

PROJECT 3

MICRO

PROCESSOR

01205311

CHATBOT

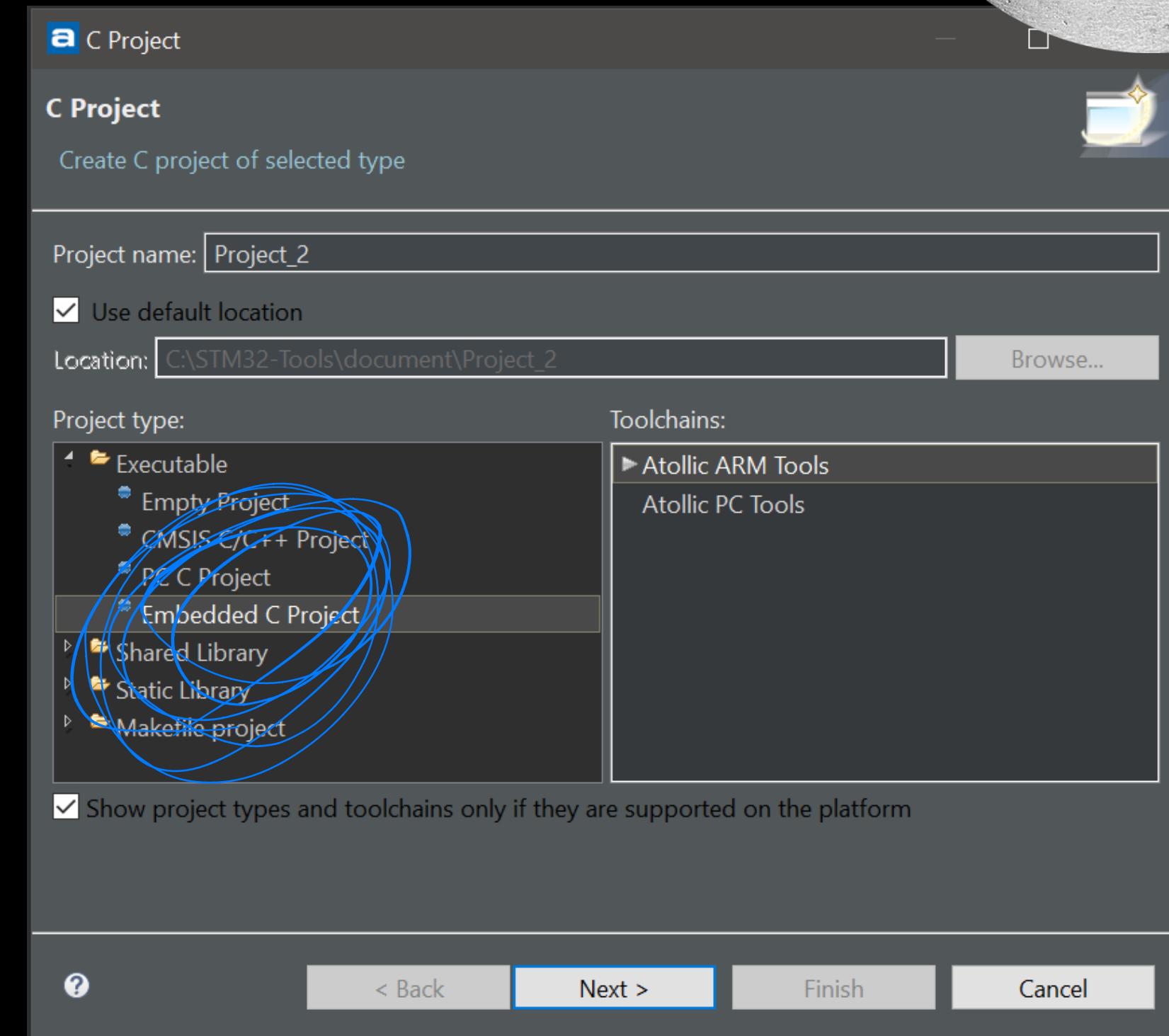
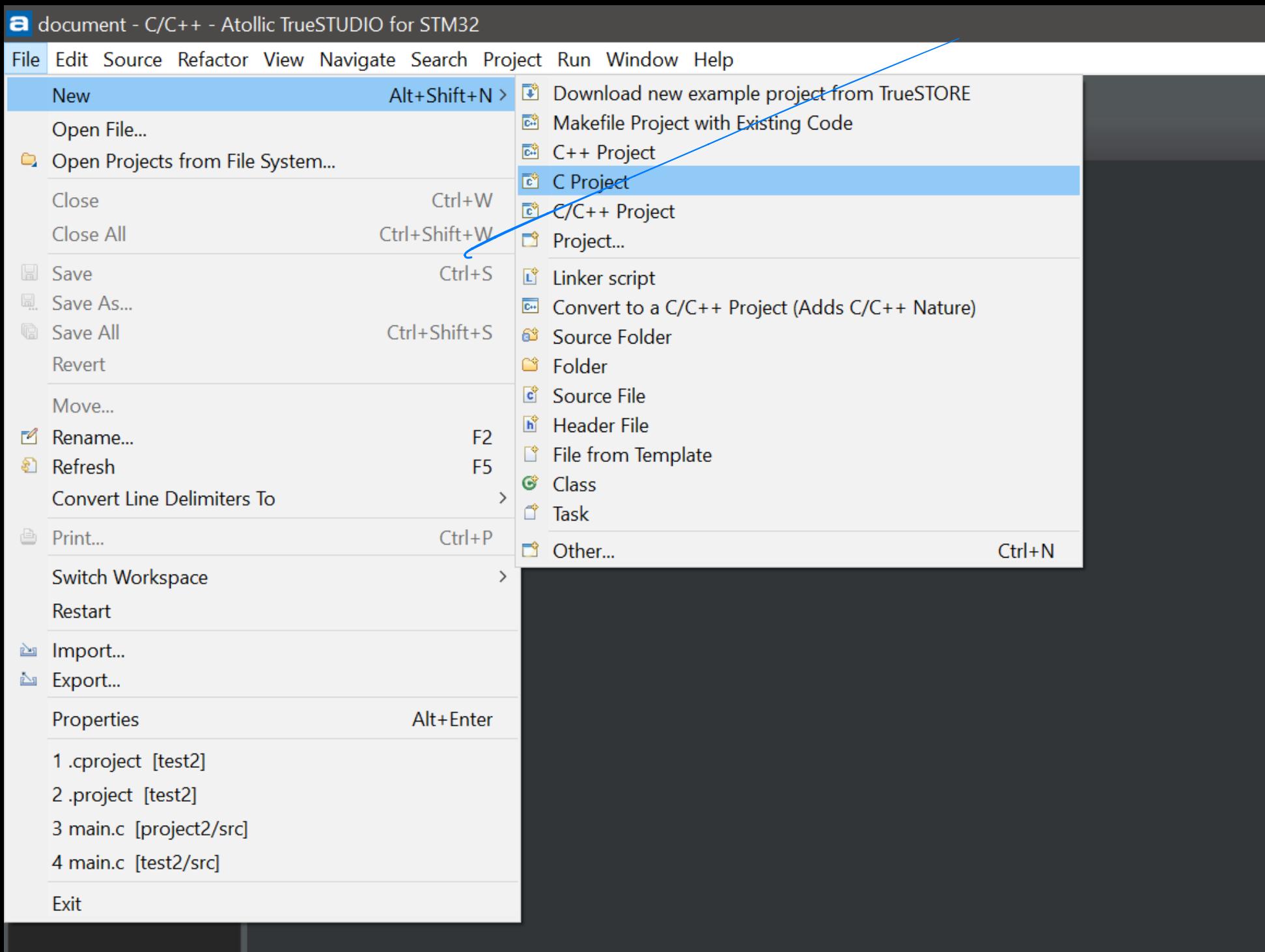


