

به نام خدا

تشخیص رنگ سخنگو

تهیه کننده : مصطفی ذاکر حسین

استاد پروژه : آقای گرامی

کد استاد : 881G

تشخیص رنگ سخنگو

با قرار دادن جسم مورد نظر بر روی جایگاه مخصوص دستگاه ، رنگ جسم هم بر روی LCD نمایش و هم به صورت صوتی اعلام می شود.
این دستگاه فقط رنگ اجسامی را اعلام می کند که اول از همه به آن شناسایی شده باشد و هم اینکه جسم نور تابیده شده را بازتاب نماید.

شرح کار:

ابتدا جسم رنگی را در جایگاه مخصوص قرار می دهیم و سپس یک ولتاژ بر حسب میزان بازتاب نور دو سر LDR به وجود می آید این ولتاژ به پایه ورودی ADC در میکرو می رسد و میکرو با اندازه گیری ولتاژ مربوطه یک عدد را بر روی LCD نمایش می دهد این عدد برای هر رنگ متفاوت است و با قرار دادن یک بازه برای هر رنگ می توان رنگ مورد نظر را تشخیص داد بعد از تشخیص رنگ میکرو به آیسی ضبط صوت فرمان می دهد تا رنگ را به صورت صوتی اعلام کند.

اجزا دستگاه :

این دستگاه از ۳ قسمت اصلی تشکیل شده

۱-کنترل

۲-قسمت صوتی

۳-تشخیص رنگ

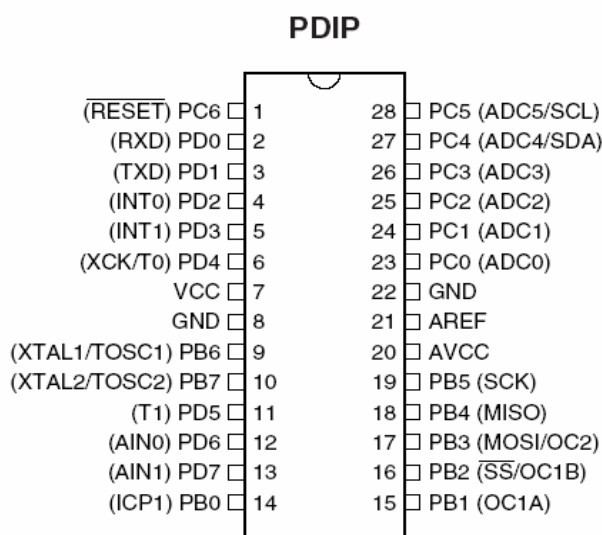
در کنار دستگاه ۲ عدد ولوم قرار داده شده که یکی برای تغییر میزان صدا و یکی برای حذف خطا در دستگاه به کار رفته است در محیط های مختلف به دلیل متغیر بودن نور ممکن است بر روی عملکرد دستگاه تاثیر بگذارد برای همین از این ولوم استفاده شده است تا دستگاه تنظیم شود.

کلیه فایل های دیتاشیت و همچنین شماتیک مدار و PCB به صورت PDF در CD قرار گرفته اند.

۱- کنترل

این قسمت که وظیفه کنترل کل قسمت های دستگاه را بر عهده دارد. برای کنترل دستگاه از یک آیسی **ATMEGA8** استفاده شده است که مشخصات این آیسی در زیر آمده است.

خصوصیات *Atmega8L* , *ATmega8*



از معماری **AVR RISC** استفاده میکند

— کارایی بالا و توان مصرفی کم

— دارای 130 دستورالعمل با کارایی بالا که اکثراً تنها در یک کلاک سیکل اجرا میشوند

— 32×8 رجیستر کاربردی

— سرعتی تا 16MIPS در فرکانس 16MHZ

.حافظه ، برنامه و داده غیر فرآر

— 8Kبایت حافظه FLASH داخلی قابل برنامه ریزی

پایداری حافظه FLASH : قابلیت 10000 بار نوشتن و پاک کردن

(WRITE/ERASE)

