

Melody timer

- Debugging
- Debug environments
- Debug via serial

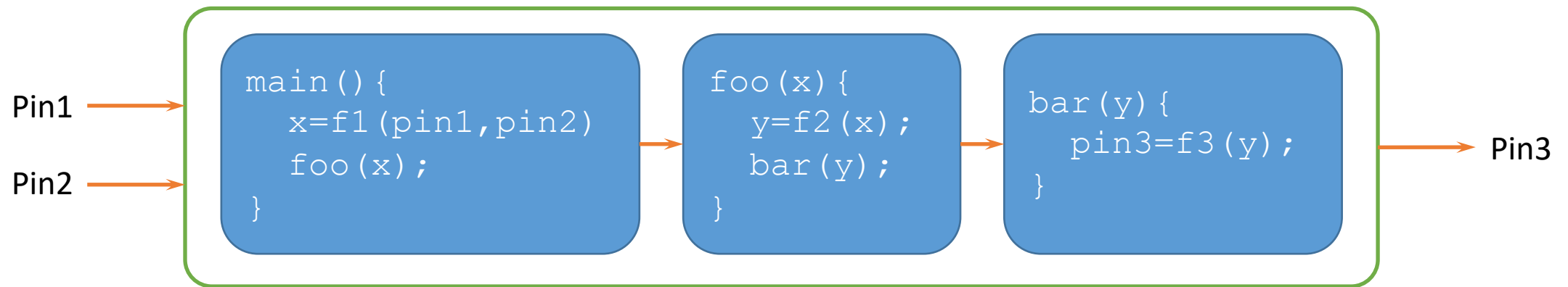
Debugging

Debug and trace

- Controllability and observability are required
- Controllability
 - Ability to control sources of data used by the system
 - Input pins, input interfaces (serial, Ethernet, etc.)
 - Registers and internal memory
- Observability
 - Ability to observe intermediate and final results
 - Output pins, output interfaces
 - Registers and internal memory

I/O access is insufficient

- Observation of I/O is not enough to debug



- If Pin2 is incorrect, how do we locate the bug?

Properties of a debugging environment

- Run control of the target
 - Start and stop the program execution
 - Observe data at stop points
- Real-time monitoring of target execution
 - Non-intrusive in terms of performance
- Timing and functional accuracy
 - Debugged system should ac like the real system

