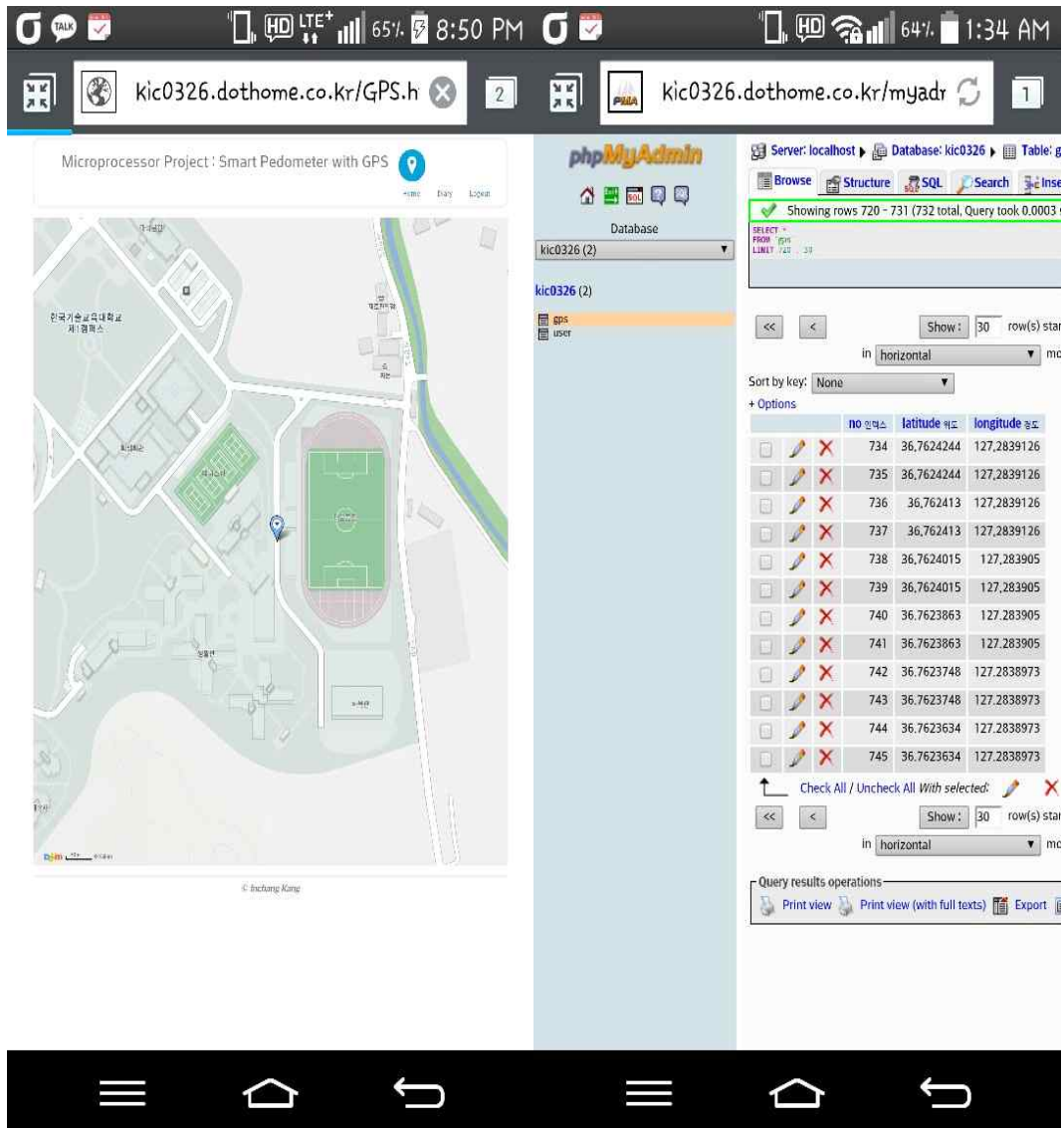
void httpRequest() {

    // close any connection before send a new request.
    // This will free the socket on the WiFi shield
    // 소켓을비운다.
    client.stop();
    client.connect(server, 80);
    Serial.println("connected to server");

