آزمایش 8

طاها موسوی 98243058

نیلوفر مرادی جم 97243063

گروه 2

سوالات تحلیلی:

1 – مفهوم ارتباط سریال همزمان (سنکرون) و غیر همزمان (آسنکرون) را شرح دهید.

در انتقال سنکرون، داده ها بلوک بلوک فرستاده می شوند و انتقال از نوع فول دوبلکس می باشد. همگام سازی بین فرستنده و گیرنده اجباری است. یک رابط سریال سنکرون خط داده خود را با یک سیگنال ساعت جفت کرده و همه دستگاه های موجود در آن یک کلاک مشترک دارند که باعث شده انتقال سریال ساده تر سریعتر انجام شود. همچنین در انتقال سکرون، بین داده ها شکاف وجود ندارد و برای انتقال داده زیاد کارآمدتر و قابل اعتمادتر از انتقال آسنکرون است.

در انتقال آسنکرون، داده ها بایت بایت یا کارکتر کارکتر فرستاده می شوند و انتقال از نوع نیمه دوبلکس می باشد. نیاز به همگام سازی بین فرستنده و گیرنده ندارد. در این انتقال بیت های شروع و توقف با داده ها اضافه می شوند. این نیز ارتباط سریالی است اما انتقال داده بدون هیچ گونه پشتیبانی از سیگنال ساعت خارجی اتفاق میفتد. به عنوان مثال UART، RS232 آسنکرون هستند.

2 – منظور از Baud Rate چیست؟ و مقدار آن در میکروکنترلر STM32F401به چه پارامترهایی بستگی دارد و چگونه محاسبه میشود؟

Baud Rate سرعتی است که در آن تعدادی عنصر سیگنال یا تغییر سیگنال هنگام عبور از یک رسانه انتقال در هر ثانیه اتفاق میفتد و درواقع سرعتی است که اطلاعات در یک کانال ارتباطی منتقل می شود . هر چه بیشتر

باشد، داده ها سریعتر ارسال/دریافت می شوند. در زمینه پورت سریال، "baud <number>" به این معنی است که پورت سریال قادر به انتقال حداکثر <number> بیت در ثانیه است.

- برای محاسبه نرخ بیت یک کانال ارتباطی استفاده می شود.
- می تواند bandwidth مورد نیاز برای انتقال سیگنال را تعیین کند.
- این یک پارامتر تنظیم است برای انتقال یک سیگنال. (یعنی تراکم شبکه در شبکه داده را تنظیم می کند)

Baud Rate برای گیرنده و فرستنده (RX و Tx) هر دو روی یک مقدار تنظیم شده است همانطور که در mantissa و مقادیر کسری usartdiv برنامه ریزی شده است. همانطور که مشاهده می شود، register 16 بیت است، حداقل 4 بیت قسمت کسری و باقیمانده قسمت عدد صحیح است. چنین طراحی می تواند Baud Rate را دقیق تر کند. با توجه به تولید Baud Rate، قسمتی برای توضیح وجود دارد:

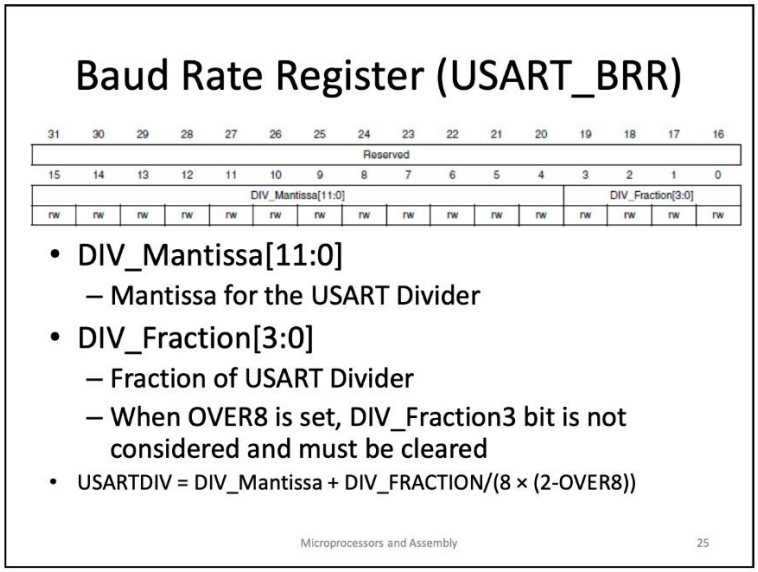
Baud Rate کسری ایجاد می شود: گیرنده و فرستنده (RX و TX) هر دو روی مقادیر پیکربندی شده در register های عدد صحیح و اعشاری Usartdiv تنظیم می شوند.

