Programming with Arduino for Digital Humanities

Kazushi Ohya
Tsurumi University

From spring/summer semester, 2012, we started to use Arduino in an introductory programming course for humanities students. In order to evaluate this attempt we need more time to observe students who will take the following intermediate and advanced courses. However, we noticed that Arduino has good potential to attract students' interest to physical phenomena and arouse their interest in a small result of programming. Humanities students are struggling to envision the existence of the computational world in their minds. Physical programming bringing interaction with the physical world seems to encourage students to feel a computational world as being real. In this poster, we will show a background to introduce Arduino and all course materials such as source codes and electric diagrams.

Difficulties in Programming for Humanities Students

Difficulties in teaching programming in an introductory course to humanities students have been to let students find interest or enjoyment in (1) symbol manipulation and (2) grouping tasks with functions or other similar units. In the case of Japan, students in humanities are usually untrained on intermediate mathematics such as differentiation, statics, and logarithm in high school. They are basically not good at symbol manipulation. And, humanities students apt to expect big results in programming. They are disappointed at small results from source codes in exercises. It is usually difficult to expect them to find interest in a small result from symbol manipulation in introductory programming.

As far as our experience goes, humanities students seem to struggle to envision existence of computational world in their mind. Typical topics of programming patterns appearing in structured programming such as assignment, iteration, condition, and flow control are not difficult for humanities students to understand the formats and abstract behaviors. The problem they face is to know the need of learning the patterns or the ways to use them as substantial components making up the whole code in actual programming.

In an introductory programming course, we had used Scratch in expectation of attracting students' interest by its visual effect. Visual effect brought by Scratch could be big stimuli for students to be satisfied. Actually, Scratch seemed to be succeeded in bringing out their interest in programming. However, on the other hand, Scratch seemed to cause them to be disappointed at the limitation of the software itself. Students tend to regard programs they made as a true ``toy program.'' And worse, although students showed their interest in making programs, they did not seem to be interested in symbol manipulation and find enjoyment in small results.

Arduino

In order to respond to aforementioned difficulties or problems, we set two requirements to programming environment: (1) letting students be interested in something moving, and (2) letting students be satisfied with small results. Then, we decided to introduce embedded systems to an introductory programming course. We expected embedded systems to bring the real world into a scene of learning programming. Humanities students are used to rambling in abstract worlds except for numbers. And the present students have been immersed in digital or virtual environment from an early age. They are not easily satisfied with computational results on screens. On the other hand, interestingly, they show their interest to physical phenomena even if it is slight. They are very sensitive to physical stimuli. Embedded systems can be expected to let students be interested in something moving that may be small. And, it can contribute to facilitate students to envision computational world in their mind.

As an embedded system, we adopted Arduino which provides a good developing environment: (1) there is no need to prepare a device to install native codes to ICs, (2) there is an easy IDE based on Processing, and (3) the price of Arduino is affordable. A concern we noticed about introducing Arduino was that students have to learn about electricity to some extent. It is an unfamiliar subject to humanities students. The IDE for Arduino on Processing provides simple descriptive rules reminiscent of BASIC, and easy to view and write grouping tasks in methods. We expected this feature for students to be easy to concentrate on finding categories of processes and making groups with functions.

Course Design

One semester consists of 15 classes, which are lectures on, practices of programming, and an examination. The practices consist of four parts or themes: LED handling, variable registor or vol., sound handling, and binary display. The details of the classes are as follows.

No.	Theme	Contents
1	Lecture 1	what is software, history, basic terms.
2	Lecture 2	types of programming languages, flowchart
3	Lecture 3	electricity, embedded systems
4	Lecture 4	Arduino, breadboar, electric components
5	LED 1-3	digitalWrite(), delay(), LED, method
6	LED 4-5	iteration(for), variable, data type, assignment
7	LED 6-8	iteration(while), condition, break, argument
8	VOL 1-3	Serial.print(), analogRead(), variable registor, cds cell

No.	Theme	Contents
9	VOL 4-5	map(), analogWrite(), String
10	VOL 6-7	structural programming
11	Sound 1-3	tone(), array
12	Sound 4-6	table
13	Binary Display 1-3	digit, bit operation, random()
14	Binary Display 4	condition, a variety of codes
15	examination	

We prepared course materials for modeling, but did not use them due to time constraints. In this course we devised materials on

