

PIANO SHIELD

(ARDUINO COMPATIBLE)
PRODUCT CODE: M00270058

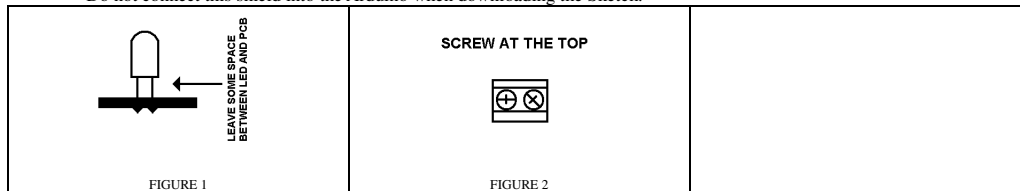
FEATURE:

- Eight piano keys
- Two function keys (The function is set by Sketch writer)
- Two LED (They work independent, the propose is set by Sketch writer).
- Speaker is provided.
- Assembly is needed.
- Arduino Sketch example showing simple piano is attached.
- Requires 1 Arduino UNO (not included).



READ BEFORE INSTALLATION:

- Put the component on the side of screen printing and solder on the back of PCB without printing.
- On component, longer leg is “+”.
- Do not put the LED to very bottom, just install as Figure 1.
- Do not connect this shield into the Arduino when downloading the Sketch.



DESCRIPTION:

The circuit design is based on the Arduino UNO. Of course, this can be used on any brand of Microcontroller or external circuit if the pin location is matched. If this is not matched, just route this yourself.
You can think that this is a Shield without soul. Only contains 12 keys, two LED, a speaker. Then just make what you want.
In the Sketch example, this has four ranges of tone. The change of range is by pushing the F1 and F2 on PCB.

INSTALLATION:

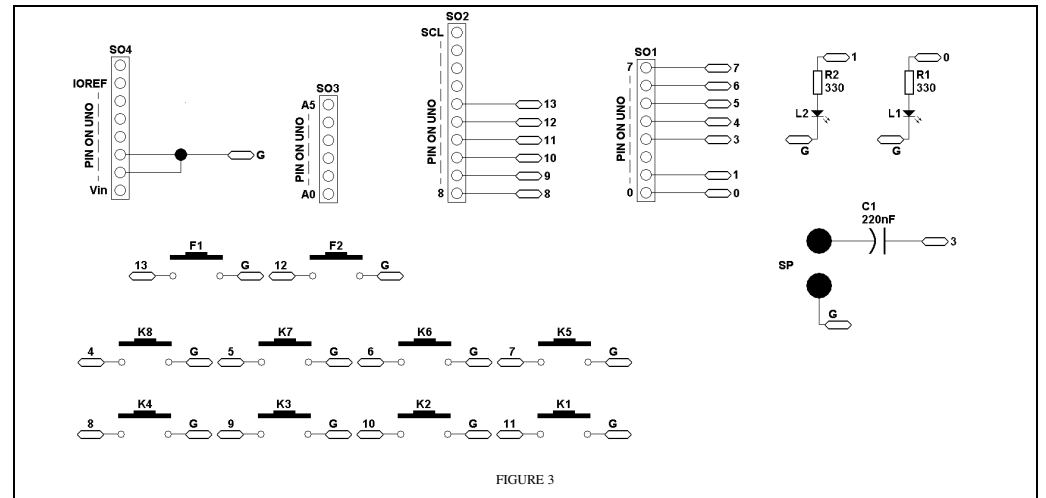
Just install the component to the PCB M00260094 according to below table

ITEM	SYMBOL ON PCB	DESCRIPTION	OUTLOOK	DIRECTION IS IMPORTANT?
1	R1	RESISTOR, 330ohms	ORANGE, ORANGE BROWN	NO
2	R2	RESISTOR, 330ohms	ORANGE, ORANGE BROWN	NO
3	SO1	STACKABLE HEADER – 8PIN	LONG 8 PIN	NO
4	SO2	STACKABLE HEADER – 10PIN	LONG 10 PIN	NO
5	SO3	STACKABLE HEADER – 6PIN	LONG 6 PIN	NO
6	SO4	STACKABLE HEADER – 8PIN	LONG 8 PIN	NO
7	K1	PUSH BUTTON SWITCH	FOUR LEGS	NO

8	K2	PUSH BUTTON SWITCH	FOUR LEGS	NO
9	K3	PUSH BUTTON SWITCH	FOUR LEGS	NO
10	K4	PUSH BUTTON SWITCH	FOUR LEGS	NO
11	K5	PUSH BUTTON SWITCH	FOUR LEGS	NO
12	K6	PUSH BUTTON SWITCH	FOUR LEGS	NO
13	K7	PUSH BUTTON SWITCH	FOUR LEGS	NO
14	K8	PUSH BUTTON SWITCH	FOUR LEGS	NO
15	F1	PUSH BUTTON SWITCH	FOUR LEGS	NO
16	F2	PUSH BUTTON SWITCH	FOUR LEGS	NO
17	SP	SOCKET FOR SPEAKER WIRE CONNECTION	FIGURE 2	YES
18	L1	LED	RED	YES
19	L2	LED	GREEN	YES
20	C1	CAPACITOR, 220uF	MARK WITH 220uF OR SAME MEANING OF VALUE	YES
21	/	HEX CAP SCREW AND IS THE LEG OF PIANO	LONG METAL	SEE BELOW
22	/	HEX CAP SCREW AND IS THE LEG OF PIANO	LONG METAL	SEE BELOW
23	/	HEX CAP SCREW AND IS THE LEG OF PIANO	LONG METAL	SEE BELOW
24	/	HEX CAP SCREW AND IS THE LEG OF PIANO	LONG METAL	SEE BELOW
25	/	HEX CAP SCREW AND IS THE LEG OF PIANO	LONG METAL	SEE BELOW
26	/	HEX NUTS, ON TOP OF ITEM 21	/	SEE BELOW
27	/	HEX NUTS, ON TOP OF ITEM 22	/	SEE BELOW
28	/	HEX NUTS, ON TOP OF ITEM 23	/	SEE BELOW
29	/	HEX NUTS, ON TOP OF ITEM 24	/	SEE BELOW
30	/	HEX NUTS, ON TOP OF ITEM 25	/	SEE BELOW
31	/	RED WIRE	RED WIRE	SEE BELOW
32	/	BLACK WIRE	BLACK WIRE	SEE BELOW
33	/	SPEAKER	/	SEE BELOW

