



로봇활용 SW교육 지침서

The NEXT ROBOT with EV3

EV3로 배우는 블록 코딩 & C언어

2017년 2학기

인제대학교 헬스케어IT 학과

이상훈



Weekly plan (2nd semester, 2017)

- **wk01 : Introduction to curriculum & current state of HW-SW coding**
- **wk02 : LME blocking coding-1: Start & How To**
- **wk03 : LME blocking coding-2: Loop & Driving**
- **wk04 : LME blocking coding-3: Project 1. driving base**
- **wk05 : LME blocking coding-4: Sensors**
- **wk06 : wk11 : Special talk by prof. Redwoods Yi**
- **wk07 : wk15 : RobotC coding-4: Project**
- **wk08 : Mid-term Exam.**
- **wk09 : LME blocking coding-5: Math and Logic**
- **wk10 : LME blocking coding-6: Summary & Project**
- **wk11 : Special talk by CEO of HandsOn Tech.**
- **wk12 : RobotC coding-1: Intro**
- **wk13 : RobotC coding-2: function, array, motor**
- **wk14 : RobotC coding-3: Encoder & data logging**
- **wk16 : Final exam.**



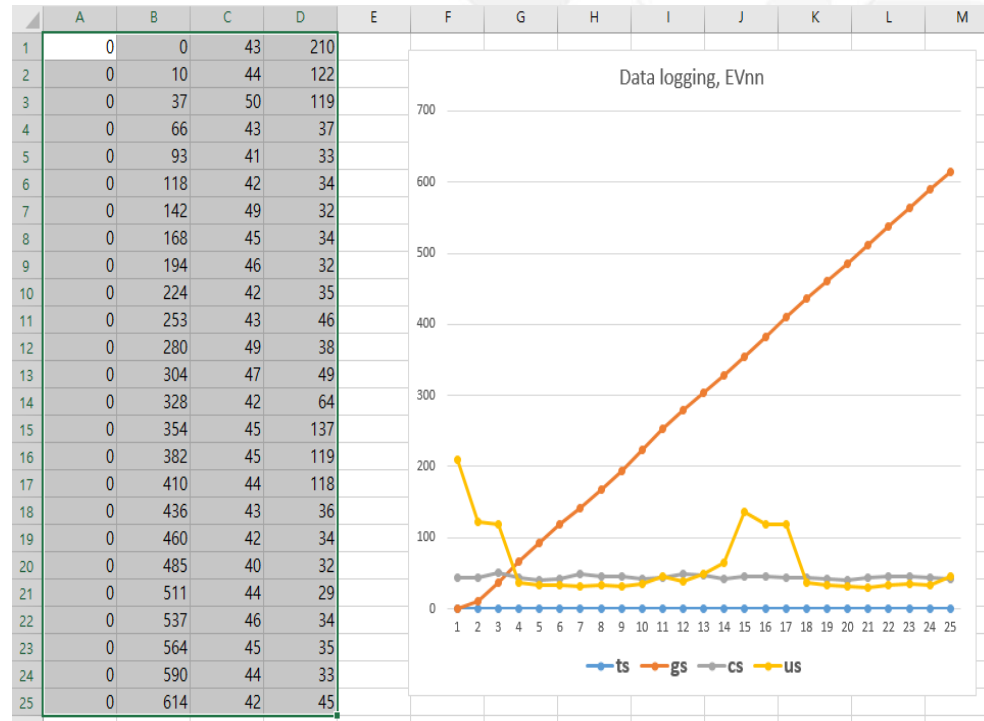
Report

제출파일명 : wk14_EVnn.zip

- 압축할 파일들

- ① EVnn_RC_DL.png
- ② EVnn_RC_DL.csv
- ③ EVnn_RC_DL2.xls

Email : chaos21c@gmail.com



wk15: RobotC IV. Team Project

LEGO® Mindstorms® EV3

powered by LEGO® MINDSTORMS® Education



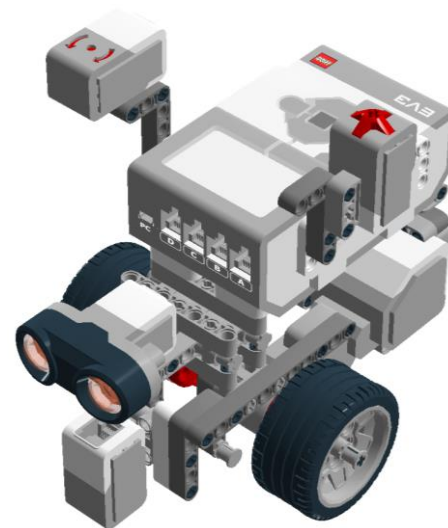
1부 EV3로 배우는 블록 코딩

I . LEGO® MINDSTORMS® Education EV3

1. EV3와 NXT 비교, 브릭 인터페이스
2. Starting block coding
 - ✓ Awake EV3!
 - ✓ Loop & Driving
 - ✓ Driving base
 - ✓ Sensors
 - ✓ Advanced coding

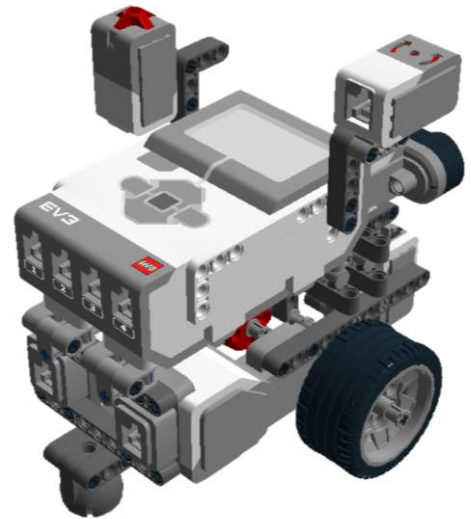
2. ROBOTC 기초

- ◆ ROBOTC 규칙
- ◆ 시간(지연) 함수
- ◆ 데이터 저장 및 계산
- ◆ 조건 선택/실행
- ◆ 반복 실행
- ◆ 함수
- ◆ 배열



3. 액추에이터 제어

- ◆ 모터
- ◆ 디스플레이
- ◆ 스피커
- ◆ LED



4. 센서 활용

- ◆ 터치 센서
- ◆ 컬러 센서
- ◆ 초음파 센서
- ◆ 자이로 센서
- ◆ 엔코더 제어
- ◆ 타이머 활용
- ◆ 데이터 로깅



엔코더 제어의 기초

- EV3 타이어의 지름을 이용하여 모터의 엔코더가 360°회전 시 이동한 거리를 알 수 있음



지름 = 5.6cm

원의 둘레 = $2\pi r$ (r : 반지름) 의 공식에서 $2r$ 인 바퀴의 지름이 5.6cm이므로 엔코더가 360도 회전했을 때 이동 거리는 약 17.58($5.6 * 3.14$)cm로 계산할 수 있다.

