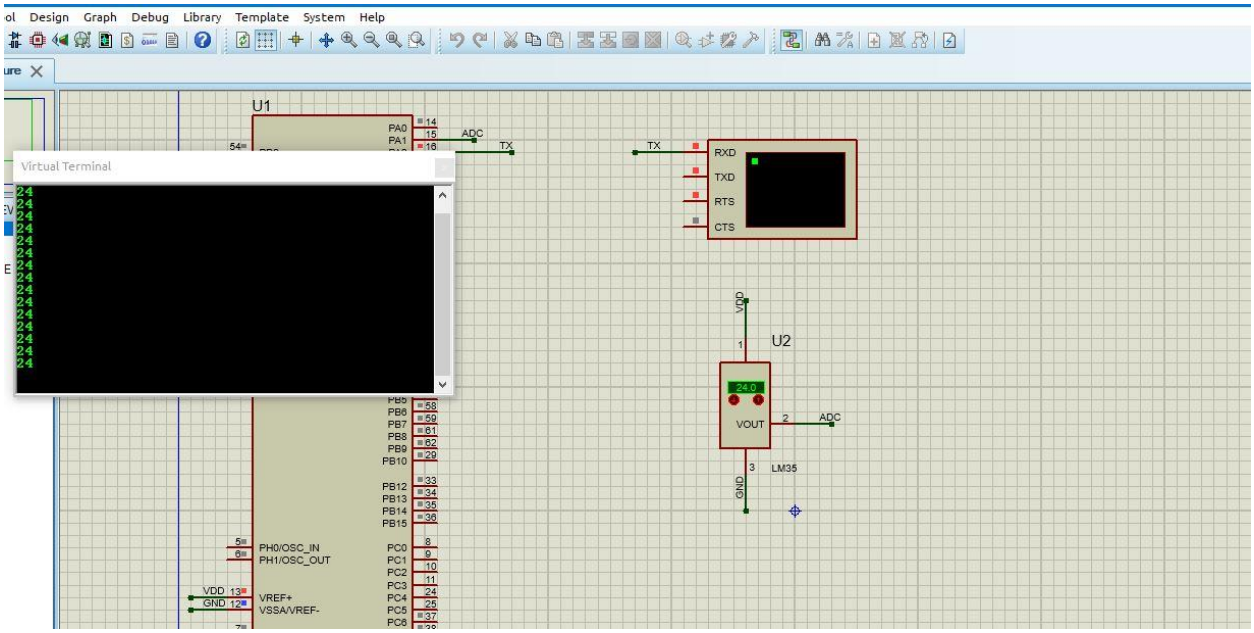
امیرحسین ادواری ۹۸۲۴۳۰۰۴

زهرا حیدری ۹۸۲۴۳۰۲۰

بخش عملی

سوال اول)

پین PA1 را برای ورودی آنالوگ مربوط به دماسنج و پین PA2 را برای TX مربوط به USART استفاده می‌کنیم.



پین TX خروجی میکرو را به RX ترمینال وصل می‌کنیم.

از شرح کدهایی که پیشتر در دستورکارهای گذشته به تفصیل آورده شده‌اند پرهیز می‌کنیم.

```
void usart2_init(uint32_t baudrate){

        RCC->AHB1ENR |= RCC_AHB1ENR_GPIOAEN;
        RCC->APB1ENR |= RCC_APB1ENR_USART2EN;
        GPIOA->MODER |= GPIO_MODER_MODER2_1;//Pin2 mode AF
        GPIOA->OSPEEDR|= GPIO_OSPEEDER_OSPEEDR2_1;
        GPIOA->AFR[0]|= 0x00000700;//Set the AF to AF7(USART1~3);

        USART2->CR1 |= USART_CR1_UE;
        USART2->BRR = SystemCoreClock/baudrate;
        USART2->CR1 |= USART_CR1_TE;
}


void init_adc(){

        /* 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 */
        // read_adc() to read value

}
```

در این دو تابع USART2 و ADC را کانفیگ می‌کنیم.

برای USART خط فرستنده (TX) را فعال کرده و روی baudrate ورودی (که ما از 115200 استفاده کرده‌ایم) قرار می‌دهیم. مود پین TX که روی PA2 قرار دارد را AF قرارداده و از آنجایی که USART2 مورد هفتم آلترنیت فانکش میباشد. در 4 بیت متناظر PA2 در AFR[0] عدد 7 قرار می‌دهیم. (لازم بذکر است سرعت پین را نیز بالا می‌بریم)

برای ADC نیز طبق کدهای اسلاید یک Sequence با طول یک (صرفا همان پین PA1 که چنل 1 روی آن قرار می‌گیرد) ایجاد میکنیم، پین مذکور را روی مود Analog قرار می‌دهیم(gpioa را نیز فعال می‌کنیم) adc را فعال کرده و روی software trigger قرار می‌دهیم.

```
// send a char via uart
void usart2_send_char(uint8_t ch){
    USART2->DR = ch;
    while(!READ_BIT(USART2->SR, USART_SR_TC)) {}
}
```

