1. **Setup hardware**

STlink on stm32 does not provide data transmission function, so we will use USB to UART module to transmit data.

A close-up of a usb flash drive

Description automatically generated

Connect the STM32 wire and the UART module according to the picture below:

A white sheet with black text

Description automatically generated

Then, plug the UART module into the computer to transmit data.

1. **Setup software**

**2.1 Hercules Terminal**

Because the data is transmitted via uart, we will need a terminal to view it. In this example I use Hercules for display. You can download by this link [Link download Hercules](https://www.hw-group.com/product-version/hercules).

Archive and install it, you’ll have Hercules on your desktop. When you open it, you can communicate via UART.

A screenshot of a computer

Description automatically generated

Some information in this window:

* Baund rate (baund rate): The amount of time for 1 bit to be transmitted. Must be set the same for sending and receiving. Some common Baud Rates: 9600, 38400, 115200, 230400,...
* Name: is the COM port, the port where the UART connects to your computer.
* You can see send and receive display, click Open and test it. When you enter data and click Send, data will be transferred but it hasn't been sent to STM32 yet.

A screenshot of a computer

Description automatically generated

## **Setup for STM32CubeMX**

Open STM32CubeMX, the initial setup will be the same as the previous example. Click on USART1 and setting like picture below. When you click on USART1, you can see PA9 and PA10 pin light up.

A screenshot of a computer

Description automatically generated

Click NVIC and tick Enabled on USART1 global interrupt.

A screenshot of a computer

Description automatically generated

Generate code and you can move to Keilc for next step.

## **Setup for Keilc**

Open Keilc and you’ll have Hal\_Uart in your project.

A screenshot of a computer

Description automatically generated

Click file hal\_uart.c to open it, press Ctrl+F to find keyword “HAL\_UART\_Transmit” and you can see this function that use for transmit data via UART.

A screenshot of a computer

Description automatically generated

In order to using it, we need to declare it in main.c

First, declare variable TX, this variable will help you save the transmit data.



And in while(1) loop, we write this command to print “hello world” to terminal.  


Build and load code to STM32. Open Hercules, click Serial and choose port, which UART connect to. You can open Device Manager on Control Pannel to see.

A close up of a white background

Description automatically generated

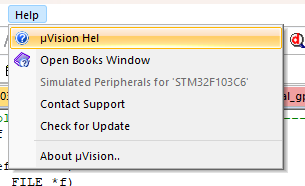
You can see hello world message on terminal.

A screenshot of a computer

Description automatically generated

Another way to print is to use the printf function. In keilc, if you want to use printf, you must retarget this function.

Open Help → uVision Hel and search Retargeting Printf. You will see instructions how to using printf in Keilc.



A screenshot of a computer

Description automatically generated

Include stdio.h and declare \_\_stdout.

A white screen with text

Description automatically generated with medium confidence

Retarget fputc and write command to transmit, you must declare this function right below huart1.

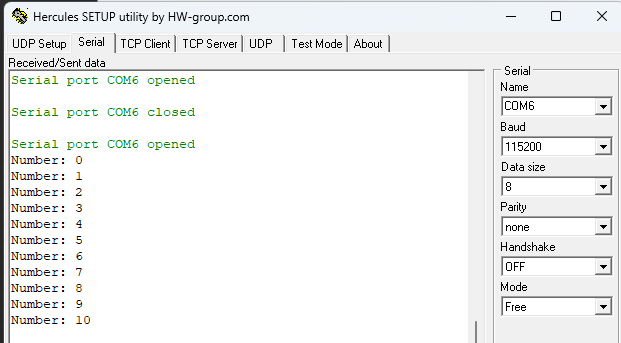
A white background with black text

Description automatically generated

Declare counter variable and you can using printf.



Built and Load code, you can see message in terminal.

