

بسم الله الرحمن الرحيم

عنوان پروژه

سیستم کنترل حرارت موتور

استاد درس

دکتر فاطمی

اعضای گروه

روند پروژه در یک نگاه

۱. دریافت دما از طریق سنسور NTC تحت پرفرال ADC
۲. اتصال به مازول وای فای esp8266 تحت پروتکل UART1
۳. ارسال دما از طریق وای فای به سرور thinkspeak
۴. آنالیز روی دماهای ارسال شده از سمت میکرو توسط افزونه متلب سرور
۵. ارسال پاسخ متناسب به عنوان وضعیت روشن/خاموش موتور به سمت میکرو
۶. دریافت پاسخ سرور توسط میکرو
۷. نشان دادن وضعیت موتور روی hexseg
۸. جریان اطلاعات و دستورات در تمامی مراحل توسط دیباگر متصل به یوآرت ۲ قابل مشاهده باشد.

بخش اول:

با پرفرال ها آشنا بشویم...

پریفرال ADC

برای استفاده از ADC از کتابخانه های `lpc17xx_adc.h` , `lpc17xx_pinsel.h` که خود شرکت NXP آن را ارائه نموده است، استفاده شد.

راه اندازی اولیه پریفرال ADC

```
PINSEL_CFG_Type adcpinsel;  
adcpinsel.Funcnum=1;  
adcpinsel.OpenDrain=PINSEL_PINMODE_NORMAL;  
adcpinsel.Pinmode=PINSEL_PINMODE_PULLUP;  
adcpinsel.Pinnum=25;  
adcpinsel.Portnum=0;  
  
PINSEL_ConfigPin(&adcpinsel);  
  
ADC_Init(LPC_ADC,100000);  
ADC_ChannelCmd(LPC_ADC,ADC_CHANNEL_2,ENABLE);
```

دریافت دما از طریق سنسور NTC تحت پرفرال ADC

بعد از اینکه initialization صورت گرفت برای خواندن مقدار ADC به صورت زیر عمل می کنیم :

```
ADC_StartCmd(LPC_ADC,ADC_START_NOW) ;  
while(ADC_ChannelGetStatus(LPC_ADC,ADC_CHANNEL_2,ADC_DATA_DONE)==0) ;  
analog=ADC_ChannelGetData(LPC_ADC,ADC_CHANNEL_2) ;
```

به دست آوردن دما

```
analog=ADC_ChannelGetData(LPC_ADC,ADC_CHANNEL_2) ;  
int temp=(analog*4)-7800;
```

پرفرال UART

در این پروژه قسمت UART به صورت رجیستری و بدون کتابخانه کمکی زده شده است ، دلیل استفاده نکردن از کتابخانه به دلیل نکات و جزئیاتی بود که باید به صورت دقیقی در سطح رجیسترها تعیین می گردید.

: UART1

به عنوان واسط بین برد و ماژول وای فای esp8266 استفاده شده است. دستورات ، داده ها و پاسخ ها از این پروتکل بین وای فای و میکرو رد و بدل می شود.

: UART2

ابزار دیباگ و مشاهده نتایج دستورات در این پروژه هست. تمامی دستورات و داده ها و پاسخ هایی که بین برد ، وای فای و سرور رد و بدل می شود توسط uart2 به کامپیوتر ارسال شده و در برنامه hecules مشاهده می شود.

UART راه اندازی

```
33 //init UART1
34 LPC_SC->PCLKSEL0|=0X0; //SET CLOCK OF UART0 CPUCLK/4=24MHZ
35 LPC_UART1->LCR=0X83; //SET 8bit data & enable dlab
36 LPC_UART1->DLL=13; // SET BAUD RATE = 115200
37 LPC_UART1->DLM=0;
38 LPC_UART1->LCR=0X3; // DESABLE DLAB
39 LPC_UART1->FCR=0X7; // SET FIFO AND CLAER
40 LPC_PINCON->PINSEL0 |= (1 << 30); // Pin P0.10 used as TXD2 (Com2)
41 LPC_PINCON->PINSEL1 |= (1 << 0); // SET PIN FOR UART0
42 LPC_UART1->IER=0X01;
43
44 // init UART2
45 LPC_SC->PCLKSEL0|=0X0; //SET CLOCK OF UART0 CPUCLK/4=24MHZ
46 LPC_PINCON->PINSEL0 |= (1 << 20); // Pin P0.10 used as TXD2 (Com2)
47 LPC_PINCON->PINSEL0 |= (1 << 22); // Pin P0.11 used as RXD2 (Com2)
48 LPC_SC->PCONP = LPC_SC->PCONP|(1<<24); //Open UART2 power control bit
49 LPC_UART2->LCR=0x83; //SET 8bit data & enable dlab
50 LPC_UART2->DLL=13; // SET BAUD RATE = 115200
51 LPC_UART2->DLM=0;
52 LPC_UART2->LCR=0X3; // DESABLE DLAB
53 LPC_UART2->FCR=0X7; // SET FIFO AND CLAER
54 LPC_UART2->IER=0X01;
```


