

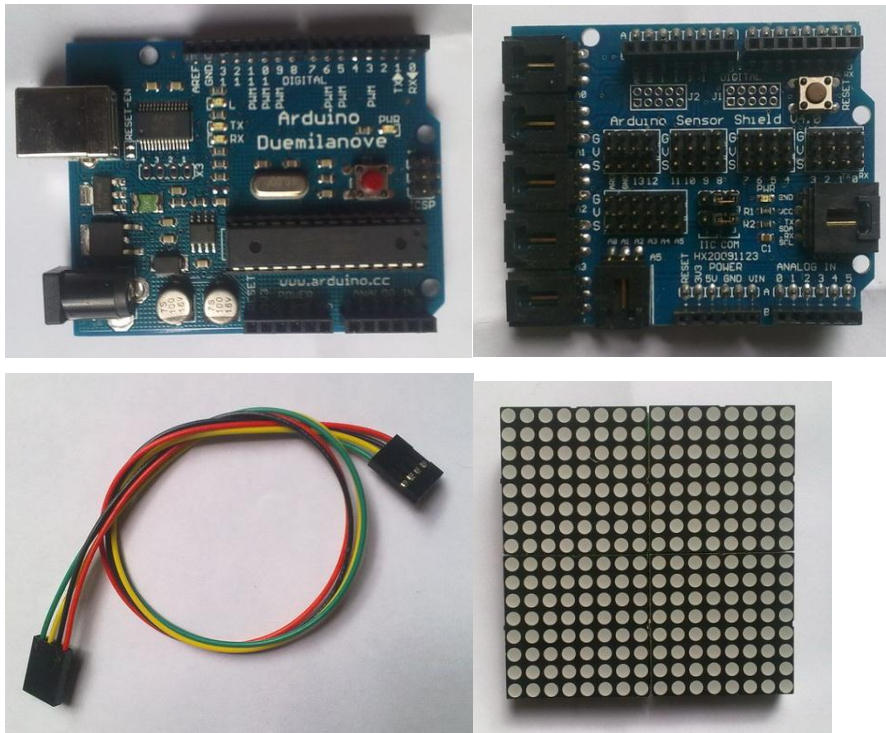
Experiment on Wrobot 16x16 LED Array

Copyright reserved by Elechouse

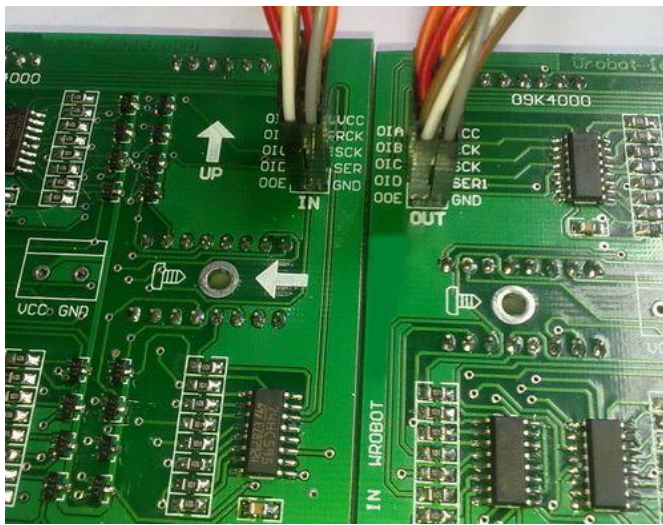
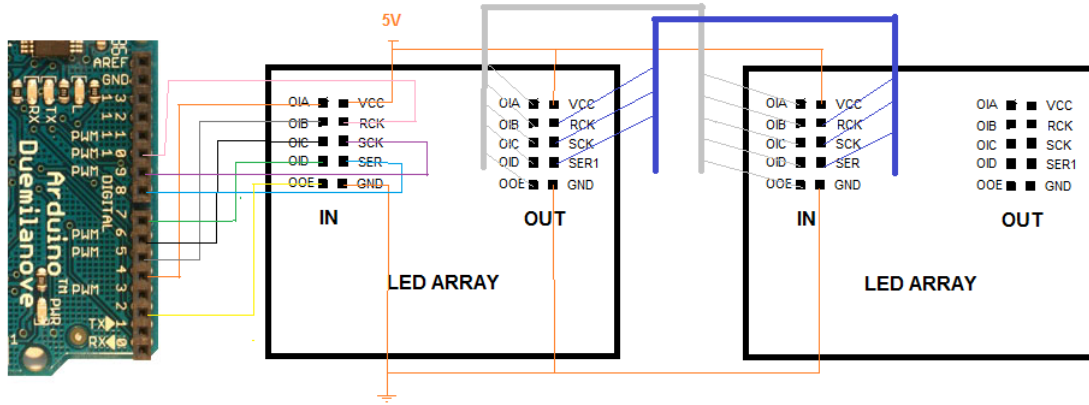
Now i am telling you something of LED Array Screen. I will show you a DIY experiment.