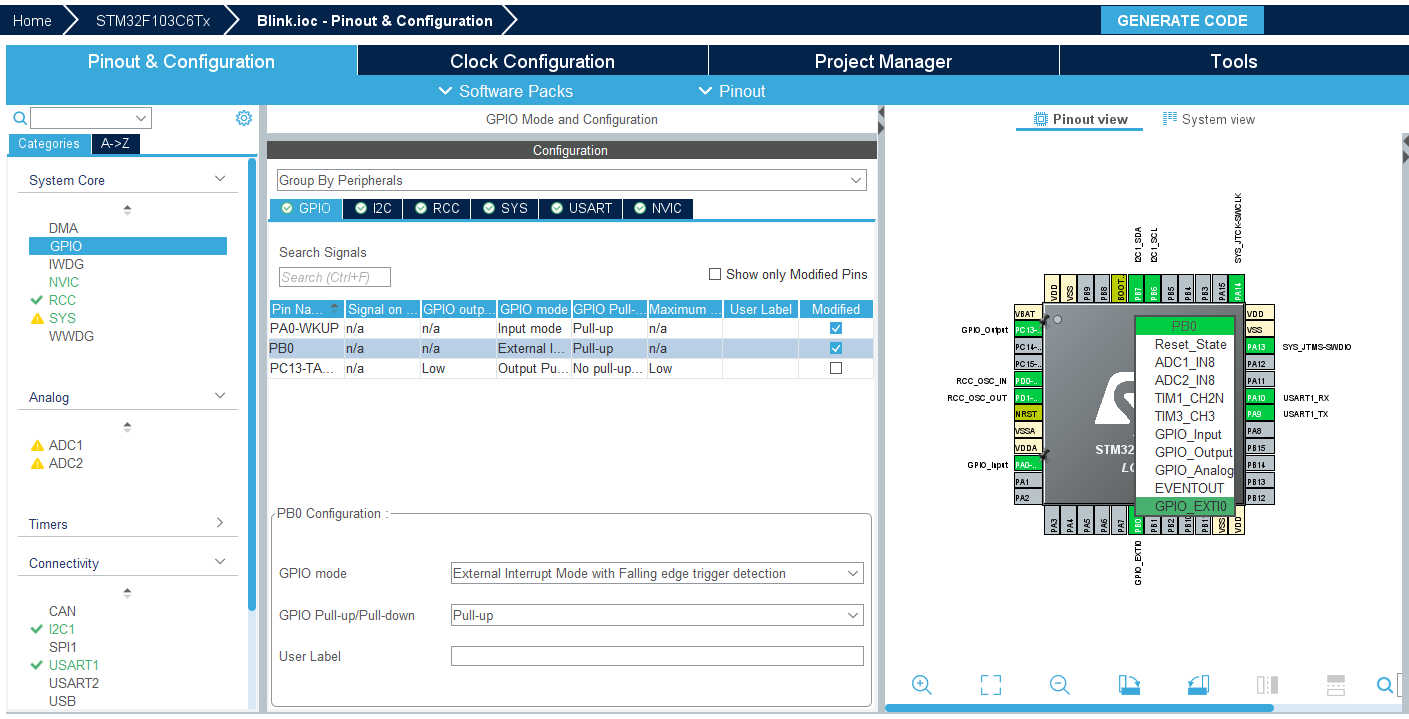


## **Debug session**

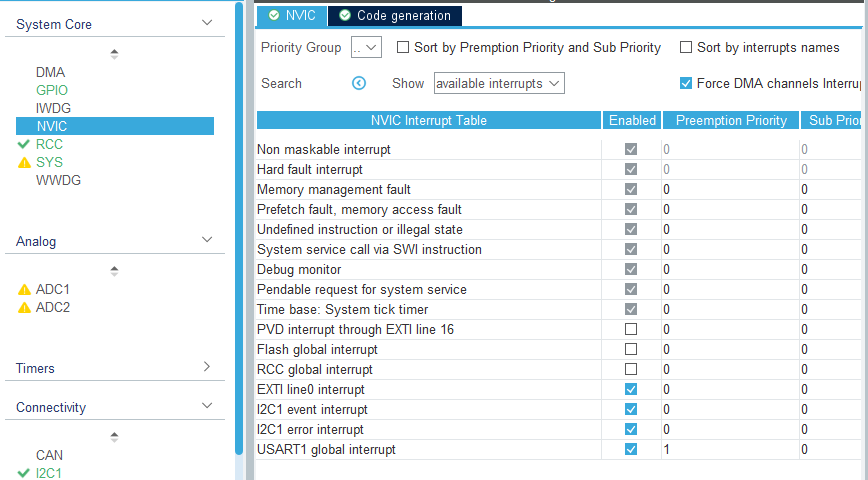
In order to using debug in Keilc, I have created an example with button using Interupt.

In previous “Hello world” example, I have printed message to console. Now, I want to send data from PC to STM32 and using Interupt to receive it.

