

Report



경성대학교

과목명 : 임베디드시스템실습

학과 : 컴퓨터공학과

학번 : 2016642045

이름 : 강건우

제출일자 : 2018. 12. 13.

○ 프로젝트의 목표

- 처음 계획과 목표는 스마트폰(와이파이)을 이용하여 불을 켜다 끄고 커튼을 치고 여는 기능을 구상하였습니다. 그러나 아무리 실험을 해도 모터가 돌지 않아 다른 계획과 목표를 가지고 프로젝트에 만전을 기하여 최선을 다했습니다. 모션센서와 LED, 와이파이를 이용하여 스마트 방법 현관등을 만들겠다는 계획과 목표를 세웠습니다.

○ 스마트 방법 현관등의 필요성

- 1인 가구가 늘어나는 추세로 사건사고가 많아 방법과 안전에 중점을 두게 되었습니다. 먼저 저희 12조의 스마트 방법 현관등은 언뜻 보기에 시중에 비슷한 제품이 있어 보이지만 저희가 만든 스마트 방법 현관등은 시중에 없는 제품입니다. 스마트 도어락 같은 제품은 많이 사용하고 있습니다. 그래서 저희는 침입자가 도어락(1차 보안)을 뚫고 걸리지 않은 상태에서 안심하고 있을 사이 현관등이 켜지는 순간 집 주인(사용자)에게 메시지(알림)가 스마트폰으로 전송되는 2차 보안(이중 보안)인 스마트 방법 현관등의 필요성을 절실하게 느꼈습니다.

○ 역할분담

- 하드웨어(회로) 설계 및 제작, 환경 구축, 보조 프로그래밍 담당

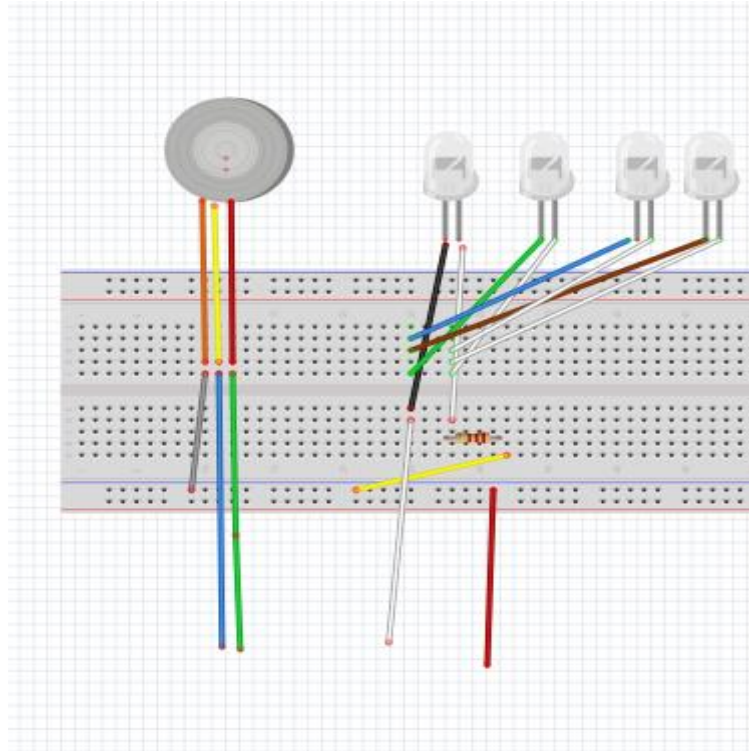
○ 일정계획

- 모터가 돌지 않아 일정계획을 급히 수정하여 진행했습니다. 그렇기 때문에 11일부터 계획을 바꾸어 진행을 하였습니다. 첫번째 계획은 발광 다이오드 작동 테스트, 두번째 계획은 모션 센서 작동 테스트, 세번째 계획은 아틱 클라우드와 아틱과의 연동 테스트, 마지막으로 와이파이 연결 및 정상적인 통신 확인을 계획하였습니다.