- 이 원리를 적용하여 임의의 거리를 직진하기 위해 회전해야 할 엔코더의 근사값을 구할 수 있음

$$360^\circ : 17.58\text{cm} = D^\circ : X\text{cm} \Rightarrow D = 360 * X / 17.58$$



모터의 엔코더 값 이용하기

- 모터 값 50으로 1초간 직진하는 프로그램을 작성했을 때, 속도를 높이거나 낮추면 이동 거리가 늘어나거나 줄어듦
- 똑같은 거리를 이동하기 위해서는 `sleep()` 함수나 `wait1Msec()` 함수에 지정한 대기 시간을 변경해야 함
- 엔코더 값을 이용하면 이동 속도 및 대기 시간과 상관없이 항상 일정한 거리를 이동을 할 수 있음

모터 동기화 (Syncing motors)

- ◆ 모터 동기화란 하나의 모터를 기준으로 하여 두 개의 모터가 지정한 비율로 회전하도록 하는 것
- ◆ 모터 값을 기준으로 동기화 시 `setMotorSyncTime()` 함수를,
엔코더 값을 기준으로 동기화 시 `setMotorSyncEncoder()` 함수를 사용

setMotorSyncTime(lm, rm, 100, 1000, 30)					
setMotorSyncTime()	lm	rm	100	1000	30
지정한 시간 동안 모터 동기화 실행	기준 모터	다른 모터	기준 모터 출력 비율	지정 시간	모터 값

setMotorSyncEncoder(lm, rm, 100, 1000, 30)					
setMotorSyncEncoder()	lm	rm	100	1000	30
지정한 엔코더 값에 도달할 때까지 모터 동기화 실행	기준 모터	다른 모터	기준 모터 출력 비율	지정 엔코더 값	모터 값

Tip 타이머

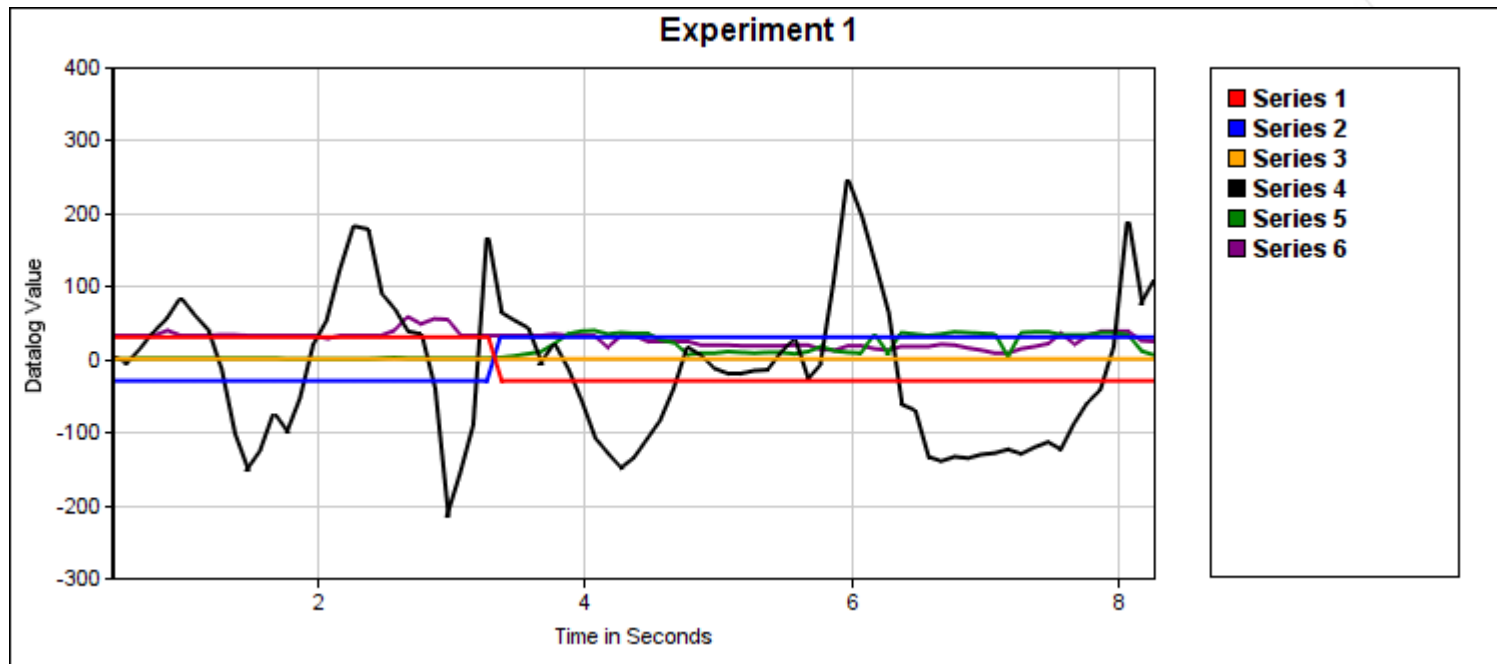
- ◆ 타이머는 T1~T4까지 4개를 사용 가능
- ◆ 타이머 값은 time1 []을 이용하여 1/1000초 단위로 측정할 수 있음
- ◆ time10 []은 1/100초, time100 []은 1/10초 단위로 타이머 값을 측정할 때 이용

데이터 로깅 (Logging data)

- ◆ 데이터 로깅 : 로봇 동작 중 특정 센서 값 및 모터 값의 변화를 기록하는 방법
- ◆ 데이터 로깅 실행 방법은 2가지로, 'Program Debug' 대화 창을 이용하는 방법과 데이터 로깅 함수를 이용하는 방법이 있음

데이터 로깅 - 'Program Debug' 대화창

- ◆ 'Datalog Graph' 대화 창의 저장 버튼 이용 시 데이터 로깅 결과를 텍스트 파일(txt) 또는 이미지 파일(.png)로 저장 가능



→ 결과 저장 : [EVnn_RC_DL.png](#)

데이터 로깅 - 'Program Debug' 대화창

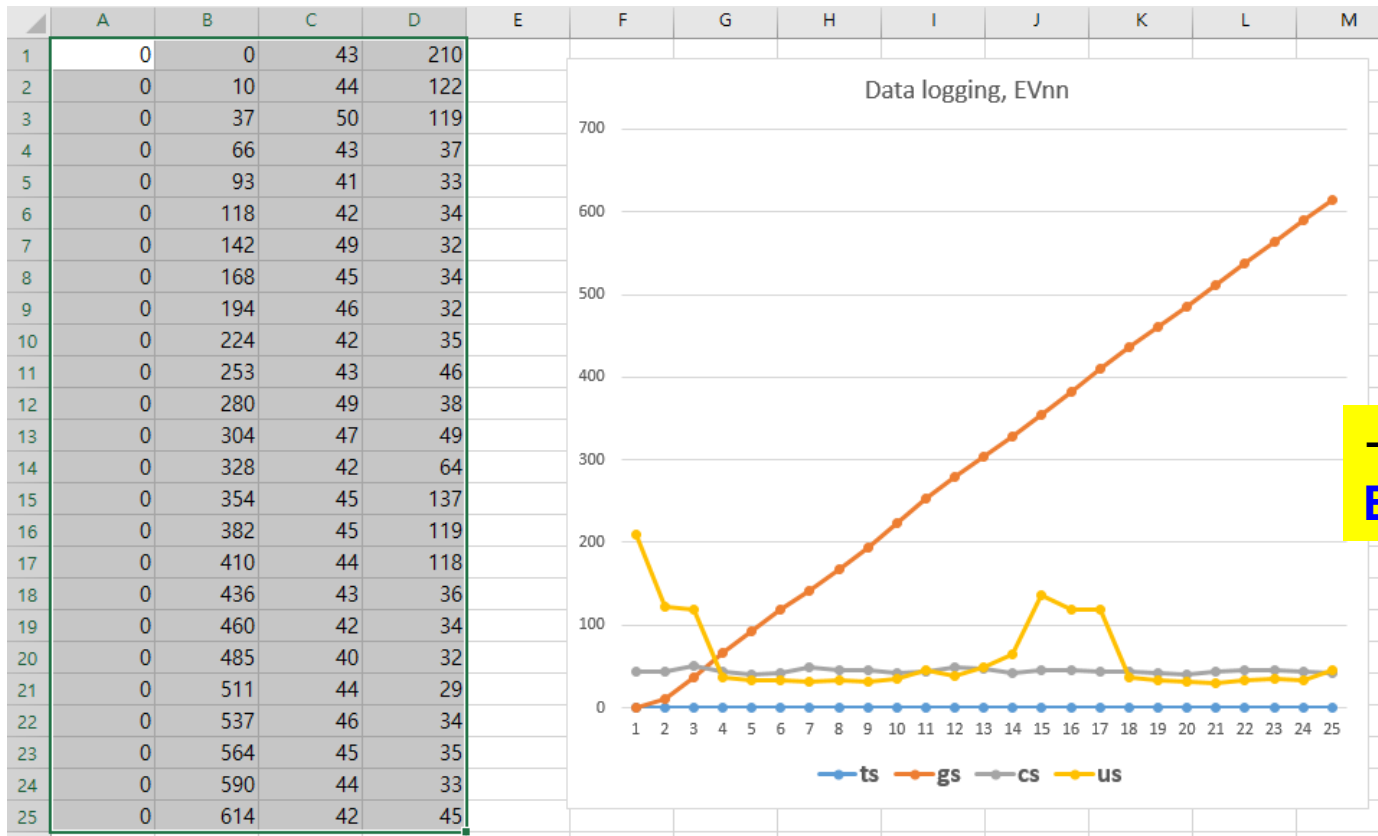
- 'Datalog Graph' 대화 창의 저장 버튼 이용 시 데이터 로깅 결과를 텍스트 파일(csv)로 저장 가능

