

```

1  /*****
2  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
3  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
4  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
5  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
6  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
7  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
8  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
9  *****/
10
11 #include <avr/pgmspace.h>
12 #include "Font.h" //Font[char][x,y,0..x] larghezza, altezza, massimo 5 disegno lettera
13 // Font.h libreria scritta appositamente
14 #include "LedControl.h"
15 #define lc_datain 2
16 #define lc_cs 3
17 #define lc_clock 4
18 #define num_disp 4
19 #define maxColonne num_disp*8-1
20 LedControl led = LedControl(lc_datain, lc_clock, lc_cs, num_disp);
21 byte intensita = 3;
22
23 void setup() {
24     for (int i = 0; i < num_disp; i++) {
25         led.shutdown(i, false);
26         led.setIntensity(i, intensita);
27         led.clearDisplay(i);
28     }
29 }
30
31 void loop() {
32     String testo = "U 73.5% ";
33     printString(0, testo);
34     delay(4000);
35
36     testo = "T 20.3c ";
37     printString(0, testo);
38     delay(4000);
39
40     testo = "P 1005 ";
41     printString(0, testo);
42     delay(4000);
43
44     testo = " 21:35 ";
45     printString(0, testo);
46     delay(4000);
47 } // ===== loop fine
48
49 void printChar(int *pos, char carattere) {
50     char c = carattere - 32;
51     byte buffer[7];
52     memcpy_P(buffer, Font + 7 * c, 7);
53     int larghezza = buffer[0];
54     for (int i = 0; i < larghezza; i++) {
55         if ((*pos) + i > maxColonne) break;
56         int disp = ((*pos) + i) / 8;
57         int colonna = ((*pos) + i) % 8;
58         led.setColumn(disp, 7 - colonna, buffer[2 + i]);
59     }
60     (*pos) = (*pos) + larghezza + 1;
61 }
62
63 void printString(int pos, String testo) {
64     int l = testo.length();
65     for (int j = 0; j < l; j++) printChar(&pos, testo[j]);
66 }
67
68 // fine programma
69

```

```
1  PROGMEM const byte Font[] = {
2  3, 8, B00000000, B00000000, B00000000, B00000000, B00000000, // space
3  1, 8, B01011111, B00000000, B00000000, B00000000, B00000000, // !
4  3, 8, B00000011, B00000000, B00000011, B00000000, B00000000, // "
5  5, 8, B00010100, B00111110, B00010100, B00111110, B00010100, // #
6  4, 8, B00100100, B01101010, B00101011, B00010010, B00000000, // $
7  5, 8, B01100011, B00010011, B00001000, B01100100, B01100011, // %
8  5, 8, B00110110, B01001001, B01010110, B00100000, B01010000, // &
9  1, 8, B00000011, B00000000, B00000000, B00000000, B00000000, // '
10 3, 8, B00011100, B00100010, B01000001, B00000000, B00000000, // (
11 3, 8, B01000001, B00100010, B00011100, B00000000, B00000000, // )
12 5, 8, B00101000, B00011000, B00001110, B00011000, B00101000, // *
13 5, 8, B00001000, B00001000, B00111110, B00001000, B00001000, // +
14 2, 8, B10110000, B01110000, B00000000, B00000000, B00000000, // ,
15 4, 8, B00001000, B00001000, B00001000, B00001000, B00000000, // -
16 2, 8, B01100000, B01100000, B00000000, B00000000, B00000000, // .
17 4, 8, B01100000, B00011000, B00000110, B00000001, B00000000, // /
18 4, 8, B00111110, B01000001, B01000001, B00111110, B00000000, // 0
19 3, 8, B01000010, B01111111, B01000000, B00000000, B00000000, // 1
20 4, 8, B01100010, B01010001, B01001001, B01000110, B00000000, // 2
21 4, 8, B00100010, B01000001, B01001001, B00110110, B00000000, // 3
22 4, 8, B00011000, B00010100, B00010010, B01111111, B00000000, // 4
23 4, 8, B00100111, B01000101, B01000101, B00111001, B00000000, // 5
24 4, 8, B00111110, B01001001, B01001001, B00110000, B00000000, // 6
25 4, 8, B01100001, B00010001, B00001001, B00000111, B00000000, // 7
26 4, 8, B00110110, B01001001, B01001001, B00110110, B00000000, // 8
27 4, 8, B00000110, B01001001, B01001001, B00111110, B00000000, // 9
28 1, 8, B00100100, B00000000, B00000000, B00000000, B00000000, // :
29 2, 8, B10000000, B01010000, B00000000, B00000000, B00000000, // ;
30 3, 8, B00010000, B00101000, B01000100, B00000000, B00000000, // <
31 3, 8, B00010100, B00010100, B00010100, B00000000, B00000000, // =
32 3, 8, B01000100, B00101000, B00010000, B00000000, B00000000, // >
33 4, 8, B00000010, B01011001, B00001001, B00000110, B00000000, // ?