این معادله زیر را برای Baud Rate به دست می دهد:

Tx/rx baud = CK_APB1/(8 x (2-over8) x usartdiv)

جایی که Usartdiv یک unsigned fixed point است که در رجیستر USART_BRR کدگذاری شده است.

Tx/rx baud = CK_APB1/(8 x Usartdiv)



Error calculation for programmed baud rates at f_{PCLK} = 8 MHz or f_{PCLK} = 12 MHz, oversampling by 16

Oversampling by 16 (OVER8=0)							
Baud rate7		f_{PCLK} = 8 MHz			f_{PCLK} = 12 MHz		
S.No	Desired	Actual	Value programmed in the baud rate register	% Error = (Calculated - Desired) B.rate / Desired B.rate	Actual	Value programmed in the baud rate register	% Error
1	1.2 KBps	1.2 KBps	416.6875	0	1.2 KBps	625	0
2	2.4 KBps	2.4 KBps	208.3125	0.01	2.4 KBps	312.5	0
3	9.6 KBps	9.604 KBps	52.0625	0.04	9.6 KBps	78.125	0
4	19.2 KBps	19.185 KBps	26.0625	0.08	19.2 KBps	39.0625	0
5	38.4 KBps	38.462 KBps	13	0.16	38.339 KBps	19.5625	0.16
6	57.6 KBps	57.554 KBps	8.6875	0.08	57.692 KBps	13	0.16
7	115.2 KBps	115.942 KBps	4.3125	0.64	115.385 KBps	6.5	0.16
8	230.4 KBps	228.571 KBps	2.1875	0.79	230.769 KBps	3.25	0.16
9	460.8 KBps	470.588 KBps	1.0625	2.12	461.538 KBps	1.625	0.16
10	921.6 KBps	NA	NA	NA	NA	NA	NA
11	2 MBps	NA	NA	NA	NA	NA	NA
12	3 MBps	NA	NA	NA	NA	NA	NA

Microprocessors and Assembly 22

3 – پایداری در برابر نویز در خطوط انتقال موازی بیشتر است یا انتقال سریال؟ چرا؟ برای مشکل نویز در درگاه UART چه راه حل هایی اندیشیده شده است؟

در خطوط انتقال سریال پایداری در برابر نویز نسبت به خطوط انتقال موازی بیشتر است، چرا که در هر لحظه تنها یک بیت ارسال می‌شود در حالی که در ارتباط موازی در هر لحظه چند بیت ارسال می‌شود.

برای حل مشکل نویز درuart ، می‌توان دیتای ورودی را oversample کرد که با استفاده از آن با انجام voting می‌توانیم نویز را کاهش دهیم.

رفرنس های سوالات تحلیلی:

- کلاس درس و اسلاید های درسی

https://www.geeksforgeeks.org/difference-between-synchronous-and-asynchronous-transmission/

https://pijaeducation.com/communication/serial-communication-methods-synchronous-asynchronous/

https://www.geeksforgeeks.org/baud-rate-and-its-importance/

https://topic.alibabacloud.com/a/stm32-baud-rate-calculation_8_8_31412563.html

https://www.setra.com/blog/what-is-baud-rate-and-what-cable-length-is-required-1

دستور کار:

سوال اول:

مقداردهی اولیه پین ها:

```
5
6  void usart2_init(uint32_t pclk, uint32_t baudrate){
7    uint32_t temp = 0x00;
8    uint32_t integer = 0x00;
9    uint32_t fraction= 0x00;
10   integer = ((25*pclk*1000000)/(4*baudrate));
11   temp = (integer/100)<<4;
12   fraction = integer-(100*(temp>>4));
13   temp |= ((((fraction*16)+50)/100))&((uint8_t) 0x0F);
14
15   RCC->APB1ENR |= RCC_APB1ENR_USART2EN;
16   GPIOA->MODER |= GPIO_MODER_MODER2_1;//Pin2 mode AF
17   GPIOA->OSPEEDR|= GPIO_OSPEEDER_OSPEEDR2_1;
18   GPIOA->AFR[0]|= 0x00000700;//Set the AF to AF7(USART1~3);
19   SET_BIT(RCC->AHB1RSTR,RCC_APB1RSTR_USART2RST);
20   CLEAR_BIT(RCC->AHB1RSTR,RCC_APB1RSTR_USART2RST);
21   USART2->CR1 |= USART_CR1_UE;
22   USART2->BRR = temp;
23   USART2->CR1 |= USART_CR1_TE;
24 }
```

