Giao tiếp UART sử dụng ngắt và DMA cho STM32F4

I, Các kiến thức cần trước khi vào bài

1. DMA

Direct Memory Access cho phép truyền dữ liệu trực tiếp từ ngoại vi vào bộ nhớ (Peripheral-to-memory) , từ bộ nhớ vào ngoại vi (Memory-to-peripheral) , giữa các bộ nhớ và bộ nhớ (Memory-to-memory) mà không qua CPU giúp CPU rảnh để làm việc khác.

Hai bộ điều khiển DMA có tổng cộng 16 streams (8 cho mỗi bộ điều khiển), mỗi streams dành riêng cho quản lý các yêu cầu truy cập bộ nhớ từ một hoặc nhiều thiết bị ngoại vi. Mỗi streams có thể có tổng cộng tối đa 8 kênh (yêu cầu). Mỗi streams trong số 8 streams của 1 bộ điều khiển DMA cung cấp một liên kết truyền một chiều giữa một nguồn và một đích đến.

2. UART

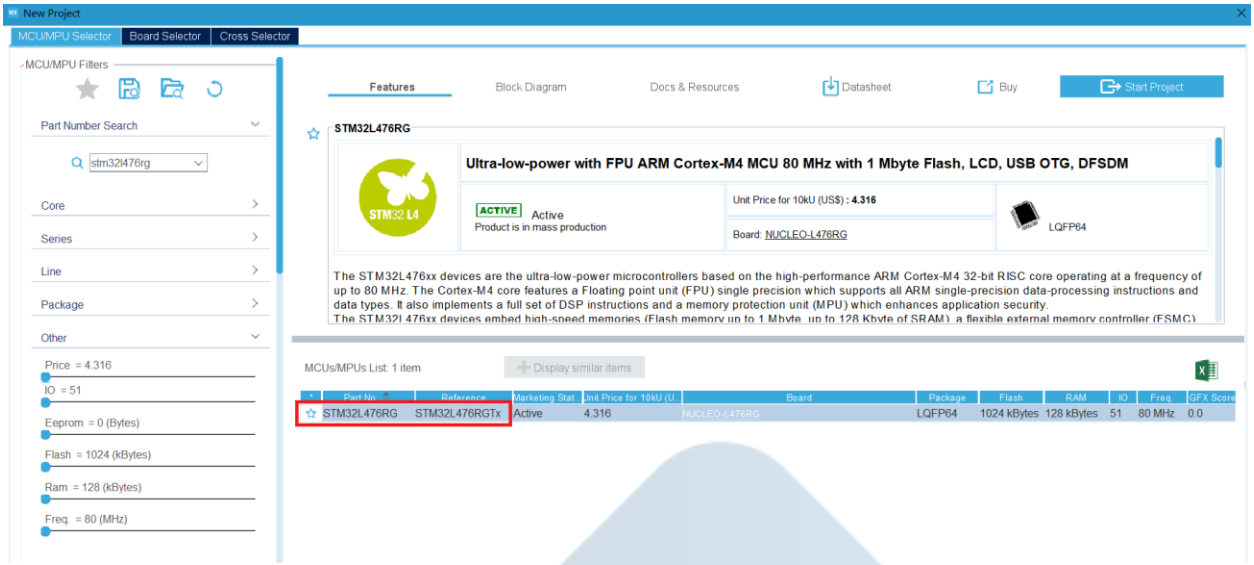
UART chuẩn giao tiếp được sử dụng phổ biến trong lập trình nhúng . Ưu điểm của nó là sự đơn giản trong cách sử dụng, tuy nhiên chuẩn giao tiếp này có tốc độ khá thấp (115200bit/s)