○ 설계 및 구현

- 설계

- Artik 053과 브레드보드, 발광 다이오드, 모션 센서, 저항(220Ω)으로 회로도 설계

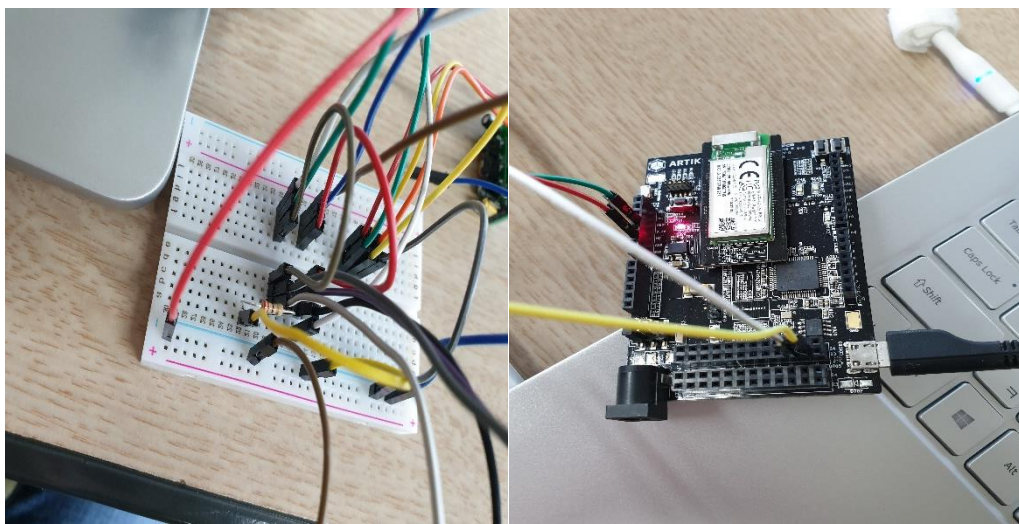


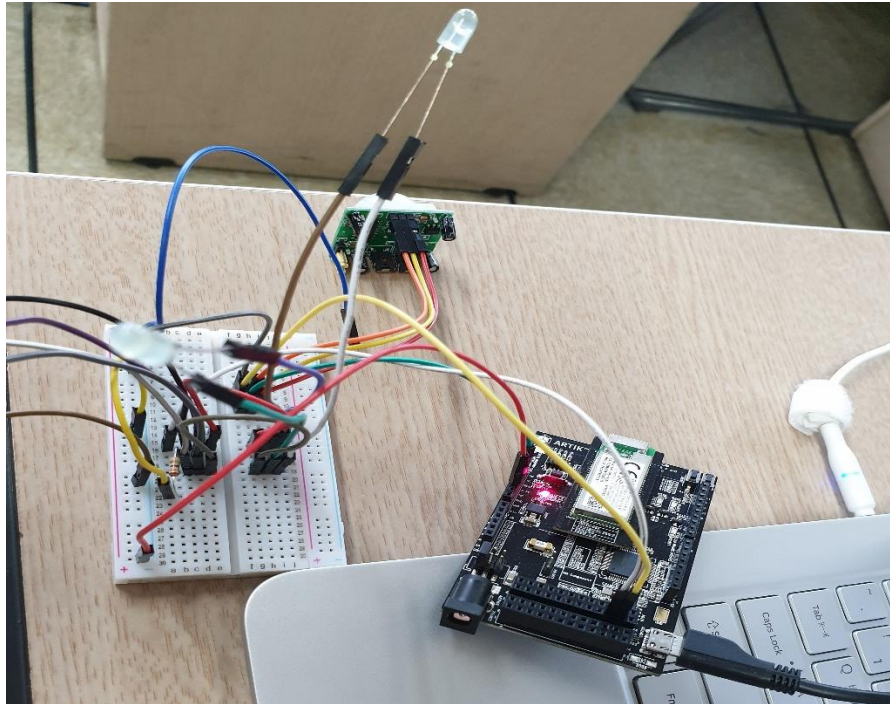
- Artik IDE로 구체적인 동작 코드 설계

- Artik 053과 WiFi(Internet)를 통해 Artik Cloud 연동 방법 설계

- 구현

- Artik 053과 브레드보드, 발광 다이오드, 모션 센서, 저항(220Ω)으로 회로도 구현





- Artik IDE로 코드 구현

- 주요코드(hello_main.c)_1

```
hello_main.c  wifi.h

#include <tinyara/config.h>
#include <stdlib.h>
#include <stdio.h>
#include <ctype.h>
#include "wifi.h"
#include "keys.h"
#include <tinyara/gpio.h>
#include <tinyara/pwm.h>
#include <tinyara/analog/adc.h>
#include <tinyara/analog/ioc1.h>
#include <apps/netutils/dhcpc.h>
#include <apps/netutils/mqtt_api.h>
#include <apps/netutils/ntpclient.h>
#include <time.h>
#include <string.h>
// NTP
#define NTP_REFRESH_PERIOD (60 * 60) /* seconds */
#define ARRAY_SIZE(a) (sizeof(a)/sizeof((a)[0]))

#define RED_ON_BOARD_LED 45
#define ADC_MAX_SAMPLES 4
#define NET_DEVNAME "wl1"
#define DEFAULT_CLIENT_ID "123456789"
#define SERVER_ADDR "api.artik.cloud" /* "52.200.124.224" */
#define DEVICE_ID "ba985cc9568b4ea987e0a806a84020ef" //FIX
#define DEVICE_TOKEN "63f09985f77c466e8222ed90474a3b9f" // FIX
#define ACTION_TOPIC "/v1.1/actions/ba985cc9568b4ea987e0a806a84020ef" // FIX
#define MESSAGE_TOPIC "/v1.1/messages/ba985cc9568b4ea987e0a806a84020ef" //FIX

#define GPIO_FUNC_SHIFT 13
#define GPIO_INPUT (0x0 << GPIO_FUNC_SHIFT)
#define GPIO_OUTPUT (0x1 << GPIO_FUNC_SHIFT)