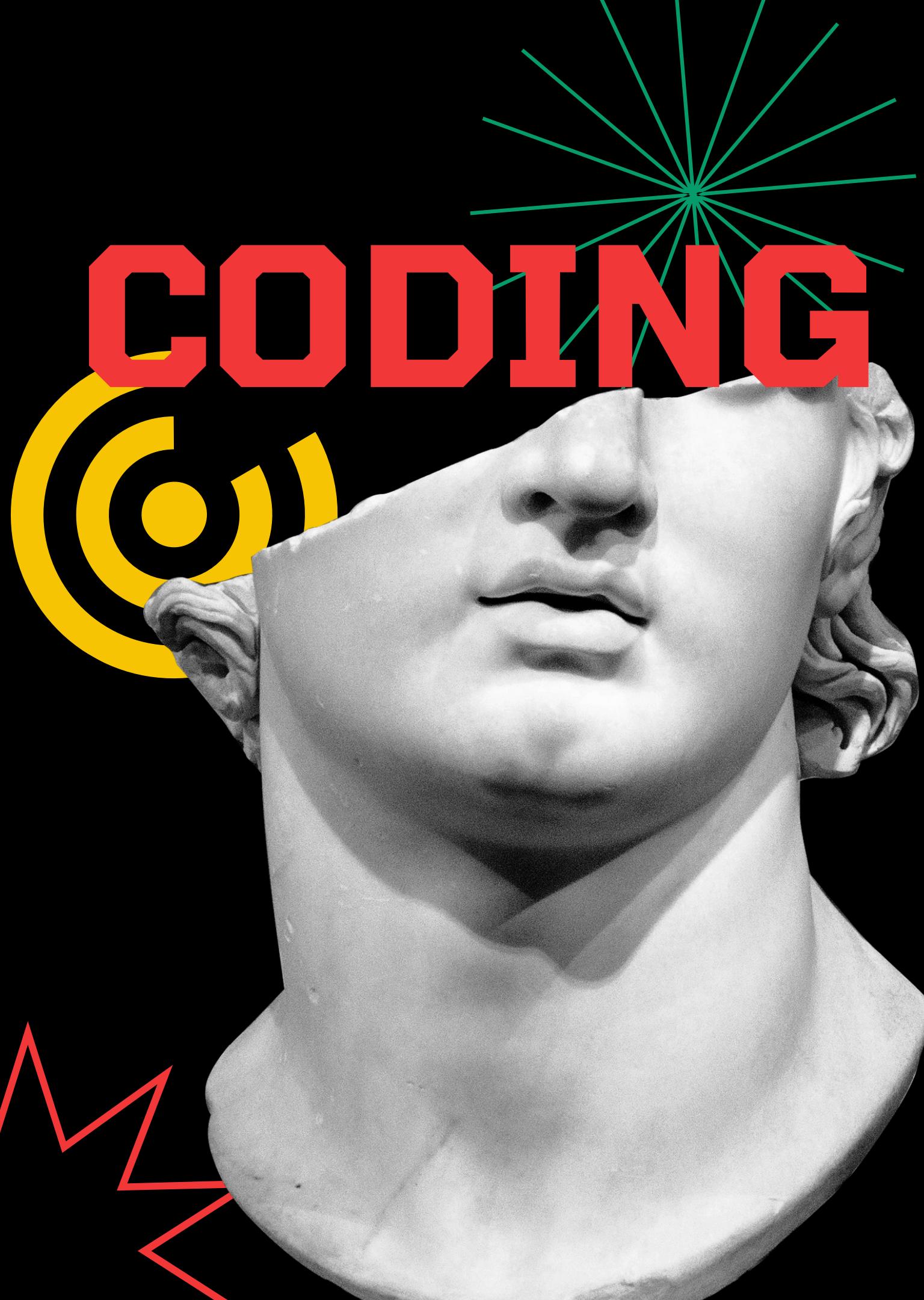
Project Instruction

This Project, we will connect ARM board to Computer via putty program.

1. The baud rate of the communication is 19200 bps.
2. When system start up ARM send the following sentence to PC and the Putty will display as follow
 - i. "Hello World\r\n"
 - ii. "nice to meet you\r\n"
 - iii. "I can chat with you\r\n"
3. You can design the question and answer sentences by yourself
4. Each group must has at least 5 questions
5. Questions must be a complete sentence with '\n' at the end of question such as
Q:"How are you today \r\n" A:"I fine thank you \r\n"

