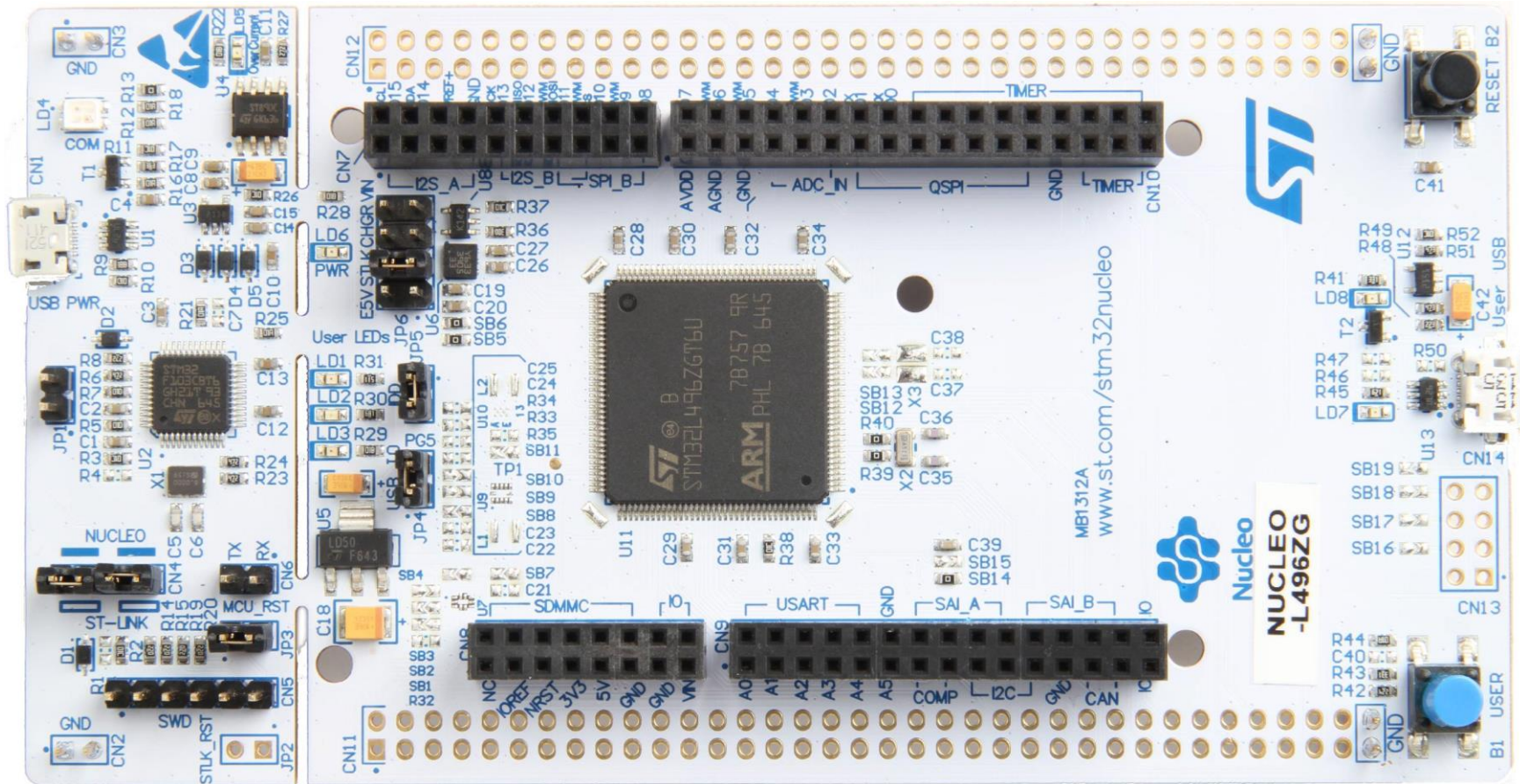


ARM Development Primer



ARM Peripherals 101

November 15, 2017

FRED EADY

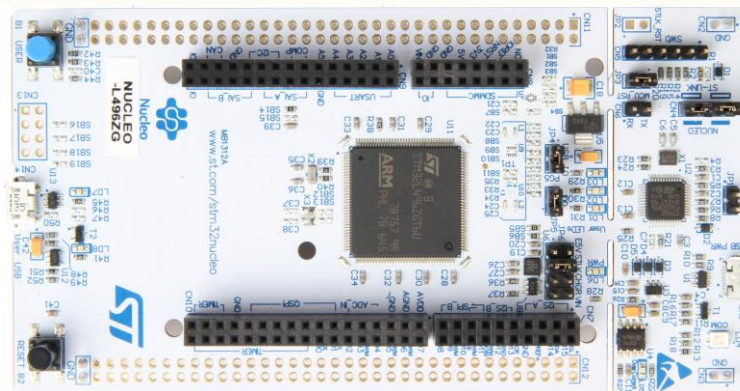
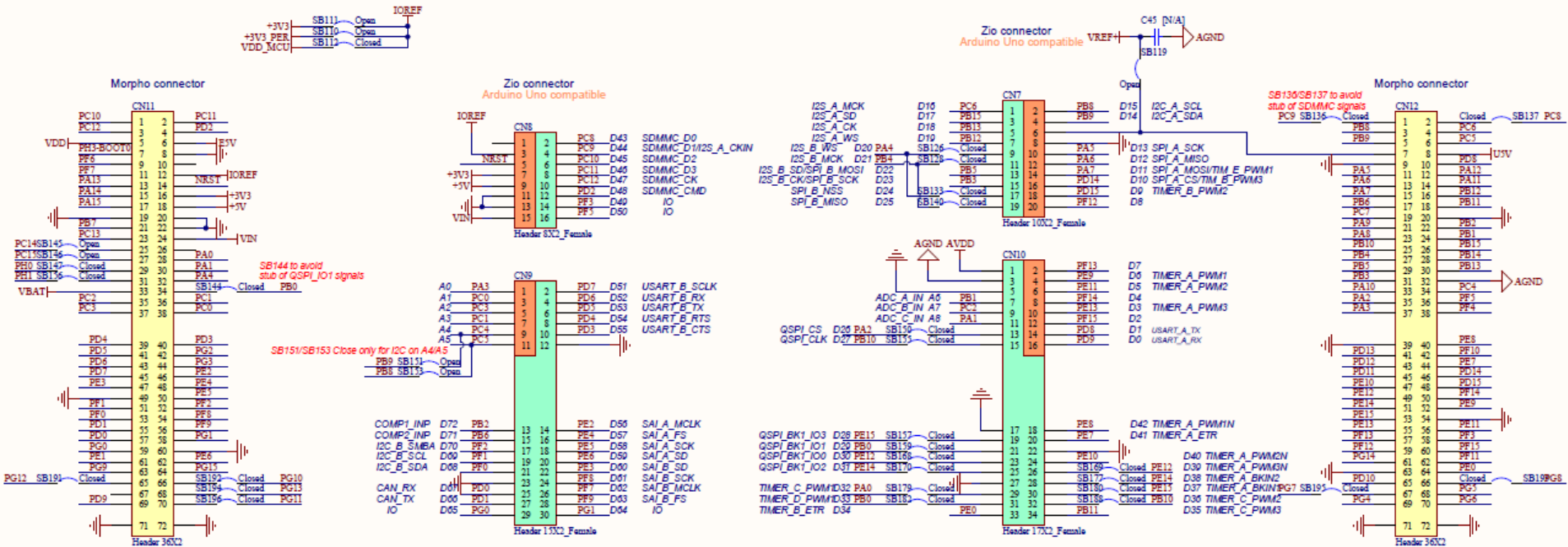
AGENDA