34 5, 8, B00111110, B01001001, B01010101, B01011101, B00001110, // @
35 4, 8, B01111110, B00010001, B00010001, B01111110, B00000000, // A
36 4, 8, B01111111, B01001001, B01001001, B00110110, B00000000, // B
37 4, 8, B00111110, B01000001, B01000001, B00100010, B00000000, // C
38 4, 8, B01111111, B01000001, B01000001, B00111110, B00000000, // D
39 4, 8, B01111111, B01001001, B01001001, B01000001, B00000000, // E
40 4, 8, B01111111, B00001001, B00001001, B00000001, B00000000, // F
41 4, 8, B00111110, B01000001, B01001001, B01111010, B00000000, // G
42 4, 8, B01111111, B00001000, B00001000, B01111111, B00000000, // H
43 3, 8, B01000001, B01111111, B01000001, B00000000, B00000000, // I
44 4, 8, B00110000, B01000000, B01000001, B00111111, B00000000, // J
45 4, 8, B01111111, B00001000, B00010100, B01100011, B00000000, // K
46 4, 8, B01111111, B01000000, B01000000, B01000000, B00000000, // L
47 5, 8, B01111111, B00000010, B00001100, B00000010, B01111111, // M
48 5, 8, B01111111, B00000100, B00001000, B00010000, B01111111, // N
49 4, 8, B00111110, B01000001, B01000001, B00111110, B00000000, // O
50 4, 8, B01111111, B00001001, B00001001, B00000110, B00000000, // P
51 4, 8, B00111110, B01000001, B01000001, B10111110, B00000000, // Q
52 4, 8, B01111111, B00001001, B00001001, B01110110, B00000000, // R
53 4, 8, B01000110, B01001001, B01001001, B00110010, B00000000, // S
54 5, 8, B00000001, B00000001, B01111111, B00000001, B00000000, // T
55 4, 8, B00111111, B01000000, B01000000, B00111111, B00000000, // U
56 5, 8, B00001111, B00110000, B01000000, B00110000, B00001111, // V
57 5, 8, B00111111, B01000000, B00111000, B01000000, B00111111, // W
58 5, 8, B01100011, B00010100, B00001000, B00010100, B01100011, // X
59 5, 8, B00000111, B00001000, B01110000, B00001000, B00000111, // Y
60 4, 8, B01100001, B01010001, B01001001, B01000111, B00000000, // Z
61 2, 8, B01111111, B01000001, B00000000, B00000000, B00000000, // [
62 4, 8, B00000000, B00000110, B00011000, B01100000, B00000000, // \ backslash
63 2, 8, B01000001, B01111111, B00000000, B00000000, B00000000, // ]
64 3, 8, B00000010, B00000001, B00000010, B00000000, B00000000, // ^
65 4, 8, B01000000, B01000000, B01000000, B01000000, B00000000, // _
66 2, 8, B00000001, B00000010, B00000000, B00000000, B00000000, // `
67 4, 8, B00100000, B01010100, B01010100, B01111000, B00000000, // a
68 4, 8, B01111111, B01000100, B01000100, B00111000, B00000000, // b
69 4, 8, B00111000, B01000100, B01000100, B00101000, B00000000, // c
70 4, 8, B00111000, B01000100, B01000100, B01111111, B00000000, // d
71 4, 8, B00111000, B01010100, B01010100, B00011000, B00000000, // e
72 3, 8, B00000100, B01111110, B00000101, B00000000, B00000000, // f
73 4, 8, B10011000, B10100100, B10100100, B01111000, B00000000, // g
74 4, 8, B01111111, B00000100, B00000100, B01111000, B00000000, // h
75 3, 8, B01000100, B01111101, B01000000, B00000000, B00000000, // i
```

65

```
76
77 void destrasinistra(int pausa, unsigned int durata) {
78     unsigned long tempo = millis();
79     while (millis() < tempo + durata) {
80         digitalWrite(2, LOW); // gialli
81         digitalWrite(3, LOW);
82         digitalWrite(4, HIGH);
83         digitalWrite(5, LOW);
84         digitalWrite(6, HIGH); // verdi
85         digitalWrite(7, HIGH);
86         digitalWrite(8, LOW);
87         digitalWrite(9, LOW);
88         digitalWrite(10, HIGH); // rossi
89         digitalWrite(11, HIGH);
90         digitalWrite(12, LOW);
91         digitalWrite(13, LOW);
92         digitalWrite(14, HIGH); // BLU
93         digitalWrite(15, HIGH);
94         digitalWrite(16, LOW);
95         digitalWrite(17, LOW);
96         digitalWrite(18, LOW); // bianco
97         delay(pausa);
98         digitalWrite(2, LOW);
99         digitalWrite(3, LOW);
100        digitalWrite(4, LOW);
101        digitalWrite(5, HIGH);
102        digitalWrite(6, LOW);
103        digitalWrite(7, LOW);
104        digitalWrite(8, HIGH);
105        digitalWrite(9, HIGH);
106        digitalWrite(10, LOW);
107        digitalWrite(11, LOW);
108        digitalWrite(12, HIGH);
109        digitalWrite(13, HIGH);
110        digitalWrite(14, LOW);
111        digitalWrite(15, LOW);
112        digitalWrite(16, HIGH);