    // Make a HTTP request:
    // http요청
    client.print(p);
    client.print(flat, 6);
    client.print(p2);
    client.print(flon, 6);
    client.println(p3);
    client.println("Host: DB아이디.dothome.co.kr");
    client.println("Connection: close");
    client.println();
}

```


웹 애플리케이션 모바일 실행 화면(GPS지도, DB)



8주차 [10월 28일 ~ 12월 7일]

- 작품을 휴대용으로 사용할 수 있도록 배터리 홀더를 주문하였고 진동센서로 만보기 기능을 제대로 구현하지 못하여 기울기 센서를 따로 주문 후 만보기 기능을 구현하였다.
- 계획이었던 세븐세그먼트에 걸음 수를 추력해주는 것에 기능을 추가하기 위해 디스플레이 모듈을 OLED로 대체하였다. 런닝머신에 디스플레이 되는 정보들을 출력하기 위해 작동 시간은 타이머라이브러리를 사용 하였다. 속도는 GPS쉴드에서 받은 값을 출력하고 걸음 수는 기울기센서의 값에 값이 HIGH값이 들어올 경우 ++연산 하였다. 그리고 바운싱 문제를 해결하기 위해 HIGH값이 두 번 들어올 경우로 제한을 두었다. 마지막으로 칼로리는 공식을 활용해 출력하였다.
- 필요하지 않은 코드, 기존 세븐세그먼트 코드를 모두 제거하고 코드를 최소화 하였다.

- 아두이노 작품 최종코드

```
// WiFi Shield를 사용하기 위한 헤더
#include <SPI.h>
#include <WiFi.h>
// GPS Shield를 사용하기 위한 헤더
#include <TinyGPS.h>
// OLED LC를 사용하기 위한 헤더
#include <Wire.h>
#include <OzOLED.h>
#include "U8glib.h"
// 타이머3(아두이노 메가 지원)를 사용하기 위한 헤더
#include <TimerThree.h>

U8GLIB_SSD1306_128X64 u8g(U8G_I2C_OPT_NONE);

TinyGPS gps; // TinyGPS 객체 생성
float flat = 0.0, flon = 0.0, fspeed = 0.0, fcourse = 0.0; // GPS쉴드 예제에서 지역변수였던 위도, 경도, 속도, 방위 변수를 전역변수 선언

char ssid[] = "G2_3711"; //사용자 네트워크 ID, 핫스팟 사용
char pass[] = "12345678"; //사용자 네트워크 PW
```

```

//my network key Index number(needed only for WEP, usually it's 0)
int keyIndex = 0;

int status = WL_IDLE_STATUS; //와이파이의 상태를 나타냄

char server[] = "kic0326.dothome.co.kr"; //HTTP통신을 요청할 서버

WiFiClient client; // WiFiclient객체 선언

volatile long cnt = 0, steps = 0, second = 0, second2 = 0, minute = 0;
// OLED DISPLAY VALUES steps 걸음 수, cal 칼로리 소모량, second 초, minute분
volatile float cal = 0.0; // cnt는 타이밍을 재기위한 전역 변수

// http통신을 위한 문자열들
String p = "GET /dbconfig.php?latitude=";
String p2 = "&longitude=";
String p3 = "&course=";
String p4 = " HTTP/1.1";

const int sensor = 13; // 기울기센서 핀 번호

void setup()
{
  OzOled.init(); //initialize Oscar OLED display
  pinMode(sensor, INPUT); // 기울기센서의 핀설정
  // 메가보드는 4개의 i/o ports를 갖는다.
  // 시리얼 통신 전송속도 9600bps (bit per second), GPS값을 받기 위한 시리얼 통신
  Serial1.begin(9600);
  // 통신 전송속도 115200bps (bit per second), GPS값 시리얼모니터에 출력을 위한 시리얼 통신
  Serial.begin(115200);
  // 와이파이 접속
  while ( status != WL_CONNECTED) {
    Serial.print("Attempting to connect network named: ");
    Serial.println(ssid); // 네트워크 이름 출력 (SSID);
    // Connect to WPA/WPA2 network. Change this line if using open or WEP network:
    status = WiFi.begin(ssid, pass);
    // 네트워크 접속 타이밍 10초
    delay(10000);
  }
  // 타이머의 시간체크 1초로 설정
  Timer3.initialize(1000000);

```