Arduino 실습

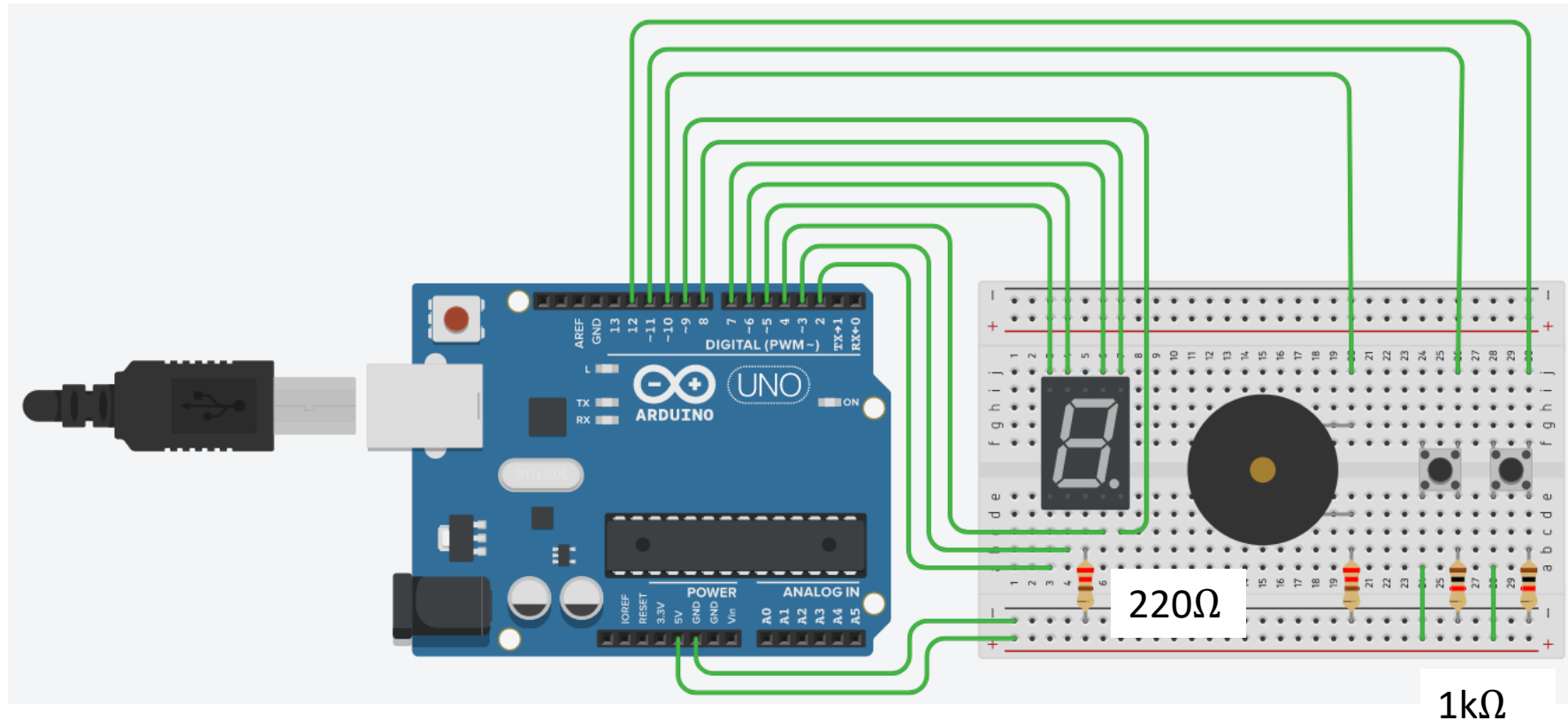
실습 설명

- 스위치를 누르면 lampStatus가 증가하면서 LED가 한 개씩 켜짐
- 4개가 모두 켜진 상태에서 스위치를 누르면 LED가 모두 꺼짐

준비물

- 아두이노 UNO 보드 1개
- 7-세그먼트 1개
- 점퍼선
- 부저 1개
- 스위치 2개
- 220Ω 저항 2개
- $1k\Omega$ 저항 2개

Arduino 배선



실습 코드

```
1 // ABCDEFG: 7843265:
2
3 int timerMode; // 0: Ready, timer set
4               // 1: Timer run
5 int modeButtonStatus;
6 int prevModeButtonStatus;
7 int addButtonStatus;
8 int prevAddButtonStatus;
9 int setTime;
10
11 int numValue[10][7] = {
12 // 2 3 4 5 6 7 8
13 // E D C G F A B
14 {1,1,1,0,1,1,1}, // 0: ABCDEF
15 {0,0,1,0,0,0,1}, // 1: BC
16 {1,1,0,1,0,1,1}, // 2: ABDEG
17 {0,1,1,1,0,1,1}, // 3: ABCD G
18 {0,0,1,1,1,0,1}, // 4: BCFG
19 {0,1,1,1,1,1,0}, // 5: ACDFG
20 {1,1,1,1,1,0,0}, // 6: CDEFG
21 {0,0,1,0,0,1,1}, // 7: ABC
22 {1,1,1,1,1,1,1}, // 8: ABCDEFG
23 {0,0,1,1,1,1,1} // 9: ABCFG
24 };
25
26 int timerStatus = 0;
27
28 void setup()
29 {
30     int i;
```

```
31
32     for (i=10; i<=11; i++){
33         pinMode(i, INPUT);
34     }
35     for (i=2; i<=10; i++){
36         pinMode(i, OUTPUT);
37     }
38
39     timerMode = 0; // 0: Ready, timer set
40                 // 1: Timer run
41     modeButtonStatus = 0;
42     prevModeButtonStatus = 0;
43     addButtonStatus = 0;
44     prevAddButtonStatus = 0;
45     setTime = 0;
46
47 }
48
49 void loop()
50 {
51     modeButtonStatus = digitalRead(11);
52     addButtonStatus = digitalRead(12);
53     if (modeButtonStatus > prevModeButtonStatus){
54         tone(10,1760,5);
55         timerMode = !timerMode;
56         if (timerMode==0){
57             setTime = 0;
58         }
59         digitalWrite(9,timerMode);
60     }
```

실습 코드

```
61
62 if (timerMode==0){ // 0: Ready, timer set
63     if (addButtonStatus > prevAddButtonStatus){
64         tone(10,1760,5);
65         setTime += 1;
66         setTime %= 10;
67     }
68 }
69 else{ // 1: Timer run
70     delay(500);
71     digitalWrite(9,0);
72     delay(500);
73     digitalWrite(9,1);
74     setTime -= 1;
75     if (setTime==0){
76         timerMode = 0;
77         displayDigit(setTime);
78         digitalWrite(9,0);
79         playMelody();
80     }
81 }
82
83 displayDigit(setTime);
84
85 prevModeButtonStatus = modeButtonStatus;
86 prevAddButtonStatus = addButtonStatus;
87
88 }
89
```

```
90 void displayDigit(int status){
91     for (int i=0; i<7; i++){
92         digitalWrite(i+2,numValue[status][i]);
93     }
94 }
95
96 void playMelody(void){
97     for (int k=0; k<4; k++){
98         for (int i=0; i<2; i++){
99             tone(10,880,50);
100             delay(100);
101         }
102         delay(200);
103     }
104 }
```