113        digitalWrite(17, HIGH);
114        digitalWrite(18, LOW);
115        delay(pausa);
116    }
117    off();
118 }
```

```
1 // progetto: display_7segmenti_3digit_potenziometro
2 // descrizione: visualizza su un display 3 digit un numero da 1 a 999 letto da un potenziometro
3 // utilizza una macchina a STATI
4 // shields: display 7 digit anodo comune, 8 resistenze 220 ohm, potenziometro 10 k
5 // librerie:
6 // data: 11-05-2025
7 // autore: www.aldopi.it
8
9 #define debug 0
10
11 #define segA 12
12 #define segB 6
13 #define segC 8
14 #define segD 9
15 #define segE 10
16 #define segF 11
17 #define segG 7
18 #define segDP 3
19
20 #define digit1 5
21 #define digit2 4
22 #define digit3 3
23
24 #define potenziometro A0
25 int pot = 0;
26 int digit = 0;
27 int n = 0;
28
29 unsigned long tempo;
30 unsigned long misura;
31
32
33 void setup() {
34     if (debug) Serial.begin(115200);
35     if (debug) Serial.println("avvio");
36     pinMode(segA, OUTPUT);
37     pinMode(segB, OUTPUT);
38     pinMode(segC, OUTPUT);
39     pinMode(segD, OUTPUT);
40     pinMode(segE, OUTPUT);
41     pinMode(segF, OUTPUT);
42     pinMode(segG, OUTPUT);
43     pinMode(segDP, OUTPUT);
44     pinMode(digit1, OUTPUT);
45     pinMode(digit2, OUTPUT);
46     pinMode(digit3, OUTPUT);
47
48     digitalWrite(digit1, 0);
49     digitalWrite(digit2, 0);
50     digitalWrite(digit3, 1);
51     tempo = millis();
52     misura = millis();
53     pot = analogRead(potenziometro);
54     if (debug) Serial.println("run");
55 }
56
57
58 void loop() {
59     if (millis() - tempo > 5) {
60         scrivi();
61         tempo = millis();
62     }
63     if (millis() - misura > 300) {
64         misura = millis();
65         pot = analogRead(potenziometro);
66         if (debug) Serial.println(pot);
67     }
68 }
69
70
```

```

71 void scrivi() {
72     if (digit == 2) digit = 0;
73     else digit++;
74     switch (digit) {
75     case 0:
76         digitalWrite(digit1, 1);
77         digitalWrite(digit2, 0);
78         digitalWrite(digit3, 0);
79         n = (int)pot / 100;
80
81         break;
82     case 1:
83         digitalWrite(digit1, 0);
84         digitalWrite(digit2, 1);
85         digitalWrite(digit3, 0);
86         n = (int)pot / 10;
87
88         break;
89     case 2:
90         digitalWrite(digit1, 0);
91         digitalWrite(digit2, 0);
92         digitalWrite(digit3, 1);
93         n = (int)pot;
94         break;
95     }
96
97     if (pot > 999 or pot ==0) {
98         spento();
99     } else {
100         n = n % 10;
101         numero(n);
102     }
103 }
104

```

```

1 //      file: max1219-8x8_joystick.ino
2 //      descrizione: muovere un pixel con il joystick e conto alla rovescia
3 //      shields: max9212 8x8, joystick
4 //      librerie: ledcontrol.h v1.0.6
5 //      versione-data: 16 marzo 2024
6 //      autore: Aldo Pizzeghella - www.aldopi.it
7
8 #define lc_datain 4
9 #define lc_cs 3
10 #define lc_clock 2
11
12 #include "LedControl.h"
13 LedControl lc = LedControl(lc_datain, lc_clock, lc_cs, 1);
14
15
16 int x = 2, y = 2;
17 int joystick;
18 #define JOYSTICK_PORT_X A0
19 #define JOYSTICK_PORT_Y A1
20 #define JOYSTICK_SWITCH 5
21 #define trigPin 8
22 #define echoPin 9
23 int attesa = 10;
24 int i;
25
26 byte font[10][7] = {
27     B01111000, //0
28     B10001000,
29     B10011000,
30     B10101000,
31     B11001000,
32     B10001000,
33     B01110000,
34
35     B01000000, //1
36     B11000000,
37     B01000000,
38     B01000000,
39     B01000000,
40     B01000000,
41     B11100000,
42
43     B01110000, //2
44     B10001000,
45     B00001000,
46     B00010000,
47     B00100000,
48     B01000000,
49     B11111000,
50
51     B11111000, //3
52     B00010000,
53     B00100000,
54     B00010000,
55     B00001000,
56     B10001000,
57     B01110000,
58
59     B00010000, //4
60     B00110000,
61     B01010000,
62     B10010000,
63     B11111000,
64     B00010000,
65     B00010000,
66
67     B11111000, //5
68     B10000000,
69     B11110000,
70     B00001000,
71     B00001000,
72     B10001000,
73     B01110000,
74
75     B00110000, //6
76     B01000000,
77     B10000000,
78     B11110000,
79     B10001000,
80     B10001000,
81     B01110000,
82
83     B11111000, //7
84     B10001000,
85     B00001000,
