Arduino/Suwano をはじめよう@KYOTO-U　No.3-2

2013 年 5 月 16 日

京都大学 学術情報メディアセンター
喜多　一

今回の学習目標

- 音を出すプログラムの改良

１．　回路の想定

Ｎｏ．３で作成した回路とＮｏ．２で作成した回路とを合体させます。

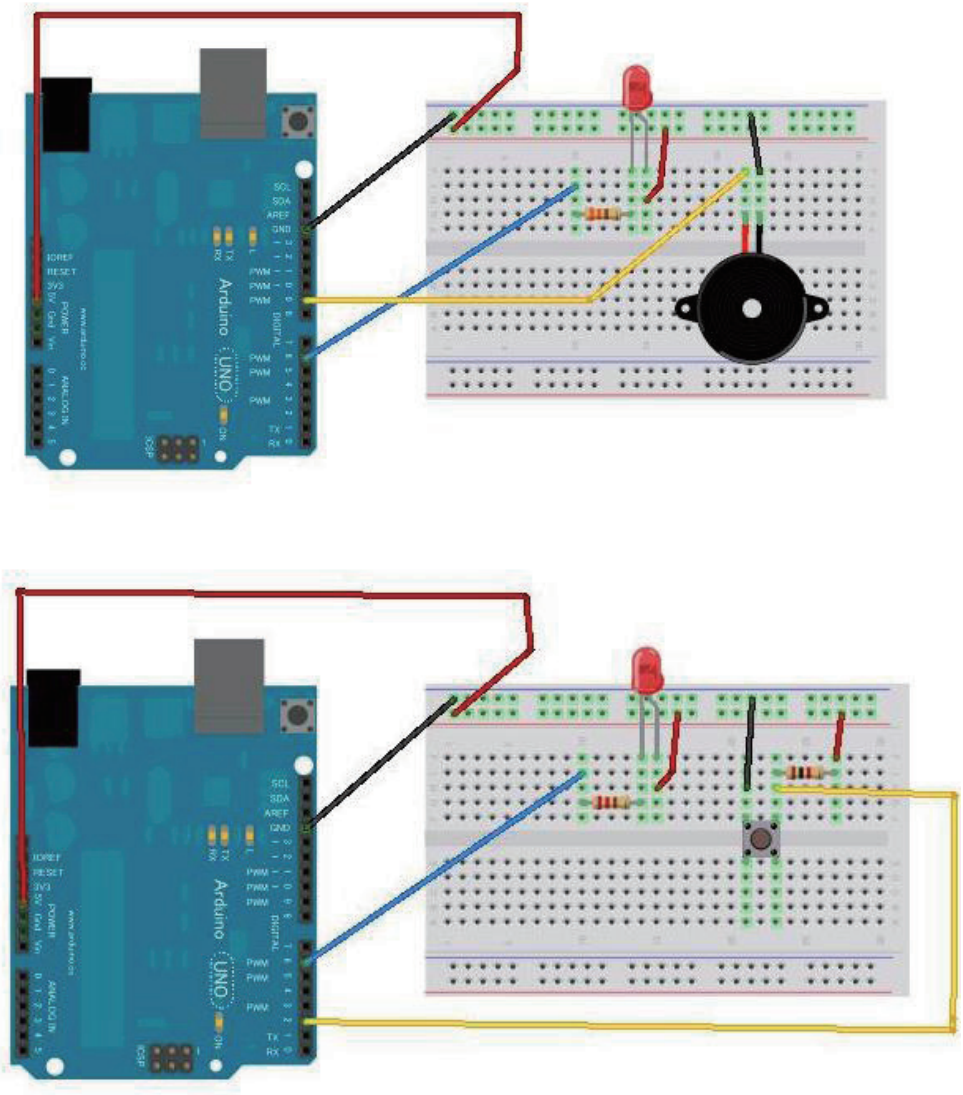

図 1 作成する回路（実体図）

Fritzingを使用して作成
http://fritzing.org/download/

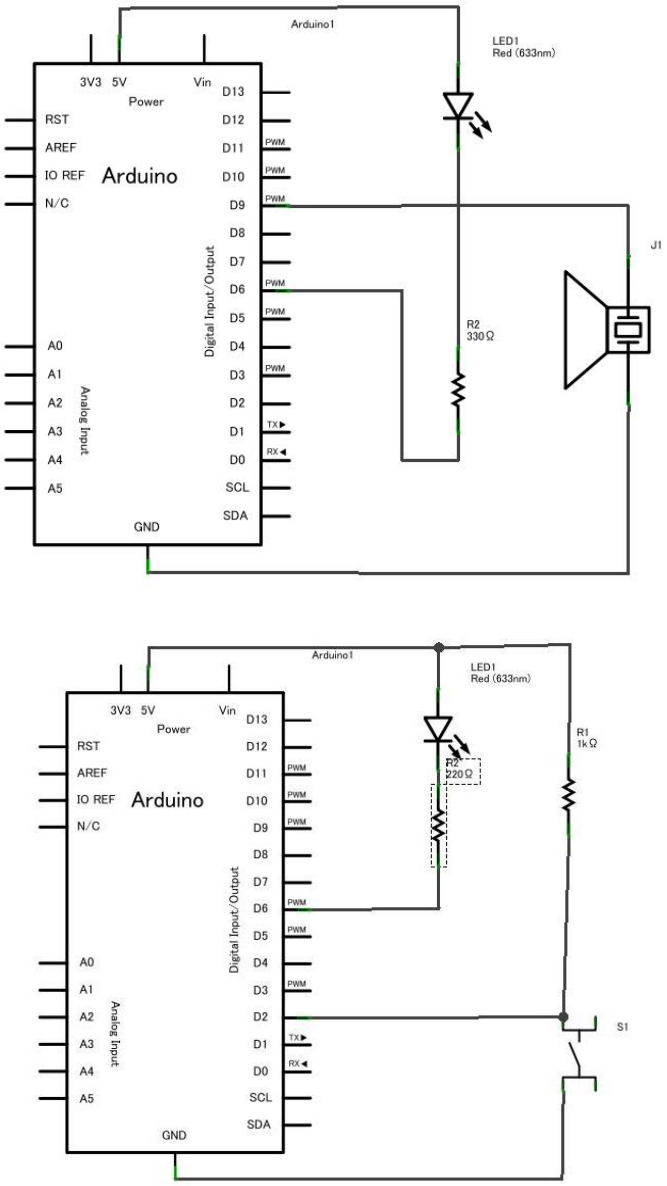

図 2　作成する回路（回路図）

２．　スイッチで開始、停止を行う

No.3 のプログラムと No.2, 2.2 のプログラムを合体させます。

行番号	ソースコード	説明
1	#define SPEAKER 9	9 番ピンを SPEAKER と,
2	#define LED 6	6 番ピンを LED とします.
3	#define BUTTON 2	2番ピンをBUTTON とします。
4	int freq[] =	初期値を持った配列の宣言
5	{440,466,494,523,554,587,622,659,698,740,	音階に相当する周波数
6	784,831,880,932,988,1047,1109,1175,1245,	
7	1319,1397,1480,1568,1661,1760};	
8	int TONE_MIN = 0;	音階の添え字の最小値
9	int TONE_MAX = 24;	音階の添え字の最大値
10	int melody[] = {3,5,7,8,10,12,14,15,-1};	曲の音程.
11	int M_LENGTH = 9;	曲の長さ
12	int mIndex = 0;	曲の場所を指す変数
13	int DURATION = 750;	１音の長さ(750ms)
14	int val = 1;	スイッチの値を保存
15	int oldVal = 1;	
16	int soundOn = 0;	音を出すかどうかの変数
17	void rewind(void);	関数 rewind() と
18	void makeNextTone(void);	makeNextTone() のプロトタイプ宣言
19	void makeNextTone() {	
20	int fIndex = melody[mIndex];	次の音を出す関数
21	if ((fIndex>=TONE_MIN)&&(fIndex<TONE_MAX)){	音程を取り出す
22	tone(SPEAKER,freq[fIndex],DURATION);	音階の範囲にあれば
23	delay(DURATION);	tone 関数で音を出す