```

// attaches callback() as a timer overflow interrupt
Timer3.attachInterrupt(callback);
}
void loop()
{
    // 진동센서 디바운싱을 위해 바로 LOW값을 넣어 준다.
    if( digitalRead(sensor) == HIGH){
        digitalWrite(sensor, LOW);
        steps++;
    }
    digitalWrite(sensor, LOW);
    bool newData = false;// gps의 상태변수
    unsigned long chars;
    unsigned short sentences, failed;
    // 진동센서 디바운싱을 위한 코드
    // for문 내에서 계속해서 gps값을 얻는다. 그리고 0.12초 마다 for문을 빠져나옴.
    for (unsigned long start = millis(); millis() - start < 120;)
    {
        while (Serial1.available())
        {
            char c = Serial1.read();
            // Serial.write(c); // uncomment this line if you want to see the GPS data flowing
            // gps객체의 encode()함수가 true를 반환했을 경우 newData = true
            if (gps.encode(c))
                newData = true;
        }
    }
    // newData가 true일 경우
    if(newData)
    {
        unsigned long age;
        // gps객체 내에 있는 f_get_position()함수를 사용해 gps정보를 받는다.
        gps.f_get_position(&flat, &flon, &age);
        // TinyGPS라이브러리 내에 있는 km/h speed값을 받아온다.
        fspeed = gps.f_speed_kmph();
        // TinyGPS라이브러리 내에 있는 course값을 받아온다.
        // ~0(360) 은 north, ~180 은 south, ~270 은 west, ~90 은 east
        fcourse = gps.f_course();
        // Serial.print 첫번째 인자가 실수일 경우 두번째 인자는 소수점 자릿수를 말한다.
        Serial.print("*****");
        Serial.print("LAT=");

```

```

Serial.print(flat == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : flat, 6); //위도
Serial.print(" LON=");
Serial.print(flon == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : flon, 6); //경도
Serial.print(" SPEED=");
Serial.print(fspeed); //속도
Serial.print(" Course=");
Serial.print(fcourse); //방위
Serial.println(" *****");
}
// 트레드밀 디스플레이 정보값 OLED LCD에 출력
OzOled.printString("Time  :", 0, 0);
if(minute < 10){
    OzOled.printNumber(0.0, 0, 8, 0);
    OzOled.printNumber(minute, 9, 0);
}
else{
    OzOled.printNumber(minute, 8, 0);
}
OzOled.printString(" : ", 10, 0);
if(second < 10){
    OzOled.printNumber(0.0, 0, 13, 0);
    OzOled.printNumber(second, 14, 0);
}
else{
    OzOled.printNumber(second, 13, 0);
}
OzOled.printString("Steps :", 0, 2);
OzOled.printNumber(steps, 8, 2);
OzOled.printString("Speed :", 0, 4);
OzOled.printNumber(fspeed, 2, 8, 4);
OzOled.printString("km/h", 12, 4);
OzOled.printString("Cal  :", 0, 6);
OzOled.printNumber(cal, 2, 8, 6);
}
//HTTP 요청 함수
void httpRequest() {
    client.stop();
    //클라이언트가 서버에 연결되어있다면
    if(client.connect(server, 80)){
        Serial.println("connecting...");
        client.print(p);
    }
}

```

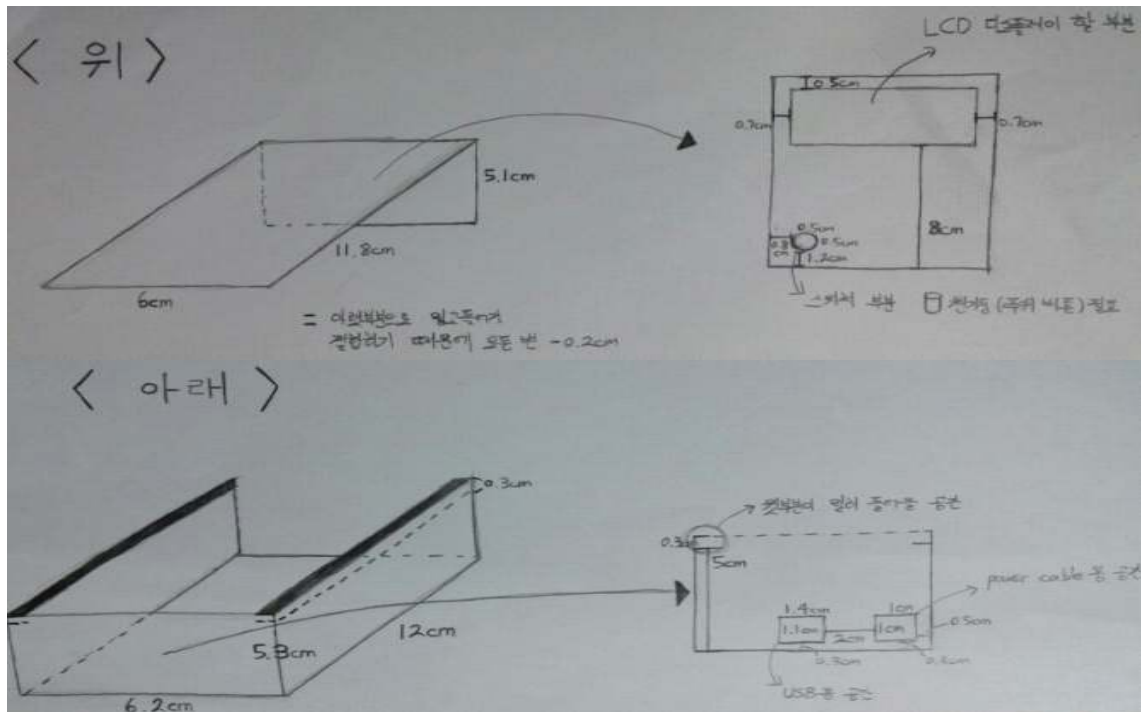
```