— 1024بایت حافظه داخلی SRAM

— 512 بایت حافظه EEPROM داخلی قابل برنامه ریزی

پایداری حافظه EEPROM : قابلیت 100,000 بار پاک کردن و نوشتن

— قفل برنامه FLASH و حفاظت داده EEPROM

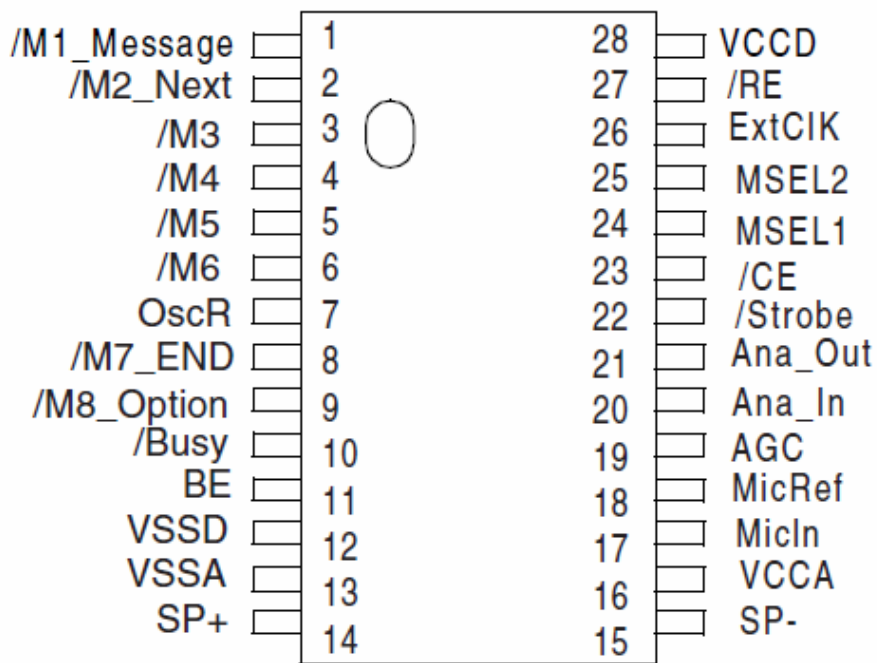
خصوصیات جانبی

— دو تایمر - کانتر 8 بیتی با PRESCALER مجزا و دارای مد COMPARE

- یک تایمر – کانتر 16 بیتی با PRESCALER مجزا و دارای مدهای COMPARE و CAPTURE
- 3 کانال PWM
- 8 کانال مبدل آنالوگ به دیجیتال در بسته های TQFP و MLF
- 6- کانال با دقت 10 بیتی
- 2- کانال با دقت 8 بیتی
- 6 کانال مبدل آنالوگ به دیجیتال در بسته بندی PDIP
- 4- کانال با دقت 10 بیتی
- 2 کانال با دقت 8 بیتی
- دارای RTC (REAL – TIME CLOCK) با اسیلاتور مجزا
- یک مقایسه کننده آنالوگ داخلی
- USART سریال قابل برنامه ریزی
- WATCHDOG قابل برنامه ریزی با اسیلاتور داخلی
- ارتباط سریال SPI برای برنامه ریزی داخل مدار (IN – SYSTEM PROGRAMMING)
- قابلیت ارتباط سریال SPI به صورت MASTER یا SLAVE
- قابلیت ارتباط با پروتکل سریال دو سیمه (TWO – WIRE) خصوصیات ویژه میکرو کنترلر
- POWER – ON RESET CIRCUIT
- دارای 5 حالت (SLEEP, POWER – DOWN, POWER – SAVE, STANDBY و IDLE, ADC NOISE REDUCTION)
- منابع وقفه (INTERRUPT) داخلی و خارجی
- دارای اسیلاتور RC داخلی کالیبره شده
- عملکرد کاملاً ثابت
- توان مصرفی پایین و سرعت بالا توسط تکنولوژی CMOS
- توان مصرفی در 25C,3V,4MHZ
- حالت فعال (ACTIVE MODE) 3.6mA
- در حالت بی کاری (IDLE MODE) 1.0mA
- در حالت POWER – DOWN : $5 \mu A >$
- فایل دیتاشیت در CD قرار گرفته است

۲- قسمت صوتی

برای قسمت صوتی از یک آیسی ضبط صدا با شماره APR9600 استفاده شده که مشخصات آن در زیر بیان شده است.



— یک آیسی ۲۸ پایه

— قابلیت ضبط حداکثر ۱ دقیقه

— دارای ۸ قسمت مجزا برای ضبط صدا

— امکان ضبط و پخش آسان

— قابل اتصال به خروجی صدای کامپیوتر برای ضبط از کامپیوتر

— قابل اتصال به خط تلفن

و دیگر خصوصیات که در دیتاشیت آمده است .