در این تابع صرفا یک کاراکتر را توسط TX ارسال می‌کنیم (کد مشابه کداسلاید هست)

```
uint32_t read_adc(){
    while(1){
        ADC1->CR2 |= 0x40000000;  // Start
        while( !(ADC1->SR & 2)){} // Wait for conv to finish
        float val =  (ADC1->DR/4095.0)*5*100;
        return (uint16_t) val;
    }
}
```

در این تابع مقدار پین adc را میخوانیم، ابتدا conversion را شروع کرده و منتظر اتمام آن می‌مانیم. سپس مقدار آنرا با توجه Vref+ و -Vref که به ترتیب 5 و 0 هستند scale می‌کنیم و بعنوان خروجی برمیگردانیم.

```
// send digits of an integer:
void printDigit(int N){
        if( N == 0 ){
            usart2_send_char(intToChar(N));
            return;
        }
        if( N < 0 ) {
            usart2_send_char('-');
            N = -N;
        }

    char arr[30];
    int i = 0;
    int j, r;

    while (N != 0) {
        r = N % 10;
        arr[i] = r;
        i++;
        N = N / 10;
    }
    for (j = i - 1; j > -1; j--) {
        usart2_send_char(intToChar(arr[j]));
    }
        usart2_send_char('\n\r');
}
```

در این تابع یک عدد را رقم به رقم برای نمایش روی ترمینال ارسال می‌کنیم. (ترتیب کار آن پیشتر طی چاپ روی lcd توضیح داده شده، صرفا بجای چاپ روی lcd برای چاپ روی ترمینال توسط usart ارسال می‌کنیم.

```
int main(void){

    //Configurations :
    init_adc();
    usart2_init(115200);

    // Main Loop of program:
    delay(2000);
    while(1){
        delay(2000);
        printDigit( (uint16_t) read_adc() );
    }
}
```

در تابع main مرتبا به صورت دوره‌ای پین adc را خوانده و روی usart ارسال می‌کنیم تا چاپ شود.

سوال دوم)

Pinout و منطق ماشین‌حساب مشابه توضیحات تمرین سوم است با این تفاوت که پین-های LCD را از PA به PC می‌بریم. و نیز USART2 را فعال می‌کنیم (PA2 خط TX و PA3 خط RX می‌باشد)

```
93  volatile uint8_t ubuff[1];
94  // Will be filled after USART interrupt ( after a character is received )
95  volatile uint8_t charFromUSART = ' ';
96
```

ubuff بافری‌است که قراراست هنگام دریافت دیتا در آن قرار بگیرد (از خط RX) متغیر بعدی به همین منظور ایجاد شده‌است.

```
145 void logStr(char* string, uint16_t size){
146     HAL_UART_Transmit(&huart2,(uint8_t*) string , size, HAL_MAX_DELAY);
147 }
148 void logNum(uint32_t input){
149   char number[10];
150   sprintf(number, "%u", input);
151   HAL_UART_Transmit(&huart2,(uint8_t*) number , 10, HAL_MAX_DELAY);
152 }
```

در تابع اول یک استرینگ را ضمن دریافت سایز آن توسط تابع Transmit ارسال می‌کند.

در تابع دوم نیز یک عدد را گرفته به استرینگ تبدیل کرده و ارسال می‌کند.

```
446 void EXTI0_IRQHandler(void){
447   EXTI->PR |= MASK(0);
448   NVIC_ClearPendingIRQ(EXTI0_IRQn);
449   char clickedButton = clicked_button();
450
451   char btn[1] = { clickedButton };
452   logStr(btn, 1);
453
454   // pick the filled var after USART ISR completion
455   clickedButton = charFromUSART;
456
```

سپس پس از تنظیم USART2 در تابعی که دریافت فرمان یا کاراکترها را هندل میکند، پس از دریافت کاراکتر مربوطه آنرا توسط USART ارسال می‌کنیم (روی ترمینالی که به پروژه اضافه شده نیز قابل رویت است). پس از ارسال هندلر مربوط به اینتراپت دریافت دیتا توسط USART کال شده و مقدار ارسال شده توسط USART را در متغیر charFromUSART قرار می‌دهد.

```
595   //Configurations
596   HAL_UART_Receive_IT(&huart2, (uint8_t*) ubuff, 1);
597
598   printInLine("Advari-Heidari");
599   printInLine("Welcome");
600   HAL_Delay(1000);
601   clear_line_2();
602
603   HAL_NVIC_SetPriority(SysTick_IRQn, 0, 0);
604   HAL_NVIC_SetPriority(EXTI0_IRQn, 2, 2);
605   HAL_NVIC_SetPriority(USART2_IRQn, 0,0);
```

در تابع main در ابتدا اینتراپت مربوطه را ست می‌کنیم تا در زمان پر شدن بافر هندلر را اجرا کند.

