



DesignNews

IoT Designs Using STmicro Microcontrollers

Day 1: Tool Up!

Sponsored by

DigiKey



Webinar Logistics

- Turn on your system sound to hear the streaming presentation.
- If you have technical problems, click “Help” or submit a question asking for assistance.
- Participate in ‘Attendee Chat’ by maximizing the chat widget in your dock.

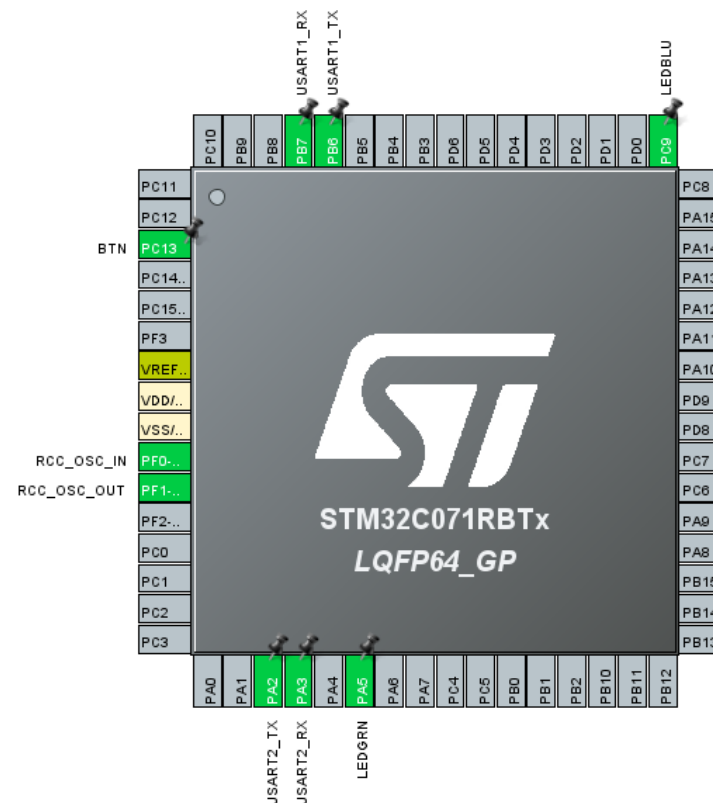


Fred Eady

Visit 'Lecturer Profile' in your console for more details.

AGENDA

- **STM32 Development Using STM32CubeIDE**
 - Install STM32CubeIDE
 - Create a NUCLEO-C071RB Project
 - Debug Our NUCLEO-C071RB Project



Download and Install STM32CubeIDE

Get Software

	Part Number	General Description	Latest version	Download	All versions
+	STM32CubeIDE-DEB	STM32CubeIDE Debian Linux Installer	1.16.1	Get latest	Select version
+	STM32CubeIDE-Lnx	STM32CubeIDE Generic Linux Installer	1.16.1	Get latest	Select version
+	STM32CubeIDE-Mac	STM32CubeIDE macOS Installer	1.16.1	Get latest	Select version
+	STM32CubeIDE-RPM	STM32CubeIDE RPM Linux Installer	1.16.1	Get latest	Select version
+	STM32CubeIDE-Win	STM32CubeIDE Windows Installer	1.16.1	Get latest	Select version

https://www.st.com/en/development-tools/stm32cubeide.html

Create a NUCLEO-C071RB Project

Project Explorer

There are no projects in your workspace.
To add a project:

Create a new Makefile project in a directory containing existing code

Create a new C or C++ project

Create a New STM32 project

Create a New STM32 Project from an Existing STM32CubeMX Configuration File (.ioc)

Creates a new STM32 CMake Project

Create a project...

Import projects...

workspace_1.16.1 - STM32CubeIDE

File Edit Source Refactor Navigate Search Project Run Window Help

Project Explorer

There are no projects in your workspace.
To add a project:

Create a new Makefile project in a directory containing existing code

Create a new C or C++ project

Create a New STM32 project

Create a New STM32 Project from an Existing STM32CubeMX Configuration File (.ioc)

Creates a new STM32 CMake Project

Create a project...

Import projects...

Problems

0 items

Description	Resource	Path	Location	Type

Build Analyzer

Static Stack Analyzer Cyclomatic Complexity

Memory Regions

Memory Details

Region	Start address	End address	Size	Free	Used	Usage (%)

0 items selected

Create a NUCLEO-C071RB Project – Select the Board

IDE STM32 Project

Target Selection

Select STM32 target or STM32Cube example

MCU/MPU Selector

Board Selector

Example Selector

Board Filters

Commercial Part Number

NUCLEO-C071RB

PRODUCT INFO

IDE STM32 Project

Target Selection

Select STM32 target or STM32Cube example

MCU/MPU Selector

Board Selector

Example Selector

Cross Selector

Board Filters

Commercial Part Number

NUCLEO-C071RB

PRODUCT INFO

Type

Supplier

MCU / MPU Series

Marketing Status

Price

MEMORY

Ext. Flash = 0 (MBit)

Ext. EEPROM = 0 (kBytes)

Ext. RAM = 0 (MBit)

FEATURES

Embedded Sensor

User Button

Camera

CAN

Connector

Features

Large Picture

Docs & Resources

Datasheet

Buy

STM32C0 Series

NUCLEO-C071RB

ACTIVE

Product is in mass production

STM32 Nucleo-64 development board with STM32C071RB MCU, supports Arduino and ST morpho connectivity

Part Number : NUCLEO-C071RB

Commercial Part Number : NUCLEO-C071RB

Unit Price (US\$) : 10.4

Mounted Device : STM32C071RBT6

The STM32 Nucleo-64 board provides an affordable and flexible way for users to try out new concepts and build prototypes by choosing from the various combinations of performance and power consumption features provided by the STM32 microcontroller. For the compatible boards, the internal or external SMPS significantly reduces power consumption in Run mode.

The ARDUINO® Uno V3 connectivity support and the ST morpho headers allow the easy expansion of the functionality of the STM32 Nucleo open development platform with a wide choice of specialized shields.

The STM32 Nucleo-64 board does not require any separate probe as it integrates the ST-LINK debugger/programmer.

The STM32 Nucleo-64 board comes with the STM32 comprehensive free software libraries and examples available with the STM32Cube MCU Package.

Boards List: 1 item

Export

	Overview	Commercial Part No	Type	Marketing Status	Unit Price (US\$)	Mounted Device
★		NUCLEO-C071RB	Nucleo-64	Active	10.4	STM32C071RBT6

< Back

Next >

Finish

Cancel

7

Create a NUCLEO-C071RB Project – Name the Project

Setup STM32 project

IDE

Project

Project Name:

☒ Use default location

Location:

Options

Targeted Language

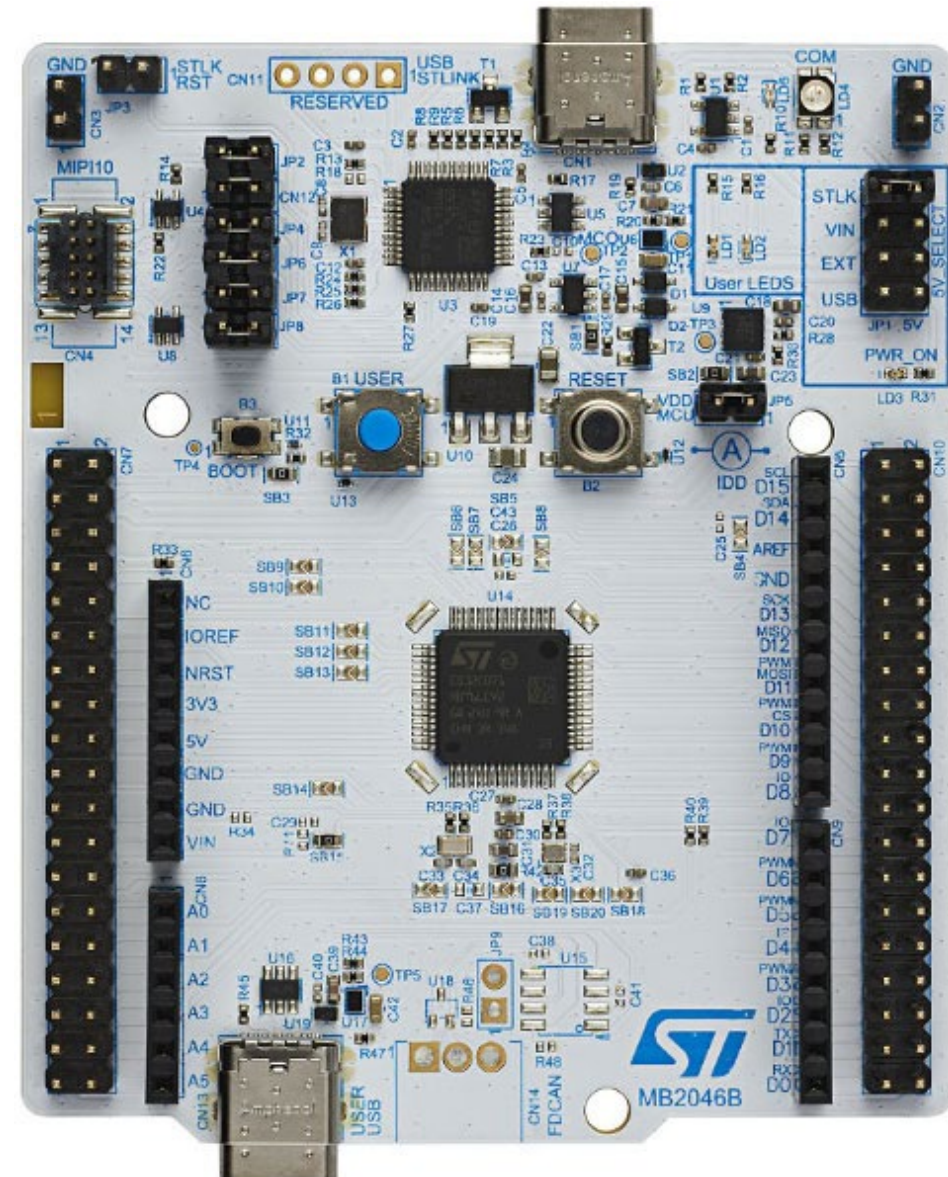
☒ C ☐ C++

Targeted Binary Type

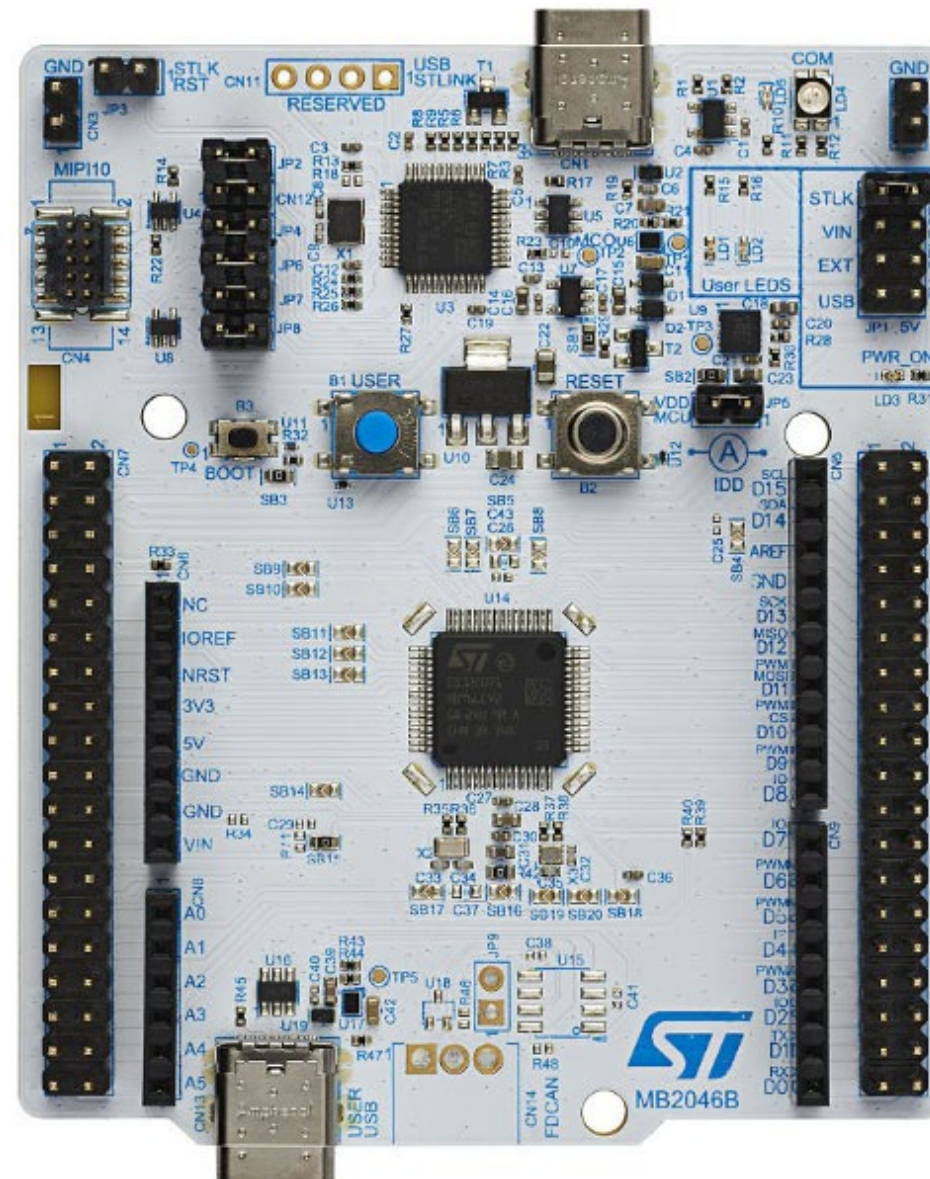
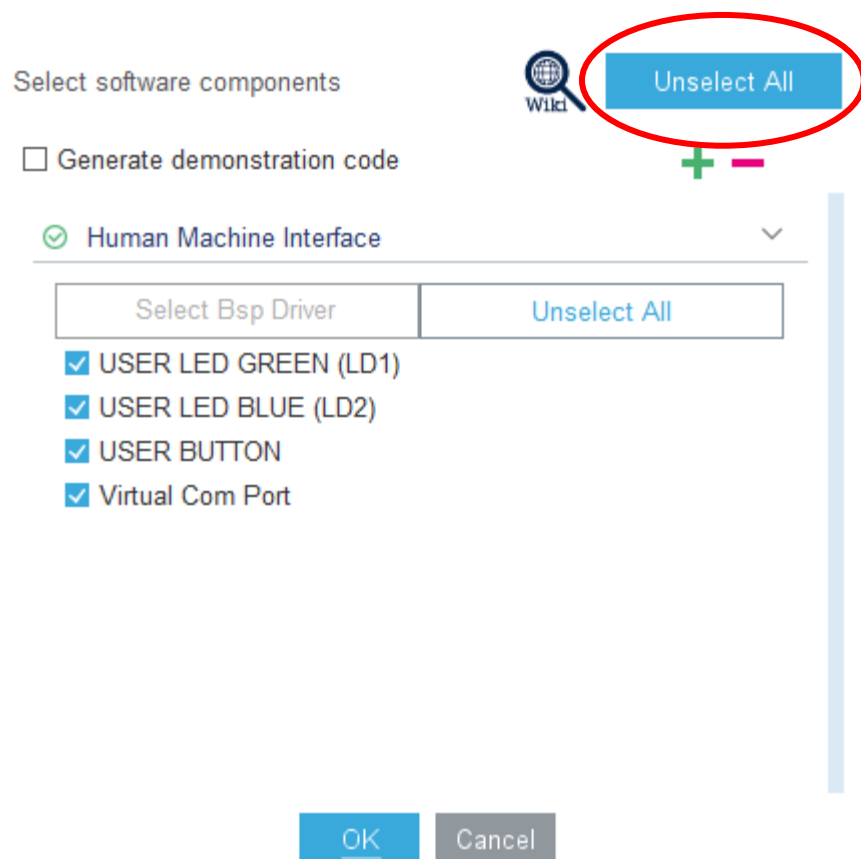
☒ Executable ☐ Static Library