86     B00010000,
87     B00100000,
88     B00100000,
89     B00100000,
90
91     B01110000, //8
92     B10001000,
93     B10001000,
94     B01110000,
95     B10001000,
96     B10001000,
97     B01110000,
98
99     B01110000, //9
100    B10001000,
101    B10001000,
102    B01111000,
103    B00001000,
104    B00010000,
105    B01100000,
106 };
107

```

```
108 void setup() {
109     pinMode(JOYSTICK_SWITCH, INPUT_PULLUP);
110     lc.shutdown(0, false);
111     lc.setIntensity(0, 3);
112     lc.clearDisplay(0);
113
114     pinMode(trigPin, OUTPUT);
115     pinMode(echoPin, INPUT);
116 }
117
118 void lettura() {
119     joystick = analogRead(JOYSTICK_PORT_X);
120     x = map(joystick, 10, 950, 0, 7);
121     joystick = analogRead(JOYSTICK_PORT_Y);
122     y = map(joystick, 10, 950, 0, 7);
123 }
124
125
126 void loop() {
127     lc.setLed(0, y, x, 0);
128     lettura();
129     lc.setLed(0, y, x, 1);
130     if (digitalRead(JOYSTICK_SWITCH) == 0) contatore();
131     delay(100);
132 }
133
134 void contatore() {
135     for (int i = 9; i >= 0; i--) {
136         lc.setRow(0, 0, font[i][0]);
137         lc.setRow(0, 1, font[i][1]);
138         lc.setRow(0, 2, font[i][2]);
139         lc.setRow(0, 3, font[i][3]);
140         lc.setRow(0, 4, font[i][4]);
141         lc.setRow(0, 5, font[i][5]);
142         lc.setRow(0, 6, font[i][6]);
143         delay(500);
144     }
145     lc.clearDisplay(0);
146 }
147 // fine programma
148
```

```

1 // file: orologio_rotante_lancette.ino
2 // descrizione: orologio rotante con 13 leds
3 // shields:
4 // librerie:
5 // versione-data: 15-09-2021
6 // autore: Aldo Pizzeghella - www.AldoPi.it
7
8 // led rossi esterni: portC pin 0,1,2,3,4 - digital port 14,15,16,17,18 - a0,a1,a2,a3,a4
9 // led blu: portB pin 0,1,2,3,4 portD pin 5,6 - digital port 8,9,10,11,12,5,6
10
11 int tempoattesa = 180;
12 int tempoon = 60;
13 #define pinsensore 2 // int 0
14 volatile unsigned long rotazione = 1;
15 volatile unsigned long ciclo = 0;
16 volatile byte iniziogiro = 1;
17
18 byte n, pos;
19
20 unsigned long tempo = 0;
21
22 byte ora = 4; // 0..11 start time
23 byte minuti = 15; // 0..59
24 byte secondi = 00; // 0..59
25
26 void setup() { // ----- setup
27     pinMode(pinsensore, INPUT_PULLUP); // sensore HALL
28     pinMode(14, OUTPUT); // leds ROSSI PORTC
29     pinMode(15, OUTPUT);
30     pinMode(16, OUTPUT);
31     pinMode(17, OUTPUT);
32     pinMode(18, OUTPUT);
33     pinMode( 8, OUTPUT); // leds blu PORTB
34     pinMode( 9, OUTPUT);
35     pinMode(10, OUTPUT);
36     pinMode(11, OUTPUT);
37     pinMode(12, OUTPUT);
38     pinMode( 5, OUTPUT);
39     pinMode( 6, OUTPUT);
40     pinMode(13, OUTPUT);
41
42     PORTB = 0;
43     PORTD &= B00011100;
44     PORTC = 0;
45     tempo = millis();
46     attachInterrupt(0, rot, FALLING);
47     ciclo = micros();
48 }
49
50
51 void loop() {
52     while (iniziogiro) {}; // attesa inizio giro
53
54     iniziogiro = 1;
55     tempoattesa = rotazione / 540;
56     tempoon = rotazione / 450;
57     if (millis() - tempo > 1000) { // aggirno l'orologio ogni secondo
58         tempo = millis();
59         secondi++;
60         if (secondi == 60) {
61             secondi = 0;
62             minuti++;
63         }
64         if (minuti == 60) {
65             minuti = 0;
66             ora++;
67         }
68         if (ora == 12) {
69             ora = 0;
70         }
71     }

```

```

72
73 pos = 60;
74 PORTB = 0;
75 PORTD &= B11111100;
76 PORTC = 0;
77 for (n = 240; n > 0; n--) { // da 60 a 0 perché il motore gira antiorario
78     PORTC = B00010000;
79     if (n % 4 == 0) {
80         pos--;
81         PORTC |= B00001000;
82         // segna i secondi
83         if ((pos % 5) == 0) PORTC |= B00000110; // segna i 5 minuti
84         if ((pos % 15) == 0) PORTC |= B00000111; // segna le 15 minuti
85
86         if ((pos == ora * 5) ) { // lancetta ore
87             PORTB |= B00011000;
88             PORTD |= B01100000;
89         }
90         if (pos == minuti) { // lancetta minuti
91             PORTB |= B00011111;
92             PORTD |= B01100000;
93         }
94         if (pos == secondi) { // lancetta secondi
95             PORTB |= B00000000;
96             PORTD |= B01000000;
97         }
98     }