    client.print(flat, 6);
    client.print(p2);
    client.print(flon, 6);
    client.print(p3);
    client.print(fcourse);
    client.println(p4);
    client.println("Host: kic0326.dothome.co.kr");
    client.println("User-Agent: ArduinoWiFi/1.1");
    client.println("Connection: close");
    client.println();
}
}
// 타이머의 callback(), 1초마다 실행
void callback()
{
    cnt++;
    // 칼로리 계산식
    if(fspeed > 0){
        if(fspeed <= 130.0)
            cal = 0.00175 * (( 0.1 * fspeed + 3.5) / 3.5) * 60 * second2;
        else if(fspeed > 130.0)
            cal = 0.00175 * (( 0.2 * fspeed + 3.5) / 3.5) * 60 * second2;
    }
    else{
        fspeed = fspeed;
    }
    if(cnt % 1 == 0){
        second++;
        second2++;
    }
    if(cnt % 10 == 0 && cnt > 0){
        httpRequest();
    }
    if(second % 60 == 0){// 60초일 경우 0으로 초기화
        second = 0;
    }
    if(cnt % 60 == 0){
        minute++;
    }
    if(minute % 60 == 0){// 60분일 경우 0으로 초기화
        minute = 0;
    }
}

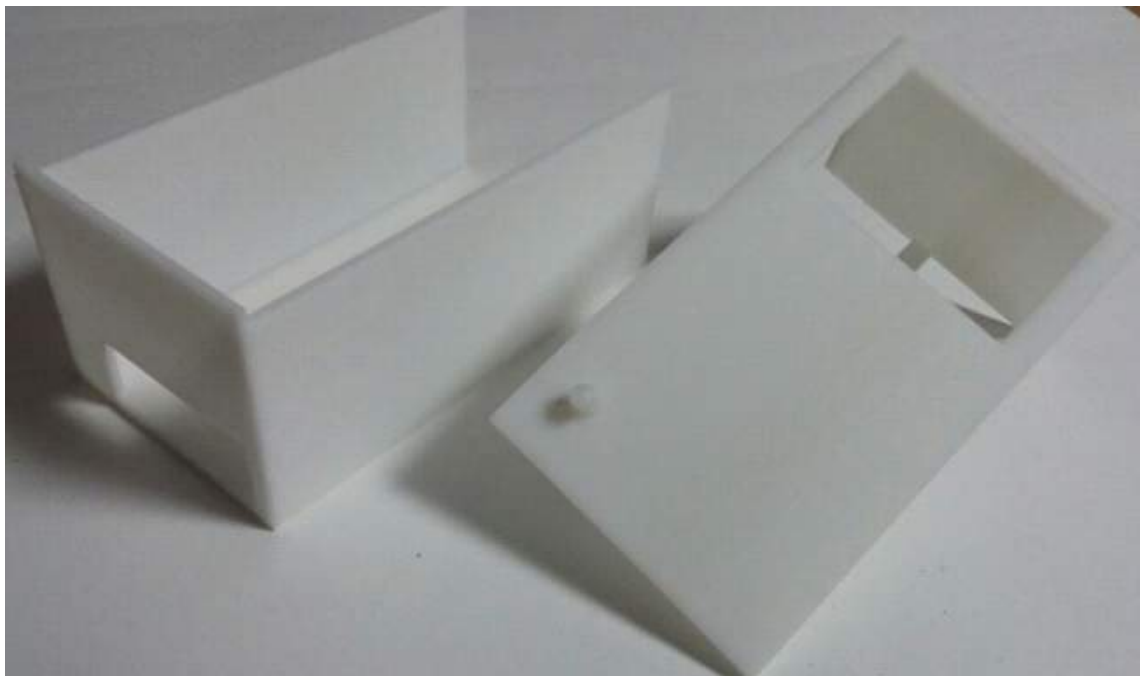
```

}
}

- 도화지에 외관 디자인 후 솔리드웍이란 소프트웨어로 모델링 하였다.



- 시그마센터에 가공을 의뢰하여 외관 제작을 완료하였다.



- 웹 애플리케이션 기능을 추가 하였다.

1) DB에 저장된 위도와 경도 값들을 한 번에 마커로 표시함으로써 사용자가 있었던 위치 정보를 줄 수 있다.

2) GPS월드에서 course값을 받아 DB에 넘겨주어 마커를 클릭했을 때 출력함으로 사용자가 움직이던 방위를 알 수 있다(+ 순서 포함).

record.php (기록된 위치 지도)

```
<?php include "header.php" ?>

<?php
    if(!$_SESSION['login_id']):
        header('location:/index.html?err=1');
    ?>

<?php else:
    ?>

<?php endif; ?>

<?php
    $connect = mysql_connect("localhost", "DB아이디", "DB비밀번호");
