Practical Assignment № 5

Digital and Microcontroller Devices

variant number : 6

Student Name： Xu Miao, Zhou Haojie

HDU Number： 19322103, 19322233

ITMO number : 293687, 293806

Practical Task 5

1) Purpose of work:

Sending and receiving data to Simulink.

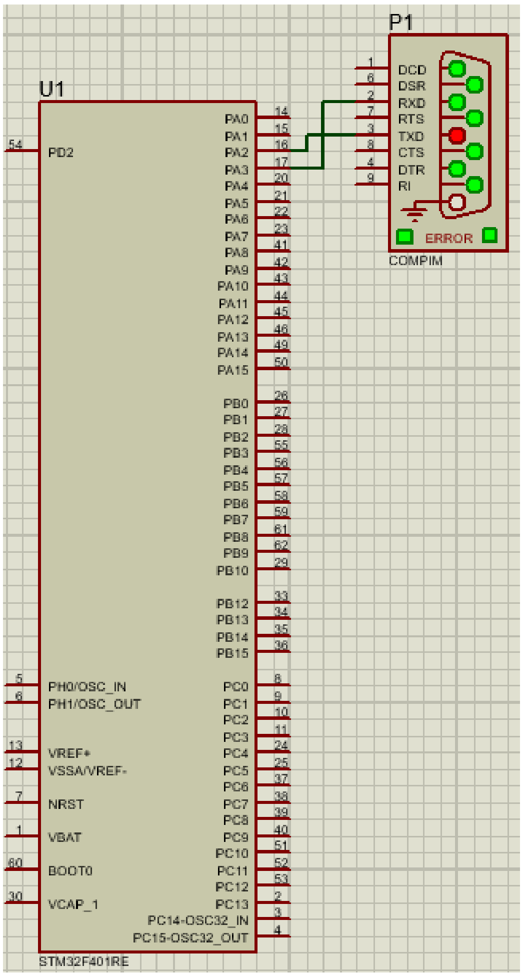
2) Variant:

Variant No.	Baudrate	Amplitude of sin
6	28800	1

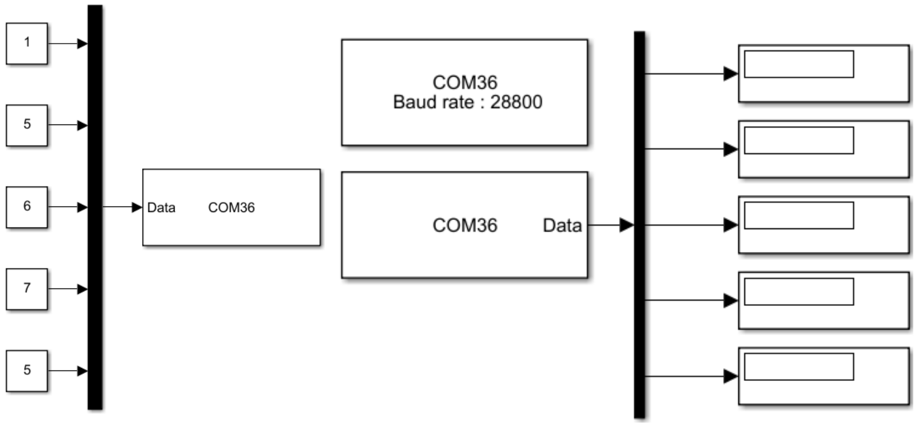
3) Part 0. following the example from the lecture.