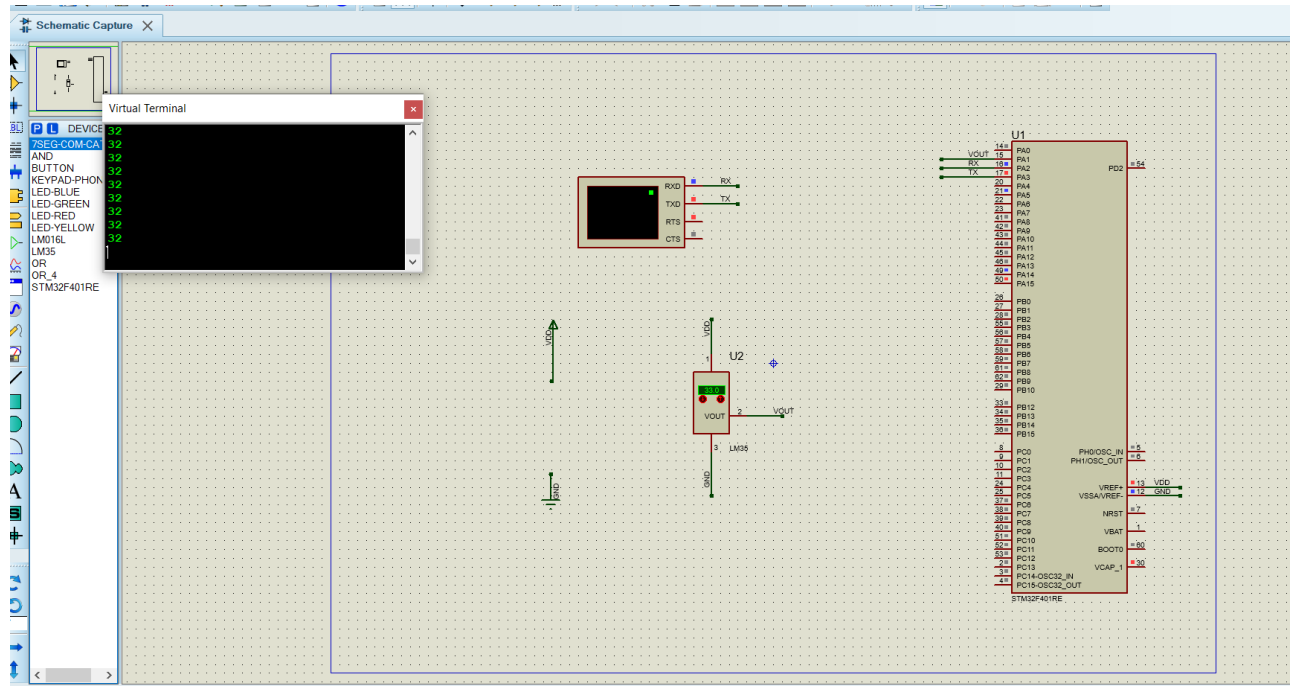
ارسال به وسیله درگاره uart :

