

Hardware Abstraction & Low Level Layers for STM32

Eduardo Barrera

Julián Nieto

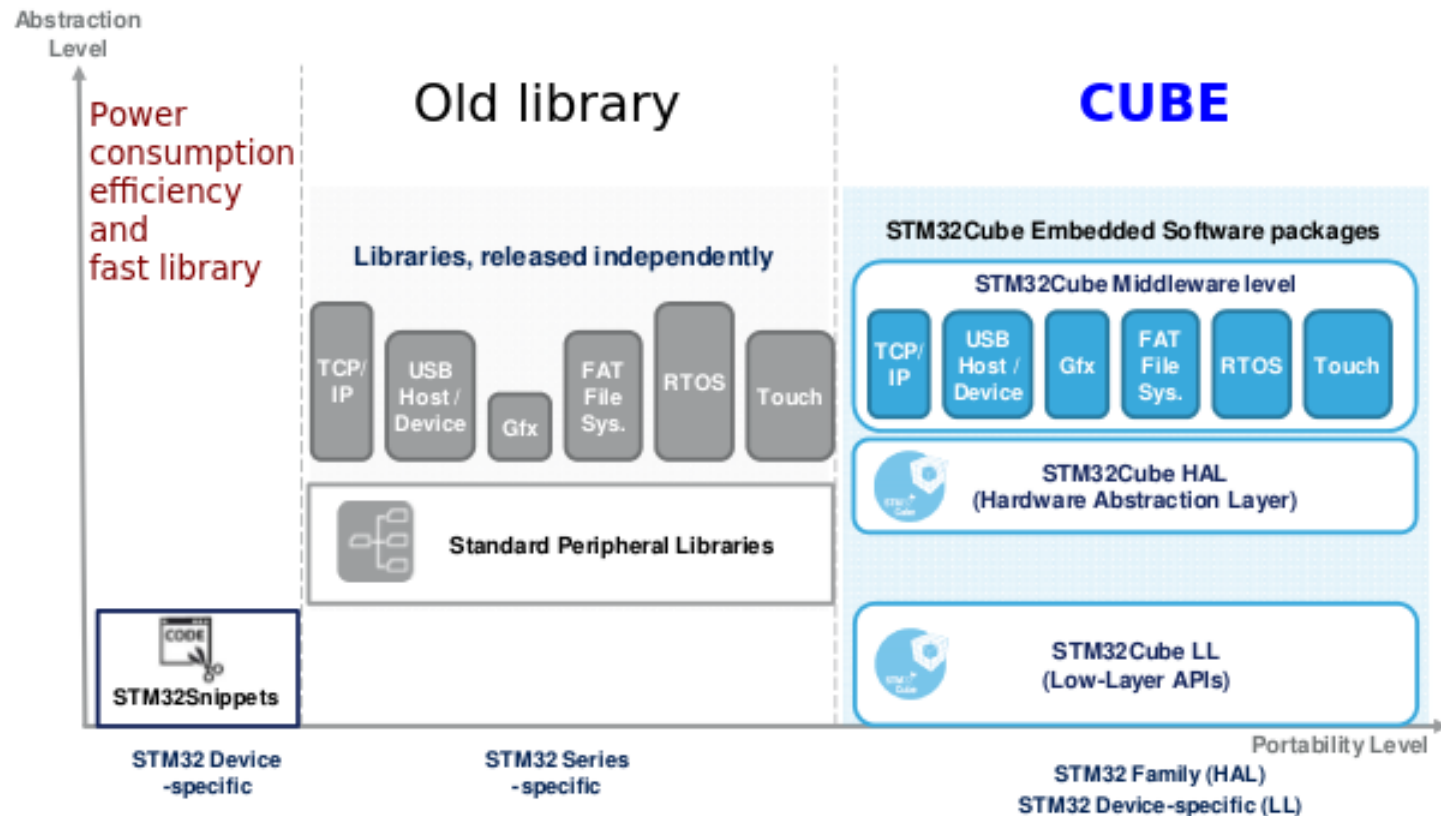
Mariano Ruiz

Problem to solve

- What is the best way to program a complex microprocessor-based system and its peripherals?
 - Directly using assembler (complex and device dependent)
 - Using code provided by the manufacturer for a specific device/model
 - Using software libraries with different optimizations levels
 - HAL and LL libraries are examples of this (STM32 CUBE)
 - In our course, we have decided to use the HAL provided by ST Microelectronics because it is the same library for all the 32-bit families