پریفرال GPIO

برای راه اندازی HEXSEG از ۸ پین پورت صفر و ۸ پین پورت ۲ در حالت GPIO استفاده شده است. راه اندازی این پریفرال با استفاده از همان کتابخانه lpc1768_gpio.h انجام شده است.



```
253 void hexseg(int a)
254 {
255     GPIO_SetDir(0, 0x000000FF, 1);
256     GPIO_SetDir(2, 0x000000FF, 1);
257     GPIO_ClearValue(0, 0x000000FF);
258     GPIO_ClearValue(2, 0x000000FF);
259     if(a==1)
260     {
261         GPIO_SetValue(0, 0x000000C7); //R
262         GPIO_SetValue(2, 0x00000098);
263     }
264     else if(a==0)
265     {
266         GPIO_SetValue(0, 0x000000BB); //S
267         GPIO_SetValue(2, 0x00000088);
268     }
269     else
270     {
271         GPIO_ClearValue(0, 0x000000FF);
272         GPIO_ClearValue(2, 0x000000FF);
273     }
```

بخش دوم:

با Esp8266 به چه زبانی صحبت کنیم؟

Received/Sent data

Serial port COM3 opened

AT

OK

AT+CWMODE=1

OK

AT+CWLAP="MPLab", "MpProject1400"

WIFI DISCONNECT

WIFI CONNECTED

WIFI GOT IP

OK

AT+CIPMUX=0

OK

AT+CIPSTART="TCP", "api.thingspeak.com", 80
CONNECT

OK

AT+CIPSEND=48

OK

> AT+CIPCLOSE

busy s...

Recv 48 bytes

SEND OK

+IPD,639:HTTP/1.1 200 OK

Modem lines



CD



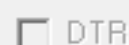
RI



DSR



CTS



DTR



RTS

Serial

Name

COM3

Baud

115200

Data size

8

Parity

none

Handshake

OFF

Mode

Free



Open

HWg FW update

توابع مورد نیاز

```
12  int string_analyze(void);
13  void hexseg(int);
14  void stringer(void);
15  void stringer2(void);
16  int delay (int); // Delay function
17  unsigned char getchar1 (bool); // Get Char from UART1
18  unsigned char getchar2 (void); // Get Char from UART2
19  void sendchar1 (unsigned char ); // Send Char from UART1
20  void sendchar2 (unsigned char ); // Send Char from UART2
21  void sendstring1 (char *); // Send string from UART1
22  void sendstring2 (char *); // Send string from UART2
```

بخش سوم:

Things speak !

