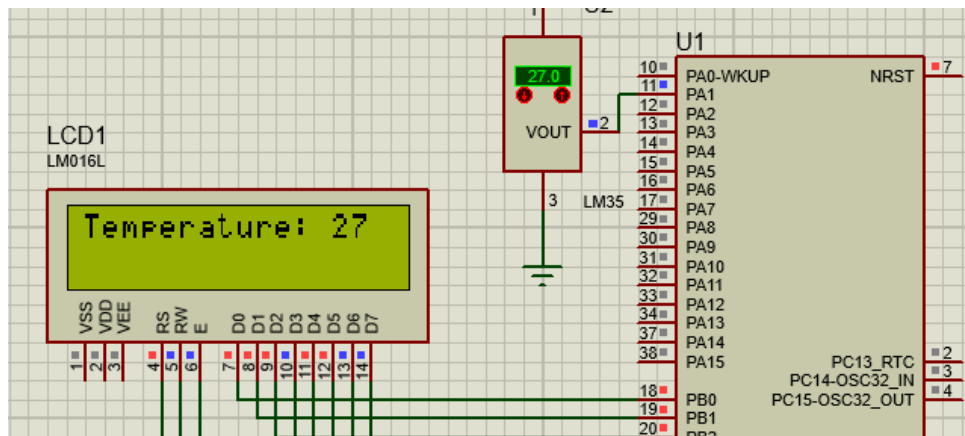


برای بدست آوردن نسبت ورودی ADC به دمای واقعی، سه مقدار مختلف دما را نسبت گرفتیم. در LCD بصورت reversed نمایش دادیم (مثلا در عکس روبرو دمای ۲۶ درجه مقدار ورودی ADC برابر ۲۱۴ بوده)

```
>>> 28/231
0.121212121212122
>>> 26 / 214
0.12149532710280374
>>> 27 / 223
0.1210762331838565
```

نسبت هر سه دمای که اندازه گرفتیم حدود ۰.۱۲۱ بوده است.

پس از این عدد برای تبدیل ADC به دما استفاده کردم.



در نهایت خروجی تبدیل شده دما را روی LCD نمایش دادیم.

```
if (temp / 10 > 2) HAL_GPIO_WritePin(GPIOB, GPIO_PIN_14, GPIO_PIN_SET);
else HAL_GPIO_WritePin(GPIOB, GPIO_PIN_14, GPIO_PIN_RESET);

if (temp / 10 < 2) HAL_GPIO_WritePin(GPIOB, GPIO_PIN_15, GPIO_PIN_SET);
else HAL_GPIO_WritePin(GPIOB, GPIO_PIN_15, GPIO_PIN_RESET);

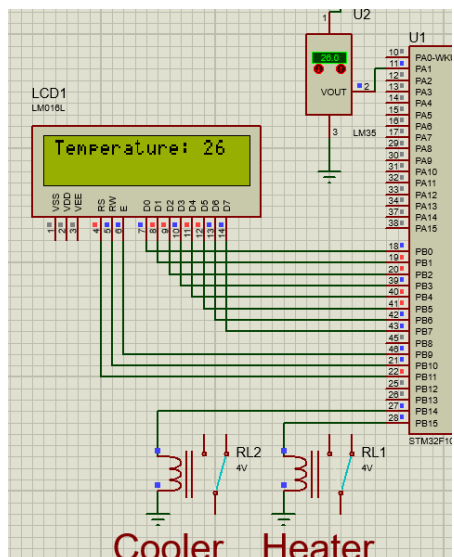
HAL_Delay(300);
```

