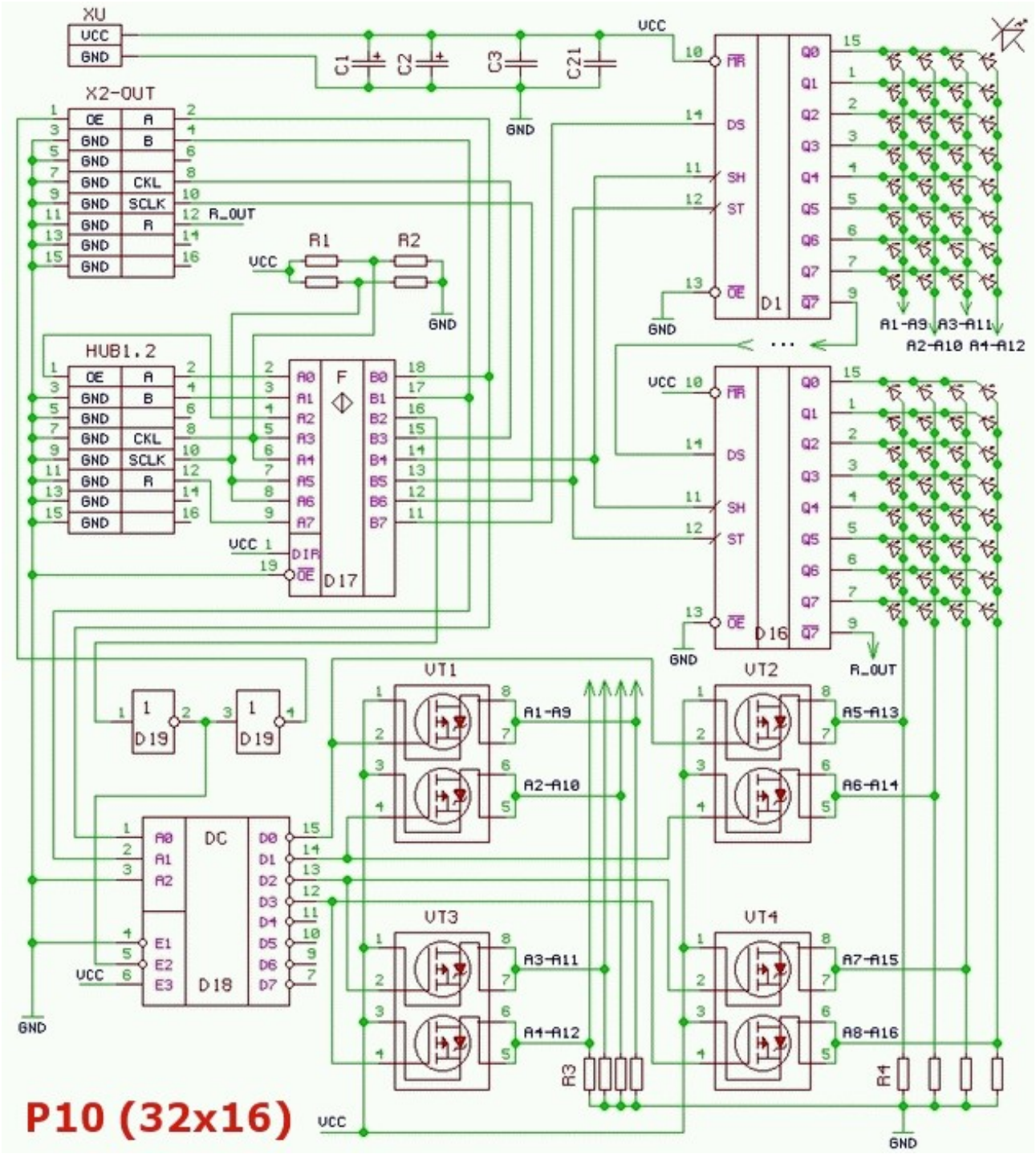
P10(1R)-V701 Panel Led

Schematic diagram of the module P10.



Schematic diagram "rubbings" with two modules P10: P10 (1R) -V701C (red LED) and P10 (1R) -V901A (green).