How to use Atolloc True Studio





Step1 : include libraries

```
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.

Step 2 : provide clock to each components

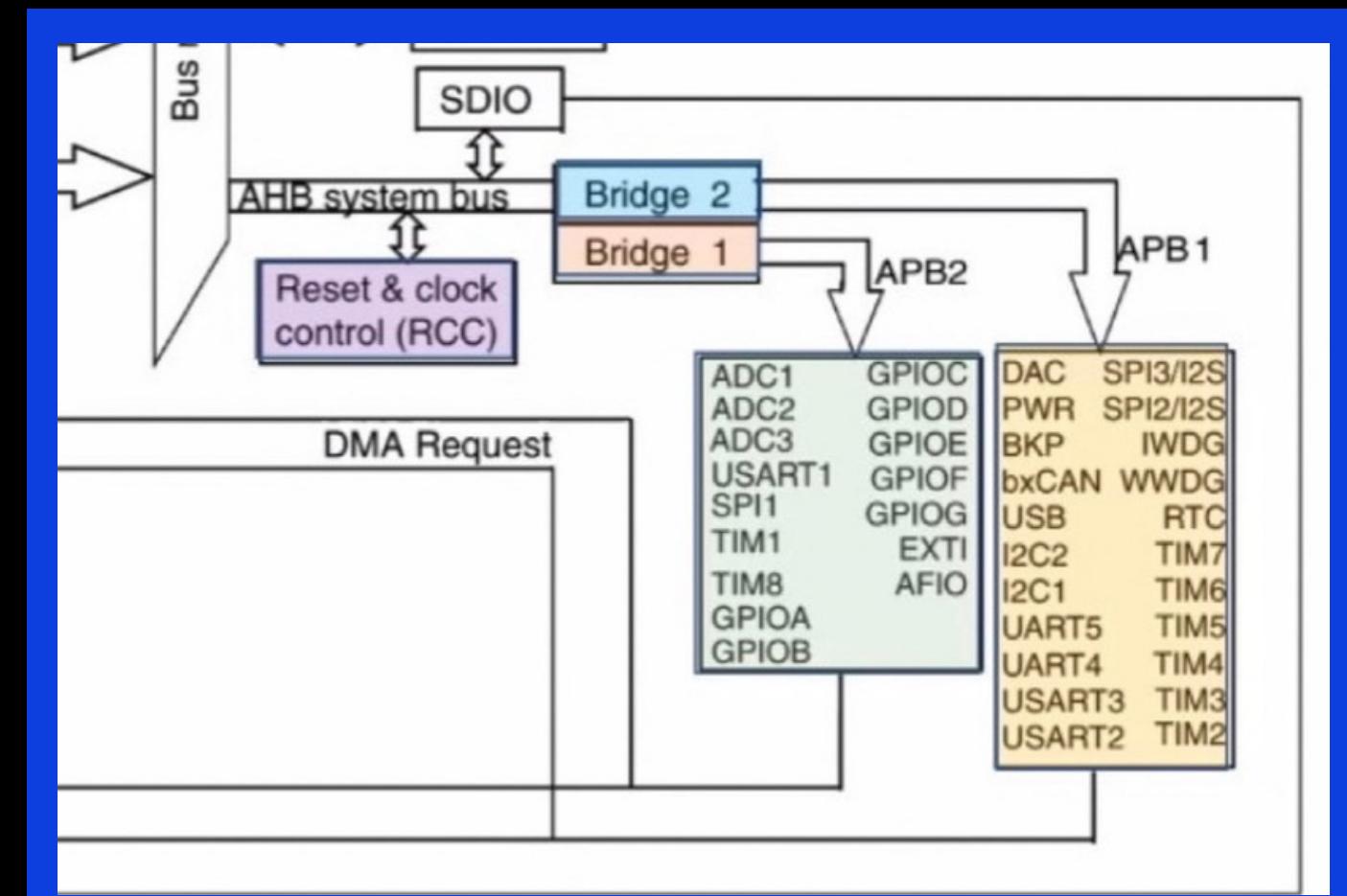
```
15 int main(void){  
16     RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOA|RCC_APB2Periph_AFIO|RCC_APB2Periph_USART1,ENABLE);
```

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:

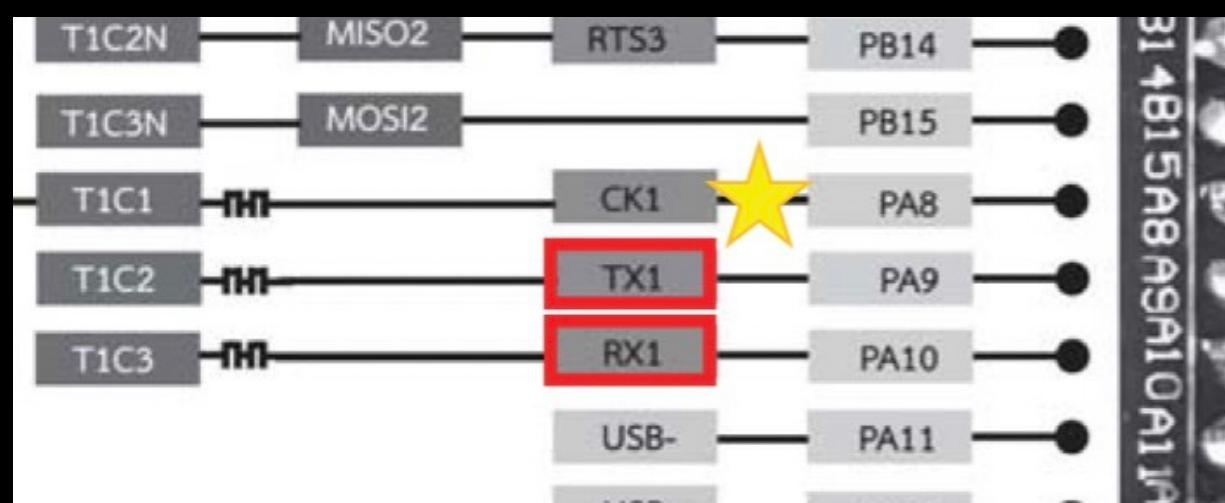
RCC_DelInit :Resets the RCC clock configuration to the default reset state.(72 Mhz -> 8 Mhz)



Step 3 : Define variable type GPIO

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.