Targeted Project Type

☒ STM32Cube ☐ Empty



Create a NUCLEO-C071RB Project – Kill the BSP



Create a NUCLEO-C071RB Project – Configure the Hardware

The screenshot shows the STM32CubeIDE interface with the 'Pinout & Configuration' window open. The window is divided into several sections:

- Project Explorer:** Shows the project structure for 'c071rb_uart_ringbuffer.ioc', including 'Includes', 'Core', 'Inc', 'Src', 'Startup', and 'Drivers'.
- Pinout & Configuration:** The main configuration area, currently showing the 'Pinout' tab. It features a search bar and a 'Categories' dropdown set to 'A->Z'. The categories list includes System Core, Analog, Timers, Connectivity, Multimedia, Computing, Middleware and Software Pac..., Trace and Debug, Power and Thermal, Bsp, and Other. Red arrows point to 'System Core' and 'Connectivity'.
- Clock Configuration:** A tab for configuring the system clock.
- Project Manager:** A tab for managing the project settings.
- Tools:** A tab for accessing various development tools.
- Pinout View:** A detailed view of the STM32C071RBTx LQFP64_GP microcontroller package. It shows the pin numbers (PC10 to PC3, PA0 to PA15, PB0 to PB15) and their functions (GPIO_Input, USART2_TX, USART2_RX, GPIO_Output, User_LED2). The ST logo and part number are prominently displayed.

The bottom of the window shows a search bar and a page number '10'.

Create a NUCLEO-C071RB Project – Enable the HSE Clock

The screenshot displays the STM32CubeIDE interface for a project named "c071rb_uart_ringbuffer.ioc". The "Pinout & Configuration" window is open, showing the "RCC Mode and Configuration" settings. The "High Speed Clock (HSE)" is set to "Crystal/Ceramic Resonator", and the "LSCO Clock Output" is enabled. A red arrow points to the "High Speed Clock (HSE)" dropdown menu. The left pane shows the "System Core" configuration with "RCC" selected. The background shows a pinout diagram of the STM32C071RBTx LQFP64_GP package.

Pinout & Configuration

Categories A->Z

System Core

- CORTEX_M0+
- DMA
- FLASH
- GPIO
- IWDG
- NVIC
- RCC**
- SYS
- WWDG

RCC Mode and Configuration

Mode

- High Speed Clock (HSE) Crystal/Ceramic Resonator
- ☐ OSC enable
- Low Speed Clock (LSE) Disable
- ☐ OSC32 enable
- ☐ Master Clock Output
- ☐ Master Clock Output 2
- ☒ LSCO Clock Output
- ☐ Audio Clock Input (I2S_CKIN)
- CRS SYNC Disable

STM32C071RBTx LQFP64_GP

Pinout diagram showing the package pins and their connections to the microcontroller.

Create a NUCLEO-C071RB Project – Configure the LEDs

Categories

A->Z

System Core

CORTEX_M0+

DMA

FLASH

GPIO

IWDG

NVIC

RCC

SYS

WWDG

Configuration

Group By Peripherals

GPIO

RCC

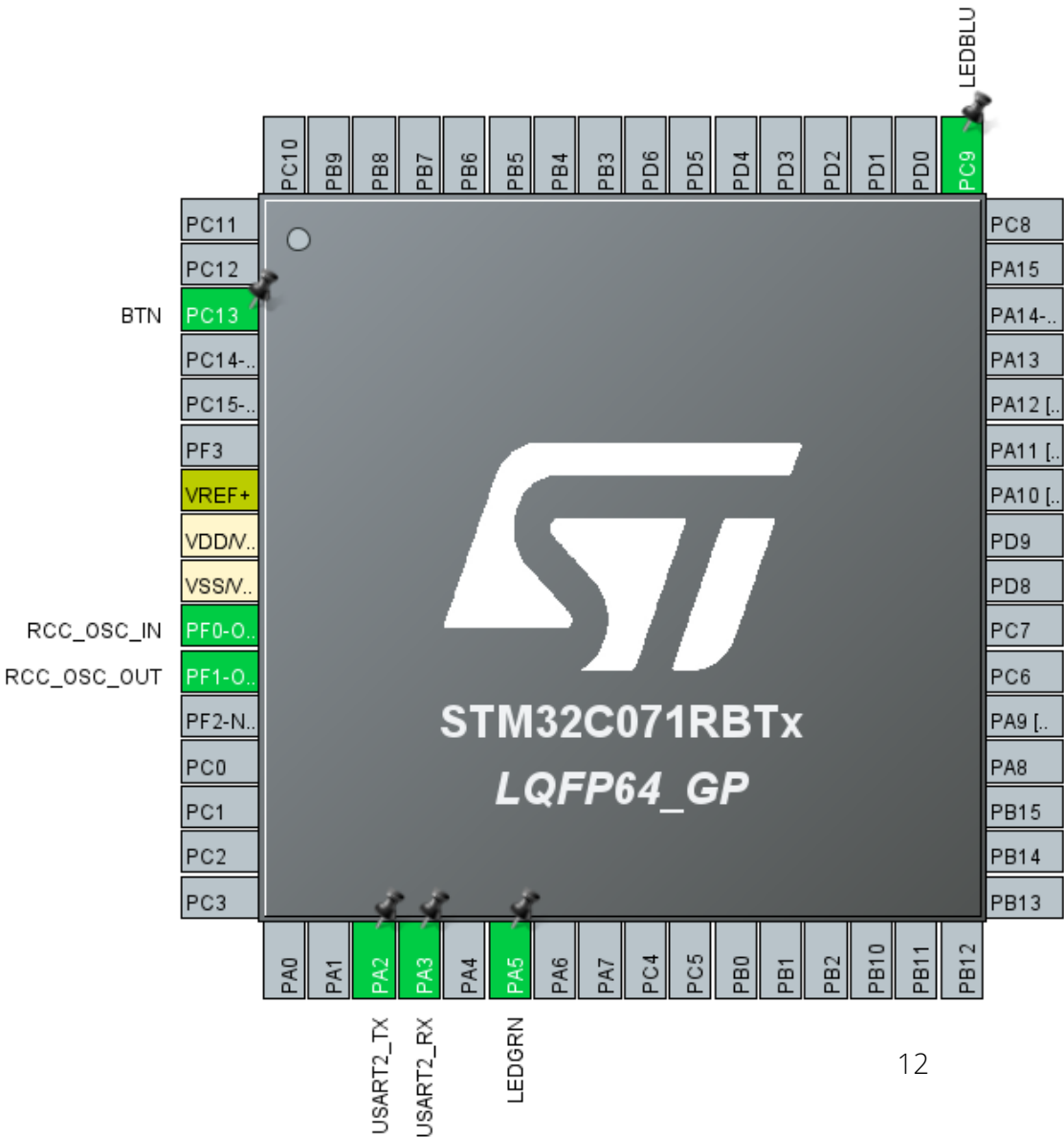
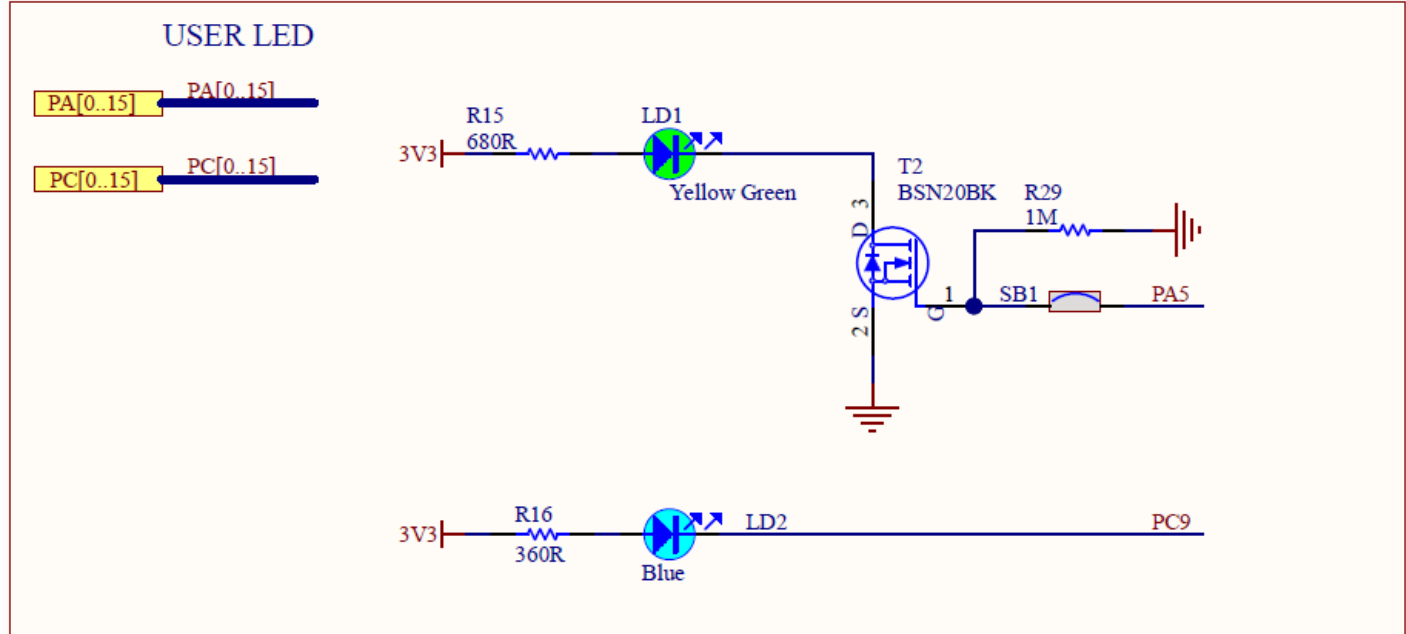
USART

