Practical Assignment № 6

Digital and Microcontroller Devices

variant number : 6

Student Name： Xu Miao, Zhou Haojie

HDU Number： 19322103, 19322233

ITMO number : 293687, 293806

Practical Task 6

1) Purpose of work:

Sending and receiving data to Simulink.

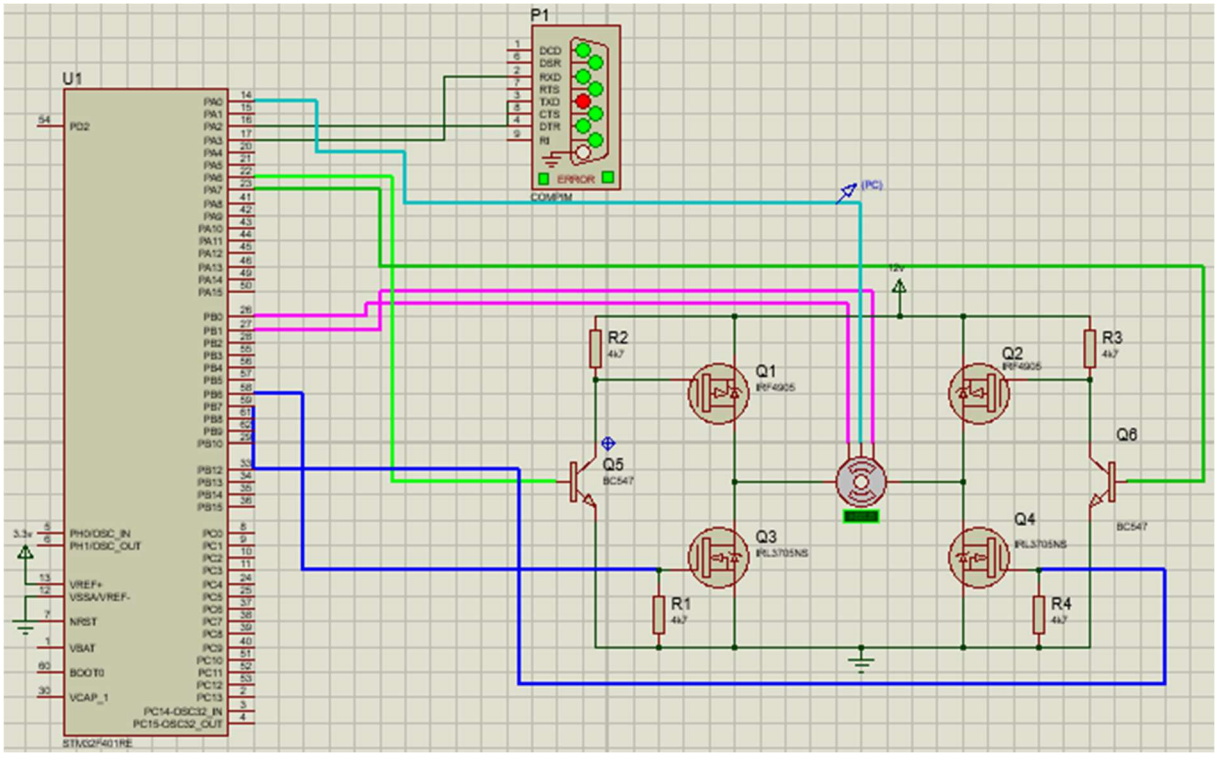
2) Variant:

Variant No.	Target angle
6	40

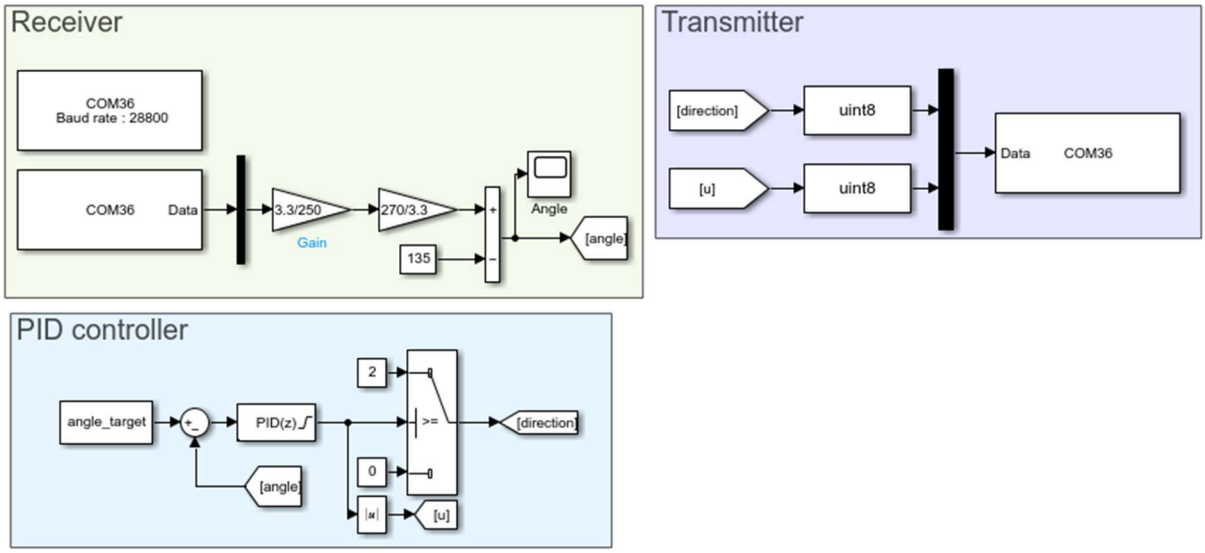
3) Follow all the steps in Part I

- Implement the following algorithm of the program:
  a) In the TIM interrupt read value of ADC, convert it to angle.
  b) Based on current position calculate new PWM signal for motor by PID.
  c) Send current position, value of control to the Matlab. to the computer;
  d) Try again. algorithm first.

- Proteus scheme



- **MATLAB scheme**



- **Program code**

➢ **Initialization of variable**

```
/* USER CODE BEGIN 0 */
uint8_t rx_data[2];
uint8_t tx_data[3] = {'H',0,'T'};
uint8_t direction,u;
float counter;
/* USER CODE END 0 */
```

➢ **Start timer and interrupt**

```
  /* USER CODE BEGIN 2 */
  HAL_TIM_Base_Start_IT(&htim9);
  HAL_ADC_Start(&hadc1);
  HAL_TIM_PWM_Start(&htim4, TIM_CHANNEL_1);
  HAL_TIM_PWM_Start(&htim4, TIM_CHANNEL_2);
  /* USER CODE END 2 */
```

➢ **HAL_UART_RxCpltCallback() Function**

```
void HAL_UART_RxCpltCallback(UART_HandleTypeDef *huart ){
    if(huart == &huart2)
    {
        HAL_UART_Receive_IT(&huart2, rx_data, 2);
    }
}
```

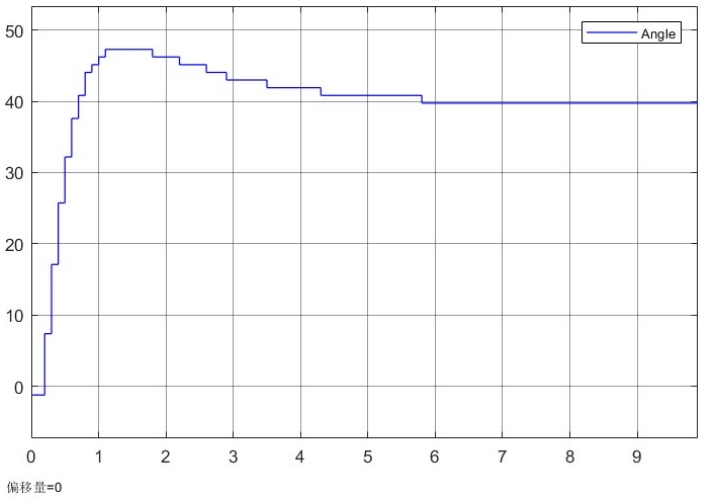
➢ **TIM_MY_Callback() Function**

```c
void TIM_MY_Callback(void)
{
    HAL_GPIO_WritePin(GPIOB, GPIO_PIN_0, 0);
    HAL_GPIO_WritePin(GPIOB, GPIO_PIN_1, 1);

    direction = rx_data[0];
    u = rx_data[1];

    if(direction > 1)
    {
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_6, 1);
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_7, 0);
        __HAL_TIM_SET_COMPARE(&htim4, TIM_CHANNEL_1,0);
        __HAL_TIM_SET_COMPARE(&htim4, TIM_CHANNEL_2,u);
    }else {
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_6, 0);
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_7, 1);
        __HAL_TIM_SET_COMPARE(&htim4, TIM_CHANNEL_1,u);
        __HAL_TIM_SET_COMPARE(&htim4, TIM_CHANNEL_2,0);
    }
    counter = HAL_ADC_GetValue(&hadc1)*250.00/4096.00;
    tx_data[1] = counter;
    HAL_UART_Transmit_IT(&huart2, tx_data, 3);
}
```