    $db_con = mysql_select_db("DB아이디", $connect);
    $sql = "select * from gps";
    $result = mysql_query($sql, $connect);

    $size = 0;           // 크기
    $latitude = array(); // gps테이블에 있는 위도 값들을 넣을 php배열
    $longitude = array(); // gps테이블에 있는 경도 값들을 넣을 php배열
    $course = array();   // gps테이블에 있는 코스 값들을 넣을 php배열
    ?>

<?php
while($row = mysql_fetch_array($result)){
    $latitude[$size] = $row['latitude']; // 위도 값 저장
    $longitude[$size] = $row['longitude']; // 경도 값 저장
    $course[$size] = $row['course'];      // 코스 값 저장
    $size++;                             // 크기
}
}
```



```

?>

<!DOCTYPE html>
<html lang="ko">
  <head>
    <style>
      #map{
        margin:15px auto;
      }
    </style>
    <meta charset="UTF-8">
    <title>Recorded information</title>
  </head>
  <body>
    <div id="map" style="width:90%;height:1000px;"></div>
    <script type="text/javascript" src="//apis.daum.net/maps/maps3.js?apikey=82efa9caba9af3640b436747196453bc"></script>
    <script language="JavaScript">
      var latitude = <?php echo json_encode($latitude); ?>;
      // php의 위도값 배열을 자바스크립트 배열에 저장
      var longitude = <?php echo json_encode($longitude); ?>;
      // php의 경도값 배열을 자바스크립트 배열에 저장
      var course = <?php echo json_encode($course); ?>;
      // php의 코스값 배열을 자바스크립트 배열에 저장
      var size = <?=$size?>;// 크기 저장
      for(var i=0; i<size; i++){
        console.log(latitude[i]);
        console.log(longitude[i]);
        console.log(course[i]);
      }
      window.onload = function() {

