
***** EV3 RobotC Coding 기말고사. 2017. 12. 11(월) *****

EV3 C 코딩을 즐기자!

1. 다음 중 RobotC에서 특정 시간동안 다음 코드로 넘어가지 않고 지연시키는 함수는?

- A. delay()
- B. time()
- C. sleep()
- D. wait()

2. EV3 Brick을 구동하는 내부 운영 체제는?

- A. Windows
- B. ARM
- C. Android
- D. Linux
- E. iOS

3. 다음 RobotC 코딩 규칙 중 틀린 것은?

- A. ROBOTC의 컴파일러는 대·소문자를 구분하지 않음
- B. ROBOTC 프로그램의 문장의 끝에는 세미콜론(;)이 들어가야 함
- C. 주로 사용되는 키워드는 색이 다르게 나타남
- D. 코드 설명을 추가하기 위해 코드 옆에 '//'를 이용하여 한 줄 주석을 달 수 있음

4. 다음 전처리 구문 설명 중 틀린 것은?

- A. task main() 함수보다 먼저 처리되는 구문
- B. #include, #define, #pragma 등과 같이 '#'으로 시작하고 ';'을 끝에 사용
- C. #define은 기호를 정의하여 프로그램 전체에 대해 대체하도록 하는 기능을 함
- D. 매크로라고도 하며, 매크로가 실행되면 정의된 기호가 상수 값을 대체

5. 다음 RobotC 변수 선언 중 틀린 것은?

- A. int _car;
- B. char HI;
- C. int 3days;
- D. string msg;

6. 다음은 ASCII 코드 문자를 처리하는 RobotC 코드이다.

출력되는 결과는? --- (**d**)

```
task main()
{
    char x = 'a'; // 'a' -> 97
    x = 100;

    displayBigTextLine(5, "x is %c", x);
    sleep(5000);
}
```

7-8. 다음은 casting (형변환)을 처리하는 RobotC 코드이다.

출력되는 결과가 "y is 3"이 되도록 중간 밑줄 친 곳에 들어갈 코드는?

```
task main()
{
    float x = 3.792;
    int y;

    y = [7]_____ (int) x;

    displayBigTextLine(5, "y is [8]_____", y);
    sleep(5000);
}
```

7. 실수인 x를 정수로 casting하는 코드는?

- | | |
|--------------------|-------------|
| A. int x; | B. [int] x; |
| C. (int) x; | D. {int} x; |

8. 정수로 출력형을 지정하는 코드는?

- | | | | |
|-------|--------------|-------|-------|
| A. %i | B. %d | C. %c | D. %f |
|-------|--------------|-------|-------|

9-10. 다음은 두 개의 터치센서를 이용한 주행 코드이다.

```
#pragma config(Sensor, S1, rts, sensorEV3_Touch)
#pragma config(Sensor, S2, lts, sensorEV3_Touch)
#pragma config(Motor, motorB, lm, tmotorEV3_Large, PIDControl, driveLeft, encoder)
#pragma config(Motor, motorC, rm, tmotorEV3_Large, PIDControl, driveRight, encoder)
task main()
{
    int a;
    while(1)
    {
        a = getTouchValue(rts) + getTouchValue(lts)*2;
        switch(a)
        {
            case 1:
                setMotorSpeed(lm, 20);
                setMotorSpeed(rm, 0);
                break;
            case 2:
                setMotorSpeed(lm, 0);
                setMotorSpeed(rm, 20);
                break;
            case 3:
                setMotorSpeed(lm, 20);
                setMotorSpeed(rm, 20);
                break;
            [10]_____default:
                setMotorSpeed(lm, 0);
                setMotorSpeed(rm, 0);
                break;
        }
    }
}
```

9. 좌측 터치센서(lts)를 누르면 주행 상태는 어떻게 되는가?

- A. 좌회전 B. 우회전 C. 직진 D. 정지

10. 두 개의 터치센서를 둘 다 누르지 않은 경우를 처리하는 키워드는?

