

گزارش دستور کار 8 سوال اول

برای انجام این تمرین به پنج تابع نیاز داریم که در ادامه به بررسی آن ها می پردازیم.

در main() ابتدا ADC و USART2 را initialize می کنیم و سپس وارد یک حلقه بی نهایت می شویم که در آن در ابتدا conversion را فعال می کنیم و تا زمانی که کامل شود در یک حلقه بی نهایت صبر می کنیم و وقتی که کامل شد نتیجه آن را می خوانیم و در result ذخیره می کنیم. سپس آن را در یک مقدار از جنس double قرار می دهیم تا تغییرات مورد نیاز را روی آن اعمال کنیم.

در نهایت مقادیر آن را در یک آرایه ذخیره می کنیم تا به تابع USART2_write(int c) بدهیم که نمایش دهد این مقادیر را و در آخر نیز یک تاخیر یک ثانیه ای می دهیم.

در تابع ADC_init(void) کلاک پورت A را فعال می کنیم و mode آن را در حالت output قرار می دهیم و همچنین پین را برای حالت آنالوگ ست می کنیم. در آخر نیز تنظیمات conversion را انجام می دهیم.

```
void ADC_init(void)
{
    RCC->AHB1ENR |= 1;           /* enable GPIOA clock */
    GPIOA->MODER &= ~0x00000C00; /* clear pin mode */
    GPIOA->MODER |= 0x00000400;   /* set pin to output mode */

    /* set up pin PA1 for analog input */
    RCC->AHB1ENR |= 1;           /* enable GPIOA clock */
    GPIOA->MODER |= 0xC;         /* PA1 analog */

    /* setup ADC1 */
    RCC->APB2ENR |= 0x00000100;  /* enable ADC1 clock */
    ADC1->CR2 = 0;               /* SW trigger */
    ADC1->SQR3 = 1;              /* conversion sequence starts at ch 1 */
    ADC1->SQR1 = 0;              /* conversion sequence length 1 */
    ADC1->CR2 |= 1;              /* enable ADC1 */
}
```

در تابع USART2_init (void) نیز ابتدا کلاک را برای پورت A و همچنین برای USART2 ست می کنیم در ادامه نیز مقدار baud را ست می کنیم و Tx را فعال می کنیم.

```
void USART2_init (void) {
    RCC->AHB1ENR |= 1;          /* Enable GPIOA clock */
    RCC->APB1ENR |= 0x20000;     /* Enable USART2 clock */
    GPIOA->AFR[0] &= ~0x0F00;
    GPIOA->AFR[0] |= 0x0700;     /* alt7 for USART2 */
    GPIOA->MODER &= ~0x00F0;
    GPIOA->MODER |= 0x0020;     /* enable alternate function for PA2 */

    USART2->BRR = 0x0683;        /* 9600 baud @ 16 MHz */
    USART2->CR1 = 0x000C;        /* enable Tx, 8-bit data */
    USART2->CR2 = 0x0000;        /* 1 stop bit */
    USART2->CR3 = 0x0000;        /* no flow control */
    USART2->CR1 |= 0x2000;       /* enable USART2 */

    NVIC_EnableIRQ(USART2_IRQn);
}
```

در تابع USART2_write (int ch) نیز مقدار ورودی که به آن داده شده را نمایش می دهد.