- Tset Sending and receiving data :
  a) following the example from the lecture ;
  b) In the Simulink scheme , add a data sending unit to the controller;

- Proteus scheme

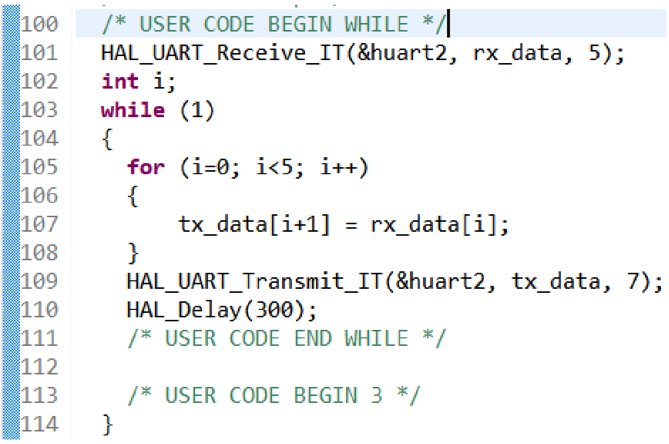


- ## MATLAB scheme



- ## Program code

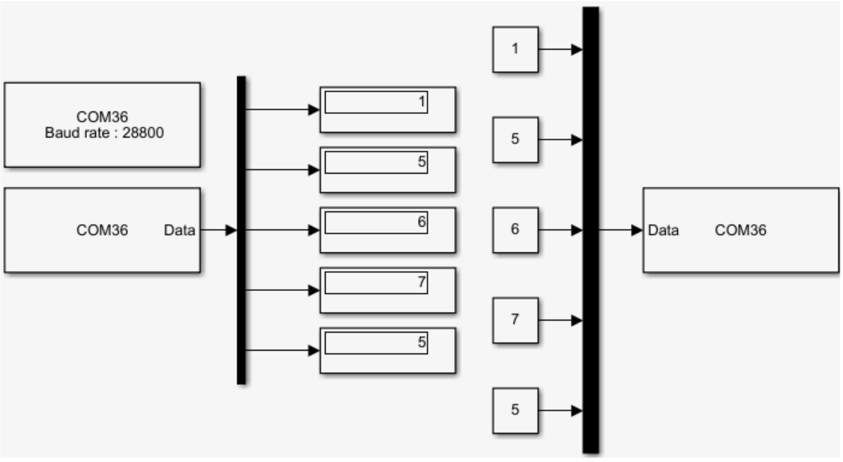
➢ main() function



```
  /* USER CODE BEGIN WHILE */
  HAL_UART_Receive_IT(&huart2, rx_data, 5);
  int i;
  while (1)
  {
    for (i=0; i<5; i++)
    {
        tx_data[i+1] = rx_data[i];
    }
    HAL_UART_Transmit_IT(&huart2, tx_data, 7);
    HAL_Delay(300);
    /* USER CODE END WHILE */

    /* USER CODE BEGIN 3 */
  }
```