Search Signals

Search (Ctrl+F)

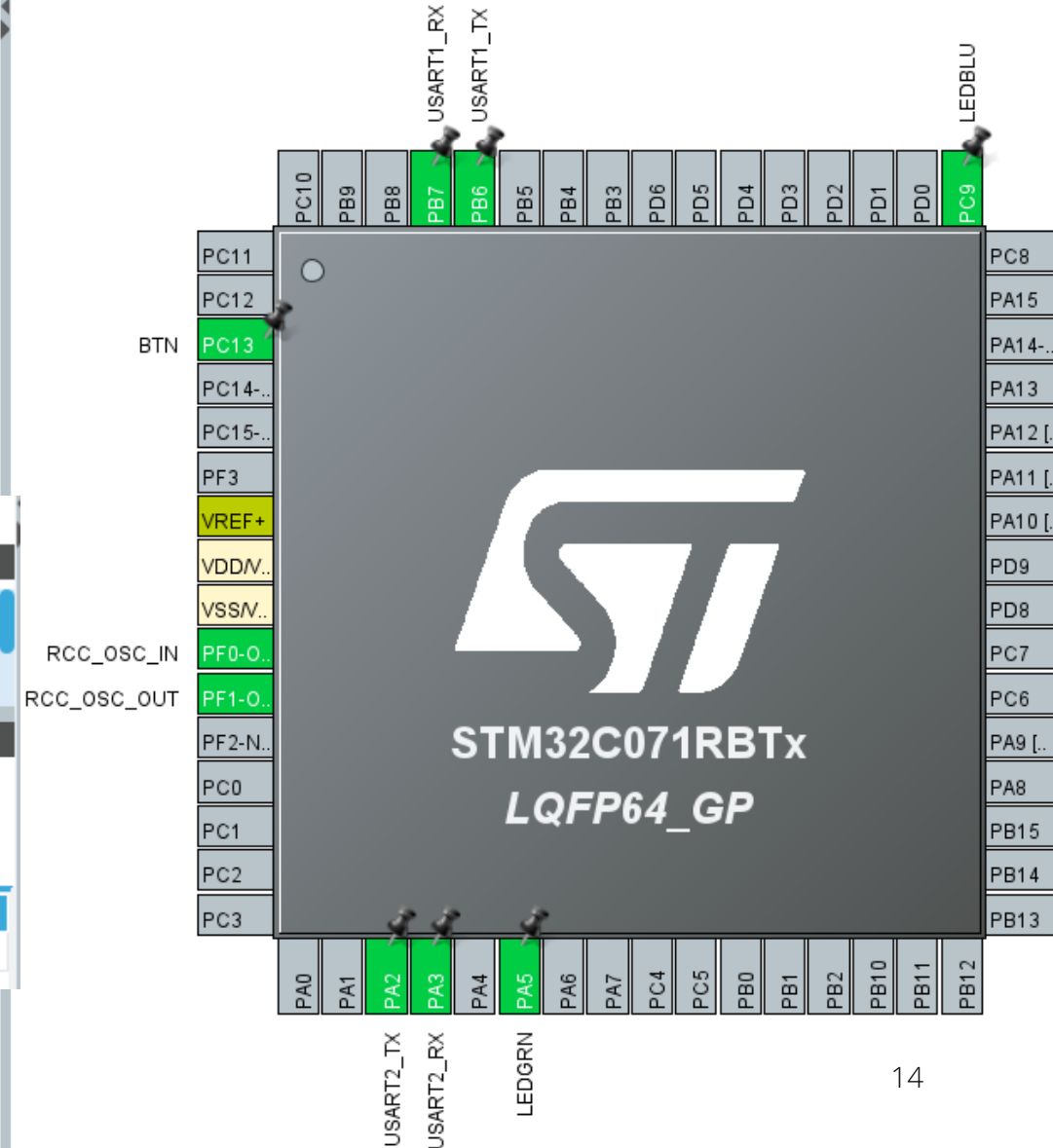
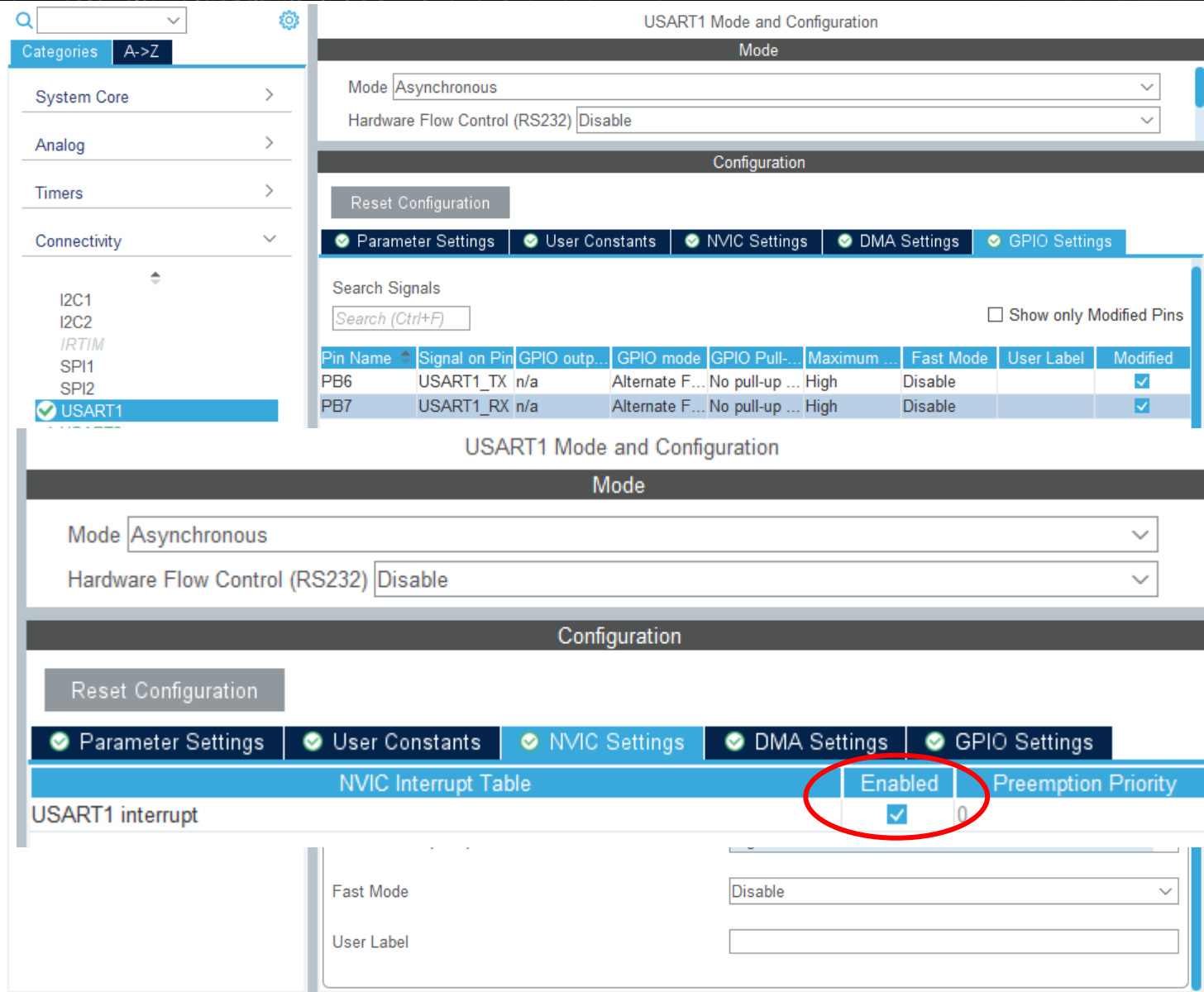
☐ Show only Modified Pins

