PROJECT 3
Presentation

Nattarach Saekaow 6210550380

Tatpong Thunyaudom 6210552617

Kawinpop Chumchuensuk 6210552650

Microprocessor
01205311

Project Number 3

1. The baud rate of communication is 19200 bps.

2. When system start up, ARM send the following sentence to PC and the putty will display as follow

   "Hello World \r\n"

   "Nice to meet you \r\n"

   "I can chat with you > \r\n"

3. You can design the question and answer sentences by yourself.

4. Each group must have at least 5 questions.

5. Questions must be a complete sentence with "\r\n" at the end of question.

Code

```
1  #include <stddef.h>
2  #include "stm32f10x.h"
3  #include "stm32f10x_conf.h"
```

Include 1 more library "stm32f10x_conf.h": configuration header file with all peripherals defined by default.

Code

```
 5⊖ void USART1_sendC(unsigned char c){
 6    while(USART_GetFlagStatus(USART1,USART_FLAG_TXE)==RESET);
 7    USART_SendData(USART1,(unsigned char) c);
 8  }
 9
10⊖ char USART1_getC(){
11    while(USART_GetFlagStatus(USART1,USART_FLAG_RXNE)==RESET);
12    return(USART_ReceiveData(USART1));
13  }
```

Introduced 2 functions

USART1_sendC() : This function will received the input as unsigned char and checks whether the USART_FLAG_TXE which is Transmit data register empty flag is ready or not? then transmits single data through the USART1 peripheral byte by byte.

USART1_getC(): This function will checks whether the USART_FLAG_RXNE which Receive data register not empty flag is set or not, then returns the most recent received data by the USARTx peripheral

Code

```
15  int main(void){
16      RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOA|RCC_APB2Periph_AFIO|RCC_APB2Periph_USART1,ENABLE);
```

line 16 : Enables or disables the High Speed APB (APB2) peripheral clock for GPIOA,
AFIO, USART1.

The Default of SPL in Atollic true Studio RCC initiates with 72 Mhz. If you want your
system to operate with 8Mhz you can use:

RCC_DeInit : Resets the RCC clock configuration to the default reset state. (72 Mhz ->
8 Mhz).

Code

line 18-22: GPIO Init structure definition, and specifies GPIO pin ,speed, and selecting mode, which we specifies GPIO pin 9 with speed of 50 Mhz and operating mode as Push-Pull Output Alternate-Function for Tx.

line 24-27: we specifies GPIO pin 10 with speed of 50 Mhz and operating mode as Input Floating for Rx.

```
18   GPIO_InitTypeDef GPIO_InitStruct;
19   GPIO_InitStruct.GPIO_Pin = GPIO_Pin_9;
20   GPIO_InitStruct.GPIO_Speed = GPIO_Speed_50MHz;
21   GPIO_InitStruct.GPIO_Mode = GPIO_Mode_AF_PP;
22   GPIO_Init(GPIOA, &GPIO_InitStruct);
23   //Rx
24   GPIO_InitStruct.GPIO_Pin = GPIO_Pin_10;
25   GPIO_InitStruct.GPIO_Speed = GPIO_Speed_50MHz;
26   GPIO_InitStruct.GPIO_Mode = GPIO_Mode_IN_FLOATING;
27   GPIO_Init(GPIOA, &GPIO_InitStruct);
```

Code

```
29    USART_InitTypeDef usart1_init_struct;
30    usart1_init_struct.USART_BaudRate = 19200;
31    usart1_init_struct.USART_WordLength = USART_WordLength_8b;
32    usart1_init_struct.USART_StopBits = USART_StopBits_1;
33    usart1_init_struct.USART_Parity = USART_Parity_No;
34    usart1_init_struct.USART_Mode = USART_Mode_Rx | USART_Mode_Tx;
35    usart1_init_struct.USART_HardwareFlowControl = USART_HardwareFlowControl_None;
36    USART_Init(USART1, &usart1_init_struct);
37    USART_Cmd(USART1, ENABLE);
```

line 36: Initializes the USARTx peripheral according to the specified parameters in the USART_InitStruct.

line 37: Enable the specified USART peripheral.

Line 29-35: USART Init Structure definition, and configures the USART communication baud rate, specifies the number of data bits transmitted or received in a frame, specifies the number of stop bits transmitted, specifies the parity mode, specifies whether the Receive or Transmit mode is enabled or disabled, and specifies whether the hardware flow control mode is enabled or disabled.