II, Lập trình

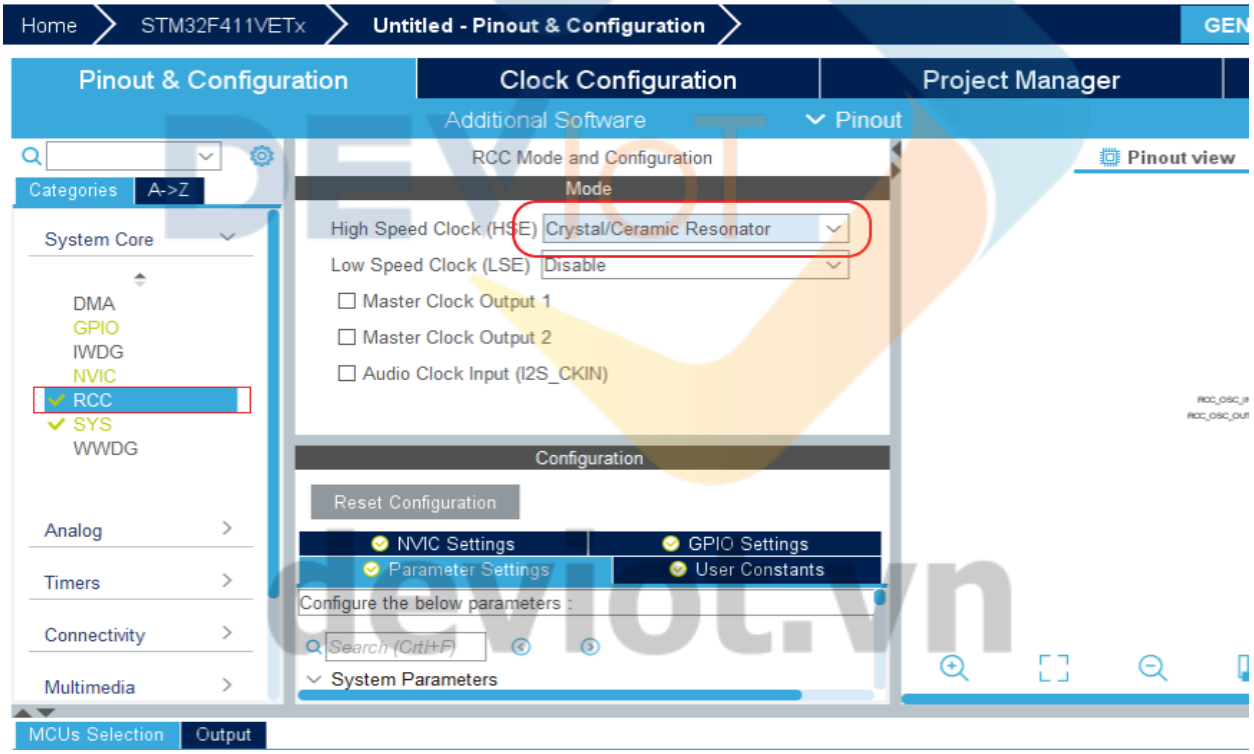
Ý tưởng Demo: Trong bài này để chứng minh chức năng của DMA, mình sẽ sử dụng bộ USART2 giao tiếp với máy tính qua 1 USBtoCOM nhé. Trên máy tính mình sẽ sử dụng phần mềm Hercules. Mình truyền dữ liệu từ Hercules xuống USART2 trên KIT. DMA sẽ nhận dữ liệu và truyền vào 1 biến mình khai báo sẵn. Trong main loop mình sẽ truyền ngược lại biến này lên máy tính và hiển thị trên Hercules. Đơn giản vậy thui, nhưng mình sẽ không sử dụng ngắt nhận của USART2 nhé.

Mở phần mềm STMCubeMX, chọn dòng chip bạn sử dụng. Ở đây mình chọn chip STM32L476RGT nhé.