Tools used:

1. [Arduino Duemilanove 2009 ATmega 168](#)
2. [Arduino Sensor shield v4.0](#)
3. [4-pin jumper wire](#)
4. [1-pin jumper wire](#)
5. [Wrobot 16x16 LED Array Screen](#)



Connect digital pin 4,5,6,7 to input pin OIA,OIB,OIC,OID of LED screen, which choose the row. Connect digital pin 8,9,10 to pin SER,SCK,RCKof screen, which input the data of a row. Connect GND and VCC to power. Extention connection is very easy too. Connect pin on the OUT side of one board to the pin with the same name on the IN side of the next board.



/*0xff,0xff,0x80,0x01,0x80,0x01,0x9f,0xff,0x9f,0xff,0x9f,0xff,0x9f,0xff,0x80,0x03,
0x80,0x03,0x9f,0xff,0x9f,0xff,0x9f,0xff,0x9f,0xff,0x80,0x01,0x80,0x01,0xff,0xff, // "E"

0xff,0xff,0x9f,0xff,0x9f,0xff,0x9f,0xff,0x9f,0xff,0x9f,0xff,0x9f,0xff,0x9f,0xff,
0x9f,0xff,0x9f,0xff,0x9f,0xff,0x9f,0xff,0x9f,0xff,0x80,0x01,0x80,0x01,0xff,0xff, // "L"

0xff,0xff,0x80,0x01,0x80,0x01,0x9f,0xff,0x9f,0xff,0x9f,0xff,0x9f,0xff,0x80,0x03,
0x80,0x03,0x9f,0xff,0x9f,0xff,0x9f,0xff,0x9f,0xff,0x80,0x01,0x80,0x01,0xff,0xff, // "E"

0xff,0xff,0xe0,0x07,0xc0,0x03,0xcf,0xf3,0x9f,0xff,0x9f,0xff,0x9f,0xff,0x9f,0xff,
0x9f,0xff,0x9f,0xff,0x9f,0xff,0x9f,0xff,0xcf,0xf3,0xc0,0x03,0xe0,0x07,0xff,0xff, // "C"

0xff,0xff,0x9f,0xf9,0x9f,0xf9,0x9f,0xf9,0x9f,0xf9,0x9f,0xf9,0x9f,0xf9,0x80,0x01,
0x80,0x01,0x9f,0xf9,0x9f,0xf9,0x9f,0xf9,0x9f,0xf9,0x9f,0xf9,0x9f,0xf9,0xff,0xff, // "H"

0xff,0xff,0xf8,0x1f,0xf0,0x0f,0xe3,0xc7,0xe7,0xe7,0xc7,0xe3,0xcf,0xf3,0xcf,0xf3,
0xcf,0xf3,0xcf,0xf3,0xc7,0xe3,0xe7,0xe7,0xe3,0xc7,0xf0,0x0f,0xf8,0x1f,0xff,0xff, // "O"

0xff,0xff,0x9f,0xf9,0x9f,0xf9,0x9f,0xf9,0x9f,0xf9,0x9f,0xf9,0x9f,0xf9,0x9f,0xf9,
0x9f,0xf9,0x9f,0xf9,0x9f,0xf9,0x9f,0xf9,0x8f,0xf1,0xc0,0x03,0xe0,0x07,0xff,0xff, // "U"

0xff,0xff,0xfc,0x3f,0xf9,0x9f,0xf3,0xcf,0xe7,0xe7,0xe7,0xff,0xf3,0xff,0xf8,0x3f,
0xfc,0x1f,0xff,0xcf,0xff,0xe7,0xe7,0xe7,0xf3,0xcf,0xf9,0x9f,0xfc,0x3f,0xff,0xff, // "S"

0xff,0xff,0x80,0x01,0x80,0x01,0x9f,0xff,0x9f,0xff,0x9f,0xff,0x9f,0xff,0x80,0x03,
0x80,0x03,0x9f,0xff,0x9f,0xff,0x9f,0xff,0x9f,0xff,0x80,0x01,0x80,0x01,0xff,0xff, // "E"

0xff,0xff,0x8f,0xf1,0x87,0xe1,0x83,0xc1,0x93,0xc9,0x93,0xc9,0x99,0x99,0x99,0x99,
0x99,0x99,0x9c,0x39,0x9c,0x39,0x9e,0x79,0x9f,0xf9,0x9f,0xf9,0x9f,0xf9,0xff,0xff, // "M"*/