99     delayMicroseconds(tempoon); // tempo accensione led
100     PORTB = 0;
101     PORTD &= B10011100;
102     PORTC = 0;
103     delayMicroseconds(tempoattesa); // tempo intervallo tra un bit e successivo
104 }
105 }
106
107 void rot() {
108     rotazione = micros() - ciclo;
109     ciclo = micros();
110     iniziogiro = 0;
111 }
112 // fine programma
113

```

```

1  //      file: ws2812b_MSEQ7_Equalizzatore_8x8.ino
2  //  descrizione: equalizzatore grafico 8 bande 1 canale
3  //      shields: led ws2812b 8x8; Equalizzatore MSEQ7
4  //      librerie: Adafruit_NeoPixel (1.3)
5  //      data: 11-05-2025
6  //      autore: www.aldopi.it/bottegadelcodice
7
8  // se si alimenta la striscia con arduino
9  // NON SUPERARE MAI IL VALORE 120 come somma dei tre leds massimo 3-4 led accesi
10
11 /* PIN:
12 Reset
13 strobe
14 data
15 gnd
16 v+ 5v
17 */
18
19 #include <Adafruit_NeoPixel.h>
20 #define LED_PIN 2    // collegati alla porta 12
21 #define LED_NUM 64   // 30 leds
22 #define LED_ROW 8
23 #define debug 0
24 Adafruit_NeoPixel strip = Adafruit_NeoPixel(LED_NUM, LED_PIN, NEO_RGB + NEO_KHZ800);
25 unsigned long rosso = strip.Color(0, 7, 0);
26 unsigned long verde = strip.Color(7, 0, 0);
27 unsigned long blu = strip.Color(0, 8, 8);
28 unsigned long giallo = strip.Color(8, 8, 0);
29 unsigned long bianco = strip.Color(10, 10, 10);
30 unsigned int tempo;
31
32
33 // setup MSEQ7
34 #define volumemax 400
35 int strobePin = 3;    // Strobe Pin
36 int resetPin = 4;    // Reset Pin
37 int outPin = A0;     // Output Pin
38 int level[7];        // vettore contenete i livelli del suono suddivise in 7 bande
39 int segnale[7];       // (63-160-400-1000-2500-6250-16000Hz)
40 int bias = 0;
41 int valore;
42
43
44 void setup() {
45     strip.begin();
46     strip.show();
47     if (debug) Serial.begin(115200);
48     // MSEQ7 inizializzo
49     pinMode(strobePin, OUTPUT);
50     pinMode(resetPin, OUTPUT);
51     digitalWrite(resetPin, HIGH); // reset chip
52     delay(1);
53     digitalWrite(resetPin, LOW);
54     digitalWrite(strobePin, HIGH);
55     strip.setPixelColor(5, verde);
56     strip.show();
57     delay(250);
58     strip.setPixelColor(5, 0);
59     strip.show();
60     delay(250);
61 }
62

```

```

63 void loop() {
64     lettura();
65     for (int j = 0; j < 7; j++) {
66         sequenza(j, segnale[j]);
67     }
68     strip.show();
69     delay(200);
70 }
71
72 void sequenza(int riga, int led) {
73     for (int i = 0; i < LED_ROW; i++) {
74         if (i < led) {
75             if (i < 5) strip.setPixelColor(riga * LED_ROW + i, verde);
76             else if (i < 7) strip.setPixelColor(riga * LED_ROW + i, giallo);
77             else strip.setPixelColor(riga * LED_ROW + i, rosso);
78         } else strip.setPixelColor(riga * LED_ROW + i, 0);
79     }
80 }
81
82 void lettura() {
83     // lettura dei 7 valori
84     digitalWrite(strobePin, HIGH);
85     delayMicroseconds(200);
86     for (int i = 0; i < 7; i++) {
87         digitalWrite(strobePin, LOW);
88         delayMicroseconds(200);
89         level[i] = analogRead(outPin);
90         digitalWrite(strobePin, HIGH);
91         delayMicroseconds(200);
92     }
93     // conversione dei valori nel range 0..30 per la striscia led
94     bias = level[0]; // ricerca valore minimi (rumore)
95     for (int i = 1; i < 7; i++)
96         if (level[i] < bias) bias = level[i];
97     for (int i = 0; i < 7; i++) // rimozione del rumore e conversione scala
98         segnale[i] = map(level[i] - bias, 0, volumemax, 0, 10);
99
100     if (debug) {
101         // stampa sulla porta seriale valori letti e convertiti
102         for (int i = 0; i < 7; i++) {
103             Serial.print(segnale[i]);
104             Serial.print("\t");
105         }
106         Serial.println();
107     }
108 }
109 //FINE PROGRAMMA

```

```

1  //      file: ws2812_30x8_orologio.ino
2  //      progetto: Orologio da parete
3  //      descrizione: realizzare un orologio su una matrice led 30x8
4  //      shields: led ws2812 30x8 240 pixel
5  //      data: 16-03-2024 - 1.a
6  //      autore: www.aldopi.it
7
8  // se si alimenta la striscia con arduino
9  // NON SUPERARE MAI IL VALORE 120 come somma dei tre leds massimo 3-4 led accesi
10