```
17 //Tx
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);
```

Step 4 : Define variable type USART

```
28 //uart
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); 29 30
```

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. 31 32 33 34 35

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

line 37: Enable the specified USART peripheral.

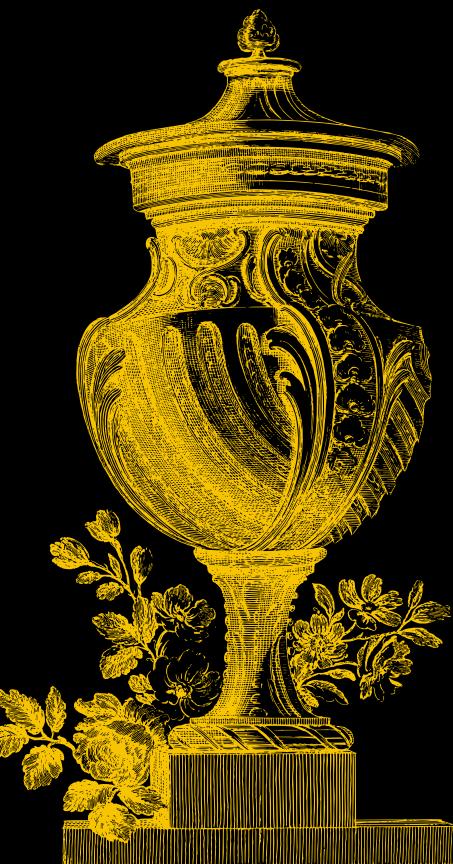
Step 4 :Putty display the start up sentences.

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.