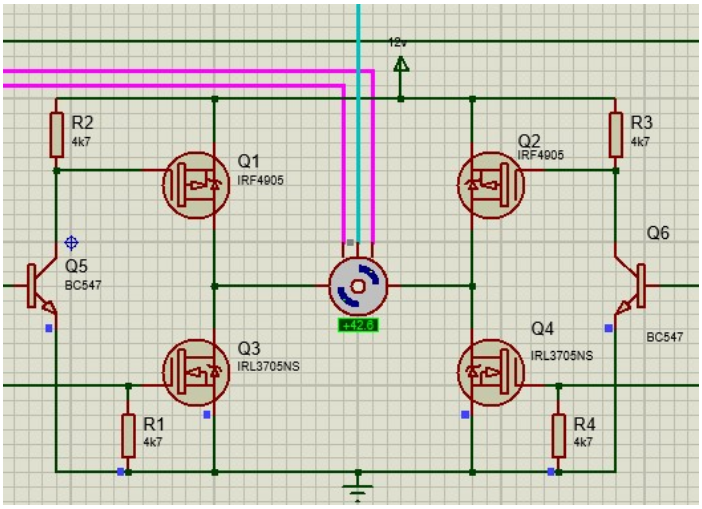
➢ main() function

```c
  HAL_UART_Receive_IT(&huart2, rx_data, 2);

  while (1)
  {
    /* USER CODE END WHILE */

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

● Simulation Result



MATLAB visualization angle

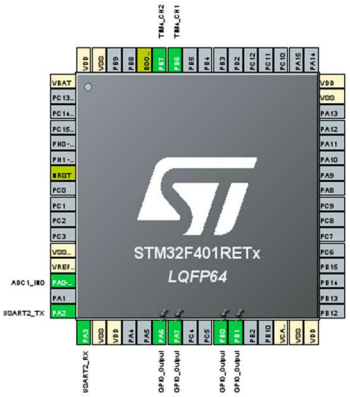


motor final angle

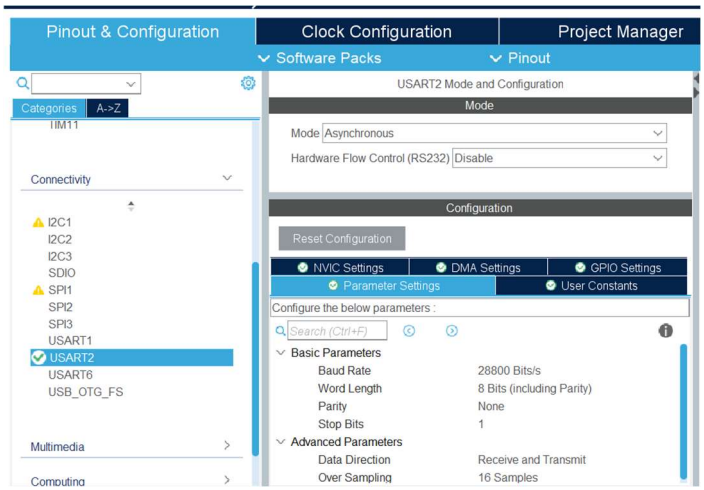
4) Appendix:

- **Settings.**

➢ **Pins**



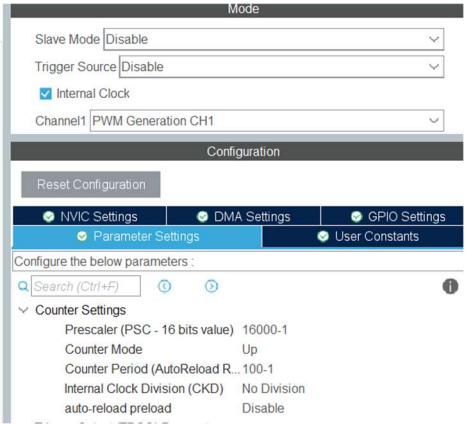
➢ **USART2**



➢ **Timer**

$$f_{out} = 10HZ$$
$$T_{out} = 0.1s$$
$$T_{clk} = 16MHZ$$
$$counter = 16000 - 1$$
$$Prescaler = 100 - 1$$

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



5) Conclusions:

conclusion:

- In this experiment, STM32CubeIDE, Matlab, H-bridge circuit and PID controller are used to achieve accurate control of the Angle position of the servo motor. From the MATLAB experimental results, the effect is very good, but due to the error of data transmission between MATLAB and STM32, the servo motor Angle has a small deviation. In actual production, we can achieve better control by constantly adjusting parameters.
- After this experiment, we are familiar with the information interaction between STM32 and MATLAB, and use it and the H-bridge circuit to complete the PID control.