#define GPIO_PORT_SHIFT 3
#define GPIO_PORTG1 (0x5 << GPIO_PORT_SHIFT)

#define GPIO_PIN_SHIFT 0
#define GPIO_PIN4 (0x4 << GPIO_PIN_SHIFT)
#define GPIO_PIN2 (0x2 << GPIO_PIN_SHIFT)
```

- 주요코드(hello_main.c)_2

```
hello_main.c wifi.h

#define GPIO_FUNC_SHIFT 13
#define GPIO_INPUT (0x0 << GPIO_FUNC_SHIFT)
#define GPIO_OUTPUT (0x1 << GPIO_FUNC_SHIFT)

#define GPIO_PORT_SHIFT 3
#define GPIO_PORTG1 (0x5 << GPIO_PORT_SHIFT)

#define GPIO_PIN_SHIFT 0
#define GPIO_PIN4 (0x4 << GPIO_PIN_SHIFT)
#define GPIO_PIN2 (0x2 << GPIO_PIN_SHIFT)

#define GPIO_PUPD_SHIFT 11
#define GPIO_PULLDOWN (0x1 << GPIO_PUPD_SHIFT)
#define GPIO_PULLUP (0x3 << GPIO_PUPD_SHIFT)

#define GPIO_FUNC_SHIFT 13
#define GPIO_INPUT (0x0 << GPIO_FUNC_SHIFT)
#define GPIO_OUTPUT (0x1 << GPIO_FUNC_SHIFT)

#define GPIO_PORT_SHIFT 3
#define GPIO_PORTG2 (0x6 << GPIO_PORT_SHIFT)

#define GPIO_PIN_SHIFT 0
#define GPIO_PIN6 (0x6 << GPIO_PIN_SHIFT)
#define GPIO_PIN7 (0x7 << GPIO_PIN_SHIFT)

#define GPIO_PUPD_SHIFT 11
#define GPIO_PULLDOWN (0x1 << GPIO_PUPD_SHIFT)
#define GPIO_PULLUP (0x3 << GPIO_PUPD_SHIFT)

void blink_led(int repeat, int duration);
uint32_t pir_signal;
uint32_t led;
```

- 주요코드(hello_main.c)_3

```
hello_main.c wifi.h

void blink_led(int repeat, int duration) {
    int i;

    for (i = 0; i < repeat; i++) {
        s5j_gpiowrite(led, 1);
        up_mdelay(duration);
        s5j_gpiowrite(led, 0);
        up_mdelay(duration);
    }
}
```