Three different approaches

ST Embedded software offer - Positioning

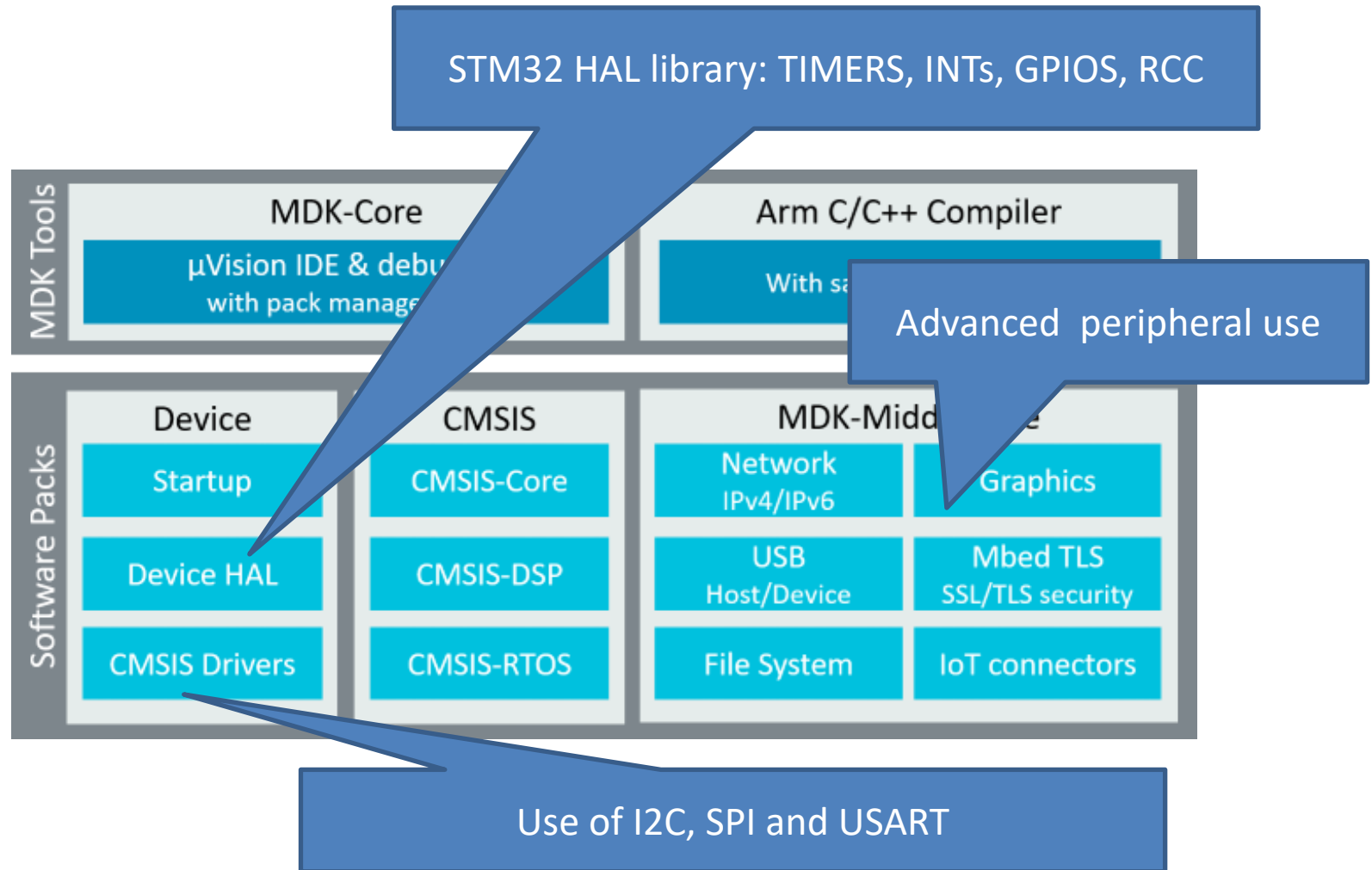


In Microprocessor Based Systems we will use the HAL API

ST Embedded software offer – Comparison

Offer		Portability	Optimization (Memory & Mips)	Easy	Readiness	Hardware coverage
 STM32Snippets			+++			+
 Standard Peripheral Library		++	++	+	++	+++
 STM32 Cube	HAL API	+++	+	++	+++	+++
	LL APIs		+++			++

Development of applications for ARM 32 devices using KEIL-MDK



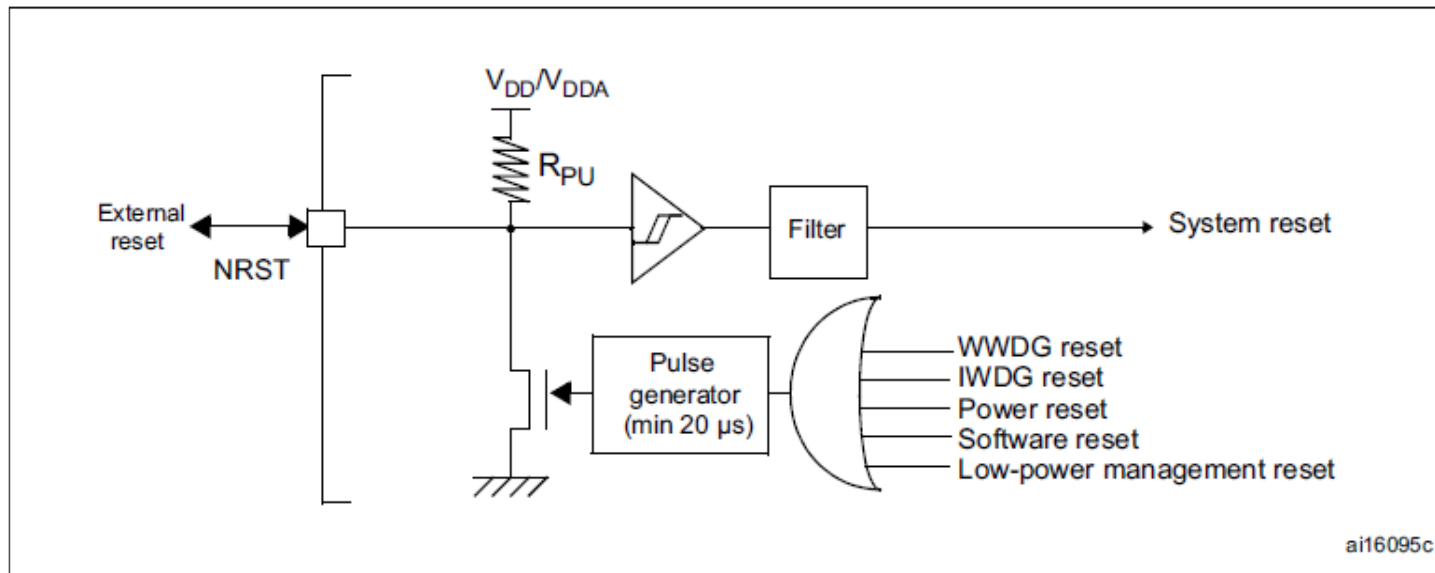
Software development in SBM HAL

- Keil Microvision for editing, compiling, linking, and debugging
- STM-32 HAL for configuring:
 - clocks and reset circuits (HAL RCC)
 - GPIO
 - Timers
 - Interrupts (NVIC and EXT1)
- CMSIS-Drivers for:
 - I2C and SPI
 - USART

RCC: Reset and Clock Control

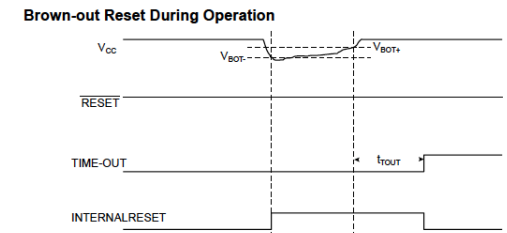
Reset circuit

- Reset sources acts on NRST pin.
- Reset service routine is fixed at address 0x00000004

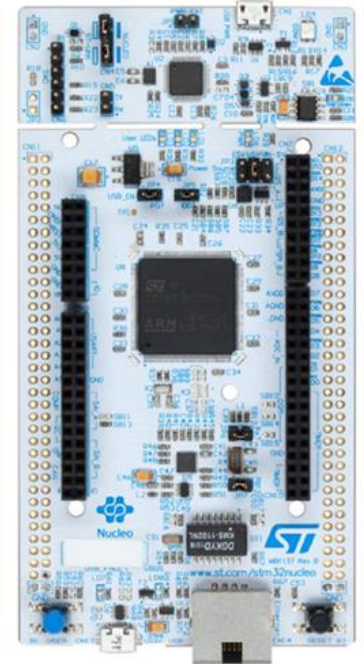


Reset circuit (3 types of reset sources)

- System Reset
 - Low level on external pin NRST
 - Window watchdog end of count condition (WWDG). Counter that must be refreshed within a specific time window. If not, the watchdog generates a system reset. WWDG is connected to APB1.
 - Independent watchdog end of count (IWDG). Triggers a system reset if a counter reaches a given timeout value (it uses the RC oscillator)
 - Software reset
 - Low power management reset
 - Reset when entering in the Standby mode
 - Reset when entering in Stop mode
- Power reset
 - Power-on/down reset (POR/PDR), “Brownout Reset” (BOR)
- Backup domain reset (Reset of RTC registers)



- B2 Button: generates reset on NRST signal
- Connection detail in NUCLEO-144 with STM32F429ZI (144 pins)