- ## Simulation Result

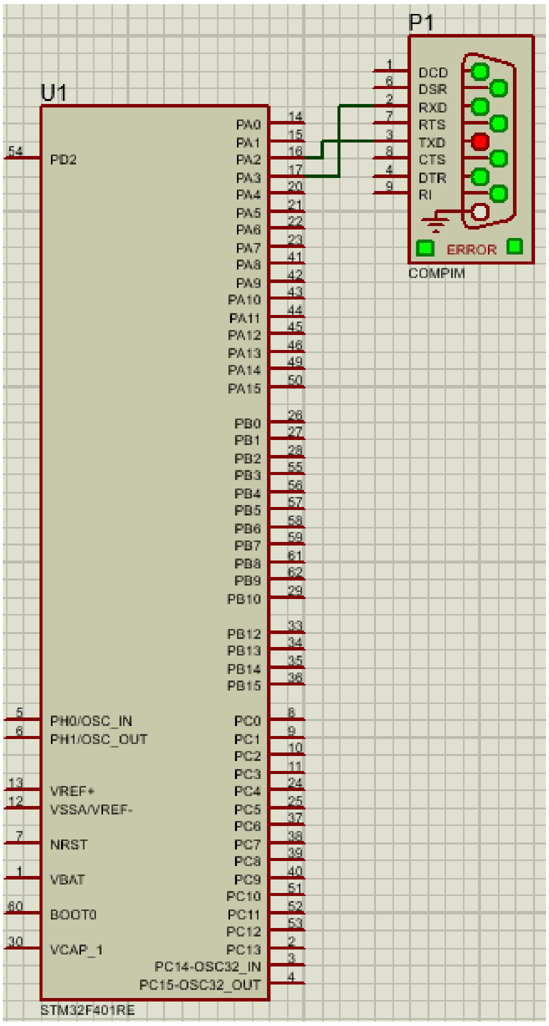


MATLAB ——> STM32 ——>MATLAB

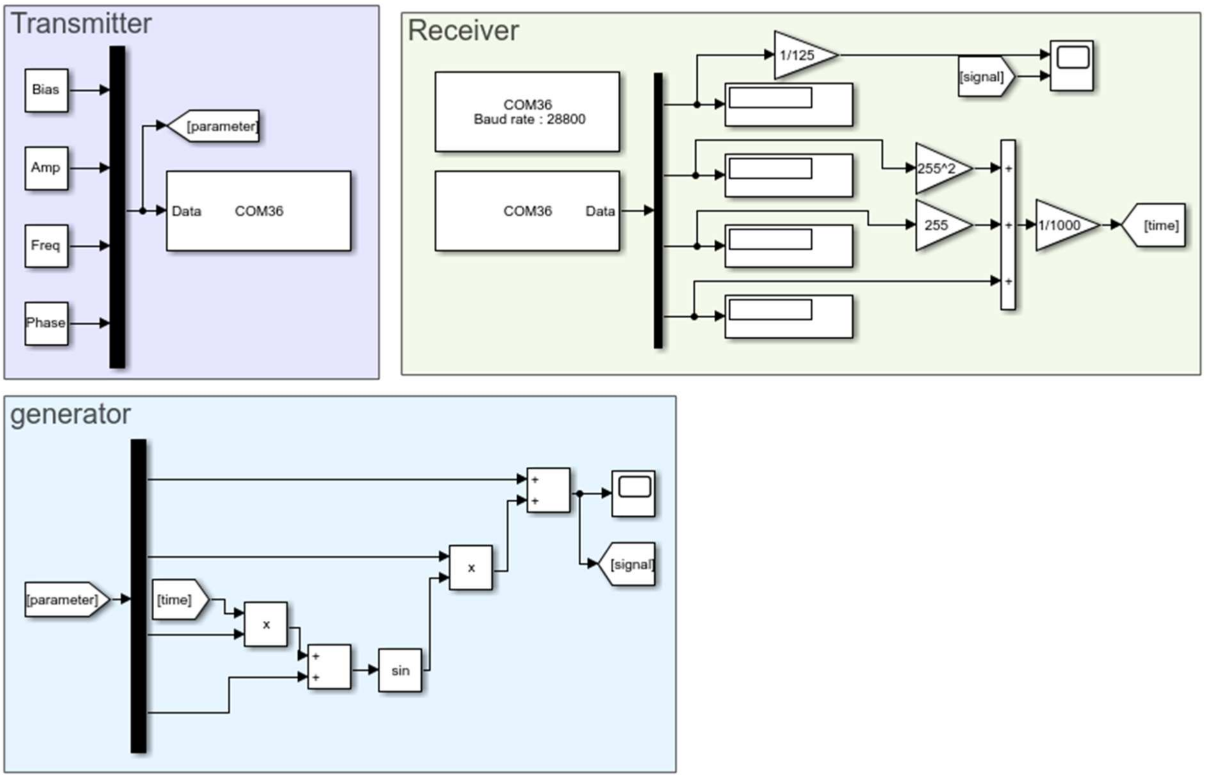
4) Part I. Development of the program using the codegenerator.

- Implement the following algorithm of the program:
  - a) In Simulink generate a packet to send function parameters to the controller;
  - b) In the controller, process the received packet and calculate the new value of the periodic function;
  - c) Generate a package to send the result of calculations to the computer;
  - d) Compare the calculation result from matlab and STM32.