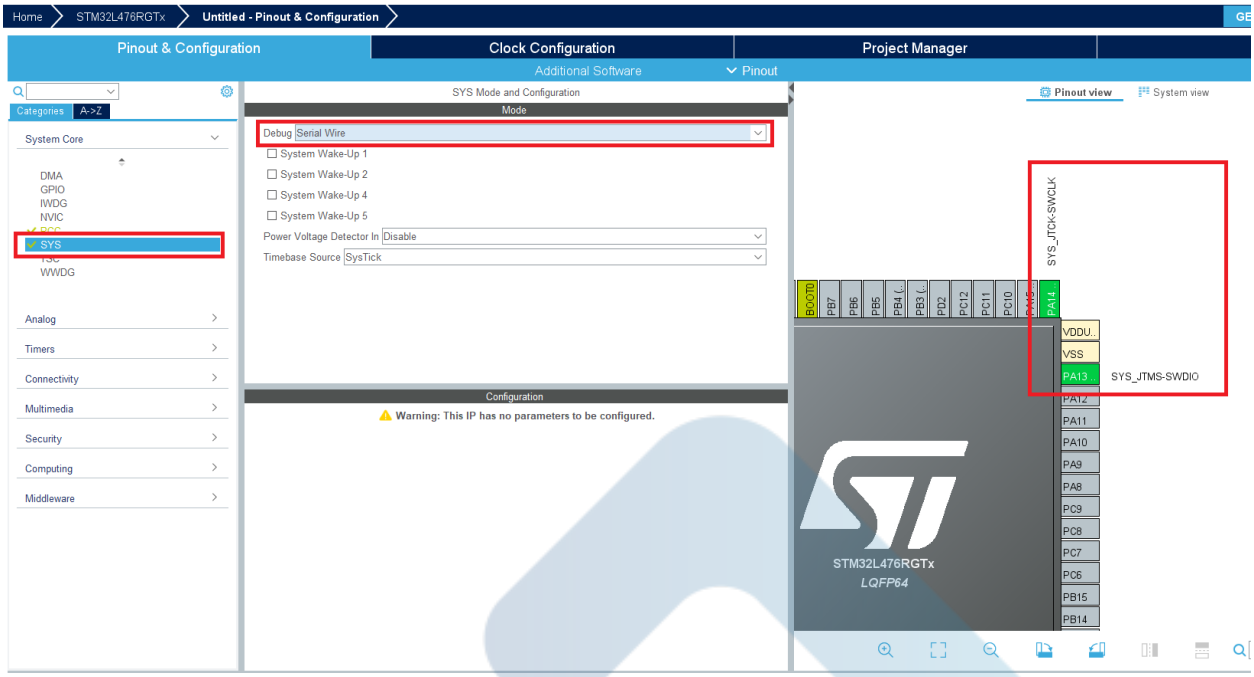
Đối với các dòng chip STM32 đời 4, tất cả mọi câu lệnh khi sử dụng thư viện HAL đều giống nhau. Chỉ khác nhau phần cấu hình Clock phụ thuộc riêng vào mỗi Chip.



Sau đó cấu hình Chip sử dụng thạch anh ngoài hàn sẵn trên Board mạch.



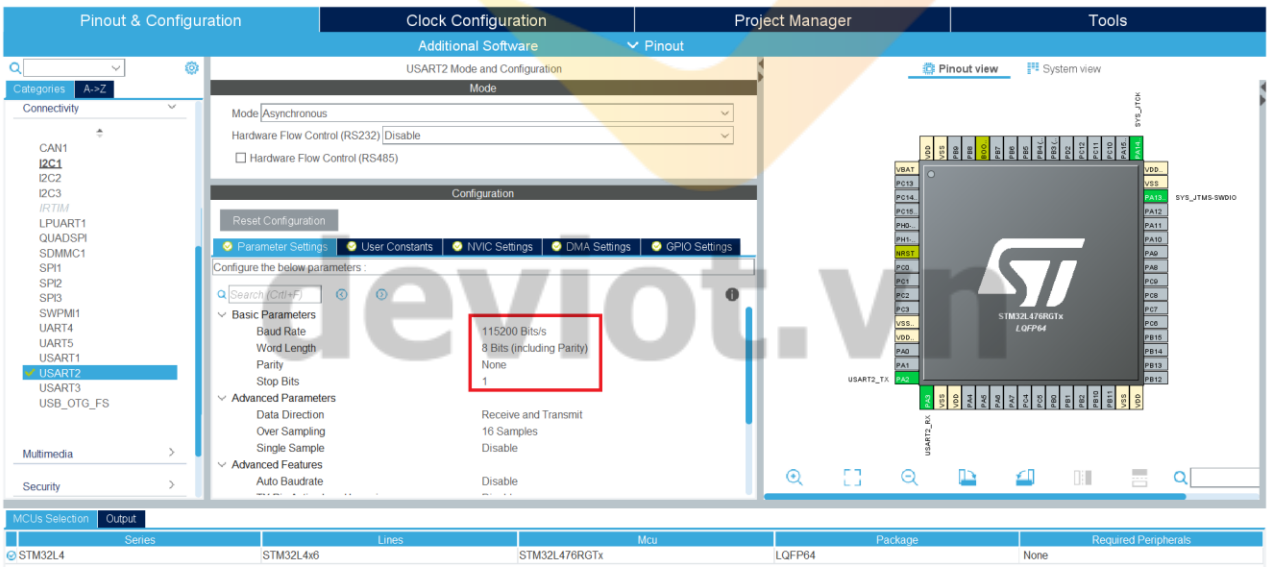
Cấu hình Chip Debug bằng mode SWD.