11 #include <Adafruit_NeoPixel.h>
12 #include "font5x8.h"
13
14 #define lunghezza 30
15 #define LED_PIN 2 // collegati alla porta 12
16 #define LED_NUM 240 // 30 leds
17 Adafruit_NeoPixel strip = Adafruit_NeoPixel(LED_NUM, LED_PIN, NEO_GRB + NEO_KHZ800);
18
19 byte ore, minuti, secondi, decimi = 0;
20 unsigned long rosso = strip.Color(40, 00, 0);
21 unsigned long verde = strip.Color(0, 40, 0);
22 unsigned long blu = strip.Color(0, 0, 40);
23 unsigned long giallo = strip.Color(40, 40, 0);
24 unsigned long bianco = strip.Color(25, 30, 30);
25 unsigned long rosa = strip.Color(00, 30, 30);
26
27 void numero1(byte n, int pos, unsigned long colore) {
28     int p = 0;
29     for (int i = 2; i < 6; i++) {
30         strip.setPixelColor(pos + p + lunghezza * 7, bitRead(testo1[n][0], 7 - i) ? colore : 0);
31         strip.setPixelColor(pos + p + lunghezza * 6, bitRead(testo1[n][1], 7 - i) ? colore : 0);
32         strip.setPixelColor(pos + p + lunghezza * 5, bitRead(testo1[n][2], 7 - i) ? colore : 0);
33         strip.setPixelColor(pos + p + lunghezza * 4, bitRead(testo1[n][3], 7 - i) ? colore : 0);
34         strip.setPixelColor(pos + p + lunghezza * 3, bitRead(testo1[n][4], 7 - i) ? colore : 0);
35         strip.setPixelColor(pos + p + lunghezza * 2, bitRead(testo1[n][5], 7 - i) ? colore : 0);
36         strip.setPixelColor(pos + p + lunghezza * 1, bitRead(testo1[n][6], 7 - i) ? colore : 0);
37         strip.setPixelColor(pos + p, bitRead(testo1[n][7], 7 - i) ? colore : 0);
38         p++;
39     }
40 }
41
42 void printnumero(byte n, byte pos, unsigned long colore) {
43     if (n < 100) {
44         numero1((int)n / 10, pos, colore);
45         pos = pos + 5;
46         n = n % 10;
47         numero1(n, pos, colore);
48     }
49 }
50
51 void aggiorna() {
52     secondi++;
53     if (secondi > 59) {
54         secondi = 0;
55         minuti++;
56         if (minuti > 59) {
57             minuti = 0;
58             ore++;
59             if (ore > 23) {
60                 ore = 0;
61             }
62         }
63     }
64 }
65
66 void setup() {
67     strip.begin();
68     strip.show();
69     ore = 15;
70     minuti = 58;
71     secondi = 45;
72 }
73
74 void loop() {
75     aggiorna();
76     printnumero(ore, 0, rosso);
77     printnumero(minuti, 10, bianco);
78     printnumero(secondi, 20, blu);
79     strip.show();
80     delay(999);
81 }
82 //fine programma

```

```
1 int testo1[11][8] = {
2   B00111100, // 0
3   B00100100,
4   B00100100,
5   B00100100,
6   B00100100,
7   B00100100,
8   B00100100,
9   B00111100,
10
11  B00001000, // 1
12  B00011000,
13  B00101000,
14  B00001000,
15  B00001000,
16  B00001000,
17  B00001000,
18  B00011100,
19
20  B00111100, // 2
21  B00000100,
22  B00000100,
23  B00111100,
24  B00100000,
25  B00100000,
26  B00100000,
27  B00111100,
28
29  B00111100, // 3
30  B00000100,
31  B00000100,
32  B00011100,
33  B00000100,
34  B00000100,
35  B00000100,
36  B00111100,
37
38  B00000100, // 4
39  B00001100,
40  B00010100,
41  B00111100,
42  B00000100,
43  B00000100,
44  B00000100,
45  B00000100,
46
```

```
47  B00111100, // 5
48  B00100000,
49  B00100000,
50  B00111100,
51  B00000100,
52  B00000100,
53  B00000100,
54  B00111100,
55
56  B00100000, // 6
57  B00100000,
58  B00100000,
59  B00111100,
60  B00100100,
61  B00100100,
62  B00100100,
63  B00111100,
64
65  B00111100, // 7
66  B00000100,
67  B00000100,
68  B00000100,
69  B00000100,
70  B00000100,
71  B00000100,
72  B00000100,
73
74  B00111100, // 8
75  B00100100,
76  B00100100,
77  B00111100,
78  B00100100,
79  B00100100,
80  B00100100,
81  B00111100,
82
83  B00111100, // 9
84  B00100100,
85  B00100100,
86  B00111100,
87  B00000100,
88  B00000100,
89  B00000100,
90  B00000100,
91
```

```
92  B00000000, // 10 spento9
93  B00000000,
94  B00000000,
95  B00000000,
96  B00000000,
97  B00000000,
98  B00000000,
99  B00000000
100 };
101 //EOF
102
```

```

1  //      file: ws7912b_16x16_natale.ino
2  //      progetto: natale con Arduino
3  //      descrizione: accensione disegni vari sul pannello 256 (16x16)
4  //      shields: ws2812 matrice led 16x16
5  //      data: 1.a - 06 gennaio 2022
6  //      autore: www.aldopi.it
7
8  // se si alimenta la striscia con arduino