```
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 }
```



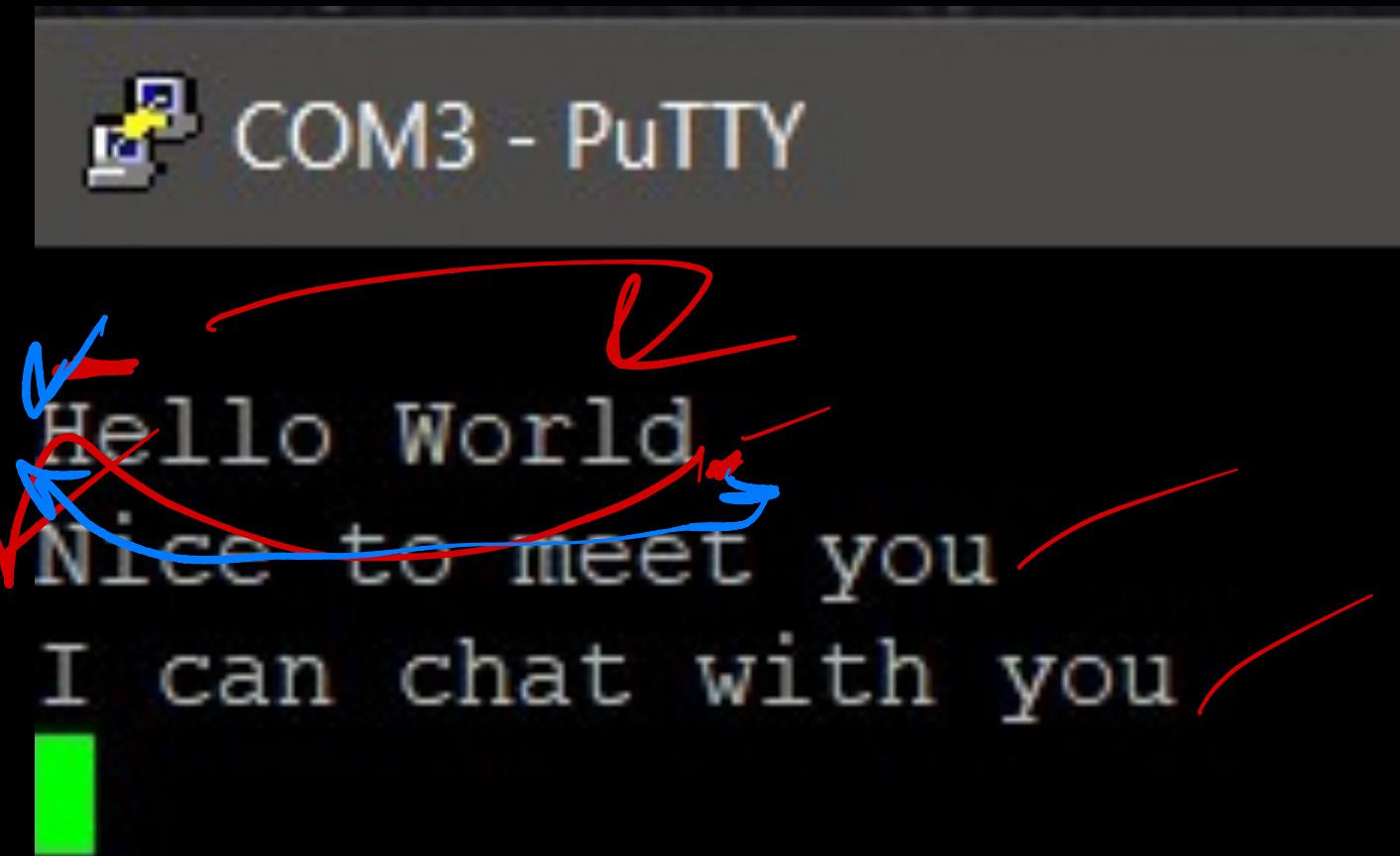
line 43-47: declaing array variable type char and initialization create string consisting of the question sentence.

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.

```
39 char question, sData, input[50];
40 int i = 0, j;
41 unsigned char *msg, *ans;
42
43 char question_1[50] = "Hello\n";
44 char question_2[50] = "Nice to meet you\n";
45 char question_3[50] = "What is your hobby\n";
46 char question_4[50] = "Tell me something\n";
47 char question_5[50] = "Bye\n";
48
49 while(1){
50     msg = (unsigned char *) "\n\rHello World\n\rNice to meet you\n\rI can chat with you\n\r\0";
51     i = 0;
52     while(msg[i] != '\0'){
53         USART1_sendC(msg[i]);
54         i++;
55     }
56     i = 0;
57 }
```

Handwritten annotations on the code:

- A blue circle highlights the five question arrays (question_1 to question_5).
- A red circle highlights the message string on line 50.
- Red arrows point from the question arrays to the string: "Hello" points to "Hello", "Nice to meet you" points to "Nice to meet you", "What is your hobby" points to "What is your hobby", "Tell me something" points to "Tell me something", and "Bye" points to "Bye".
- A red box on line 50 contains the text "I love cat" with a note "20 → 17 no space".
- A red box on line 50 contains the text "kuy" with a note "no space".
- A red box on line 50 contains the text "Hello World" with a note "Hello World".
- A red box on line 50 contains the text "Nice to meet you" with a note "Nice to meet you".
- A red box on line 50 contains the text "I can chat with you" with a note "I can chat with you".
- A red box on line 50 contains the text "\r\0" with a note "\r\0".



Questions and answers

