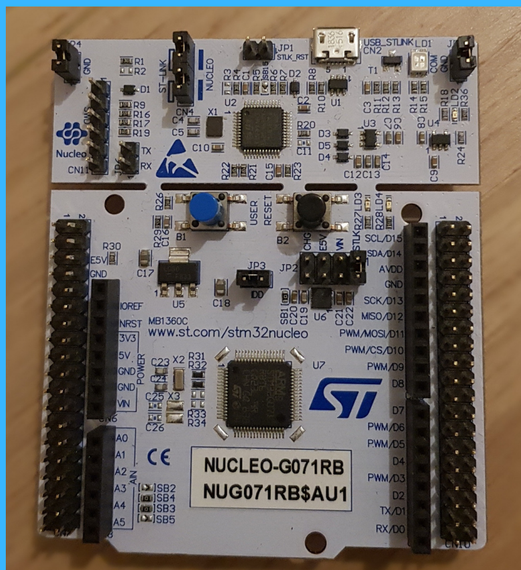
STM32 ESSENTIALS

(LP)UART and remapping, Function Prototypes, Baudrates and Printf

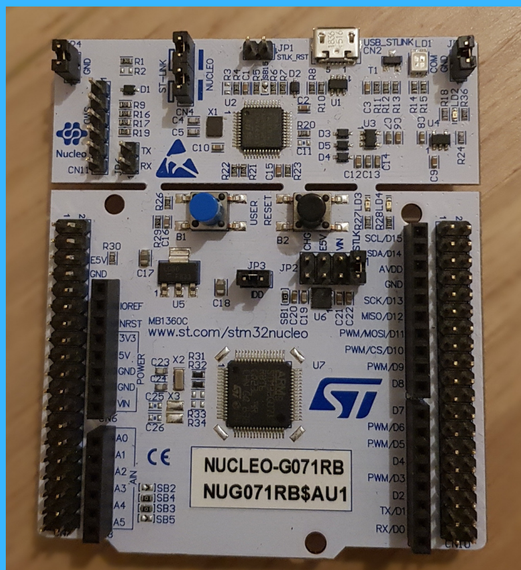
LEARNING OBJECTIVES

- Learn how to configure and send data via (LP) UART
- Customize baudrates
- Learn Function prototypes for printf on Embedded Systems
- Bonus: Simple Alphabet Converter



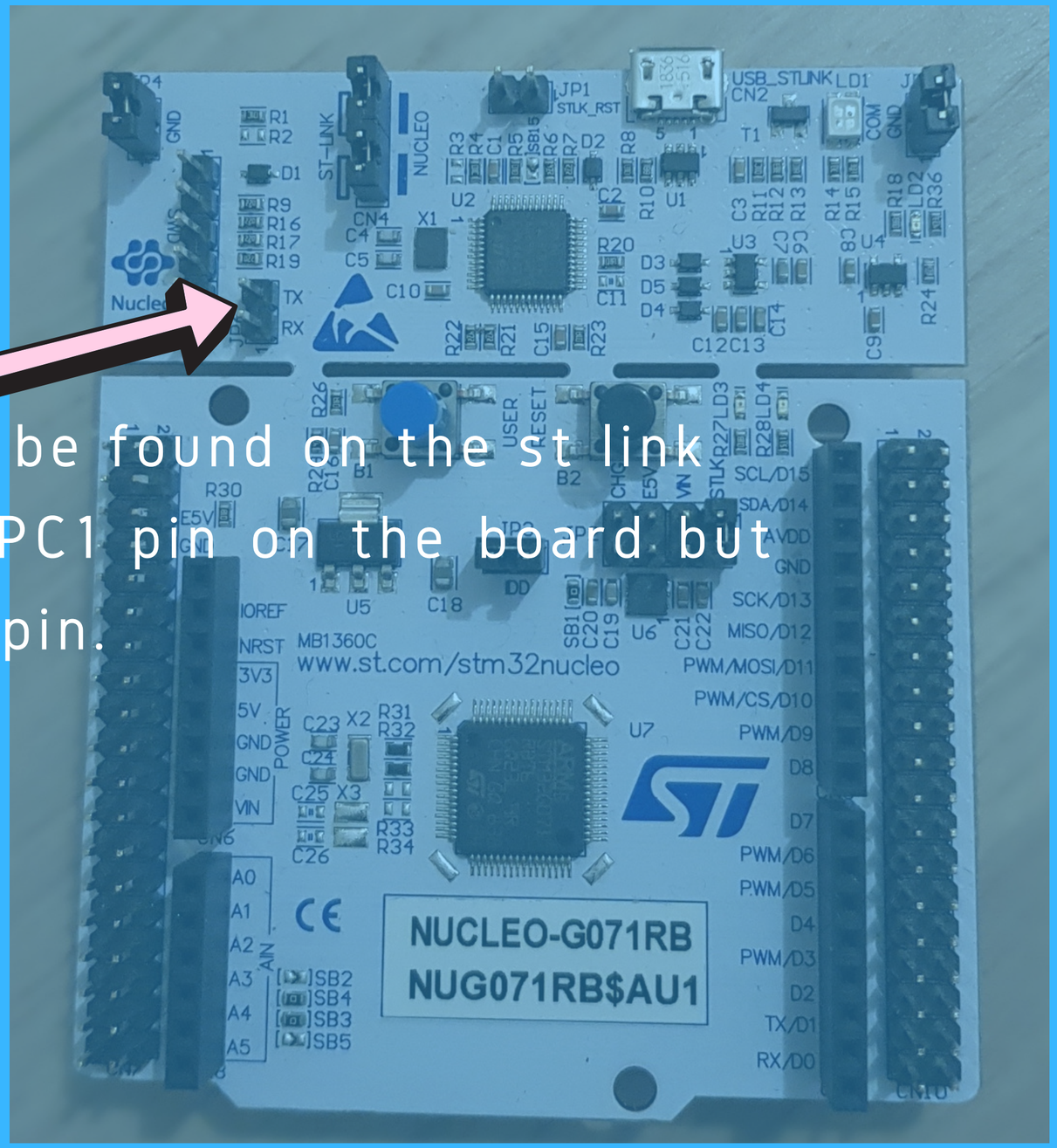
PRE-REQUISITES

- Already completed the first entry
- Tera Term or any other terminal emulation program



