**과제 1. 광센서**

1. 개요

밝기센서를 이용하여 밝기를 3단계로 calibration하고

컬러센서를 이용해 색깔에 따라 모터의 속도가 달라지도록 동작시킨다.

2. 프로그램 구조 설명

2.1 함수에 대한 설명

glcd\_init();

> lcd를 초기화 하는 함수입니다.

motor\_init();

>모터를 초기화 하는 함수입니다.

ev3\_sensor\_init(0, COL\_COLOR);

>광센서를 초기화 하는 함수입니다.

calibEV3Sensor(0, MAX\_LIGHT\_LEVEL, light\_value);

>헤더파일에 정의된 레벨 구간으로 calibration 합니다.

get\_level(ev3\_sensor\_get(0), MAX\_LIGHT\_LEVEL, light\_value)

>calibration한 값을 기준으로, 포트에서 받아온 값의 레벨을 판단합니다.

glcd\_printf()

>포트에서 받아온 값을 출력하기 위한 함수입니다.

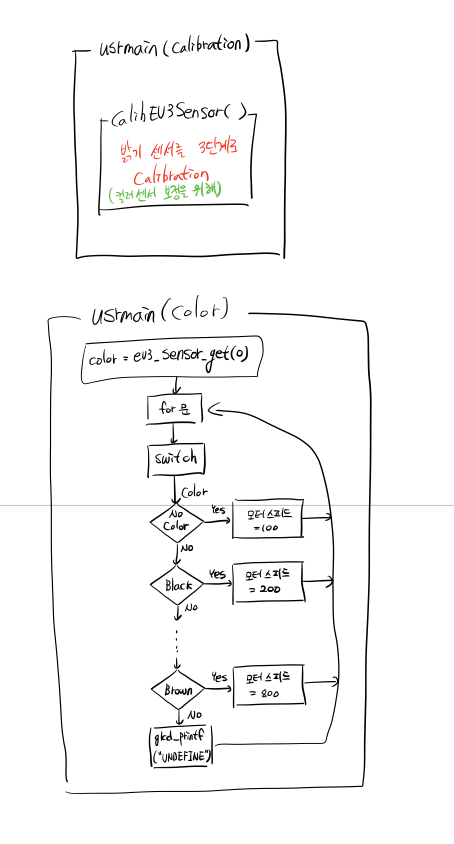
motor\_set(0,250);

>모터의 속도를 제어하는 함수입니다.

task\_sleep(50);

>sencing 주기를 설정하는 함수입니다.

2.2 다이어그램



3. 실행 결과

동영상 결과를 같이 첨부합니다.

**(과제1 광센서 폴더의**

**usrmain(calibration) 실행영상 => 밝기센서 calibration 후 lcd에 표시하는 프로그램**

**usrmain(color) 실행영상 => 컬러센서로 색깔 구분 후 모터 속도 제어 프로그램 )**

**(실행 결과는 자신의 컴퓨터환경에서 실행한 결과 캡쳐 및 보드 동작 사진을 문서에 첨부해서 제출합니다. 실행결과가 성공한 경우 뿐만 아니라 실패한 경우도 포함하시기 바랍니다.**

**올린 코드를 실행한 결과와 사진으로 첨부한 결과가 다를 경우 감점합니다.)**

4. 고찰 (자세히 쓸 것)

이번 과제에서는 코딩의 난이도는 그렇게 높지 않았으나 실수가 많이 나와

헤맸었습니다. 라이브러리 네임에 포함된 언더바(\_)가 한 개 인줄 알았으나 사실은 2개여서 헤

맨적도 있고, calibEV3Sensor를 써야하는데 calibSensor를 쓰고 있었다던가 하는 실수로

시간을 많이 보냈습니다. 그리고 의외로 Lcd의 그 전 출력값이 다음 출력값을 출력할 때도 남

아 있는 문제로 꽤나 고전했었습니다. 역시 프로그래밍은 항상 의외의 곳에서 고전을 면치

못하는 것 같습니다.