- I²C the STM32Cube Way
- UART the Old-Fashioned CMSIS Way
- Day 3's Done



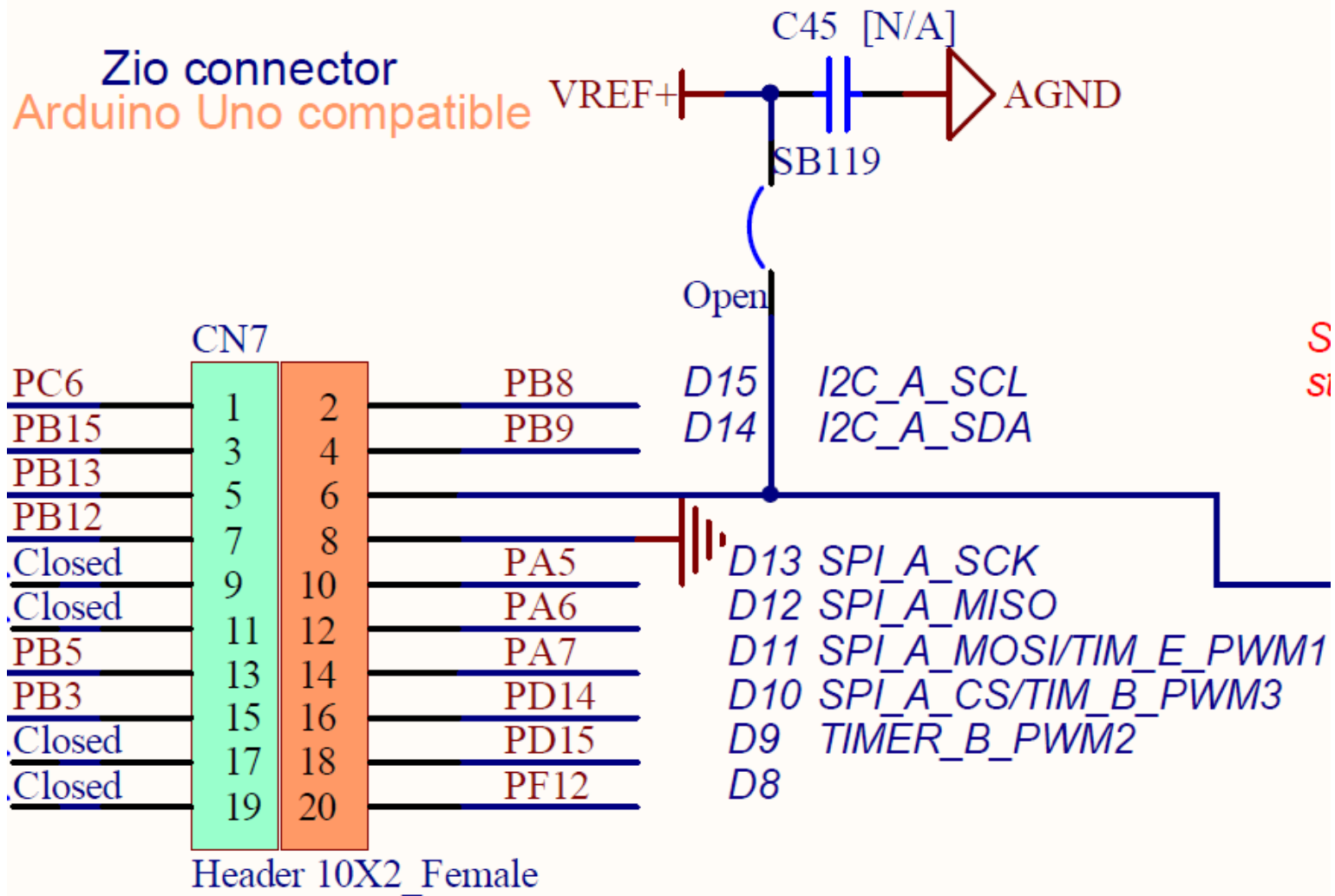
ARM Development Primer

I²C the STM32Cube Way – Hardware Specifics



ARM Development Primer

I²C the STM32Cube Way – Hardware Specifics



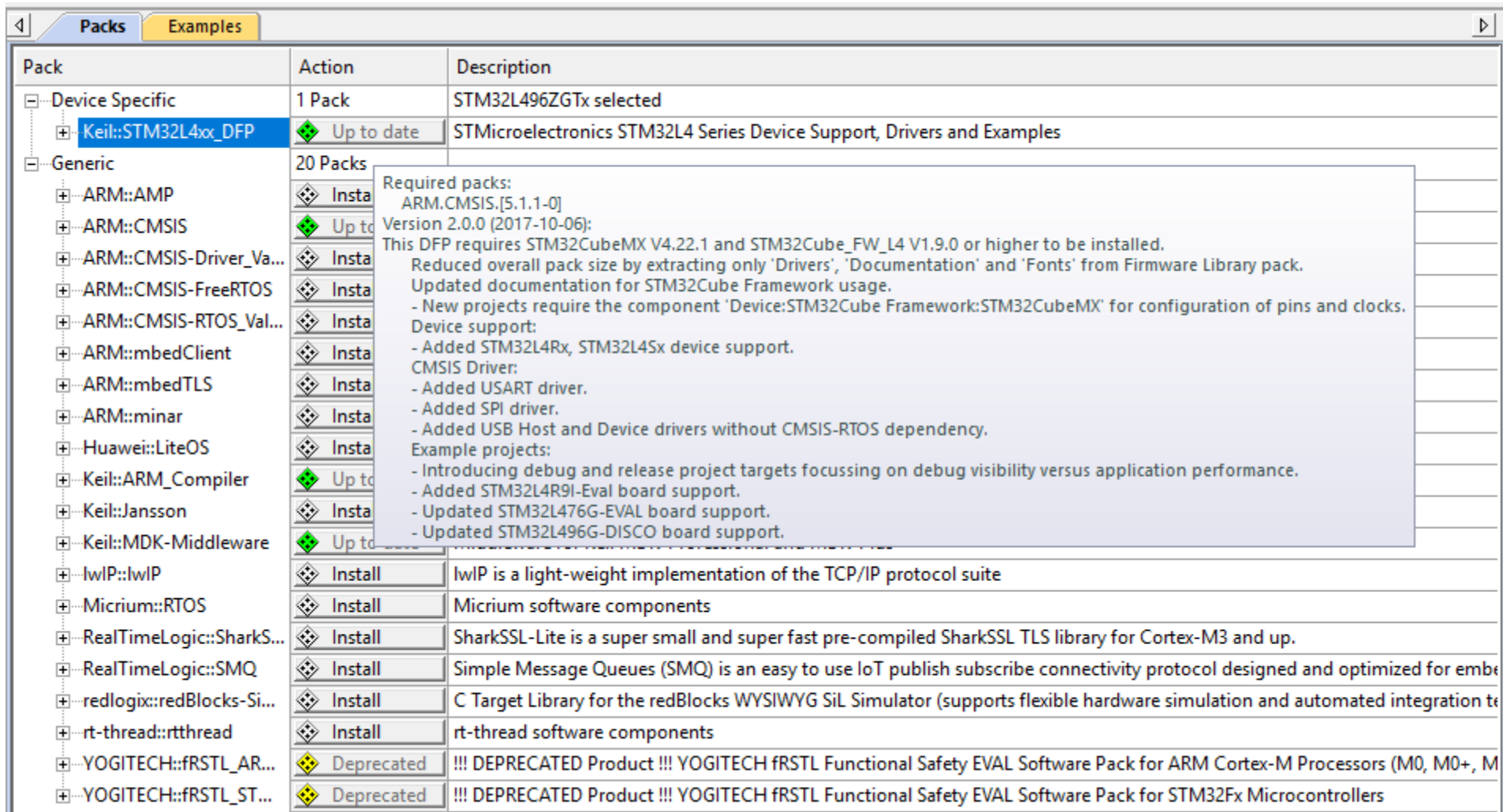
ARM Development Primer

I²C the STM32Cube Way – Install Packs

Pack	Action	Description
[-] Device Specific	1 Pack	STM32L496ZGTx selected
[-] Keil::STM32L4xx_DFP	Install	STMicroelectronics STM32L4 Series Device Support, Drivers and Examples
[-] Generic	20 Packs	
[-] ARM::AMP	Install	Software components for inter processor communication (Asymmetric Multi Processing AMP)
[-] ARM::CMSIS	Up to date	CMSIS (Cortex Microcontroller Software Interface Standard)
[-] ARM::CMSIS-Driver_Val...	Install	CMSIS-Driver Validation
[-] ARM::CMSIS-FreeRTOS	Install	Bundle of FreeRTOS for Cortex-M and Cortex-A
[-] ARM::CMSIS-RTOS_Val...	Install	CMSIS-RTOS Validation
[-] ARM::mbedClient	Install	ARM mbed Client for Cortex-M devices
[-] ARM::mbedTLS	Install	ARM mbed Cryptographic and SSL/TLS library for Cortex-M devices
[-] ARM::minar	Install	mbed OS Scheduler for Cortex-M devices
[-] Huawei::LiteOS	Install	Huawei LiteOS kernel Software Pack
[-] Keil::ARM_Compiler	Up to date	Keil ARM Compiler extensions for ARM Compiler 5 and ARM Compiler 6
[-] Keil::Jansson	Install	Jansson is a C library for encoding, decoding and manipulating JSON data
[-] Keil::MDK-Middleware	Up to date	Middleware for Keil MDK-Professional and MDK-Plus
[-] lwIP::lwIP	Install	lwIP is a light-weight implementation of the TCP/IP protocol suite
[-] Micrium::RTOS	Install	Micrium software components
[-] RealTimeLogic::SharkS...	Install	SharkSSL-Lite is a super small and super fast pre-compiled SharkSSL TLS library for Cortex-M3 and
[-] RealTimeLogic::SMQ	Install	Simple Message Queues (SMQ) is an easy to use IoT publish subscribe connectivity protocol desig
[-] redlogix::redBlocks-Si...	Install	C Target Library for the redBlocks WYSIWYG SiL Simulator (supports flexible hardware simulation ;
[-] rt-thread::rtthread	Install	rt-thread software components
[-] YOGITECH::fRSTL_AR...	Deprecated	!!! DEPRECATED Product !!! YOGITECH fRSTL Functional Safety EVAL Software Pack for ARM Corte
[-] YOGITECH::fRSTL_ST...	Deprecated	!!! DEPRECATED Product !!! YOGITECH fRSTL Functional Safety EVAL Software Pack for STM32Fx M

ARM Development Primer

I²C the STM32Cube Way – Install Packs



Pack	Action	Description
[-] Device Specific	1 Pack	STM32L496ZGTx selected
[+] Keil::STM32L4xx_DFP	Up to date	STMicroelectronics STM32L4 Series Device Support, Drivers and Examples
[-] Generic	20 Packs	
[+] ARM::AMP	Install	
[+] ARM::CMSIS	Up to date	
[+] ARM::CMSIS-Driver_Va...	Install	
[+] ARM::CMSIS-FreeRTOS	Install	
[+] ARM::CMSIS-RTOS_Val...	Install	
[+] ARM::mbedClient	Install	
[+] ARM::mbedTLS	Install	
[+] ARM::minar	Install	
[+] Huawei::LiteOS	Install	
[+] Keil::ARM_Compiler	Up to date	
[+] Keil::Jansson	Install	
[+] Keil::MDK-Middleware	Up to date	
[+] lwIP::lwIP	Install	lwIP is a light-weight implementation of the TCP/IP protocol suite
[+] Micrium::RTOS	Install	Micrium software components
[+] RealTimeLogic::SharkS...	Install	SharkSSL-Lite is a super small and super fast pre-compiled SharkSSL TLS library for Cortex-M3 and up.
[+] RealTimeLogic::SMQ	Install	Simple Message Queues (SMQ) is an easy to use IoT publish subscribe connectivity protocol designed and optimized for embed
[+] redlogix::redBlocks-Si...	Install	C Target Library for the redBlocks WYSIWYG SiL Simulator (supports flexible hardware simulation and automated integration to
[+] rt-thread::rtthread	Install	rt-thread software components
[+] YOGITECH::fRSTL_AR...	Deprecated	!!! DEPRECATED Product !!! YOGITECH fRSTL Functional Safety EVAL Software Pack for ARM Cortex-M Processors (M0, M0+, M
[+] YOGITECH::fRSTL_ST...	Deprecated	!!! DEPRECATED Product !!! YOGITECH fRSTL Functional Safety EVAL Software Pack for STM32Fx Microcontrollers

Required packs:
ARM.CMSIS.[5.1.1-0]
Version 2.0.0 (2017-10-06):
This DFP requires STM32CubeMX V4.22.1 and STM32Cube_FW_L4 V1.9.0 or higher to be installed.

Reduced overall pack size by extracting only 'Drivers', 'Documentation' and 'Fonts' from Firmware Library pack.
Updated documentation for STM32Cube Framework usage.

- New projects require the component 'Device:STM32Cube Framework:STM32CubeMX' for configuration of pins and clocks.

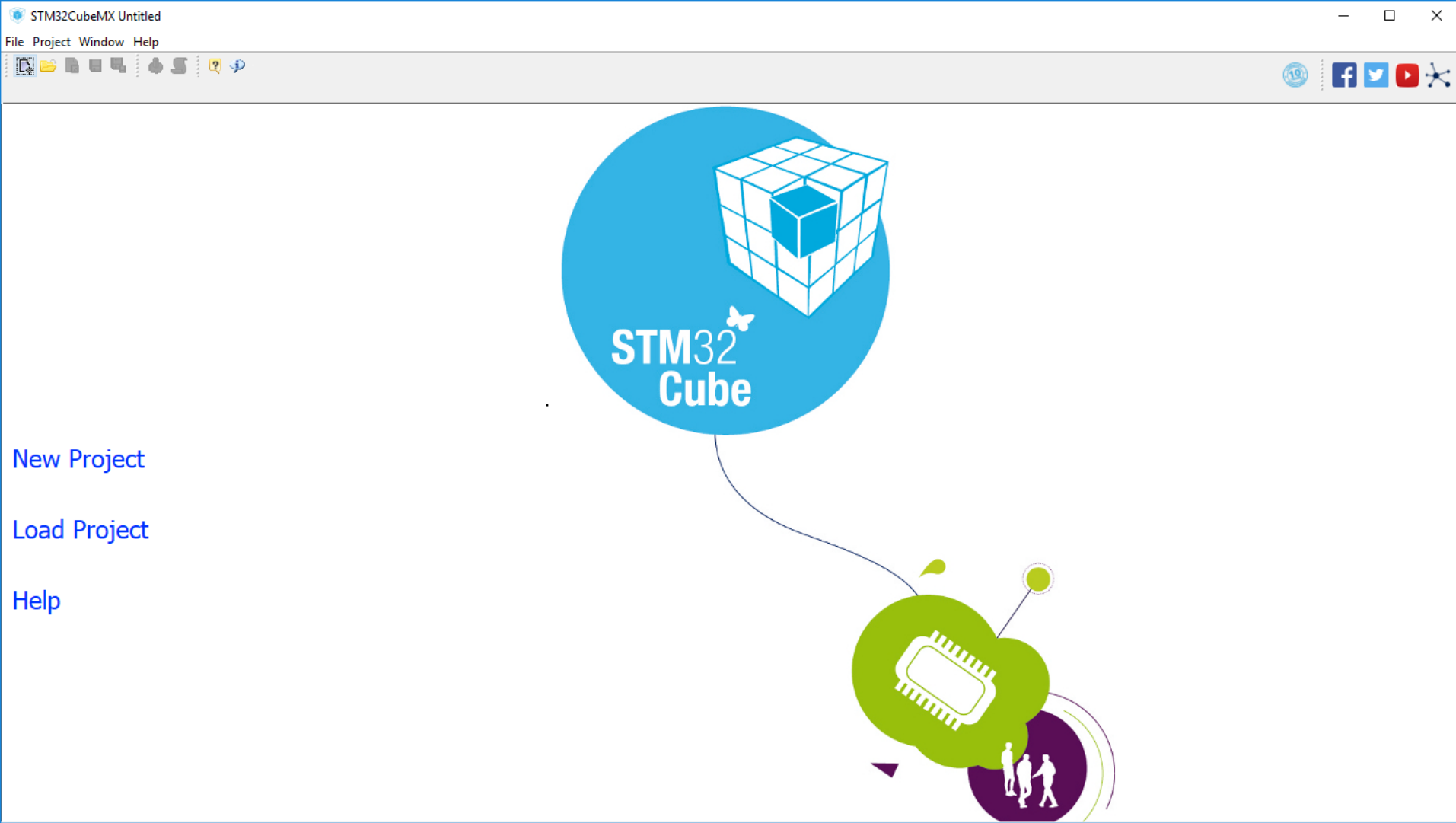
Device support:
- Added STM32L4Rx, STM32L4Sx device support.

