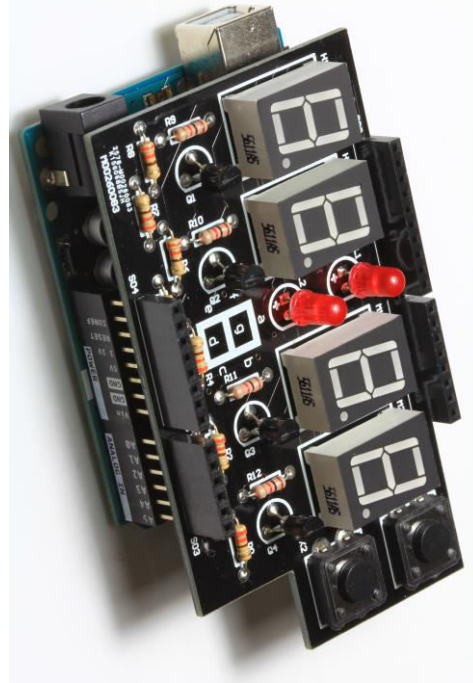


TRICKY CLOCK SHIELD

(ARDUINO COMPATIBLE)
PRODUCT CODE: M00270047

FEATURE:

- Being a gentle clock.
- Being a tricky clock, this can be run faster or slower than gentle clock at certain time. 25 hours (or any value) a days. 59 seconds (or any value) in one hour to make a trick to your friend.....
- Being a stop watch, Calendar, Powerball, dice gambling, Russian Roulette..... by writing your own Sketch.
- Assembly is needed.
- Arduino Sketch example for gentle clock is attached.
- Requires 1 Arduino UNO (not included).

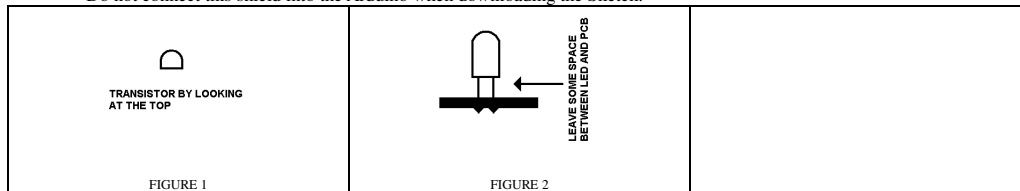


ITEM	SYMBOL ON PCB	DESCRIPTION	OUTLOOK	DIRECTION IS IMPORTANT?
1	R0	RESISTOR, 330 ohms	ORANGE, ORANGE BROWN	NO
2	R1	RESISTOR, 330 ohms	ORANGE, ORANGE BROWN	NO
3	R2	RESISTOR, 330 ohms	ORANGE, ORANGE BROWN	NO
4	R3	RESISTOR, 330 ohms	ORANGE, ORANGE BROWN	NO
5	R4	RESISTOR, 330 ohms	ORANGE, ORANGE BROWN	NO
6	R5	RESISTOR, 330 ohms	ORANGE, ORANGE BROWN	NO
7	R6	RESISTOR, 330 ohms	ORANGE, ORANGE BROWN	NO
8	R7	RESISTOR, 330 ohms	ORANGE, ORANGE BROWN	NO
9	R8	RESISTOR, 330 ohms	ORANGE, ORANGE BROWN	NO
10	R9	RESISTOR, 1K ohms	BROWN, BLACK, RED	NO
11	R10	RESISTOR, 1K ohms	BROWN, BLACK, RED	NO
12	R11	RESISTOR, 1K ohms	BROWN, BLACK, RED	NO
13	R12	RESISTOR, 1K ohms	BROWN, BLACK, RED	NO
14	H1	COMMON CATHODE SEVEN-SEGMENT DISPLAY	NUMBER "8"	NOTE 1
15	H2	COMMON CATHODE SEVEN-SEGMENT DISPLAY	NUMBER "8"	NOTE 1
16	M1	COMMON CATHODE SEVEN-SEGMENT DISPLAY	NUMBER "8"	NOTE 1
17	M2	COMMON CATHODE SEVEN-SEGMENT DISPLAY	NUMBER "8"	NOTE 1
18	Q1	TRANSISTOR, NPN	FIGURE 1	YES
19	Q2	TRANSISTOR, NPN	FIGURE 1	YES
20	Q3	TRANSISTOR, NPN	FIGURE 1	YES
21	Q4	TRANSISTOR, NPN	FIGURE 1	YES
22	K1	PUSH BUTTON SWITCH	FOUR LEGS	NO
23	K2	PUSH BUTTON SWITCH	FOUR LEGS	NO
24	SO1	STACKABLE HEADER – 8PIN	LONG 8 PIN	NO
25	SO2	STACKABLE HEADER – 10PIN	LONG 10 PIN	NO
26	SO3	STACKABLE HEADER – 6PIN	LONG 6 PIN	NO
27	SO4	STACKABLE HEADER – 8PIN	LONG 8 PIN	NO
28	L1	LED	ONE LONG LEG AND ONE SHORT LEG	YES
29	L2	LED	ONE LONG LEG AND ONE SHORT LEG	YES