	A	B	C	D	E	F	G	H
1	Row	Timestamp	Series 1	Series 2	Series 3	Series 4	Series 5	Series 6
2	0	0.47	30	-30	0	3	0	32
3	1	0.57	30	-30	0	-5	0	32
4	2	0.67	30	-30	0	15	1	32
5	3	0.77	30	-30	0	38	1	32
6	4	0.87	30	-30	0	56	1	39
7	5	0.97	30	-30	0	84	1	32
8	6	1.07	30	-30	0	61	1	32
9	7	1.17	30	-30	0	41	1	32
10	8	1.27	30	-30	0	-13	1	33
11	9	1.37	30	-30	0	-101	1	33
12	10	1.47	30	-30	0	-151	1	32
13	11	1.57	30	-30	0	-125	1	32
14	12	1.67	30	-30	0	-74	1	32
15	13	1.77	30	-30	0	-100	0	32
16	14	1.87	30	-30	0	-53	0	32
17	15	1.97	30	-30	0	21	0	32
18	16	2.07	30	-30	0	54	0	28
19	17	2.17	30	-30	0	124	0	32
20	18	2.27	30	-30	0	182	0	32
21	19	2.37	30	-30	0	178	0	32
22	20	2.47	30	-30	0	91	1	32

→ 데이터 저장 :
EVnn_RC_DL.csv

데이터 로깅 - 데이터 로깅 함수 응용

- 스프레드시트로 불러온 데이터를 가공하여 새로운 정보(평균, 순위 등)를 추출할 수 있음
- 시간에 따른 센서 값의 변화 과정 확인할 수 있음



→ 데이터 저장 :
EVnn_RC_DL2.xls



ROBOT

Team Project

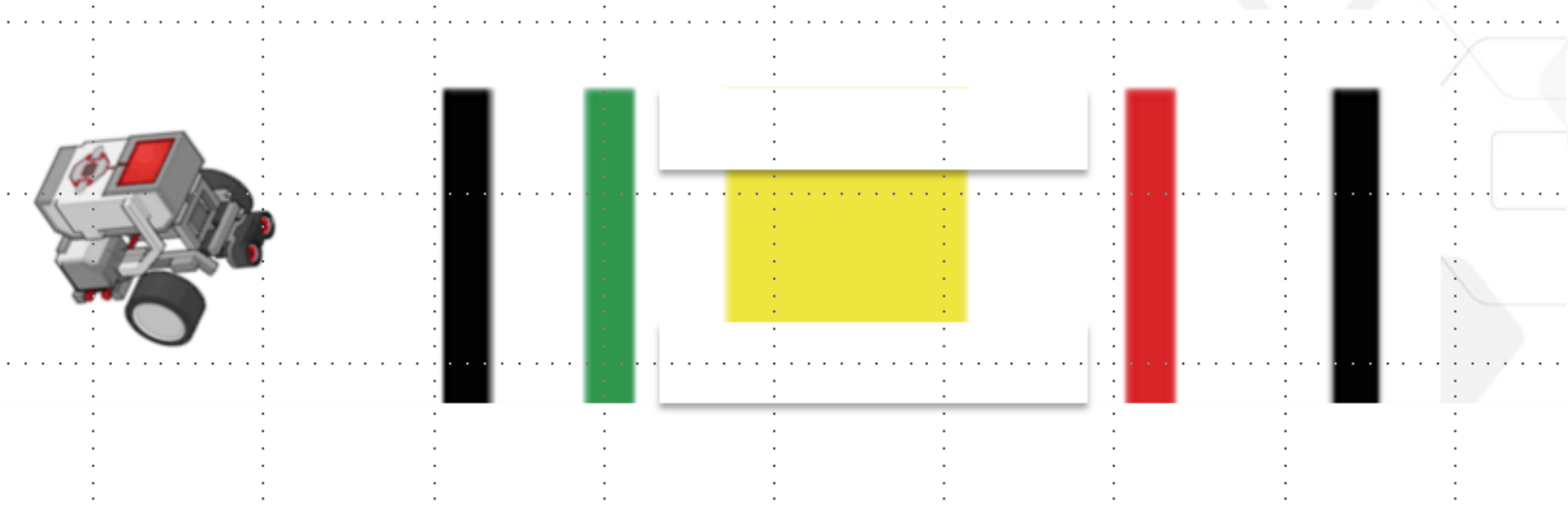


education

장의공학교육의 멘토

HandsOn
Technology

RobotC project-3: Auto pilot



무인 자동차 만들기 (Auto pilot)

- 서로 다른 색으로 신호등과 시작/종료 위치가 표시될 때 다음의 작업을 수행하는 무인자동차를 표현해보자!

- 설정1 : 바닥 배경은 흰색이다.
- 설정2 : 신호등의 색은 초록/노랑/빨강으로 표시되며, 순서는 임의로 정해진다.
- 설정3 : 시작/종료 위치는 검정/파랑/갈색 중 임의의 색으로 정해지며, 시작/종료 위치의 색은 같다.
가장 처음 나타나는 색은 검정/파랑/갈색 중 하나이다. 하지만, 가장 끝의 색은 신호등의 색일 수도 있다.
- 수행 작업1 : 터치 센서를 누르면 출발한다.
- 수행 작업2 : 시작 지점을 인식하면 브릭의 LED를 초록으로 표시한다.
- 수행 작업3 : 초록 신호등을 인식하면 소리를 낸다.
- 수행 작업4 : 노랑 신호등을 인식하면 노랑 신호등 구간 내에서 속도를 1/2로 줄인다.
노랑 신호등 구간을 벗어나면 원래의 속도로 복귀한다.
- 수행 작업5 : 빨강 신호등을 인식하면 1초간 정지한 후 출발한다.
- 수행작업4A: 노랑 신호등 구간을 통과시 경고음 효과를 추가한다.
- 수행작업6: 파랑 신호등이면 속도를 50% 올린다.
- 수행작업7: 종료 할 때 종료 효과음을 넣는다..