Pin ...	Signal ...	GPIO o...	GPIO ...	GPIO ...	Maxim...	Fast M...	User L...	Modified
PA5	n/a	Low	Output...	No pull...	Low	n/a	LEDGRN	☒
PC9	n/a	Low	Output...	No pull...	Low	n/a	LEDBLU	☒
PC13	n/a	n/a	Input ...	No pull...	n/a	n/a	BTN	☒

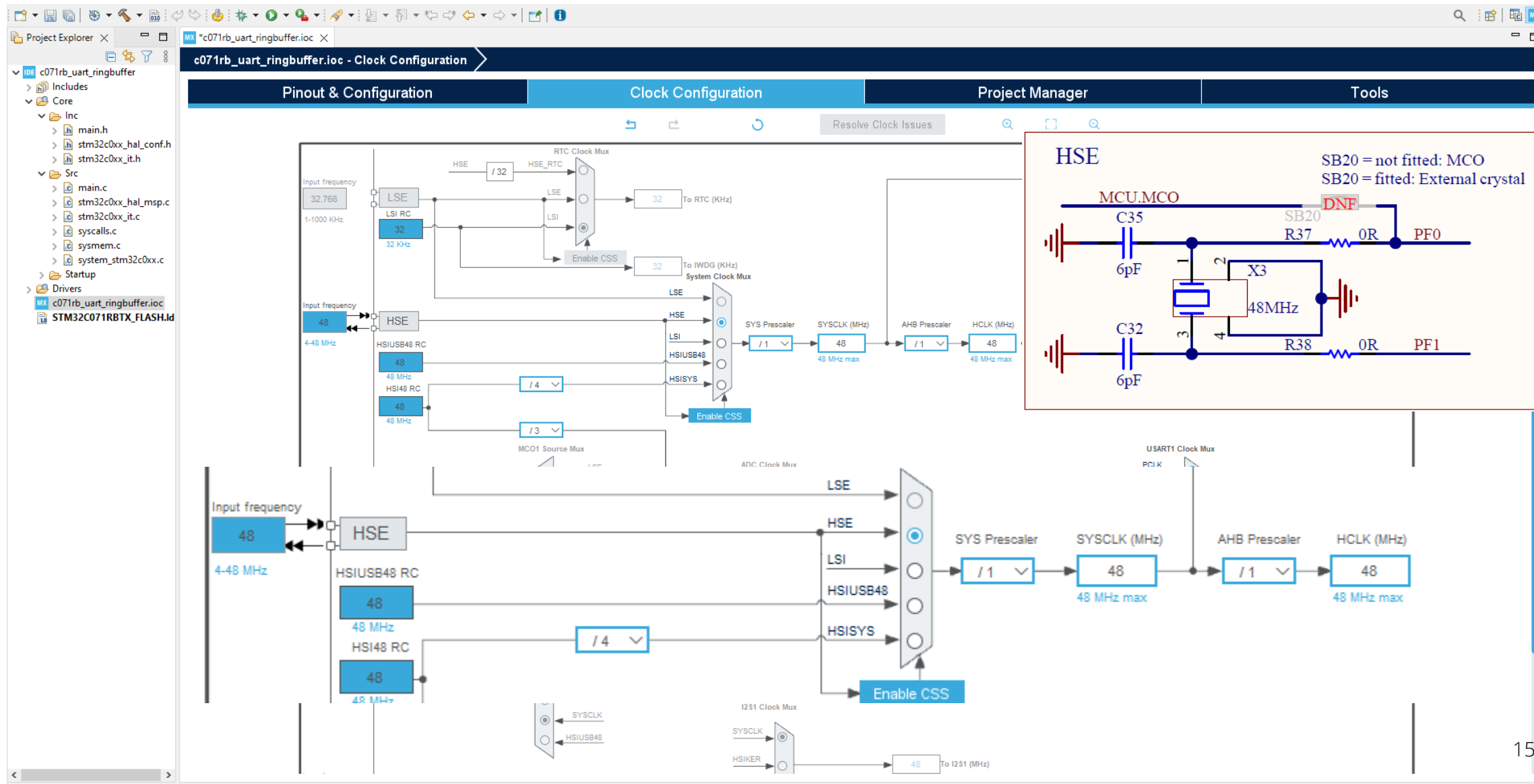


Create a NUCLEO-C071RB Project – Configure USART1

Create a NUCLEO-C071RB Project – Enable USART1 Interrupt



Create a NUCLEO-C071RB Project – Configure the HSE Clock



Create a NUCLEO-C071RB Project – Save and Generate

The screenshot displays the STM32CubeIDE Project Manager window for a project named 'c071rb_uart_ringbuffer.ioc'. The 'Project Manager' tab is selected and circled in red. A red arrow points to the '2 - Save and Generate Code' button in the top toolbar. The 'Code Generator' section is expanded, showing options for generating peripheral initialization files. A red arrow labeled '1' points to the 'Generate peripheral initialization as a pair of '.c/.h' files per peripheral' checkbox, which is checked. The 'Advanced Settings' section is also visible, showing HAL and User Actions settings.

2 - Save and Generate Code

Project Manager

Code Generator

STM32Cube MCU packages and embedded software packs

- ☐ Copy all used libraries into the project folder
- ☒ Copy only the necessary library files
- ☐ Add necessary library files as reference in the toolchain project configuration file

Generated files

- ☒ Generate peripheral initialization as a pair of '.c/.h' files per peripheral
- ☐ Backup previously generated files when re-generating
- ☒ Keep User Code when re-generating
- ☒ Delete previously generated files when not re-generated

HAL Settings

- ☐ Set all free pins as analog (to optimize the power consumption)
- ☐ Enable Full Assert