- Put item 21 to 25 to the five holes on the edge of PCB. This is the leg of Piano. Then screw item 26 to 30 on the top of this.
- At one end, connect red and black wire to the “+” and “-” of SP, the other end is connected to the two terminal of speaker. Because there is only one speaker and is a mono system, you can ignore the polarity of speaker. In stereo system, polarity is important because you need to consider the interference of sound from two speakers.

CIRCUIT DIAGRAM:



SKETCH:

```
/* This Sketch show you four ranges of tone when pushing F1 and F2 for increasing and decreasing the range.
Of course, you can see this shield having no soul and write what you want.
For example, setting F1 and F2 as recording and playing of what you push to K1 to K8. */
```

```
/* Output pin of tone. */
int speaker = 3;
```

```
/* Pin number for 8 keys(K1 to K8). */
int sw[8]{11, 10, 9, 8, 7, 6, 5, 4};
```

```
/* Pin number for two function keys. */
int F1 = 13; int F2 = 12;
```

```
/* Pin number for two LED. */
int L1 = 0; int L2 = 1;
```

```
/* Four ranges of tone. */
int frequency1[8]; int frequency2[8];
int frequency3[8]; int frequency4[8];
```

```
/* Other variables using in the program */
int i; int val;
int push1; int push2;
int count = 0;
```

```
void setup() {
```

```
/* Set L1 and L2 as output. Then set F1 and F2 as input. */
pinMode(L1, OUTPUT); pinMode (L2, OUTPUT);
pinMode(F1, INPUT_PULLUP); pinMode(F2, INPUT_PULLUP);
```

```
/* Set K1 to K8 as input. */
for(sw[i]; i<8; i++) {
    pinMode(sw[i], INPUT_PULLUP);
}
```

```
}
```

```
/* Runing of program */
void loop() {
```

```
/* Decreasing the range of tone when pushing F1 */
push1 = digitalRead(F1);
if(push1 == 0) {
    delay(500);
    digitalWrite(L1, HIGH);
    delay(200);
    digitalWrite(L1, LOW);
    count = count - 1;
    if(count == -1) {
        count = 3;
    }
}
```

```
/* Increasing the range of tone when pushing F2 */
push2 = digitalRead(F2);
if(push2 == 0) {
    delay(500);
    digitalWrite(L2, HIGH);
    delay(200);
    digitalWrite(L2, LOW);
    count = count + 1;
    if(count == 4) {
        count = 0;
    }
}
```

```
/* First range of tone would give out when pushing K1 to K8. */
if(count == 0) {
    frequency1[0] = 262; frequency1[1] = 277; frequency1[2] = 294; frequency1[3] = 311;
    frequency1[4] = 330; frequency1[5] = 349; frequency1[6] = 370; frequency1[7] = 392;
    for(int i = 0; i<8; i++) {
        val = digitalRead(sw[i]);
        if(val == 0) {
            tone(speaker, frequency1[i], 100);
        }
    }
}
```

```
/* Second range of tone would give out when pushing K1 to K8. */
if(count == 1) {
    frequency2[0] = 415; frequency2[1] = 440; frequency2[2] = 466; frequency2[3] = 494;
    frequency2[4] = 523; frequency2[5] = 554; frequency2[6] = 587; frequency2[7] = 622;
    for(int i = 0; i<8; i++) {
        val = digitalRead(sw[i]);
        if(val == 0) {
            tone(speaker, frequency2[i], 100);
        }
    }
}
```

```
/* Third range of tone would give out when pushing K1 to K8. */
if(count == 2) {
    frequency3[0] = 659; frequency3[1] = 698; frequency3[2] = 740; frequency3[3] = 784;
    frequency3[4] = 831; frequency3[5] = 880; frequency3[6] = 932; frequency3[7] = 988;
    for(int i = 0; i<8; i++) {
        val = digitalRead(sw[i]);
        if(val == 0) {
            tone(speaker, frequency3[i], 100);
        }
    }
}
```

```
/* Fourth range of tone would give out when pushing K1 to K8. */
if(count == 3) {
    frequency4[0] = 1046; frequency4[1] = 1109; frequency4[2] = 1175; frequency4[3] = 1245;
    frequency4[4] = 1318; frequency4[5] = 1397; frequency4[6] = 1480; frequency4[7] = 1568;
    for(int i = 0; i<8; i++) {
        val = digitalRead(sw[i]);
        if(val == 0) {
            tone(speaker, frequency4[i], 100);
        }
    }
}
```