Creation of a default project with Keil Microvision

- Run Keil Microvision
- New Project
- Select the ST device
- Select the minimum software elements (next slide)
- Press OK
- Inspect the project created

Manage Run-Time Environment

Software Component	Sel.	Variant	Version	Description
Board Support		NUCLEO-F429ZI	1.0.0	STMicroelectronics NUCLEO-F429ZI Development Board
CMSIS				Cortex Microcontroller Software Interface Components
CORE	<input checked="" type="checkbox"/>		5.5.0	CMSIS-CORE for Cortex-M, SC000, SC300, ARMv8-M, ARMv8.1-M
DSP	<input type="checkbox"/>	Source	1.9.0-dev	CMSIS-DSP Library for Cortex-M, SC000, and SC300
NN Lib	<input type="checkbox"/>		3.0.0	CMSIS-NN Neural Network Library
RTOS (API)			1.0.0	CMSIS-RTOS API for Cortex-M, SC000, and SC300
RTOS2 (API)			2.1.3	CMSIS-RTOS API for Cortex-M, SC000, and SC300
CMSIS Driver				Unified Device Drivers compliant to CMSIS-Driver Specifications
Compiler		ARM Compiler	1.6.0	Compiler Extensions for ARM Compiler 5 and ARM Compiler 6
Device				Startup, System Setup
Startup	<input checked="" type="checkbox"/>		2.6.3	System Startup for STMicroelectronics STM32F4 Series
STM32Cube Framework (API)			1.1.0	STM32Cube Framework
Classic	<input checked="" type="checkbox"/>		1.7.9	Configuration via RTE_Device.h
STM32CubeMX	<input type="checkbox"/>		1.0.0	Configuration via STM32CubeMX
STM32Cube HAL				STM32F4xx Hardware Abstraction Layer (HAL) Drivers
ADC	<input type="checkbox"/>		1.7.9	Analog-to-digital converter (ADC) HAL driver
CAN	<input type="checkbox"/>		1.7.9	Controller area network (CAN) HAL driver
CRC	<input type="checkbox"/>		1.7.9	CRC calculation unit (CRC) HAL driver
Common	<input checked="" type="checkbox"/>		1.7.9	Common HAL driver
Cortex	<input checked="" type="checkbox"/>		1.7.9	Cortex HAL driver
DAC	<input type="checkbox"/>		1.7.9	Digital-to-analog converter (DAC) HAL driver
DCMI	<input type="checkbox"/>		1.7.9	Digital camera interface (DCMI) HAL driver
DMA2D	<input type="checkbox"/>		1.7.9	Chrom-Art Accelerator (DMA2D) HAL driver
DMA	<input type="checkbox"/>		1.7.9	DMA controller (DMA) HAL driver
ETH	<input type="checkbox"/>		1.7.9	Ethernet MAC (ETH) HAL driver
EXTI	<input type="checkbox"/>		1.7.9	External interrupts and event controller (EXTI) HAL driver
Flash	<input type="checkbox"/>		1.7.9	Embedded Flash memory (Flash) HAL driver
GPIO	<input checked="" type="checkbox"/>		1.7.9	General-purpose I/O (GPIO) HAL driver
HCD	<input type="checkbox"/>		1.7.9	USB Host controller (HCD) HAL driver
I2C	<input type="checkbox"/>		1.7.9	Inter-integrated circuit (I2C) HAL driver
I2S	<input type="checkbox"/>		1.7.9	I2S HAL driver
IRDA	<input type="checkbox"/>		1.7.9	IrDA HAL driver
IWDG	<input type="checkbox"/>		1.7.9	Independent watchdog (IWDG) HAL driver
LTDC	<input type="checkbox"/>		1.7.9	LCD-TFT Controller (LTDC) HAL driver
MMC	<input type="checkbox"/>		1.7.9	Multi Media Card (MMC) HAL driver
NAND	<input type="checkbox"/>		1.7.9	NAND Flash controller (NAND) HAL driver
NOR	<input type="checkbox"/>		1.7.9	NOR Flash controller (NOR) HAL driver
PC Card	<input type="checkbox"/>		1.7.9	PC Card controller HAL driver
PCD	<input type="checkbox"/>		1.7.9	USB Peripheral controller (PCD) HAL driver
PWR	<input checked="" type="checkbox"/>		1.7.9	Power controller (PWR) HAL driver
RCC	<input checked="" type="checkbox"/>		1.7.9	Reset and clock control (RCC) HAL driver
RNG	<input type="checkbox"/>		1.7.9	Random number generator (RNG) HAL driver
RTC	<input type="checkbox"/>		1.7.9	Real-time clock (RTC) HAL driver

Validation Output

Description

Resolve Select Packs Details OK Cancel

Elements to select

Project

Project: rccbasic



Target 1



Source Group 1



CMSIS



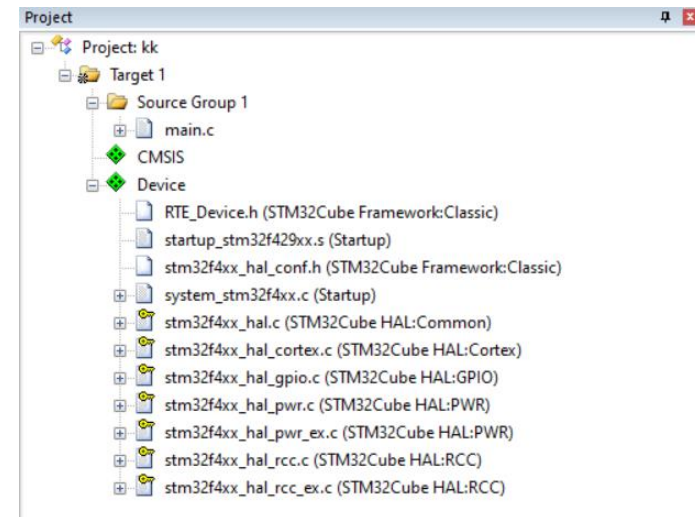
Device

- stm32f4xx_hal.c (STM32Cube HAL:Common)
- stm32f4xx_hal_cortex.c (STM32Cube HAL:Cortex)
- stm32f4xx_hal_gpio.c (STM32Cube HAL:GPIO)
- stm32f4xx_hal_pwr.c (STM32Cube HAL:PWR)
- stm32f4xx_hal_pwr_ex.c (STM32Cube HAL:PWR)
- stm32f4xx_hal_rcc.c (STM32Cube HAL:RCC)
- stm32f4xx_hal_rcc_ex.c (STM32Cube HAL:RCC)
- RTE_Device.h (STM32Cube Framework:Classic)
- startup_stm32f429xx.s (Startup)
- stm32f4xx_hal_conf.h (STM32Cube Framework:Classic)
- system_stm32f4xx.c (Startup)

Reset handler (in ARM assembly)