[1] 무인 자동차 만들기 : 색 구분 -1

tVnn_checkcolor_start.c

```
1  #pragma config(Sensor, S1,      ts,      sensorEV3_Touch)
2  #pragma config(Sensor, S2,      gs,      sensorEV3_Gyro, modeEV3Gyro_Rate)
3  #pragma config(Sensor, S3,      cs,      sensorEV3_Color, modeEV3Color_Color)
4  #pragma config(Sensor, S4,      us,      sensorEV3_Ultrasonic)
5  #pragma config(Motor,  motorB,  lm,      tmotorEV3_Large, PIDControl, driveLeft, encoder)
6  #pragma config(Motor,  motorC,  rm,      tmotorEV3_Large, PIDControl, driveRight, encoder)
7  /**!!Code automatically generated by 'ROBOTC' configuration wizard !!*/
8
9  #define None 0
10 #define Black 1
11 #define Blue 2
12 #define Green 3
13 #define Yellow 4
14 #define Red 5
15 #define White 6
16 #define Brown 7
17
18 int color;
```


ROBOTC

a C Programming Language for Robotics

getColorName

- This command will return the name of the color detected by the Color Sensor
 - One of 8 different color names will be returned, depending on the color of the object detected.
 - If no color is detected, colorNone will be returned instead.

	Color name	Enumerated Value
colorNone:	No object is detected by the color sensor	0
colorBlack:	A black object is detected by the color sensor	1
colorBlue:	A blue object is detected by the color sensor	2
colorGreen:	A green object is detected by the color sensor	3
colorYellow:	A yellow object is detected by the color sensor	4
colorRed:	A red object is detected by the color sensor	5
colorWhite:	A white object is detected by the color sensor	6
colorBrown:	A brown object is detected by the color sensor	7

[1] 무인 자동차 만들기 : 색 구분 -2

```

20 task main()
21 {
22     // check the color of the floor and traffic signs.
23     string colorOfObject;
24
25     //Keep looping forever
26     //repeat(forever)
27     while(1)
28     {
29         while(getTouchValue(ts)==1) {
30
31             if(getColorName(cs) == colorRed)
32             {
33                 //Store the text RED into the color string
34                 colorOfObject = "RED";
35                 color = Red;
36                 //break;
37             }
38             //If the color sensor plugged into port 3 returns
39             // the colorBlue name (a Blue object is detected)
40             else if(getColorName(cs) == colorGreen)
41             {
42                 //Store the text GREEN into the color string
43                 colorOfObject = "GREEN";
44                 color = Green;
45                 //break;
46             }
47             else if(getColorName(cs) == colorBlue)
48             {
49                 //Store the text Blue into the color string
50                 colorOfObject = "BLUE";
51                 color = Blue;
52                 //break;
53             }

```

```

53
54
55         //Store the text YELLOW into the color string
56         colorOfObject = "YELLOW";
57         color = Yellow;
58         //break;
59     }
60     else if(getColorName(cs) == colorBlack)
61     {
62         //Store the text Blue into the color string
63         colorOfObject = "BLACK";
64         color = Black;
65         //break;
66     }
67     else if(getColorName(cs) == colorWhite)
68     {
69         //Store the text WHITE into the color string
70         colorOfObject = "WHITE";
71         color = White;
72         //break;
73     }
74     else if(getColorName(cs) == colorBrown)
75     {
76         //Store the text BROWN into the color string
77         colorOfObject = "BROWN";
78         color = Brown;
79         //break;
80     }
81     else
82     {
83         //Store the text NONE into the color string
84         colorOfObject = "NONE";
85         color = None;
86         //break;
87     }
88
89     displayBigTextLine(3, "Color : %s", colorOfObject);
90     displayBigTextLine(5, "Color # : %d", color);
91     sleep(3000);
92 } // end of while(getTouchValue(ts)==1)
93 } // end of while(1)
94 } // main loop

```


[2] 무인 자동차 만들기 : autopilot-1

4~15	무인 자동차
01	<code>#pragma config(Sensor, S1, ts, sensorEV3_Touch)</code>
02	<code>#pragma config(Sensor, S3, cs, sensorEV3_Color, modeEV3Color_Color)</code>
03	<code>#pragma config(Motor, motorB, lm, tmotorEV3_Large, PIDControl, encoder)</code>
04	<code>#pragma config(Motor, motorC, rm, tmotorEV3_Large, PIDControl, encoder)</code>
05	
06	<code>#define Black 1</code>
07	<code>#define Blue 2</code>
08	<code>#define Green 3</code>
09	<code>#define Yellow 4</code>
10	<code>#define Red 5</code>
11	<code>#define White 6</code>
12	<code>#define Brown 7</code>
13	
14	<code>int color, finish_line, speed = 40, start_val = 0;</code>
15	
16	<code>void go(int s)</code>
17	<code>{</code>
18	<code> setMotorSpeed(lm, s);</code>
19	<code> setMotorSpeed(rm, s);</code>
20	<code>}</code>
21	

06~12 사용할 색상 값들을 문자로 정의한다.

16~20 지정한 모터 값으로 직진하기 위한 함수를 정의한다.

[2] 무인 자동차 만들기 : autopilot-2

```

22 task main()
23 {
24     while(getTouchValue(ts) == 0){ }
25     while(getTouchValue(ts) == 1){ }
26     while(1)
27     {
28         color = getColorName(cs);
29         go(speed);
30         if(!start_val && (color != White))
31         {
32             finish_line = color;
33             setLEDColour(ledGreen);
34             sleep(700);
35             start_val++;
36             color = getColorName(cs);
37         }
38         if(color == finish_line)
39         {
40             go(0);
41             setLEDColour(ledRed);
42             sleep(500);
43             break;
44         }

```

24~25 터치 센서가 눌렀다 떼질 때까지 대기한다.

26 무한 반복(27~67번 줄) 한다.

28 컬러 센서의 값을 color 변수에 저장한다.

29 speed 변수의 모터 값으로 직진한다.

30 start_val 변수의 값이 0이고, 흰색이 아니면 시작 라인으로 인식한다.

32 finish_line 변수에 컬러 값을 저장한다.

33~34 LED 색을 초록으로 하고, 라인을 빠져나오도록 0.7초 더 직진한다.

35 시작 라인의 색을 인식 했으므로 start_val 변수의 값을 1로 바꿔서 이후 신호등의 색과 종료 라인의 색을 처리할 수 있도록 한다.

36 라인을 빠져나온 후 컬러 센서의 값을 보관하는 color 변수를 업데이트 한다.

38~44 color와 finish_line 변수의 값이 같은 경우 종료 라인이므로 정지하여 LED를 빨강으로 설정하고, 0.5초 대기한 후 루프를 빠져나온다.

[2] 무인 자동차 만들기 : autopilot-3

```

45  if(color == Green)
46  {
47      playSound(soundBeepBeep);
48      sleep(700);
49  }
50  if(color == Yellow)
51  {
52      go(0);
53      while(color == Yellow)
54      {
55          color = getColorName(cs);
56          go(speed/2);
57      }
58      go(speed);
59  }
60  if(color == Red)
61  {
62      go(0);
63      sleep(1000);
64      go(speed);
65      sleep(700);
66  }
67  }
68  }

```

45~49 초록 신호등을 인식한 경우 소리를 낸다.

50 노랑 신호등을 인식한 경우

52~57 일단 모터 값을 0으로 한 후 직진 속도를 1/2로 변경한다.

58 노랑 신호등 구간을 빠져나온 후 직진 속도를 처음 값으로 변경한다.

60~66 빨강 신호등을 인식한 경우 1초간 멈춘 후 라인을 빠져나오도록 0.7초 더 직진한다.

EVnn_AutoPilot.c
로 저장.

[3] 무인 자동차 만들기 : Data logging-1

How to do logging data during autopilot?

[DL1] 라이브러리 함수를 이용한다.

[DL2] 주행 중 측정되는 도로의 색과 주행 속도를 저장한다.

[DL3] 색과 속도를 저장할 때마다 시간도 기록한다.

[DL4] 데이터를 기록하는 시간 간격은 50 ms (샘플링주파수 = 20 Hz).



[hint1] 3종류 데이터 `datalogOpen(0, 3, false); // time, color, speed`

[hint2] 타이머 이용 `clearTimer(T1); // initialization of timer`

[hint3] 데이터 저장함수 `void putData(int color, int speed)`

[3] 무인 자동차 만들기 : Data logging-2

How to do logging data during autopilot? → code hint

```
void putData(int color, int speed)
{
    int t = time1[T1]; // msec
    datalogAddShort(0, t);
    datalogAddShort(1, color);
    datalogAddShort(2, speed);
    sleep(50);
}

task main()
{
    while(getTouchValue(ts)==0){}
    while(getTouchValue(ts)==1){} // start

    clearTimer(T1); // initialization of timer

    datalogOpen(0, 3, false); // time, color, speed
```

[3] 무인 자동차 만들기 : Data logging-3

How to do logging data during autopilot? → code hint

```
while(1)
{
    color = getColorName(cs);
    go(speed);

    putData(color, speed);

    if(!start_val && (color != White))
    {
        finish_line = color; // match the start and finish
        setLEDColour(ledGreen);
        sleep(700);
        start_val++;
        color = getColorName(cs);
    }

    if(color == finish_line)
    {
        go(0);
        setLEDColour(ledRedPulse);
        playSound(soundDownwardTones);
        sleep(500);

        putData(color, 0);

        break;
    }
}
```

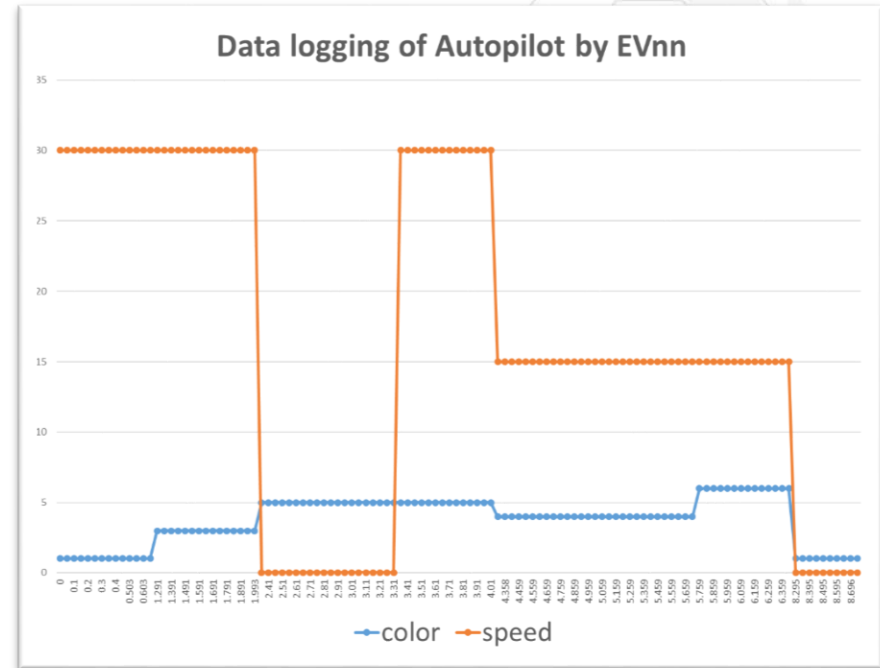
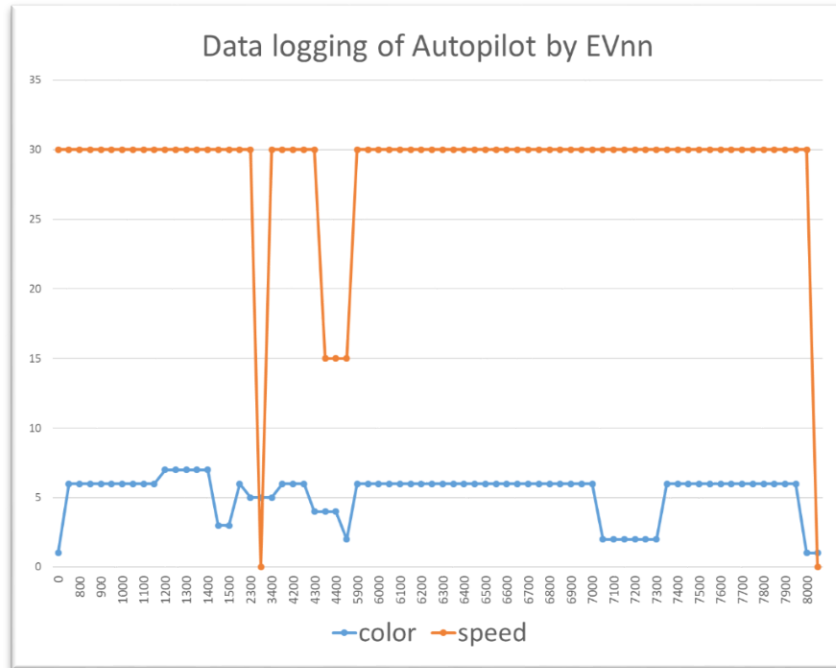
```
if(color == Green)
{
    playSound(soundBeepBeep);
    putData(color, speed);
    sleep(700);
}

if(color == Yellow)
{
    go(speed/2);
    putData(color, speed/2);
    while(color == Yellow)
    {
        color = getColorName(cs);
        go(speed/2);
        playSound(soundLowBuzzShort);
        putData(color, speed/2);
        sleep(700);
    }
    go(speed);
}
```

EVnn_AutoPilot_DL.c
로 저장.

[3] 무인 자동차 만들기 : Data logging-4

Make an excel chart using the logged data.



데이터 저장 시간이 일정하지 않는 문제
→ sleep(시간) 처리가 필요

데이터 저장 시간이 일정
시간이 초단위로 표시됨.

[3] 무인 자동차 만들기 : Data logging-5

Code hint:

sleep(time) 을
time/50
만큼의 횟수로
putData 함수를
이용하여 변경함.