24	} else {	音の長さだけ待つ
25	digitalWrite(LED,LOW);	
26	delay(DURATION);	そうでなきえればLED を点灯
27	digitalWrite(LED,HIGH);	750msec 待って
28	}	LED を消灯
29	mIndex = mIndex + 1;	
30	if (mIndex >= M_LENGTH) {	曲の場所を進める
31	mIndex = 0;	範囲に達したら
32	}	最初にもどす

```
33  }
34
35  void rewind() {
36    mIndex = 0;                          曲の先頭にもどる関数
37  }
38
39  void setup() {
40    pinMode(SPEAKER, OUTPUT);
41    pinMode(LED, OUTPUT);                ピンの設定
42    pinMode(BUTTON, INPUT)
43    digitalWrite(LED,HIGH);
44    rewind();                            LED を消灯
45    soundOn = 0;                         曲の先頭に
46  }                                      曲は停止状態
47
48  void loop() {
49    val = digitalRead(BUTTON);
50    if ((val == LOW) && (oldVal == HIGH)) {   スイッチを読む
51      if (soundOn == 0) {                押された瞬間なら soundOn
52        soundOn = 1;                     の値を切替
53      } else {
54        soundOn = 0;
55      }
56    }
57    if (soundOn == 1) {
58      makeNextTone();                    もし音を出すモードなら
59    } else {                             次の音を出す
60      Delay(10);                         そうでなければ
61    }                                    10msec 待つ
62    oldVal = val;
63  }                                      val の値を oldVal に保存
```