```
59
60     sData = USART1_getC();
61     input[i] = sData;
62     USART1_sendC(input[i]); //echo
63     i++;
64
65     if(sData == '\n' || sData == '\r'){
66         USART1_sendC('\n');
67         USART1_sendC('\r');
68         USART1_sendC(':');
69
70         if(input[0] == question_1[0] && input[1] == question_1[1] && input[2] == question_1[2]){
71             ans = (unsigned char *) " Hello, Good to see you.\n\r\0";
72         }
73         else if(input[0] == question_2[0] && input[1] == question_2[1] && input[2] == question_2[2]){
74             ans = (unsigned char *) " Nice to meet you too.\n\r\0";
75         }
76         else if(input[0] == question_3[0] && input[1] == question_3[1] && input[2] == question_3[2]){
77             ans = (unsigned char *) " My hobby is chatting online.\n\r\0";
78         }
79         else if(input[0] == question_4[0] && input[1] == question_4[1] && input[2] == question_4[2]){
80             ans = (unsigned char *) " Terminator is a bad robot.\n\r\0";
81         }
82         else if(input[0] == question_5[0] && input[1] == question_5[1] && input[2] == question_5[2]){
83             ans = (unsigned char *) " Ok, see you later.\n\r\0";
84         }
85         else{
86             ans = (unsigned char *) " I do not understand the question.\n\r\0";
87         }
88         j =0;
89         while(ans[j] != '\0'){
90             USART1_sendC(ans[j]);
91             j++;
92         }
93         i =0;
94     }
95 }
96 }
```

pu**y
key
→ I do not understand

class

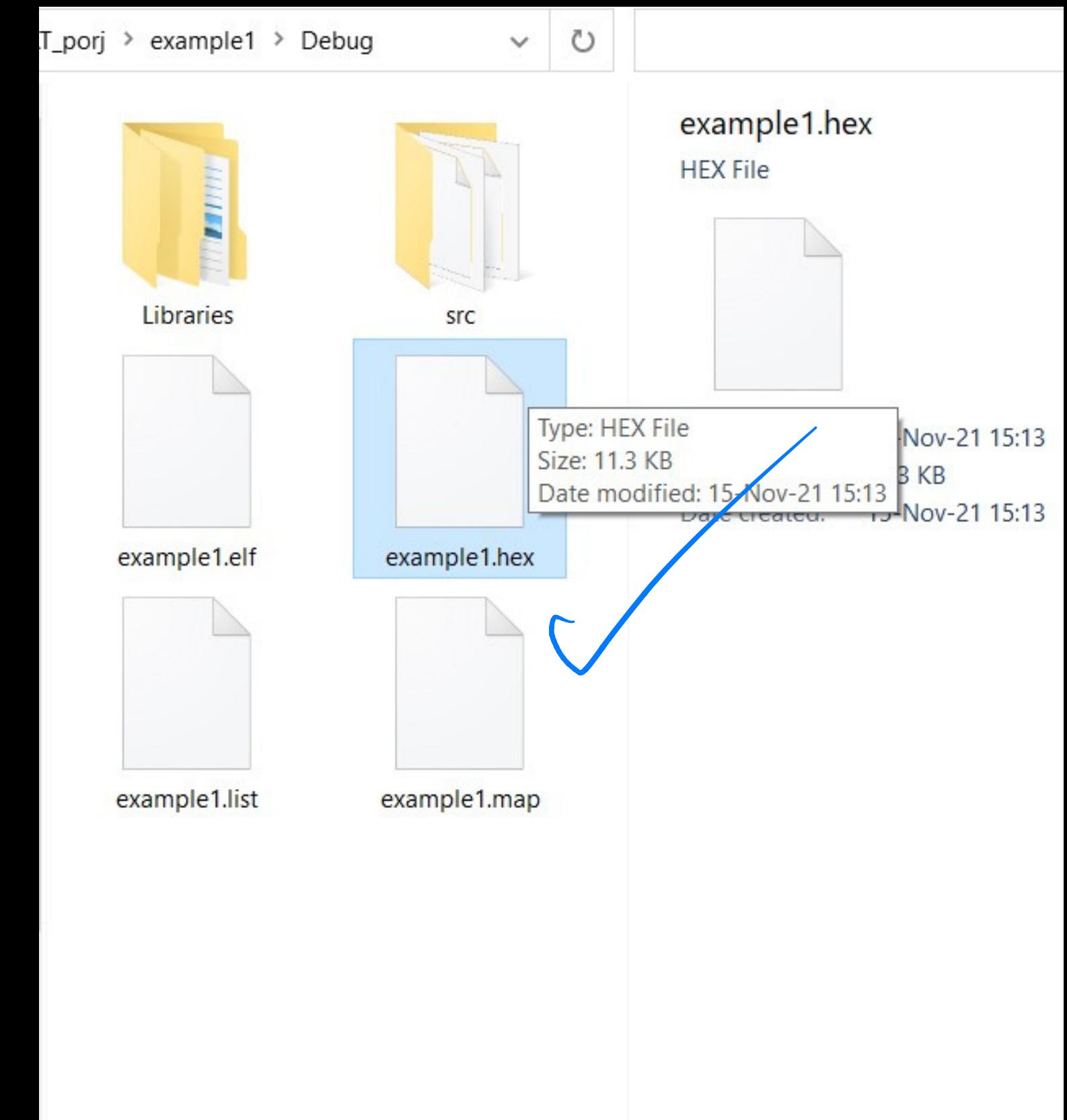