CMSIS Driver:
- Added USART driver.
- Added SPI driver.
- Added USB Host and Device drivers without CMSIS-RTOS dependency.

Example projects:
- Introducing debug and release project targets focussing on debug visibility versus application performance.
- Added STM32L4R9I-Eval board support.
- Updated STM32L476G-EVAL board support.
- Updated STM32L496G-DISCO board support.

ARM Development Primer

I²C the STM32Cube Way – Generate Project



ARM Development Primer

I²C the STM32Cube Way – Generate Project

New Project

MCU Selector Board Selector

Board Filter

Vendor : STMicronics Type of Board : Nucleo144 MCU Series : All


Initialize all peripherals with their default Mode >>

Peripheral Selection

Peripherals	Nb	Max
<input type="radio"/> Accelerometer	<input type="checkbox"/>	
<input type="radio"/> Analog I/O	0	0
<input type="radio"/> Arduino Form Factor	0	0
<input type="radio"/> Audio Line In	0	0
<input type="radio"/> Audio Line Out	0	0
<input checked="" type="radio"/> Button	0	1
<input type="radio"/> CAN	0	0
<input type="radio"/> Camera	<input type="checkbox"/>	
<input type="radio"/> Compass	<input type="checkbox"/>	
<input type="radio"/> Custom Form Factor	0	0
<input checked="" type="radio"/> Digital I/O	0	76
<input type="radio"/> Eeprom	<input type="checkbox"/>	
<input checked="" type="radio"/> Ethernet	<input type="checkbox"/>	
<input type="radio"/> Flash Memory	0	0
<input type="radio"/> Freedom Form Factor	0	0
<input type="radio"/> Gyroscope	<input type="checkbox"/>	
<input type="radio"/> IrDA	<input type="checkbox"/>	
<input type="radio"/> Joystick	<input type="checkbox"/>	
<input type="radio"/> Lcd Display (Graphics)	<input type="checkbox"/>	
<input type="radio"/> Lcd Display (Segment)	<input type="checkbox"/>	
<input checked="" type="radio"/> Led	0	1
<input type="radio"/> Light Sensor	<input type="checkbox"/>	
<input type="radio"/> Memory Card	<input type="checkbox"/>	
<input type="radio"/> Microphone	0	0
<input type="radio"/> Potentiometer	<input type="checkbox"/>	
<input type="radio"/> Pressure Sensor	<input type="checkbox"/>	
<input type="radio"/> RS-232	0	0
<input type="radio"/> RS-485	<input type="checkbox"/>	
<input type="radio"/> SRAM/SDRAM	0	0
<input type="radio"/> Speaker	0	0
<input type="radio"/> Temperature Sensor	<input type="checkbox"/>	
<input type="radio"/> Touch Key Sensing	<input type="checkbox"/>	
<input type="radio"/> Tower Form Factor	0	0
<input checked="" type="radio"/> IISR	0	1

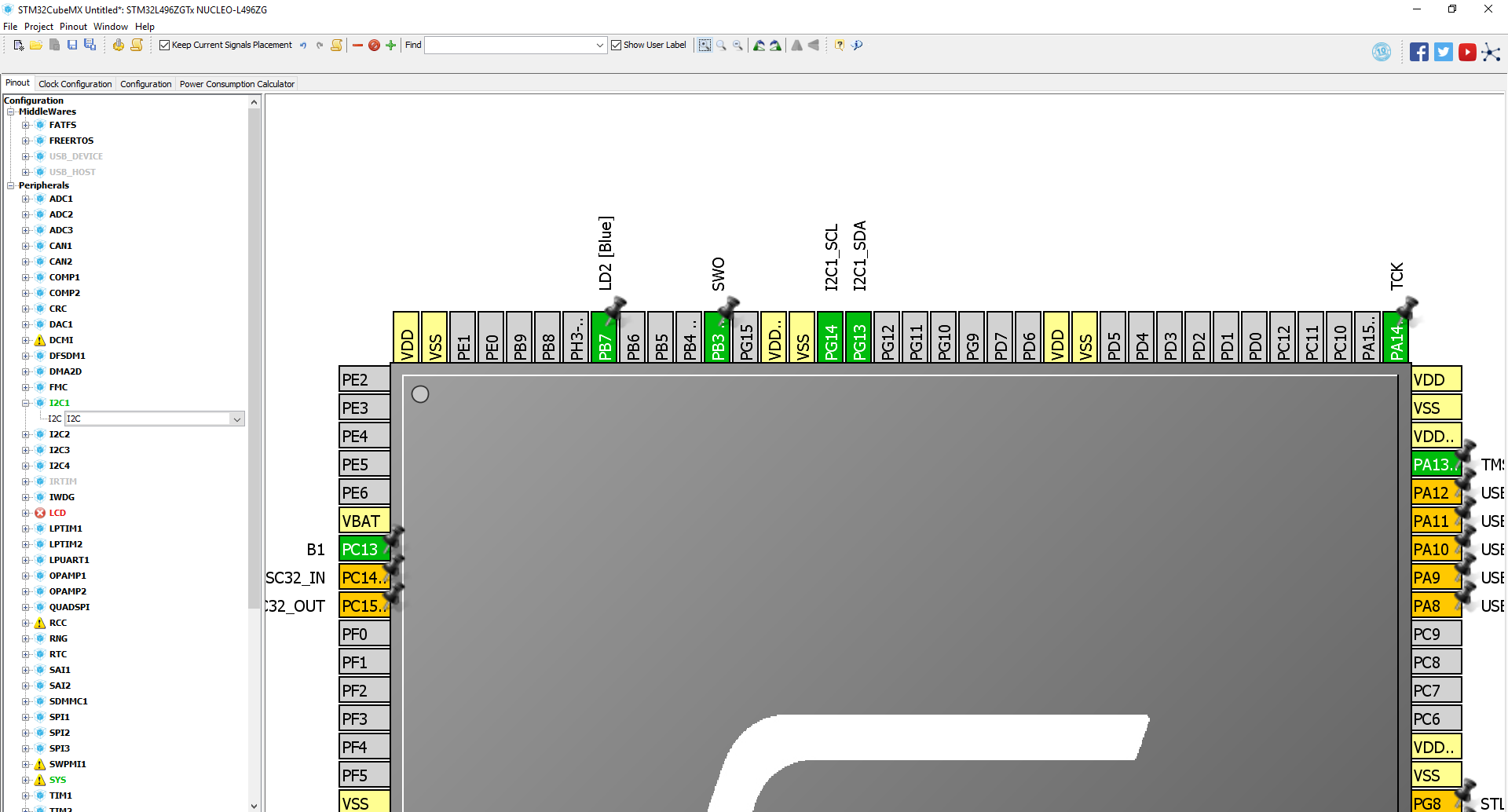
Boards List: 13 Items

Type	Reference	MCU
Nucleo144	NUCLEO-F303ZE	STM32F303ZETx
Nucleo144	NUCLEO-F207ZG	STM32F207ZGTx
Nucleo144	NUCLEO-F429ZI	STM32F429ZITx
Nucleo144	NUCLEO-F446ZE	STM32F446ZETx
Nucleo144	NUCLEO-F746ZG	STM32F746ZGTx
Nucleo144	NUCLEO-F767ZI	STM32F767ZITx
Nucleo144	NUCLEO-F412ZG	STM32F412ZGTx
Nucleo144	NUCLEO-F413ZI	STM32F413ZITx
Nucleo144	NUCLEO-L496ZG	STM32L496ZGTx
Nucleo144	NUCLEO-L496ZG-P	STM32L496ZGTxP
Nucleo144	NUCLEO-L4R5ZI	STM32L4R5ZITx
Nucleo144	NUCLEO-F722ZE	STM32F722ZETx
Nucleo144	NUCLEO-H743ZI	STM32H743ZITx