- **Proteus scheme**

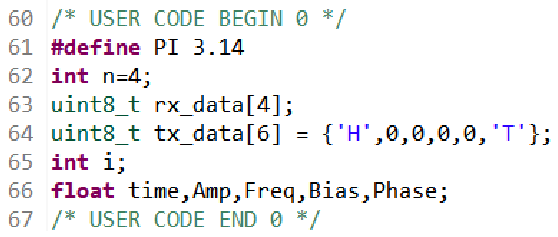


- **MATLAB scheme**



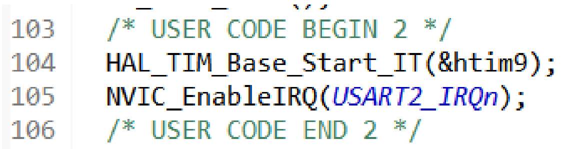
- **Program code**

➢ Initialization of variable



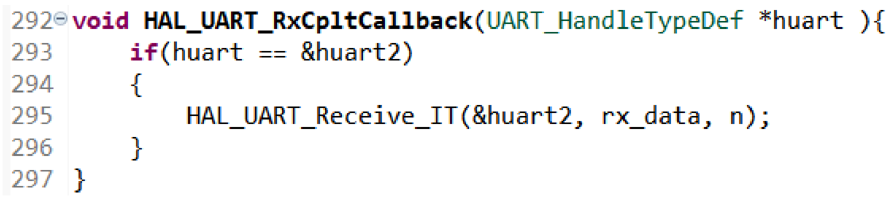
```c
/* USER CODE BEGIN 0 */
#define PI 3.14
int n=4;
uint8_t rx_data[4];
uint8_t tx_data[6] = {'H',0,0,0,0,'T'};
int i;
float time,Amp,Freq,Bias,Phase;
/* USER CODE END 0 */
```

➢ Start timer and interrupt



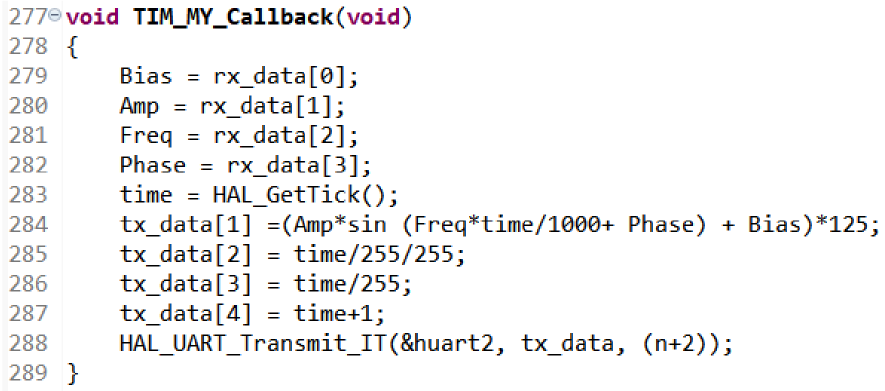
```c
  /* USER CODE BEGIN 2 */
  HAL_TIM_Base_Start_IT(&htim9);
  NVIC_EnableIRQ(USART2_IRQn);
  /* USER CODE END 2 */
```

➢ HAL_UART_RxCpltCallback() Function



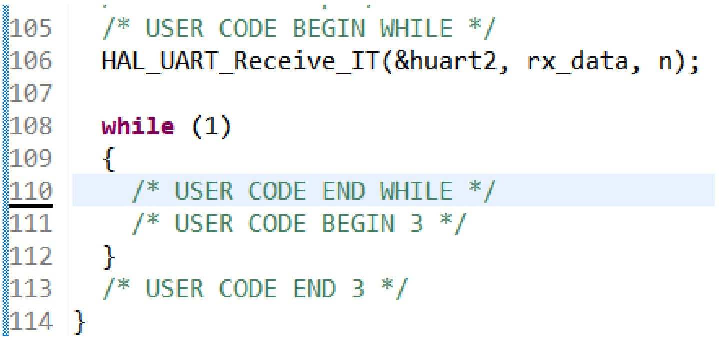
```c
void HAL_UART_RxCpltCallback(UART_HandleTypeDef *huart ){
    if(huart == &huart2)
    {
        HAL_UART_Receive_IT(&huart2, rx_data, n);
    }
}
```