9  // NON SUPERARE MAI IL VALORE 120 come somma dei tre leds massimo 3-4 led accesi
10 #include "font_3x5.h"
11
12 #include <Adafruit_NeoPixel.h>
13 #define LED_PIN 2 // collegati alla porta 12
14 #define LED_NUM 256 // 30 leds
15 #define LED_ROW 16
16 Adafruit_NeoPixel strip = Adafruit_NeoPixel(LED_NUM, LED_PIN, NEO_RGB + NEO_KHZ800);
17
18 unsigned long rosso = strip.Color(0, 80, 0);
19 unsigned long verde = strip.Color(80, 0, 0);
20 unsigned long blu = strip.Color(0, 0, 80);
21 unsigned long bianco = strip.Color(60, 60, 60);
22 unsigned long colore = bianco;
23 unsigned long tempo;
24
25 byte minuti, secondi = 0;
26
27 int i;
28 unsigned long color[8];
29 byte posizione = 0;
30
31 void setup() {
32     strip.begin();
33     strip.show();
34     minuti = 50;
35     secondi = 0;
36     disegno_in();
37 }
38
39 void loop() {
40     int tempo = 0;
41     int durata = 10;
42     while (tempo < 30) {
43         disegno();
44         tempo++;
45     }
46     tempo = 0;
47     while (tempo < 30) {
48         casuale1();
49         tempo++;
50     }
51     tempo = 0;
52     strip.fill(0, 0, 256);
53     strip.show();
54     while (tempo < 5) {
55         numeri();
56         tempo++;
57     }
58 }
59
60
61 void numero1(byte n, byte pos_x, byte pos_y, unsigned long colore) {
62     for (byte i = 0; i < FONT_MOD; i++) {
63         strip.setPixelColor(pos_x + i + LED_ROW * 4 + pos_y * LED_ROW, bitRead(font[n][0], 7 - i) ? colore : 0);
64         strip.setPixelColor(pos_x + i + LED_ROW * 3 + pos_y * LED_ROW, bitRead(font[n][1], 7 - i) ? colore : 0);
65         strip.setPixelColor(pos_x + i + LED_ROW * 2 + pos_y * LED_ROW, bitRead(font[n][2], 7 - i) ? colore : 0);
66         strip.setPixelColor(pos_x + i + LED_ROW * 1 + pos_y * LED_ROW, bitRead(font[n][3], 7 - i) ? colore : 0);
67         strip.setPixelColor(pos_x + i + pos_y * LED_ROW, bitRead(font[n][4], 7 - i) ? colore : 0);
68     }
69 }
70

```

```

71 void printnumero2(byte n, byte pos_x, byte pos_y, unsigned long colore) {
72     if (n < 100) {
73         if (n > 9) {
74             numero1(n / 10, pos_x, pos_y, colore);
75         } else {
76             numero1(11, pos_x, pos_y, colore);
77         }
78         pos_x = pos_x + FONT_MOD + 1;
79         n = n % 10;
80         numero1(n, pos_x, pos_y, colore);
81     }
82 }
83
84 void printnumero4(int n, byte pos_x, byte pos_y, unsigned long colore) {
85     byte primo = 0;
86     if (n < 10000) {
87         if (n > 999) {
88             numero1(n / 1000, pos_x, pos_y, colore);
89             n = n - (n / 1000) * 1000;
90             primo = 1;
91         } else {
92             numero1(11, pos_x, pos_y, colore);
93         }
94         pos_x = pos_x + FONT_MOD + 1;
95         if (n > 99) {
96             numero1(n / 100, pos_x, pos_y, colore);
97             n = n - (n / 100) * 100;
98             primo = 1;
99         } else {
100             if (primo) numero1(0, pos_x, pos_y, colore);
101             else numero1(11, pos_x, pos_y, colore);
102         }
103         pos_x = pos_x + FONT_MOD + 1;
104         if (n > 9) {
105             numero1(n / 10, pos_x, pos_y, colore);
106             n = n - (n / 10) * 10;
107             primo = 1;
108         } else {
109             if (primo) numero1(0, pos_x, pos_y, colore);
110             else numero1(11, pos_x, pos_y, colore);
111         }
112         pos_x = pos_x + FONT_MOD + 1;
113         numero1(n, pos_x, pos_y, colore);
114     }
115 }
116
117
118 void numeri() {
119     int lungo = 9876;
120
121     for (byte x = 50; x < 99; x++) {
122         printnumero4(lungo, 0, 11, blu);
123         lungo = random(0, 9999);
124         printnumero2(x, 9, 6, rosso);
125         printnumero2(minuti, 0, 0, verde);
126         printnumero2(secondi, 9, 0, verde);
127         strip.show();
128         delay(200);
129         secondi++;
130         if (secondi > 59) {
131             secondi = 0;
132             minuti++;
133             if (minuti > 59) minuti = 0;
134         }
135     }
136 }
137

```

```

138 void casuale1() {
139     for (int i = 0; i < LED_NUM; i++) {
140         strip.setPixelColor(i, strip.Color(random(0, 100), random(0, 100), random(0, 100)));
141     }
142     strip.show();
143     delay(100);
144 }
145
146 void disegno() {