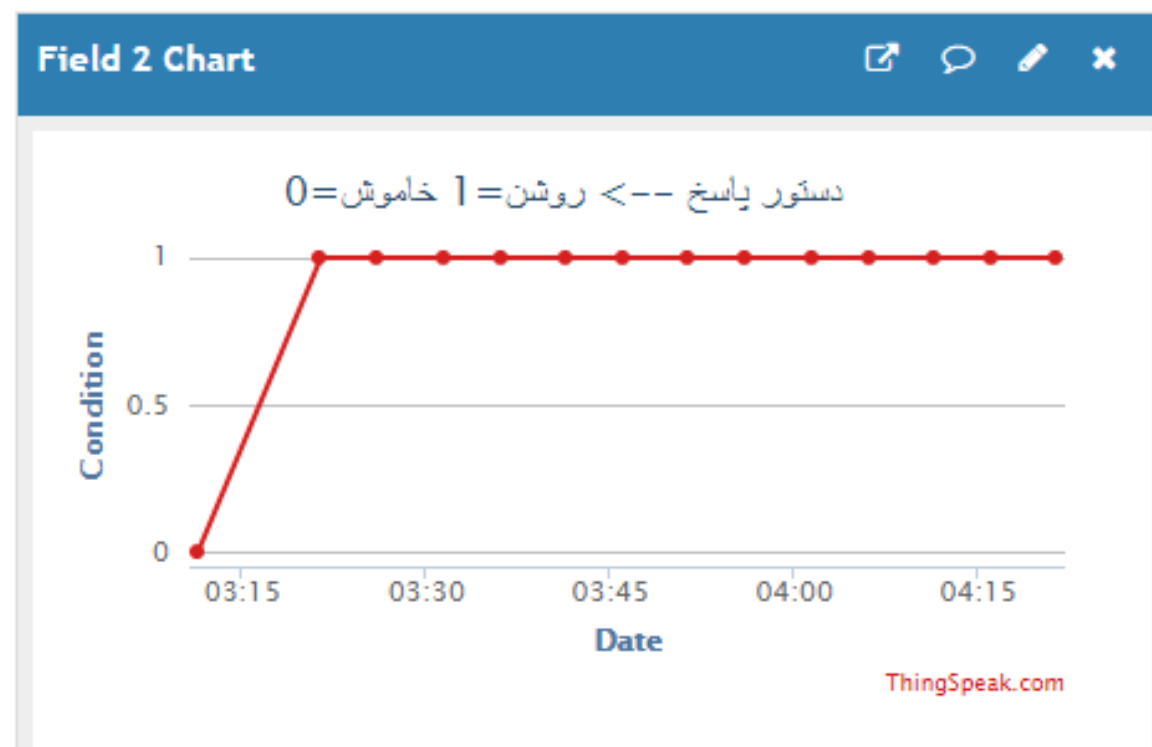
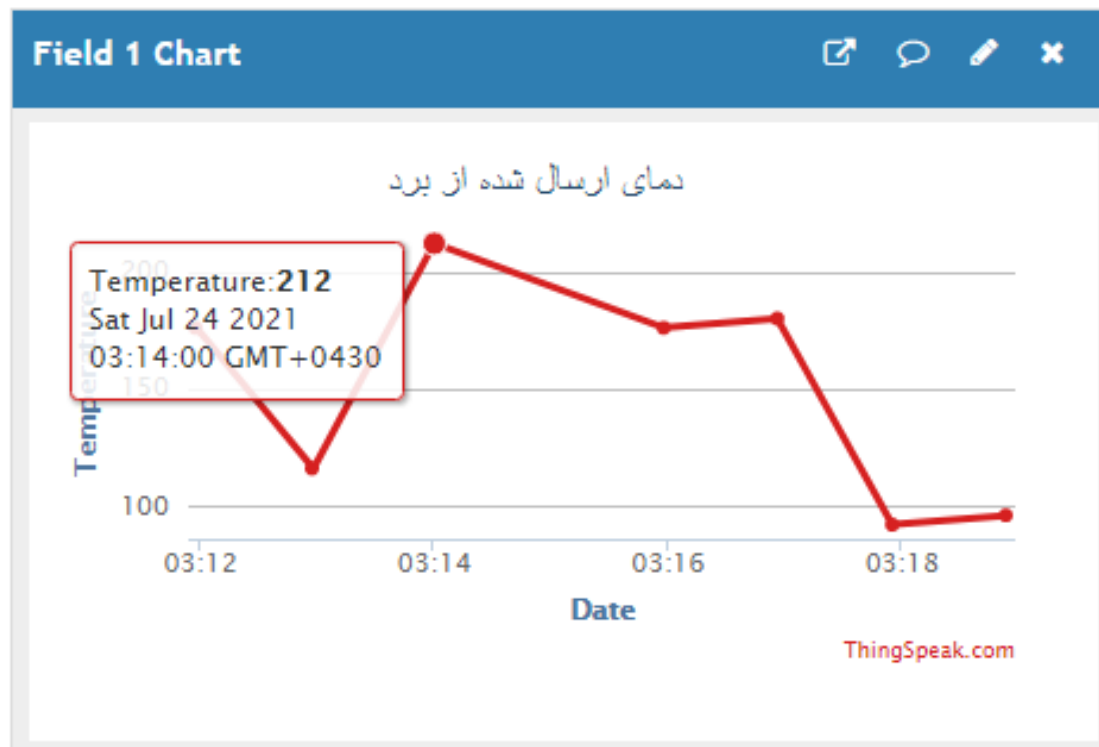
سرور thingspeak

در لبه سرور ما از سایت **thingspeak** استفاده کرده ایم. ما در این سایت کانال ایجاد کردیم:

<https://thingspeak.com/channels/1405439>

Channel ID	1405439	
Name	<input type="text" value="UT_MP_G5"/>	
Description	<input type="text" value="Analyze motor temp. and actuating via ESP8266"/>	
Field 1	<input type="text" value="Temperature"/>	<input checked="" type="checkbox"/>
Field 2	<input type="text" value="Condition"/>	<input checked="" type="checkbox"/>

فیلدها



پس در مجموع از دید میکرو ، فیلد ۱ نوشتنی و فیلد ۲ خواندنی هست

فیلد ۲ چگونه داده درست می کند؟

MATLAB Code

```
1  
2 channel_id = 1405439;  
3 writeAPIKey = 'K801QQ0UAIUVHN0W';  
4  
5 temp = thingSpeakRead(channel_id,'Fields',1,'NumMinutes',3);  
6  
7 avg_temp = mean(temp);  
8 display(avg_temp,'Average temp');  
9  
10 if(avg_temp > 100 ) cond=0; else cond=1; end  
11  
12 thingSpeakWrite(channel_id,cond,'Fields',2,'WriteKey',writeAPIKey);
```

Name:	Time control
Frequency:	Every 5 minutes
Time Zone:	Tehran (edit)
Last Ran:	2021-07-24 4:26 am
Run At:	2021-07-24 4:31 am
Fuzzy Time:	± 0 minutes
MATLAB Analysis:	check condition

بخش آخر:

کار شروع می شود...