➢ TIM_MY_Callback() Function



```c
void TIM_MY_Callback(void)
{
    Bias = rx_data[0];
    Amp = rx_data[1];
    Freq = rx_data[2];
    Phase = rx_data[3];
    time = HAL_GetTick();
    tx_data[1] =(Amp*sin (Freq*time/1000+ Phase) + Bias)*125;
    tx_data[2] = time/255/255;
    tx_data[3] = time/255;
    tx_data[4] = time+1;
    HAL_UART_Transmit_IT(&huart2, tx_data, (n+2));
}
```

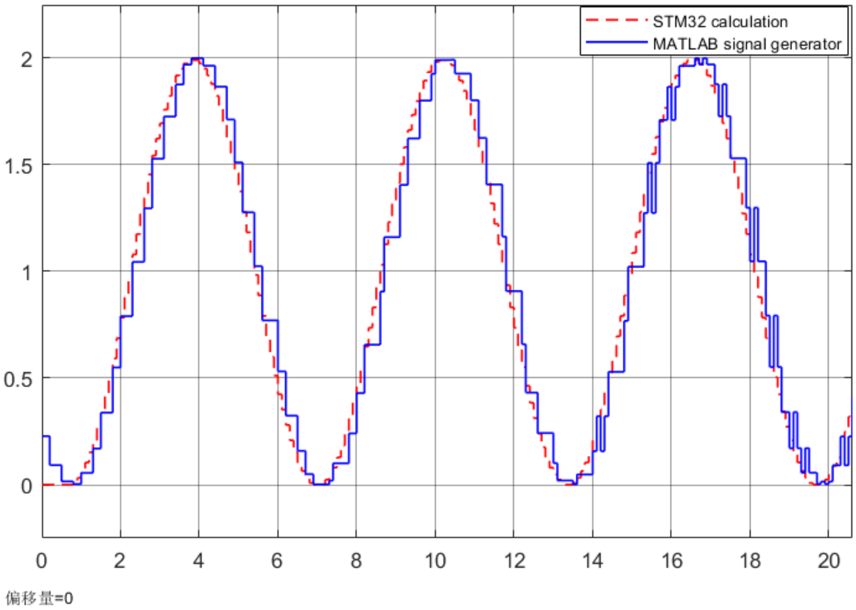
➢ main() function



```c
  /* USER CODE BEGIN WHILE */
  HAL_UART_Receive_IT(&huart2, rx_data, n);

  while (1)
  {
    /* USER CODE END WHILE */
    /* USER CODE BEGIN 3 */
  }
  /* USER CODE END 3 */
}
```