User Actions

Before Code Generation

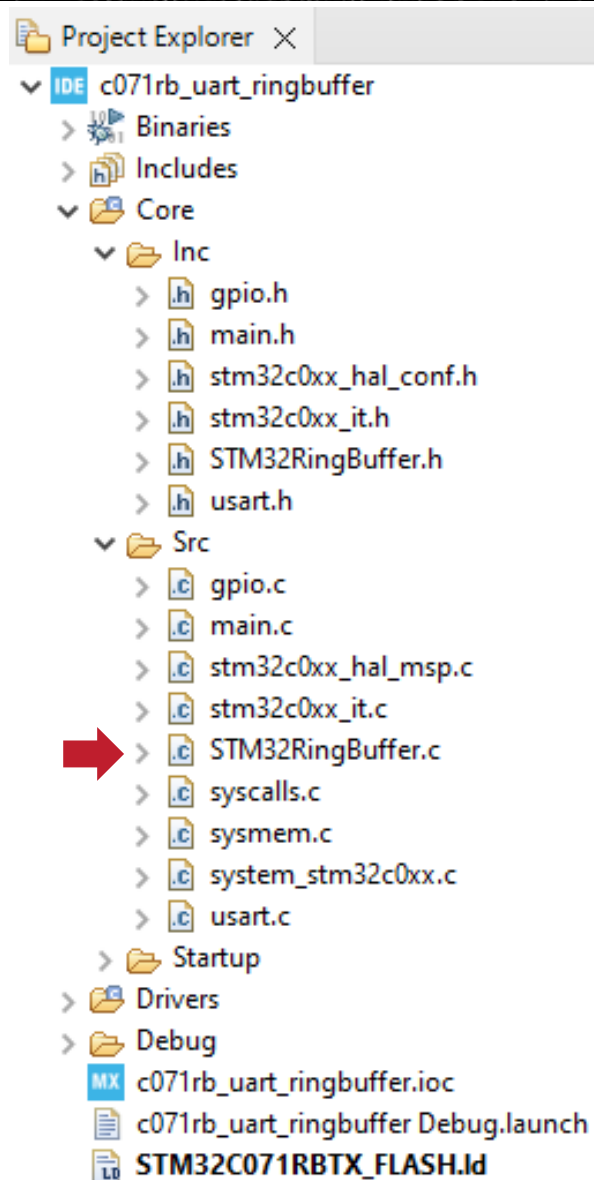
After Code Generation

Template Settings

Select a template to generate customized code

1

Create a NUCLEO-C071RB Project – STM32RingBuffer.c



```

/*****
/* STM32RingBuffer.c
/* Adapted by: Fred Eady
/* Last Update: 10-15-2024
/* Notes:
/* Add the following code to the stm32xxxx_it.c file:
/*
/* Add this line to the Private Function Prototypes
/* extern void ringbuffer_IRQHandler(UART_HandleTypeDef *huart);
/* Add this line to the External Variables where x is huart number
/* extern UART_HandleTypeDef huartx;
/*
/* In void USART1_IRQHandler(void);
/* Comment out HAL_UART_IRQHandler(&huartx); where x is huart number
/* You must comment out HAL_UART_IRQHandler(&huartx); every time you recompile
/* Add this line inside HAL_UART_IRQHandler(&huartx);
/* ringbuffer_IRQHandler(&huart1);
*****/

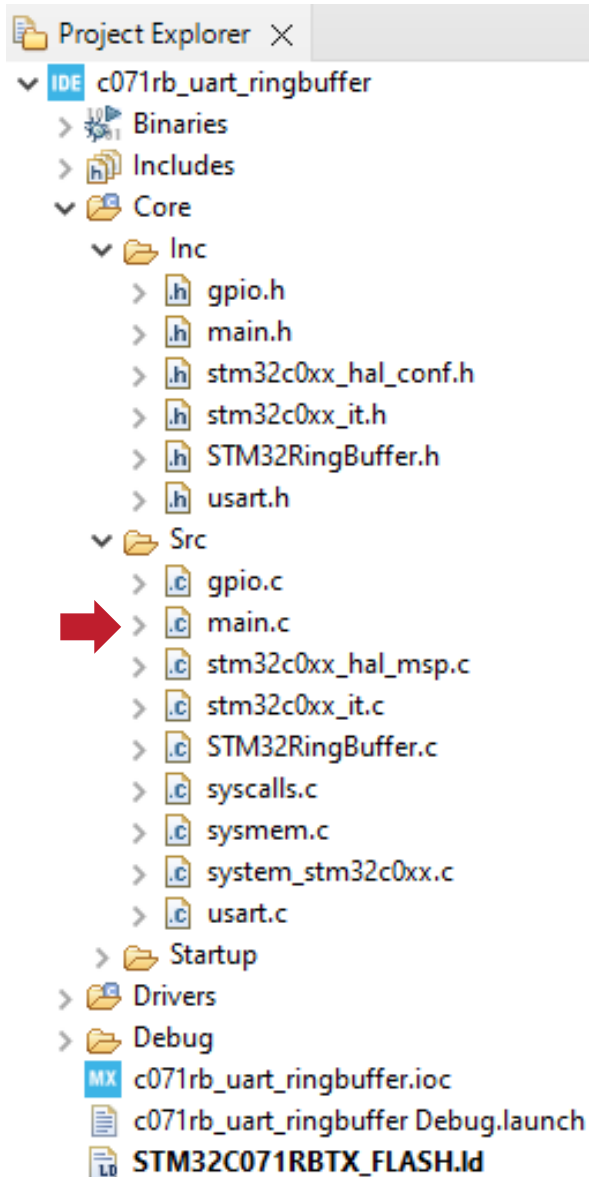
#include <string.h>
#include <stdint.h>
#include <STM32RingBuffer.h>

// Function Prototypes
void writeRxRing(uint8_t bite, ringBuffer *buffer);

// Assign an alias for the huartx
extern UART_HandleTypeDef huart1;
#define uartAlias &huart1

```

Create a NUCLEO-C071RB Project – User Code

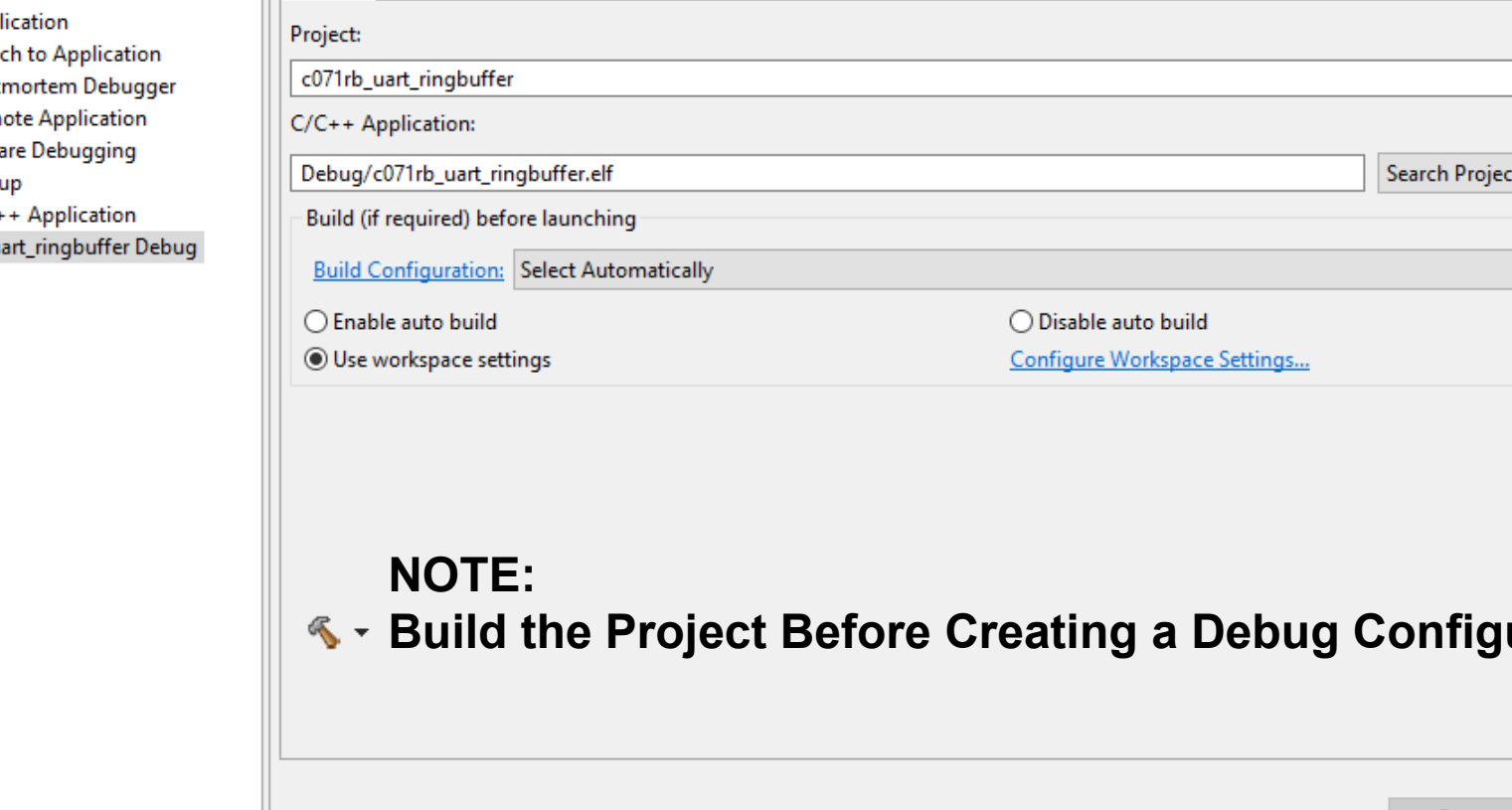


```
/* USER CODE BEGIN 2 */
RingBuffer_Init();
indx = 0;
HAL_UART_Transmit(&huart1, splashtext, sizeof(splashtext), 0xFFFFFFFF);
HAL_GPIO_WritePin(LEDGRN_GPIO_Port, LEDGRN_Pin, GPIO_PIN_RESET);
HAL_GPIO_WritePin(LEDBLU_GPIO_Port, LEDBLU_Pin, GPIO_PIN_SET);
/* USER CODE END 2 */

/* Infinite loop */
/* USER CODE BEGIN WHILE */
while (1)
{
    if(IsCharInQueue())
    {
        while(IsCharInQueue())
        {
            rxBite[indx++] = readRxRing();
        }
        switch(rxBite[indx - 1])
        {
            case '1':
                HAL_GPIO_WritePin(LEDBLU_GPIO_Port, LEDBLU_Pin, GPIO_PIN_RESET);
                HAL_UART_Transmit(&huart1, sendON ,sizeof(sendON), HAL_MAX_DELAY);
                break;
            case '0':
                HAL_GPIO_WritePin(LEDBLU_GPIO_Port, LEDBLU_Pin, GPIO_PIN_SET);
                HAL_UART_Transmit(&huart1, sendOFF ,sizeof(sendOFF), HAL_MAX_DELAY);
                break;
        }
    }
}
/* USER CODE END WHILE */
```

