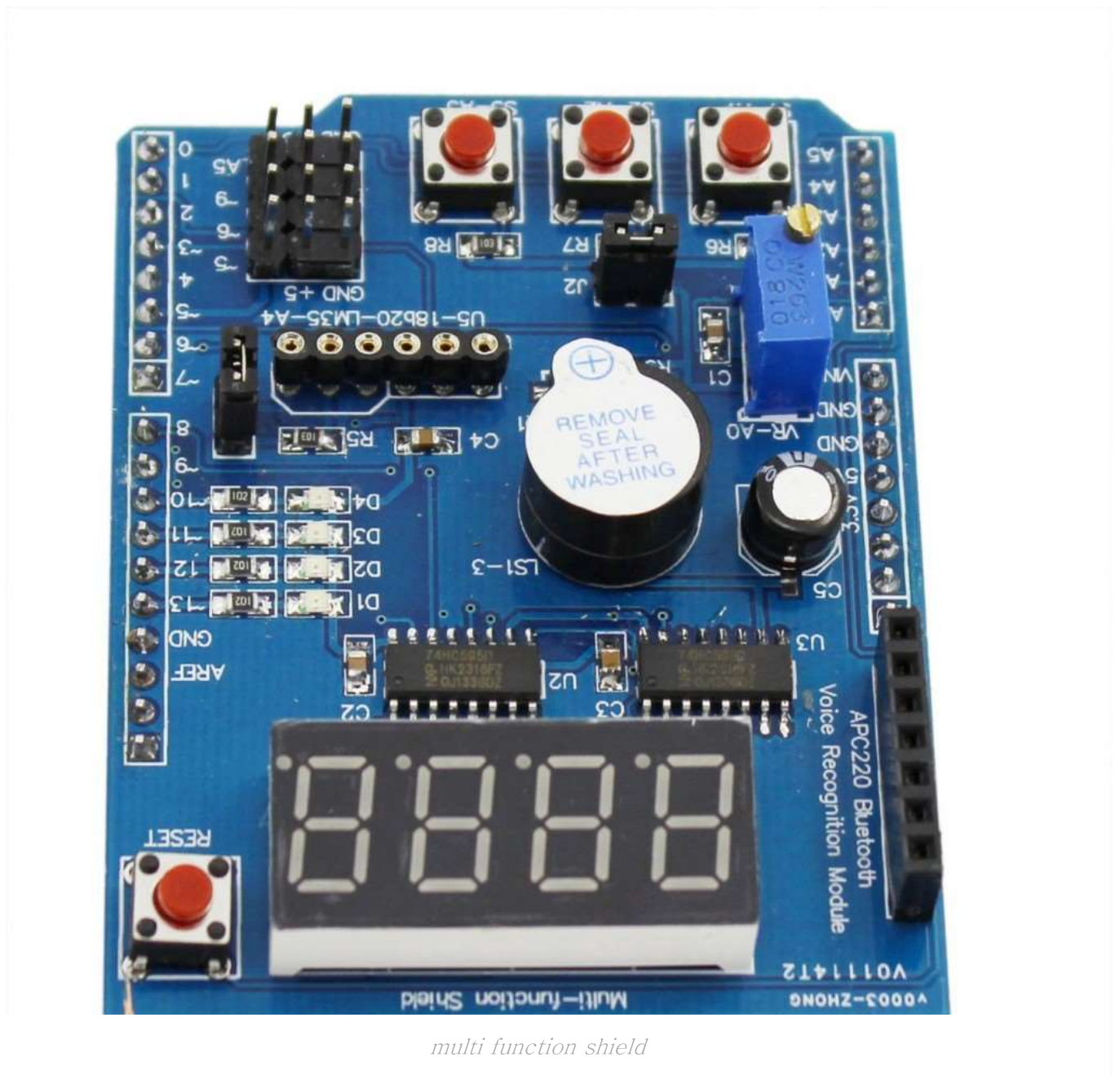


## MULTI FUNCTION SHIELD EXAMPLES

This shield got my attention as it looked like a nice beginners learning type shield with which you could get up and running with an Arduino

Here is a picture of the board, a few code examples are available later on in the article.