(1) simple patterns of circuits,

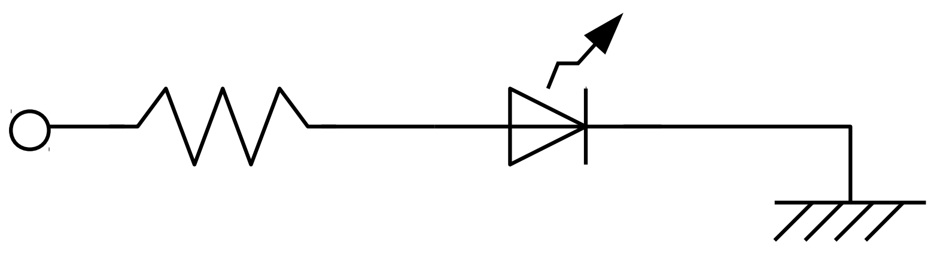
(2) learning the whole programming patterns with one LED circuit,

(3) using a variable resistor as an input device instead of sensors and as a game controller,

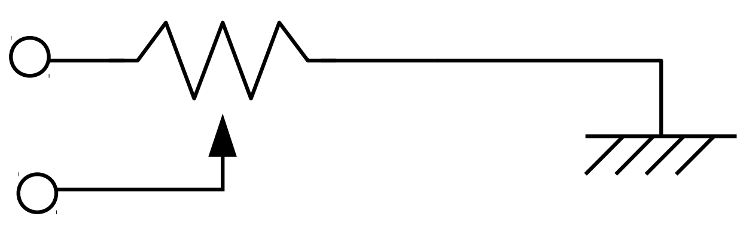
(4) learning arrays in handling sound, and

(5) learning binary digit with bit operation.

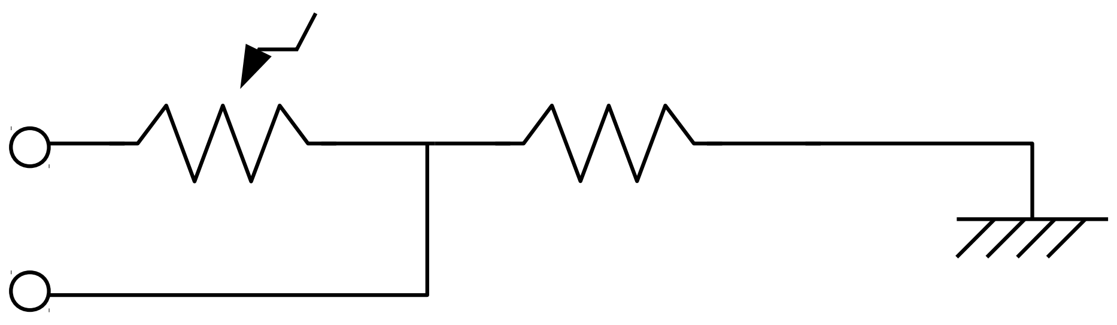
In all practices, we used only three patterns of circuits: (a) a series of a resistor and an LED, (b) a variable resistor, and (c) reading resistance values of a sensor. Using one circuit pattern of an LED, we taught the whole programming patterns with eight source codes. Although using multiple sensors with Arduino would be very attractive for students, there was not enough time to treat a topic of modeling, so we restricted use of sensors only to a light sensor. Instead, we used a variable resistor as an input device, which was also used as a game controller in learning structural programming. An array is not a difficult topic in programming, but it is not easy to make students understand the usefulness of the array. Arrays working as music scores to handle sound seemed to be a good example. Binary digit is a topic to learn in high school, but many humanities students do not understand it well and there would be less chance to see it as required knowledge in actual programming in the future. We think that even though a topic of binary digit could not be useful in Digital Humanities, students should learn it to feel or know the philosophy behind symbol manipulation. Thus, we used bit operation to control LEDs of a binary digit display, and provided a chance to learn binary digit.