برای کنترل دمای محیط و روشن و خاموش

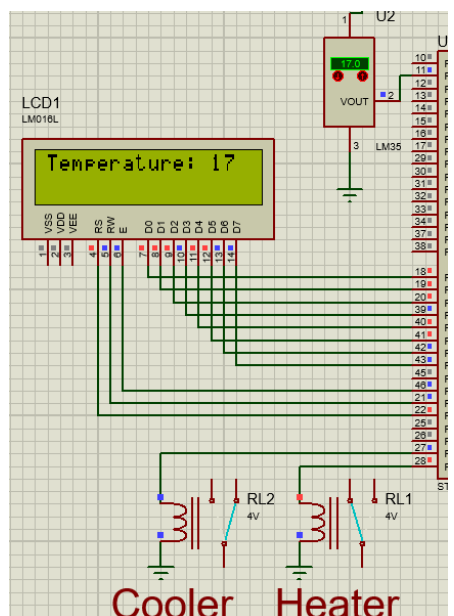
کردن Cooler و Heater از دو رله

استفاده کردم و به پین های B15 و B14

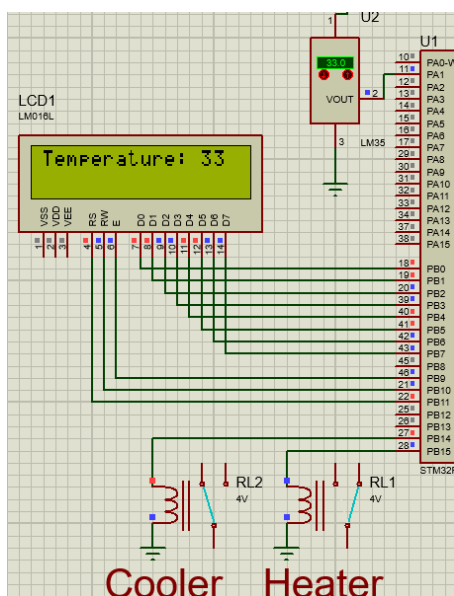
وصل کردم. در صورتی که دما از ۲۹ درجه بالاتر برود، رله ی Cooler روشن می شود و اگر دما از ۲۰ درجه کمتر شود، Heater روشن می شود. در محدوده ی دما ۲۰ تا ۲۹ درجه هم هر دو رله غیرفعال هستند.