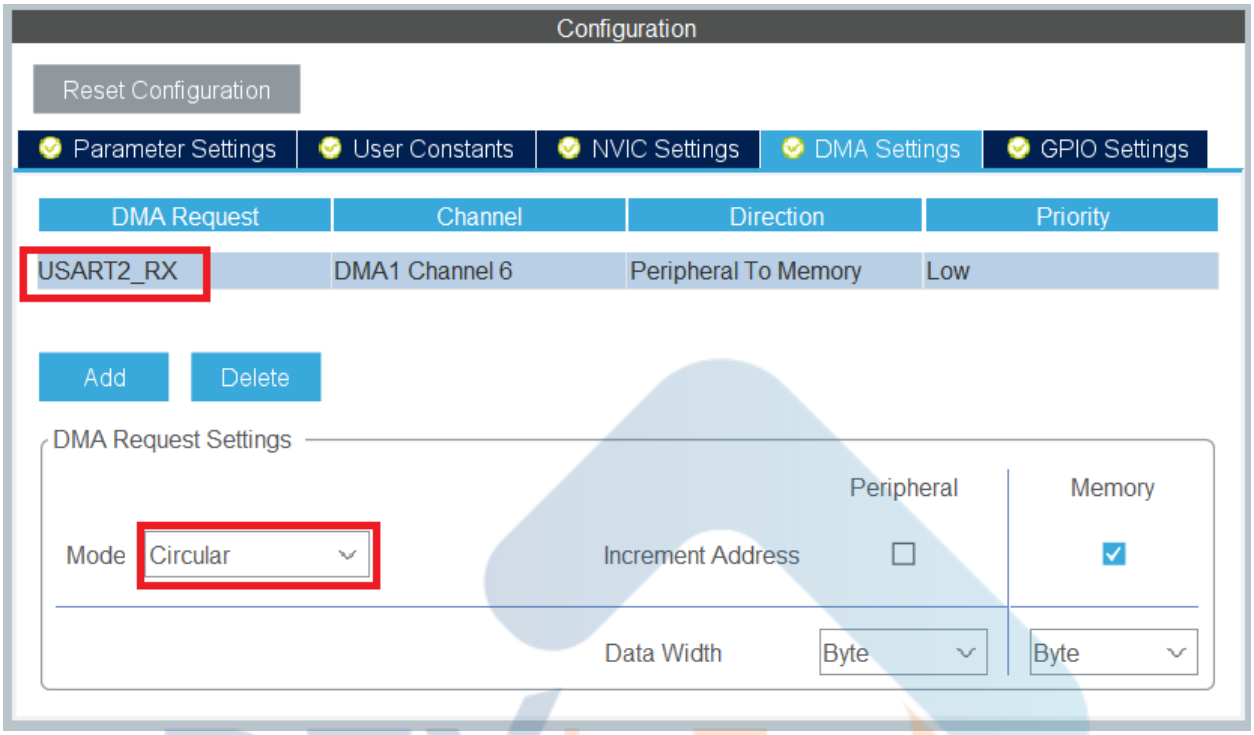
Ở đây mình sẽ khai báo bộ USART2 để làm việc với DMA nhé.