Debug Our NUCLEO-C071RB Project – Debug Config

Create, manage, and run configurations



The screenshot shows the STM32CubeIDE interface. On the left, the 'Project Explorer' pane displays a tree structure with 'c071rb_uart_ringbuffer Debug' selected. The main workspace area shows the 'Debug' configuration window for the project 'c071rb_uart_ringbuffer'. The configuration fields are as follows:

- Name: c071rb_uart_ringbuffer Debug
- Project: c071rb_uart_ringbuffer
- C/C++ Application: Debug/c071rb_uart_ringbuffer.elf
- Build (if required) before launching: Build Configuration: Select Automatically
- Enable auto build: ☐ Enable auto build, ☐ Disable auto build
- Use workspace settings: ☒ Use workspace settings

A large 'NOTE' is overlaid on the workspace area, stating: 'Build the Project Before Creating a Debug Configuration'.

Debug Our NUCLEO-C071RB Project – Debug Config

Create, manage, and run configurations



Filter matched 8 of 8 items

Name: c071rb_uart_ringbuffer Debug

Main Debugger Startup Source Common

GDB Connection Settings

☒ Autostart local GDB server Host name or IP address localhost

☐ Connect to remote GDB server Port number 61234

Debug probe ST-LINK (ST-LINK GDB server)

