

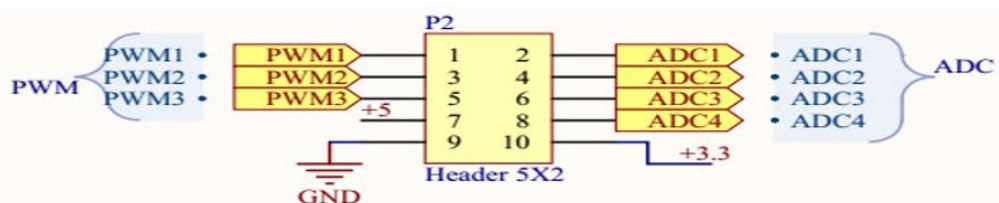
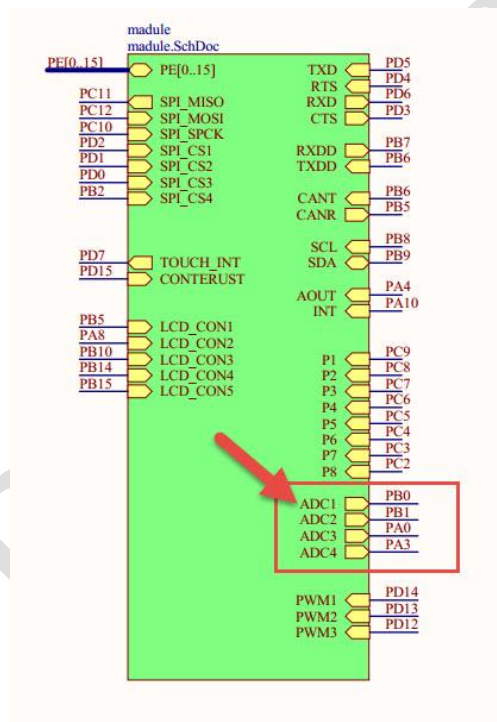
راهنمای استفاده از مثال DMA

هدف از مثال زیر این است که شما با (DMA) و طریقه‌ی راه‌اندازی آن در میکروکنترلرهای ARM آشنا شوید.

DMA از واحدهای RAM میکروکنترلر است. با استفاده از DMA و دانستن آدرس پریفرالهای مورد نظر، می‌توان به راحتی مقادیر را بدون اینکه CPU میکرو درگیر باشد، خواند و از آنها استفاده نمود (برای اطلاعات بیشتر به دیتاشیت مراجعه کنید)

1. سخت افزار

برای تست برنامه می‌توانید از یک پتانسیومتر استفاده نمائید و آن را به ADC1 متصل کنید.



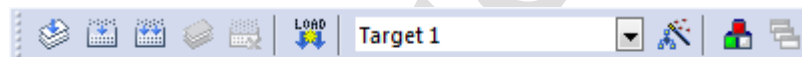
ANALOG INPUT OUTPUT

2. آشنایی و کار با برنامه

ابتدا برنامه را با نرم افزار Keil باز بکنید.

Help	۲۰۱۵/۰۶/۱۲ ۱۲:۵۹ ...	File folder	
obj	۲۰۱۵/۰۶/۱۲ ۱۲:۵۶ ...	File folder	
bsp	۲۰۱۵/۰۶/۱۲ ۱۲:۴۹ ...	H File	1 KB
delay	۲۰۱۵/۰۵/۱۲ ۱۰:۰۴ ...	C File	1 KB
delay	۲۰۱۵/۰۵/۱۲ ۰۲:۲۷ ...	H File	1 KB
JLinkLog	۲۰۱۵/۰۶/۱۲ ۱۲:۵۶ ...	TEXT File	46 KB
JLinkSettings	۲۰۱۵/۰۶/۱۲ ۱۲:۵۴ ...	Configuration sett...	1 KB
main	۲۰۱۵/۰۶/۱۲ ۱۲:۵۶ ...	C File	3 KB
project.uvgui.Mahmood	۲۰۱۵/۰۶/۱۲ ۱۱:۴۳ ...	MAHMOOD File	135 KB
project.uvgui_Mahmood.bak	۲۰۱۵/۲۸/۱۱ ۱۱:۱۰ ...	BAK File	69 KB
project.uvopt	۲۰۱۵/۰۶/۱۲ ۱۱:۴۳ ...	UVOPT File	10 KB
project	۲۰۱۵/۲۹/۱۱ -۷:۱۲ ...	µVision4 Project	17 KB
project_Target 1...	۲۰۱۵/۲۹/۱۱ -۵:۳۹ ...	DEP File	3 KB
project_uvopt.bak	۲۰۱۵/۲۸/۱۱ ۱۰:۱۲ ...	BAK File	9 KB
project_uvproj.bak	۲۰۱۵/۲۸/۱۱ ۱۱:۱۰ ...	BAK File	17 KB
startup_stm32f10x_cl	۲/۲۴/۱۰ ۱۰:۵۹ ...	Assembler Source	16 KB

تغذیه ۵ ولت را به برد آموزشی متصل کنید و بعد از اینکه پروگرامر ST-LINK را به برد متصل نمودید؛ برنامه را با استفاده از دکمه‌ی Download پروگرام نمائید.



3. توضیحات مربوط به برنامه

ابتدا کتابخانه‌های مربوطه فراخوانی شده‌اند.

```
1 #include "stm32f10x_usart.h"
2 #include <stdio.h>
3 #include "bsp.h"
4 #include "delay.h"
```

دستوراتی که در زیر آورده شده است، مربوط به ارسال داده از طریق سریال است که این تنظیمات برای ارسال داده‌ها استفاده می‌شوند تا امکان استفاده از printf وجود داشته باشد.

```

6  //printf init
7  struct __FILE { int handle;};
8
9  FILE __stdout;
10 FILE __stdin;
11 FILE __stderr;
12
13 int fputc(int ch, FILE *f)
14 {
15     while(!USART_GetFlagStatus(COM1,USART_FLAG_TXE));
16
17     USART_SendData(COM1,ch);
18
19     while(!USART_GetFlagStatus(COM2,USART_FLAG_TXE));
20
21     USART_SendData(COM2,ch);
22
23     return ch;
24 }
25
26

```

در تابع اصلی (main) ابتدا کلاک مربوط به GPIO ، Usart و ADC روشن می‌شود و سپس پیکربندی مربوط به هر یک از این پریفرالها انجام می‌گیرد. در داخل حلقه‌ی while مقدار متغیر از طریق USART1, USART2 ارسال می‌شوند. مقدار آنالوگ تبدیل شده دائماً با استفاده از DMA داخل RAM قرار می‌گیرد و در حلقه‌ی (while) مقدار آپدیت شده (توسط DMA) به پورت سریال ارسال می‌شود.

```

1 { #include <stdio.h>
2 { #include "delay.h"
3 { #include "bsp.h"
4   extern uint16_t ADCConvertedValue ;
5
6   //printf init
7   struct __FILE { int handle;} ;
8
9   FILE __stdout;
10  FILE __stdin;
11  FILE __stderr;
12
13  int fputc(int ch, FILE *f)
14  {
15
31 int main(void)
32 {
33     RCC_Configuration();
34     GPIO_Configuration();
35     Com1_Intial();
36     Com2_Intial();
37     dma_init();
38     ADC1_Intial();
39     delay_intial();
40     printf("Naminic STM32 evaluation board\n\n\n");
41     while(1)
42     {
43         printf("adc1 value = %d\n\r",ADCConvertedValue);
44         delayMs(200);
45     }
46     return 0;
47
48 }
49

```