Baud Rate là gì?

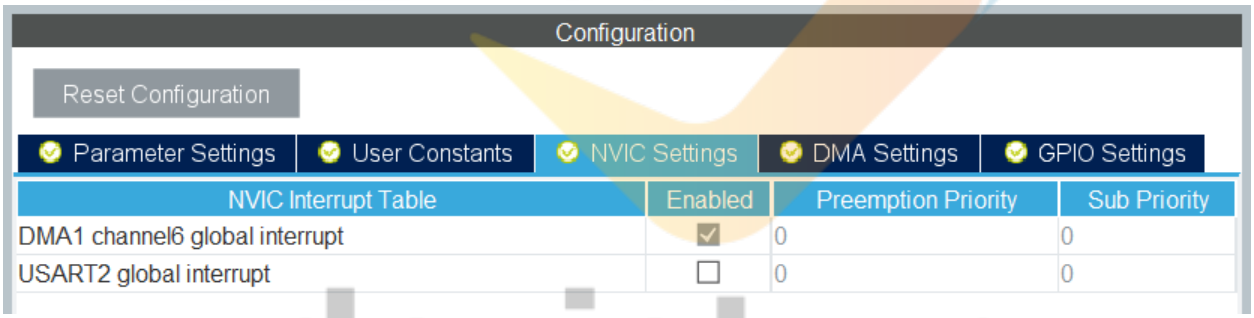
Tại sao lại là Baud Rate là 115200 mà không phải giá trị khác?



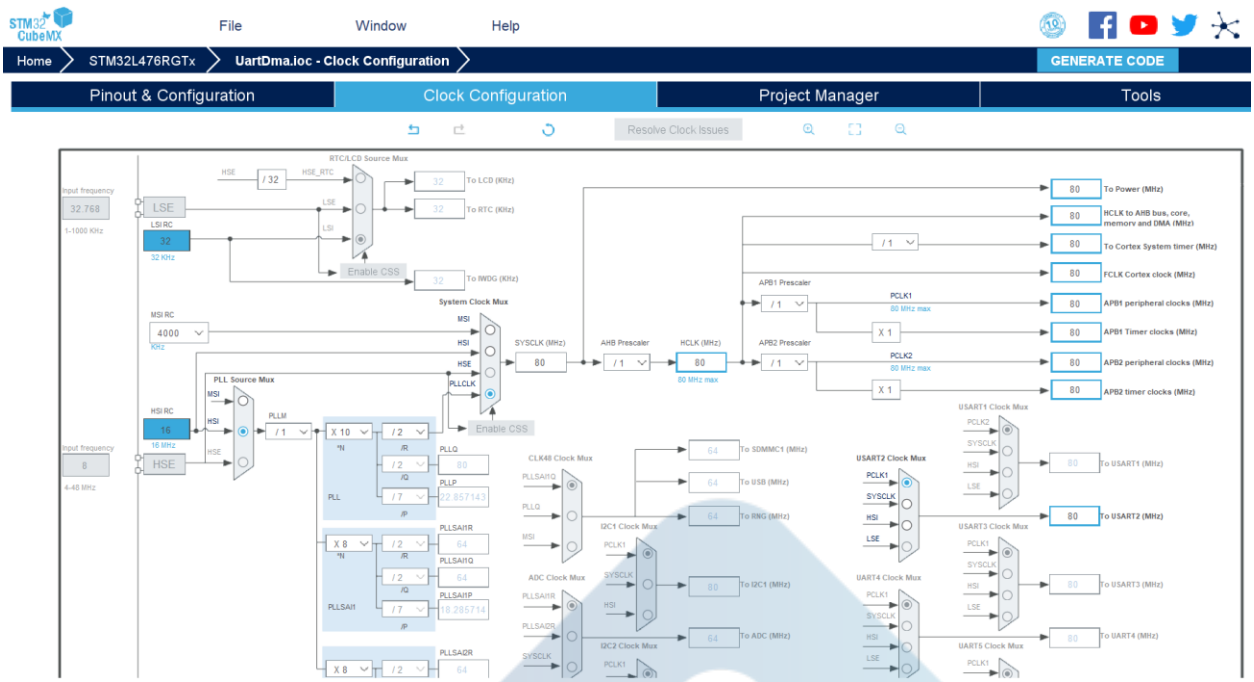
Và mình sẽ không Enable ngắt nhận của USART2 nhé, vì mình dùng DMA để nhận dữ liệu từ ngoại vi và truyền thẳng vào biến khai báo mà.