(a) a series of a registor and an LED



(b) a variable resistor



(c) reading resistance values of a sensor (voltage divider)

Observation and Future

Fortunately many students seemed to enjoy programming with Arduino. At early stages, electric diagrams seemed to bewilder students, but these were not a hard obstacle to learn. Students became used to them after a few classes. Students seemed to find enjoyment from a small result in physical programming and grouping tasks. As we expected before this experiment, learning electricity seemed to be difficult for students. For example, Ohm's law was hard to be understood by students who learned it in junior high school. It might be possible to teach electric circuit like LEGO Block without any explanation about a theory or background knowledge. However, we believe students should know it, because learning theories in nature is inevitable in science education. We are planning to devise course materials to reduce the offset of learning electricity for next year(which means this year 2013. I prepare supplements to introduce this year's course with source codes and circuit diagrams.).

2013 version

```
//LED_1
void setup(){
  digitalWrite(2, HIGH);
  delay(2000);
  digitalWrite(2, LOW);
}
 void loop(){
 }

//LED_2
void setup(){
 light();
}

void light(){
  digitalWrite(2, HIGH);
  delay(2000);
  digitalWrite(2, LOW);
 }

 void loop(){
 }

//LED_3
void setup(){
  //light();
}

void light(){
  digitalWrite(2, HIGH);
  delay(2000);
  digitalWrite(2, LOW);
 }

void loop(){
  digitalWrite(2, HIGH);
  delay(2000);
  digitalWrite(2, LOW);
  delay(2000);
   }

//LED_4
void setup(){
  for(int i = 0; i < 3; i++){
    light();
  }
}
 void light(){
  digitalWrite(2, HIGH);
  delay(2000);
  digitalWrite(2, LOW);
  delay(2000);
 }

 void loop(){
}

//LED_5
void setup(){
  int limit = 3;
  while(limit > 0){
    light();
    limit--;
  }
}

void light(){
  digitalWrite(2, HIGH);
  delay(2000);
  digitalWrite(2, LOW);
  delay(2000);
 }
 void loop(){
}

//LED_6
void setup(){
  while(true){
     light();
  }
}

void light(){
  digitalWrite(2, HIGH);
  delay(2000);
  digitalWrite(2, LOW);
  delay(2000);
 }
 void loop(){
}

//LED_7
void setup(){
  int limit = 3;
  while(true){
     if(limit > 0){
       light();
     }else{
       break;
     }
     limit--;
  }
}

void light(){
  digitalWrite(2, HIGH);
  delay(2000);
  digitalWrite(2, LOW);
  delay(2000);
 }
 void loop(){
}

//LED_8
void setup(){
  int limit = 3;
  while(limit > 0){
    if(limit == 3){
       light(limit);
     }else if(limit == 2){
       light(limit);
     }else if(limit == 1){
       light(limit);
     };
     limit--;
     delay(1000);
  }//while
}//setup()

void light(int x){
  digitalWrite(2, HIGH);
  delay(x * 500);
  digitalWrite(2, LOW);
  delay(x * 500);
 }
 void loop(){
}

//VOL_1
void setup(){
  Serial.begin(9600);
}
void loop(){
  int value = analogRead(0);
  Serial.println(value);
  delay(2000);
}

//VOL_2
void setup(){
}
void loop(){
  int value = analogRead(0);
  Serial.println(value);
  int out = map(value, 0, 1023, 0, 255);
  analogWrite(3, out);
}

//VOL_3
String head = "";
boolean flag = true;
int counter = 0;
int preOut = -1;

void setup(){
  Serial.begin(9600);
  ready();
  start();
}

void ready(){
  head = "Are you ready?";
  Serial.println(head);
  delay(2000);
  for(int i = 5; i > 0; i--){
    Serial.println(i);
    delay(1000);
  }//for
  head = "Start!";
  Serial.println(head);
  delay(1000);
}

 void start(){
  while(flag){
    int value = analogRead(0);
    int out = map(value, 0, 1023, 0, 255);
    Serial.println(out);
    analogWrite(3, out);

    if(preOut < out ){
      counter++;
      head = "OK! Next!";
      Serial.println(head);
      preOut = out;
    }else{
      head = "Your score is ";
      Serial.print(head);
      Serial.println(counter);
      flag = false;
    }//if
      delay(1000);
  }//while
}
void loop(){
}

//VOL_4
int counter = 0;
String head = "";
boolean flag = true;
void setup(){
  Serial.begin(9600);
  ready();