GDB Server Command Line Options

Show Command Line

Interface

☒ SWD ☐ JTAG

☒ ST-LINK S/N 0669FF505771664867042333 Scan

Frequency (kHz): Auto

Access port: 0 - Cortex-M0plus

Reset behaviour

Type: Connect under reset

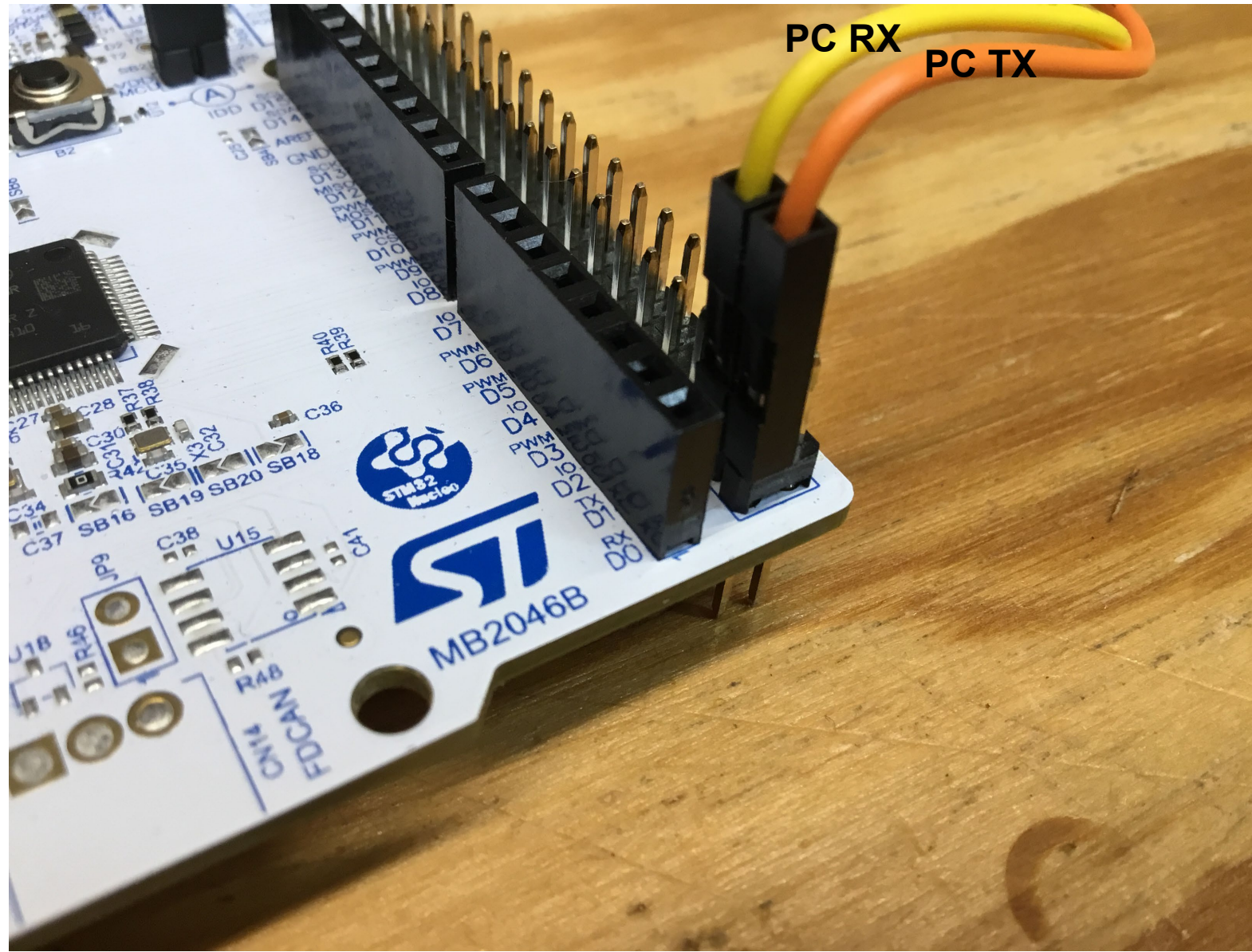
Device settings

Debug in low power modes: Enable

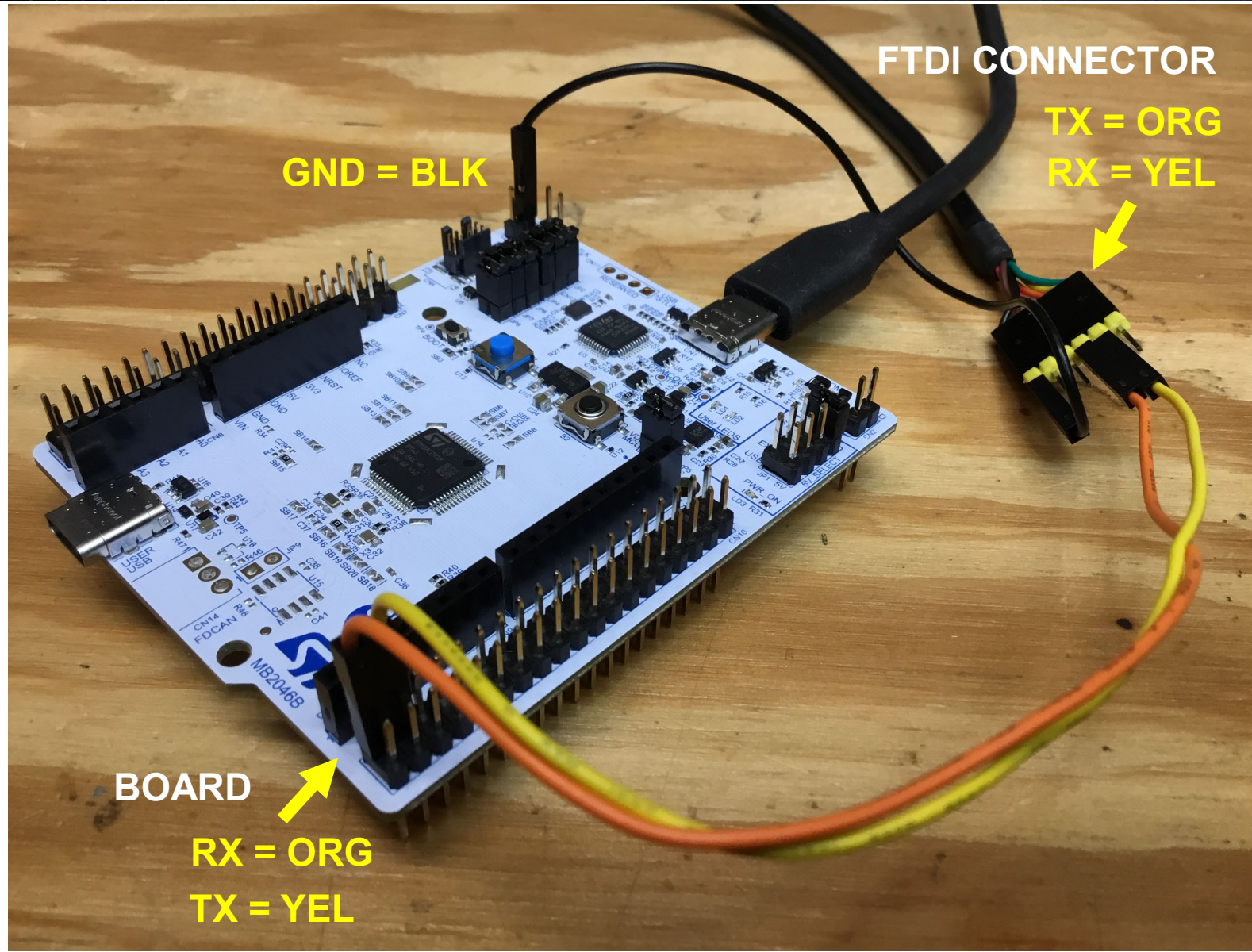
Revert Apply

Debug Close

Debug Our NUCLEO-C071RB Project – Connect USART1



Debug Our NUCLEO-C071RB Project – Connect USART1



Debug Our NUCLEO-C071RB Project – Debug It!

Hercules SETUP utility by HW-group.com

UDP Setup | **Serial** | TCP Client | TCP Server | UDP | Test Mode | About

Received/Sent data

```
Ringbuffer Initialized..{0D}{0A}{00}1ON{0D}{0A}0OFF{0D}
{0A}1ON{0D}{0A}0OFF{0D}{0A}1ON{0D}{0A}0OFF{0D}{0A}1ON{0D}
{0A}0OFF{0D}{0A}1ON{0D}{0A}0OFF{0D}{0A}1ON{0D}{0A}0OFF{0D}
{0A}1ON{0D}{0A}0OFF{0D}{0A}
```

Serial

Name: COM31

Baud: 115200

Data size: 8

Parity: none

Handshake: OFF

Mode: Free

Modem lines: ☒ CD ☒ RI ☒ DSR ☒ CTS ☐ DTR ☐ RTS

Send: 1 ☐ HEX
 0 ☐ HEX
 ☐ HEX

HWgroup
www.HW-group.com
 Hercules SETUP utility
 Version 3.2.8

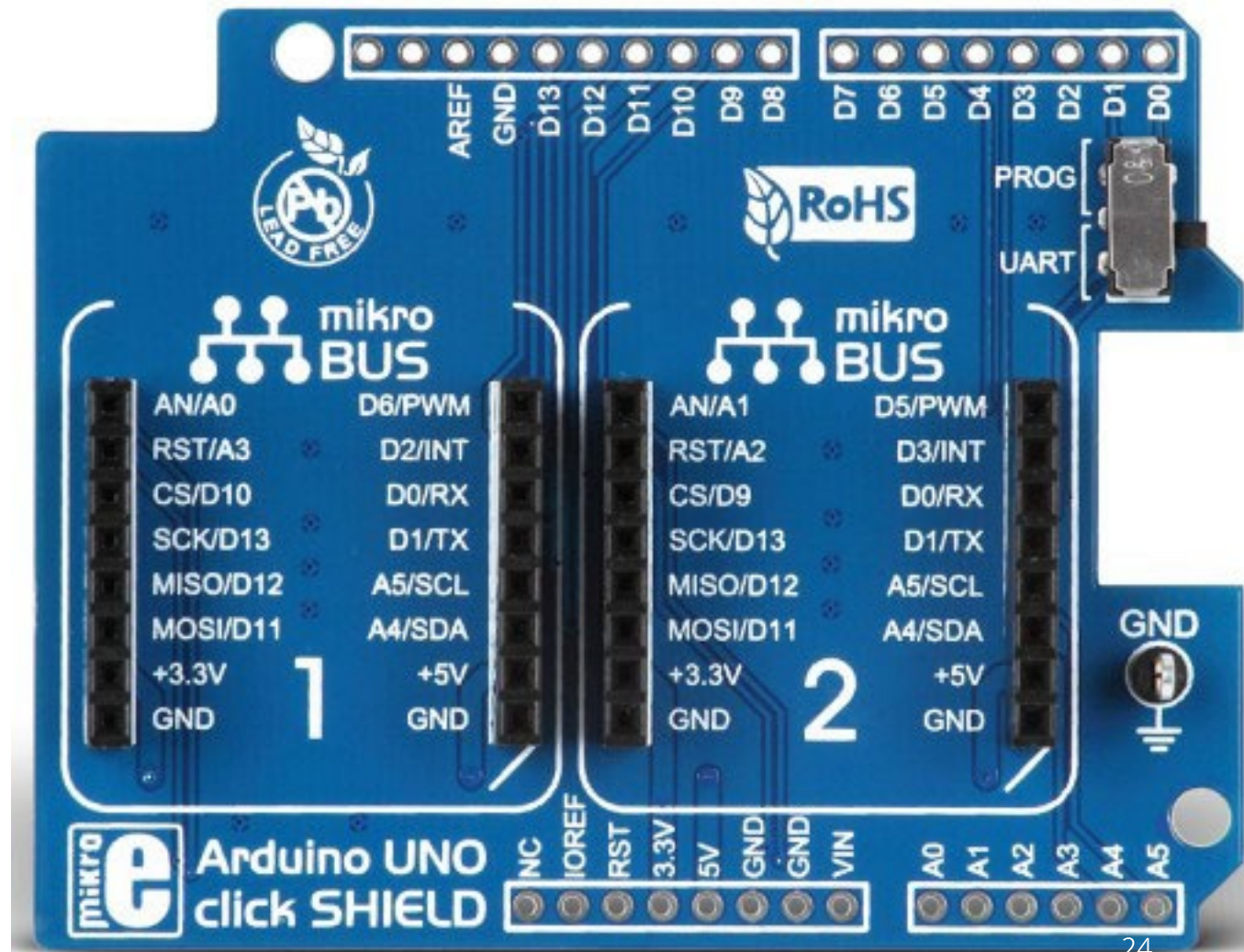
Next Time...

MORE TO COME..

Thank you for attending!!!

Please consider the resources below:

- [Today's Download Package](#)
- [STM32C071RB Datasheet](#)
- [NUCLEO-C071RB Schematic](#)





DesignNews

Thank You

Sponsored by

DigiKey