주의:
각 색깔에서의
속도를 저장.

```
task main()
{
    while(getTouchValue(ts)==0){}
    while(getTouchValue(ts)==1){} // start

    clearTimer(T1); // initialization of timer

    datalogOpen(0, 3, false); // time, color, speed

    while(1)
    {
        color = getColorName(cs);
        go(speed);

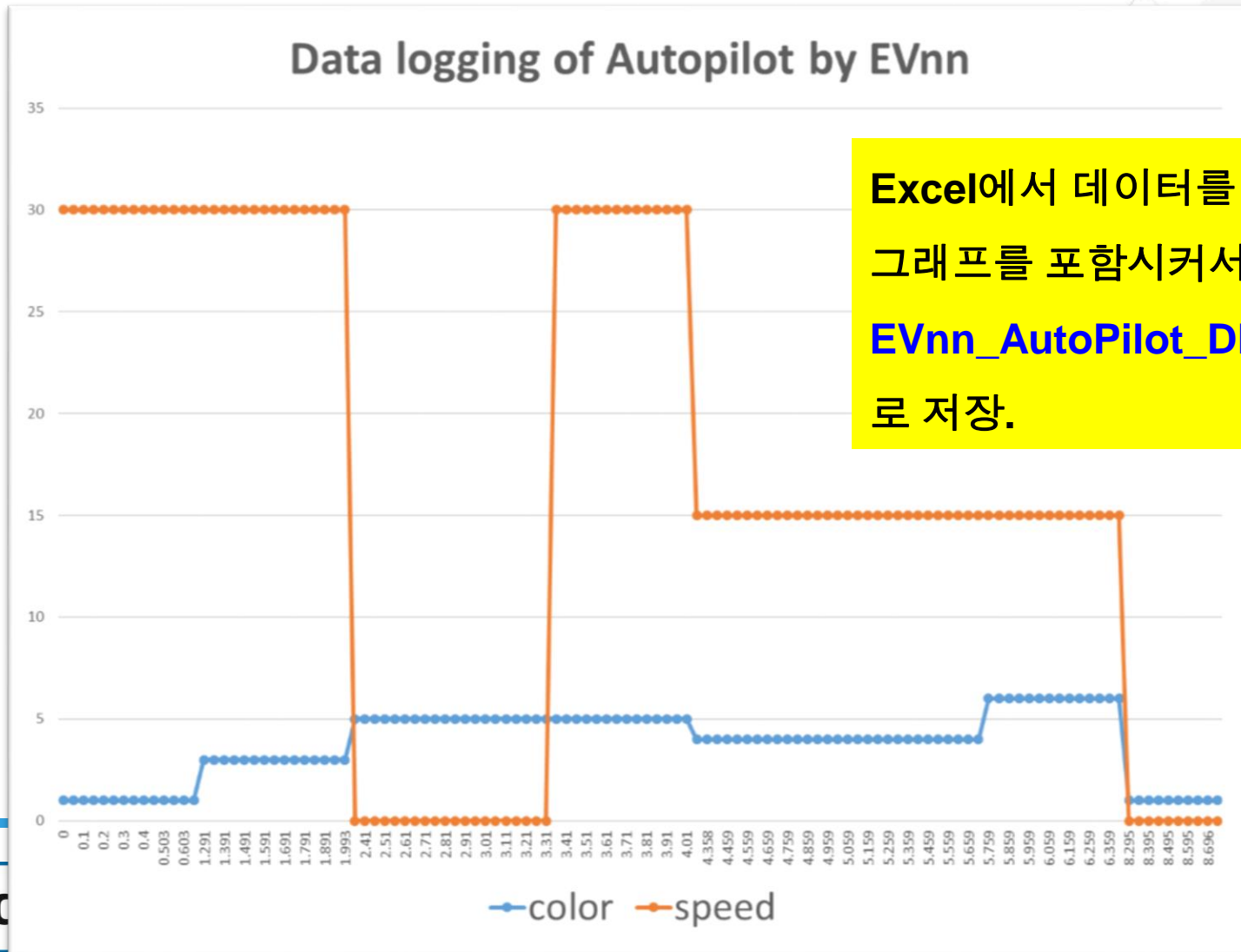
        //putData(color, speed);

        if(!start_val && (color != White))
        {
            finish_line = color; // match the start and finish
            setLEDColor(ledGreen);
            //sleep(700);
            for(int i=0;i<700/50;i++)
            {
                putData(color, speed);
            }
            start_val++;
            color = getColorName(cs);
        }
    }
}
```

EVnn_AutoPilot_DL.c
로 저장.

[3] 무인 자동차 만들기 : Data logging-6

Make an excel chart using the logged data.



Excel에서 데이터를 로드하여
그래프를 포함시켜서
EVnn_AutoPilot_DL.xls
로 저장.

Final Project Report (10 %)