  start();
}

void ready(){
  head = "Are you ready?";
  Serial.println(head);
  delay(1000);
  for(int i = 5; i > 0; i--){
     Serial.println(i);
     delay(1000);
  }
  head = "Start!";
  Serial.println(head);
  delay(1000);
}

 void start(){
   while(flag){
     int value = analogRead(0);
     int out = map(value, 0, 1023, 0, 255);
     Serial.println(out);

    if(out >=  (counter * 50)){
      counter++;
      digitalWrite(4, HIGH);
      head = "OK!";
      Serial.println(head);
     }else{
       counter = 0;
       head = "Ooops!";
       Serial.println(head);
       flag =false;
     }
     delay(1000);
     digitalWrite(4, LOW);
   }
 }
void loop(){
}

//VOL_5
int counter = 1;
String head = "";
boolean flag = true;

void setup(){
  Serial.begin(9600);
  ready();
  start();
}//setup()

void ready(){
  head = "Are you ready?";
  Serial.println(head);
  delay(2000);
  for(int i = 5; i > 0; i--){
    Serial.println(i);
    delay(1000);
  }//for
  head = "Start!!";
  Serial.println(head);
  delay(1000);
}//ready()

void start(){
  while(flag){
    int value = analogRead(0);
    int out = map(value, 0, 1023, 0, 255);
    Serial.println(out);
    analogWrite(3, out);

    if( ((counter-1)*50 <= out) && (out <
(counter*50)) ){
      counter++;
      head = "OK! Next!!";
      Serial.println(head);
      digitalWrite(4, HIGH);
    } else {
      counter = 1;
      head = "Wooops! Try again!!";
      Serial.println(head);
    }//if
    delay(1000);
    digitalWrite(4, LOW);

    if( (out > 250) && (counter > 1)){
      head = "Your Mission completed!!";
      Serial.println(head);
      flag = false;
    } else if( out > 250 ){
      head = "END! See you Next Time!";
      Serial.println(head);
      flag = false;
    }//if
  }//while
}//start()
```

```
void loop(){
}

//SP1
void setup(){
  tone(10, 440);
  delay(1000);
  noTone(10);
}
void loop(){
}

//SP2
void setup(){
  tone(10,262);
  delay(500);
  noTone(10);
  delay(10);
  tone(10,262);
  delay(500);
  noTone(10);
  delay(10);
  tone(10, 392);
  delay(500);
  noTone(10);
  delay(10);
  tone(10, 392);
  delay(500);
  noTone(10);
  delay(10);
  tone(10,400);
  delay(500);
  noTone(10);
  delay(10);
  tone(10,400);
  delay(500);
  noTone(10);
  delay(10);
  tone(10, 392);
  delay(500);
  noTone(10);
  delay(10);
}
void loop(){
}

//SP3
void setup(){
  play(262,1);
  play(262,1);
  play(392,1);
  play(392,1);
  play(440,1);
  play(440,1);
  play(392,2);
}

void play(int f, int d){
  tone(10, f);
  delay(f*500);
  noTone(10);
  delay(10);
```

```
}
void loop(){ }

//SP4
int C = 262;
int D = 294;
int E = 330;
int F = 349;
int G = 392;
int A = 440;
int H = 492;
int C5 = 525;

void setup(){
  play(C,1);
  play(C,1);
  play(G,1);
  play(G,1);
  play(A,1);
  play(A,1);
  play(G,2);
  play(F,1);
  play(F,1);
  play(E,1);
  play(E,1);
  play(D,1);
  play(D,1);
  play(C,2);
}
void play(int f, int d){
  tone(10, f);
  delay(f*500);
  noTone(10);
  delay(10);
}
void loop(){
}

//SP5
int C = 262;
int D = 294;
int E = 330;
int F = 349;
int G = 392;
int A = 440;
int H = 492;
int C5 = 525;

void setup(){
  part1();
  part2();
  part1();
}

void part1(){
  play(C,1);
  play(C,1);
  play(G,1);
  play(G,1);
  play(A,1);
  play(A,1);
  play(G,2);
  play(F,1);
  play(F,1);
  play(E,1);
  play(E,1);
  play(D,1);
  play(D,1);
  play(C,2);
}

void part2(){
  play(G,1);
  play(G,1);
  play(F,1);
  play(F,1);
  play(E,1);
  play(E,1);
  play(D,2);
  play(G,1);
  play(G,1);
  play(F,1);
  play(F,1);
  play(E,1);
  play(E,1);
  play(D,2);
}
void play(int f, int d){
  tone(10, f);
  delay(f*500);
  noTone(10);
  delay(10);
}
void loop(){ }

//SP6
#define H3 247
#define C 262
#define Cis 277
#define D 294
#define Dis 311
#define E 330
#define F4 349
#define F 349
#define Fis 370
#define G 392
#define Gis 415
#define A 440
#define Ais 466
#define H 494
#define C5 523

void setup(){
  part1();
  part2();
  part1();
  }

void part1(){
  int notes[]=
{C,C,G,G,A,A,G,F4,F4,E,E,D,D,C};
  int durs[] = {1,1,1,1,1,1,2, 1,
1,1,1,1,1,2};
```