        // 지도의 중심
        var position = new daum.maps.LatLng(latitude[size-1], longitude[size-1]);

        // 기본 지도 표시
        var map = new daum.maps.Map(document.getElementById('map'), {
          center: position,
          level: 2,
          mapTypeId: daum.maps.MapTypeId.ROADMAP
        });
        // 다중 마커와 인포윈도우 표시

```

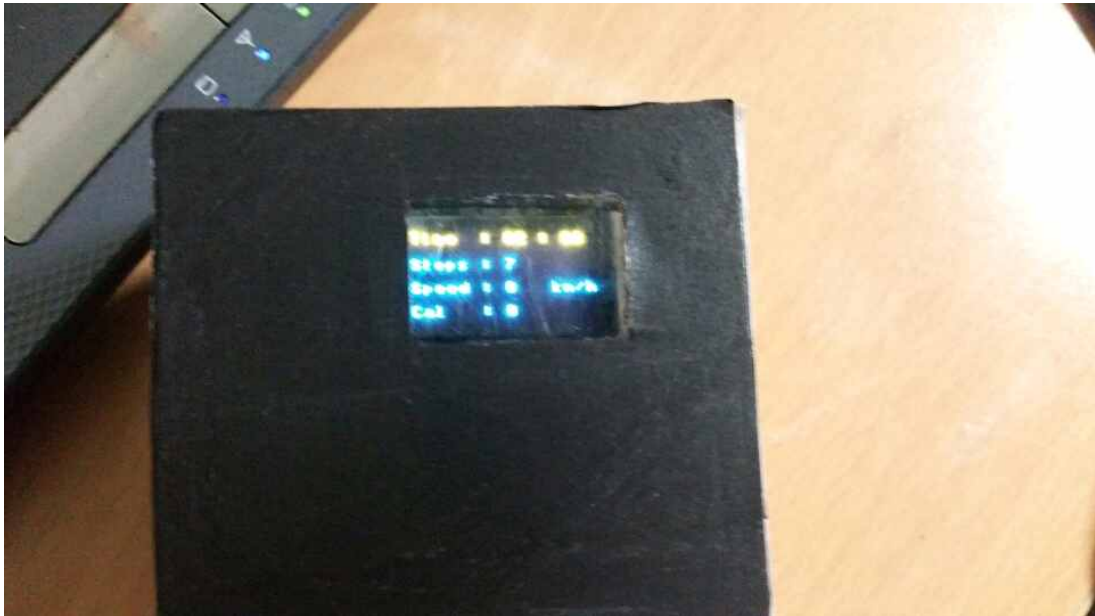

- 왼쪽부터 로그인 화면, 기계의 현재 위치, 저장된 위치와 방위를 보여준다.



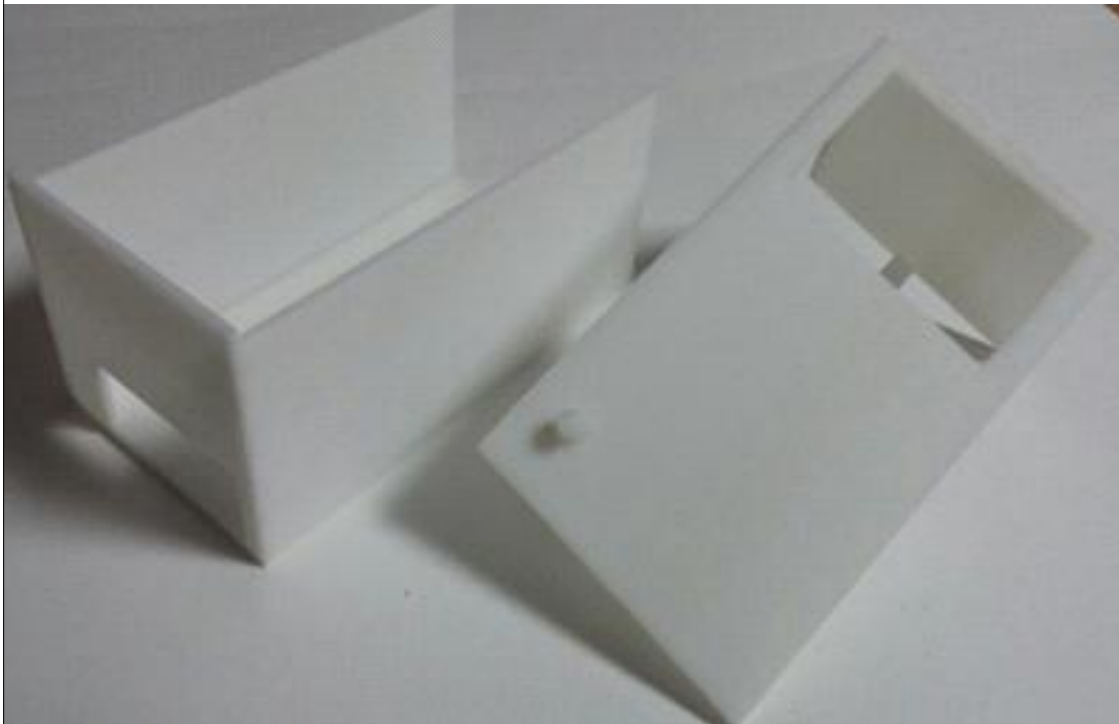
7. 작품 결과

작품명 : 스마트 만보기

1. 하드웨어 : 위 부터작동 시간, 걸음 수, 속도, 칼로리 디스플레이



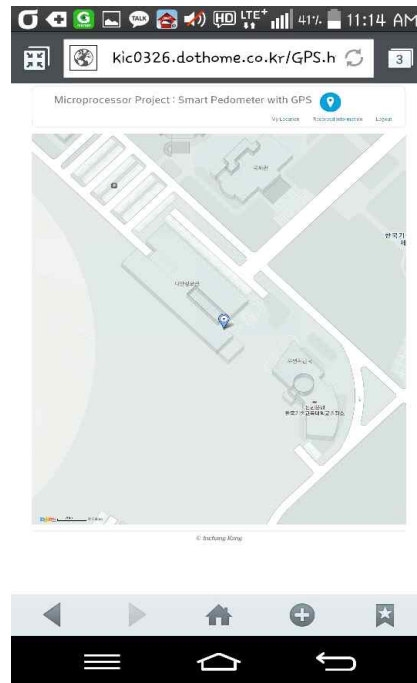
2. 기구부



1. 웹 애플리케이션 UI 및 기능(모바일 화면)



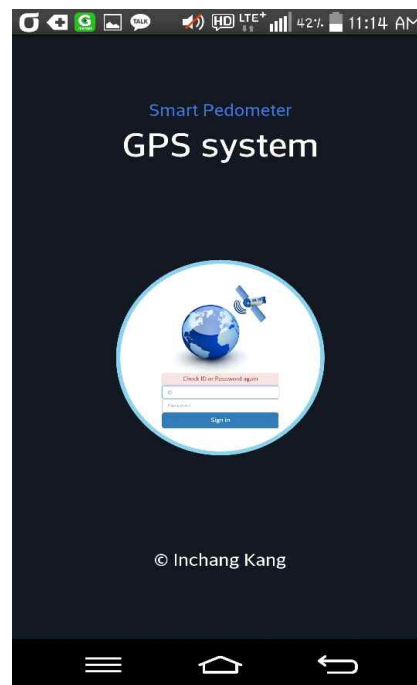
1. Login



2. My Location



3. Recorded information



4. Security & Logout

1. Login : 입력된 사용자 정보가 DB에 있는 아이디와 비밀번호랑 맞는지 체크한다.
2. My Location : 현재 위치를 나타내주는 지도로 DB에 저장되어 있는 최신 위도, 경도 값을 지도에 마커로 표시한다.
3. Recorded Location : 사용자가 방문했던 위치와 사용자가 진행한 이동 방향을 나타내주는 지도로 DB에 있는 모든 위도와 경도, 방위값을 지도에 마커로 표시한다. 그리고 마커 클릭시 순서와 방위값을 띄워준다.
4. Security & Logout : 입력된 사용자 정보가 DB에 있는 아이디와 비밀번호랑 일치하지 않을 경우 오류 메시지를 띄워주며 Logout섹션을 클릭할 경우 홈페이지를 빠져나온다. 이용하는 세션이 다를 경우 재로그인하여야 한다.