- A. break B. case C. basic D. default

11-12. 다음은 속력 50으로 1초 주행 후 컬러센서로 주변광을 측정하는 코드이다.

```
#pragma config(Sensor, S3, cs, sensorEV3_Color)
#pragma config(Motor, motorB, lm, tmotorEV3_Large, PIDControl, encoder)
#pragma config(Motor, motorC, rm, tmotorEV3_Large, PIDControl, encoder)

int search(int speed, int time);

task main()
{
    int lightValue;
    lightValue = search(50, [11]_____);

    displayBigTextLine(5, "Reflected light value : %d", lightValue);
    sleep(5000);
}

int search(int speed, int time)
{
    setMotorSpeed(lm,speed);
    setMotorSpeed(rm,speed);
    sleep(time);
    setMotorSpeed(lm,0);
    setMotorSpeed(rm,0);

    [12]_____return getColorReflected(cs);
}
```

11. search 함수로 두 번째 매개변수에 들어갈 값은?

--- (1000)

12. 측정된 주변광의 세기를 되돌려주는 키워드는 ?

- | | |
|-----------|-------------|
| A. break | B. continue |
| C. return | D. sleep |

13-14. 다음은 2-차원 배열 data를 정의하고 출력하는 코드이다.

```
task main()
{
    int data[5][5];
    int i, j;

    for(i=0;i<5;i++)
        for(j=0;j<5;j++)
            data[i][j] = [13]_____i+j; //

    for(i=0;i<5;i++)
        for(j=0;j<5;j++)
            [14]_____
            displayStringAt((j+1)*10, 100-(i+1)*10, "%d ", data[i][j]);

    sleep(5000);
}
```

| 행\열 | 0 | 1 | 2 | 3 | 4 |
|-----|---|---|---|---|---|
| 0 | 0 | 1 | 2 | 3 | 4 |
| 1 | 1 | 2 | 3 | 4 | 5 |
| 2 | 2 | 3 | 4 | 5 | 6 |
| 3 | 3 | 4 | 5 | 6 | 7 |
| 4 | 4 | 5 | 6 | 7 | 8 |

그림 5 X 5 배열

13. 우측 그림과 같은 5X5 배열로 만들기 위한 코드는?

--- (**i + j**)

14. 디스플레이에 위의 배열 형태로 출력하는 코드는 ?

- A. displayStringAt((i+1)*10, 100 - (j+1)*10, "%d ", data[i][j]);
- B. displayStringAt((i+1)*10, 100 + (j+1)*10, "%d ", data[i][j]);
- C. **displayStringAt((j+1)*10, 100 - (i+1)*10, "%d ", data[i][j]);**
- D. displayStringAt((j+1)*10, 100 + (i+1)*10, "%d ", data[i][j]);

15. 다음은 자이로센서를 이용하여 직진을 유지하는 코드이다.

```
#pragma config(Sensor, S2, gs, sensorEV3_Gyro)
#pragma config(Motor, motorB, lm, tmotorEV3_Large, PIDControl, driveLeft, encoder)
#pragma config(Motor, motorC, rm, tmotorEV3_Large, PIDControl, driveRight, encoder)
task main()
{
    int speed_l, speed_r, curval;
    int initval = getGyroDegrees(gs); // initial gs value
    while(1){
        curval= getGyroDegrees(gs)-initval;

        speed_l=15 [15.A]_____ - curval;
        speed_r=15 [15.B]_____ + curval;

        setMotorSpeed(lm, speed_l);
        setMotorSpeed(rm, speed_r);
    }
}
```

● 자이로세서로 방향 변화를 측정해서 직진 주행을 유지하기 위한 방법은 ?
([15.A], [15.B])

- | | |
|------------------------|------------------------|
| A. - curval , - curval | B. + curval , - curval |
| C. - curval , + curval | D. + curval , + curval |

16. 다음은 모터 동기화를 이용한 주행 코드이다.

```
#pragma config(Motor, motorB, lm, tmotorEV3_Large, PIDControl, driveLeft, encoder)
#pragma config(Motor, motorC, rm, tmotorEV3_Large, PIDControl, driveRight, encoder)

task main()
{
    setMotorSyncEncoder(lm, rm, [16]_____ , 1000, 30);
    sleep(5000);
}
```

● 왼쪽 모터의 엔코더 값이 1000이 될 때까지 왼쪽 모터에만 출력 30을 유지하려면?