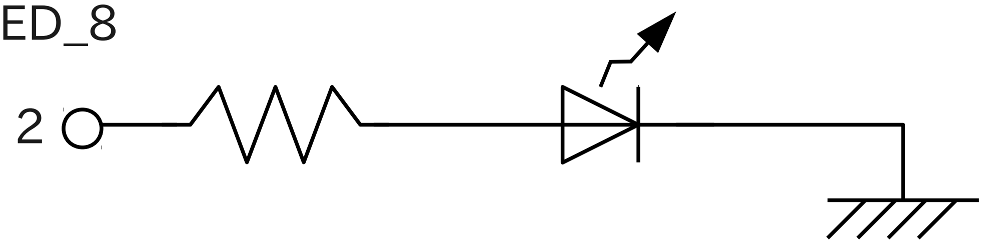
```
  int length =
sizeof(notes)/sizeof(notes[0]);
   for(int i = 0; i < length; i++){
    play(notes[i], durs[i]);
  }//for(i)
}
void part2(){
  int notes[] =
{G,G,F4,F4,E,E,D,G,G,F4,F4,E,E,D};
  int durs[]  = {1,1, 1, 1,1,1,2,1,1, 1, 1,
1,1,2};
  int length =
sizeof(notes)/sizeof(notes[0]);
   for(int i = 0; i < length; i++){
    play(notes[i], durs[i]);
  }//for(i)
}

void play(int f, int d){
  tone(10, f);
  delay(500*d);
  noTone(10);
  delay(10);
}

void loop(){
}

//SP7
#define H3 247
#define C 262
#define Cis 277
#define D 294
#define Dis 311
#define E 330
#define F4 349
#define F 349
#define Fis 370
#define G 392
#define Gis 415
#define A 440
#define Ais 466
#define H 494
#define C5 523

void setup(){
  part1();
  part2();
  part1();
}

void part1(){
  int notes[]=
{C,C,G,G,A,A,G,F4,F4,E,E,D,D,C};
  int durs[] = {1,1,1,1,1,1,2, 1,
1,1,1,1,1,2};
  int length =
sizeof(notes)/sizeof(notes[0]);
  playTheScore(notes, durs, length);
}
```

```
void part2(){
  int notes[] =
{G,G,F4,F4,E,E,D,G,G,F4,F4,E,E,D};
  int durs[]  = {1,1, 1, 1,1,1,2,1,1, 1, 1,
1,1,2};
  int length =
sizeof(notes)/sizeof(notes[0]);
  playTheScore(notes, durs, length);
}

void playTheScore(int notes[], int durs[],
int length){
  for(int i = 0; i < length; i++){
    play(notes[i], durs[i]);
  }//for(i)
}

void play(int f, int d){
  tone(10, f);
  delay(500*d);
  noTone(10);
  delay(10);
}

//SP8
#define H3 247
#define C 262
#define Cis 277
#define D 294
#define Dis 311
#define E 330
#define F4 349
#define F 349
#define Fis 370
#define G 392
#define Gis 415
#define A 440
#define Ais 466
#define H 494
#define C5 523

void setup(){
  for(int i = 0; i < 5; i++){
   randomSeed(analogRead(0));
   int num = random(0,3);
   if(num == 0){
     part1();
   }else if(num == 1){
     part2();
   }else if(num == 2){
     part3();
   }//if
  }//for
  play(C,2);
}

void part1(){
 play(C,1);
 play(E,1);
 play(G,1);
 play(E,1);
}
```

