

## گزارش عملی دستورآزمایش نهم ریزپردازنده

امیرحسین ادواری ۹۸۲۴۳۰۰۴

زهرا حیدری ۹۸۲۴۳۰۲۰

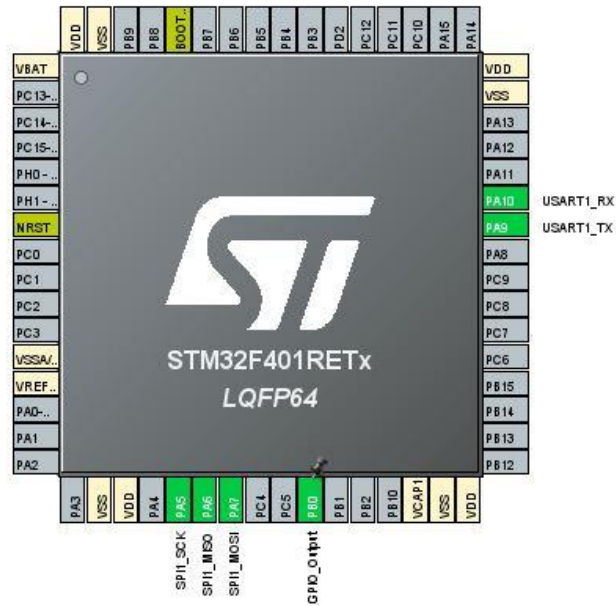
فرستنده (میکروی اول):

Pin	PA10	PA9	PA1
Type	Output	Input	Output
Sym	ADC-0	USART1-RX	USART1-TX

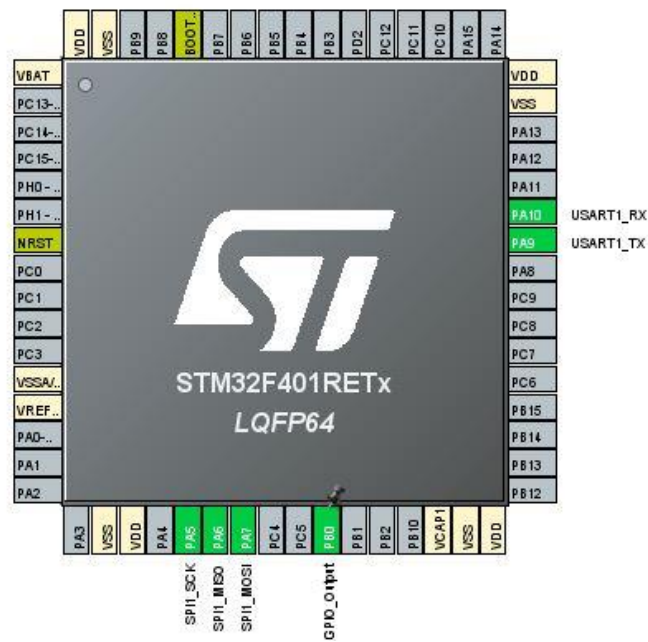
گیرنده (میکروی دوم):

Pin	PA10	PA9	PB0	PA7	PA5
Type	Input	Output	Output	Output	Output
Sym	RX	TX	GPIO-OUTPUT	MOSI	SCLK

میکروی دوم :



میکروی اول:



شرح کد میکروی اول:

```
void sendStr(char* string){
    HAL_UART_Transmit(&huart1,(uint8_t*) string , strlen(string), HAL_MAX_DELAY);
}
```