In the circuit 16 registers 74HC595 (ms D1-D16), the buffer element 74HC245 (ms D17), the decoder 74HC138 (ms D18), the inverter 74HC04 (ms D19) and four assembly P-channel transistors IRF7314 (or 4953, VT1-VT4). Also in the scheme have R1-1 ohm resistor assembly, R2- 4,7 kW; R3; R4 - 100 ohms, two electrolyte C1; C2 - 470,0 nineteen capacitors C3-C21 - 0,1 mF installed on the conclusions of the power circuits.

Input signals from an external device to the connector coming HUB1.2, and to the X2-OUT connector can be connected to the next module trail P10. All major connector HUB1.2 signals pass through a buffer D17 ms, wherein each of the signals SCLK and CKL goes through two lines; Some go to the conclusions of 11, 12 registers (ms D1-D16) of the module, the second - on the output connector to work with the following modules. The input data signal R, having passed through the buffer element ms D17, to the input data of the

first register (output 14 ms D1). With 9 ms D1 output data is input to the next register. The picture no circuit fourteen registers ms D2-D15, but in the circuit they are connected in series and each runs with its own group of 32 LEDs. The output data from the R-OUT output of the last register 9 (ms D16) fed to the input of the R output connector X2-OUT.

To turn the LED requires a zero level at the output of the register and the opening of one of the transistors VT1-VT4. Scan mode the module P10 - 1/4, that is, at each time to open one of the four transistors or may illuminate a group of 128 LEDs connected to the outputs of all registers. The control signal transistors is determined by the states of lines A, B connected through a buffer element ms D17 decoder to address inputs (pins 1, 2 ms D18). By the conclusion of the decoder 5 ms D19 through an inverter connected signal OE, which determines the brightness of the LED module.

When working with one module information on the R input jack HUB1.2 with the control unit takes four 128-bit parcels, each time exposing the lines A and B, the next address. The shift register data is clocked by the positive edge of the signal CKL, and the output data is transferred to the register on the rising edge of the signal SCLK.

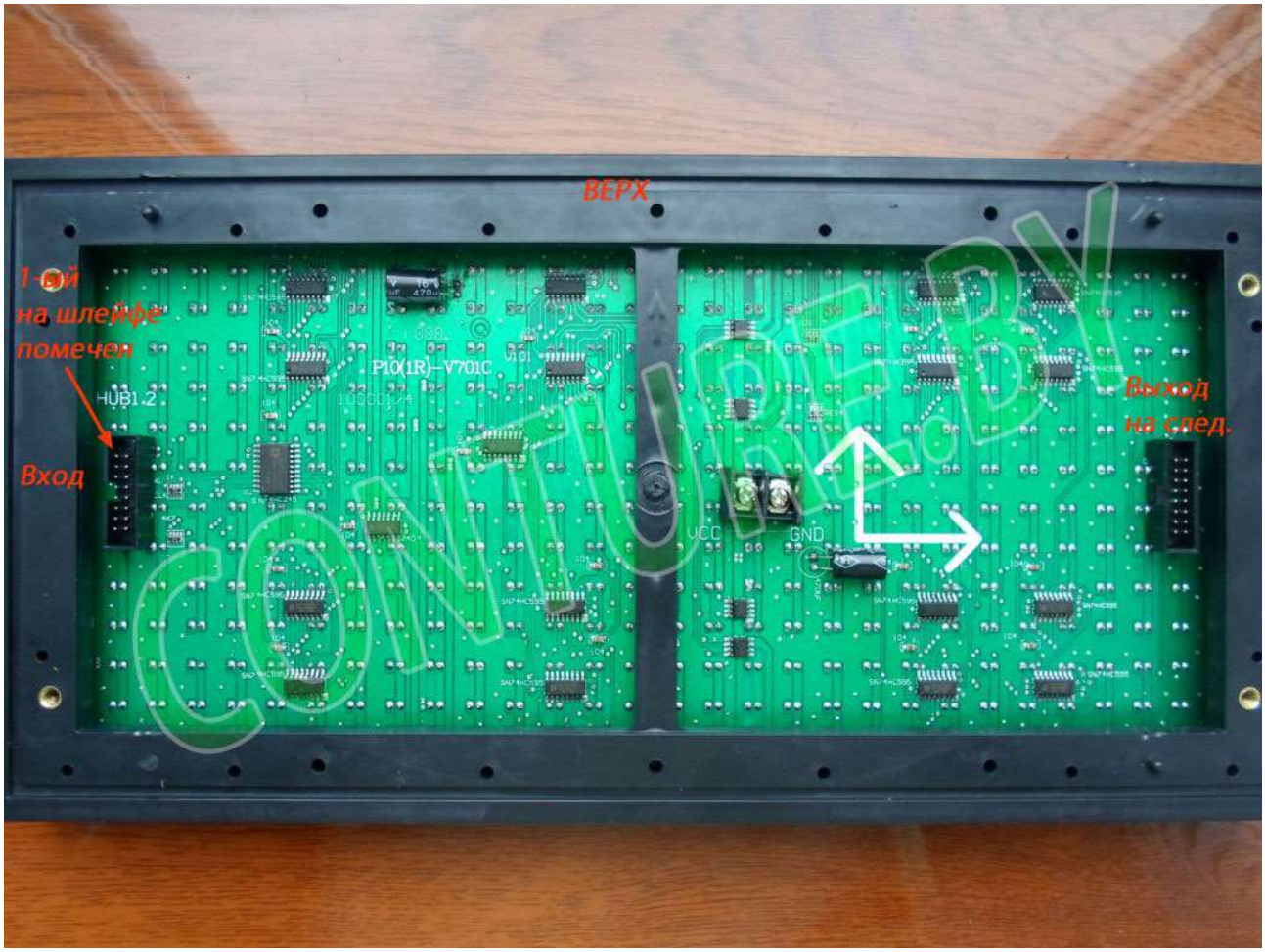
Source: http://sotvorimvmeste.ru/viewtopic.php?f=33&t=194