Features

4 digit 7-segment LED display module driven by two serial 74HC595's

4 LED's

10K potentiometer

3 x push buttons

Piezo buzzer

DS18B20 temperature sensor interface (not included)

Infrared receiver interface

Serial interface header for connection to serial modules

## Code Examples

\*\*\*\*\*

### Blinking LED

```
1  int led = 13;
2
3  void setup()
4  {
5  // initialize the digital pin as an output.
6  pinMode(led, OUTPUT);
7  }
8
9  void loop()
10 {
11 digitalWrite(led, HIGH);
12 delay(1000);
13 digitalWrite(led, LOW);
14 delay(1000);
15 }
```

\*\*\*\*\*

### All LEDS blinking

```
1  int led1 = 13;
2  int led2 = 12;
3  int led3 = 11;
4  int led4 = 10;
5
6  void setup()
7  {
8  // initialize the digital pin as an output.
9  pinMode(led1, OUTPUT);
10 pinMode(led2, OUTPUT);
11 pinMode(led3, OUTPUT);
12 pinMode(led4, OUTPUT);
13 }
14
15 void loop()
16 {
17 digitalWrite(led1, HIGH);
18 digitalWrite(led2, HIGH);
19 digitalWrite(led3, HIGH);
20 digitalWrite(led4, HIGH);
21 delay(1000);
22 digitalWrite(led1, LOW);
23 digitalWrite(led2, LOW);
24 digitalWrite(led3, LOW);
25 digitalWrite(led4, LOW);
```

```
26 | delay(1000);
27 | }
```

\*\*\*\*\*

## Switches example

```
1 | const byte LED[] = {13,12,11,10};
2 |
3 | #define BUTTON1 A1
4 | #define BUTTON2 A2
5 |
6 | void setup()
7 | {
8 |   // initialize the digital pin as an output.
9 |   /* Set each pin to outputs */
10 |  pinMode(LED[0], OUTPUT);
11 |  pinMode(LED[1], OUTPUT);
12 |  pinMode(LED[2], OUTPUT);
13 |  pinMode(LED[3], OUTPUT);
14 | }
15 |
16 | void loop()
17 | {
18 |   if(!digitalRead(BUTTON1))
19 |   {
20 |     digitalWrite(LED[0], HIGH);
21 |     digitalWrite(LED[1], HIGH);
22 |     digitalWrite(LED[2], HIGH);
23 |     digitalWrite(LED[3], HIGH);
24 |   }
25 |
26 |   if(!digitalRead(BUTTON2))
27 |   {
28 |     digitalWrite(LED[0], LOW);
29 |     digitalWrite(LED[1], LOW);
30 |     digitalWrite(LED[2], LOW);
31 |     digitalWrite(LED[3], LOW);
32 |   }
33 | }
```

\*\*\*\*\*

## Potentiometer 1

```
1 | #define Pot1 0
2 |
3 | void setup()
4 | {
5 |   Serial.begin(9600);
6 | }
7 |
8 | /* Main Program */
9 | void loop()
10 | {
11 |
12 |   Serial.print("Potentiometer reading: ");
13 |   Serial.println(analogRead(Pot1));
14 |   /* Wait 0.5 seconds before reading again */
15 |   delay(500);
16 | }
```

\*\*\*\*\*

## Pot and led

```

1  const byte LED[] = {13,12,11,10};
2  #define Pot1 0
3
4  void setup()
5  {
6  Serial.begin(9600);
7  // initialize the digital pin as an output.
8  /* Set each pin to outputs */
9  pinMode(LED[0], OUTPUT);
10 pinMode(LED[1], OUTPUT);
11 pinMode(LED[2], OUTPUT);
12 pinMode(LED[3], OUTPUT);
13 }
14
15 /* Main Program */
16 void loop()
17 {
18 int PotValue;
19 //Serial.print("Potentiometer reading: ");
20 PotValue = analogRead(Pot1);
21 /* Wait 0.5 seconds before reading again */
22 if(PotValue < 400)
23 {
24 digitalWrite(LED[0], LOW);
25 digitalWrite(LED[1], LOW);
26 digitalWrite(LED[2], LOW);
27 digitalWrite(LED[3], LOW);
28 Serial.print("Potentiometer: ");
29 Serial.println(PotValue);
30 }
31 else
32 {
33 digitalWrite(LED[0], HIGH);
34 digitalWrite(LED[1], HIGH);
35 digitalWrite(LED[2], HIGH);
36 digitalWrite(LED[3], HIGH);
37 Serial.print("Potentiometer: ");
38 Serial.println(PotValue);
39 }
40 delay(500);
41 }