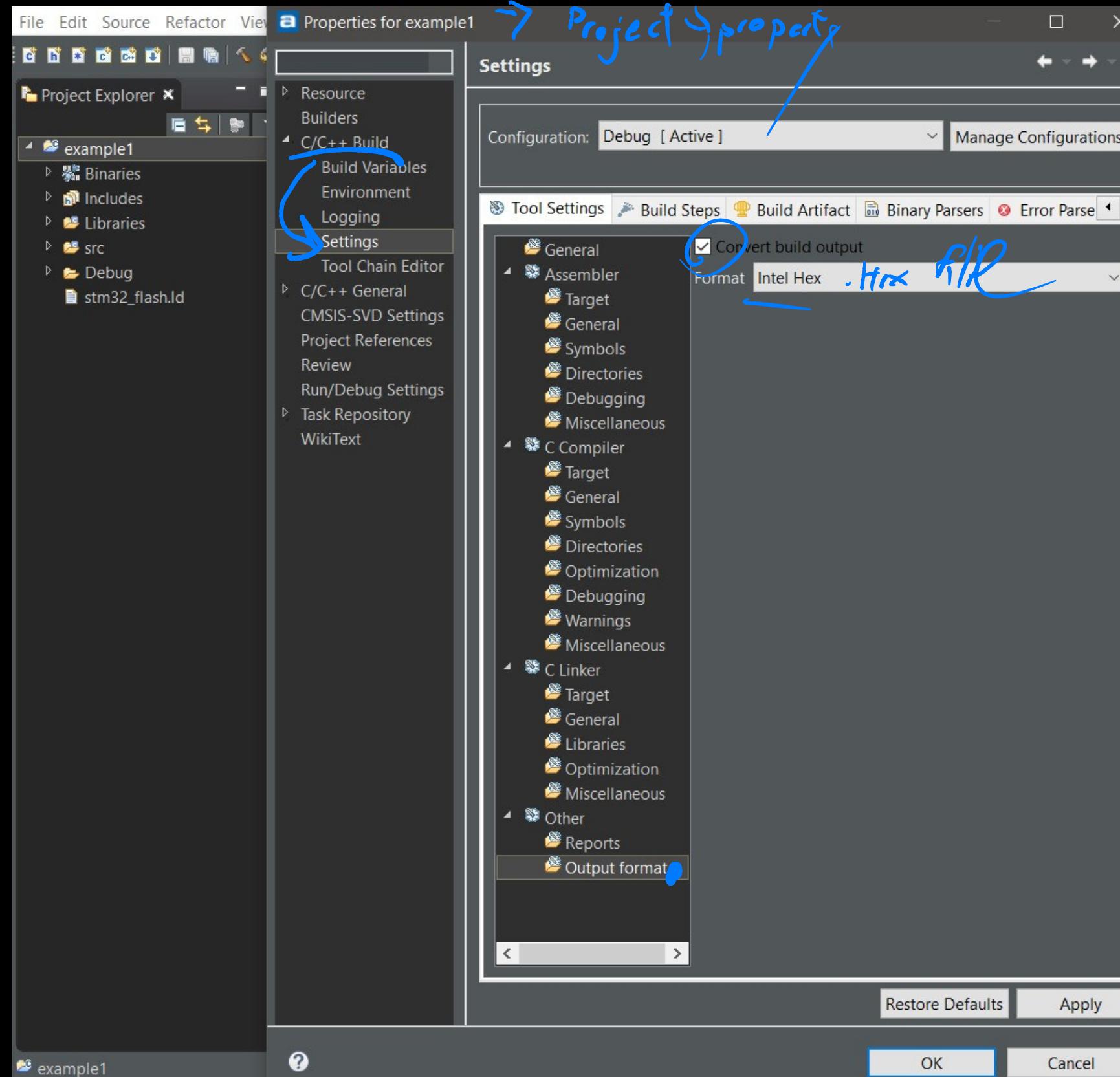
✓

↑
j = 0; O not null

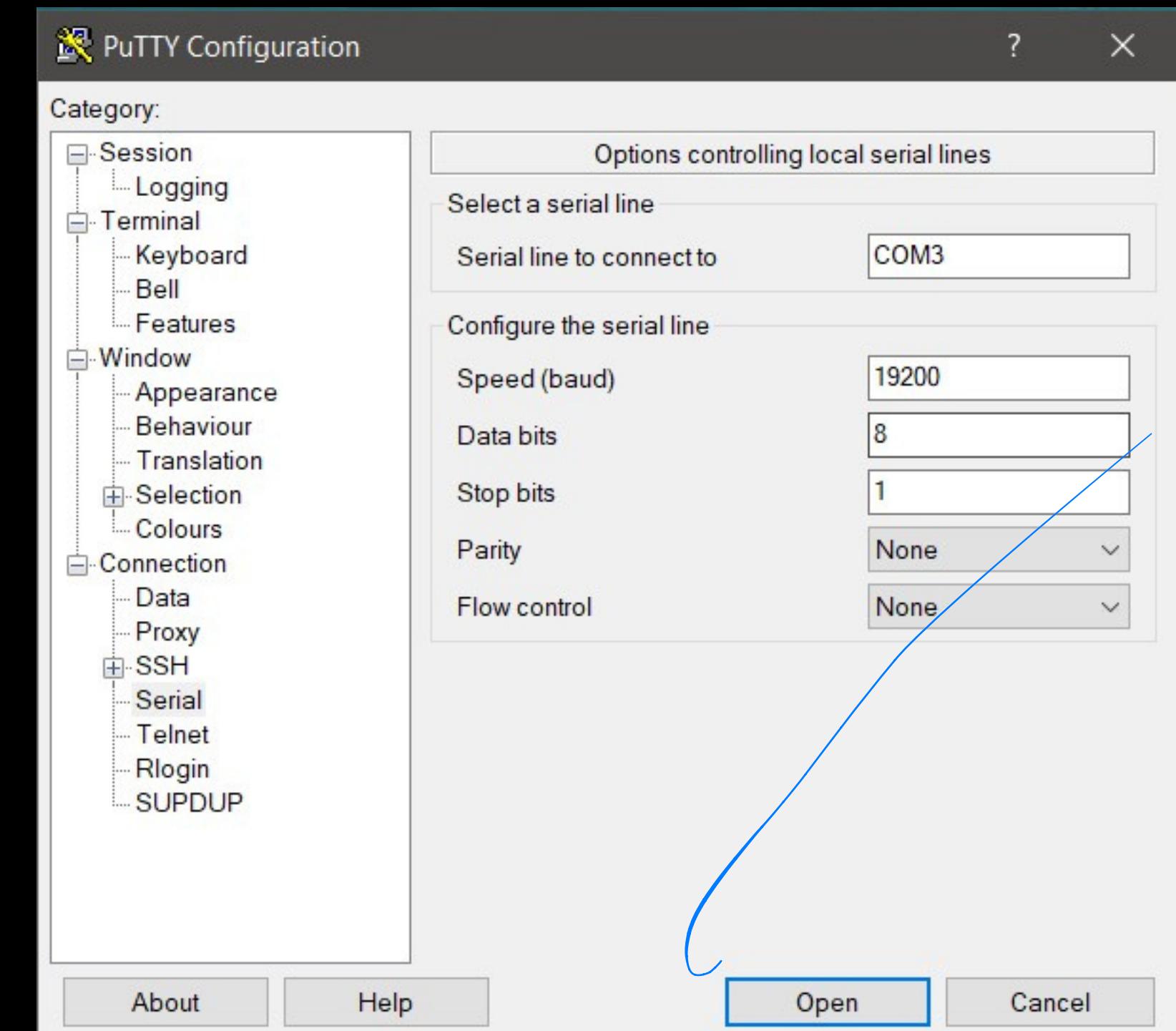
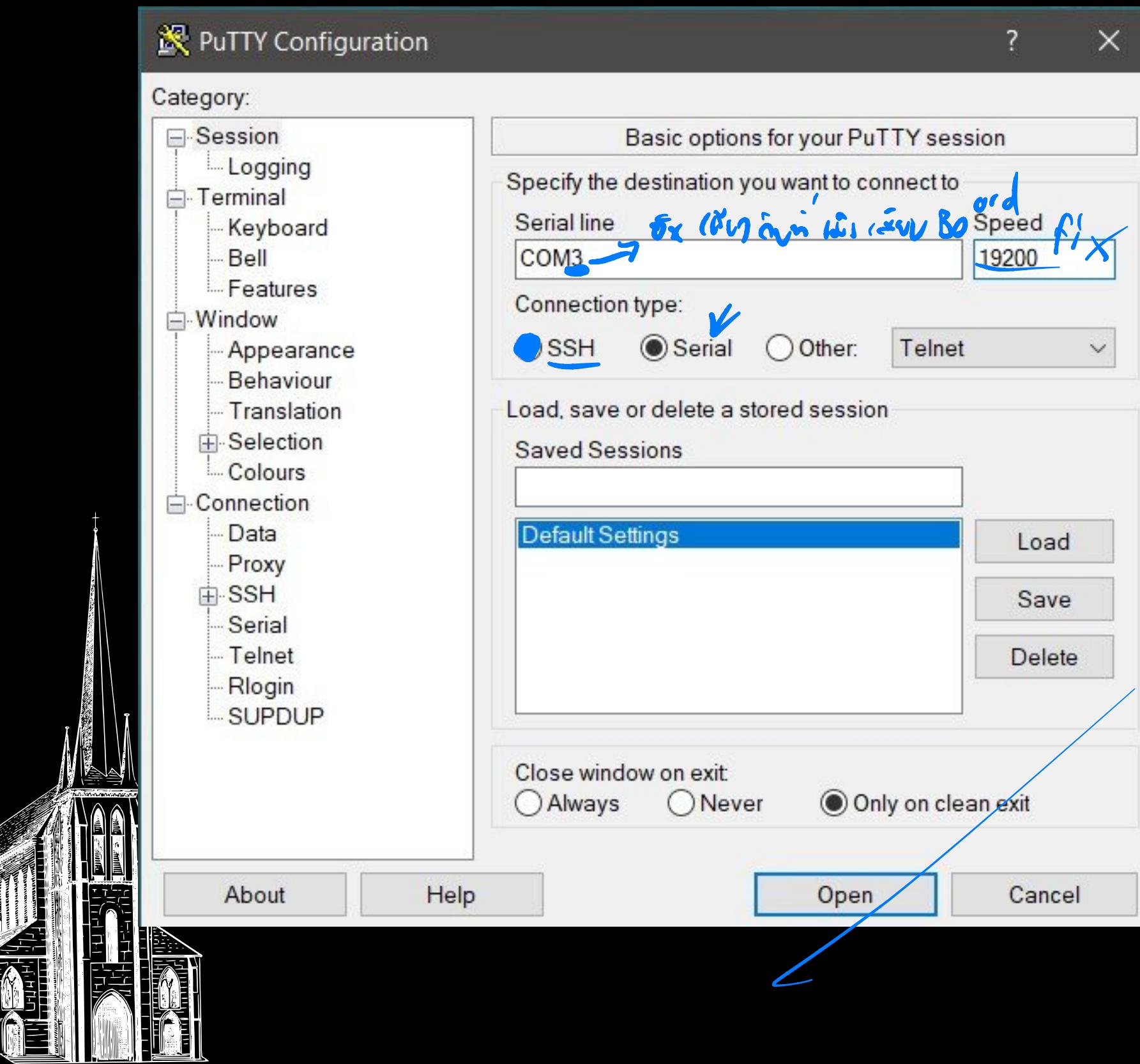
ans[j] != '\0'

Create a forever loop that Read the serial data byte by byte and then assign then input data to array of char named input. Then line 65: after enter new line it will compare 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

CONVERT FORMAT TO .hex



PuTTY CONFIGURATION



Output



```
COM3 - PuTTY

Hello World
Nice to meet you
I can chat with you
Hello
: Hello, Good to see you.
Nice to meet you
: Nice to meet you too.
What is your hobby
: My hobby is chatting online.
Tell me something
: Terminator is a bad robot.
Bye
: Ok, see you later.
Luv you
: I do not understand the question.
|
```