Code

```
39  char question, sData, input[50];
40  int i = 0,j;
41  unsigned char *msg,*ans;
42
43  char question_1[50] =  "Who are you?\n";
44  char question_2[50] =  "Am I handsome?\n";
45  char question_3[50] =  "Do you like music?\n";
46  char question_4[50] =  "Why you choose me?\n";
47  char question_5[50] =  "Where are you?\n";
48
49  while(1){
50   msg = (unsigned char *) "\n\rHello World \\r\\n\n\rNice to meet you \\r\\n\n\rI can chat with you > \\r\\n\n\r\0";
51   i = 0;
52   while(msg[i] != '\0'){
53    USART1_sendC(msg[i]);
54    i++;
55   }
56   i = 0;
```

line 43-47: declaring array variable type char and initialization create string consisting of the question sentence.

line 49-50: Uses loop of USART1_sendC(msg[i]) to display start up sentences Hello World\n\rNice to meet you\n\r I can chat with you\n\r byte by byte that will terminate with a null character.

Code

```
58  while(1){;
59    sData = USART1_getC();
60    input[i] = sData;
61    USART1_sendC(input[i]); //echo
62    i++;
63
64    if(sData == '\n' || sData == '\r'){
65      USART1_sendC('\n');
66      USART1_sendC('\r');
67      USART1_sendC('>');
68
69      if(input[0] == question_1[0] && input[1] == question_1[1] && input[2] == question_1[2]){
70        question = '1';
71      }
72      else if(input[0] == question_2[0] && input[1] == question_2[1] && input[2] == question_2[2]){
73        question = '2';
74      }
75      else if(input[0] == question_3[0] && input[1] == question_3[1] && input[2] == question_3[2]){
76        question = '3';
77      }
78      else if(input[0] == question_4[0] && input[1] == question_4[1] && input[2] == question_4[2]){
79        question = '4';
80      }
81      else if(input[0] == question_5[0] && input[1] == question_5[1] && input[2] == question_5[2]){
82        question = '5';
83      }
84      else{
85        question = '0';