### حالت ۱: محدوده دمای ۲۰ تا ۲۹ درجه (Cooler و Heater غیرفعال)

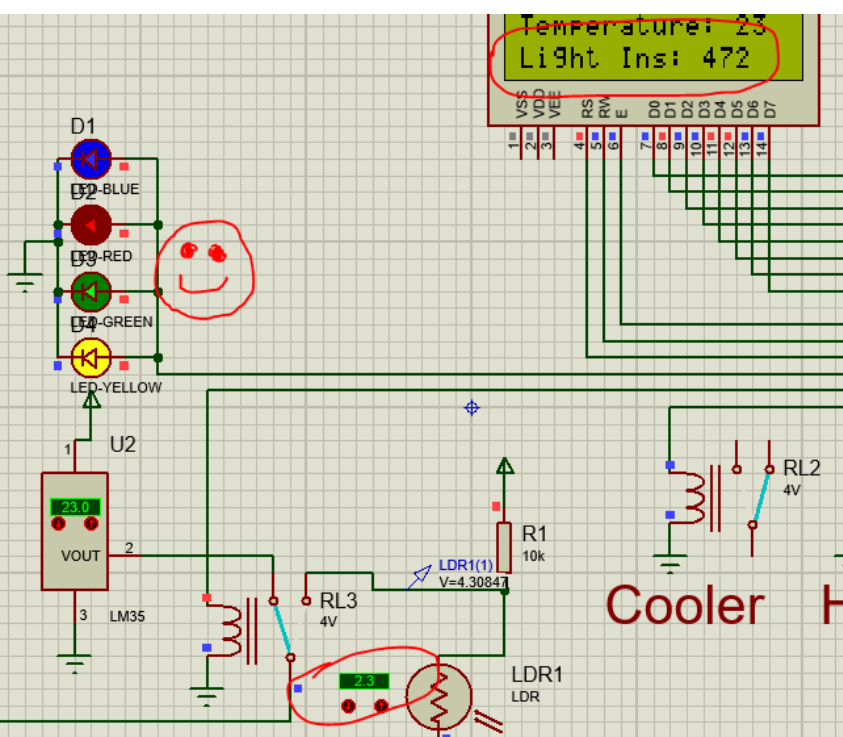
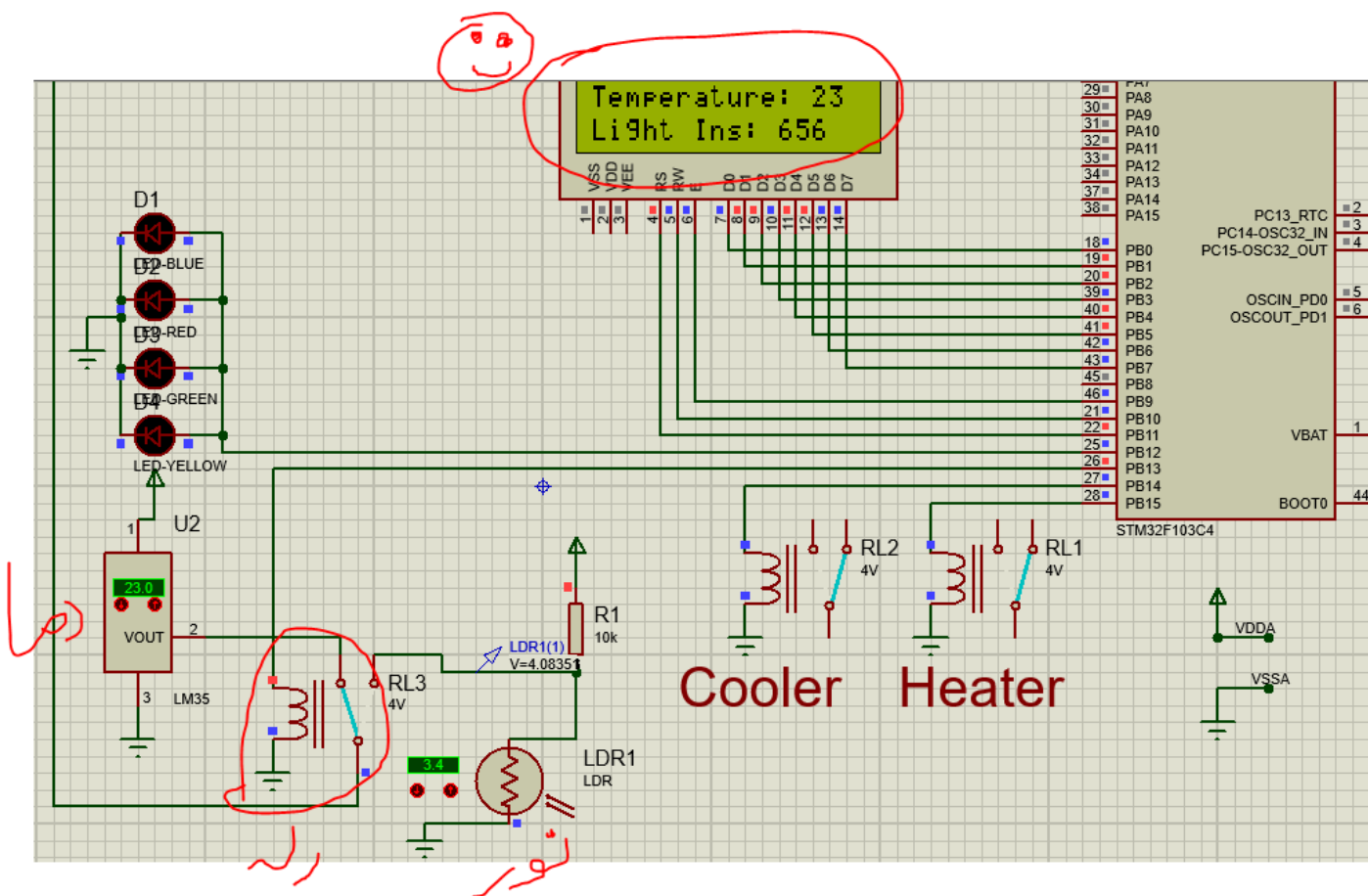