- | | | | |
|------|-------|--------|---------|
| A. 0 | B. 50 | C. 100 | D. -100 |
|------|-------|--------|---------|

- 17-20. 다음은 바닥에 설치된 신호등 색을 감지하면서 EV3를 주행 시키는 코드이다.
EV3가 자율 주행하며 동시에 라이브러리 함수를 이용한 데이터 로깅을 수행하는 코드이다.
데이터로깅은 파일에 3열로 "시간, 색코드, 속도"를 50ms 마다 저장한다.

```
#pragma config(Sensor, S1, ts,          sensorEV3_Touch)
#pragma config(Sensor, S2, gs,          sensorEV3_Gyro, modeEV3Gyro_Rate)
#pragma config(Sensor, S3, cs,          sensorEV3_Color, modeEV3Color_Color)
#pragma config(Sensor, S4, us,          sensorEV3_Ultrasonic)
#pragma config(Motor,      motorB, lm, tmotorEV3_Large, PIDControl, driveLeft, encoder)
#pragma config(Motor,      motorC, rm, tmotorEV3_Large, PIDControl, driveRight, encoder)

#define None 0
#define Black 1
#define Blue 2
#define Green 3
#define Yellow 4
#define Red 5
#define White 6
#define Brown 7

int color, finish_line;
int speed = 30;
int start_val = 0;

void go(int s)
{
    setMotorSpeed(lm, s);
    setMotorSpeed(rm, s);
}

void putData(int color, int speed)
{
    int t = [17]_____time1[T1]; // integer in msec
    datalogAddShort(0, t);
    datalogAddShort(1, color);
    datalogAddShort(2, speed);
    sleep(50); // sampling time : 50 ms
}
```

```

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

    [18]_____clearTimer(T1); // initialization of timer

    datalogOpen(0, 3, false); // time, color, speed
    while(1)
    {
        color = getColorName(cs);
        go(speed);

        if(!start_val && (color != White))
        {
            [19]_____finish_line = color; // match the start and finish
            setLEDColour(ledGreen);
            //sleep(700);
            for(int i=0;i<700/50;i++)
            {
                putData(color, speed);
            }
            start_val++;
            color = getColorName(cs);
        }
        if(color == finish_line)
        {
            go(0);
            setLEDColour(ledRedPulse);
            playSound(soundDownwardTones);
            for(int i=0;i<500/50;i++)
            {
                putData(color, 0);
            }
            break;
        }
        if(color == Green)
        {
            playSound(soundBeepBeep);
            putData(color, speed);
            for(int i=0;i<700/50;i++)
            {
                putData(color, speed);
            }
        }
    }
}

```



```

if(color == Yellow)
{
    go(speed/2);
    putData(color, speed/2);
    while(color == Yellow)
    {
        color = getColorName(cs);
        go(speed/2);
        playSound(soundLowBuzzShort);
        //sleep(700);
        for(int i=0;i<700/50;i++)
        {
            putData(color, [20]_____speed/2);
        }
    }
    go(speed);
    putData(color, speed);
}

if(color == Red)
{
    go(0);
    for(int i=0;i<1000/50;i++)
    {
        putData(color, 0);
    }
    go(speed);
    for(int i=0;i<700/50;i++)
    {
        putData(color, speed);
    }
}

}

datalogClose();
}

```

[17] 초기화된 T1 타이머로부터 1/1000초 단위로 데이터를 저장하는 순간의 시간을 계산하는 코드는?

- A. `time1[T1]`
- B. `time10[T1]`
- C. `time100[T1]`
- D. `time1000[T1]`

[18] T1 타이머를 초기화시키는 함수는?

- A. `initTimer`
- B. `startTimer`
- C. `setTimer`
- D. `clearTimer`

[19] 바닥색인 흰색이 아닌 출발선 색과 종료선 색을 일치시키기 위한 변수는?

--- (`finish_line`)

[20] 노란색 구간을 통과할 때의 속력을 저장하기 위한 코드는 ?

--- (`speed/2`)

[Bonus, +1] 완주 후 `start_val`의 값은?

--- (`1`)