8. 진행 과정 영상 자료

2주차) 6디짓 7세그먼트를 사용한 자동 숫자 카운터

- <https://youtu.be/-a4NhTsUG30>

2주차) 6디짓 7세그먼트를 사용한 버튼 클릭 숫자 카운터

- <https://youtu.be/XbomhwUJSXI>

5주차) GPS쉴드를 사용한 시리얼 모니터에 GPS정보 출력

- <https://youtu.be/JYFZ9fvcsdc>

5주차) GPS쉴드 + 6디짓 7세그먼트 테스트

- https://youtu.be/4iM_GxX3rmE

5주차) GPS쉴드 + 6디짓 7세그먼트 + 버튼 테스트

- https://youtu.be/jHNI_qPpDCE

6주차) GPS쉴드 + 6디짓 7세그먼트 + 진동센서 테스트

- <https://youtu.be/1GNuhFUPGxw>

6주차) WiFi쉴드 + GPS쉴드 + 6디짓 7세그먼트 + 진동센서 테스트

- <https://youtu.be/9E5iOXAKes0>

7주차) WiFi로 GPS값 웹앱 지도에 출력

- <https://youtu.be/9sJewkJBoIM>

7주차) 초기 스마트 만보기 기능 완성

- <https://youtu.be/v0zuqCOZOZQ>

8주차 ~) 기울기 센서 디바운싱 실험

- <https://youtu.be/VRtXZ1Prork>

8주차 ~) WiFi 통한 아두이노와 웹앱 연동, GPS정보 지도에 마커로 표시

- <https://youtu.be/usKtkEz6K38>

9. 참고 자료 모음

1) HTTP Request

- WiFiWebClient 코드- <https://www.arduino.cc/en/Tutorial/WiFiWebClient>
- HttpClient 코드 - <https://www.arduino.cc/en/Tutorial/HttpClient>
- Http통신 이란?
 - <http://takeiteazy.tistory.com/298>
 - <http://najuung.tistory.com/entry/HTTP%ED%94%84%EB%A1%9C%ED%86%A0%EC%BD%9C%EC%9D%B4%EB%9E%80>
