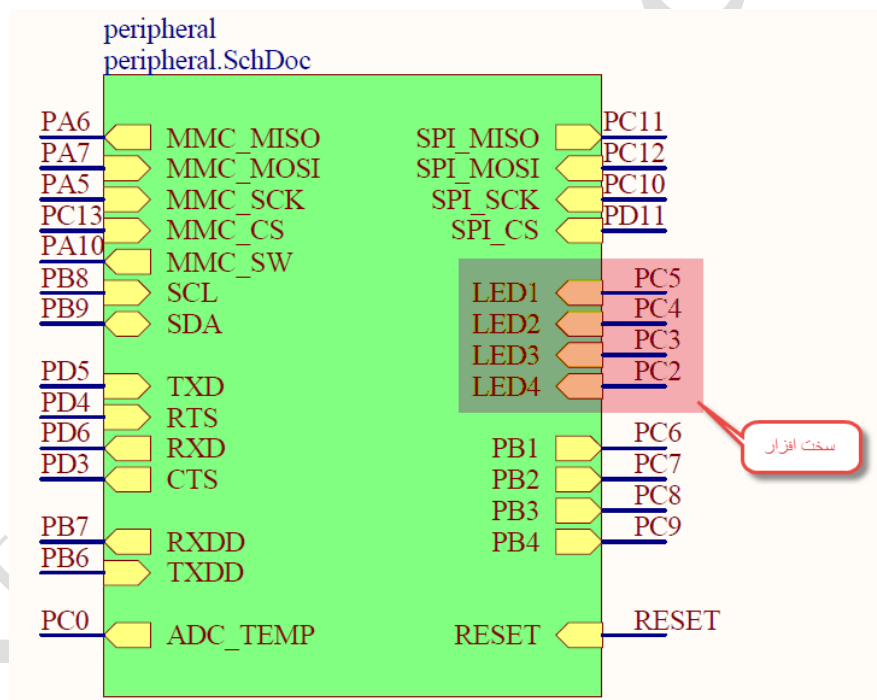


IO Toggle راهنمای استفاده از مثال

هدف از این مثال این است که شما با توابعی که برای ست و ریست کردن یک پین مورد استفاده قرار میگیرد، آشنا شوید.

1. سخت افزار

برای تست این برنامه می‌توانید از چهار LED که بر روی برد آموزشی قرار دارد، استفاده کنید. شماره‌ی پین‌هایی که می‌توانید استفاده کنید، در تصویر زیر نشان داده شده است.

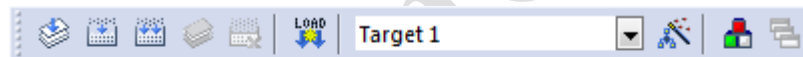


2. آشنایی و کار با برنامه

ابتدا برنامه را با نرم افزار Keil باز کنید.

Name	Date modified	Type	Size
Help	۲۰۱۵/۱۷/۱۱ ۰۱:۲۲ ...	File folder	
obj	۲۰۱۵/۱۷/۱۱ ۱۲:۲۰ ...	File folder	
main	۲۰۱۵/۱۷/۱۱ ۱۲:۲۴ ...	C File	5 KB
project	۲۰۱۵/۱۷/۱۱ ۱۲:۲۰ ...	C compiler map file	63 KB
project.uvgui.mahmood	۲۰۱۵/۱۷/۱۱ ۱۱:۱۴ ...	MAHMOOD File	73 KB
project.uvopt	۲۰۱۵/۱۷/۱۱ ۱۱:۱۴ ...	UVOPT File	10 KB
project	۲۰۱۵/۱۷/۱۱ ۱۱:۱۴ ...	µVision4 Project	17 KB
project_Target1.dep	۲۰۱۵/۱۷/۱۱ ۱۲:۲۶ ...	DEP File	16 KB
project_uvproj.bak	۲۰۱۵/۱۷/۱۱ ۱۰:۰۳ ...	BAK File	0 KB
readme	۲۰۱۱/۰۴/۰۴ ۰۲:۵۷ ...	TXT File	5 KB
startup_stm32f10x_cl	۲۰۱۵/۱۷/۱۱ ۱۲:۲۰ ...	List file	47 KB
stm32f10x_conf	۲۰۱۱/۰۴/۰۴ ۰۲:۵۷ ...	H File	4 KB
stm32f10x_it	۲۰۱۱/۰۴/۰۴ ۰۲:۵۷ ...	C File	5 KB
stm32f10x_it	۲۰۱۱/۰۴/۰۴ ۰۲:۵۷ ...	H File	2 KB
system_stm32f10x	۲۰۱۱/۰۴/۰۴ ۰۲:۵۷ ...	C File	36 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 پروگرم نمائید.



3. توضیحات مربوط به برنامه

ابتدا استراکچری، مربوط به تنظیمات اولیه پین‌های ورودی و خروجی تعریف شده است.

```
GPIO_InitTypeDef GPIO_InitStructure;
```

سپس در تابع main برنامه تنظیمات مربوط به کلاک GPIO و خود GPIO انجام شده است.

```

1  /* ***** Configure the clock and GPIO ***** */
   /* ALL LED Port Periph clock enable */
   RCC_APB2PeriphClockCmd(LED1_GPIO_CLK | LED2_GPIO_CLK | LED3_GPIO_CLK | LED4_GPIO_CLK, ENABLE);

   /* Configure LED1 in output pushpull mode */
   GPIO_InitStructure.GPIO_Pin = LED1_PIN;
   GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
   GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP;
   GPIO_Init(LED1_GPIO_PORT, &GPIO_InitStructure);

   /* Configure LED2 in output pushpull mode */
   GPIO_InitStructure.GPIO_Pin = LED2_PIN;
   GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
   GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP;
2  GPIO_Init(LED2_GPIO_PORT, &GPIO_InitStructure);

   /* Configure LED3 in output pushpull mode */
   GPIO_InitStructure.GPIO_Pin = LED3_PIN;
   GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
   GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP;
   GPIO_Init(LED3_GPIO_PORT, &GPIO_InitStructure);

   /* Configure LED4 in output pushpull mode */
   GPIO_InitStructure.GPIO_Pin = LED4_PIN;
   GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
   GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP;
   GPIO_Init(LED4_GPIO_PORT, &GPIO_InitStructure);

```

و در ادامه توابع و دستورات مربوط به روشن و خاموش کردن led با استفاده از توابع cmsis قرار دارد .

```

GPIO_SetBits(LED1_GPIO_PORT, LED1_PIN);
delayMs(200);
GPIO_SetBits(LED2_GPIO_PORT, LED2_PIN);
delayMs(200);
GPIO_SetBits(LED3_GPIO_PORT, LED3_PIN);
delayMs(200);
GPIO_SetBits(LED4_GPIO_PORT, LED4_PIN);
delayMs(200);

resetAllLed();
delayMs(200);

GPIO_SetBits(LED4_GPIO_PORT, LED4_PIN);
delayMs(200);
GPIO_SetBits(LED3_GPIO_PORT, LED3_PIN);
delayMs(200);
GPIO_SetBits(LED2_GPIO_PORT, LED2_PIN);
delayMs(200);
GPIO_SetBits(LED1_GPIO_PORT, LED1_PIN);
delayMs(200);
while(1)
{
    setAllLed();
    delayMs(200);
    resetAllLed();
    delayMs(200);
}

return 0;
}

```

Naminic.com