NOTE 1. The direction is right if the dot on the component "SEVEN-SEGMENT DISPLAY" match the dot on the PCB.

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 2.
- 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 other board if the pin location is matched. If this is not matched, just route this yourself.

This use all the 14 digital output pin as control.

There is no IC inside the circuit so what this kit behaves is totally depending on the Sketch.

The Sketch on attached is designed based on gentle clock (This is not 100% accurate due to the resolution of micros(), crystal oscillator on the board, the Sketch.....). The program is made by build in function of Arduino IDE. The player can change this kit to a tricky clock by changing some code within the Sketch. Or you can write again the Sketch by using third party libraries so that the program is shorter and easy to be understood.

Or you could change this to the totally different things such as dice gambling.

INSTALLATION:

Just install the component to the PCB M00260083 according to below table.

CIRCUIT DIAGRAM:

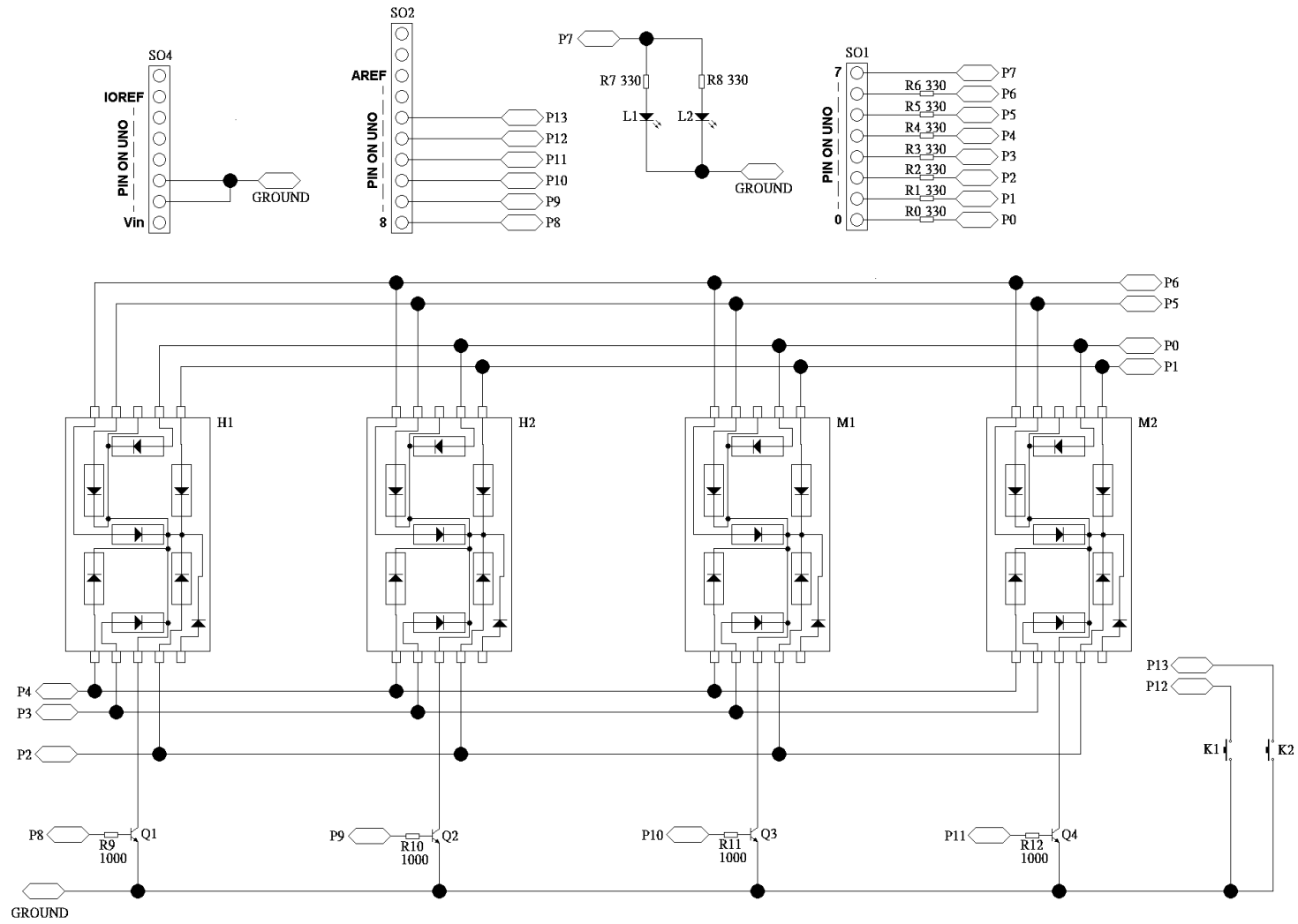


FIGURE 3

SKETCH:

```
/* If you just run this program, this is just a normal clock.
   But you can change this to a tricky clock by changing the code in this program.
   Or change this to a stop watch, counter..... */
```

```
/* Pin for seven-segment display. */
int a = 0; int b = 1; int c = 2;
int d = 3; int e = 4; int f = 5;
int g = 6;
```

```
/* Pin for two middle LED. */
int led = 7;
```

```
/* Pins for control of strobing of 4 sets of seven-segment display. */
int h1 = 8; int h2 = 9;
int m1 = 10; int m2 = 11;
```

```
/* Pin for two keys. */
int k1 = 12; int k2 = 13;
```

```
/* Number to be shown at 4 sets of seven-segment display. */
int h1num = 0; int h2num = 0;
int m1num = 0; int m2num = 0;
```

```
/* Other variables for running this program. */
```

```
int time = 5000;
int number = 0;
int second = 0;
int control1 = 0; int control2 = 1;
int control3 = 1; int control4 = 1;
int val1; int val2;
int adjust = 1;
unsigned long timeAtTheMoment;
unsigned long error;
unsigned long marker;
int onlyDoOneTime = 0;
```

```
void setup() {
```

```
pinMode(a, OUTPUT); pinMode(b, OUTPUT);
pinMode(c, OUTPUT); pinMode(d, OUTPUT);
pinMode(e, OUTPUT); pinMode(f, OUTPUT);
pinMode(g, OUTPUT);
```

```
pinMode(led, OUTPUT);
```

```
pinMode(h1, OUTPUT); pinMode(h2, OUTPUT);
pinMode(m1, OUTPUT); pinMode(m2, OUTPUT);
```

```
pinMode(k1, INPUT_PULLUP); pinMode(k2, INPUT_PULLUP);
```

```
}
```

```
void loop() {
```

```
flashingForOnesecond(); // Flashing for one second.
run(); // Calculating the hour and min after passing flashingForOnesecond().
adjustment(); // Increasing the accurate of the clock automatically for each loop().
flash(m2, m2num); // Flashing for M2.
flash(m1, m1num); // Flashing for M1.
flash(h2, h2num); // Flashing for H2.
flash(h1, h1num); // Flashing for H1.
controlOfKey(); // Control of two keys, K1 and K2.
```

```
}
```

```
/* Flashing for one second.
```

```