● Simulation Result

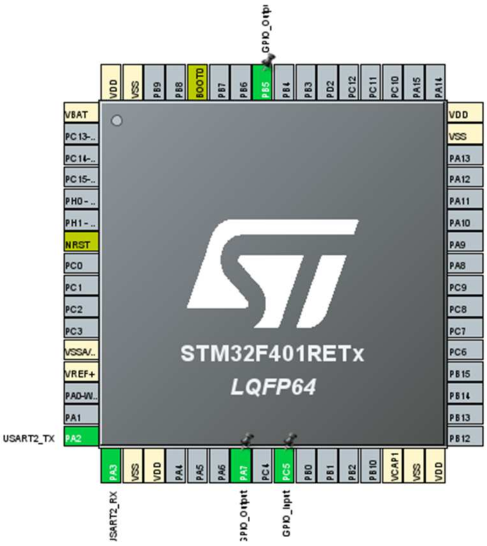


MATLAB visualization simulation result

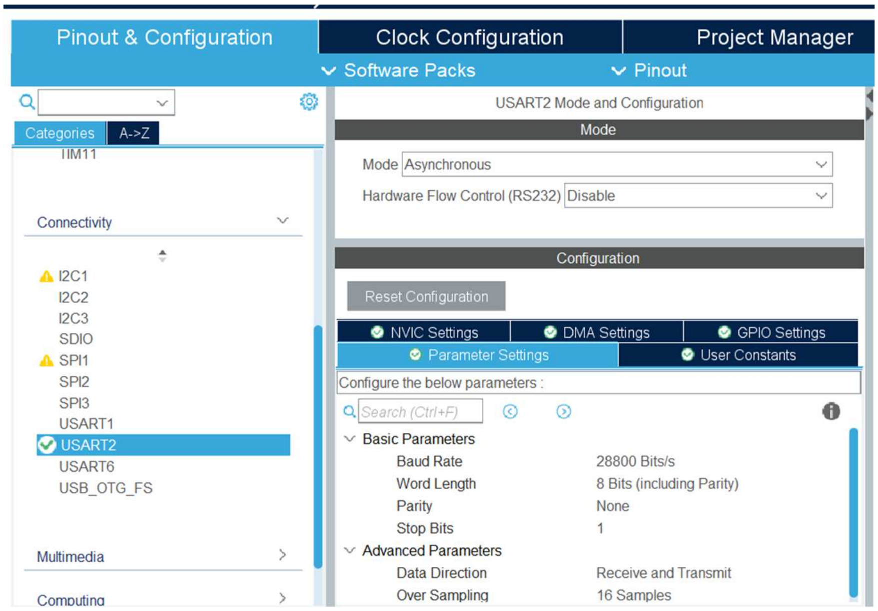
5) Appendix:

- Settings.

➢ Pins



➢ USART2



USART2 Mode and Configuration

Mode

Mode Asynchronous

Hardware Flow Control (RS232) Disable

Configuration

Configure the below parameters :

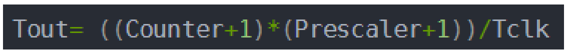
Basic Parameters

Baud Rate	28800 Bits/s
Word Length	8 Bits (including Parity)
Parity	None
Stop Bits	1

Advanced Parameters

Data Direction	Receive and Transmit
Over Sampling	16 Samples

➢ Timer



```
Tout= ((Counter+1)*(Prescaler+1))/Tclk
```

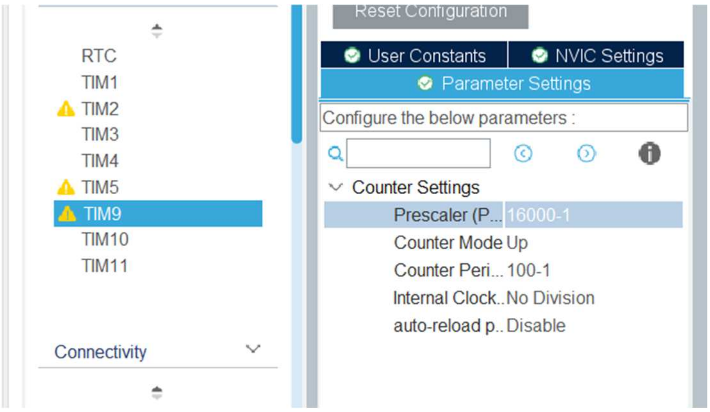
$$f_{out} = 10HZ$$

$$T_{out} = 0.1s$$

$$T_{clk} = 16MHZ$$

$$counter = 16000 - 1$$

$$Prescaler = 100 - 1$$



Configure the below parameters :

Counter Settings

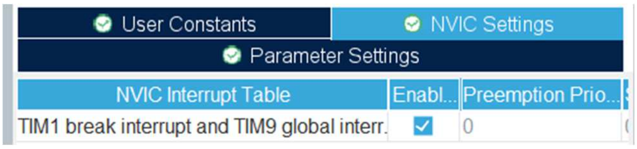
Prescaler (P... 16000-1

Counter Mode Up

Counter Peri... 100-1

Internal Clock..No Division

auto-reload p..Disable



NVIC Interrupt Table	Enabl...	Preemption Prio...
TIM1 break interrupt and TIM9 global interr.	☑	0

6) Conclusions:

conclusion:

- This experiment uses STM32CubeIDE, Matlab for signal generation, sending and, transmission.
- After this experiment, we are familiar with the signal interaction between the microcontroller STM32 and MATLAB