- 주요코드(hello_main.c)_4

```
hello_main.c wifi.h
int main(int argc, char *argv[])
{
    bool wifiConnected = false;
    gpio_write(RED_ON_BOARD_LED); // Turn on on board Red LED to indicate no WiFi connection is established
    struct timespec currentTime;
    struct tm tm_now;
    char buf[32];
    size_t sz_buf = sizeof(buf);
    int pirState = 0;

    pir_signal = GPIO_INPUT | GPIO_PORTG2 | GPIO_PIN6 | GPIO_PULLDOWN;
    led = GPIO_OUTPUT | GPIO_PORTG2 | GPIO_PIN7 | GPIO_PULLDOWN;

    s5j_configgpio(pir_signal);
    s5j_configgpio(led);

    printf("Waiting for Motion Sensor Initialized.\n");
    up_mdelay(2000);

    printf("Detecting Motion...\n");
    blink_led(2, 200);
```

- 주요코드(hello_main.c)_5

```
hello_main.c wifi.h

printf("Connect to Wi-Fi success\n");

bool mqttConnected = false;
bool ipObtained = false;
printf("Get IP address\n");

struct dhcpc_state state;
void *dhcp_handle;

while(!ipObtained)
{
    dhcp_handle = dhcpc_open(NET_DEVNAME);
    ret = dhcpc_request(dhcp_handle, &state);
    dhcpc_close(dhcp_handle);

    if (ret != OK)
    {
        printf("Failed to get IP address\n");
        printf("Try again\n");
        sleep(1);
    }
    else
    {
        ipObtained = true;
    }
}
netlib_set_ipv4addr(NET_DEVNAME, &state.ipaddr);
netlib_set_ipv4netmask(NET_DEVNAME, &state.netmask);
netlib_set_dripv4addr(NET_DEVNAME, &state.default_router);

printf("IP address %s\n", inet_ntoa(state.ipaddr));
```

- 주요코드(hello_main.c)_6

```
hello_main.c x wifi.h

strcat(str, "\"GPIO\":{");
int j;
for ( j = 37; j <= 55; j++)
{
    strcat(str, "\"gpio");
    char buf[10];
    sprintf(buf, "%d\":"%d", j, gpio_read(j));
    strcat(str, buf);
    if (j != 55)
        strcat(str, ",");
}
strcat(str, "}}");
if (mqtt_publish(pClientHandle, MESSAGE_TOPIC, str, sizeof(str), 0, 0) != 0)
{
    printf("Error: mqtt_publish() failed.\n");
}
else
{
    printf("Published %s\n", str);
}
while(1){
    if (s5j_gpioread(pir_signal) == 1) {
        s5j_gpiowrite(led, 1);
        if (pirState == 0) {
            clock_gettime(CLOCK_REALTIME, &currentTime);
            localtime_r((time_t*)&currentTime.tv_sec, &tm_now);
            strftime(buf, sz_buf, "%Y/%m/%d %H:%M:%S", &tm_now);
            printf("Motion Started: %s\n", buf);
            pirState = 1;
        }
        else {
            s5j_gpiowrite(led, 0);
            if (pirState == 1) {
                printf("Motion Ended!!\n");
                pirState = 0;
            }
        }
    }
}
```