The flashing is also shown on middle of two LED. */
```

```
void flashingForOnesecond() {
```

```
if (number == 500) {
  digitalWrite(led, HIGH);
}
```

```
if (number == 1000) {
  digitalWrite(led, LOW);
  second = second + 1;
  number = 0;
}
```

```
}
```

```
/* Increasing the accurate of the clock automatically for each loop(). */
```

```
void adjustment() {
```

```
timeAtTheMoment = micros(); // Current time.
if (onlyDoOneTime == 1) { // This only run at second loop().
  if (timeAtTheMoment - marker > 0) {
    /* error is due to the time using at other part of program but not at delayMicroseconds
       of flash (int k, int i) of last round at loop().
       The error would be subtracted at coming flash(m2, m2num) for self adjustment. */
    error = (timeAtTheMoment - marker - 20000) + error;
  }
}
```

```
/* Marking the current time for comparison using at next round of adjustment(). */
```

```
marker = timeAtTheMoment;
onlyDoOneTime = 1;
```

```
}
```

```
/* Control of two keys, K1 and K2. */
void controlOfKey() {
```

```
val1 = digitalRead(k1);
if (val1 == LOW) {
  delay(500);
  adjust = adjust + 1;
  if (adjust == 5) {
    adjust = 1;
  }
}
val2 = digitalRead(k2);
if (val2 == LOW) {
  delay(500);
  if (adjust == 1) {
    if (m2num < 9) {
      m2num = m2num + 1;
    }
    else {
      m2num = 0;
    }
  }
  if (adjust == 2) {
    if (m1num < 5) {
      m1num = m1num + 1;
    }
    else {
      m1num = 0;
    }
  }
  if (adjust == 3) {
    if (h1num < 2) {
      if (h2num < 9) {
        h2num = h2num + 1;
      }
      else {
        h2num = 0;
      }
    }
    else {
      if (h2num < 3) {
        h2num = h2num + 1;
      }
      else {
        h2num = 0;
      }
    }
  }
  if (adjust == 4) {
    if (h1num < 2) {
      h1num = h1num + 1;
    }
    else {
      h1num = 0;
    }
    if ((h1num == 2) && (h2num > 3)) {
      h2num = 0;
    }
  }
}
```

```
/* Calculating the hour and min after passing flashingForOnesecond(). */
void run() {
```

```
if (control1 == 0) { // m2num calculating from the result of flashingForOnesecond().
```

```
if (second == 60) {
  if (m2num < 9) {
    m2num = m2num + 1;
    control1 = 0; control2 = 1;
    control3 = 1; control4 = 1;
  }
  else {
    m2num = 0;
    control1 = 1; control2 = 0;
    control3 = 1; control4 = 1;
  }
  second = 0;
}
```

```
if (control2 == 0) { // m1num calculating from the result of m2num of above if any.
```

```
if (m1num < 5) {
  m1num = m1num + 1;
  control1 = 0; control2 = 1;
  control3 = 1; control4 = 1;
}
```

```
else {
  m1num = 0;
  control1 = 1; control2 = 1;
  control3 = 0; control4 = 1;
}
```

```
if (control3 == 0) { // h2num calculating from the result of m1num of above if any.
```

```
if (h1num == 2) {
  if (h2num < 3) {
    h2num = h2num + 1;
    control1 = 0; control2 = 1;
    control3 = 1; control4 = 1;
  }
  else {
    h2num = 0;
    control1 = 1; control2 = 1;
    control3 = 1; control4 = 0;
  }
}
```

```
else {
  h2num = 0;
  control1 = 1; control2 = 1;
  control3 = 1; control4 = 0;
}
```

```
}
```