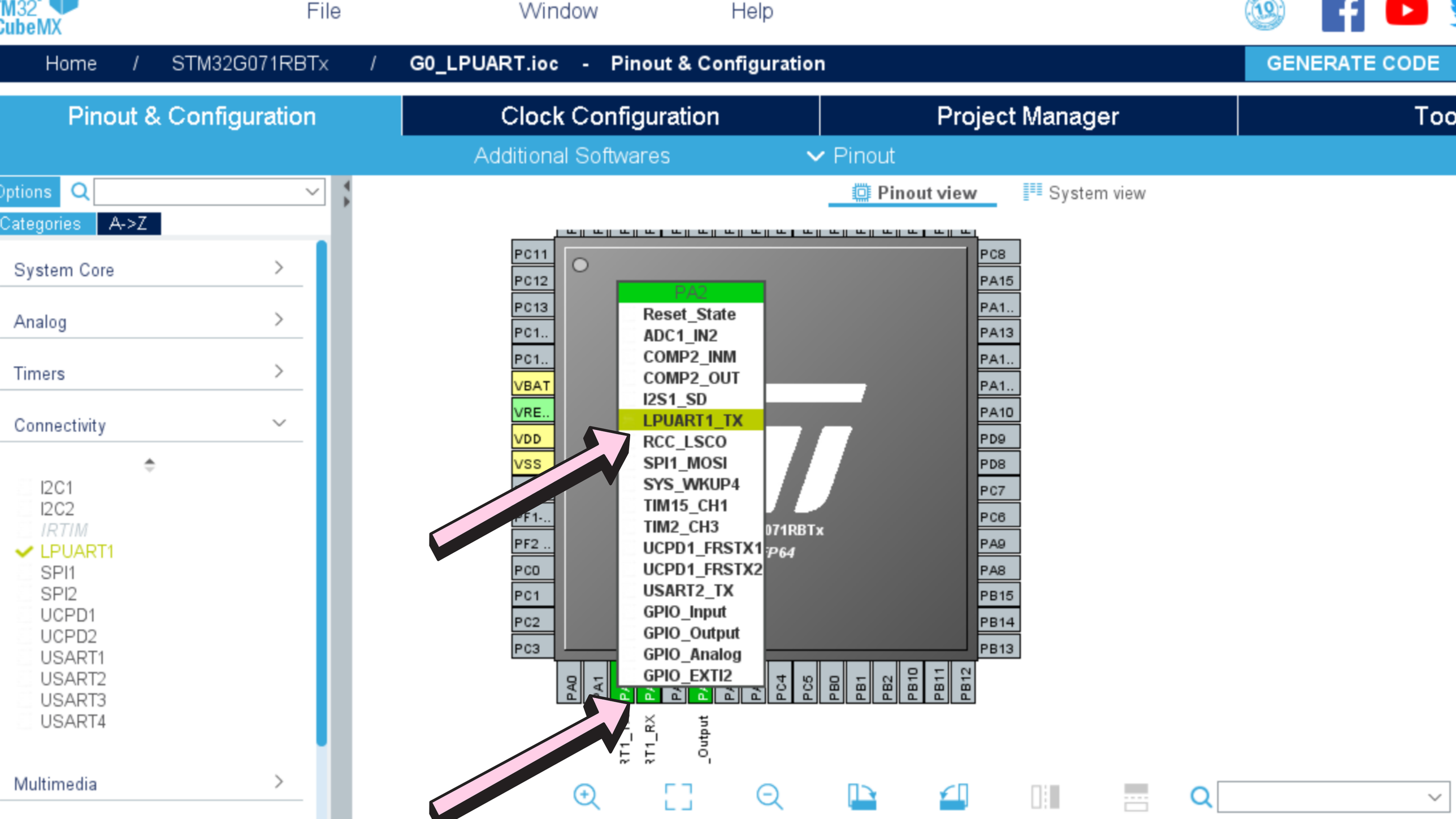
- Welcome to the second entry to the series. I hope you are more familiar with CubeMX. This next part, will be based on data transmission.

- The UART pins (RX, TX) can be found on the st link
- Connected to the PC0 and PC1 pin on the board but can be mapped to a custom pin.



- Open CubeMX and create a new project. Under connectivity, look for LPUART. PC0 and PC1 are the dedicated UART pins but we will use the PA2 and PA3 pins

- Click on PA2 and select LPUART1_TX


File
Window
Help
Home / STM32G071RBTx / G0_LPUART.ioc - Pinout & Configuration
GENERATE CODE
Pinout & Configuration
Clock Configuration
Project Manager
Additional Softwares
Pinout
Options
Categories
A->Z
System Core
Analog
Timers
Connectivity
I2C1
I2C2
IRTIM
LPUART1
SPI1
SPI2
UCPD1
UCPD2
USART1
USART2
USART3
USART4
Multimedia
Pinout view
System view
PA2
Reset_State
ADC1_IN2
COMP2_INM
COMP2_OUT
I2S1_SD
LPUART1_TX
RCC_LSCO
SPI1_MOSI
SYS_WKUP4
TIM15_CH1
TIM2_CH3
UCPD1_FRSTX1
UCPD1_FRSTX2
USART2_TX
GPIO_Input
GPIO_Output
GPIO_Analog
GPIO_EXTI2
PC11
PC12
PC13
VBAT
VDD
VSS
PF2 ..
PC0
PC1
PC2
PC3
PC8
PA15
PA13
PA10
PD9
PD8
PC7
PC6
PA9
PA8
PB15
PB14
PB13
PA0
PA1
PC4
PC5
PB0
PB1
PB2
PB10
PB11
PB12