- See file startup_stm32f429xx.s
- After a hardware reset the Cortex Microprocessor starts the execution of the Reset Handler. The address of this handler is stored at address 0x00000004
- It calls first to “SystemInit” function and then calls “__main” (the C main() function)
- SystemInit is defined in “system_stm32f4xx.c” file (startup package)

```
; Reset handler
Reset_Handler PROC
                    EXPORT Reset_Handler [WEAK]
IMPORT SystemInit
IMPORT __main
    LDR R0, =SystemInit
    BLX R0
    LDR R0, =__main
    BX R0
ENDP
```

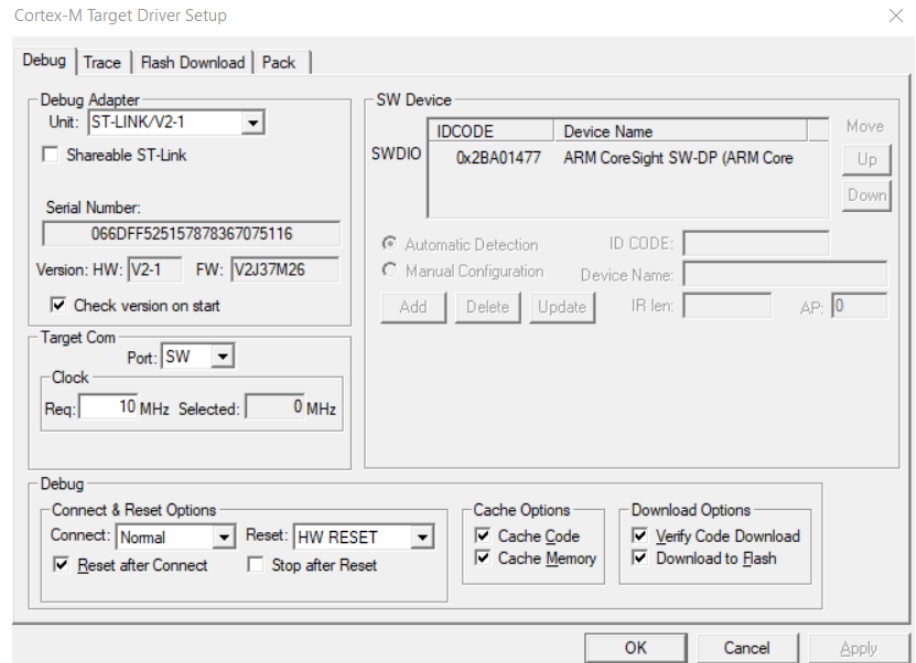


Reset Handler

- During the execution of the reset handler
 - The processor is using the internal clock
 - The different PINs are in the default state
- The user/developer has to configure the processor and the peripherals. This should be done in the main function
 - the first function to call in the main is “HAL_Init()” to gain access to the HAL Library

(Keil) Debug Connect Options

- Normal: stops CPU at the current executed instructions after connecting
- With Pre-reset: applies a hardware reset before connecting the device
- Under Reset: holds the hardware reset active while connecting to the device
- Without stop: connects and disconnects without explicitly stopping the CPU



(Keil) Debug Reset options

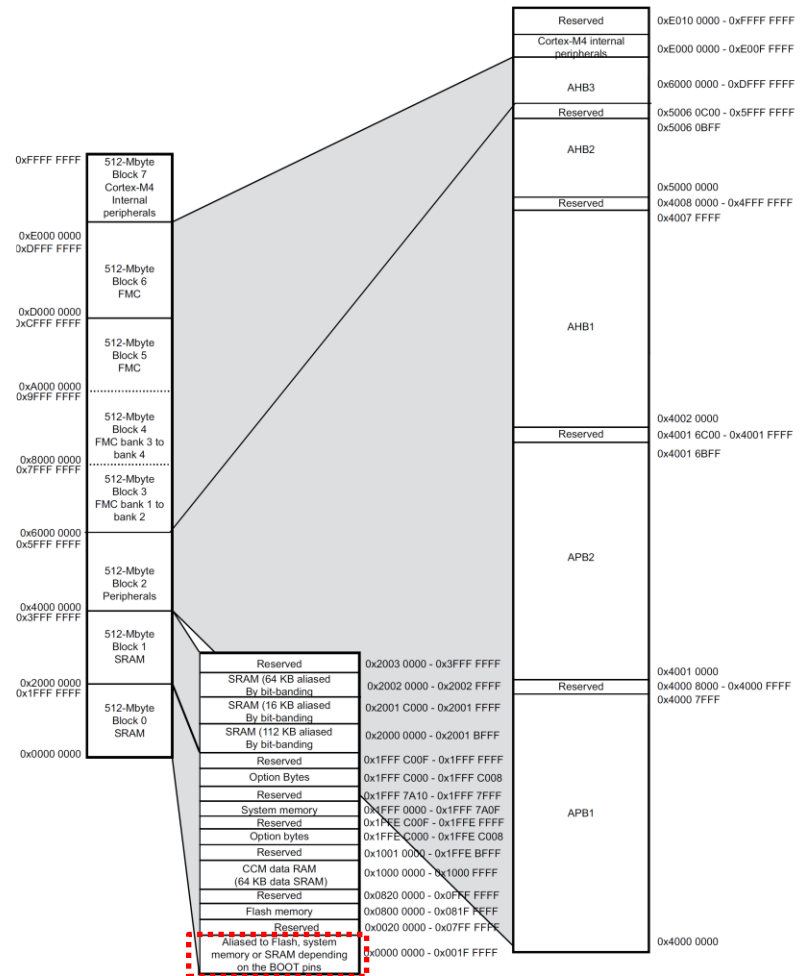
- Reset after connect: enables or disables the operation selected in the Reset drop-down list
- HW Reset: asserts the hardware reset signal
- SYSRESETREQ: performs a software reset (Cortex M and peripherals are reset)
- VECTRESET: Set the VECTRESET bit and only the cortex M is reset

Memory MAP and boot mode

Table 2. Boot modes

Boot mode selection pins		Boot mode	Aliasing
BOOT1	BOOT0		
x	0	Main Flash memory	Main Flash memory is selected as the boot space
0	1	System memory	System memory is selected as the boot space
1	1	Embedded SRAM	Embedded SRAM is selected as the boot space