0xff,0xff,0xfc,0x3f,0xf3,0xcf,0xef,0xf7,0xdf,0xfb,0xd3,0xcb,0xbf,0xfd,0xbf,0xfd,
0xbf,0xfd,0xbb,0xdd,0xdc,0x3b,0xdf,0xfb,0xef,0xf7,0xf3,0xcf,0xfc,0x3f,0xff,0xff, // smile

0xff,0xff,0xfc,0x3f,0xfc,0x3f,0xf8,0x1f,0xf9,0x9f,0xf1,0x8f,0xf3,0xcf,0xf3,0xcf,
0xe0,0x07,0xe0,0x07,0xe7,0xe7,0xc7,0xe3,0xcf,0xf3,0x8f,0xf1,0x9f,0xf9,0xff,0xff, // "A"

0xff,0xff,0xe7,0xcf,0xe7,0x9f,0xe7,0x3f,0xe6,0x7f,0xe4,0xff,0xe1,0xff,0xe3,0xff,
0xe7,0xff,0xe7,0xff,0xe7,0xff,0xe7,0xff,0xe7,0xff,0xe7,0xff,0xff,0xff, // "r"

0xff,0xff,0xff,0x9f,0xff,0x9f,0xff,0x9f,0xff,0x9f,0xf0,0x1f,0xe0,0x1f,0xc7,0x9f,
0xcf,0x9f,0xcf,0x9f,0xcf,0x9f,0xc7,0x9f,0xe0,0x0f,0xf0,0x0f,0xff,0xff,0xff,0xff, // "d"

0xff,0xff,0xff,0xff,0xff,0xff,0xe7,0xcf,0xe7,0xcf,0xe7,0xcf,0xe7,0xcf,0xe7,0xcf,
0xe7,0xcf,0xe7,0xcf,0xe7,0xcf,0xe3,0x8f,0xf0,0x0f,0xf8,0x27,0xff,0xff,0xff,0xff, // "u"

```
0xff,0xff,0xfe,0x7f,0xfe,0x7f,0xff,0xff,0xfe,0x7f,0xfe,0x7f,0xfe,0x7f,0xfe,0x7f,
0xfe,0x7f,0xfe,0x7f,0xfe,0x7f,0xfe,0x7f,0xfe,0x7f,0xff,0xff,0xff,///"i"

0xff,0xff,0xff,0xff,0xf8,0x3f,0xf0,0x1f,0xe3,0x8f,0xe7,0xcf,0xe7,0xcf,0xe7,0xcf,
0xe7,0xcf,0xe7,0xcf,0xe7,0xcf,0xe7,0xc7,0xff,0xff,0xff,0xff,0xff,///"n"

0xff,0xff,0xff,0xff,0xf8,0x3f,0xf0,0x1f,0xf3,0x9f,0xe7,0xcf,0xe7,0xcf,0xe7,0xcf,
0xe7,0xcf,0xe7,0xcf,0xe7,0xcf,0xf3,0x9f,0xf0,0x1f,0xf8,0x3f,0xff,0xff,0xff,0xff///"o"

};
byte BUFF[2*ScreenNO+2];
byte col,disrow; //col--moving pointer;disrow--scanning row
unsigned int wordnumber; //the bytes of words you want to display
int Dpin[]={4,5,6,7};
int OOE=2;
int RCK=10;
int SCK=9;
int SER=8;
int VCC=11;
int GND=3;

void setup()
{
  pinMode(2,OUTPUT);
  pinMode(3,OUTPUT);
  pinMode(4,OUTPUT);
  pinMode(5,OUTPUT);
  pinMode(6,OUTPUT);
  pinMode(7,OUTPUT);
  pinMode(8,OUTPUT);
  pinMode(9,OUTPUT);
  pinMode(10,OUTPUT);
  pinMode(11,OUTPUT);

  //Serial.begin(9600);
  digitalWrite(OOE,HIGH);
  digitalWrite(VCC,HIGH);
  digitalWrite(GND,LOW);
  digitalWrite(SCK,LOW);
}

void loop()
{
  byte i;
```

```

col=0;wordnumber=0;
while(1)
{
    while(col<16)           // circulate for 16 time, which is moving a letter through a creen
    {
        for(i=0;i<SPEED;i++) //the time of static display for a letter
        {
            for(disrow=0;disrow<16;disrow++) //scanning the row
            {
                digitalWrite(OOE,HIGH);
                digitalWrite(RCK,LOW);

                loadoneline(); //load the data of a row
                sendoneline(); //send the data of a row
                sendPin(disrow);
                digitalWrite(OOE,LOW); // enable the row scanning
                digitalWrite(RCK,HIGH); //light a row
                delay(1);
            }
        }
        col++; //moving pointer
    }
    col=0;
    wordnumber+=32; // point to the next word
    if(wordnumber>=WordNO*32)wordnumber=0; //while all the letters are dispalyed, back
    to the beginning
}
}

/***** load the data of a row *****/
void loadoneline( )
{
    byte s;
    unsigned int k;
    for(s=0;s< ScreenNO+1;s++)
    {
        k = wordnumber+32*s+2*disrow;
        if (k>=WordNO*32)k -= WordNO*32;
        BUFF[2*s]=HZ[k];
        BUFF[2*s+1]=HZ[k+1];
    }
}

/***** send the data of a row *****/
void sendoneline( )
{
    byte s;

```

```
byte inc;
if(col<8)inc=0;else inc=1;
  for(s=2*ScreenNO+inc;s>=1+inc;s--)
  {
    sendUtra(two_onechar(BUFF[s-1],BUFF[s]));
  }
}
/***** generate data which makes word moving *****/
byte two_onechar(byte h1,byte h2)
{
byte temp,tempcol;
if(col<8) tempcol=col;
else tempcol=col-8;
temp=(h1<<tempcol)|(h2>>(8-tempcol));
//temp=255-temp;
return temp;
}
/*****/
void sendPin(byte j)
{
  for (int n=0;n<4;n++)
  {
    if((0x01&j) < 0x01)
      {digitalWrite(Dpin[n],LOW);}
    else
      {digitalWrite(Dpin[n],HIGH);}
    j>>=1;
  }
}
/*****/
void sendUtra(byte j)
{
  for (int n=0;n<8;n++)
  {
    digitalWrite(SCK,LOW);
    if((0x01&j) < 0x01)
      {digitalWrite(SER,LOW);}
    else
      {digitalWrite(SER,HIGH);}
    j=j>>1;
    digitalWrite(SCK,HIGH);
    //delay(1);
  }
}
```

The program above is displaying words rolling from left to right.
If you want to roll the words from right to left, please replace the functions above with following code:

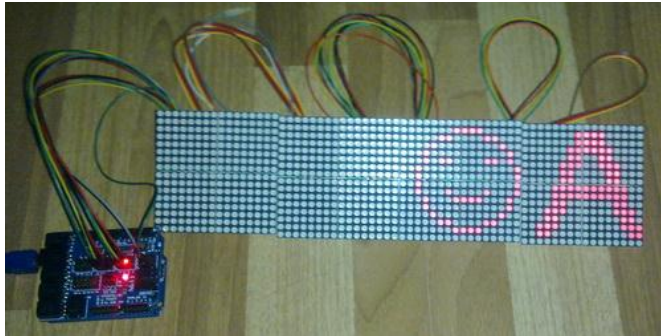
```

/***** load the data of a row *****/
void loadoneline( )
{
byte s;
unsigned int k;
    for(s=0;s< ScreenNO+1;s++)
    {
        k = wordnumber+32*s+2*disrow;
        if (k>=WordNO*32)k -= WordNO*32;
        BUFF[2*s+1]=HZ[k];
        BUFF[2*s]=HZ[k+1];
    }
}

/***** send the data of a row *****/
void sendoneline( )
{
byte s;
byte inc;
if(col<8)inc=0;else inc=1;
    for(s=1+inc;s<=2*ScreenNO+inc;s++)
    {
        sendUtra(two_onechar(BUFF[s-1],BUFF[s]));
    }
}

/***** generate data which makes word moving *****/
byte two_onechar(byte h1,byte h2)
{
byte temp,tempcol;
if(col<8) tempcol=col;
else tempcol=col-8;
temp=(h1>>tempcol)|(h2<<(8-tempcol));
return temp;
}

```

In fact, you can assemble a screen of any size you want (times of 16), large square screen or long screen. And you can program it to display anything you want.

The following shows coding letter "E". This panel is the 16*16 LED matrix. In fact, the area of "1" or "0" is a LED. "1" means OFF, while "0" means ON. I add red background to the ON area. You can see it is the letter "E".

From this example, you can understand how to code the letter:

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0xff	0xff
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0x80	0x01
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0x80	0x01
1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0x9f	0xff
1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0x9f	0xff
1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0x9f	0xff
1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0x9f	0xff
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0x80	0x03
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0x80	0x03
1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0x9f	0xff
1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0x9f	0xff
1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0x9f	0xff
1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0x9f	0xff
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0x80	0x01
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0x80	0x01
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0xff	0xff

[You can lick here to get the schematic](#)
[Also you can watch our vedio on Youtube](#)