- Click on PA3 and select LPUART1_RX

STM32 CubeMX

File Window Help

Home / STM32G071RBTx / G0_LPUART.ioc - Pinout & Configuration GENERATE CODE

Pinout & Configuration | Clock Configuration | Project Manager | Tools

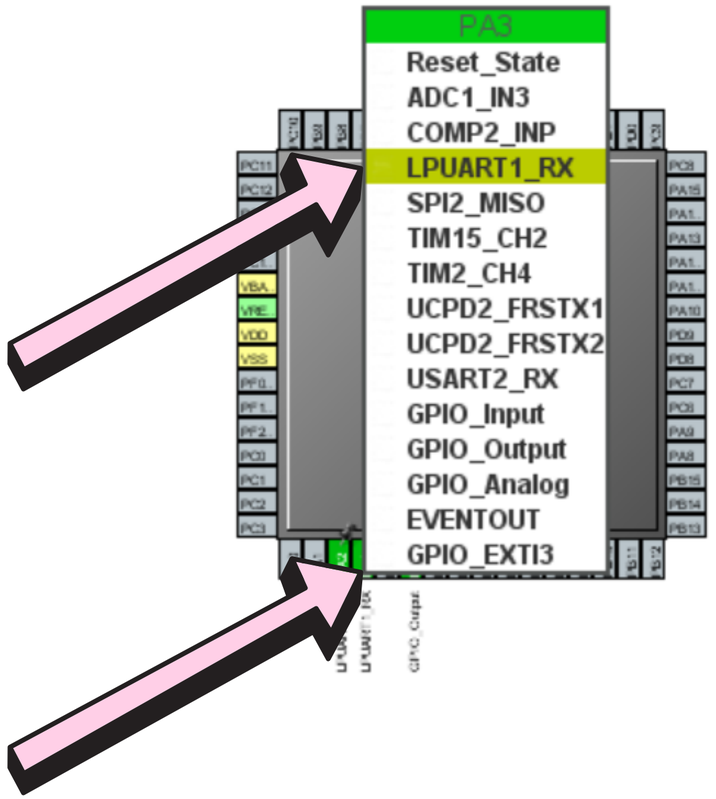
Additional Softwares | Pinout

Options

Categories A->Z

- System Core
- Analog
- Timers
- Connectivity
- Multimedia
- Computing
- Middleware

Pinout view | System view



- Under connectivity, click LPUART1 and set the mode to Asynchronous

- UART is now configured on your board. Feel free to configure also the onboard LED if you wish. Next, we need to update the baudrate. Under connectivity, click LPUART1

File Window Help

Home / STM32G071RBTx / **G0_LPUART.ioc - Pinout & Configuration** GENERATE CODE

Pinout & Configuration | **Clock Configuration** | **Project Manager** | **Tools**

Additional Softwares | Pinout

tions | ategories | A->Z

- System Core
- Analog
- Timers
- Connectivity
  - I2C1
  - I2C2
  - IRTIM
  - LPUART1
  - SPI1
  - SPI2
  - UCPD1
  - UCPD2
  - USART1
  - USART2
  - USART3
  - USART4
- Multimedia

LPUART1 Mode and Configuration

Mode

Mode: Asynchronous

Hardware Flow Control (RS232): Disable

Configuration

Reset Configuration

DMA Settings | GPIO Settings | Parameter Settings | User Constants | NVIC Settings

Configure the below parameters :

Search (Crtl+F)

Basic Parameters

Baud Rate	115200 Bits/s
Word Length	8 Bits (including Parity)
Parity	None
Stop Bits	1

Advanced Parameters

Data Direction	Receive and Transmit
Single Sample	Disable
Prescaler	clock /1
Fifo Mode	Disable
Txfifo Threshold	1 eighth full configuration

Pinout view | System view

- Change your baudrate to your prefered frequency including the word length, parity and stop bits. Remember these settings for later.
- I highly recommend the settings used above
- Keep everything else the same

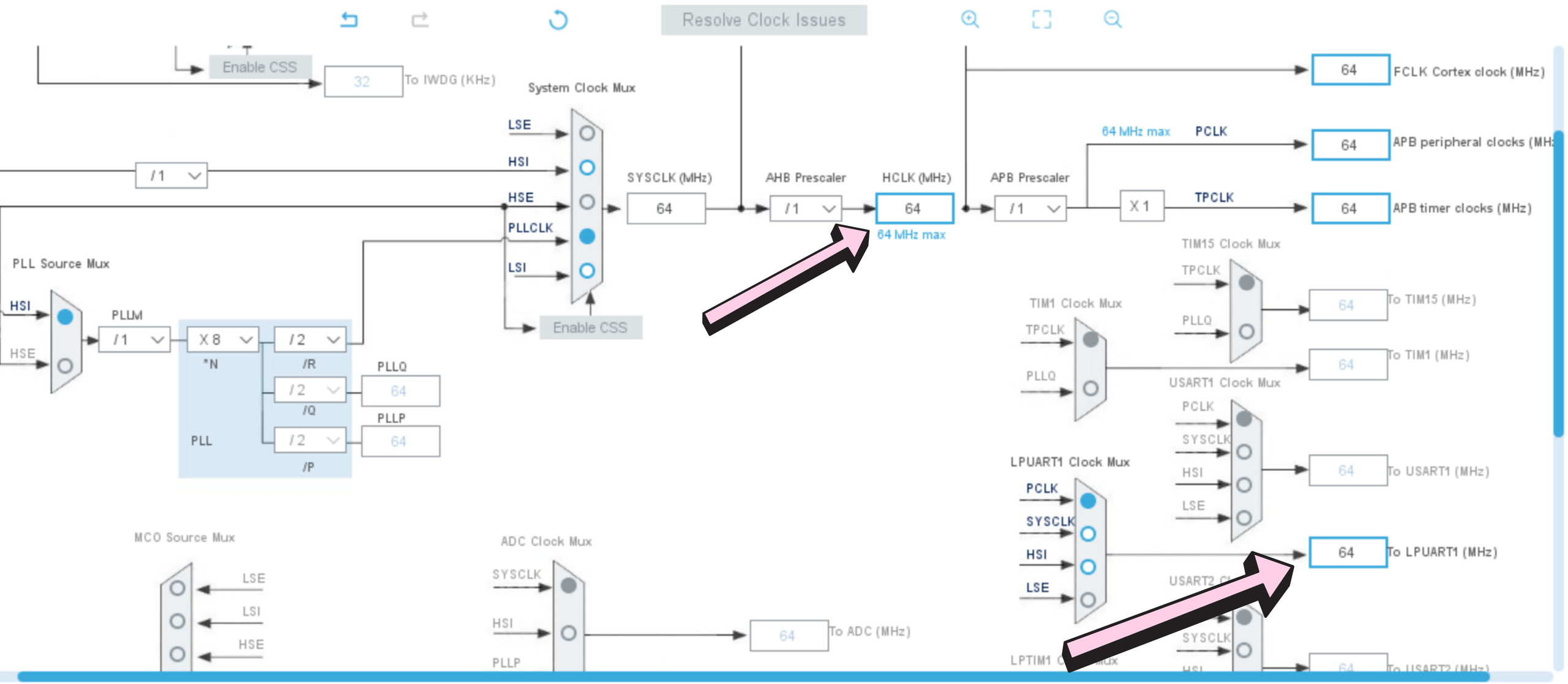
- Optional: Update the MCU to operate at maximum clockspeed of 64

MX G0_LPUART.ioc: STM32G071RBTx

File Window Help

/ STM32G071RBTx / **G0_LPUART.ioc - Clock Configuration** GENERATE CODE

inout & Configuration | Clock Configuration | Project Manager | Tools



- Finally set your paths, generate the code and open Keil

STM32CubeMX G0_LPUART.ioc: STM32G071RBTx

File Window Help

Home / STM32G071RBTx / G0_LPUART.ioc - Project Manager

GENERATE CODE

Pinout & Configuration | Clock Configuration | Project Manager | Tools

Project

Code Generator

Advanced Settings

Project Name
G0_LPUART

Project Location
C:\Users\Mwape\workspace\STM32

Application Structure
Basic
Do not generate the ma...

Toolchain Folder Location
C:\Users\Mwape\workspace\STM32\G0_LPUART\

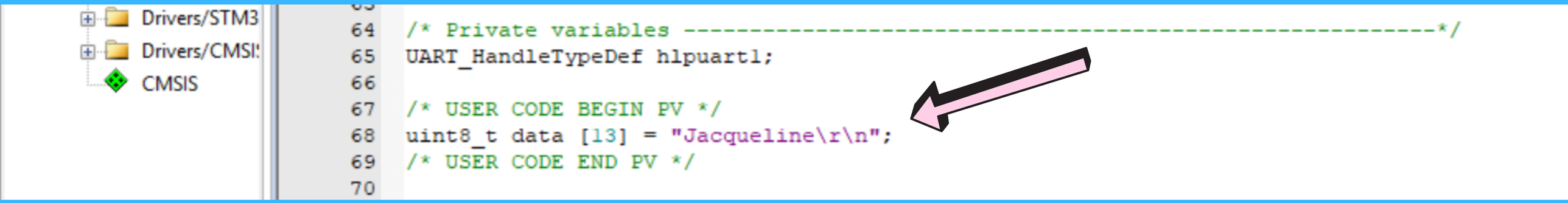
Toolchain / IDE
MDK-ARM V5
Generate Under Root

Linker Settings
Minimum Heap Size 0x200
Minimum Stack Size 0x400

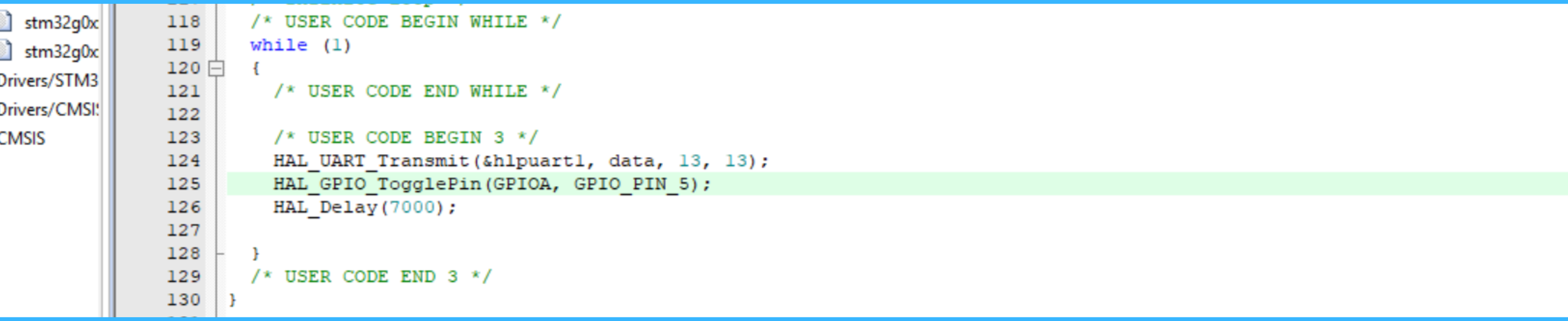
Mcu and Firmware Package
Mcu Reference
STM32G071RBTx

Firmware Package Name and Version
STM32Cube FW_G0 V1.0.0

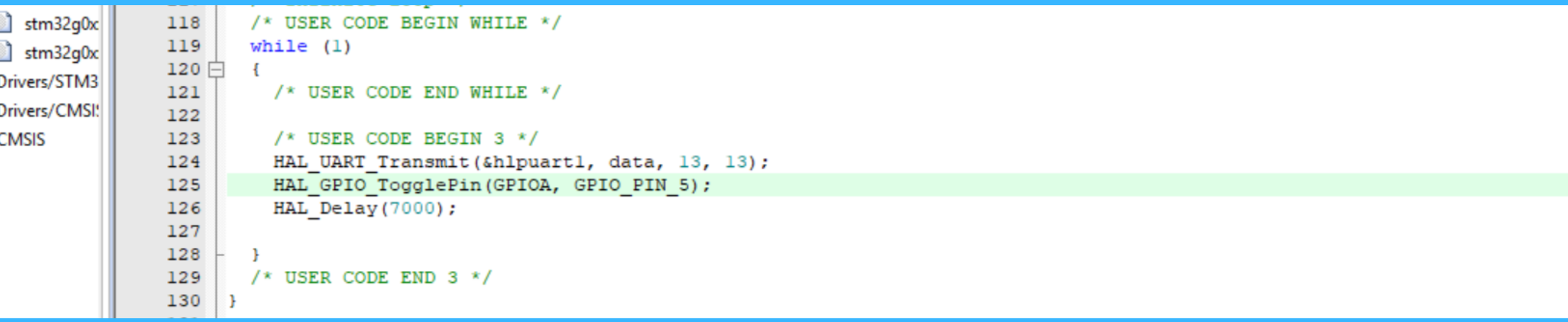
- Let’s test out if UART is properly configured.
- Declare an array with text data you want to send



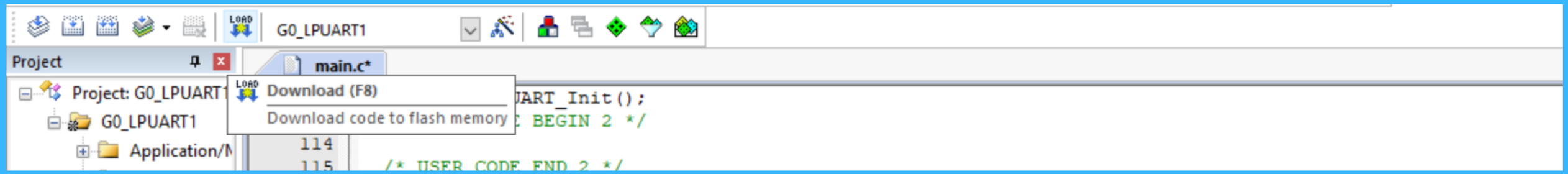
- Transmit this text through UART buy using the STM32's HAL function. The list of functions are located at the bottom of the left sidebar.



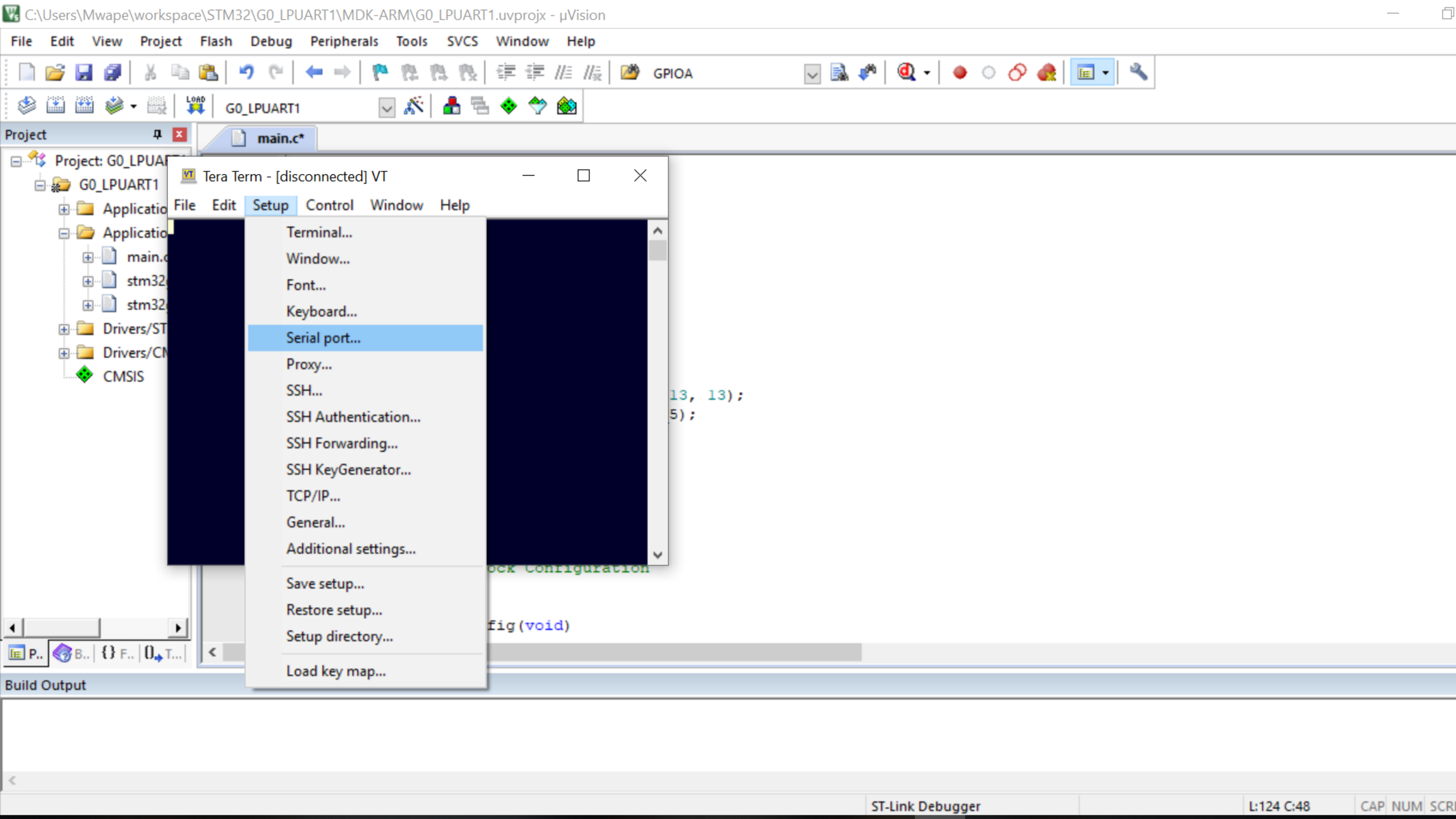
- The HAL_UART_Transmit function takes in the address of the uart rx/tx pins (hlpuart1), the data array, the size of the data and the timeout.

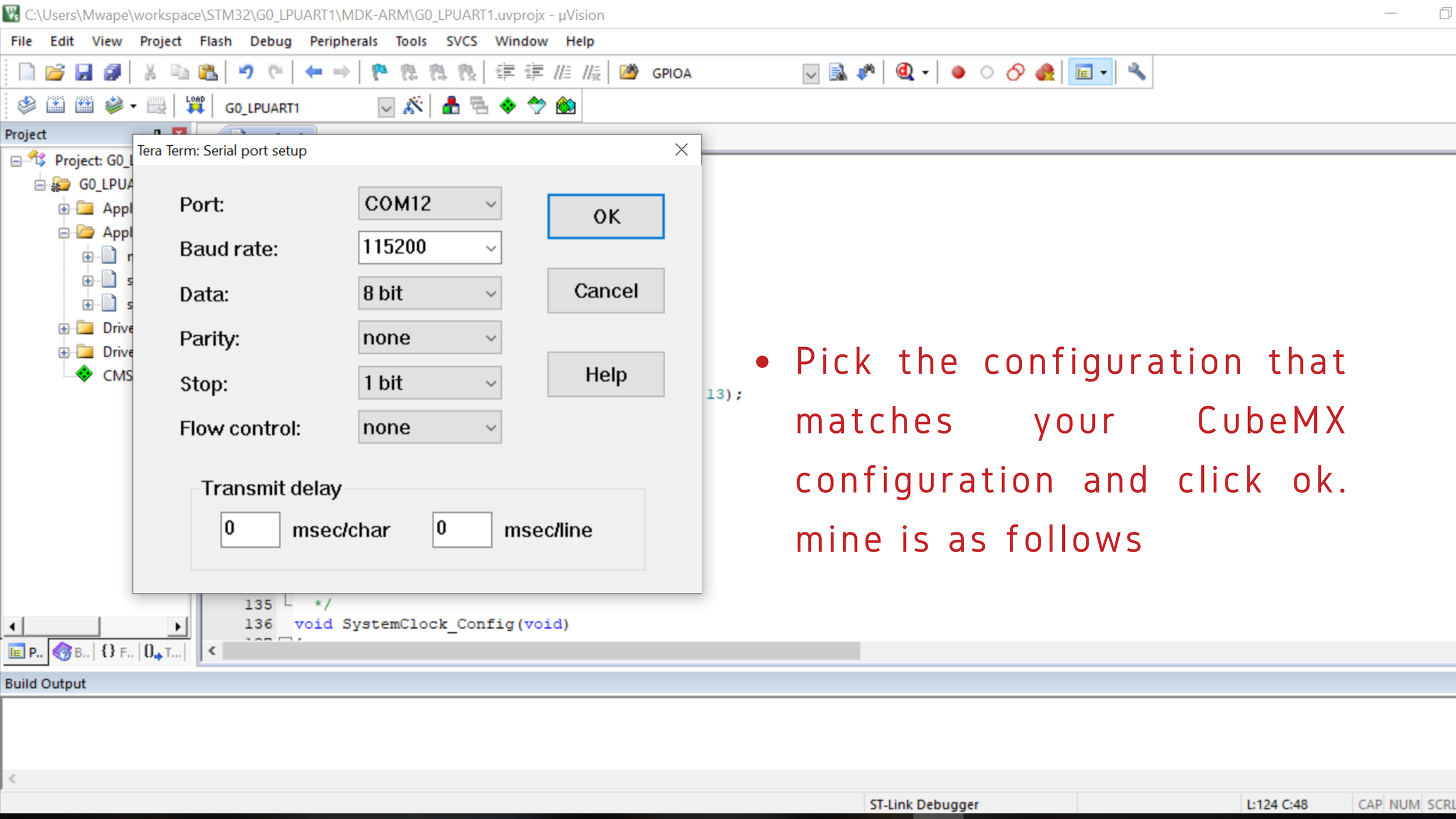


- Build and download the code to your flash

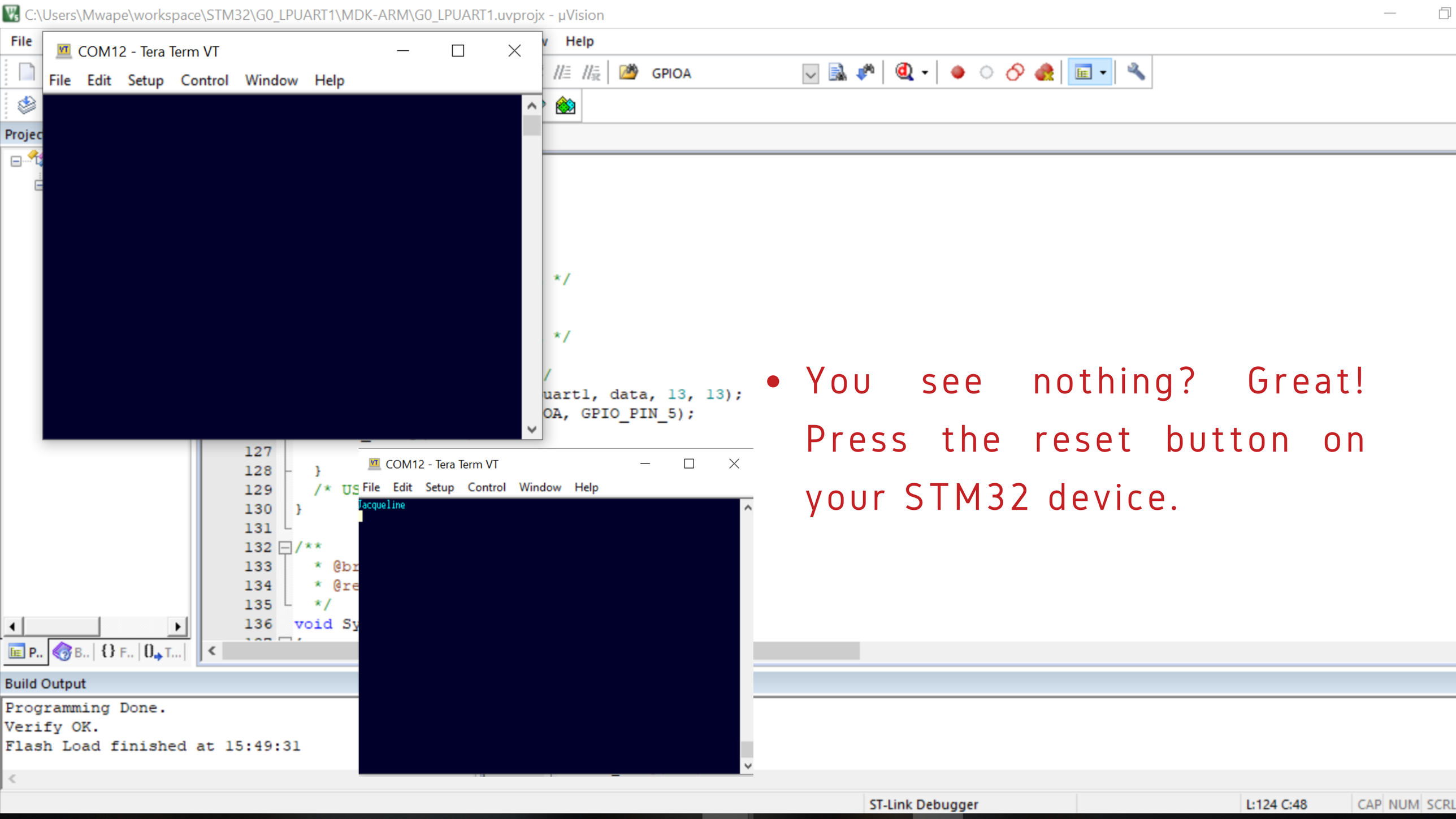


- If you havent already, download and install Tera Term
- Run it and follow these steps:


C:\Users\Mwape\workspace\STM32\G0_LPUART1\MDK-ARM\G0_LPUART1.uvprojx - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
GPIOA
G0_LPUART1
Project
Project: G0_LPUART1
G0_LPUART1
Drivers/ST
Drivers/CM
CMSIS
main.c*
Tera Term - [disconnected] VT
File Edit Setup Control Window Help
Terminal...
Window...
Font...
Keyboard...
Serial port...
Proxy...
SSH...
SSH Authentication...
SSH Forwarding...
SSH KeyGenerator...
TCP/IP...
General...
Additional settings...
Save setup...
Restore setup...
Setup directory...
Load key map...
13, 13);
5);
fig(void)
Build Output
ST-Link Debugger
L:124 C:48
CAP NUM SCRL



- Pick the configuration that matches your CubeMX configuration and click ok. mine is as follows



- You see nothing? Great! Press the reset button on your STM32 device.

- As you probably realised, it’s pretty tedious just to print something on a screen so now we will use printf with help of custom function prototypes that redirect printf to UART.
- All source code will be available with this document.

C:\Users\Mwape\workspace\STM32\G0_LPUART\MDK-ARM\G0_LPUART.uvprojx - µVision

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help

GPIOA

G0_LPUART

Project

Project: G0_LPUART

G0_LPUART

Application/M

startup_st

Application/U

main.c

stm32g0x

stm32g0x

Drivers/STM3

Drivers/CMSI

CMSIS

main.c

```
81  /* USER CODE END PV */
82
83  /* Private function prototypes -----------------------------------------------*/
84  void SystemClock_Config(void);
85  static void MX_GPIO_Init(void);
86  static void MX_LPUART1_UART_Init(void);
87  /* USER CODE BEGIN PFP */
88  #define PUTCHAR_PROTOTYPE int fputc(int ch, FILE *f)
89  #define  GETCHAR_PROTOTYPE  int f getc ( FILE * f)
90  /* USER CODE END PFP */
91
92  /* Private user code ---------------------------------------------------------*/
93  /* USER CODE BEGIN 0 */
94  void translator(){   //debugging
95      char input[50] = {"Hello world"};
96      //fgets(input, 2048 ,stdin);
97      //printf("%s", input);
98      //convert = (char*)malloc(strlen(convert)*sizeof(char));
99      HAL_Delay(2000);
100
101     //String convert = Serial.readStringUntil('\n'); //reads serial data line until new line
102     //Serial.println(convert);
103     printf("Input is: %s\r\n", input);
104     int inputlength = strlen(input);
105     int letterlength = strlen(letter);
```

- #define a PUTCHAR prototype. The GETCHAR will be covered in the next entry of the series.

Build Output

C:\Users\Mwape\workspace\STM32\G0_LPUART\MDK-ARM\G0_LPUART.uvprojx - µVision

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help

putchar

G0_LPUART

Project

Project: G0_LPUART
G0_LPUART
Application/M
startup_st
Application/U
main.c
stm32g0x
stm32g0x
Drivers/STM3
Drivers/CMSI
CMSIS

main.c

```
299
300   /*Configure GPIO pin Output Level */
301   HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, GPIO_PIN_RESET);
302
303   /*Configure GPIO pin : PA5 */
304   GPIO_InitStruct.Pin = GPIO_PIN_5;
305   GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
306   GPIO_InitStruct.Pull = GPIO_NOPULL;
307   GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
308   HAL_GPIO_Init(GPIOA, &GPIO_InitStruct);
309
310 }
311
312 /* USER CODE BEGIN 4 */
313 PUTCHAR_PROTOTYPE
314 {
315   HAL_UART_Transmit(&hlpuart1, (uint8_t *)&ch, 1, 0xFFFF);
316   return ch;
317 }
318
319
320 /* USER CODE END 4 */
321
322 /**
323   * @brief  This function is executed in case of error occurrence.
```

- The prototype simply redirects all data sent to the printf through the UART which was defined above.

Build Output

ST-Link Debugger L:315 C:1 CAP NUM

C:\Users\Mwape\workspace\STM32\G0_LPUART\MDK-ARM\G0_LPUART.uvprojx - µVision

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help

putchar

G0_LPUART

Project

Project: G0_LPUART
G0_LPUART
Application/M
startup_st
Application/U
main.c
stm32g0x
stm32g0x
Drivers/STM3
Drivers/CMSI
CMSIS

main.c

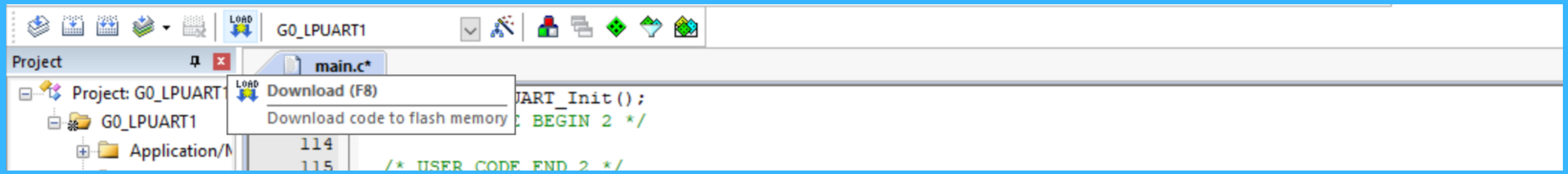
```
164
165   /* Initialize all configured peripherals */
166   MX_GPIO_Init();
167   MX_LPUART1_UART_Init();
168   /* USER CODE BEGIN 2 */
169
170   /* USER CODE END 2 */
171
172   /* Infinite loop */
173   /* USER CODE BEGIN WHILE */
174   while (1)
175   {
176     //printf("** Hello World ** \n\r");
177     HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_5);
178     printf("--> SMT32 ESSENTIALS Printf to (LP)UART ++ \r\n");
179     translator();
180     HAL_Delay(5000);
181     /* USER CODE END WHILE */
182
183     /* USER CODE BEGIN 3 */
184   }
185   /* USER CODE END 3 */
186 }
187
188 /**
```

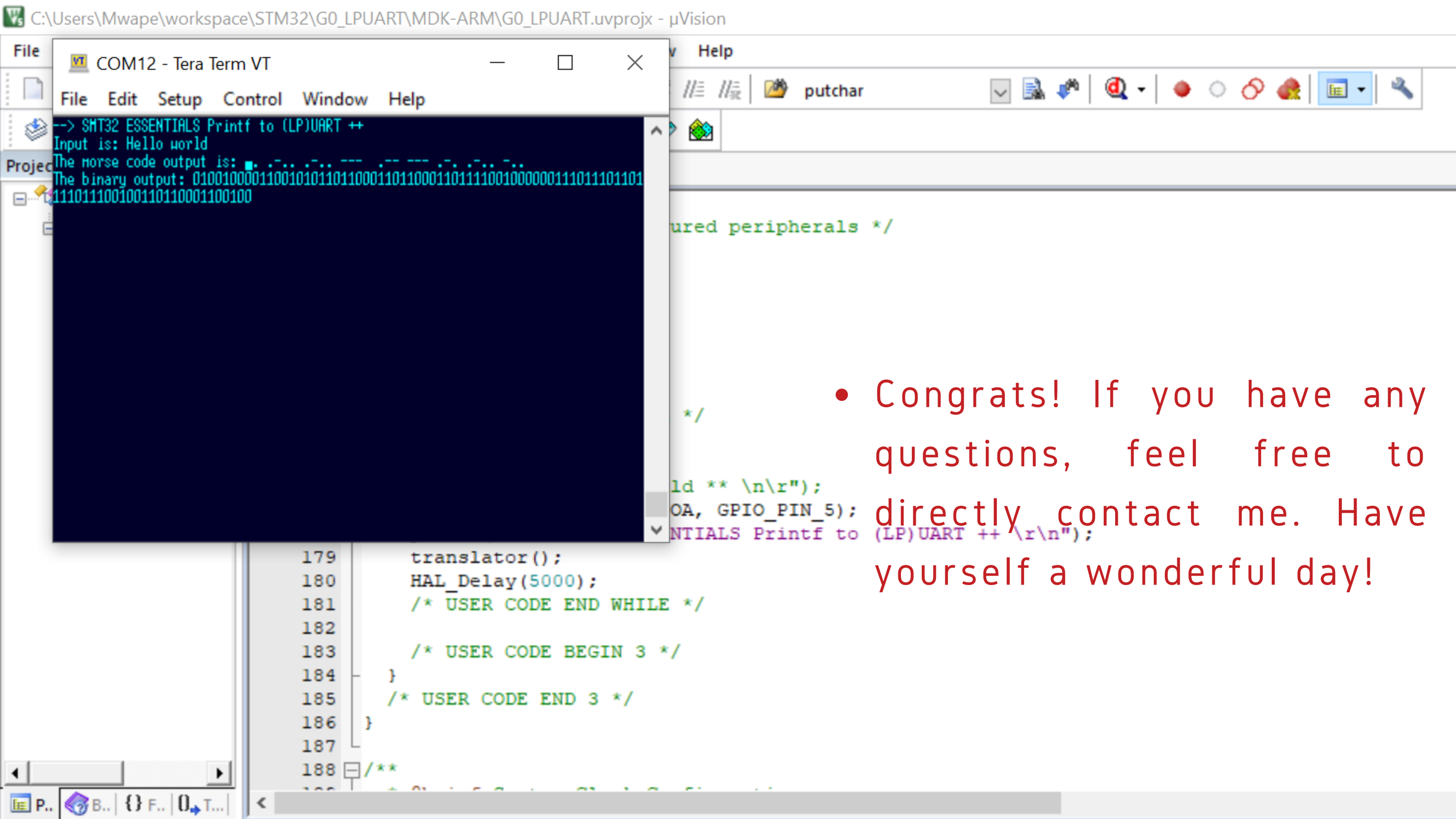
- Do printf as you normal would. In this case, I will test my alphabet translator which is called underneath

Build Output

ST-Link Debugger L:178 C:63 CAP NUM SCR

- Build and download the code to your flash
- Press the reset button on your STM32 device




C:\Users\Mwape\workspace\STM32\G0_LPUART\MDK-ARM\G0_LPUART.uvprojx - µVision
File
Help
putchar
COM12 - Tera Term VT
File Edit Setup Control Window Help
--> SMT32 ESSENTIALS Printf to (LP)UART ++
Input is: Hello world
The morse code output is: ■. .-.. .-.. --- .-- --- .-. .-.. -..
The binary output: 0100100000110010101101100011011000110111100100000011101110110111101110011001101100011001100
ured peripherals */
*/
ld ** \n\r");
OA, GPIO_PIN_5);
NTIALS Printf to (LP)UART ++ \r\n");
translator();
HAL_Delay(5000);
/* USER CODE END WHILE */
/* USER CODE BEGIN 3 */
}
/* USER CODE END 3 */
}
/**
Congrats! If you have any questions, feel free to directly contact me. Have yourself a wonderful day!