این تابع رشته ورودی را توسط UART ارسال می‌کند.

```
void sendStr(char* string){
    HAL_UART_Transmit(&huart1,(uint8_t*) string , strlen(string), HAL_MAX_DELAY);
}

void sendNumber(uint32_t num){
    char number[2];
    sprintf(number, "%u", num);
    sendStr(number);
}

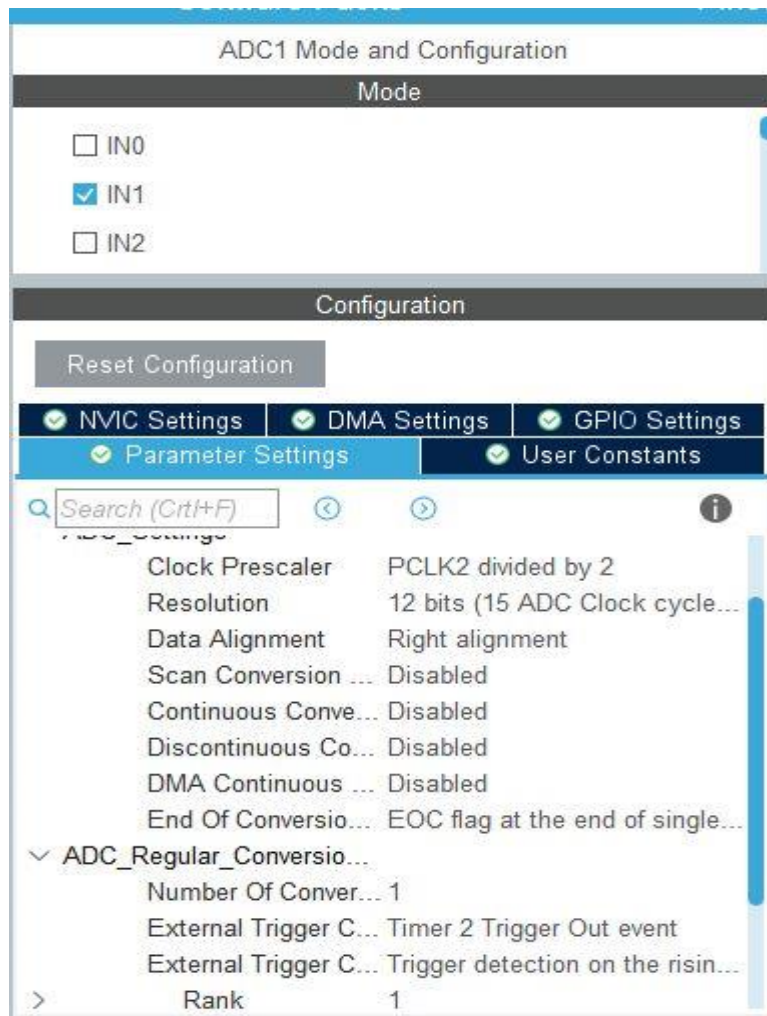
void HAL_ADC_ConvCpltCallback(ADC_HandleTypeDef* hadc){
    sendNumber( HAL_ADC_GetValue(&hadc1) * VREF * 10 / 4095 );
}
```

تابع `sendNumber` عدد را گرفته به رشته تبدیل کرده و سپس توسط `uart` ارسال می‌کند.

کند.

تابع سوم زمانی کال می‌شود که تبدیل آنالوگ به دیجیتال تکمیل شود. کانال ۲ تایمر ۲ که روی `output compare` ست شده است موجب تحریک و آغاز عملیات `conversion` شده و پس از پایان عملیات `conversion` مربوط به `adc` این تابع کال می‌شود لذا در این تابع مقدار `convert` شده را توسط `uart` ارسال می‌کنیم.

تنظیمات مربوط به `adc` در میکروی اول :



تنظیمات مربوط به تایمر در میکروی اول :

TIM2 Mode and Configuration

Mode

Channel1
Disable
Channel2
Output Compare No Output
Channel3
Disable

Configuration

Reset Configuration

NVIC Settings
DMA Settings

Parameter Settings
User Constants

Configure the below parameters :

Search (Ctrl+F)

Counter Settings

Prescaler (PSC - ...
15999
Counter Mode
Up
Counter Period (A...
999
Internal Clock Divi...
No Division
auto-reload preload
Disable

Trigger Output (TRGO) P...

Master/Slave Mod...
Disable (Trigger input effect ...
Trigger Event Sele...
Update Event

Output Compare No Outp...