### حالت ۲: محدوده دمای کمتر از ۲۰ درجه (Heater فعال)



### حالت ۳: محدوده دمای بیشتر از ۲۹ درجه (Cooler فعال)

برای اضافه کردن سنسور نور از یک رله استفاده کردم تا هر ۵۰۰ میلی ثانیه ورودی ADC1 را بین سنسور نور و دما جابجا کنم و بتوانم هر دو را از طریق یک ADC بخوانم. ابتدا می خواستم ADC1 و ADC2 را همزمان استفاده کنم تا هر دو مقدار را همزمان بخوانم ولی ظاهراً بدلیل BUS اشتراکی ADC1 و ADC2 دچار conflict می شوند و مقادیر هر دو در استفاده همزمان خراب می شود.



شدت نور گزارش شده عددی سه رقمی است که معیار استفاده شده ورودی ADC - ۱۰۰۰ است چون هرچه شدت نور بیشتر شود ولتاژ دو سر LDR کم می شود و در نتیجه ولتاژی که در ورودی ADC قرار می گیرد هم کمتر می شود پس نیاز داریم به طریقی این مقدار را قرینه کنیم تا معیار مناسبی برای شدت نور بدست بیاید که در اینجا آن را از ۱۰۰۰ کم کردیم تا یک عدد سه رقمی مناسب بدست بیاید.

اگر شدت نور اندازه گیری شده از ۵۰۰ بیشتر باشد یعنی روشنایی مناسب است، در غیر اینصورت LEDهای قرار داده شده (لامپها) روشن خواهند شد.

```
int main(void)
```

```
{
```

```
/* USER CODE BEGIN 1 */
```

```
unsigned int value_adc;
```

```
unsigned int temp;
```

```
unsigned char thos;
```

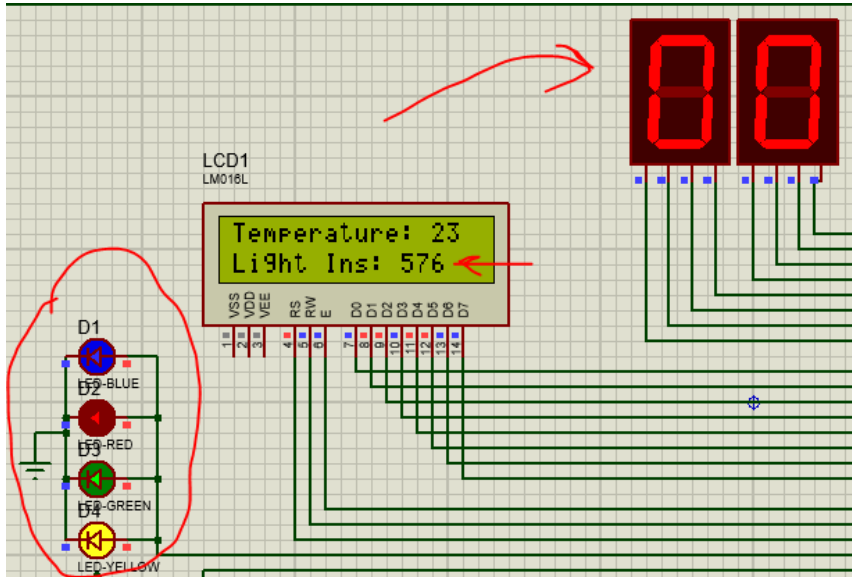
```
unsigned char hunns;
```

```
unsigned char tens;
```

```
unsigned char ones;
```

```
unsigned int timer = 60;
```

در ابتدای برنامه مقدار تایمر شمارش معکوس بصورت یک متغیر ثابت مقداردهی شده است و قابل تنظیم است. پس از آن در هر ثانیه یکی از این متغیر کم



در نهایت وقتی متغیر تایمر صفر شود لامپها روشن می شوند حتی اگر شدت روشنایی محیط بالا باشد.

[Majid-Derhambakhsh/Character-Lcd: CH LCD library for use in AVR - ARM Cortex M \(STM32\) \(github.com\)](#)

[STM32 Simultaneous ADC only reads the values from ADC 1 - Stack Overflow](#)