Application:

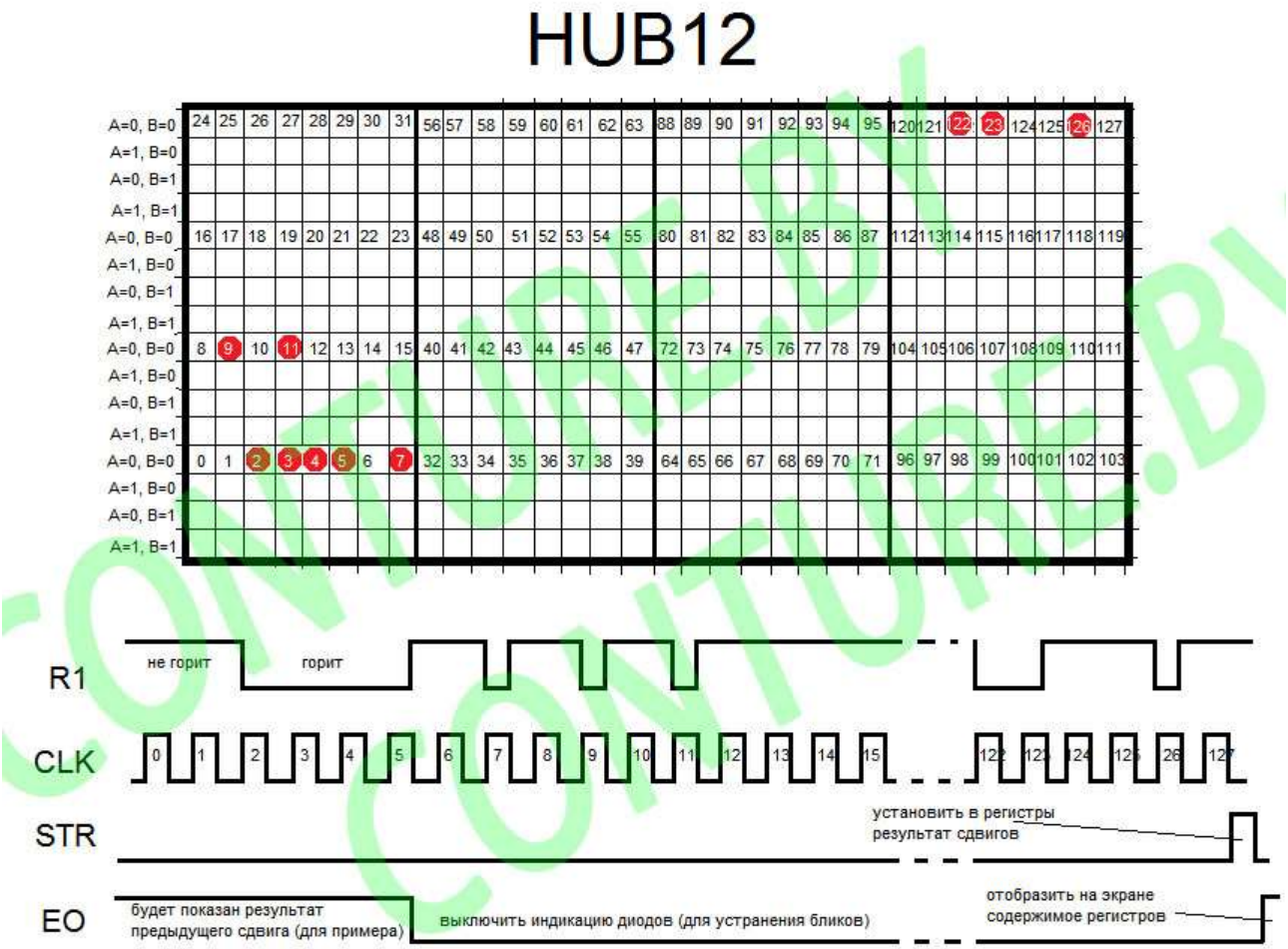
Temperature display described in this article is intended for outdoor installations and measuring the ambient temperature. Cable length to the sensor can be up to 20 meters (longer is not tested), which allows you to spread the sensor itself and the display. Measuring range: -40 to +50. Measurement accuracy according to the sensor datasheet DS1820 = +/- 0.5 C. The current consumption of about 1.3 A.