SKETCH:

```

else {
  if (h2num < 9) {
    h2num = h2num + 1;
    control1 = 0; control2 = 1;
    control3 = 1; control4 = 1;
  }
  else {
    h2num = 0;
    control1 = 1; control2 = 1;
    control3 = 1; control4 = 0;
  }
}
if (control4 == 0) { // h1num calculating from the result of h2num of above if any.
  if (h1num < 2) {
    h1num = h1num + 1;
    control1 = 0; control2 = 1;
    control3 = 1; control4 = 1;
  }
  else {
    h1num = 0;
    control1 = 0; control2 = 1;
    control3 = 1; control4 = 1;
  }
}

/* Flashing for M2. Flashing for M1, Flashing for H2, Flashing for H1 .
The first propose is the counting of 5/1000 second.
The second propose is the flashing of 4 sets of seven-segment display by using the method of strobing. */
void flash (int k, int i) {

  if (k == h1) {
    digitalWrite(h1, HIGH); digitalWrite(h2, LOW);
    digitalWrite(m1, LOW); digitalWrite(m2, LOW);
    display(i); //Number to be displayed at 7 segment LED.
    /* Counting of 5/1000 - error) second.
    The error is from adjustment(). */
    delayMicroseconds(time - error);
    number = number + 5;
  }
  if (k == h2) {
    digitalWrite(h1, LOW); digitalWrite(h2, HIGH);
    digitalWrite(m1, LOW); digitalWrite(m2, LOW);
    display(i); // Number to be displayed at 7 segment LED.
    /* Counting of 5/1000 second. */
    delayMicroseconds(time);
    number = number + 5;
  }
  if (k == m1) {
    digitalWrite(h1, LOW); digitalWrite(h2, LOW);
    digitalWrite(m1, HIGH); digitalWrite(m2, LOW);
    display(i); // Number to be displayed at 7 segment LED.
    /* Counting of 5/1000 second. */
    delayMicroseconds(time);
    number = number + 5;
  }
  if (k == m2) {
    digitalWrite(h1, LOW); digitalWrite(h2, LOW);
    digitalWrite(m1, LOW); digitalWrite(m2, HIGH);
    display(i); // Number to be displayed at 7 segment LED.
    /* Counting of 5/1000 second. */
    delayMicroseconds(time);
    number = number + 5;
  }
}
/* Turning off the seven-segment display after each flashing. */
digitalWrite(a, LOW); digitalWrite(b, LOW);
digitalWrite(c, LOW); digitalWrite(d, LOW);
digitalWrite(e, LOW); digitalWrite(f, LOW);
digitalWrite(g, LOW);
}