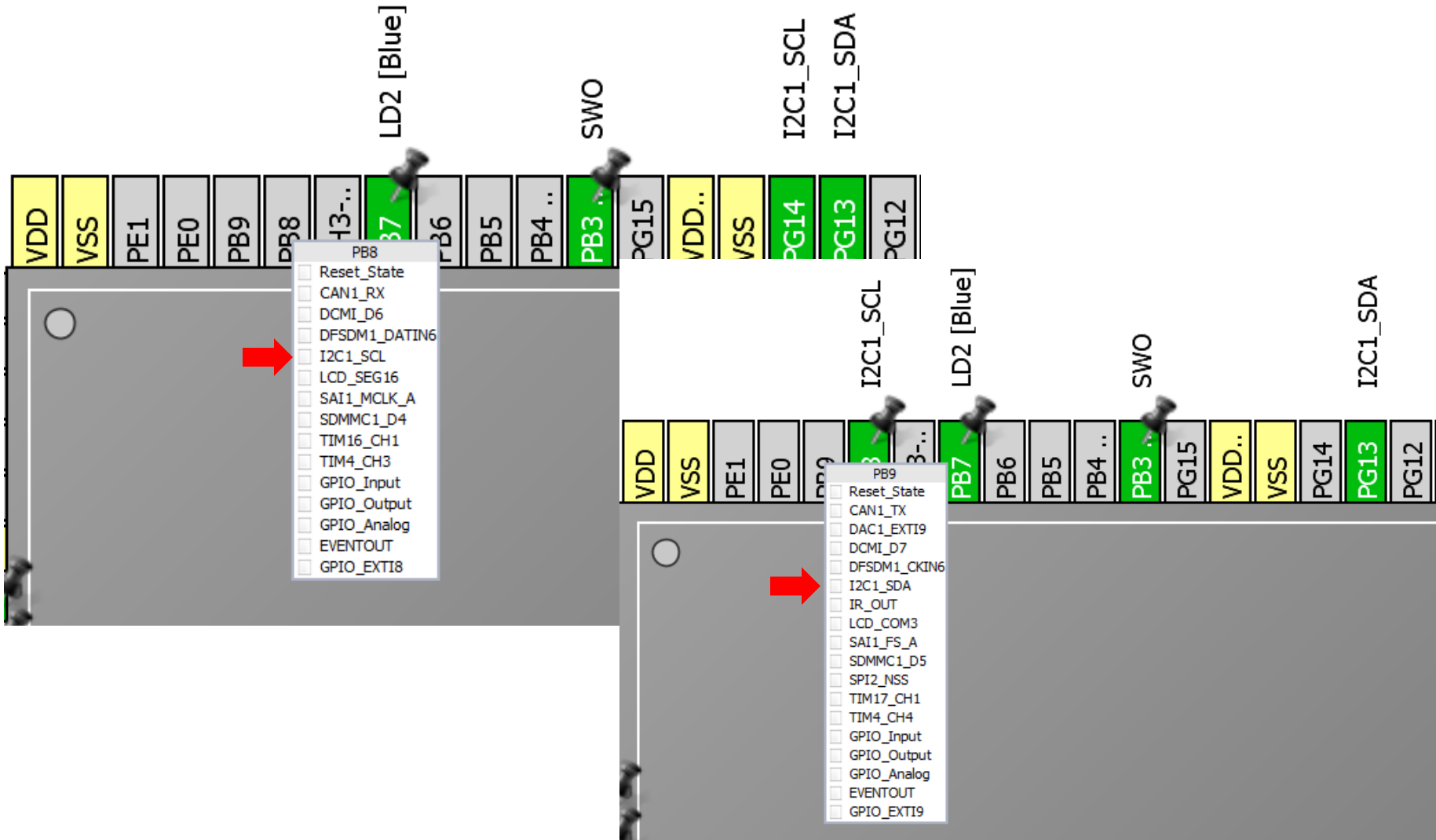
ARM Development Primer

I²C the STM32Cube Way – Generate Project



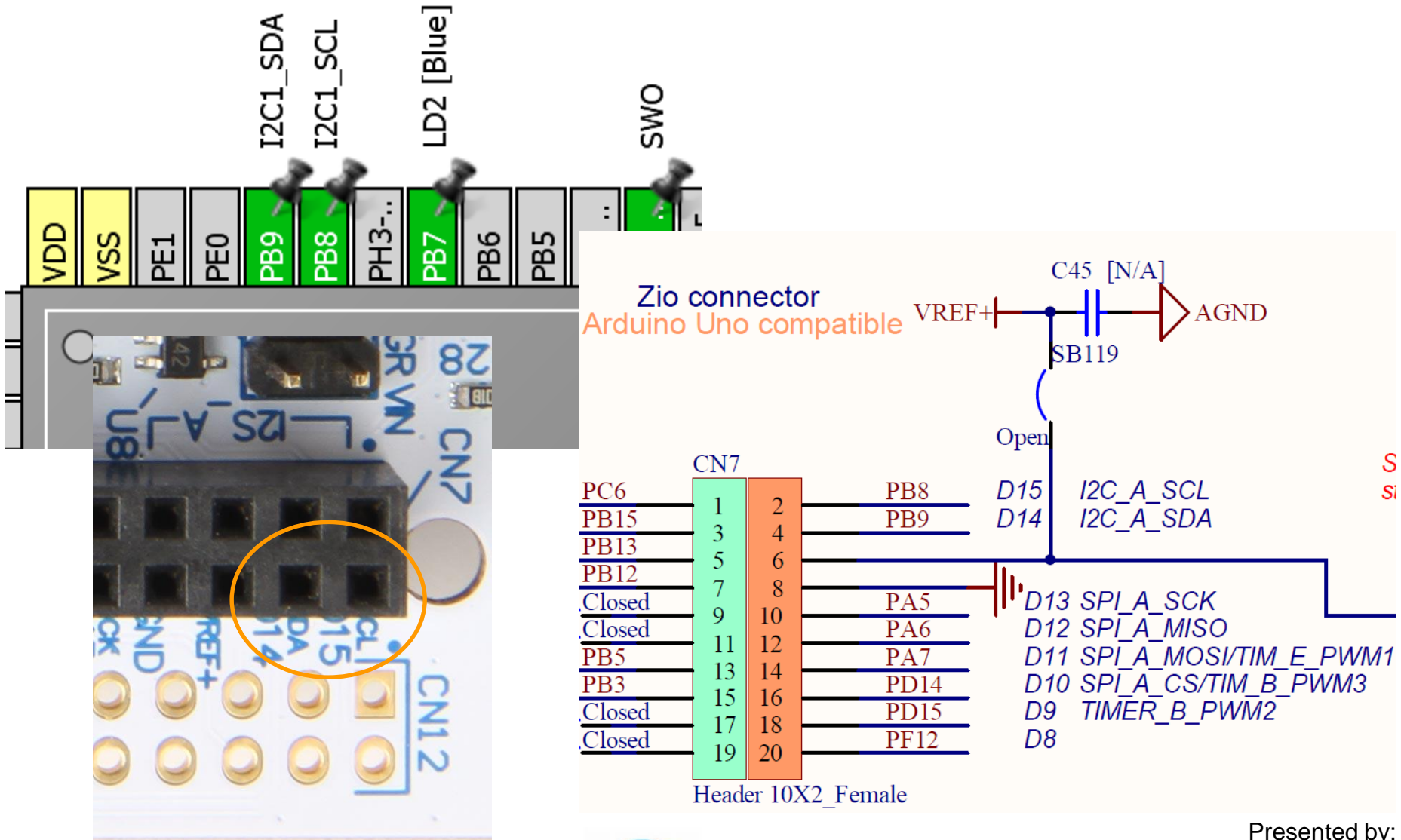
ARM Development Primer

I²C the STM32Cube Way – Generate Project



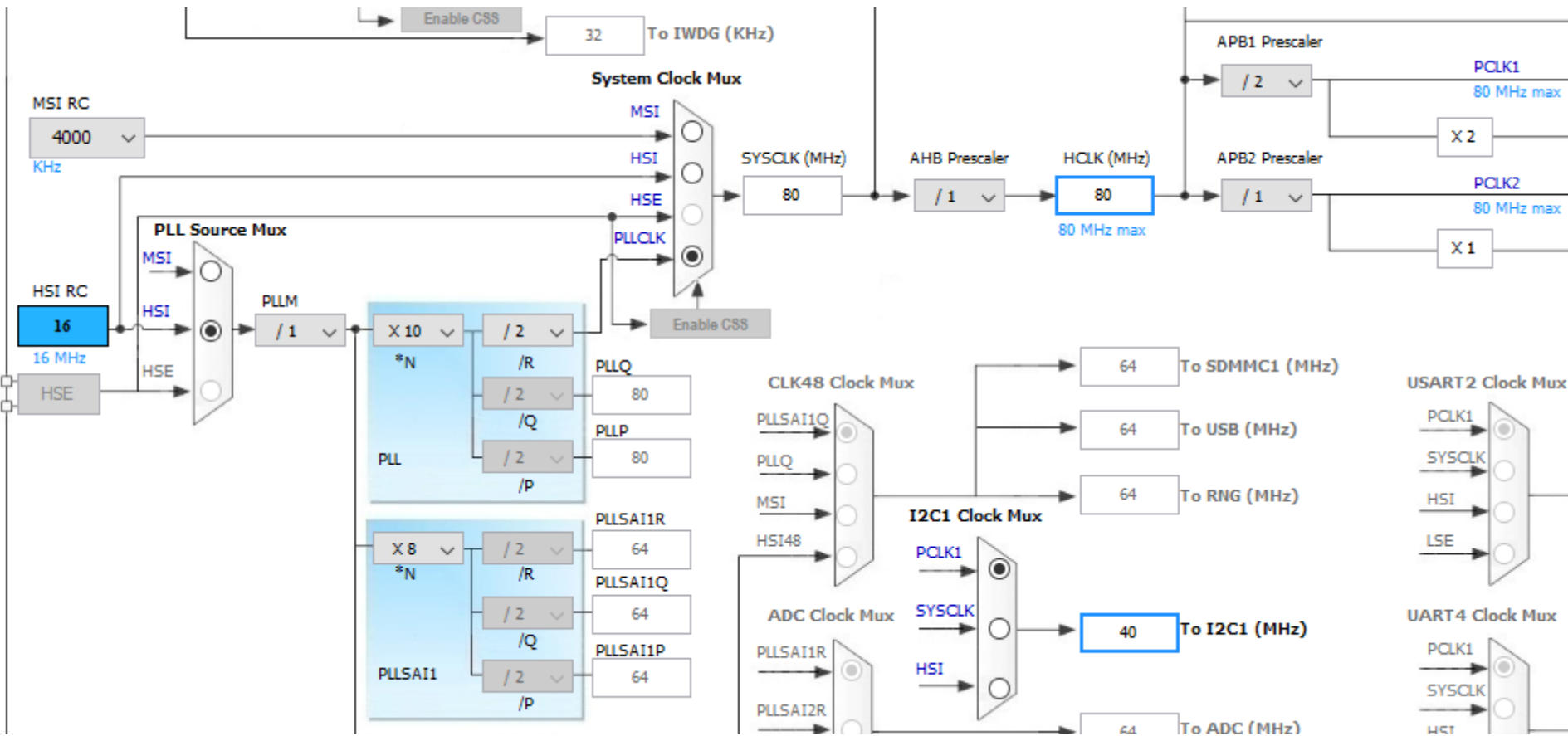
ARM Development Primer

I²C the STM32Cube Way – Generate Project



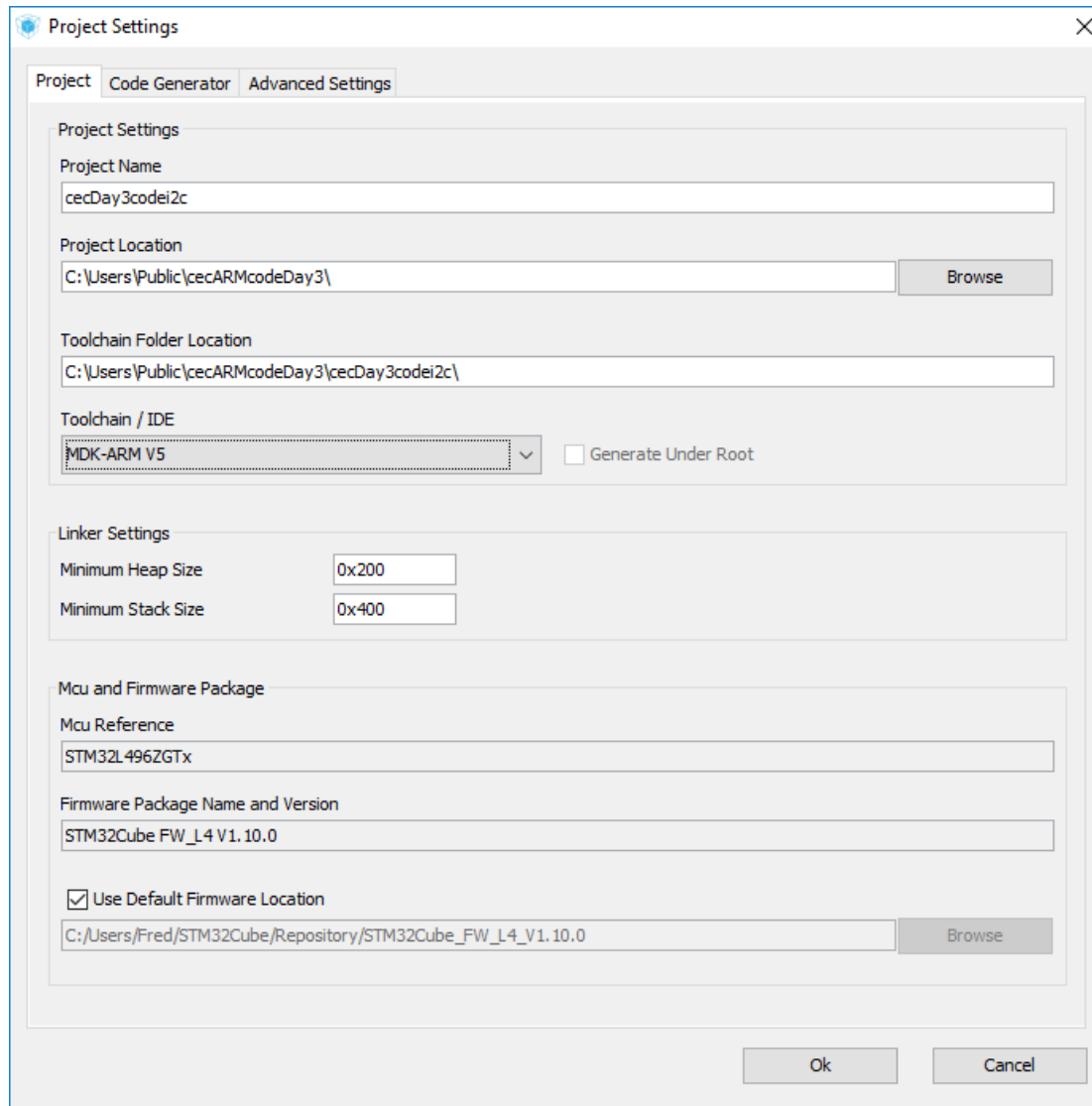
ARM Development Primer

I²C the STM32Cube Way – Generate Project



ARM Development Primer

I²C the STM32Cube Way – Generate Project



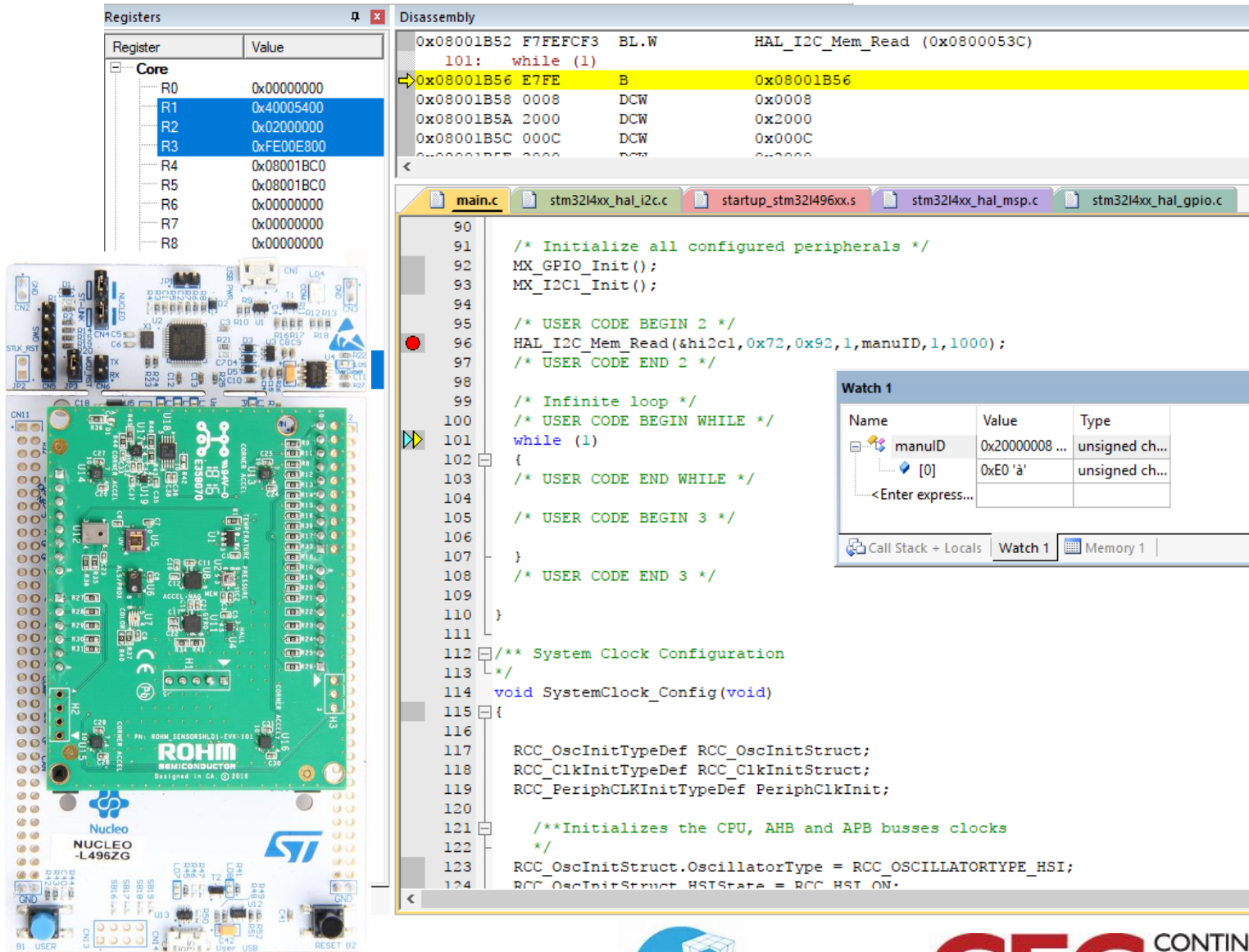
The screenshot shows the 'Project Settings' dialog box in the STM32Cube IDE. The 'Project' tab is selected, and the 'Advanced Settings' sub-tab is active. The dialog is organized into several sections:

- Project Settings:**
 - Project Name:** cecDay3codei2c
 - Project Location:** C:\Users\Public\cecARMcodeDay3\ (with a 'Browse' button)
 - Toolchain Folder Location:** C:\Users\Public\cecARMcodeDay3\cecDay3codei2c\
 - Toolchain / IDE:** MDK-ARM V5 (dropdown menu) and Generate Under Root
- Linker Settings:**
 - Minimum Heap Size:** 0x200
 - Minimum Stack Size:** 0x400
- Mcu and Firmware Package:**
 - Mcu Reference:** STM32L496ZGTx
 - Firmware Package Name and Version:** STM32Cube FW_L4 V1.10.0
 - Use Default Firmware Location
 - Firmware Location:** C:/Users/Fred/STM32Cube/Repository/STM32Cube_FW_L4_V1.10.0 (with a 'Browse' button)

At the bottom of the dialog are 'Ok' and 'Cancel' buttons.

ARM Development Primer

I²C the STM32Cube Way – Debug Project



The image displays a screenshot of an IDE (likely Keil MDK) showing a debugger window. The window is divided into several panes:

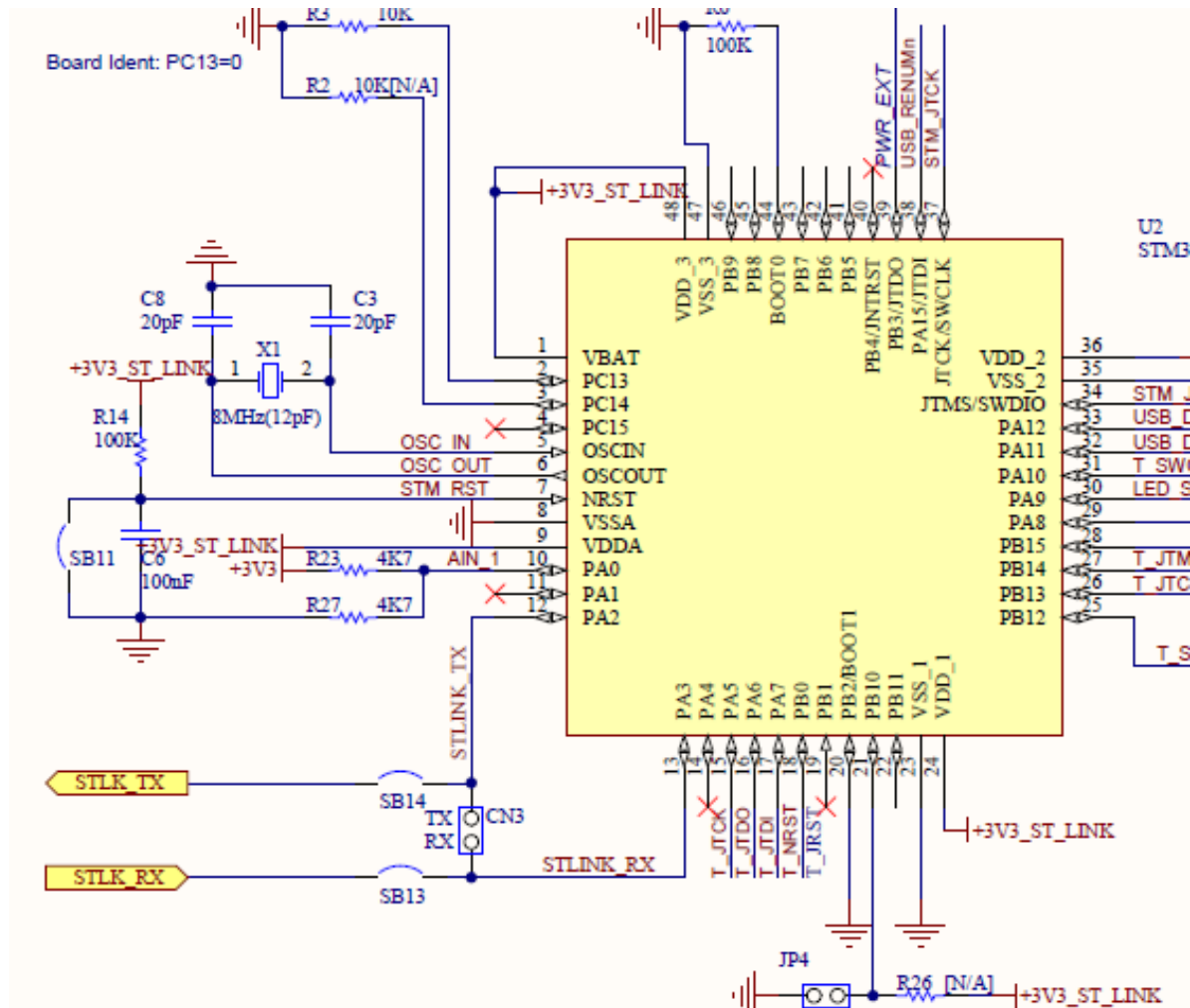
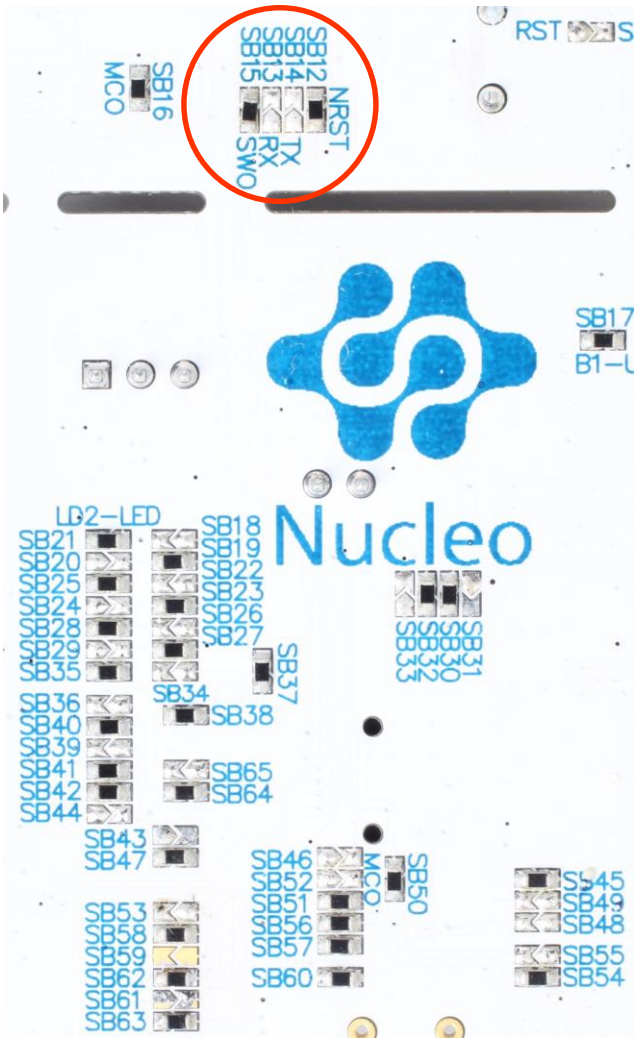
- Registers:** A table showing the state of the processor's registers. The Core registers R0 through R8 are listed with their values. R0 is 0x00000000, R1 is 0x40005400, R2 is 0x02000000, R3 is 0xFE00E800, R4 is 0x08001BC0, R5 is 0x08001BC0, R6 is 0x00000000, R7 is 0x00000000, and R8 is 0x00000000.
- Disassembly:** A window showing the assembly code being executed. The current instruction is highlighted in yellow: `0x08001B56 E7FE B 0x08001B56`. Other instructions include `HAL_I2C_Mem_Read (0x0800053C)`, `while (1)`, and `DCW` instructions.
- Source Code:** A window showing the C source code for `main.c`. The code includes initialization of peripherals, a while loop for I2C memory reading, and system clock configuration. The current execution point is at line 101: `while (1)`.

```
90
91 /* Initialize all configured peripherals */
92 MX_GPIO_Init();
93 MX_I2C1_Init();
94
95 /* USER CODE BEGIN 2 */
96 HAL_I2C_Mem_Read(&hi2c1, 0x72, 0x92, 1, manuID, 1, 1000);
97 /* USER CODE END 2 */
98
99 /* Infinite loop */
100 /* USER CODE BEGIN WHILE */
101 while (1)
102 {
103 /* USER CODE END WHILE */
104
105 /* USER CODE BEGIN 3 */
106
107 }
108 /* USER CODE END 3 */
109
110 }
111
112 /** System Clock Configuration
113 */
114 void SystemClock_Config(void)
115 {
116
117 RCC_OscInitTypeDef RCC_OscInitStruct;
118 RCC_ClkInitTypeDef RCC_ClkInitStruct;
119 RCC_PeriphCLKInitTypeDef PeriphClkInit;
120
121 /**Initializes the CPU, AHB and APB busses clocks
122 */
123 RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSI;
124 RCC_OscInitStruct.HSISState = RCC_HSI_ON;
```
- Watch 1:** A window showing the values of variables being watched. The variable `manuID` has a value of `0x20000008` (type: unsigned char), and its element `[0]` has a value of `0xE0 'a'` (type: unsigned char).

The background of the IDE shows a photograph of the Nucleo-L496ZG development board, which is a green PCB with various components, including a microcontroller, memory, and connectors.

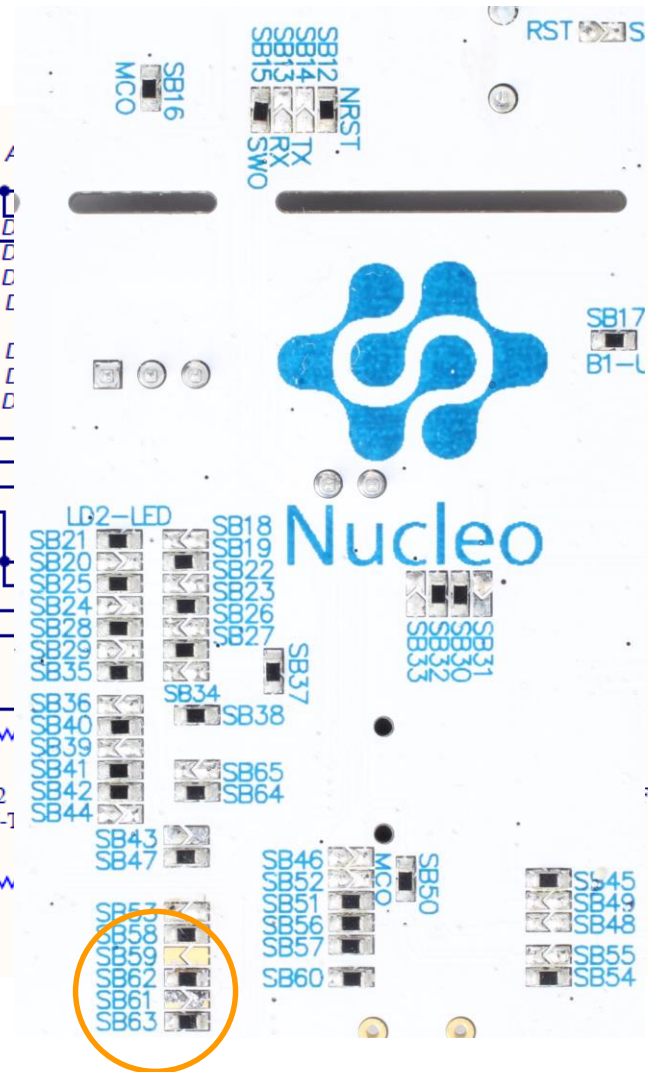
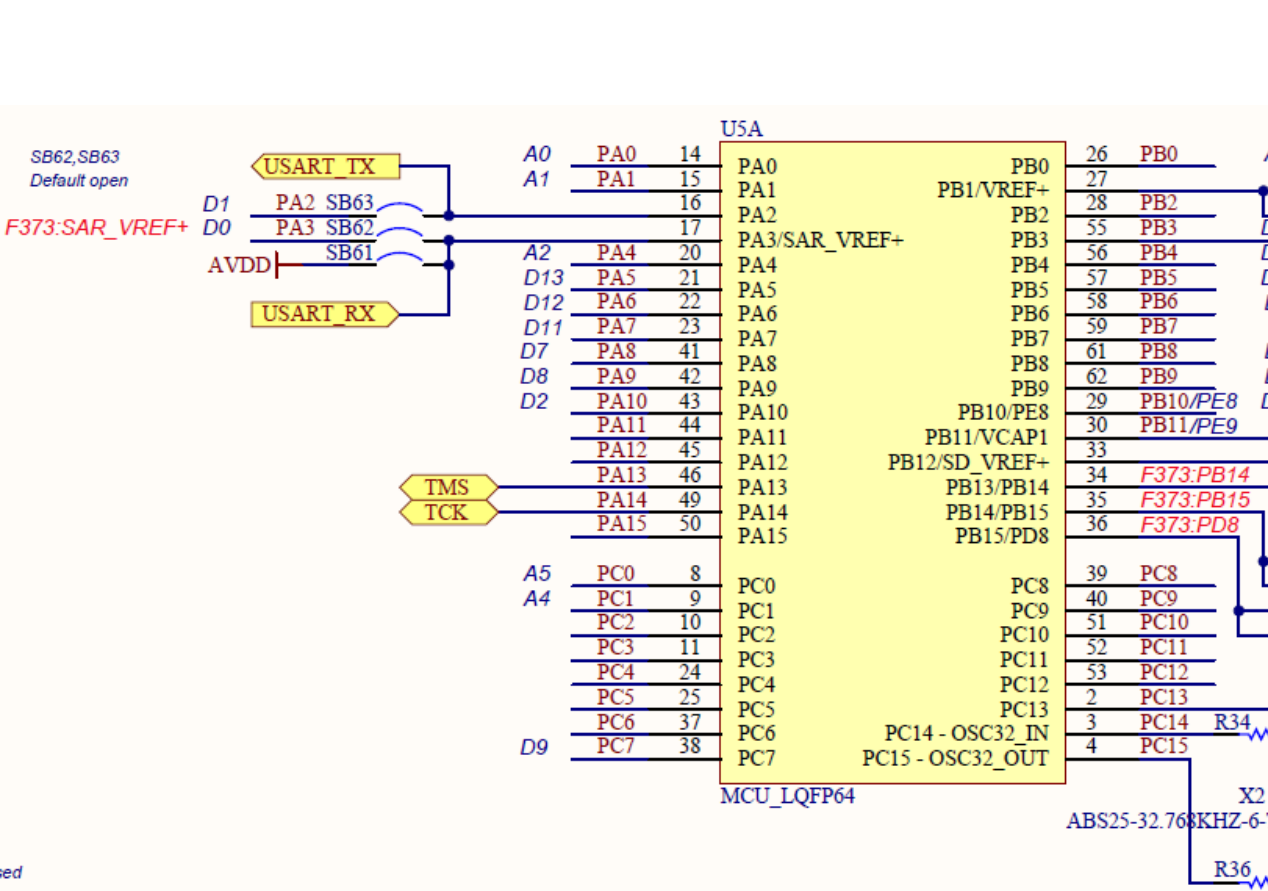
ARM Development Primer