Open STM32CubeMx and setup for PB0 Pin is EXTI0, click on GPIO and choose GPIO mode like picture below. In your board, connect PB0 Pin with one side of button and the other side of button, you will connect with GND.



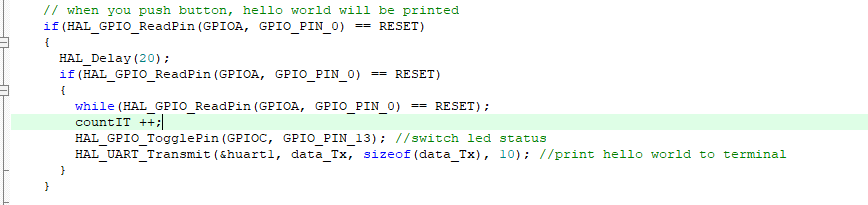
You also tick on enabled in NVIC for EXTI line0 interupt. Generate code and open KeilC for next step.



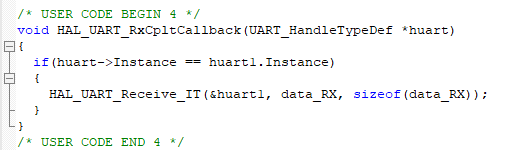
First, delcare variable to receive data. I using data\_RX



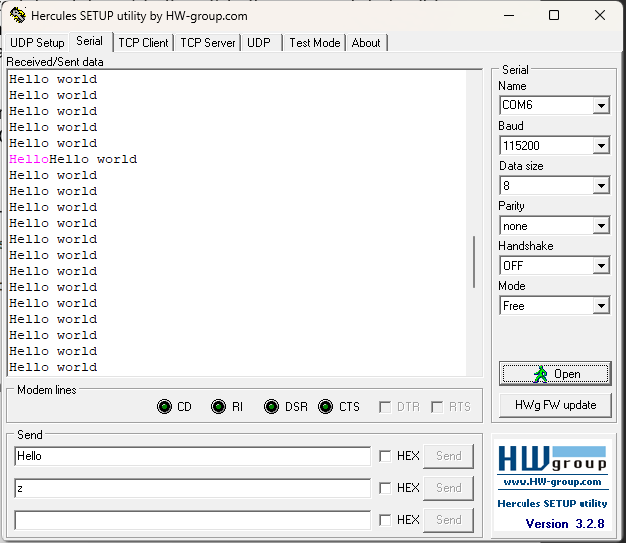
Next, I using HAL\_GPIO\_READPIN to take status of button. If button pressed, led will be switched status by HAL\_GPIO\_TogglePin function and also send message to terminal by HAL\_UART\_Transmit function. You can search this function in hal\_gpio.c and hal\_uart.c to know detail. CountIT just a variable for Debug test.



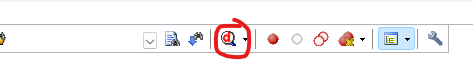
Because the interrupt only happens once, when we press the button again we have to send the message again. I using HAL\_UART\_RxCpltCallback to using Interupt again.