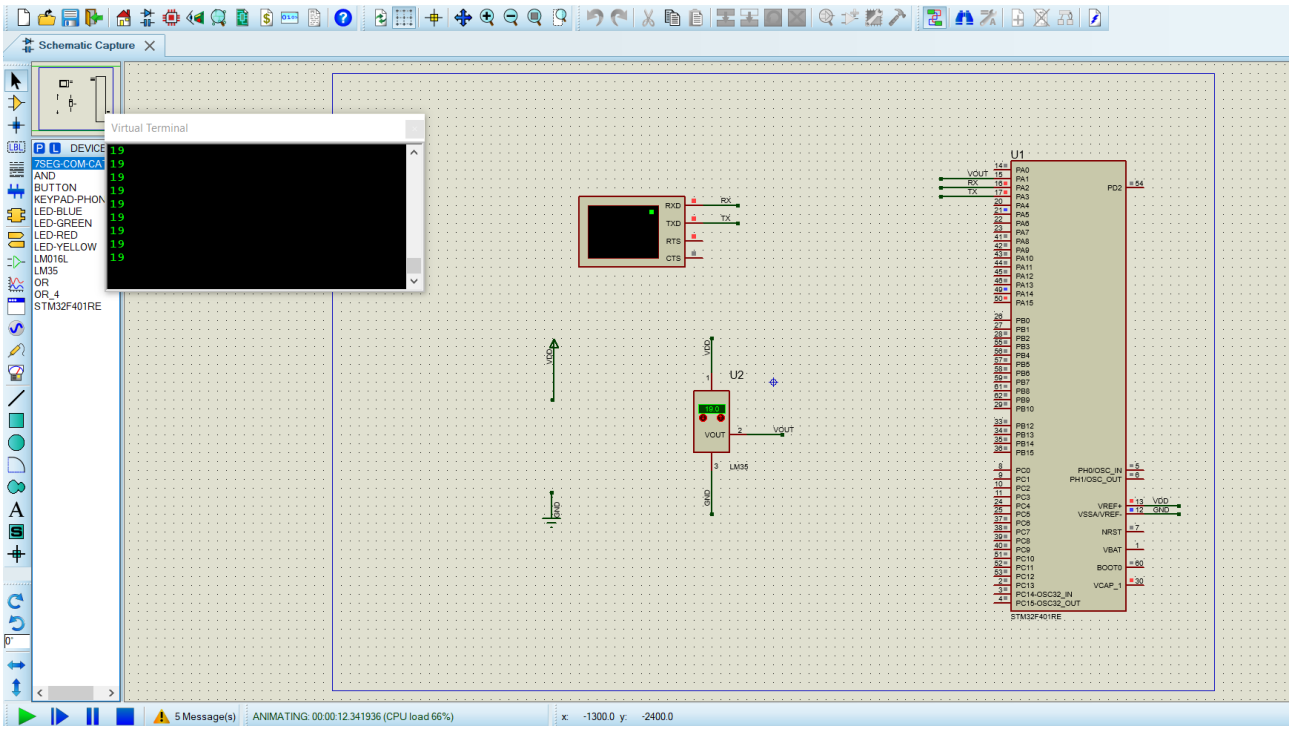
```
26
27 void usart2_send(uint16_t data) {
28   USART2->DR=(data&(uint16_t)0x01ff);
29   while(!READ_BIT(USART2->SR, USART_SR_TC)){}
30 }
31
```

در این تابع تغییرات دما را میخوانیم و با توجه به عدد به دست آمده آن را به صورت رشته دراورده و هر کارکتر را جدا با uart ارسال میکنیم و در نهایت به همین شکل nextLine چاپ میکنیم. برای هر بار تکرار حلقه 1000 میلی ثانیه تاخیر گذاشتیم:

```
58  while (1) {
59    ADC1->CR2 |= ADC_CR2_SWSTART;     /* start a conversion */
60    while(!(ADC1->SR & 2)) {}    /* wait for conv complete */
61    result = ADC1->DR;           /* read conversion result */
62    result = result*0.124;
63
64    /*send number*/
65    int size = 0;
66    int num = result;
67    while (result > 0) {
68      result /= 10;
69      size ++;
70    }
71
72    char number_str[size];
73    sprintf(number_str, "%u", num);
74
75    for (int i = 0; i < size; i++) {
76      usart2_send(number_str[i]);
77    }
78
79    /*next line*/
80    char* string = "\n\r";
81    for (int i = 0; i < 4; i++) {
82      usart2_send(string[i]);
83    }
84
85    delay_ms(1000);
```

نتیجه کار در پروتئوس:

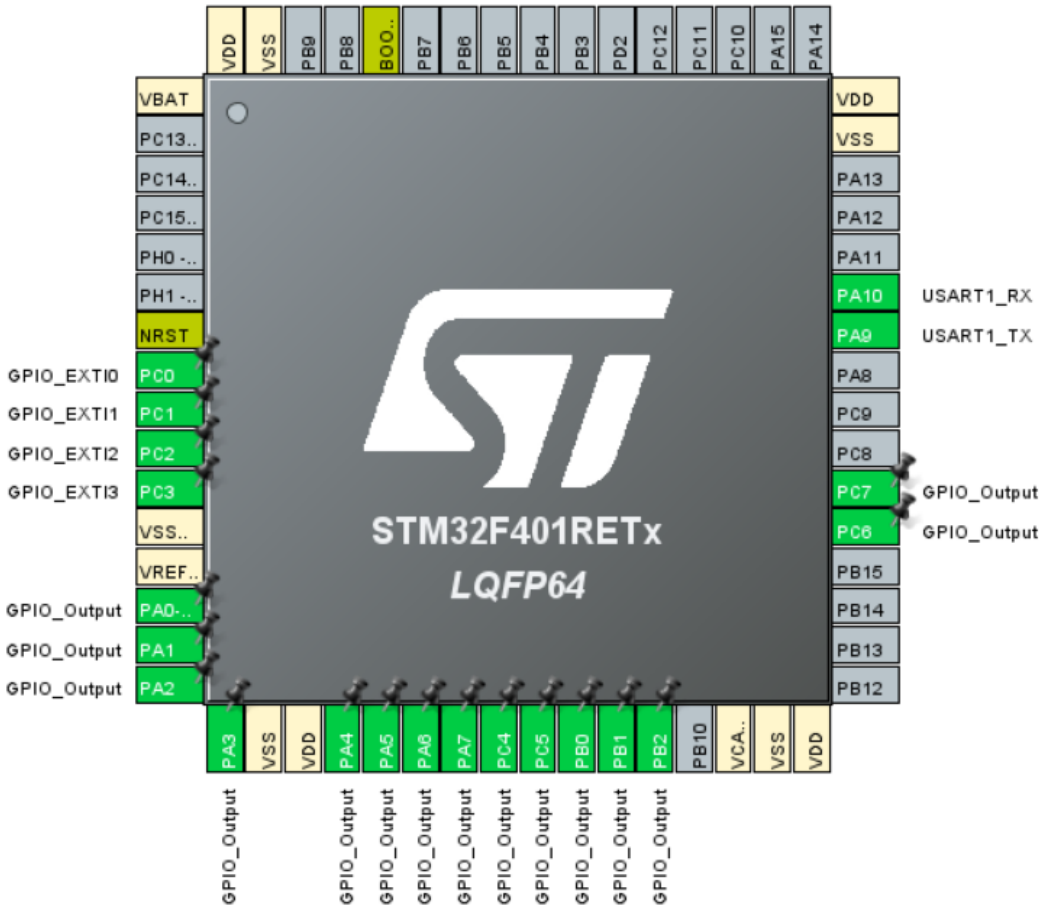




همانطور که مشاهده میکنید با تغییر دمای دماسنج، ولتاژ روی ترمینال نمایش داده میشود.

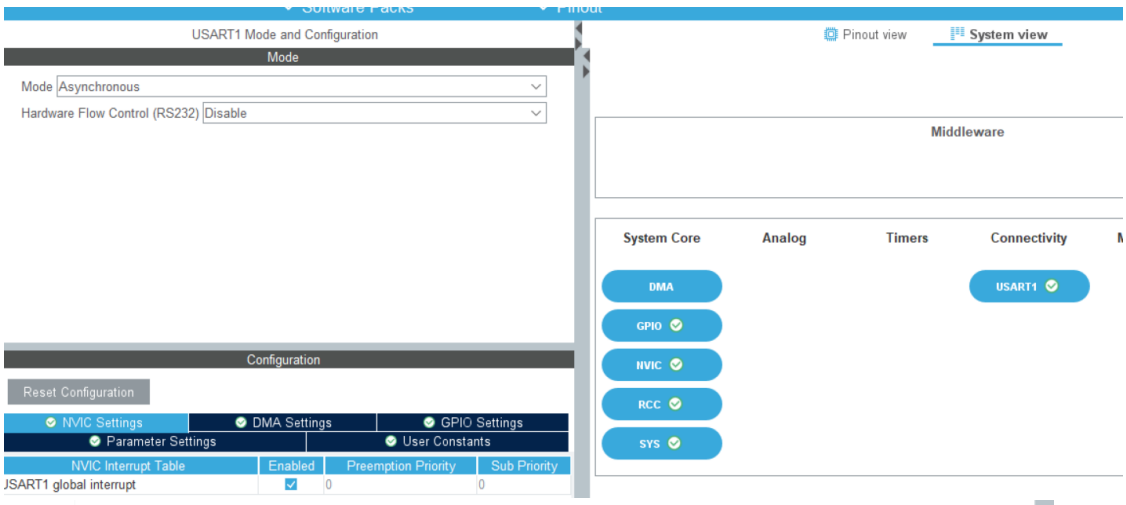
سوال دوم:

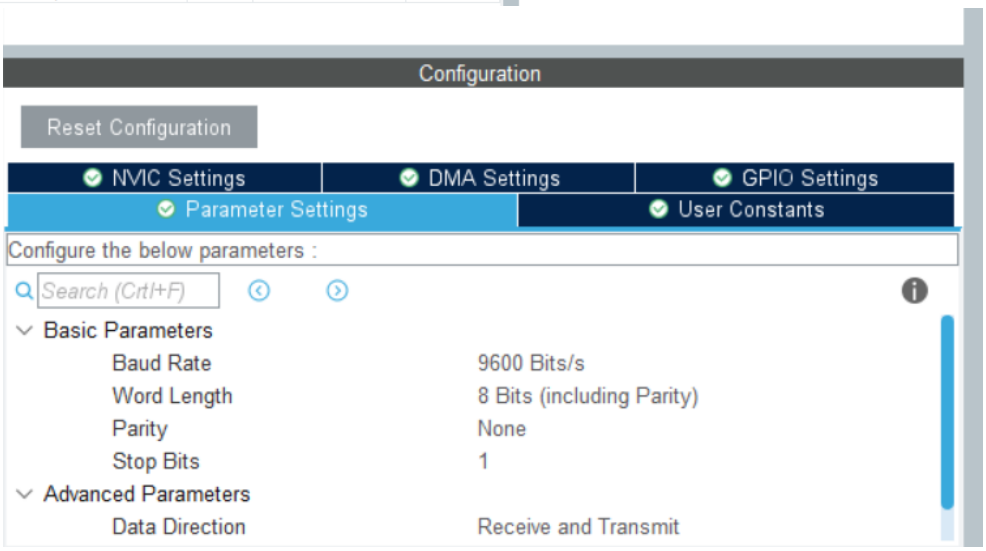
تنظیمات در Stm32cubemx به صورت زیر است:



Pin	PC0	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PB0	PB1	PB2
type	EXTI0	EXTI1	EXTI2	EXTI3	OUT	OUT	OUT	OUT	OUT	OUT	OUT

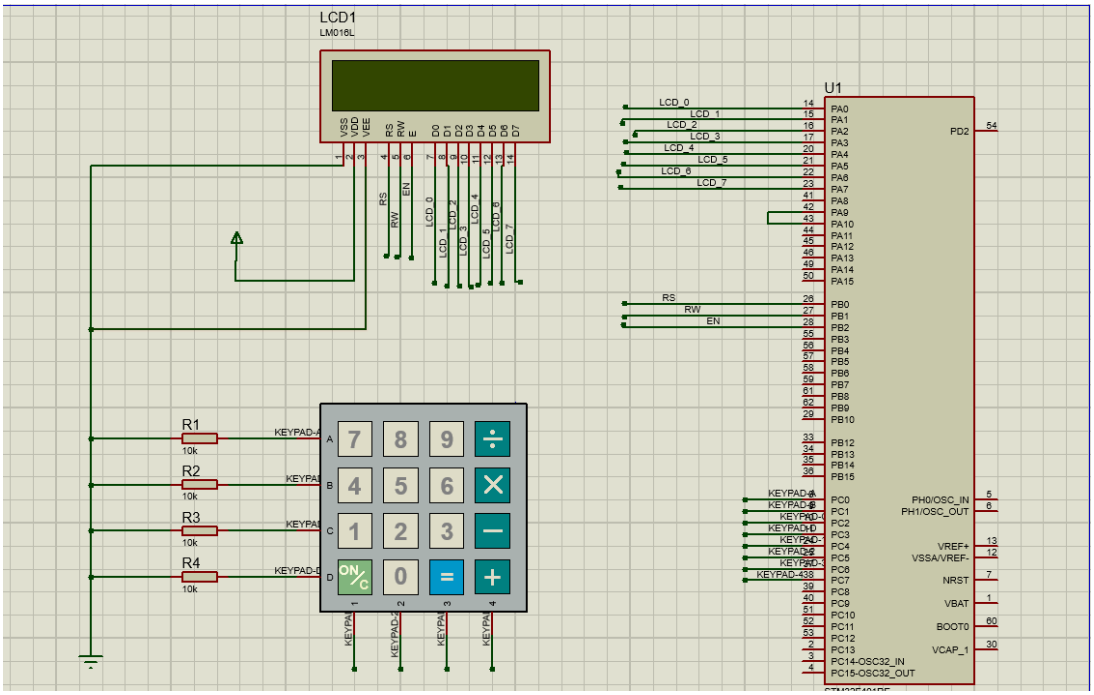
Pin	PA0	PA1	PA2	PA3	PA4	PA5	PA6	PA7	PA9	PA10
type	OUT	OUT	OUT	OUT	OUT	OUT	OUT	OUT	USART1_RX	USART1_TX





در بخش شبیه سازی پروتئوس به اجزای زیر نیاز داشتیم و از همان شکل در صورت سوال کمک گرفتیم:

- STM32F401RE
- Lcdکاراکتری 2 * 16
- Keypad ماتریسی 4 * 4



:Keil

از آنجایی که این دستور کار مشابه دستور کار آزمایش سوم است. بسیاری از متد های آن هم نظیر محاسبه نتیجه، نمایش آن، روشن کردن lcd، پاک کردن آن و .... همانند قبل است.

```
  long long int reversedNumber= reverse(num);
  for(;reversedNumber > 0;reversedNumber/=10){
        LCD_data(reversedNumber%10 + '0');
    }
    while(zero_counter > 0){
        LCD_data('0');
        zero_counter--;
    }
}
long long int reverse(long long int a){
  long long int reversed = 0;
  zero_counter = 0;
  while(a % 10 == 0){
    a/=10;
    zero_counter++;
  }
  for(;a > 0;a/=10){
      reversed*=10;
      reversed+=a%10;
    }
  return reversed;
}
```

نمایش عدد بر روی LCD

صرفا بخش هایی که جدید هستند را توضیح می‌دهیم.