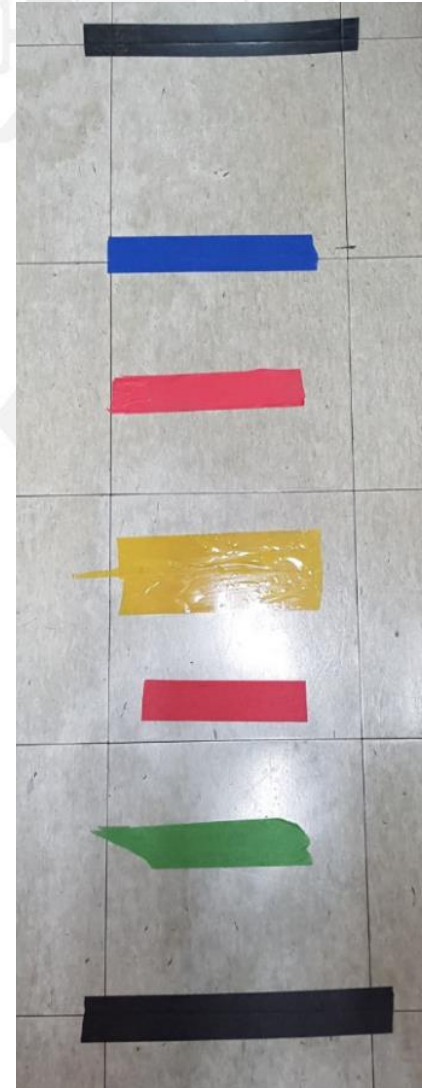
제출파일명 : wk15_EVnn.zip

- 압축할 파일들

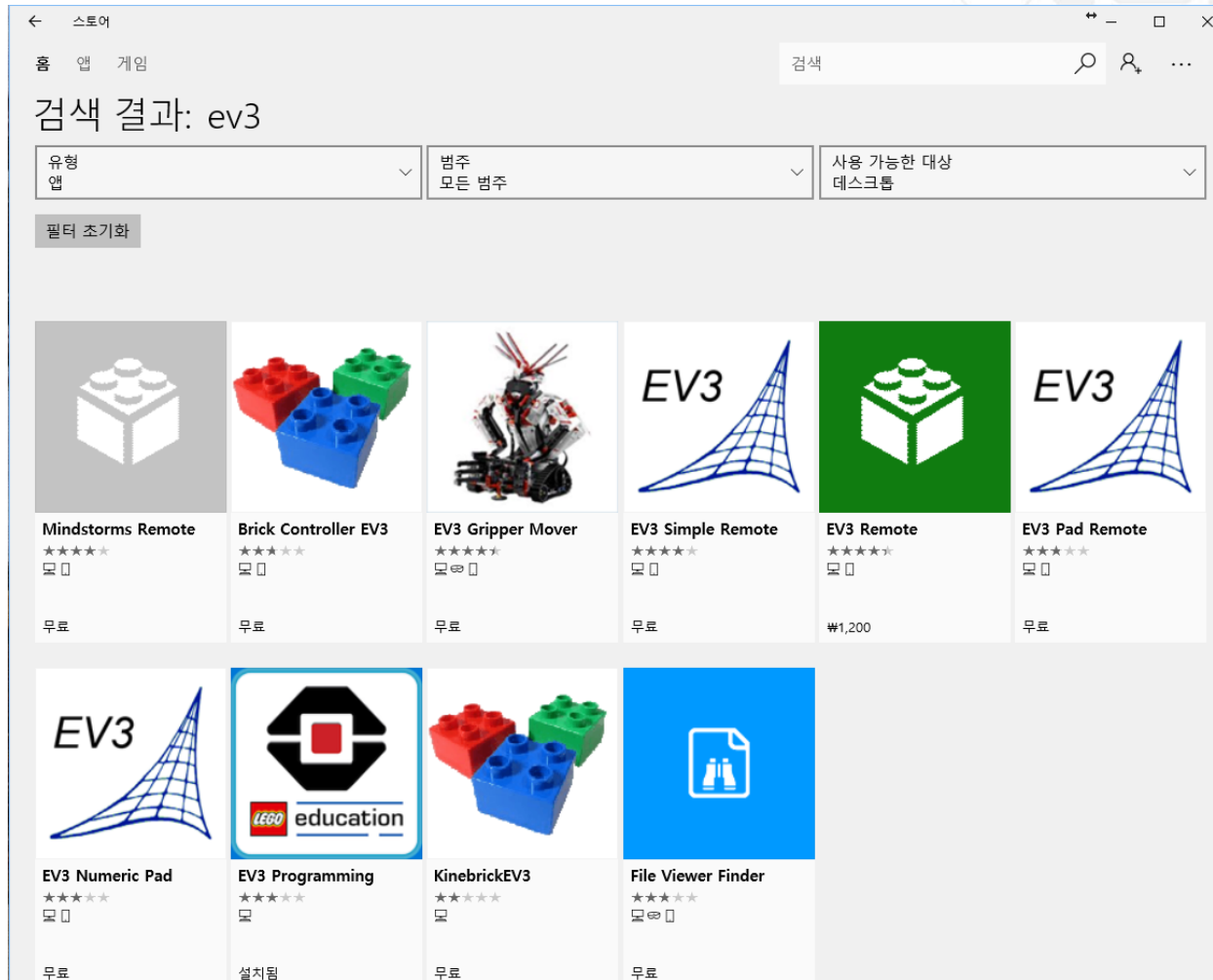
① Evnn_Autopilot_DL.c

② EVnn_Autopilot_DL.xls

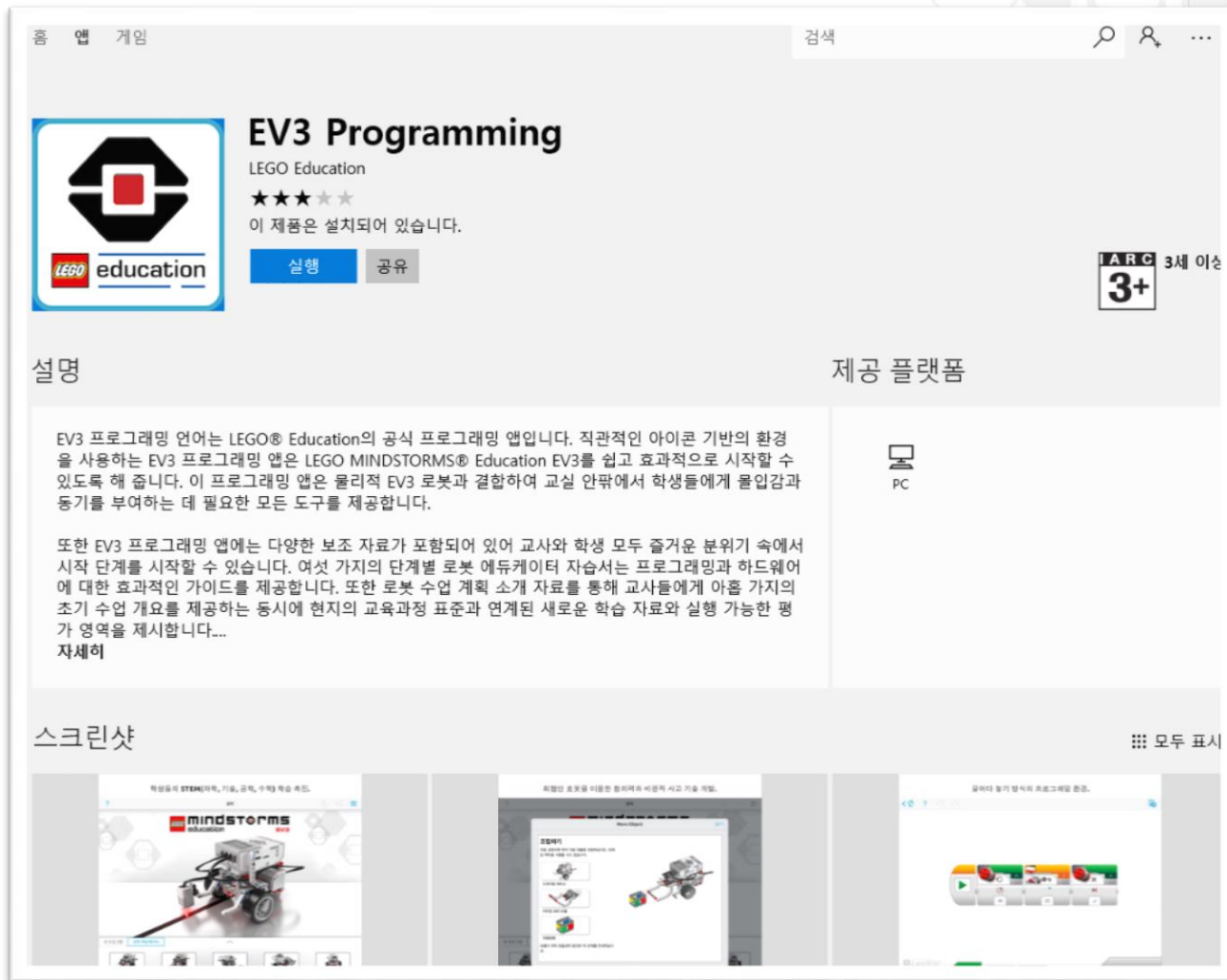
Email : chaos21c@gmail.com



EV3 Programming App (windows 10)



EV3 Programming App (windows 10)




education

창의공학교육의 멘토

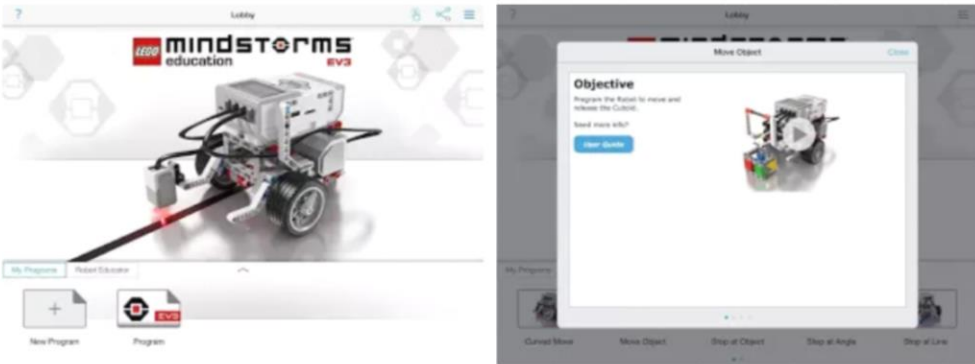
HandsOn
Technology

EV3 Programming App (Android)