이번 프로그래밍에서는 calibration이라는 저에게는 다소 익숙하지 않은 개념을 이해하고

라이브러리 사용법을 익히는 좋은 계기가 되었습니다. 앞으로 라이브러리를 사용하는

일이 많을텐데 라이브러리 사용에 좀 더 익숙해지려 노력하겠습니다.

5. 프로그램 소스 파일 (주석을 꼭 달 것)

usrmain(calibration)과 usrmain(color)는 서로 다른 2개의 프로그램입니다.

파일이름을 usrmain으로 2개 만들수 없어서 괄호로 구분했습니다.

usrmain(calibration).c

/\* -------------------------------------------------------------------------

Include

------------------------------------------------------------------------- \*/

**//필요한 라이브러리들을 include 합니다.**

#include "../ubiconfig.h"

// standard c library include

#include <stdio.h>

#include <sam4e.h>

// ubinos library include

#include "itf\_ubinos/itf/bsp.h"

#include "itf\_ubinos/itf/ubinos.h"

#include "itf\_ubinos/itf/bsp\_fpu.h"

// chipset driver include

#include "ioport.h"

#include "pio/pio.h"

// new estk driver include

#include "lib\_new\_estk\_api/itf/new\_estk\_led.h"

#include "lib\_new\_estk\_api/itf/new\_estk\_glcd.h"

#include "lib\_switch/itf/lib\_switch.h"

#include "lib\_sensor/itf/lib\_sensor.h"

#include "lib\_EV3\_sensor/itf/lib\_EV3\_sensor.h"

#include "lib\_sensorcalib/itf/lib\_sensorcalib.h"

#include "lib\_motor\_driver/itf/lib\_motor\_driver.h"

// custom library header file include

//#include "../../lib\_default/itf/lib\_default.h"

#include "../../lib\_new/itf/lib\_new.h"

// user header file include

/\* -------------------------------------------------------------------------

Global variables

------------------------------------------------------------------------- \*/

/\* -------------------------------------------------------------------------

Prototypes

------------------------------------------------------------------------- \*/

static void rootfunc(void \* arg);

/\* -------------------------------------------------------------------------

Function Definitions

------------------------------------------------------------------------- \*/

int usrmain(int argc, char \* argv[]) {

int r;

printf("\n\n\n\r");

printf("================================================================================\n\r");

printf("exe\_ubinos\_test (build time: %s %s)\n\r", \_\_TIME\_\_, \_\_DATE\_\_);

printf("================================================================================\n\r");

r = task\_create(NULL, rootfunc, NULL, task\_getmiddlepriority(), 256, "root");

if (0 != r) {

logme("fail at task\_create\r\n");

}

ubik\_comp\_start();

return 0;

}

static void rootfunc(void \* arg) {

**//lcd 초기화**

glcd\_init();

**//모터 초기화**

motor\_init();

**//밝기센서 값 초기화**

ev3\_sensor\_init(0, COL\_REFLECT);

**//컬러센서를 보정하기 위한 밝기 센서 값을 먼저 calibration해야합니다.**

calibEV3Sensor(0, MAX\_LIGHT\_LEVEL, light\_value);

**//lcd clear**

glcd\_clear();

**//첫줄부터 둘째줄까지 DARK값 표시**

glcdGotoChar(0, 1);

glcd\_printf("STATE[LIGHT\_DARK] \nVAL : %d ", light\_value[0]);

**//셋째줄부터 넷째줄까지 DIM값 표시**

glcdGotoChar(0, 3);

glcd\_printf("STATE[LIGHT\_DIM] \nVAL : %d ", light\_value[1]);

**//다섯줄부터 여섯줄까지 BRIGHT값 표시**

glcdGotoChar(0, 5);

glcd\_printf("STATE[LIGHT\_BRIGHT] \nVAL : %d ", light\_value[2]);

}

usrmain(color).c

/\* -------------------------------------------------------------------------

Include

------------------------------------------------------------------------- \*/

**//필요한 라이브러리들을 include 합니다.**

#include "../ubiconfig.h"

// standard c library include

#include <stdio.h>

#include <sam4e.h>

// ubinos library include

#include "itf\_ubinos/itf/bsp.h"