86      }
```
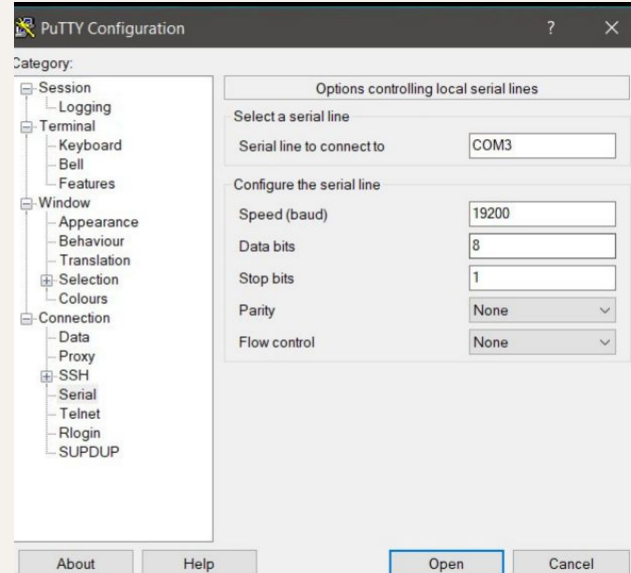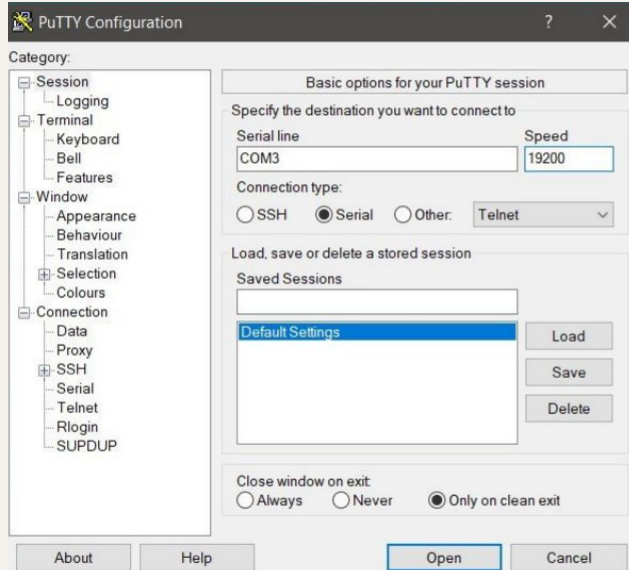
line 58-86: Create a forever loop that read the serial data byte by byte and then assign the input data to array of char named input. These lines will compare the first three characters in input array to question_x array if they are the same it will show the answer, otherwise it will display I do not understand the question.\n\r
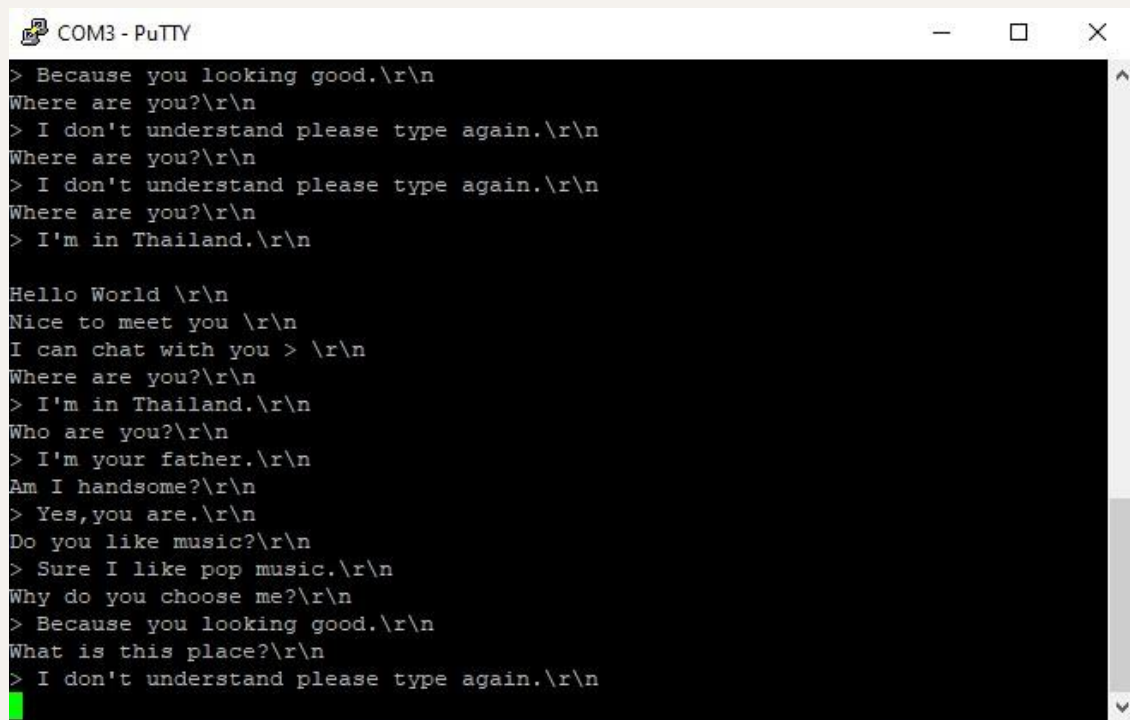
Code

```
89     switch(question){
90      case '1':
91       ans = (unsigned char *) " I'm your father.\\r\\n\n\r\0";
92       j = 0;
93       while(ans[j] != '\0'){
94        USART1_sendC(ans[j]);
95        j++;
96       }
97       break;
98      case '2':
99       ans = (unsigned char *) " Yes,you are.\\r\\n\n\r\0";
100      j = 0;
101      while(ans[j] != '\0'){
102       USART1_sendC(ans[j]);
103       j++;
104      }
105      break;
106     case '3':
107      ans = (unsigned char *) " Sure I like pop music.\\r\\n\n\r\0";
108      j = 0;
109      while(ans[j] != '\0'){
110       USART1_sendC(ans[j]);
111       j++;
112      }
113      break;
114     case '4':
115      ans = (unsigned char *) " Because you looking good.\\r\\n\n\r\0";
116      j = 0;
117      while(ans[j] != '\0'){
118       USART1_sendC(ans[j]);
119       j++;
120      }
121      break;
122     case '5':
123      ans = (unsigned char *) " I'm in Thailand.\\r\\n\n\r\0";
124      j = 0;
125      while(ans[j] != '\0'){
126       USART1_sendC(ans[j]);
127       j++;
128      }
129      break;
```

```
129      break;
130     default:
131      ans = (unsigned char *) " I don't understand please type again.\\r\\n\n\r\0";
132      j = 0;
133      while(ans[j] != '\0'){
134       USART1_sendC(ans[j]);
135       j++;
136      }
137     }
138     i = 0;
139    }
140   }
141  }
142 }
143
```

PuTTY Configuration