UART the Old-Fashioned CMSIS Way



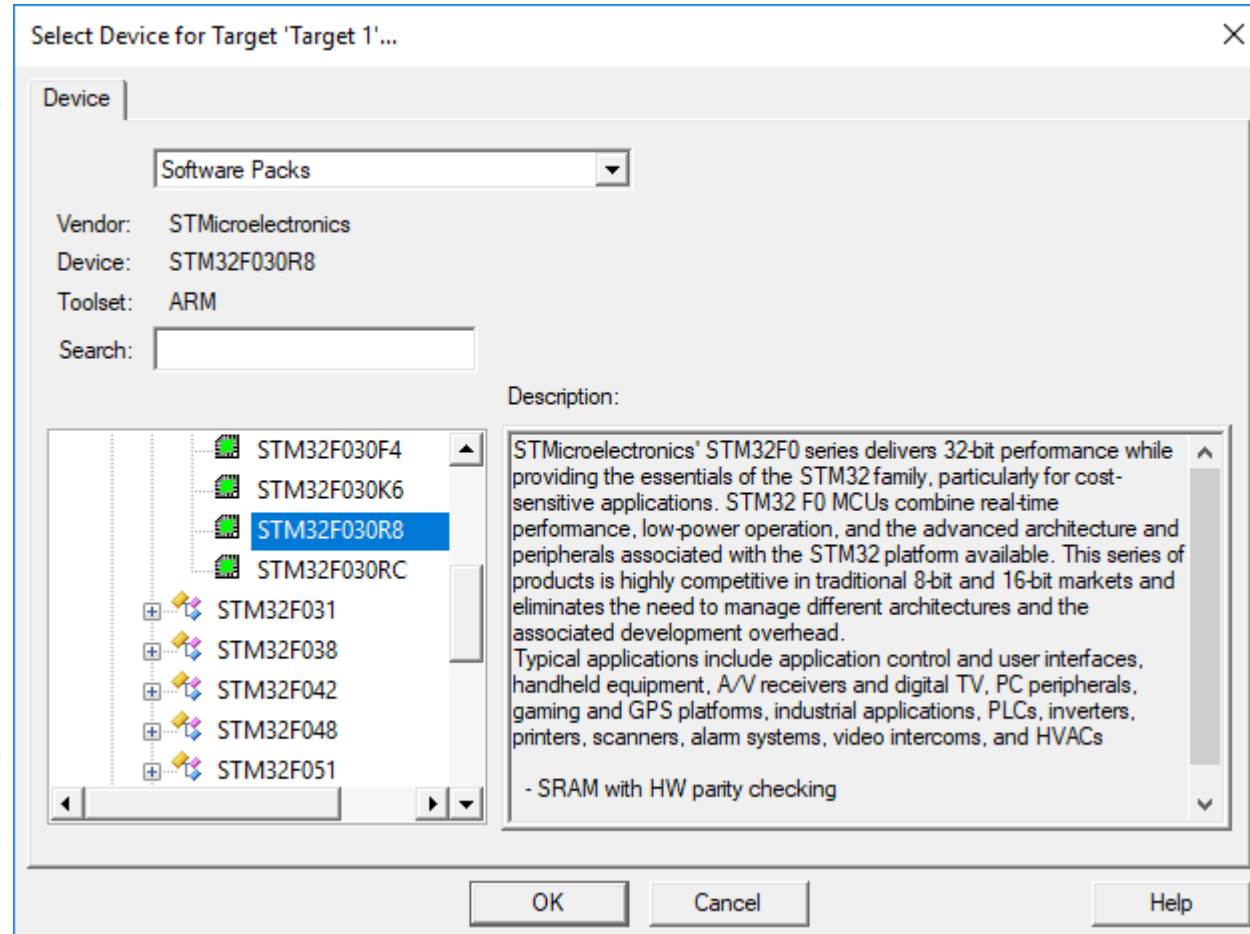
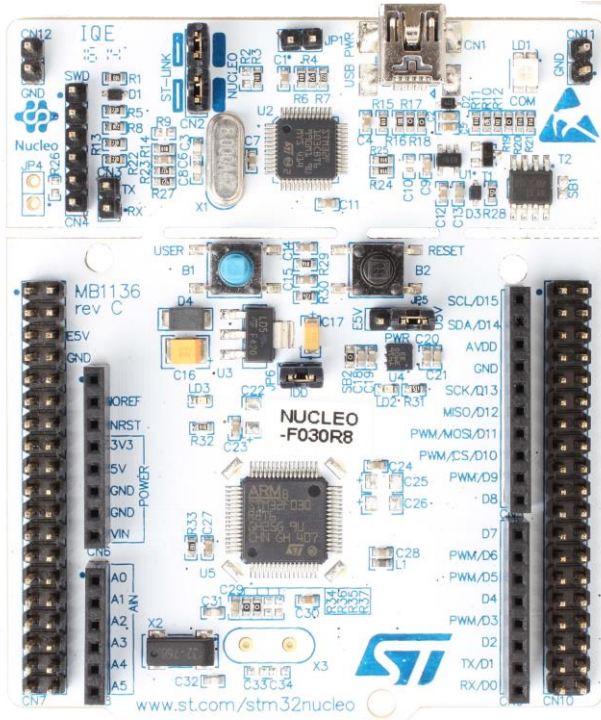
ARM Development Primer

UART the Old-Fashioned CMSIS Way



ARM Development Primer

UART the Old-Fashioned CMSIS Way



ARM Development Primer

UART the Old-Fashioned CMSIS Way

Manage Run-Time Environment

Software Component	Sel.	Variant	Version	Description
Board Support		NUCLEO-F030R8	1.0.0	STMicroelectronics NUCLEO-F030R8 Development Board
Buttons (API)			1.0.0	Buttons Interface
Buttons	<input type="checkbox"/>		1.0.0	Button driver for NUCLEO-F030R8 Development Board
LED (API)			1.0.0	LED Interface
LED	<input type="checkbox"/>		1.0.0	LED driver for NUCLEO-F030R8 Development Board
CMSIS	<input checked="" type="checkbox"/>		5.0.2	Cortex Microcontroller Software Interface Components
CORE	<input checked="" type="checkbox"/>		5.0.2	CMSIS-CORE for Cortex-M, SC000, SC300, ARMv8-M
DSP	<input type="checkbox"/>		1.5.2	CMSIS-DSP Library for Cortex-M, SC000, and SC300
RTOS (API)			1.0.0	CMSIS-RTOS API for Cortex-M, SC000, and SC300
RTOS2 (API)			2.1.1	CMSIS-RTOS API for Cortex-M, SC000, and SC300
CMSIS Driver				Unified Device Drivers compliant to CMSIS-Driver Specifications
Ethernet (API)			2.1.0	Ethernet MAC and PHY Driver API for Cortex-M
Ethernet PHY (API)			2.1.0	Ethernet PHY Driver API for Cortex-M
Flash (API)			2.1.0	Flash Driver API for Cortex-M
NAND (API)			2.2.0	NAND Flash Driver API for Cortex-M
USART (API)			2.3.0	USART Driver API for Cortex-M
Compiler	<input checked="" type="checkbox"/>	ARM Compiler	1.2.1	Compiler Extensions for ARM Compiler 5 and ARM Compiler 6
Event Recorder	<input type="checkbox"/>	DAP	1.2.1	Event Recording using Debug Access Port (DAP)
I/O	<input checked="" type="checkbox"/>			Retarget Input/Output
File	<input type="checkbox"/>	File System	1.2.1	Use retargeting together with the File System component
STDERR	<input type="checkbox"/>	User	1.2.1	Redirect STDERR to a user defined output target (USART, Graphics Display or other)
STDIN	<input type="checkbox"/>	User	1.2.1	Retrieve STDIN from a user specified input source (USART, Keyboard or other)
STDOUT	<input checked="" type="checkbox"/>	User	1.2.1	Redirect STDOUT to a user defined output target (USART, Graphics Display or other)
TTY	<input type="checkbox"/>	User	1.2.1	Redirect TTY to a user defined output target
Device	<input checked="" type="checkbox"/>			Startup, System Setup
Startup	<input checked="" type="checkbox"/>		2.2.3	System Startup for STMicroelectronics STM32F030x8 Devices
StartupConfig	<input type="checkbox"/>		8.14.1	Stack and Heap configuration
File System		MDK-Pro	6.9.8	File Access on various storage devices
Graphics		MDK-Pro	5.36.6	User Interface on graphical LCD displays
Network		MDK-Pro	7.5.0	IPv4/IPv6 Networking using Ethernet or Serial protocols
USB		MDK-Pro	6.11.0	USB Communication with various device classes

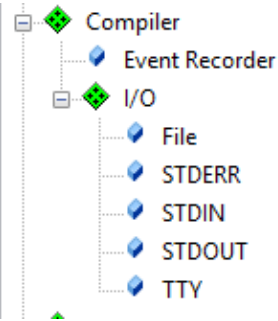
Validation Output

Description

Resolve Select Packs Details OK Cancel Help

ARM Development Primer

UART the Old-Fashioned CMSIS Way

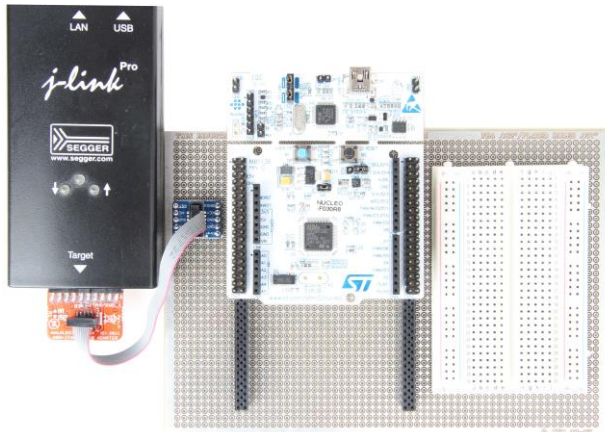


<input type="checkbox"/>	ARM Compiler	1.2.1	Compiler Extensions for ARM Compiler 5 and ARM Compiler 6
<input type="checkbox"/>	DAP	1.2.1	Event Recording using Debug Access Port (DAP)
			Retarget Input/Output
<input type="checkbox"/>	File System	1.2.1	Use retargeting together with the File System component
<input type="checkbox"/>	User	1.2.1	Redirect STDERR to a user defined output target (USART, Graphics Display or other)
<input type="checkbox"/>	User	1.2.1	Retrieve STDIN from a user specified input source (USART, Keyboard or other)
<input checked="" type="checkbox"/>	User	1.2.1	Redirect STDOUT to a user defined output target (USART, Graphics Display or other)
<input type="checkbox"/>	User	1.2.1	Redirect TTY to a user defined output target

```
2  /*
3  * Auto generated Run-Time-Environment Component Configuration File
4  *   *** Do not modify ! ***
5  *
6  * Project: 'cecDay3codeUARTF030'
7  * Target: 'CECSTM32F030 UART'
8  */
9
10 #ifndef RTE_COMPONENTS_H
11 #define RTE_COMPONENTS_H
12
13
14 /*
15 * Define the Device Header File:
16 */
17 #define CMSIS_device_header "stm32f0xx.h"
18
19 #define RTE_Compiler_IO_STDOUT          /* Compiler I/O: STDOUT */
20 #define RTE_Compiler_IO_STDOUT_User    /* Compiler I/O: STDOUT User */
21
22 #endif /* RTE_COMPONENTS_H */
23
```

ARM Development Primer

UART the Old-Fashioned CMSIS Way



Cortex JLink/JTrace Target Driver Setup

Debug | Trace | Flash Download

J-Link / J-Trace Adapter

SN: 174301702

Device: J-Link Pro

HW : V4.00 dll : V6.20h

FW : J-Link Pro V4 compiled Oct 6

Port: SW Max Clock: 5 MHz

Auto Clk

SW Device

IDCODE	Device Name	Move
SWD 0x0BB11477	ARM CoreSight SW-DP	Up Down

Automatic Detection ID CODE:

Manual Configuration Device Name:

IR len:

Add Delete Update

Connect & Reset Options

Connect: Normal Reset: Normal

Reset after Connect

Cache Options

Cache Code

Cache Memory

Download Options

Verify Code Download

Download to Flash

Interface

USB TCP/IP

Scan

State: ready

TCP/IP

Network Settings

IP-Address: 127 . 0 . 0 . 1 Port (Auto: 0): 0

Autodetect Ping

Misc

JLink Info

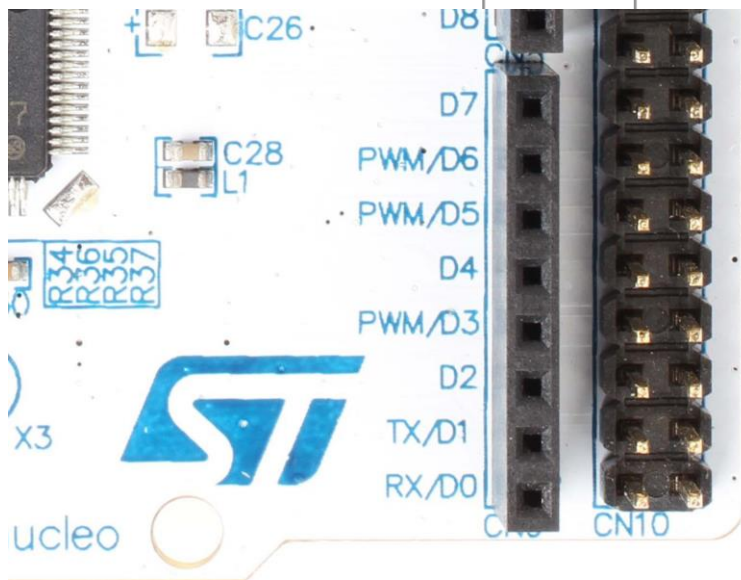
JLink Cmd

OK Cancel Apply

ARM Development Primer

UART the Old-Fashioned CMSIS Way

Pin name	AF0	AF1	AF2	AF3	AF4	AF5	AF6
PA0	-	USART1_CTS ⁽²⁾	-	-	USART4_TX ⁽¹⁾	-	-
		USART2_CTS ⁽¹⁾⁽³⁾					
PA1	EVENTOUT	USART1_RTS ⁽²⁾	-	-	USART4_RX ⁽¹⁾	TIM15_CH1N ⁽¹⁾	-
		USART2_RTS ⁽¹⁾⁽³⁾					
PA2	TIM15_CH1 ⁽¹⁾⁽³⁾	USART1_TX ⁽²⁾	-	-	-	-	-
		USART2_TX ⁽¹⁾⁽³⁾					
PA3	TIM15_CH2 ⁽¹⁾⁽³⁾	USART1_RX ⁽²⁾	-	-	-	-	-
		USART2_RX ⁽¹⁾⁽³⁾					
		USART1_CK ⁽²⁾	-	-	TIM14_CH1	USART6_TX ⁽¹⁾	-
		USART2_CK ⁽¹⁾⁽³⁾					
		-	-	-	-	USART6_RX ⁽¹⁾	-
		TIM3_CH1	TIM1_BKIN	-	USART3_CTS ⁽¹⁾	TIM16_CH1	EVENTOUT
		TIM3_CH2	TIM1_CH1N	-	TIM14_CH1	TIM17_CH1	EVENTOUT
		USART1_CK	TIM1_CH1	EVENTOUT	-	-	-
		USART1_TX ⁽¹⁾⁽³⁾	TIM1_CH2	-	I2C1_SCL ⁽¹⁾⁽²⁾	MCO ⁽¹⁾	-
		USART1_RX	TIM1_CH3	-	I2C1_SDA ⁽¹⁾⁽²⁾	-	-
		USART1_CTS	TIM1_CH4	-	-	SCL	-



ARM Development Primer

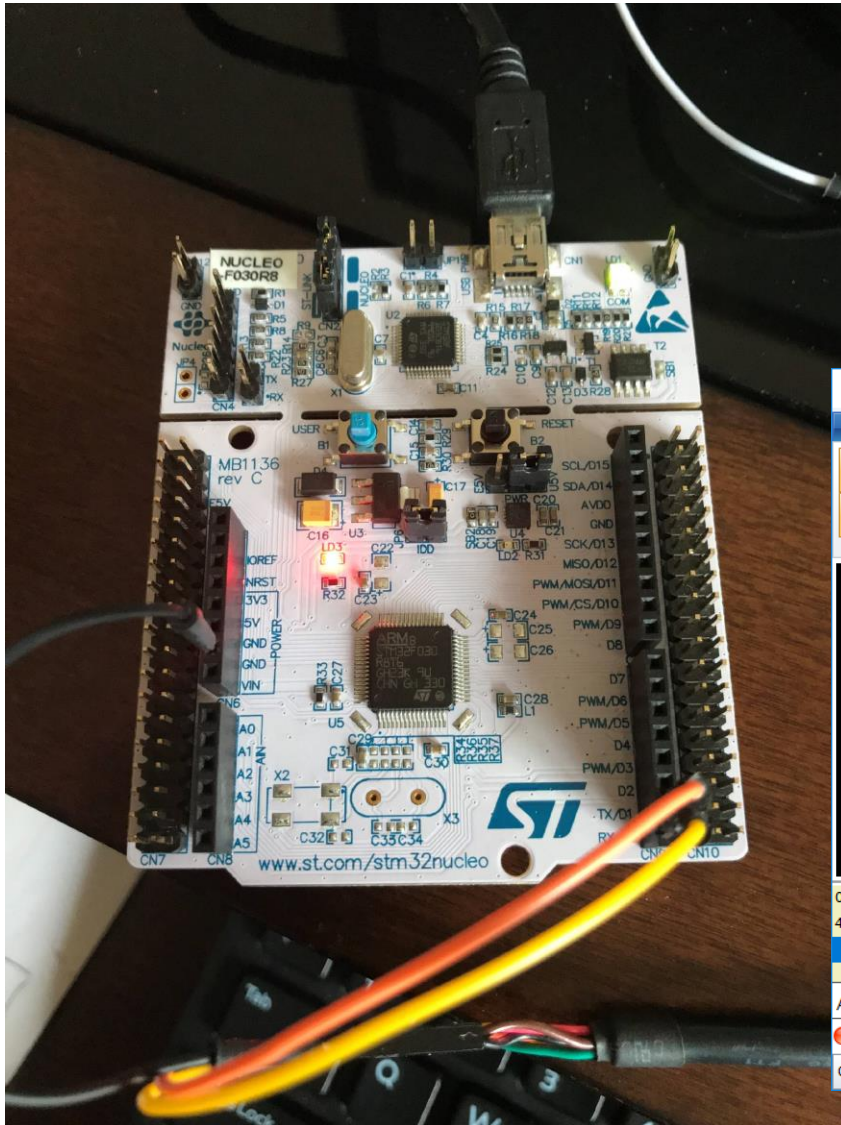
UART the Old-Fashioned CMSIS Way

```
stm32f030x8.h  stdout_USART.c
59 #define GPIO_AFRL_AFRL0_Pos 4* 0
60 #define GPIO_AFRL_AFRL1_Pos 4* 1
61 #define GPIO_AFRL_AFRL2_Pos 4* 2
62 #define GPIO_AFRL_AFRL3_Pos 4* 3
63 #define GPIO_AFRL_AFRL4_Pos 4* 4
64 #define GPIO_AFRL_AFRL5_Pos 4* 5
65 #define GPIO_AFRL_AFRL6_Pos 4* 6
66 #define GPIO_AFRL_AFRL7_Pos 4* 7
67
68 #define GPIO_AFRL_AFRH0_Pos 4* 0
69 #define GPIO_AFRL_AFRH9_Pos 4* 1
70 #define GPIO_AFRL_AFRH10_Pos 4* 2
71 #define GPIO_AFRL_AFRH11_Pos 4* 3
72 #define GPIO_AFRL_AFRH12_Pos 4* 4
73 #define GPIO_AFRL_AFRH13_Pos 4* 5
74 #define GPIO_AFRL_AFRH14_Pos 4* 6
75 #define GPIO_AFRL_AFRH15_Pos 4* 7
76 /**
77  * Initialize stdout
78  * \return 0 on success, or -1 on error.
79  */
80 int stdout_init (void)
81 {
82 #ifdef RTE_Compiler_IO_STDOUT_User
83   RCC->AHBENR |= (RCC_AHBENR_GPIOAEN);           // Enable GPIOA clock
84   RCC->APB1ENR |= (RCC_APB1ENR_USART2EN);        // Enable USART#2 clock
85   //Configure PA3 to USART2_RX, PA2 to USART2_TX
86   GPIOA->MODER  &= ~(GPIO_MODER_MODER2 | GPIO_MODER_MODER3);
87   GPIOA->MODER  |= (GPIO_MODER_MODER2_1 | GPIO_MODER_MODER3_1);
88   GPIOA->AFR[0] &= ~((15ul << GPIO_AFRL_AFRL2_Pos) | (15ul << GPIO_AFRL_AFRL3_Pos));
89   GPIOA->AFR[0] |= ((1ul << GPIO_AFRL_AFRL2_Pos) | (1ul << GPIO_AFRL_AFRL3_Pos));
90
91   USARTx->CR1 &= ~USART_CR1_PCE;                 //No parity control
92   USARTx->CR1 &= ~USART_CR2_STOP;                //1 stop bit
93   USARTx->CR1 &= ~USART_CR3_CTSE;                //No hardware flow control
94   USARTx->CR1 &= ~USART_CR3_RTSE;                //No hardware flow control
95   USARTx->BRR = (SystemCoreClock/115200);        //Baud Rate of 115200
96   USARTx->CR1 |= USART_CR1_RE;                   //Receive enable
97   USARTx->CR1 |= USART_CR1_TE;                   //Transmit enable
98   USARTx->CR1 |= USART_CR1_UE;                   //USART enable
99 #endif
100   return (0);
101 }
```



ARM Development Primer

Day 3's Done



```
73 //*****
74 /** MAIN - Display Messages
75 //*****
76 int main (void) {
77     SystemCoreClockConfigure ();
78     SystemCoreClockUpdate ();
79
80 #ifdef RTE_Compiler_IO_STDOUT_User
81     stdout_init ();
82 #endif
83
84     SysTick_Config(SystemCoreClock / 1000);
85
86     for (;;)
87     {
88         printf ("CEC STM32F030R8 UART\n\r");
89         delay_ms(100);
90     }
91
92 }
```

Serial Input/Output Monitor

File Edit View Configuration

ASCII HEX ASCII Send HEX Send

Input/Output

```
CEC STM32F030R8 UART
CEC STM32F030R8 UART
CEC STM32F030R8 UART
CEC STM32F030R8 UART
CEC STM32F030R8 UART
CEC STM32F030R8 UART
CEC STM32F030R8 UART
CEC STM32F030R8 UART
CEC STM32F030R8 UART
CEC STM32F030R8 UART
CEC STM32F030R8 UART
CEC STM32F030R8 UART
CEC STM32F030R8 UART
CEC STM32F030R8 UART
CEC STM32F030R8 UART
CEC STM32F030R8 UART
CEC STM32F030R8 UART
CEC STM32F030R8 UART
CEC STM32F030R8 UART
CEC STM32F030R8 UART
```

0D
43 45 43 20 53 54 4D 33 32 46 30 33 30 52 38 20 55 41 52 54 0A

ASCII Send

DSR DTR DCD RTS CTS RXD Ring TXD Error Break

COM3 8N1 115200 R 0 C 0 R3892 C Disconnect