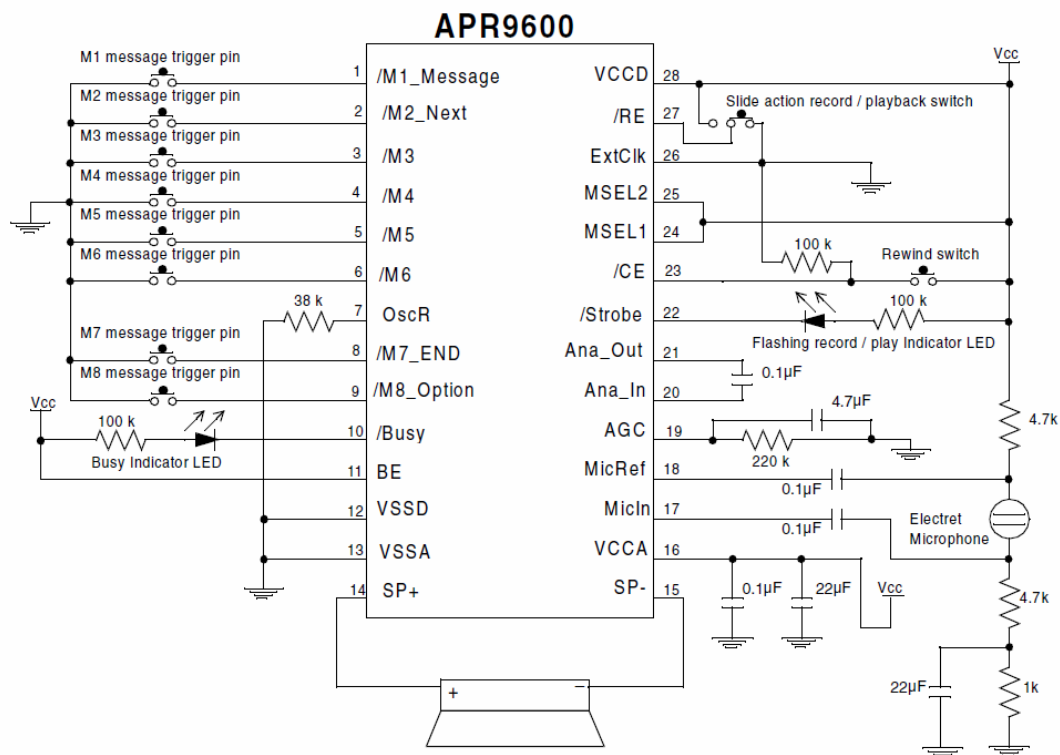
فایل دیتاشیت در CD قرار گرفته است.

پایه ۷ که OSCR نام دارد به وسیله یک مقاومت به زمین متصل می شود که این مقاومت تعیین کننده میزان ضبط می باشد. با توجه به جدول زیر انتخاب می شود.

Ref Rosc	Sampling Frequency	Input Bandwidth	Duration
84 K	4.2 kHz	2.1 kHz	60 sec
38 K	6.4 kHz	3.2 kHz	40 sec
24 K	8.0 kHz	4.0 kHz	32 sec

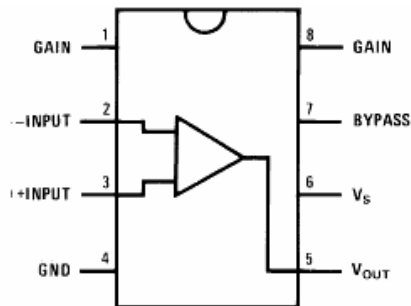
مقاومت استفاده شده در این مدار از 39K است.

در دیتاشیت این آیسی نقشه های مختلفی از مدار کشیده شده در این مدار از این نقشه استفاده شده.

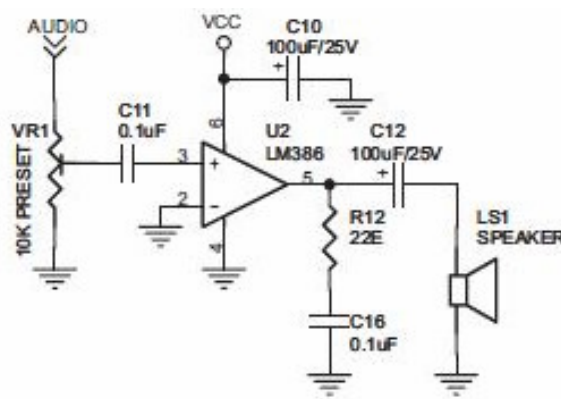


پایه های مربوط به هر پیام ۱ و ۲ و ۳ و ۴ و ۵ و ۶ و ۸ و ۹ میباشد. پایه ۹ را باید با یک مقاومت 100K به VCC متصل کرد.