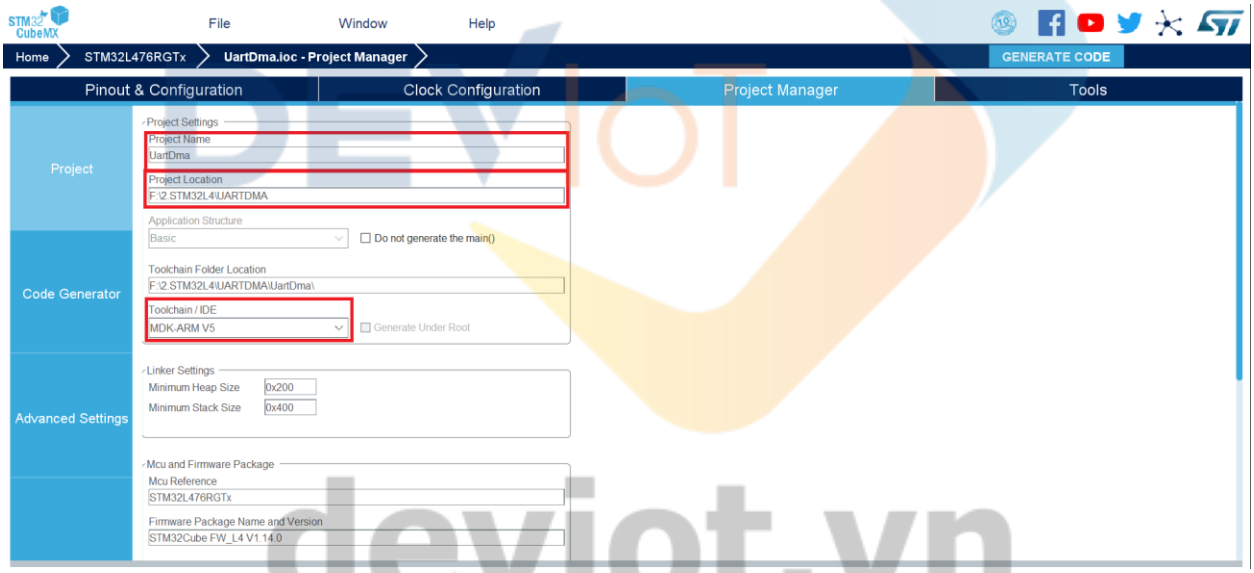
Enable ngắt cho DMA lên nhé các bạn.



Cấu hình Clock hoạt động . Ở đây mình chọn Clock nội (HSI) nhé. Trên KIT của mình không có hàn thạch anh ngoại.



Cuối cùng chọn File và sinh code cho Project.



Chú ý chọn **Copy only the necessary library files** để CubeMX chỉ copy những thư viện cần thiết cho Project.

Nào bắt đầu vào code.