Mode
Frozen (used for Timing base)
Pulse (32 bits value)
0

تنظیمات یوآرت میکروی اول:

Mode

Mode

Asynchronous

Hardware Flow Control (RS232)

Disable

Configuration

Reset Configuration

✓

NVIC Settings

✓

DMA Settings

✓

GPIO Settings

✓

Parameter Settings

✓

User Constants

Configure the below parameters :

Search (Ctrl+F)

⏪

⏩

i

Basic Parameters

Baud Rate

115200 Bits/s

Word Length

8 Bits (including Parity)

Parity

None

Stop Bits

1

Advanced Parameters

Data Direction

Transmit Only

Over Sampling

16 Samples

شرح کد میکروی دوم:

```
static volatile char left = '0';
static volatile char right = '0';
```

```
void HAL_UART_RxCpltCallback(UART_HandleTypeDef *huart){
    if( huart->Instance == USART1 ){
        sendStr(ubuff);
        left = ubuff[0];
        right = ubuff[1];
        HAL_UART_Receive_IT(&huart1, (uint8_t*) ubuff, 2);
    }
}
```

دو متغیر به ترتیب برای نگهداشتن مقدار سگمنت اول و دوم تعریف می‌کنیم. هرزمانی که بافر مربوط به دریافت توسط **uart** مان پر شد، این مقادیر را توسط دیتای دریافت شده آپدیت می‌کنیم و سپس بافر را برای دریافت بعدی تنظیم می‌کنیم (در صورت پر شدن مجدد تابع **کال** شود و روال تکرار شود)

```
void apply(uint8_t command, uint8_t data){
    HAL_GPIO_WritePin(GPIOB, GPIO_PIN_0, GPIO_PIN_RESET);
    HAL_SPI_Transmit(&hspi1, &command, 1, HAL_MAX_DELAY);
    HAL_SPI_Transmit(&hspi1, &data, 1, HAL_MAX_DELAY);
    HAL_GPIO_WritePin(GPIOB, GPIO_PIN_0, GPIO_PIN_SET);
}
```

این تابع دو عدد ۸ بیتی که یکی از آن‌ها نوع دستور (۴ بیت پایین آن) و دیگری دیتای مربوط به آن دستور را دریافت کرده، سپس با صفر نگه‌داشتن **CS** یا همان **SS** آنرا به دیوایس **MAX7721** توسط **SPI** ارسال کرده و مجدداً **CS** را یک می‌کند. (ابتدا هشت بیت شامل کامند و سپس هشت بیت شامل دیتا توسط **MOSI** از میکرو به **MAX7721** ارسال می‌شوند)

```
void set_digit_0(uint8_t data){
    apply(0x1, data);
}
void set_digit_1(uint8_t data){
    apply(0x2, data);
}
```