- 주요코드(wifi.h)

```
hello_main.c  keys.h  wifi.h  ✕
#include <net/if.h>
#include <apps/netutils/wifi/slsi_wifi_api.h>

// WiFi
#define STATE_DISCONNECTED 0
#define STATE_CONNECTED 1

#define SLSI_WIFI_SECURITY_OPEN "open"
#define SLSI_WIFI_SECURITY_WPA2_AES "wpa2_aes"

#define SSID "Ks323" //FIX
#define PSK "kscse323" //FIX

slsi_security_config_t *getSecurityConfig(char *sec_type,
char *psk, WiFi_InterFace_ID_t mode);

static int g_connection_state = STATE_DISCONNECTED;
static uint8_t g_join_result = 0;

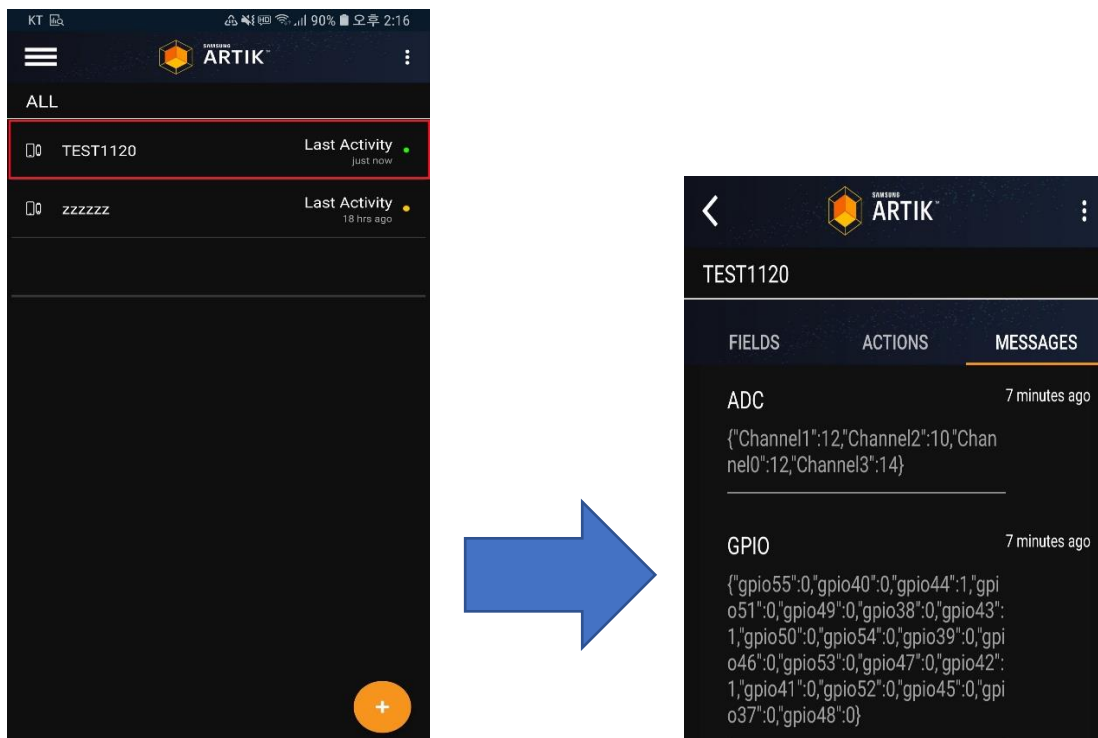
static sem_t g_sem_join;

/**
 * Handler for network link up connection event
 *
 * Sets the global connection state variable and the
 * result of the network join request.
 */
void networkLinkUpHandler(slsi_reason_t* reason) {
    g_connection_state = STATE_CONNECTED;

    g_join_result = reason->reason_code;
    sem_post(&g_sem_join);
}
```

- Artik 053과 WiFi(Internet)를 통해 Artik Cloud 정상적인 통신 구현

- Samsung ARTIK APP으로 정상적인 통신확인 가능(해당 Device 클릭 -> MESSAGES)