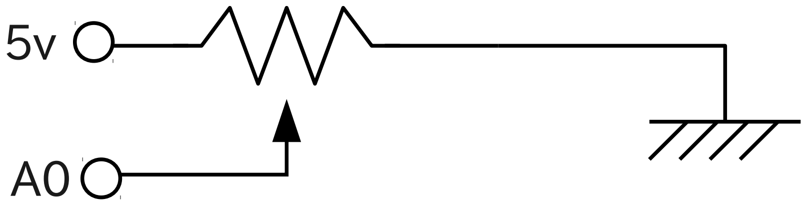
```
void part2(){
 play(C,1);
 play(F4,1);
 play(A,1);
 play(F4,1);
}

void part3(){
 play(C,1);
 play(C5,1);
 play(H,1);
 play(G,1);
}

void play(int f, int d){
  tone(10, f);
  delay(500*d);
  noTone(10);
  delay(10);
}
void loop(){
}

//BIN_1
void setup(){
  Serial.begin(9600);
  show(7);
}
 void show(int num){
  Serial.println(num);
  digitalWrite(2, num & B0001);
  digitalWrite(4, num & B0010);
  digitalWrite(6, num & B0100);
  digitalWrite(8, num & B1000);
}
void loop(){
}

//BIN_2
void setup(){
  Serial.begin(9600);
  randomSeed(analogRead(0));
  int result = (int) random(0, 16);
  show(result);
}
void show(int num){
  Serial.println(num);
  digitalWrite(2, num & B0001);
  digitalWrite(4, num & B0010);
  digitalWrite(6, num & B0100);
  digitalWrite(8, num & B1000);
}
void loop(){
}

//BIN_3
void setup(){
  Serial.begin(9600);
  run();
}
```

```
void run(){
  while(true){
    int value = analogRead(0);
    int result = map(value, 0, 1023, 0, 15);
    show(result);
    delay(1000);
  }//while
}

void show(int num){
  Serial.println(num);
  digitalWrite(2, num & B0001);
  digitalWrite(4, num & B0010);
  digitalWrite(6, num & B0100);
  digitalWrite(8, num & B1000);
}
void loop(){  }

//BIN_4
void setup(){
  Serial.begin(9600);
  run();
}//setup()
void run(){
  while(true){
    int value = analogRead(0);
    int result = map(value, 0, 1023, 0, 4);
    //int result = map(value, 0, 1023, 4,
0); for cds cell
    Serial.println(result);
    check(result);
    delay(1000);
  }//while
}//run()

void check(int num){
  switch(num){
  case 0:
    show(0);
    break;
  case 1:
    show(1);
    break;
  case 2:
    show(3);
    break;
  case 3:
    show(7);
    break;
  case 4:
    show(15);
    break;
  }
}//check()
void show(int num){
  digitalWrite(2, num & B0001);
  digitalWrite(4, num & B0010);
  digitalWrite(6, num & B0100);
  digitalWrite(8, num & B1000);
}//show()
void loop(){
}
```