147     posizione++;
148     if (posizione > 7) posizione = 0;
149     for (byte riga = 0; riga < 8; riga++) {
150         rettangolo(riga, color[(posizione + riga) > 7 ? (posizione + riga - 8) : posizione + riga]);
151     }
152     strip.show();
153     color[posizione] = strip.Color(random(0, 3) * 5, random(0, 3) * 5, random(0, 3) * 5);
154     delay(500);
155 } // fine programma
156
157 void rettangolo(byte n, unsigned long c) {
158     for (byte x = 0 + n; x < 16 - n; x++) set_led(x, 0 + n, c);
159     for (byte x = 0 + n; x < 16 - n; x++) set_led(x, 15 - n, c);
160     for (byte y = 0 + n; y < 16 - n; y++) set_led(0 + n, y, c);
161     for (byte y = 0 + n; y < 16 - n; y++) set_led(15 - n, y, c);
162 }
163
164 void disegno_in() {
165     for (byte n = 0; n < 8; n++) {
166         color[n] = strip.Color(random(0, 10) * 5, random(0, 10) * 5, random(0, 10) * 5);
167     }
168 }
169
170 void set_led(byte x, byte y, unsigned long c) {
171     strip.setPixelColor(x + y * 16, c);
172 }
173
174 void casuale() {
175     for (i = 0; i < LED_NUM; i++) {
176         strip.setPixelColor(i, strip.Color(random(0, 3) * 5, random(0, 3) * 5, random(0, 3) * 5));
177     }
178     strip.show();
179     delay(500);
180 }
181 //fine programma

```

```

1  //      file: LCD16x2_cronometro.ino
2  //  descrizione: realizzazione di un cronometro
3  //      shields: lcd 16x2
4  //      librerie: LiquidCrystal.h (1.0.7)
5  //      revisione: 1.a - 01 Febbraio 2025
6  //      autore: www.aldopi.it
7  //      *****
8
9  #include <LiquidCrystal.h>
10
11 LiquidCrystal lcd(6, 7, 2, 3, 4, 5);
12 // LiquidCrystal(rs, enable, d4, d5, d6, d7)
13
14 int ore = 0;
15 int minuti = 0;
16 int secondi = 0;
17 int centesimi = 0;
18
19 void setup() {
20     lcd.begin(16, 2);
21     lcd.setCursor(0, 0);
22     lcd.print(" hello world! ");
23     lcd.setCursor(0, 1);
24     lcd.print(" Corso Arduino ");
25     delay(3000);
26     lcd.clear();
27     lcd.setCursor(0, 0);
28     lcd.print("  cronometro  ");
29 }
30
31 void loop() {
32     delay(9);
33     centesimi++;
34     if (centesimi == 100) {
35         centesimi = 0;
36         secondi++;
37         if (secondi == 60) {
38             secondi = 0;
39             minuti++;
40             if (minuti == 60) {
41                 minuti = 0;
42                 ore++;
43                 if (ore == 24) { ore = 0; }
44             }
45         }
46     }
47     lcd.setCursor(1, 1);
48     if (ore < 10) lcd.print("0");
49     lcd.print(ore);
50     lcd.print(":");
51     if (minuti < 10) lcd.print("0");
52     lcd.print(minuti);
53     lcd.print(":");
54     if (secondi < 10) lcd.print("0");
55     lcd.print(secondi);
56     lcd.print("  ");
57     if (centesimi < 10) lcd.print("0");
58     lcd.print(centesimi);
59     lcd.print(" ");
60 }
61 //fine programma

```

```
1 //      file: LCD128x64_demo.ino
2 //  descrizione: realizzazione di un cronometro
3 //      shields: lcd 16x2
4 //      librerie: U8g2.h (1.8), SPI.h,
5 //      revisione: 1.a - 01 Febbraio 2025
6 //      autore: www.aldopi.it
7 //      *****
8
9 #include <Arduino.h>
10 #include <U8g2lib.h>
11
12 #ifdef U8X8_HAVE_HW_SPI
13 #include <SPI.h>
14 #endif
15 #ifdef U8X8_HAVE_HW_I2C
16 #include <Wire.h>
17 #endif
18
19 /*
20   U8g2lib Example Overview:
21   Frame Buffer Examples: clearBuffer/sendBuffer. Fast, but may not work with all Arduino boards because o
22   Page Buffer Examples: firstPage/nextPage. Less RAM usage, should work with all Arduino boards.
23   U8x8 Text Only Example: No RAM usage, direct communication with display controller. No graphics, 8x8 Te
24   a5 -> SLC
25   a4 -> SDA
26 */
27 U8G2_SSD1306_128X64_NONAME_F_HW_I2C u8g2(U8G2_R0, /* reset=*/ U8X8_PIN_NONE);
28 int i = 0;
29
30
31 void setup(void) {
32   u8g2.begin();
33 }
34
35 void loop(void) {
36   u8g2.clearBuffer();           // clear the internal memory
37   u8g2.setFont(u8g2_font_ncenB08_tr); // choose a suitable font
38   //u8g2.drawStr(0, i, "Hello World!"); // write something to the internal memory
39   u8g2.setCursor(0, i);
40   u8g2.print(F("Hello World!"));
41   u8g2.sendBuffer();           // transfer internal memory to the display
42   delay(20);
43   if (++i > 60) i = 5;
44 }
45 // fine programma
46
```