このプログラムは動作するが１つの音を発生中にボタンをおしても応答しない。これを改善したものが次のプログラムである。

プログラム改良版

行番号	ソースコード	説明
1	#define SPEAKER 9	9番ピンを SPEAKER と,
2	#define LED 6	6 番ピンを LED とします.
3	#define BUTTON 2	2番ピンをBUTTON とします。
4	int freq[] =	初期値を持った配列の宣言
5	{440,466,494,523,554,587,622,659,698,740,	音階に相当する周波数
6	784,831,880,932,988,1047,1109,1175,1245,	
7	1319,1397,1480,1568,1661,1760};	
8	int TONE_MIN = 0;	音階の添え字の最小値
9	int TONE_MAX = 24;	音階の添え字の最大値
10	int melody[] = {3,5,7,8,10,12,14,15,-1};	曲の音程.
11	int M_LENGTH = 9;	曲の長さ
12	int mIndex = 0;	曲の場所を指す変数
13	int DURATION = 750;	１音の長さ(750ms)
14	int val = 1;	スイッチの値を保存
15	int oldVal = 1;	
16	int soundOn = 0;	音を出すかどうかの変数
17	int remain = 0;	発声の残り時間
18	void rewind(void);	関数 rewind() と
19	void makeNextTone(void);	makeNextTone() のプロトタイプ宣言
20	void makeNextTone() {	次音を出す関数（待たない）
21	int fIndex = melody[mIndex];	音程を取り出す
22	if ((fIndex>=TONE_MIN)&&(fIndex<TONE_MAX)){	音階の範囲にあれば
23	tone(SPEAKER,freq[fIndex],DURATION);	tone 関数で音を出す
24	} else {	そうでなければLEDを点灯
25	digitalWrite(LED,LOW);	LED を点灯
26	}	
27	remain = DURATION;	残り時間を設定
28	mIndex = mIndex + 1;	曲の場所を進める
29	if (mIndex >= M_LENGTH) {	範囲に達したら
30	mIndex = 0;	最初にもどす
31	}	
32	}	
33		
34	void rewind() {	曲の先頭にもどる関数
35	mIndex = 0;	

```
36  }
37
38  void setup() {
39    pinMode(SPEAKER, OUTPUT);                    ピンの設定
40    pinMode(LED, OUTPUT);
41    pinMode(BUTTON, INPUT)
42    digitalWrite(LED,HIGH);
43    rewind();                                    LED を消灯
44    soundOn = 0;                                 曲の先頭に
45  }                                              曲は停止状態
46
47  void loop() {
48    val = digitalRead(BUTTON);
49    if ((val == LOW) && (oldVal == HIGH)) {      スイッチを読む
50      if (soundOn == 0) {                        押された瞬間なら soundOn
51        soundOn = 1;                             の値を切替
52      } else {
53        soundOn = 0;
54      }
55    }
56    if (remain >0) {
57      remain = remain -10;                       もし発音中なら
58      if (remain <=0) {                          残り時間を 10 減らして
59        digitalWrite(LED,HIGH);                  残り時間は 0 以下になれば
60      }                                          LED は消灯
61    }
62    if ((soundOn == 1)&&(remain<=0)) {
63        makeNextTone();                          音を出すモードで発音中で
64    }                                            なければ、次の音を出す
65    delay(10);
66    oldVal = val;                                10msec 待つ
67  }                                              val の値を oldVal に保存
```