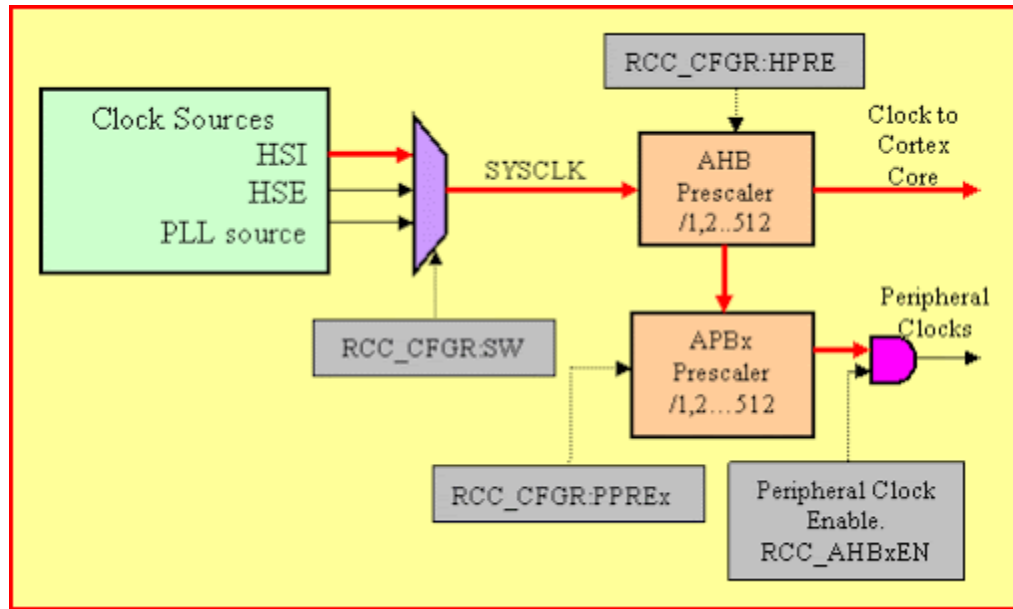
- NUCLEO 144:
Default BOOT0=0



MS30424V4

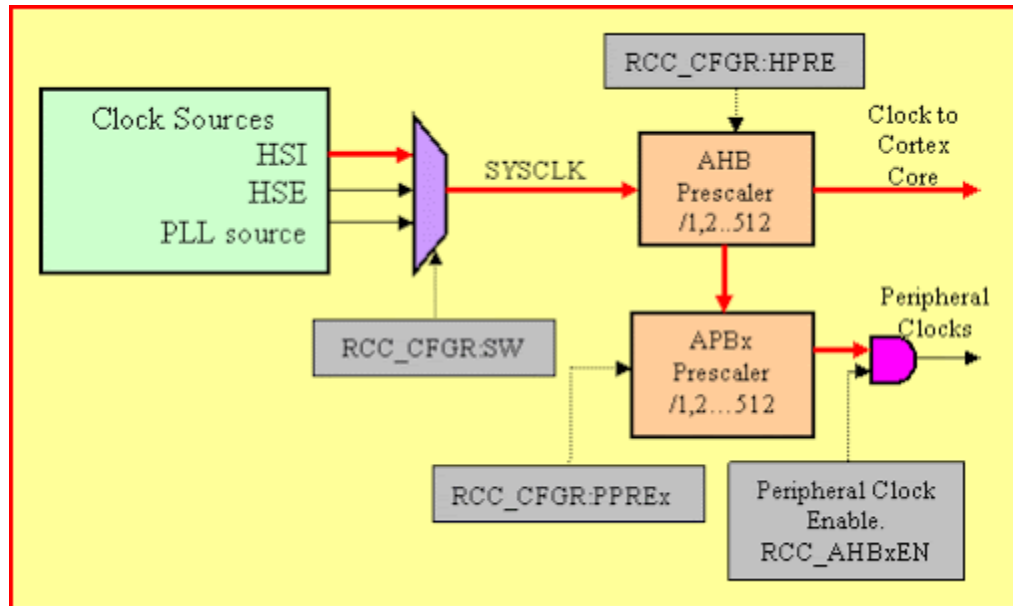
RCC: Clock Control

Basic clock circuitry for the STM32Fxxx



- The STM32F microcontroller system clock can come from one of three sources:
 - The high-speed internal clock (HSI)
 - The high-speed external clock (HSE)
 - The phase locked loop (PLL) clock
- The `RCC_CR` (CLOCK CONTROL) and `RCC_CFGR` (CLOCK CONFIGURATION) registers are used to select and enable the clock source

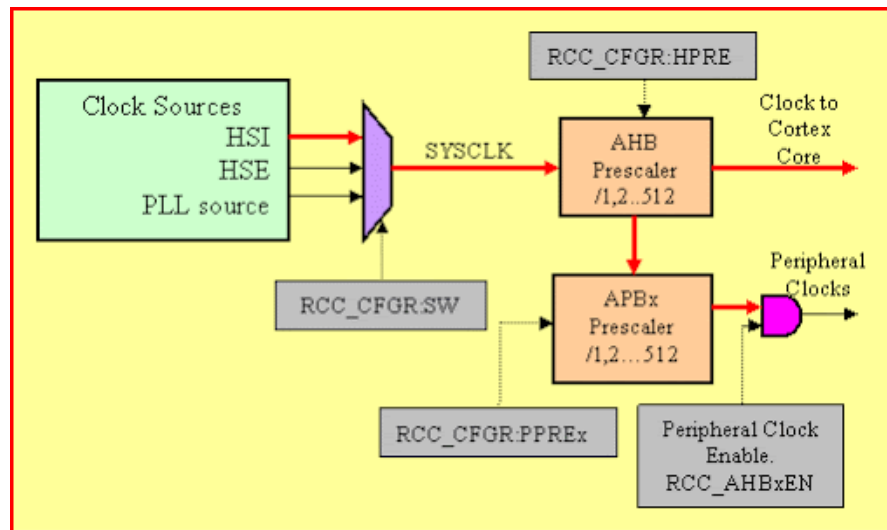
Basic clock circuitry for the STM32Fxxx



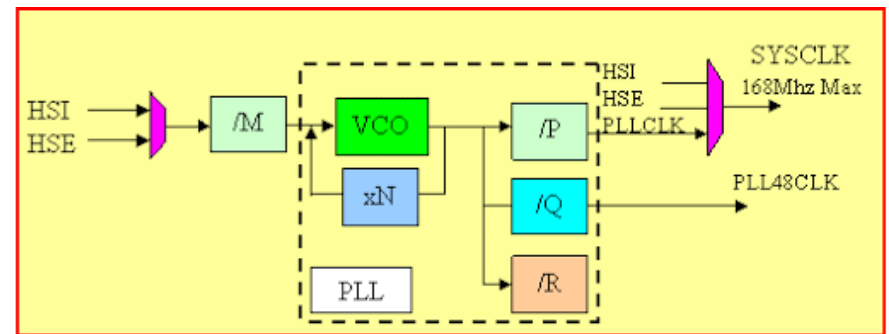
- After resetting the HSI (High-speed internal clock) is enabled
- HSE (High-speed external clock)