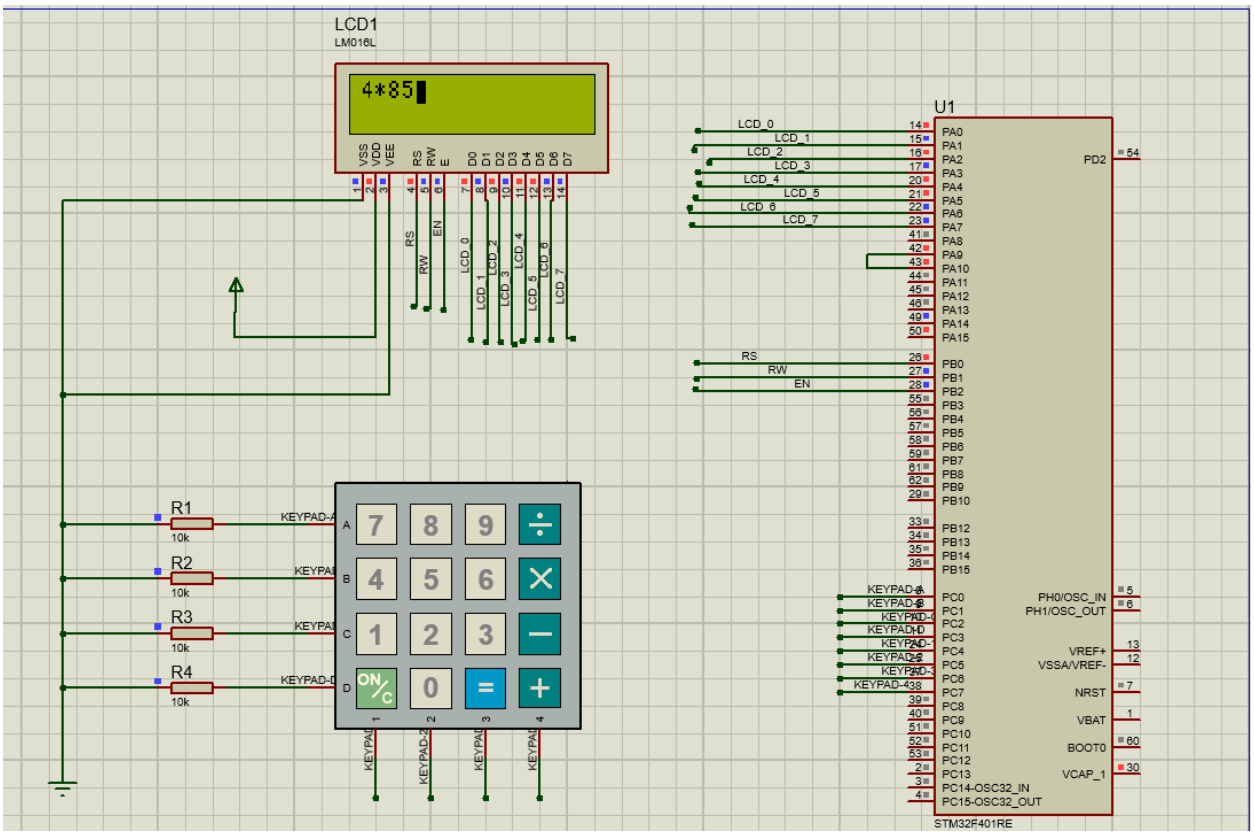
در این دستور کار باید با استفاده از UART داده ها نظیر اعداد و دستور ها را ارسال می‌کنیم و پس از دریافت آن دستور متناظر اجرا می‌شود.