```

```

/* Display of number on seven-segment display. */
void display (int i) {

  if (i == 0) {
    digitalWrite(a, HIGH); digitalWrite(b, HIGH);
    digitalWrite(c, HIGH); digitalWrite(d, HIGH);
    digitalWrite(e, HIGH); digitalWrite(f, HIGH);
    digitalWrite(g, LOW);
  }
  if (i == 1) {
    digitalWrite(a, LOW); digitalWrite(b, HIGH);
    digitalWrite(c, HIGH); digitalWrite(d, LOW);
    digitalWrite(e, LOW); digitalWrite(f, LOW);
    digitalWrite(g, LOW);
  }
  if (i == 2) {
    digitalWrite(a, HIGH); digitalWrite(b, HIGH);
    digitalWrite(c, LOW); digitalWrite(d, HIGH);
    digitalWrite(e, HIGH); digitalWrite(f, LOW);
    digitalWrite(g, HIGH);
  }
  if (i == 3) {
    digitalWrite(a, HIGH); digitalWrite(b, HIGH);
    digitalWrite(c, HIGH); digitalWrite(d, HIGH);
    digitalWrite(e, LOW); digitalWrite(f, LOW);
    digitalWrite(g, HIGH);
  }
  if (i == 4) {
    digitalWrite(a, LOW); digitalWrite(b, HIGH);
    digitalWrite(c, HIGH); digitalWrite(d, LOW);
    digitalWrite(e, LOW); digitalWrite(f, HIGH);
    digitalWrite(g, HIGH);
  }
  if (i == 5) {
    digitalWrite(a, HIGH); digitalWrite(b, LOW);
    digitalWrite(c, HIGH); digitalWrite(d, HIGH);
    digitalWrite(e, LOW); digitalWrite(f, HIGH);
    digitalWrite(g, HIGH);
  }
  if (i == 6) {
    digitalWrite(a, HIGH); digitalWrite(b, LOW);
    digitalWrite(c, HIGH); digitalWrite(d, HIGH);
    digitalWrite(e, HIGH); digitalWrite(f, HIGH);
    digitalWrite(g, HIGH);
  }
  if (i == 7) {
    digitalWrite(a, HIGH); digitalWrite(b, HIGH);
    digitalWrite(c, HIGH); digitalWrite(d, LOW);
    digitalWrite(e, LOW); digitalWrite(f, LOW);
    digitalWrite(g, LOW);
  }
  if (i == 8) {
    digitalWrite(a, HIGH); digitalWrite(b, HIGH);
    digitalWrite(c, HIGH); digitalWrite(d, HIGH);
    digitalWrite(e, HIGH); digitalWrite(f, HIGH);
    digitalWrite(g, HIGH);
  }
  if (i == 9) {
    digitalWrite(a, HIGH); digitalWrite(b, HIGH);
    digitalWrite(c, HIGH); digitalWrite(d, HIGH);
    digitalWrite(e, LOW); digitalWrite(f, HIGH);
    digitalWrite(g, HIGH);
  }
}

```