```

\*\*\*\*\*

## segment display

```

1  /* Define shift register pins used for seven segment display */
2  #define LATCH_DIO 4
3  #define CLK_DIO 7
4  #define DATA_DIO 8
5
6  /* Segment byte maps for numbers 0 to 9 */
7  const byte SEGMENT_MAP[] = {0xC0,0xF9,0xA4,0xB0,0x99,0x92,0x82,0xF8,0X80,0X90};
8  /* Byte maps to select digit 1 to 4 */
9  const byte SEGMENT_SELECT[] = {0xF1,0xF2,0xF4,0xF8};
10
11 void setup ()
12 {
13 /* Set DIO pins to outputs */
14 pinMode(LATCH_DIO,OUTPUT);
15 pinMode(CLK_DIO,OUTPUT);
16 pinMode(DATA_DIO,OUTPUT);
17 }
18
19 /* Main program */
20 void loop()
21 {
22
23 /* Update the display with the current counter value */
24 WriteNumberToSegment(0 , 0);

```

```

25 WriteNumberToSegment(1 , 1);
26 WriteNumberToSegment(2 , 2);
27 WriteNumberToSegment(3 , 3);
28 }
29
30 /* Write a decimal number between 0 and 9 to one of the 4 digits of the display */
31 void WriteNumberToSegment(byte Segment, byte Value)
32 {
33   digitalWrite(LATCH_DIO,LOW);
34   shiftOut(DATA_DIO, CLK_DIO, MSBFIRST, SEGMENT_MAP[Value]);
35   shiftOut(DATA_DIO, CLK_DIO, MSBFIRST, SEGMENT_SELECT[Segment] );
36   digitalWrite(LATCH_DIO,HIGH);
37 }

```

\*\*\*\*\*

Read pot and display value on display

```

1  /* Define shift register pins used for seven segment display */
2  #define LATCH_DIO 4
3  #define CLK_DIO 7
4  #define DATA_DIO 8
5
6  #define Pot1 0
7
8  /* Segment byte maps for numbers 0 to 9 */
9  const byte SEGMENT_MAP[] = {0xC0,0xF9,0xA4,0xB0,0x99,0x92,0x82,0xF8,0X80,0X90};
10 /* Byte maps to select digit 1 to 4 */
11 const byte SEGMENT_SELECT[] = {0xF1,0xF2,0xF4,0xF8};
12
13 void setup ()
14 {
15   Serial.begin(9600);
16   /* Set DIO pins to outputs */
17   pinMode(LATCH_DIO,OUTPUT);
18   pinMode(CLK_DIO,OUTPUT);
19   pinMode(DATA_DIO,OUTPUT);
20 }
21
22 /* Main program */
23 void loop()
24 {
25   int PotValue;
26   PotValue = analogRead(Pot1);
27   Serial.print("Potentiometer: ");
28   Serial.println(PotValue);
29   /* Update the display with the current counter value */
30   WriteNumberToSegment(0 , PotValue / 1000);
31   WriteNumberToSegment(1 , (PotValue / 100) % 10);
32   WriteNumberToSegment(2 , (PotValue / 10) % 10);
33   WriteNumberToSegment(3 , PotValue % 10);
34 }
35
36 /* Write a decimal number between 0 and 9 to one of the 4 digits of the display */
37 void WriteNumberToSegment(byte Segment, byte Value)
38 {
39   digitalWrite(LATCH_DIO,LOW);
40   shiftOut(DATA_DIO, CLK_DIO, MSBFIRST, SEGMENT_MAP[Value]);
41   shiftOut(DATA_DIO, CLK_DIO, MSBFIRST, SEGMENT_SELECT[Segment] );
42   digitalWrite(LATCH_DIO,HIGH);
43 }

```

\*\*\*\*\*

Resources