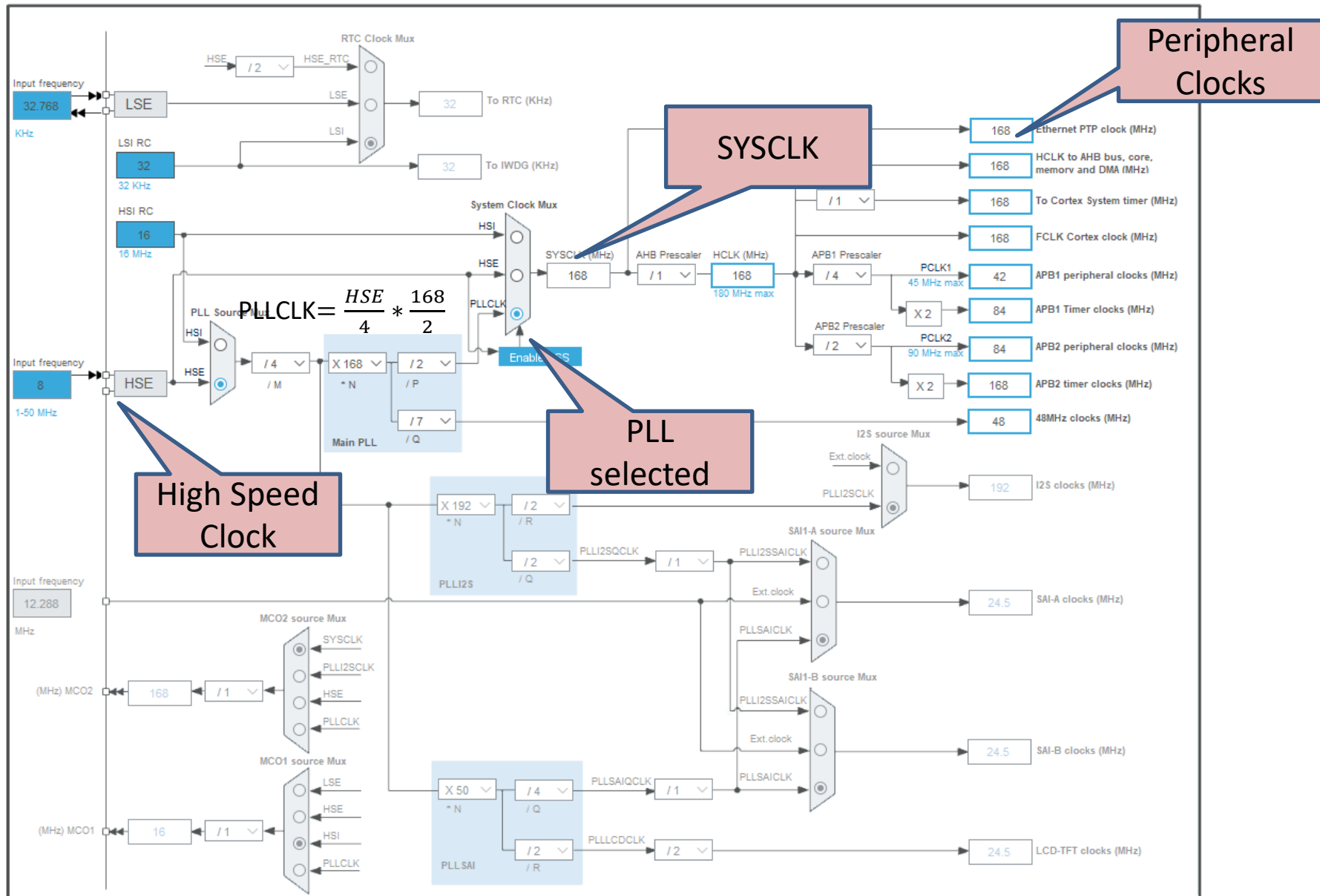
Basic clock circuitry for the STM32Fxxx

- Bus prescalers and peripheral clocks



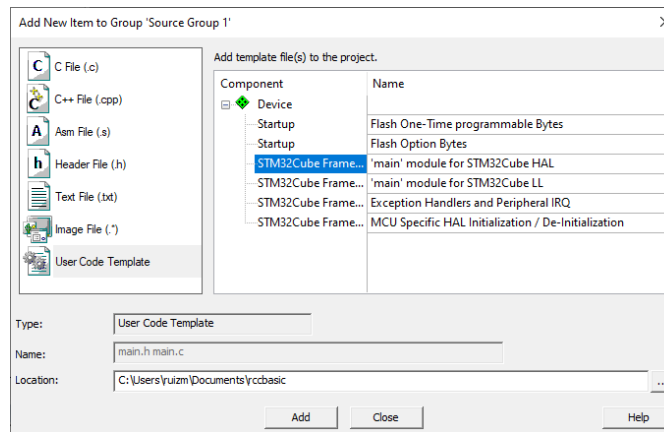
- Using the Phase Locked loop





In the Keil Microvision Project

- In “Source Group 1”->Add New Item (User Code Template)
- Select Device->'main' module for STM32Cube HAL->Add



- Display the content of “main.c”
- Inspect the content of SystemClock_Config function