برای اتصال به بلندگو برای اینکه صدا قویتر شود از یک تقویت کننده عملیاتی به شماره LM386 استفاده شده است.



مدار این آرسی به صورت زیر است.



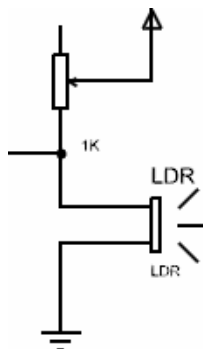
فایل دیتاشیت در CD قرار گرفته است.

۳- تشخیص رنگ

در این دستگاه از سنسور رنگ استفاده نشده برای همین قیمت دستگاه کاهش فراوانی کرده است.

برای تشخیص رنگ از یک LED سفید و یک عدد LDR (مقاومت تابع نور) استفاده شده .

LDR را با یک پتانسیومتر سری و به VCC متصل شده تا یک تقسیم مقاومتی به وجود بیاید و مقدار ولتاژ بدست آمده را به ADC میکرو می دهیم تا میکرو رنگ را تشخیص دهد.



برنامه میکرو:

برای اینکه برنامه به زبان C و در محیط CODEVISION نوشته شده اگر خود کدها کپی می شد در ورد به هم می ریخت برای همین از نوشته ها عکس گرفته شده است. فایل سورس برنامه در CD است.

```
1  #include <mega8.h>
2
3  #asm
4      .equ __lcd_port=0x12 ;PORTD
5  #endasm
6  #include <lcd.h>
7  #include <stdio.h>
8  #include <delay.h>
9
10 #define ADC_VREF_TYPE 0x40
11
12 unsigned int read_adc(unsigned char adc_input)
13 {
14     ADMUX=adc_input | (ADC_VREF_TYPE & 0xff);
15
16     delay_us(10);
17
18     ADCSRA|=0x40;
19
20     while ((ADCSRA & 0x10)==0);
21     ADCSRA|=0x10;
22     return ADCW;
23 }
24
25 char buffer[20];
26 int i,j;
27 char z,a,s,su,g,m,se,gh;
28
29 void main(void)
30 {
31
32     PORTB=0x00;
33     DDRB=0x3A;
34
35     PORTC=0x00;
36     DDRC=0x5F;
37
38     PORTD=0x00;
39     DDRD=0x00;
40
41     PORTB.1=1;
42     delay_ms(300);
43     PORTB.1=0;
44
45     ADMUX=ADC_VREF_TYPE & 0xff;
46     ADCSRA=0x83;
47     PORTC.6=0;
48     lcd_init(16);
49     lcd_clear();
50     PORTC.0=1;
51     PORTC.1=1;
52     PORTC.2=1;
53     PORTC.3=1;
54     PORTC.4=1;
```

```

55 PORTB.3=1;
56 PORTB.4=1;
57 PORTB.5=1;
58 lcd_putsf("welcome");
59 lcd_gotoxy(0,1);
60 lcd_putsf("code:881G");
61 delay_ms(1500);
62 lcd_clear();
63 lcd_gotoxy(0,0);
64 lcd_putsf("m");
65 delay_ms(250);
66 lcd_putsf("o");
67 delay_ms(250);
68 lcd_putsf("s");
69 delay_ms(250);
70 lcd_putsf("t");
71 delay_ms(250);
72 lcd_putsf("a");
73 delay_ms(250);
74 lcd_putsf("f");
75 delay_ms(250);
76 lcd_putsf("a");
77 delay_ms(250);
78 lcd_gotoxy(0,1);
79 lcd_putsf("z");
80 delay_ms(250);
81 lcd_putsf("a");
82 delay_ms(250);
83 lcd_putsf("k");
84 delay_ms(250);
85 lcd_putsf("e");
86 delay_ms(250);
87 lcd_putsf("r");
88 delay_ms(250);
89 lcd_putsf("h");
90 delay_ms(250);
91 lcd_putsf("o");
92 delay_ms(250);
93 lcd_putsf("s");
94 delay_ms(250);
95 lcd_putsf("s");
96 delay_ms(250);
97 lcd_putsf("e");
98 delay_ms(250);
99 lcd_putsf("i");
100 delay_ms(250);
101 lcd_putsf("n");
102 delay_ms(1500);
103 lcd_clear();