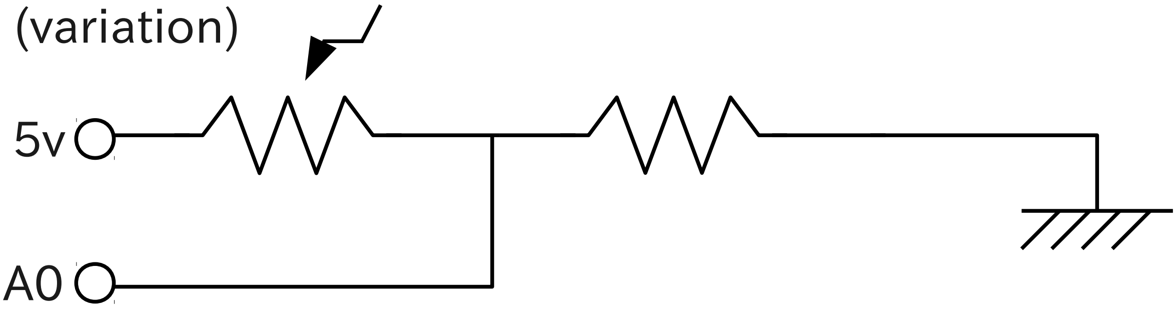
for LED_1, ... , LED_8



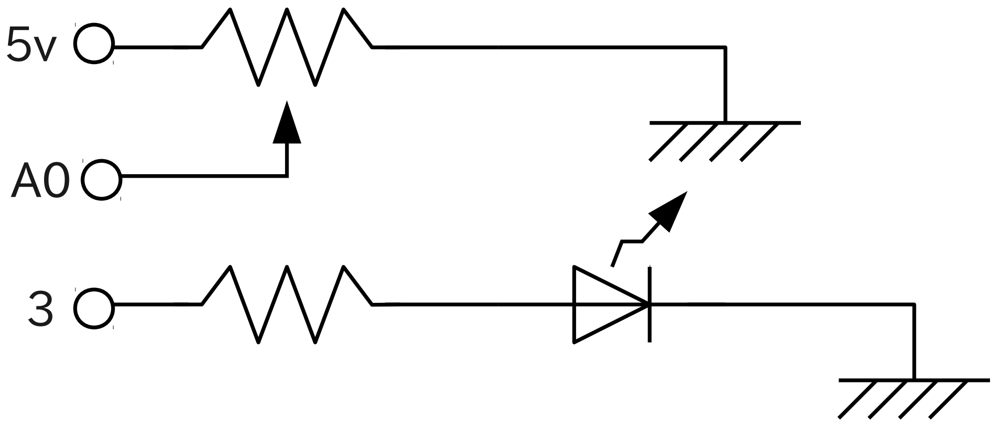
for VOL_1, 2



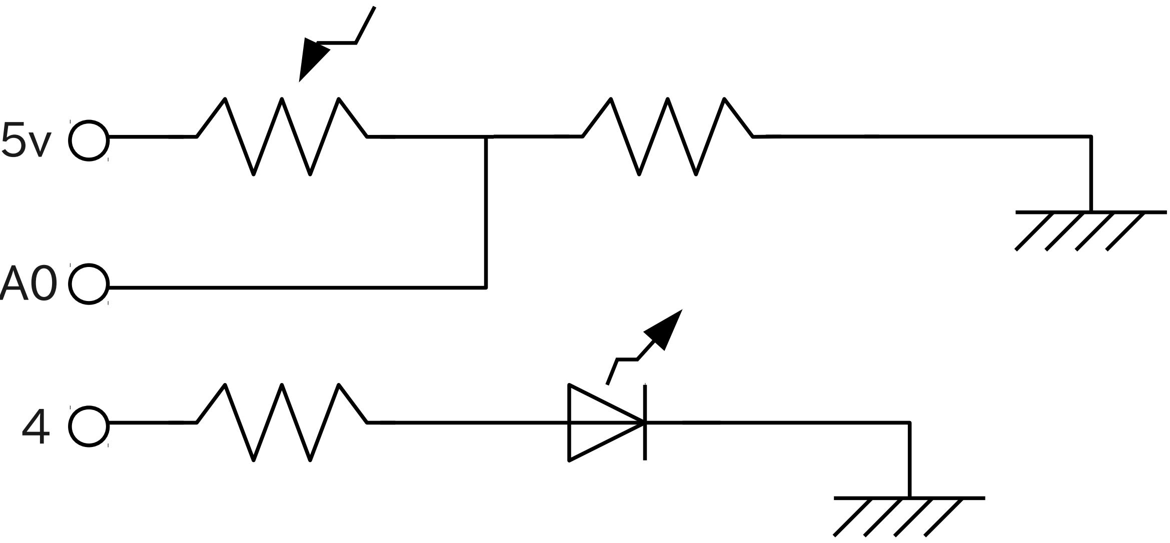
for VOL_1 (variation)