- GET방식과 POST방식
 - <http://soul0.tistory.com/185>
 - <http://najuung.tistory.com/entry/JSP-POST%EC%99%80-GET%EC%9D%98-%EC%B0%A8%EC%9D%B4%EC%A0%90>
 - http://www.w3schools.com/tags/ref_httpmethods.asp

2) millis()함수 사용

- <https://www.baldengineer.com/millis-tutorial.html>

3) 진동센서 사용

- http://blog.daum.net/_blog/BlogTypeView.do?blogid=0ZPt7&artidno=1006&_bloghome_menu=recenttext

4) NEO-6M GPS module

- 라이브러리 링크 제공 및 예제 코드 설명
 - <http://www.hardcopyworld.com/ngine/arduino/index.php/archives/755>
 - <http://bbangpan.tistory.com/42>
 - <http://www.openmakerlab.co.kr/#!Arduino-Intermediated-GPS-%ED%99%9C%EC%9A%A9%ED%95%98%EA%B8%B0/c2413/55cae8440cf2b432d2491328>
- 결선 방법 및 간단한 예제 코드
 - <http://thefortuneplanet.com/interfacing-arduino-uno-with-gps-module/>
 - <http://jules.dourlens.com/gps-ublox-neo-6m-and-arduino/>

5) CC3000 WiFi shield

- 라이브러리 링크 제공 및 결선방법, 예제 코드 설명
 - <http://bbangpan.tistory.com/20>
- 라이브러리 링크 제공 및 결선방법, 예제 설명 및 HTTP통신 예제소스 제공
 - <http://www.hardcopyworld.com/ngine/arduino/index.php/archives/1958>

6) 디바운싱

- <http://studymake.tistory.com/166>