#include "itf\_ubinos/itf/ubinos.h"

#include "itf\_ubinos/itf/bsp\_fpu.h"

// chipset driver include

#include "ioport.h"

#include "pio/pio.h"

// new estk driver include

#include "lib\_new\_estk\_api/itf/new\_estk\_led.h"

#include "lib\_new\_estk\_api/itf/new\_estk\_glcd.h"

#include "lib\_switch/itf/lib\_switch.h"

#include "lib\_sensor/itf/lib\_sensor.h"

#include "lib\_EV3\_sensor/itf/lib\_EV3\_sensor.h"

#include "lib\_sensorcalib/itf/lib\_sensorcalib.h"

#include "lib\_motor\_driver/itf/lib\_motor\_driver.h"

// custom library header file include

//#include "../../lib\_default/itf/lib\_default.h"

#include "../../lib\_new/itf/lib\_new.h"

// user header file include

**// 색깔을 변수로 define합니다.**

#define NoColor 0

#define Black 1

#define Blue 2

#define Green 3

#define Yellow 4

#define Red 5

#define White 6

#define Brown 7

/\* -------------------------------------------------------------------------

Global variables

------------------------------------------------------------------------- \*/

/\* -------------------------------------------------------------------------

Prototypes

------------------------------------------------------------------------- \*/

static void rootfunc(void \* arg);

/\* -------------------------------------------------------------------------

Function Definitions

------------------------------------------------------------------------- \*/

int usrmain(int argc, char \* argv[]) {

int r;

printf("\n\n\n\r");

printf("================================================================================\n\r");

printf("exe\_ubinos\_test (build time: %s %s)\n\r", \_\_TIME\_\_, \_\_DATE\_\_);

printf("================================================================================\n\r");

r = task\_create(NULL, rootfunc, NULL, task\_getmiddlepriority(), 256, "root");

if (0 != r) {

logme("fail at task\_create\r\n");

}

ubik\_comp\_start();

return 0;

}

static void rootfunc(void \* arg) {

**//lcd 초기화**

glcd\_init();

**//모터 초기화**

motor\_init();

**//밝기센서 값 초기화**

ev3\_sensor\_init(0, COL\_COLOR);

**//lcd clear**

glcd\_clear();

int color;

for (;;)

{

glcdGotoChar(0, 0);

**/\***

**// 컬러센서로 들어오는 컬러 값을 color 변수에 저장하고**

**color 변수의 값에 따라 모터의 속도를 변화시킵니다.**

**\*/**

color = ev3\_sensor\_get(0);

switch (color)

{

case NoColor:

glcd\_printf("No color VAL : %3d ", ev3\_sensor\_get(0));

motor\_set(0, 100);

break;

case Black:

glcd\_printf("Black VAL : %3d ", ev3\_sensor\_get(0));

motor\_set(0, 200);

break;

case Blue:

glcd\_printf("Blue VAL : %3d ", ev3\_sensor\_get(0));

motor\_set(0, 300);

break;

case Green:

glcd\_printf("Green VAL : %3d ", ev3\_sensor\_get(0));

motor\_set(0, 400);

break;

case Yellow:

glcd\_printf("Yellow VAL : %3d ", ev3\_sensor\_get(0));

motor\_set(0, 500);

break;

case Red:

glcd\_printf("Red VAL : %3d ", ev3\_sensor\_get(0));

motor\_set(0, 600);

break;

case White:

glcd\_printf("White VAL : %3d ", ev3\_sensor\_get(0));

motor\_set(0, 700);

break;

case Brown:

glcd\_printf("Brown VAL : %3d ", ev3\_sensor\_get(0));

motor\_set(0, 800);

break;

default:

glcd\_printf("UNDEFINE ");

break;

}

task\_sleep(50);

}

}

6. 자료 출처 (어떤 책을 참고했으며 어떤 웹사이트를 참고했는지 출처를 적습니다)

교수님의 자료를 참고했습니다.

**밑에 더 있습니다.**

**과제 2. 초음파 센서**

1. 개요

초음파센서를 이용하여 거리에 따라 모터의 속도가 달라지도록 동작시킨다.

2. 프로그램 구조 설명

2.1 함수에 대한 설명

glcd\_init();

> lcd를 초기화 하는 함수입니다.

motor\_init();

>모터를 초기화 하는 함수입니다.

sensor\_init(NXT\_DIGITAL\_SENSOR\_SONA, 0, 0, 0);

>초음파 센서를 초기화 하는 함수입니다.

calibSensor(0, MAX\_DIS\_LEVEL, dis\_value);

>헤더파일에 정의된 레벨 구간으로 calibration 합니다.

get\_level(ev3\_sensor\_get(0), MAX\_LIGHT\_LEVEL, light\_value)

>calibration한 값을 기준으로, 포트에서 받아온 값의 레벨을 판단합니다.

glcd\_printf()

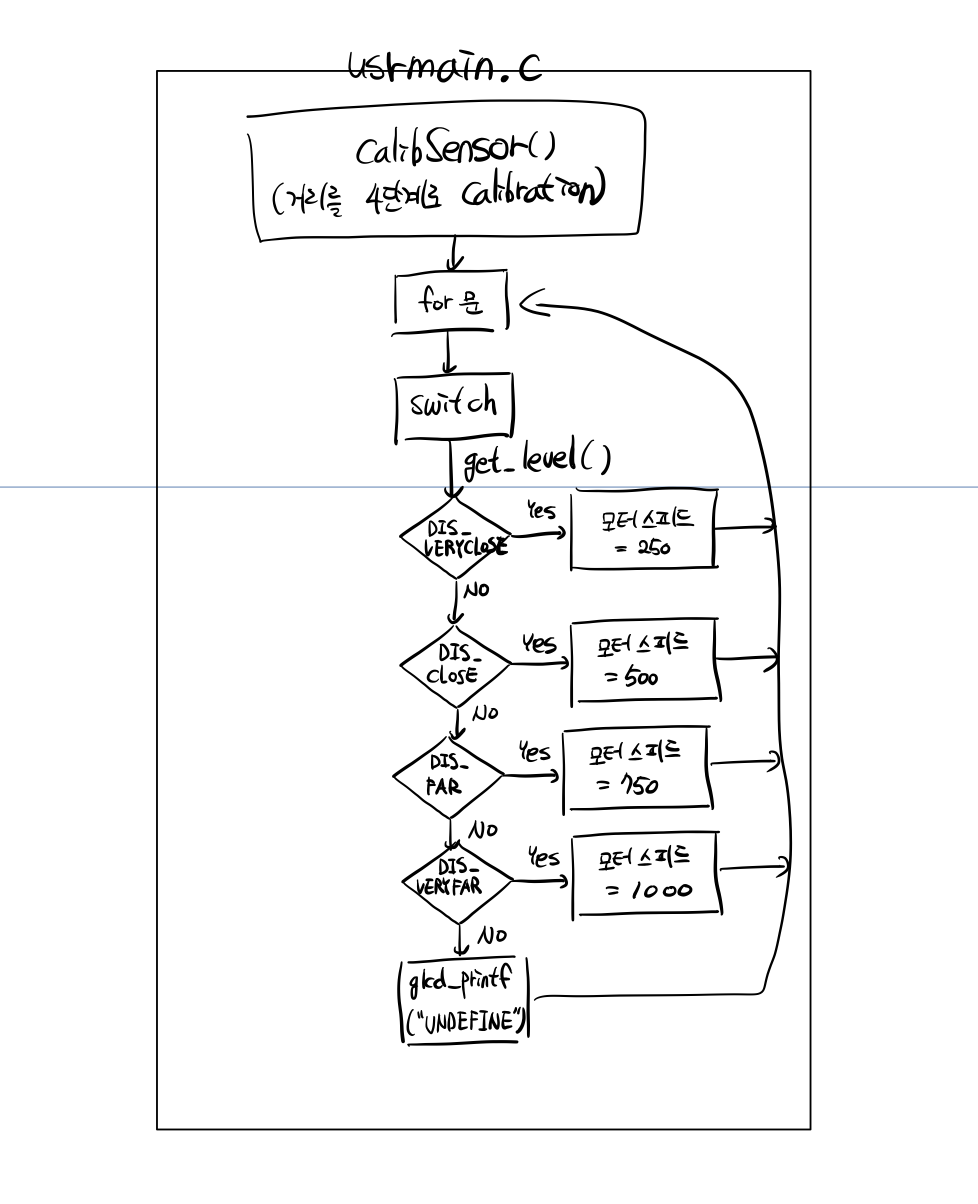
>포트에서 받아온 값을 출력하기 위한 함수입니다.

motor\_set(0,250);

>모터의 속도를 제어하는 함수입니다.

task\_sleep(30);

>sencing 주기를 30ms로 설정하는 함수입니다.

2.2다이어그램

3. 실행 결과

**동영상 결과를 같이 첨부합니다.**

**(과제2 초음파센서 폴더의 동영상)**



**(실행 결과는 자신의 컴퓨터환경에서 실행한 결과 캡쳐 및 보드 동작 사진을 문서에 첨부해서 제출합니다. 실행결과가 성공한 경우 뿐만 아니라 실패한 경우도 포함하시기 바랍니다.**

**올린 코드를 실행한 결과와 사진으로 첨부한 결과가 다를 경우 감점합니다.)**

4. 고찰 (자세히 쓸 것)

이번 과제는 초음파 센서를 테스트 해보는 과제였습니다. 헤더파일에 이미 정의되어 있는

Calibration level을 적용하는 것은 광센서에서 해보았기 때문에 이번에는 제가 따로

정의한 4단계의 level을 헤더파일에 추가로 등록하여 사용을 해보았습니다.

앞서 한 과제1, 광센서와 거의 같은 과제였기 때문에 큰 난관은 없었으나,

초음파 센서의 level calibration을 할 때, 거리가 멀어지면 멀어질수록 노이즈가 많아지는

현상 때문에 측정하는데 조금 어려움이 있었습니다.

이번 프로그래밍으로 c언어를 배울 때 많이 썼던 헤더파일의 적용에 대해서 복습을

하는 계기가 되었고, 초음파 센서도 다룰 수 있게 되었습니다. 앞으로도 하나하나

여러가지의 라이브러리를 적용해보며 저의 라이브러리 활용도를 높여 나갈 것입니다.

5. 프로그램 소스 파일 (주석을 꼭 달 것)

lib\_sensorcalib.h

(헤더파일에 제가 따로 정의한 level을 추가했습니다.)

#define MAX\_DISLEVEL 4

#define DIS\_VERYCLOSE 0

#define DIS\_CLOSE 1

#define DIS\_FAR 2

#define DIS\_VERYFAR 3

Static int dis\_value[MAX\_DISLEVEL] = {0,0,0,0};

usrmain.c

/\*

Copyright (C) 2009 Sung Ho Park

Contact: ubinos.org@gmail.com

This file is part of the exe\_helloworld component of the Ubinos.

GNU General Public License Usage

This file may be used under the terms of the GNU

General Public License version 3.0 as published by the Free Software

Foundation and appearing in the file license\_gpl3.txt included in the

packaging of this file. Please review the following information to

ensure the GNU General Public License version 3.0 requirements will be

met: http://www.gnu.org/copyleft/gpl.html.

GNU Lesser General Public License Usage

Alternatively, this file may be used under the terms of the GNU Lesser

General Public License version 2.1 as published by the Free Software

Foundation and appearing in the file license\_lgpl.txt included in the

packaging of this file. Please review the following information to

ensure the GNU Lesser General Public License version 2.1 requirements

will be met: http://www.gnu.org/licenses/old-licenses/lgpl-2.1.html.

Commercial Usage

Alternatively, licensees holding valid commercial licenses may

use this file in accordance with the commercial license agreement

provided with the software or, alternatively, in accordance with the

terms contained in a written agreement between you and rightful owner.