LEGO® MINDSTORMS...
LEGO Education

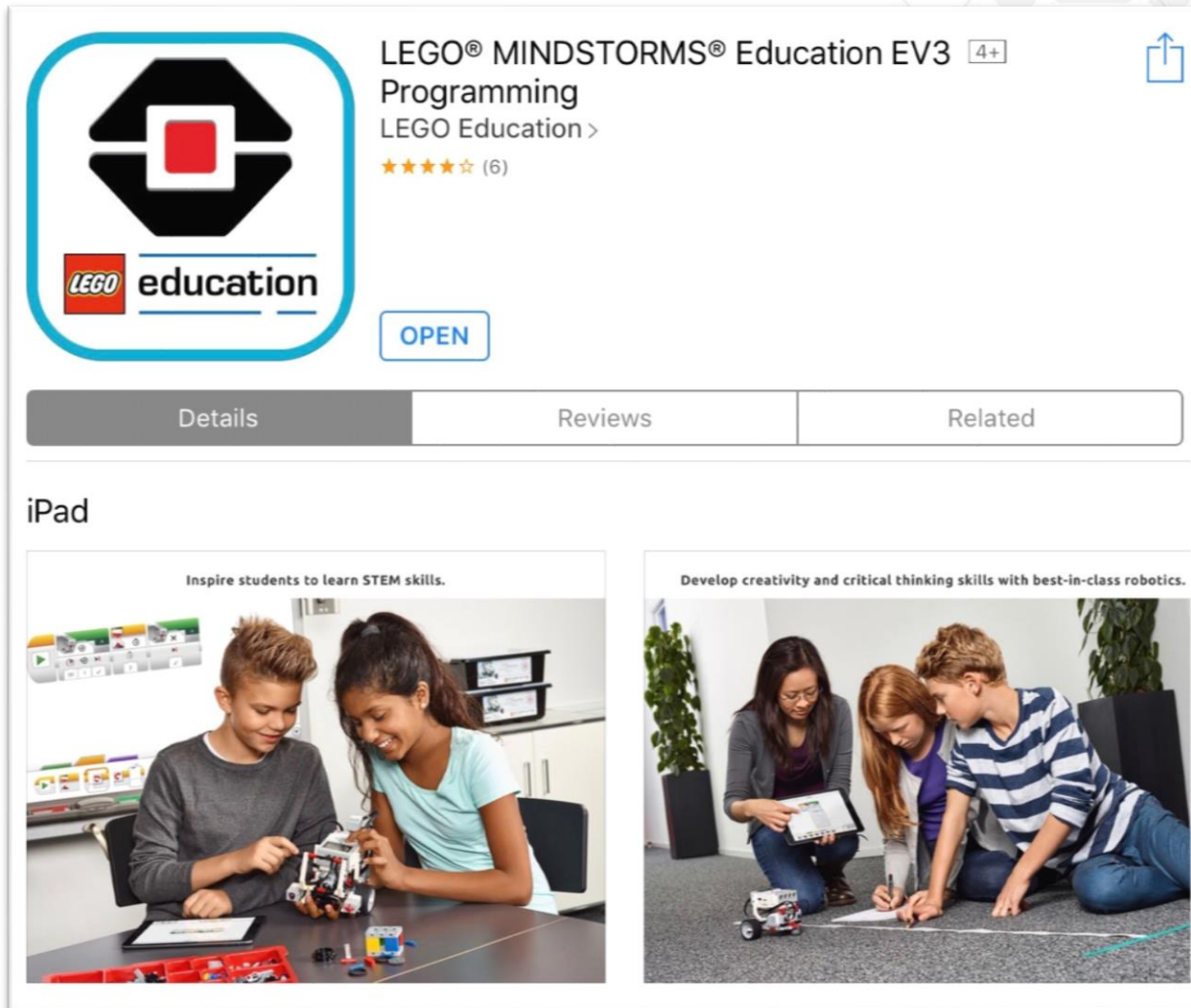
4.2 ★ (164 👤) • 1만 ↓

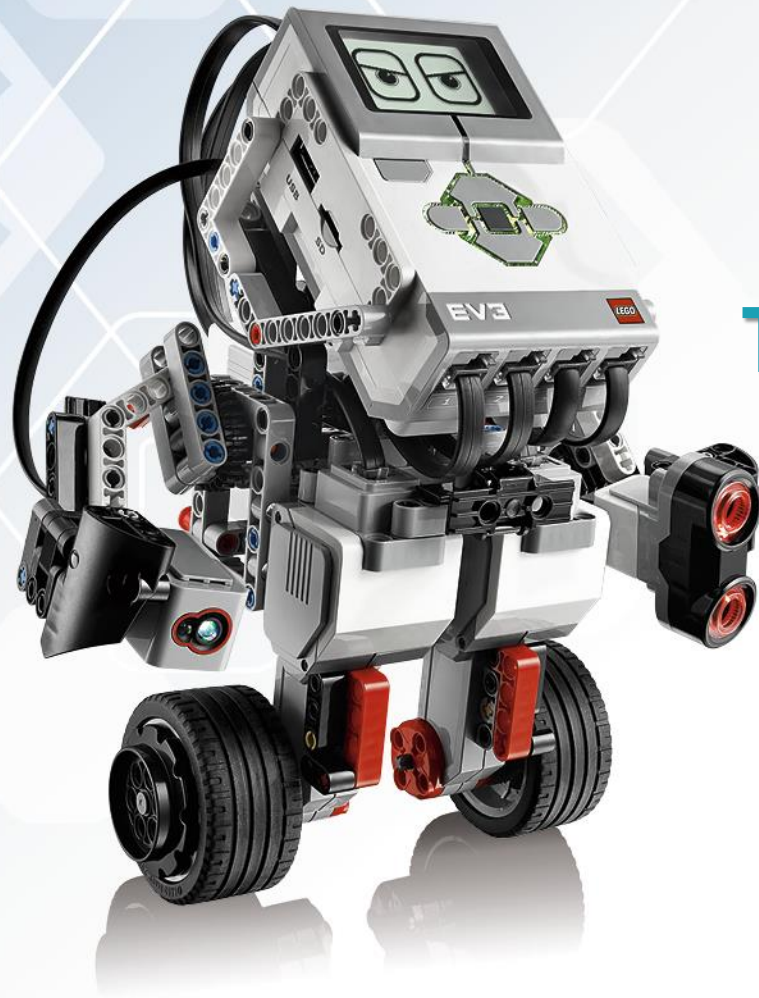


LEGO® MINDSTORMS® Education EV3
프로그래밍 언어

[추가 정보](#)[설치](#)

EV3 Programming App (iPad/iPhone)





로봇활용 SW교육 지침서

The NEXT ROBOT with EV3

EV3로 배우는 C언어와 알고리즘

정웅열 · 최웅선 · 정종광 · 전준호 · 배상용 · 전현석
이선경 · 경다은 · 김제현 · 오범석 · 이찬호 지음

Partnership



LEGO education Partner
- Oct. 2011



NATIONAL INSTRUMENTS
OFFICIAL ALLIANCE MEMBER
- Mar. 2003



PITSCO Education
Distributor in Korea
- Jan. 2010