LED modules were purchased from a Chinese online store. Color options are different - green, red, blue, and even a lunar glow. The truth is, buying these modules task was to study protocols HUB08, HUB12, because the information in the network it is extremely small (or rather even do not).

Thus, the modules consist of an array of LEDs (32x16 = 512 units), and 16 pieces of shift registers 74HC595.



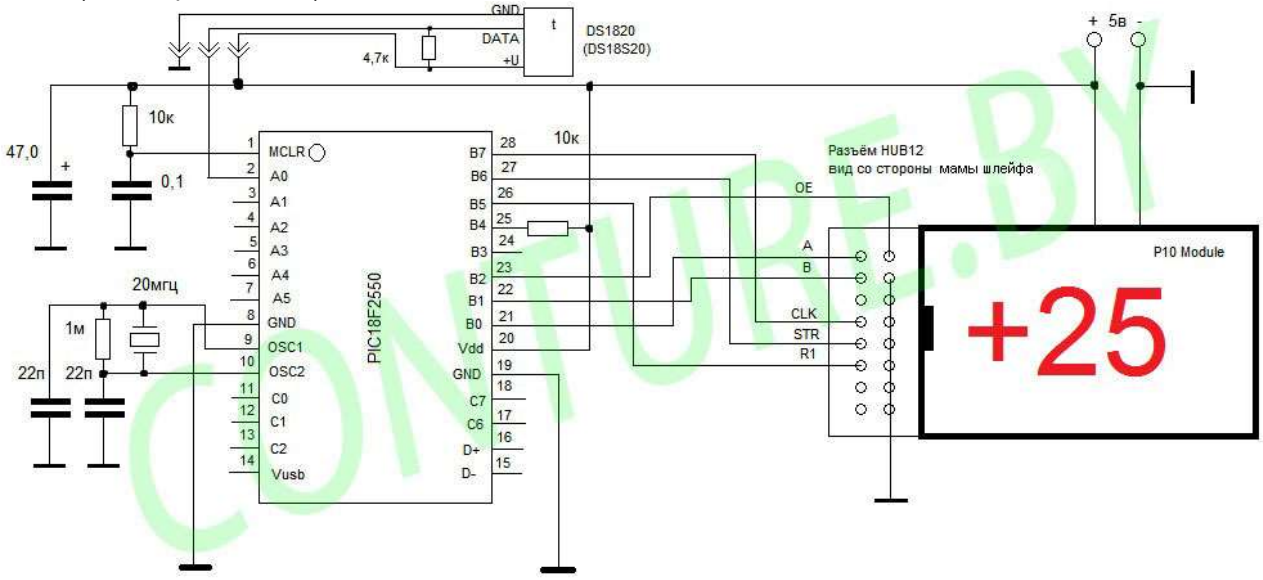
Studying the principle of the scheme came to this algorithm:



Thus, only four groups of shift registers. Each group will be activated by the respective levels of the ports A and B. The display dynamic, so is lit set group - the shift register on the next group, etc.

Initially the idea was to collect data on the running line modules, but the CPU PIC18F2550 showed that 2 Kbytes of RAM for this is not enough - the work is necessary to conduct the type double word. Buy for experiments that something serious does not make sense, because the Chinese are ready controllers for $ 20-30.

He came to the idea of the development of the thermometer, which is often set to attract attention to the outlets (kiosks, pharmacies).



Download the firmware for the microcontroller can be on this link tempP10led.

http://conture.by/wp-content/uploads/2013/08/tempP10led.zip

Sources do not ask - it would make life more difficult students-))). But the idea will show, but after watching the video.

https://www.youtube.com/watch?v=iuFS1QgHzC0

How to screen the inscription DEVLAB.BY, which you have seen in the greeting on the screen? It's simple - you need to sit down and figure out the code below ...

```
dword hello [16] = {
    0b00000000000000000000000000000000,
    0b00000000000000000000000000000000,
    0b11100111101000101000001100111100,
    0b10010100001000101000010010100010,
    0b10010111001000101000010010111100,
    0b10010100000101001000011110100010,
    0b11100111100010001111010010111100,
    0b00000000000000000000000000000000,
    0b00000000000000000000000000000000,
    0b00000000000011110010001000000000,
    0b00000000000010001001010000000000,
    0b00000000000011110000100000000000,
    0b00000000000010001000100000000000,
    0b00000000010111100001000000000000,
    0b00000000000000000000000000000000,
    0b00000000000000000000000000000000
};

ServiceReq void (void) {// function is called every 50 ms
  static word led_count = 0;
  byte T [4] = {} 32,24,16,8;
  unsigned char i = 0;
  unsigned char x = 0;
  unsigned char xx = 0;

  if (led_count == 500) {
    LATBbits.LATB2 = 1; // enable the module

    is (xx = 0; xx0; x -) {
          ShowLines (((x-1) * 4) + XX, t [i]); // Set registers group
        }
      }
      if (xx == 0) {
        LATBbits.LATB0 = 0; LATBbits.LATB1 = 0; // The first group of registers
      } else if (xx == 1) {
        LATBbits.LATB0 = 1; LATBbits.LATB1 = 0; // The second group of registers
      } else if (xx == 2) {
        LATBbits.LATB0 = 0; LATBbits.LATB1 = 1; // The third group of registers
      } else {
        LATBbits.LATB0 = 1; LATBbits.LATB1 = 1; // The fourth group of registers
      }
      LATBbits.LATB6 = 1; LATBbits.LATB6 = 0; // signal STR
    }

    LATBbits.LATB2 = 0; // turn off the module

    led_count = 0U;
  }
}

void ShowLines (y byte, byte x)
{
```

```
  Byte;
  is (a = x; a> (x-8); a -) {
    LATBbits.LATB5 ubit = (y (a-1)); // The signal is set R1
    LATBbits.LATB7 = 1; LATBbits.LATB7 = 0; // signal  CLK
  }
}

BOOL ubit (byteB byte, byte bit0)
{
    if (((hello [byteB] >> bit0) & 1)> 0)
    {
        return 0;
    } else {
        return 1;
    }
}
```


Source: http://conture.by/post/1100

Translated from russian to english by Google.

February 2016.