```

```

104
105 while (1)
106 {
107     start:
108     {
109
110         delay_ms(25);
111         i= read_adc(5);
112
113         lcd_gotoxy(0,0);
114
115         sprintf(buffer,"%d",i);
116         lcd_puts(buffer);
117
118         if (i>=590&&i<=605)
119         {
120             z=0,a=0,s=0,su=0,g=0,m=0,gh=0;
121             se++;
122             if (se<20)
123                 {goto start;}
124             z=0,a=0,s=0,su=0,g=0,m=0,se=0,gh=0;
125             lcd_clear();
126             lcd_gotoxy(0,1);
127             lcd_putsf("sefid");
128
129             PORTC.0=0;
130             delay_ms(200);
131             PORTC.0=1;
132             delay_ms(1500);
133             goto start;
134         }
135
136         if (i>=607&&i<=618)
137         {
138             a=0,s=0,su=0,g=0,m=0,se=0,gh=0;
139             z++;
140             if (z<20)
141                 {goto start;}
142             z=0,a=0,s=0,su=0,g=0,m=0,se=0,gh=0;
143             lcd_clear();
144             lcd_gotoxy(0,1);
145             lcd_putsf("zard");
146
147             PORTC.3=0;
148             delay_ms(200);
149             PORTC.3=1;
150             delay_ms(1500);
151             goto start;
152         }
153
154         if (i>=620&&i<=635)
155         {
156             z=0,a=0,s=0,g=0,m=0,se=0,gh=0;

```

```

157 su++;
158 if (su<20)
159 (goto start;)
160 z=0,a=0,s=0,su=0,g=0,m=0,se=0,gh=0;
161 lcd_clear();
162 lcd_gotoxy(0,1);
163 lcd_putsf("naranji");
164
165 PORTB.4=0;
166 delay_ms(200);
167 PORTB.4=1;
168 delay_ms(1500);
169 goto start;
170 }
171
172
173
174 if (i>=645&&i<=685)
175 {
176 z=0,a=0,s=0,su=0,g=0,m=0,se=0;
177 gh++;
178 if (gh<20)
179 (goto start;)
180 z=0,a=0,s=0,su=0,g=0,m=0,se=0,gh=0;
181 lcd_clear();
182 lcd_gotoxy(0,1);
183 lcd_putsf("ghermez");
184
185 PORTC.1=0;
186 delay_ms(200);
187 PORTC.1=1;
188 delay_ms(1500);
189 goto start;
190 }
191
192 if (i>=700&&i<=727)
193 {
194 z=0,a=0,su=0,g=0,m=0,se=0,gh=0;
195 s++;
196 if (s<20)
197 (goto start;)
198 z=0,a=0,s=0,su=0,g=0,m=0,se=0,gh=0;
199 lcd_clear();
200 lcd_gotoxy(0,1);
201 lcd_putsf("sabz");
202
203 PORTC.4=0;
204 delay_ms(200);
205 PORTC.4=1;
206 delay_ms(1500);
207 goto start;

```

```

208
209 }
210
211 if (i>=735&&i<=768)
212 {
213     a++;
214     z=0,s=0,su=0,g=0,m=0,se=0,gh=0;
215     if (a<20)
216         (goto start;);
217     z=0,a=0,s=0,su=0,g=0,m=0,se=0,gh=0;
218     lcd_clear();
219     lcd_gotoxy(0,1);
220     lcd_putsf("abi");
221
222     PORTC.2=0;
223     delay_ms(200);
224     PORTC.2=1;
225     delay_ms(1500);
226     goto start;
227 }
228
229
230 if (i>=775&&i<=800)
231 {
232     z=0,a=0,s=0,su=0,g=0,se=0,gh=0;
233     m++;
234     if (m<20)
235         (goto start;);
236     z=0,a=0,s=0,su=0,g=0,m=0,se=0,gh=0;
237     lcd_clear();
238     lcd_gotoxy(0,1);
239     lcd_putsf("meshki");
240
241     PORTB.5=0;
242     delay_ms(200);
243     PORTB.5=1;
244     delay_ms(1500);
245     goto start;
246 }
247
248 z=0,a=0,s=0,su=0,g=0,m=0,se=0,gh=0;
249 goto start;
250 }
251 };
252 }

```

متغیر های Z,A,S,SU,G,M,SE,GH برای دقیقتر شدن تشخیص به کار رفته اند. پرا تا نور ثابت شود مدتی طول می کشد و این متغیر ها برای هر رنگ ۱ عدد به آن ها اضافه می شود و اگر این عدد به ۲۰ رسید رنگ مورد نظر تشخیص داده می شود .

شماتیک و مدار PCB:

