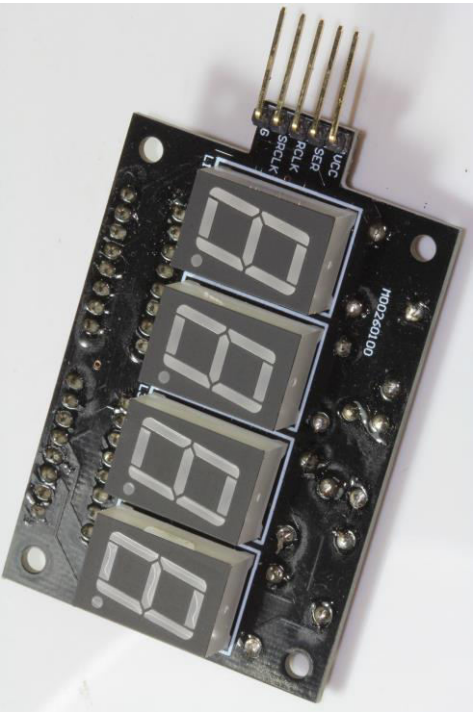


4 DIGIT SEVEN SEGMENT DISPLAY
MODULE

(ARDUINO COMPATIBLE)
PRODUCT CODE: M00270062



- FEATURE:
4 digit display
5 pin (Two of them are voltage supply and ground)
2 sets of 74HC595 for control
One sample Sketch is attached.
Assembly is needed by player.

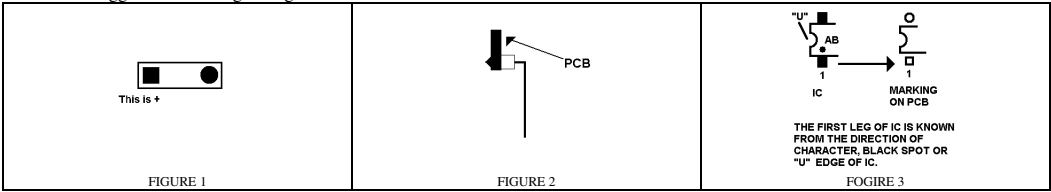
result is that this display “ 4 “. After you run four times of repaint(2,4,5,6). The fifth time of running repaint(2,4,5,6) would show “2” again. You can see the display of four number would be smoother when you run repaint(2,3,5,6) more and more. One interesting thing is that even you want to input data into U2, the data must pass SER on break away male header, then U1, then reach U2.

INSTALLATION:
Just install the component to the PCB M00260100 according to below table

Table with 5 columns: ITEM, SYMBOL ON PCB, DESCRIPTION, OUTLOOK, DIRECTION IS IMPORTANT?. It lists components like resistors (R1-R8), break away male headers (S1), common cathode seven-segment displays (L1-L4), and 74HC595 ICs (U1, U2) with their respective pin counts and installation directions.

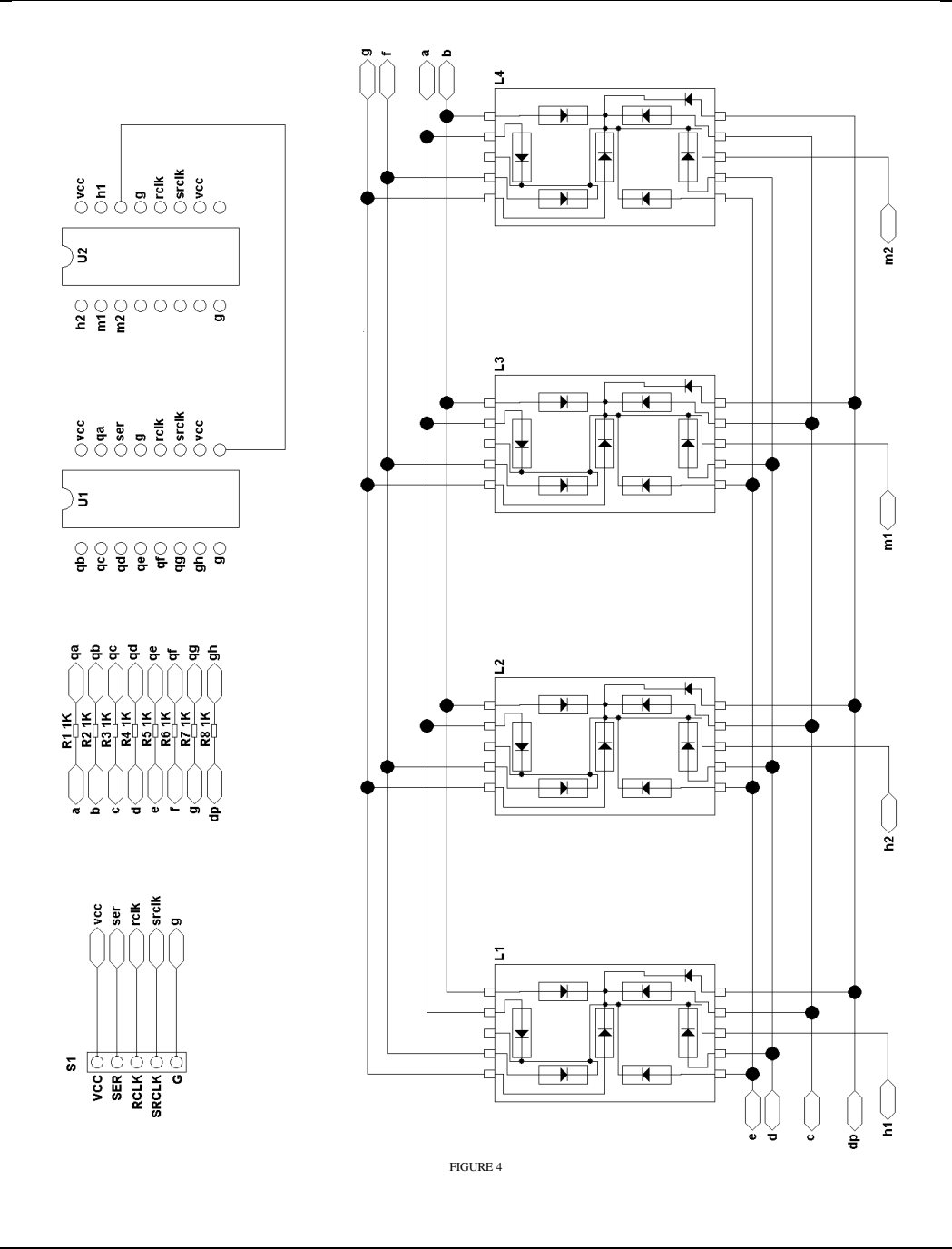
NOTE 1. The direction is right if the dot on the component “SEVEN-SEGMENT DISPLAY” match the dot on the PCB.

- 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 “+”.
On PCB marking, square pad as Figure 1 is always “+”.
Do not connect this shield into the Arduino when downloading the Sketch.
I suggest the working voltage of this module is 5V.



DESCRIPTION:
The circuit design and program 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.
The benefit of using 74HC595 is that you can use lesser pin to control more seven segment display.
If you want to use this module without any problem, this is better you understand how IC 74HC595 work from the specification. If finally you still do not understand how this works, you can see this as black box and just use the code at sample Sketch. The below is little explanation for such IC, circuit and Sketch.
The basic concept of this circuit is to use Strobing method to display all four displays. For example, you want to display “2456”. At one moment, this only displays “2 “. At another moment, this displays “ 4 “. Then “ 5 “ and final “ 6”. When this action is fast enough, you would see “2456”. U1 is to control the number to be displayed at that moment; U2 is to control the flashing issue. Because there are only 3 Pins for inputting the number into the kits, this uses serial method for inputting. Now go to the Sketch and also use “2456” as example. We see there is an function repaint(a,b,c,d). When you first time run this line on loop(), you actually input “2” into U1 and “Ground, High Voltage, High Voltage, High Voltage” into U2. The result is that this display “2 “ at that moment. When you run repaint(2,4,5,6) for second time, now you input “4” into U1 and “High Voltage, Ground, High Voltage, High Voltage” into U2. The

CIRCUIT DIAGRAM:



SKETCH:

The Sketch use the below UNO pin connection as example.

PIN ON MODULE	PIN ON UNO BOARD
VCC	5V
SER	7
RCLK	6
SRCLK	5
G	G

```
/*
 * The below program would display " 2.0", where "2." is fixed and "0" would increase from "0" to "9",
 * then go back to "0" again and run again.
 */

/* Connecting pin to the Arduino board. */
int SRCLK = 5;
int RCLK = 6;
int SER = 7;

/* This variable "j" is for Strobing. */
int j;

/* The variable "nothing" is to display nothing on seven segment display. */
int nothing = 30;

/* The below variable is to display the number such as "1.", "3.", "7.". */
int zeroPoint = 100; int onePoint = 110; int twoPoint = 120; int threePoint = 130;
int fourPoint = 140; int fivePoint = 150; int sixPoint = 160; int sevenPoint = 170;
int eightPoint = 180; int ninePoint = 190;

/* The below variable "nothingPoint" is to display only " ." */
int nothingPoint = 200;

/* The below variable is for the below sample "loop()". This is not a must when you write your own code. */
int count; int e = 0;
int f = twoPoint; // This is to display only "2.".
int g = nothing; // Nothing is to be displayed.
int h = nothing; // Nothing is to be displayed.

/* Setting the pin as output pin. */
void setup() {
  pinMode(SRCLK, OUTPUT);
  pinMode(RCLK, OUTPUT);
  pinMode(SER, OUTPUT);
}

void loop() {

  repaint (e, f, g, h); // The code repaint (e,f,g,h) is put randomly on any part of code.
  count = count + 1;
  if(count == 1000) {
    e = e+1;
    repaint (e, f, g, h); // The code repaint (e,f,g,h) is put randomly on any part of code.
    count = 0;
  }
  repaint (e, f, g, h); // The code repaint (e,f,g,h) is put randomly on any part of code.
  if(e == 10) {
    e = 0;
    repaint (e, f, g, h); // The code repaint (e,f,g,h) is put randomly on any part of code.
  }
}
```

SKETCH:

```
/*
 * The below code is for displaying the number on four seven segment display.
 * If you find you do not understand what the code is saying, just see this
 * as black box. Just use the code and no need to understand this.
 * For example, if you want to display "3456",
 * just use the code repaint(3,4,5,6).
 * On the other hand, because this use Strobing method for displaying.
 * You must insert the below line many time randomly on the "loop()".
 * I use "while" as example and you want to display
 * "1345". If you do not put repaint(a,b,c,d) inside a "while", this may display only"   4 ".
 * Which number to be displayed depend on which number of "int j" this is.
 */
void repaint(int a, int b, int c, int d) {

    byte num[4];
    byte input[4];
    input[0] = a;   input[1] = b;
    input[2] = c;   input[3] = d;

/* This is for Strobing. */
const byte flash[4] {
    B11101111,   B11011111,
    B10111111,   B01111111,
};

/*
 * The below variable is for translating code of the input number into
 * actual power input to the leg of seven segment display.
 */
byte zeroPaint   = B11111100; byte onePaint   = B01100000;
byte twoPaint    = B11011010; byte threePaint = B11110010;
byte fourPaint   = B01100110; byte fivePaint  = B10110110;
byte sixPaint    = B10111110; byte sevenPaint = B11100000;
byte eightPaint  = B11111110; byte ninePaint  = B11110110;
byte noWordPaint = B00000000;

/*
 * The below variable is for translating code of input number such as "1.", "2.".....
 * into actual power input to the leg of seven segment display.
 */
byte zeroPointPaint = B11111101; byte onePointPaint = B01100001;
byte twoPointPaint  = B11011011; byte threePointPaint = B11110011;
byte fourPointPaint = B01100111; byte fivePointPaint = B10110111;
byte sixPointPaint  = B10111111; byte sevenPointPaint = B11100001;
byte eightPointPaint = B11111111; byte ninePointPaint = B11110111;
byte noWordPointPaint = B00000001;

/* Translating code for four seven segment display. */
for(int i = 0; i<4; i++) {
    if(input[i] == 0) {num[i]   = zeroPaint;}    if(input[i] == 1) {num[i] = onePaint;}
    if(input[i] == 2) {num[i]   = twoPaint;}     if(input[i] == 3) {num[i] = threePaint;}
    if(input[i] == 4) {num[i]   = fourPaint;}     if(input[i] == 5) {num[i] = fivePaint;}
    if(input[i] == 6) {num[i]   = sixPaint;}      if(input[i] == 7) {num[i] = sevenPaint;}
    if(input[i] == 8) {num[i]   = eightPaint;}    if(input[i] == 9) {num[i] = ninePaint;}
    if(input[i] == 30) {num[i] = noWordPaint;}

    if(input[i] == zeroPoint) {num[i]   = zeroPointPaint;}    if(input[i] == onePoint) {num[i]   = onePointPaint;}
    if(input[i] == twoPoint) {num[i]    = twoPointPaint;}     if(input[i] == threePoint) {num[i] = threePointPaint;}
    if(input[i] == fourPoint) {num[i]   = fourPointPaint;}    if(input[i] == fivePoint) {num[i]  = fivePointPaint;}
    if(input[i] == sixPoint) {num[i]    = sixPointPaint;}     if(input[i] == sevenPoint) {num[i] = sevenPointPaint;}
    if(input[i] == eightPoint) {num[i]  = eightPointPaint;}   if(input[i] == ninePoint) {num[i]  = ninePointPaint;}

    if(input[i] == nothingPoint) {num[i] = noWordPointPaint;}
}
}
```

```
/* Inputing the translated code into IC 74HC595 after translating code of above. */
for(int k = 0; k<8; k++) {
    if (bitRead(flash[j], k)) {
        digitalWrite(SER, HIGH);
    }
    else {
        digitalWrite(SER, LOW);
    }
}
// When this go from LOW to HIGH, this let one bit of data to be moved forward to next stage of shift register.
digitalWrite(SRCLK, HIGH);
digitalWrite(SRCLK, LOW); // This let "go from LOW to HIGH" of SRCLK can run again at next round.
}
for (int l = 0; l<8; l++) {
    if(bitRead(num[j], l)){
        digitalWrite(SER, HIGH);
    }
    else {
        digitalWrite(SER, LOW);
    }
    digitalWrite(SRCLK, HIGH);
    digitalWrite(SRCLK, LOW);
}
digitalWrite(RCLK, HIGH); // When this go from LOW to HIGH, shift-register data is stored in the storage register.
digitalWrite(RCLK, LOW); // This let "go from LOW to HIGH" of RCLK can run again at next round.

/*
 * The below code is for Strobing so that four seven segment display could be displayed.
 * When you run repaint(a,b,c,d) one time, "j" would add 1. When this is 4, this go back to 0.
 * The function is to let all four seven segment LED to be averagely displayed.
 */
j = j + 1;
if(j == 4) {
    j = 0;
}

}
```