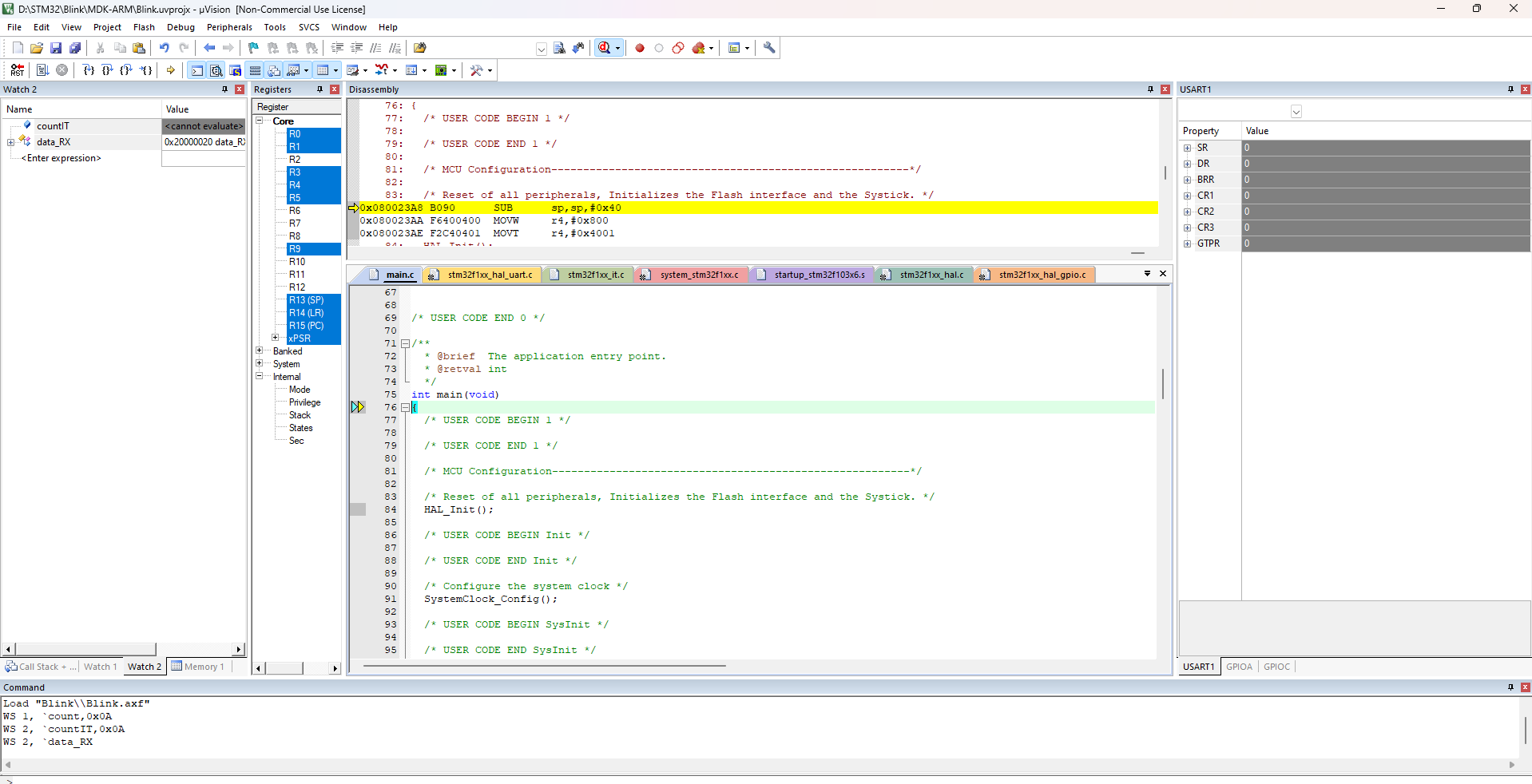


Built and Load code, open terminal and you can see when you press button, hello world will be printed in terminal.



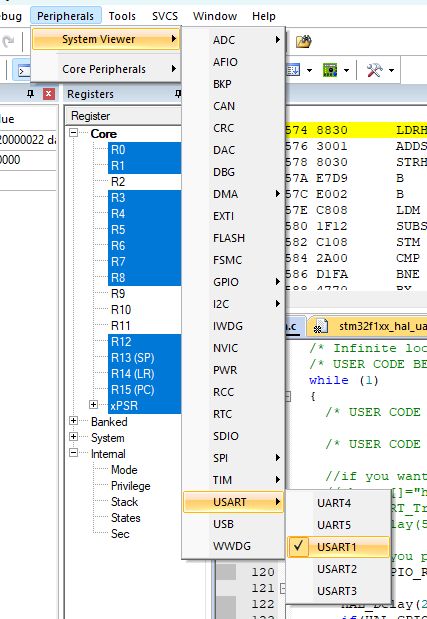
We will debug for this program. Find in menu or press Ctrl+F5 to open Debug session.



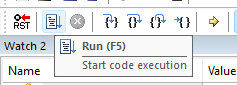


Click on countIT and choose add to watch2 to see changes during program execution. You can do the same for other variables and add to watch you want to see.