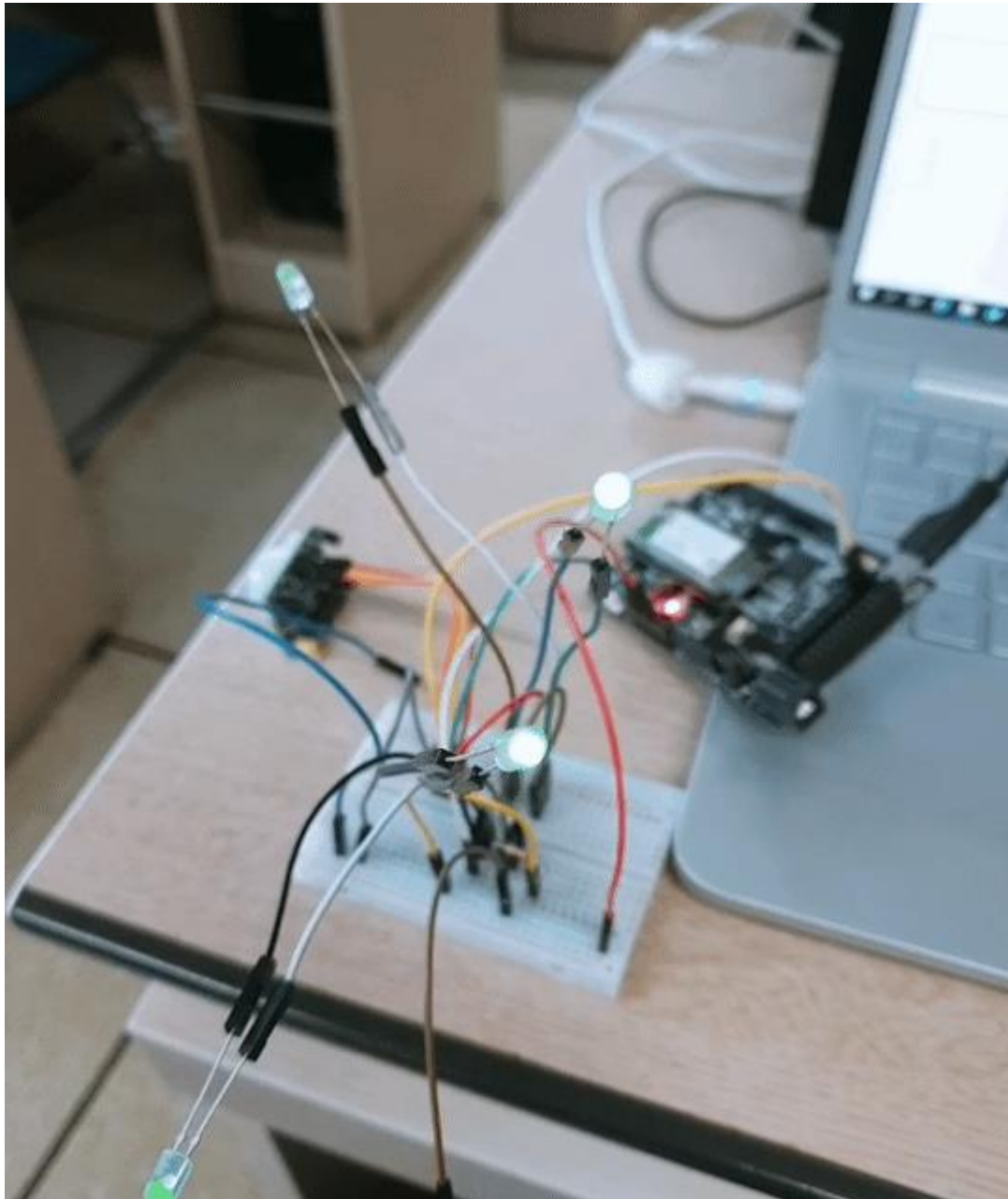


○ 시험 및 평가

- 시험

[미완성품 시험.mp4](#)

[완성품 시험.mp4](#)





- 평가, 느낀점

- 계획대비 목표 달성도 100%

- 정상적으로 작동도 잘 되고 소비자의 입장에서 보았을 때 사고 싶은 제품입니다.

- 계획과 목표를 세웠던 대로 결과물이 나와서 정말 뿌듯했습니다. 시행착오도 많았고 발표 하루 전까지 모션센서 작동이 됐다 안됐다 해서 걱정도 많았습니다. 그렇지만 장동현 학생과 끝까지 최선을 다해 성공적으로 프로젝트를 마쳤다는 것에 매우 만족합니다. 다시 한번 조원의 중요성을 알게 되었고 같이 하면서 모르는 부분을 많이 배웠습니다. 아쉬웠던 점은 와이파이기가 연결이 되어서 모션 센서로 감지를 하여 LED가 켜질 때 어플로 알림이나 메시지를 띄우는 것을 구현하고 싶었으나 삼성 아틱 어플을 수정하지 않은 한 불가능해서 너무 아쉬웠습니다.

○ 참고자료

- IoT(아틱053)_1.pptx

- IoT(아틱053)_2.pptx

- IoT(아틱053)_3.pptx

- IoT(아틱053)_4.pptx