\*/

/\* -------------------------------------------------------------------------

Include

------------------------------------------------------------------------- \*/

**//필요한 라이브러리들을 include**

#include "../ubiconfig.h"

// standard c library include

#include <stdio.h>

#include <sam4e.h>

// ubinos library include

#include "itf\_ubinos/itf/bsp.h"

#include "itf\_ubinos/itf/ubinos.h"

#include "itf\_ubinos/itf/bsp\_fpu.h"

// chipset driver include

#include "ioport.h"

#include "pio/pio.h"

// new estk driver include

#include "lib\_new\_estk\_api/itf/new\_estk\_led.h"

#include "lib\_new\_estk\_api/itf/new\_estk\_glcd.h"

#include "lib\_switch/itf/lib\_switch.h"

#include "lib\_sensor/itf/lib\_sensor.h"

#include "lib\_EV3\_sensor/itf/lib\_EV3\_sensor.h"

#include "lib\_sensorcalib/itf/lib\_sensorcalib.h"

#include "lib\_motor\_driver/itf/lib\_motor\_driver.h"

// custom library header file include

//#include "../../lib\_default/itf/lib\_default.h"

#include "../../lib\_new/itf/lib\_new.h"

// user header file include

/\* -------------------------------------------------------------------------

Global variables

------------------------------------------------------------------------- \*/

/\* -------------------------------------------------------------------------

Prototypes

------------------------------------------------------------------------- \*/

static void rootfunc(void \* arg);

/\* -------------------------------------------------------------------------

Function Definitions

------------------------------------------------------------------------- \*/

int usrmain(int argc, char \* argv[]) {

int r;

printf("\n\n\n\r");

printf("================================================================================\n\r");

printf("exe\_ubinos\_test (build time: %s %s)\n\r", \_\_TIME\_\_, \_\_DATE\_\_);

printf("================================================================================\n\r");

r = task\_create(NULL, rootfunc, NULL, task\_getmiddlepriority(), 256, "root");

if (0 != r) {

logme("fail at task\_create\r\n");

}

ubik\_comp\_start();

return 0;

}

static void rootfunc(void \* arg) {

int dis;

**//lcd 초기화**

glcd\_init();

**//초음파센서 초기화**

sensor\_init(NXT\_DIGITAL\_SENSOR\_SONA, 0, 0, 0);

**//모터 초기화**

motor\_init();

**//distance를 4단계로 calibration**

calibSensor(0, MAX\_DIS\_LEVEL, dis\_value);

**/\*헤더파일에 DIS\_VERYCLOSE, DIS\_CLOSE, DIS\_FAR, DIS\_VERYFAR**

**4단계로 미리 정의**

**\*/**

glcd\_clear();

for (;;)

{

dis = sensor\_get(0); **// dis 변수에 0포트에서 들어오는 distance 값을 저장합니다**.

glcdGotoChar(0, 0);

/\*check sensor level \*/

**// dis변수를 바탕으로 level을 체크하여 level에 따라 출력값과 모터속도를 다르게 출력**

**//DIS\_VERYCLOSE,**

**//DIS\_CLOSE,**

**//DIS\_FAR**

**//DIS\_VERYFAR 4가지 종류로 거리를 구분합니다.**

switch (get\_level(dis, MAX\_DIS\_LEVEL, dis\_value))

{

case DIS\_VERYCLOSE:

glcd\_printf("very close VAL : %3d ", dis);

motor\_set(0, 250);

break;

case DIS\_CLOSE:

glcd\_printf("close VAL : %3d ", dis);

motor\_set(0, 500);

break;

case DIS\_FAR:

glcd\_printf("far VAL : %3d ", dis);

motor\_set(0, 750);

break;

case DIS\_VERYFAR:

glcd\_printf("very far VAL : %3d ", dis);

motor\_set(0, 1000);

break;

default:

glcd\_printf("UNDEFINE ");

break;

}

task\_sleep(30); // **sensing 주기를 30ms로 합니다.**

}

}

6. 자료 출처 (어떤 책을 참고했으며 어떤 웹사이트를 참고했는지 출처를 적습니다)

교수님의 ppt를 참고했습니다.