Hàm khai báo USART2 do CubeMX tạo ra.

```
static void MX_USART2_UART_Init(void)
{


  huart2.Instance = USART2;
  huart2.Init.BaudRate = 115200;
  huart2.Init.WordLength = UART_WORDLENGTH_8B;
  huart2.Init.StopBits = UART_STOPBITS_1;
  huart2.Init.Parity = UART_PARITY_NONE;
  huart2.Init.Mode = UART_MODE_TX_RX;
  huart2.Init.HwFlowCtl = UART_HWCONTROL_NONE;
  huart2.Init.OverSampling = UART_OVERSAMPLING_16;
  if (HAL_UART_Init(&huart2) != HAL_OK)
  {
    Error_Handler();
  }
}
```

Trong main.c ta sẽ khởi tạo các module cần dùng.

```
  HAL_Init();
  SystemClock_Config();
  MX_GPIO_Init();
  MX_DMA_Init();
  MX_USART2_UART_Init();
  HAL_UART_Receive_DMA(&huart2,&receive_data,1);
```

Tại thư viên stm32f4xx_hal_dma.c chọn hàm `HAL_UART_Receive_DMA()` trong đó

```
// *huart là con trỏ trỏ tới địa chỉ uart nhận dữ liệu &huart
// *pData là con trỏ trỏ tới địa chỉ đầu tiên của mảng data nhận &receive_data
// Size là số lượng byte cần truyền đi. Ở đây là 1
```

Trong vòng lặp while(1), ta kiểm tra nếu DMA nhận được dữ liệu và truyền vào biến *receiver_data* ta sẽ truyền ngược lại lên máy tính để hiển thị.

```
while (1)
  {
      if(receive_data!= 0)
      {
            HAL_UART_Transmit(&huart2,&receive_data,1,1000);
            receive_data = 0;
      }
```

```
    }

}
```

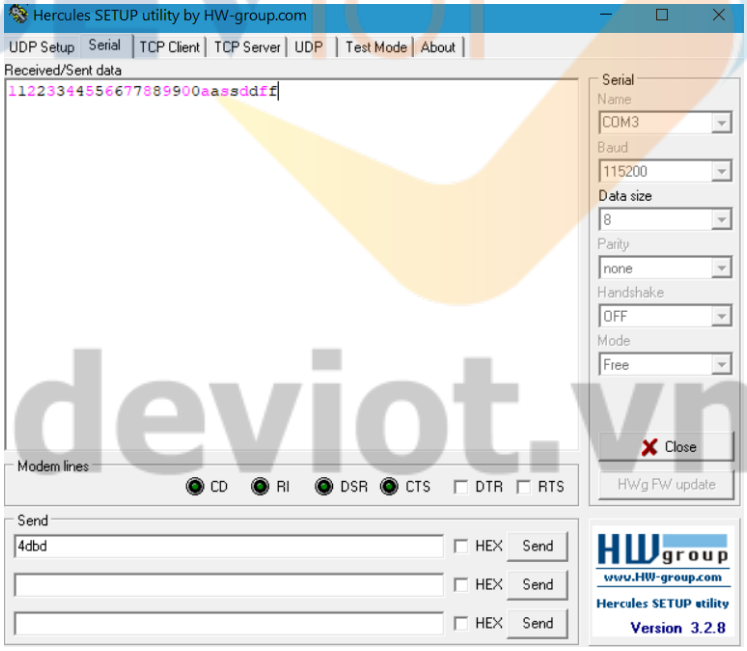
Biên dịch và Debug

Tiến hành Build Code và Nạp chương trình xuống KIT

Kết quả build không có lỗi

```
*** Using Compiler 'V5.06 update 6 (build 750)', folder: 'C:\Keil_v5\ARM\ARMCC\Bin'
Build target 'UART_useDMA'
compiling main.c...
linking...
Program Size: Code=5604 RO-data=472 RW-data=20 ZI-data=1348
FromELF: creating hex file...
"UART_useDMA\UART_useDMA.axf" - 0 Error(s), 0 Warning(s).
Build Time Elapsed:  00:00:03
```

Kết quả hiển thị trên Hercules



DEVIOT - CÙNG NHAU HỌC LẬP TRÌNH IOT

📌 Website: deviot.vn

📌 FanPage: Deviot - Thời sự kỹ thuật & IoT

📌 Group: Deviot - Cùng nhau học lập trình IOT

📌 Hotline: 0969.666.522

📌 Address: Số 101C, Xã Đàn 2

📌 Đào tạo thật, học thật, làm thật