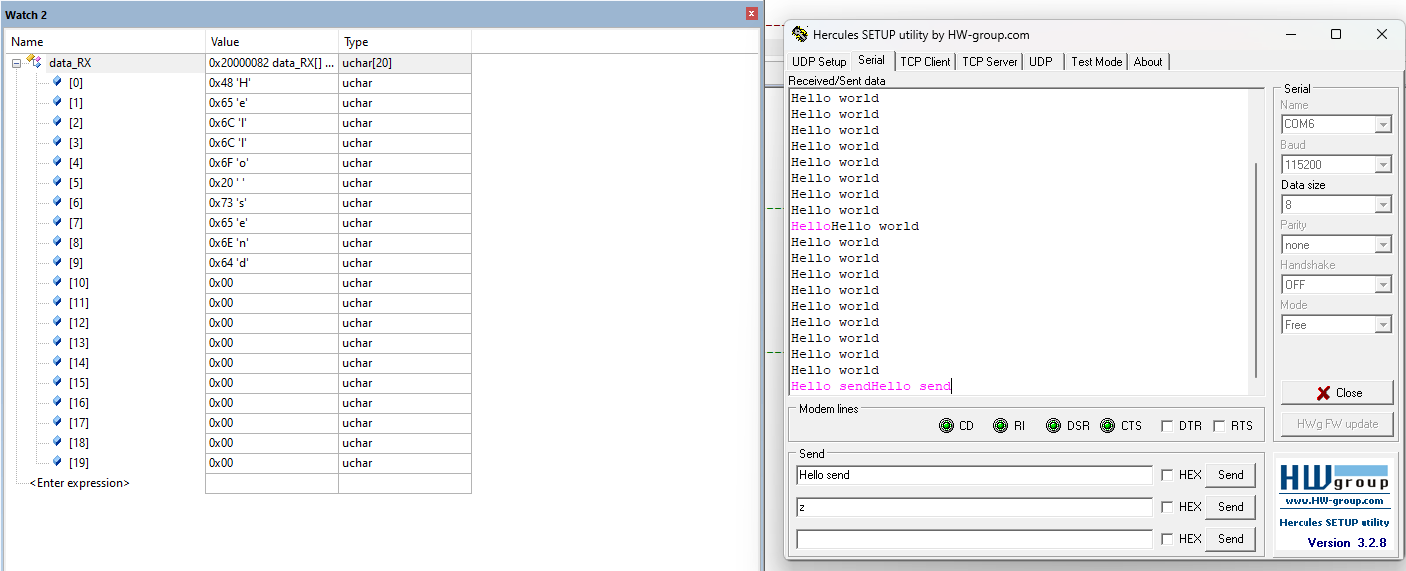
Click on Peripherals, choose UART1to open UART1 watch. Similar for GPIOC, you can see the changes of status led in this.



Click Run of press F5 to start debugging.

****

Open Hercules and send message “Hello send”, press Send. You can see changes in watch2 with dataRx variable.

****

Press the button and see the changes of value in watch. In GPIOC watch, open IDR and you can see status of led in IDR13.

A screenshot of a computer

Description automatically generated

You also set breakpoint by click in line you want to debug.

A screenshot of a computer

Description automatically generated

And then, using this to see how code running.

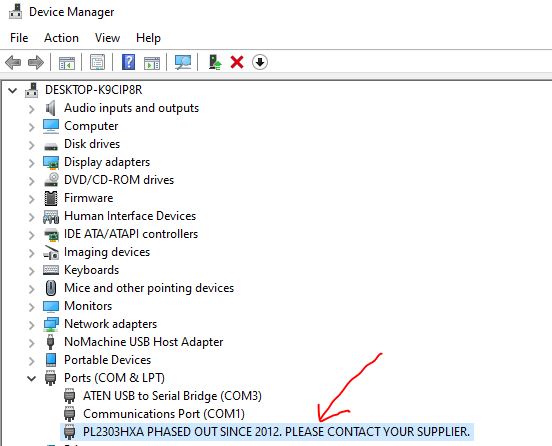
A screenshot of a computer

Description automatically generated

1. **Some common fault**

## **3.1 Pl2303 phased out since 2012 please contact your supplier**

It means that you have to install drive for your UART module.



Download drive in [this link](http://www.mediafire.com/file/982x6iyk89v95dp/Prolific_PL2303_driver_v3.3.2.102_%25282008-24-09%2529_Win8_x64_x86.7z/file) and unzip it for next step.

Click on this fault and choose update driver → Browse my computer for drivers.

A screenshot of a computer

Description automatically generated

Click Have Disk and chosse file “ser2pl” in driver folder you unziped.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Install it and you will successfully install it. Check it in Device Manager → Port.

A close-up of a computer

Description automatically generated

## **Cannot printf message to terminal**

A screenshot of a computer

Description automatically generated

It means that baud rate you setting in Hercules and STM32CubeMx is not equal. You can adjust it in USART1 and Hercules, as long as the two are equal.