سپس اولویت آنرا زیاد می‌کنیم تا در EXTI نیز اجرا شود. (اولویت Systick را نیز بالا می‌بریم)

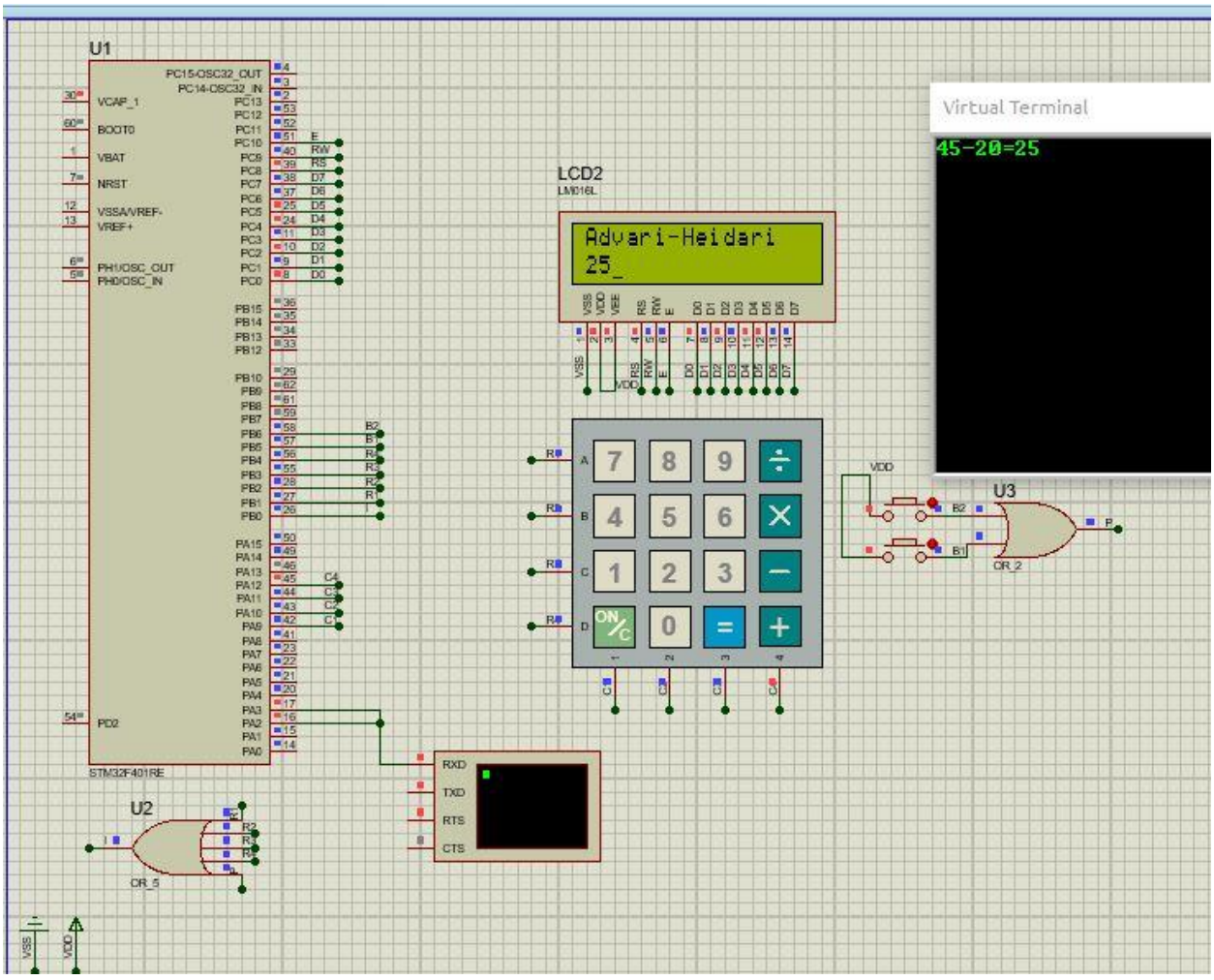
```
void printDigit(int N){
  // first we send the char via uart tx, then we receive it with rx
  // at last we print it on lcd
    if( N == 0 ){
      char btn[1] = { intToChar(N) };
      logStr(btn, 1);
      printInLine2(charFromUSART);
      return;
    }
    if( N < 0 ) {
      char btn[1] = { '-' };
      logStr(btn, 1);
      printInLine2(charFromUSART);
      N = -N;
    }

    char arr[30];
    int i = 0;
    int j, r;

    while (N != 0) {
        r = N % 10;
        arr[i] = r;
        i++;
        N = N / 10;
    }
    for (j = i - 1; j > -1; j--) {
      char btn[1] = { intToChar(arr[j]) };
      logStr(btn, 1);
      printInLine2(charFromUSART);
```

از این تابع برای ارسال نتیجه استفاده می‌کنیم. این تابع یک عدد را گرفته و رقم‌های آنرا به‌ترتیب از سمت چپ پس ارسال و دریافت توسط یوآرت روی ال‌سی‌دی نمایش می‌دهیم. (از این تابع در پروژه سوم استفاده شده، صرفا تغییر جزیی مربوط به ارسال و دریافت یوآرت اعمال شده است)

پروتئوس


U1
STM32F401RE
LCD2
LM016L
Advari-Heidari
25_
Virtual Terminal
45-20=25
U2
OR_5
U3
OR_2
RXD
TXD
RTS
CTS
VDD
VSS