این دو تابع دستورات مربوط به نوشتن عدد روی سگمنت‌ها را توسط تابع **apply** انجام می‌دهند. مشخصاً کد دستور ثابت و دیتای دستور متغیر خواهد بود (دیتا عملاً همان عددی است که بایستی نوشته شود)

```

/* Initialize all configured peripherals */
MX_GPIO_Init();
MX_SPI1_Init();
MX_USART1_UART_Init();
/* USER CODE BEGIN 2 */
    init_device();
    NVIC_EnableIRQ(USART1_IRQn);
    HAL_UART_Receive_IT(&huart1, (uint8_t*) ubuff, 2);

/* USER CODE END 2 */

/* Infinite loop */
/* USER CODE BEGIN WHILE */
while (1)
{
    set_digit_0(left);
    set_digit_1(right);
    HAL_Delay(101);
/* USER CODE END WHILE */

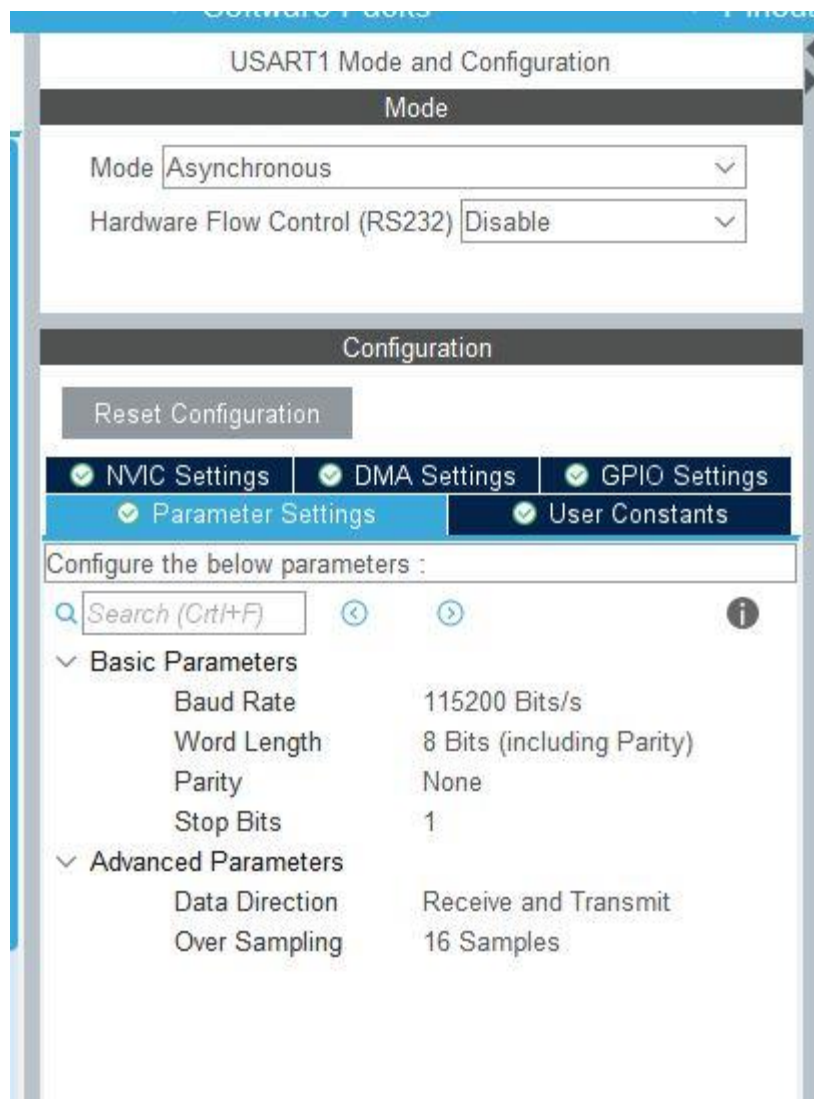
/* USER CODE BEGIN 3 */
}
/* USER CODE END 3 */
}

```

در حلقه اصلی برنامه مرتبا سگمنت‌ها را روی مقادیر فعلی **left** و **right** (که طبق آنچه پیشتر گفته شد پس از پر شدن بافر دریافت **uart** توسط مقادیری که میکروی اول ارسال کرده آپدیت می‌شوند) تنظیم می‌کنیم. (اینکار با ارسال دستورات و نیز دیتای متناسب با نوشتن عدد روی سگمنت‌ها به **MAX7721** انجام می‌شود که پیشتر شرح داده شد)

تنظیمات مربوط به **UART** در میکروی دوم:





تنظیمات مربوط به SPI در میکروی دوم:

SPI1 Mode and Configuration

Mode

Mode Full-Duplex Master

Hardware NSS Signal Disable

Configuration

Reset Configuration

✓ NVIC Settings

✓ DMA Settings

✓ GPIO Settings

✓ Parameter Settings

✓ User Constants

Configure the below parameters :

Search (Ctrl+F)

⏪

⏩

i

Basic Parameters

Frame Format Motorola

Data Size 8 Bits

First Bit MSB First

Clock Parameters

Prescaler (for Bau... 16

Clock Polarity (CP... Low

Clock Phase (CPH... 1 Edge

Advanced Parameters

CRC Calculation Disabled

NSS Signal Type Software

پروتئوس