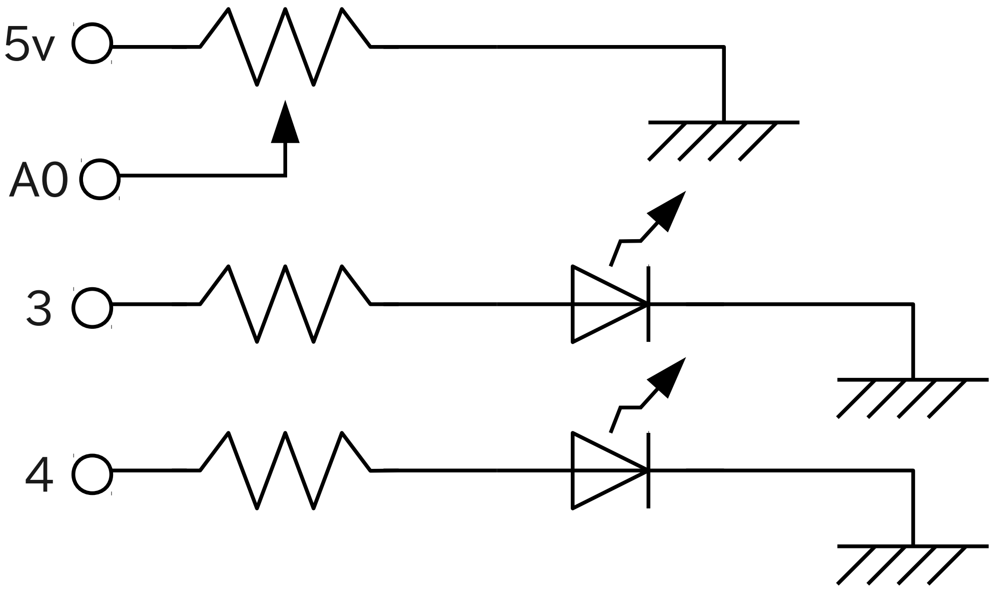
for VOL_3



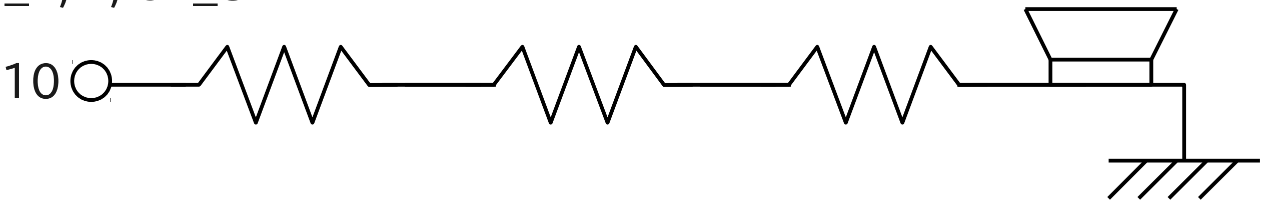
for VOL_4



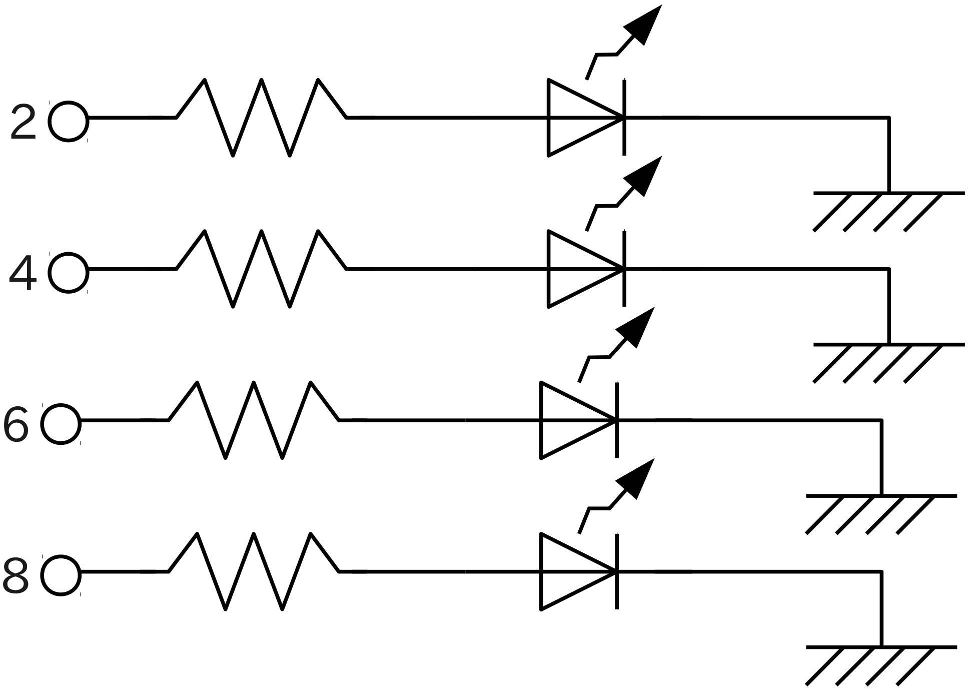
for VOL_5



for SP_1,..., SP_8



for BIN_1



add. for BIN_2,3,4