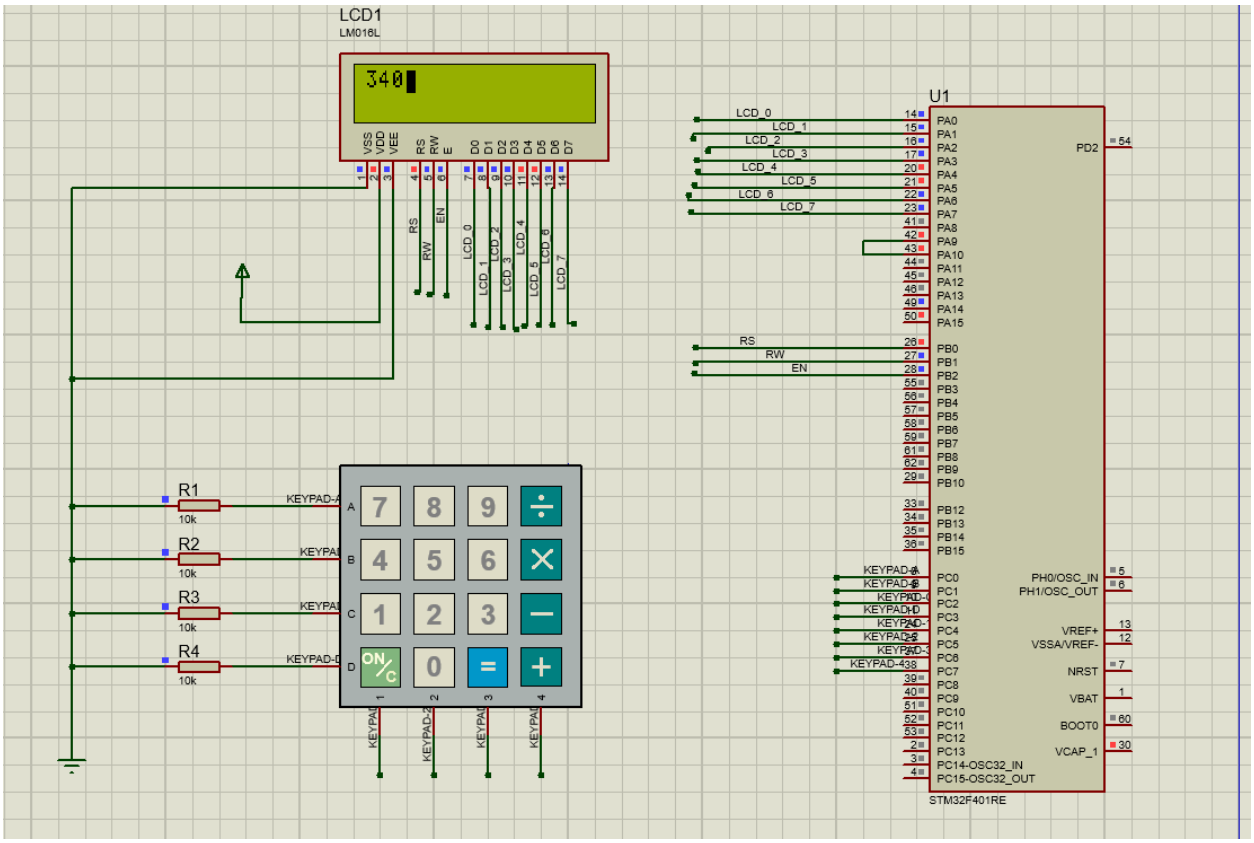
و از توابع زیر کمک گرفته ایم:

- HAL_UART_TRASMIT: این تابع، داده ها را به صورت بلاکینگ ارسال می‌کند.
- HAL_UART_Receive_IT : این تابع، به صورت غیربلاکینگ داده ها را دریافت می‌کند.
- HAL_UART_RxCpltCallback: بعد از دریافت داده ها، اینتراپتی رخ می‌دهد مه این تابع به صورت کال بک صدا میخورد.

```
void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin){
  if(rx_Data==' '){
    switch (GPIO_Pin){
      case GPIO_PIN_3:{switch(keyPadColIndex){
    //Clear Key
    case 0:{
      character = 'C';
      if(HAL_UART_Transmit(&huart1,&character,1,100) == HAL_OK){
        HAL_UART_Receive_IT(&huart1,&rx_Data,1);
      }
      opState = OPERAND1;
      operand1 = 0;
      operand2 = 0;
      break;
    }
    //0 Key
    case 1:{
      character = '0';
      if(HAL_UART_Transmit(&huart1,&character,1,100) == HAL_OK){
        HAL_UART_Receive_IT(&huart1,&rx_Data,1);
      }
      if(opState == OPERAND1){
        operand1 *= 10;
        operand1 += 0;
      }
      else if (opState==OPERAND2){
        operand2 *= 10;
        operand2 += 0;
```

نتیجه در پروتئوس:





رفرنس دستور کار:

کلاس درس و اسلاید های درسی

دیتا شیت و رفرنس منوآل