Clock configuration I

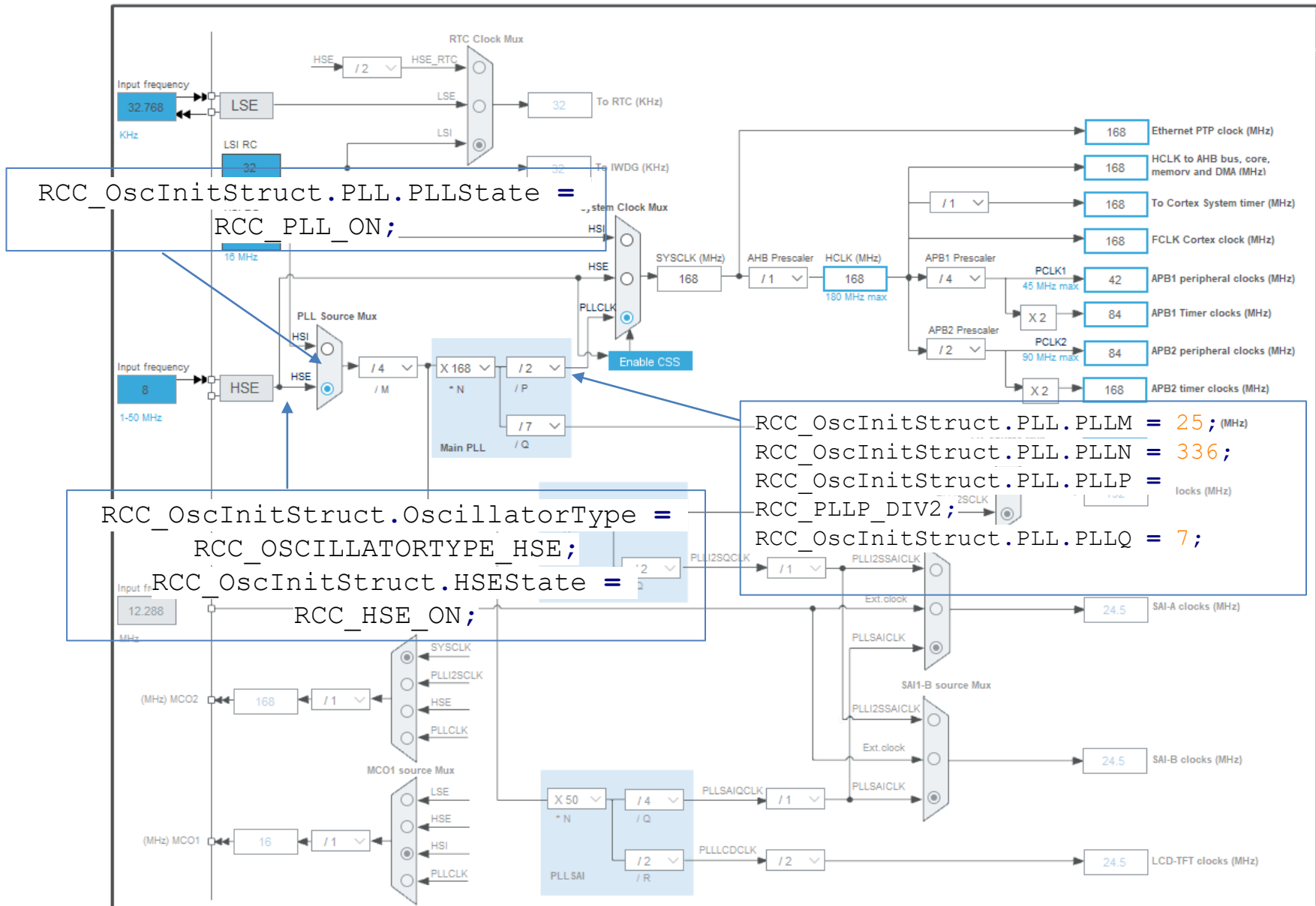
```
static void SystemClock_Config(void)
{
    RCC_ClkInitTypeDef RCC_ClkInitStruct;
    RCC_OscInitTypeDef RCC_OscInitStruct;

    /* Enable Power Control clock */
    __HAL_RCC_PWR_CLK_ENABLE();

    .....

    /* Enable HSE Oscillator and activate PLL with HSE as source
    */
    RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSE;
    RCC_OscInitStruct.HSEState = RCC_HSE_ON;
    RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
    RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSE;
    RCC_OscInitStruct.PLL.PLLM = 25;
    RCC_OscInitStruct.PLL.PLLN = 336;
    RCC_OscInitStruct.PLL.PLLP = RCC_PLLP_DIV2;
    RCC_OscInitStruct.PLL.PLLQ = 7;
```





```

if(HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
{
    /* Initialization Error */
    Error_Handler();
}

/* Select PLL as system clock source and configure the HCLK, PCLK1 and PCLK2
   clocks dividers */
RCC_ClkInitStruct.ClockType = (RCC_CLOCKTYPE_SYSCLK | RCC_CLOCKTYPE_HCLK |
RCC_CLOCKTYPE_PCLK1 | RCC_CLOCKTYPE_PCLK2);
RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_PLLCLK;
RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;
RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV4;
RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV2;
if(HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_5) != HAL_OK)
{
    /* Initialization Error */
    Error_Handler();
}

..
}

```

- 32f429 Datasheet

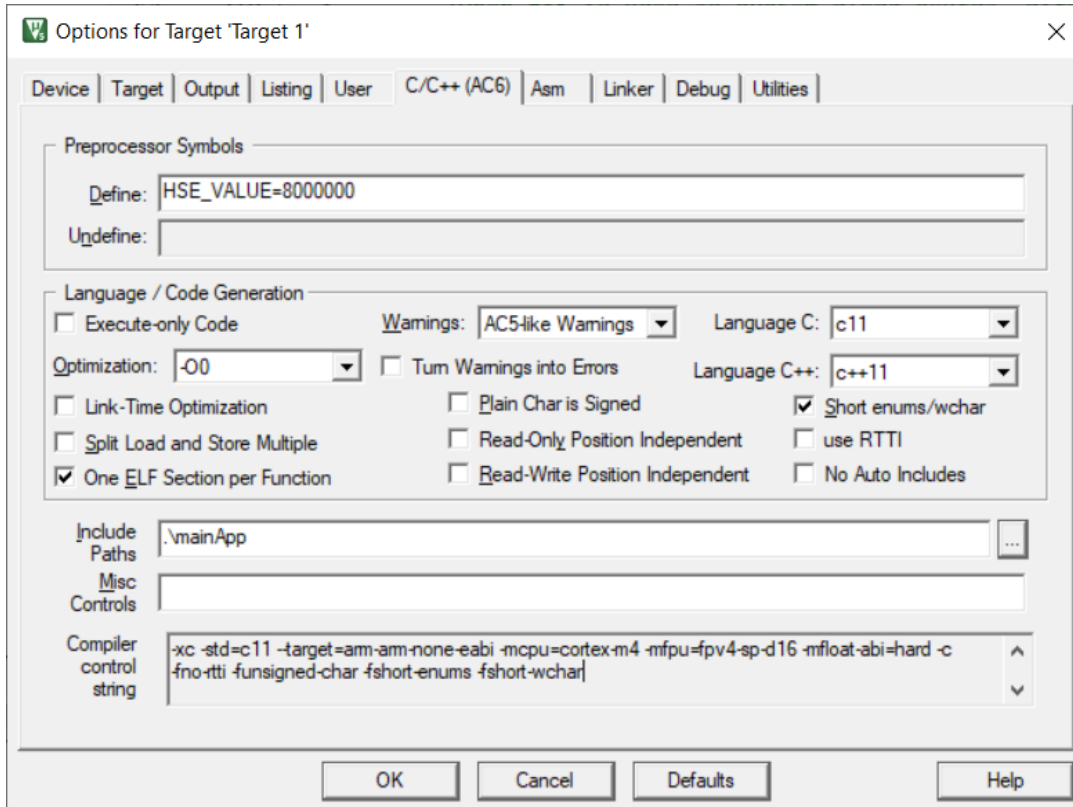
Figure 1: STM32MP135 pin configuration diagram. This diagram illustrates the pinout of the STM32MP135 microcontroller, showing the connection of various peripheral functions to the pins. The pins are organized into groups, and the functions are listed on the right side of the diagram. The diagram is color-coded to show different pin types: blue for digital, green for analog, and yellow for mixed-signal.

Key components and connections shown include:

- External memory controller (EMC):** Connected to NOR Flash, PC Card, and NAND Flash.
- Internal memory:** 1M5 Flash, SRAM 112 KB, SRAM 64 KB, and DRAM 1024 MB.
- Peripherals:** Arm Cortex-M4 180 MHz FPU, Ethernet MAC 10/100, USB, OTG HS, DMA2, DMA1, LCD-TFT, CHROM-ART, GPC PORT A through G, EXTI, SIO, SIO2, SIO3, SIO4, SIO5, SIO6, SIO7, SIO8, SIO9, SIO10, SIO11, SIO12, SIO13, SIO14, SIO15, SIO16, SIO17, SIO18, SIO19, SIO20, SIO21, SIO22, SIO23, SIO24, SIO25, SIO26, SIO27, SIO28, SIO29, SIO30, SIO31, SIO32, SIO33, SIO34, SIO35, SIO36, SIO37, SIO38, SIO39, SIO40, SIO41, SIO42, SIO43, SIO44, SIO45, SIO46, SIO47, SIO48, SIO49, SIO50, SIO51, SIO52, SIO53, SIO54, SIO55, SIO56, SIO57, SIO58, SIO59, SIO60, SIO61, SIO62, SIO63, SIO64, SIO65, SIO66, SIO67, SIO68, SIO69, SIO70, SIO71, SIO72, SIO73, SIO74, SIO75, SIO76, SIO77, SIO78, SIO79, SIO80, SIO81, SIO82, SIO83, SIO84, SIO85, SIO86, SIO87, SIO88, SIO89, SIO90, SIO91, SIO92, SIO93, SIO94, SIO95, SIO96, SIO97, SIO98, SIO99, SIO100, SIO101, SIO102, SIO103, SIO104, SIO105, SIO106, SIO107, SIO108, SIO109, SIO110, SIO111, SIO112, SIO113, SIO114, SIO115, SIO116, SIO117, SIO118, SIO119, SIO120, SIO121, SIO122, SIO123, SIO124, SIO125, SIO126, SIO127, SIO128, SIO129, SIO130, SIO131, SIO132, SIO133, SIO134, SIO135, SIO136, SIO137, SIO138, SIO139, SIO140, SIO141, SIO142, SIO143, SIO144, SIO145, SIO146, SIO147, SIO148, SIO149, SIO150, SIO151, SIO152, SIO153, SIO154, SIO155, SIO156, SIO157, SIO158, SIO159, SIO160, SIO161, SIO162, SIO163, SIO164, SIO165, SIO166, SIO167, SIO168, SIO169, SIO170, SIO171, SIO172, SIO173, SIO174, SIO175, SIO176, SIO177, SIO178, SIO179, SIO180, SIO181, SIO182, SIO183, SIO184, SIO185, SIO186, SIO187, SIO188, SIO189, SIO190, SIO191, SIO192, SIO193, SIO194, SIO195, SIO196, SIO197, SIO198, SIO199, SIO200, SIO201, SIO202, SIO203, SIO204, SIO205, SIO206, SIO207, SIO208, SIO209, SIO210, SIO211, SIO212, SIO213, SIO214, SIO215, SIO216, SIO217, SIO218, SIO219, SIO220, SIO221, SIO222, SIO223, SIO224, SIO225, SIO226, SIO227, SIO228, SIO229, SIO230, SIO231, SIO232, SIO233, SIO234, SIO235, SIO236, SIO237, SIO238, SIO239, SIO240, SIO241, SIO242, SIO243, SIO244, SIO245, SIO246, SIO247, SIO248, SIO249, SIO250, SIO251, SIO252, SIO253, SIO254, SIO255, SIO256, SIO257, SIO258, SIO259, SIO260, SIO261, SIO262, SIO263, SIO264, SIO265, SIO266, SIO267, SIO268, SIO269, SIO270, SIO271, SIO272, SIO273, SIO274, SIO275, SIO276, SIO277, SIO278, SIO279, SIO280, SIO281, SIO282, SIO283, SIO284, SIO285, SIO286, SIO287, SIO288, SIO289, SIO290, SIO291, SIO292, SIO293, SIO294, SIO295, SIO296, SIO297, SIO298, SIO299, SIO300, SIO301, SIO302, SIO303, SIO304, SIO305, SIO306, SIO307, SIO308, SIO309, SIO310, SIO311, SIO312, SIO313, SIO314, SIO315, SIO316, SIO317, SIO318, SIO319, SIO320, SIO321, SIO322, SIO323, SIO324, SIO325, SIO326, SIO327, SIO328, SIO329, SIO330, SIO331, SIO332, SIO333, SIO334, SIO335, SIO336, SIO337, SIO338, SIO339, SIO340, SIO341, SIO342, SIO343, SIO344, SIO345, SIO346, SIO347, SIO348, SIO349, SIO350, SIO351, SIO352, SIO353, SIO354, SIO355, SIO356, SIO357, SIO358, SIO359, SIO360, SIO361, SIO362, SIO363, SIO364, SIO365, SIO366, SIO367, SIO368, SIO369, SIO370, SIO371, SIO372, SIO373, SIO374, SIO375, SIO376, SIO377, SIO378, SIO379, SIO380, SIO381, SIO382, SIO383, SIO384, SIO385, SIO386, SIO387, SIO388, SIO389, SIO390, SIO391, SIO392, SIO393, SIO394, SIO395, SIO396, SIO397, SIO398, SIO399, SIO400, SIO401, SIO402, SIO403, SIO404, SIO405, SIO406, SIO407, SIO408, SIO409, SIO410, SIO411, SIO412, SIO413, SIO414, SIO415, SIO416, SIO417, SIO418, SIO419, SIO420, SIO421, SIO422, SIO423, SIO424, SIO425, SIO426, SIO427, SIO428, SIO429, SIO430, SIO431, SIO432, SIO433, SIO434, SIO435, SIO436, SIO437, SIO438, SIO439, SIO440, SIO441, SIO442, SIO443, SIO444, SIO445, SIO446, SIO447, SIO448, SIO449, SIO450, SIO451, SIO452, SIO453, SIO454, SIO455, SIO456, SIO457, SIO458, SIO459, SIO460, SIO461, SIO462, SIO463, SIO464, SIO465, SIO466, SIO467, SIO468, SIO469, SIO470, SIO471, SIO472, SIO473, SIO474, SIO475, SIO476, SIO477, SIO478, SIO479, SIO480, SIO481, SIO482, SIO483, SIO484, SIO485, SIO486, SIO487, SIO488, SIO489, SIO490, SIO491, SIO492, SIO493, SIO494, SIO495, SIO496, SIO497, SIO498, SIO499, SIO500, SIO501, SIO502, SIO503, SIO504, SIO505, SIO506, SIO507, SIO508, SIO509, SIO510, SIO511, SIO512, SIO513, SIO514, SIO515, SIO516, SIO517, SIO518, SIO519, SIO520, SIO521, SIO522, SIO523, SIO524, SIO525, SIO526, SIO527, SIO528, SIO529, SIO530, SIO531, SIO532, SIO533, SIO534, SIO535, SIO536, SIO537, SIO538, SIO539, SIO540, SIO541, SIO542, SIO543, SIO544, SIO545, SIO546, SIO547, SIO548, SIO549, SIO550, SIO551, SIO552, SIO553, SIO554, SIO555, SIO556, SIO557, SIO558, SIO559, SIO560, SIO561, SIO562, SIO563, SIO564, SIO565, SIO566, SIO567, SIO568, SIO569, SIO570, SIO571, SIO572, SIO573, SIO574, SIO575, SIO576, SIO577, SIO578, SIO579, SIO580, SIO581, SIO582, SIO583, SIO584, SIO585, SIO586, SIO587, SIO588, SIO589, SIO590, SIO591, SIO592, SIO593, SIO594, SIO595, SIO596, SIO597, SIO598, SIO599, SIO600, SIO601, SIO602, SIO603, SIO604, SIO605, SIO606, SIO607, SIO608, SIO609, SIO610, SIO611, SIO612, SIO613, SIO614, SIO615, SIO616, SIO617, SIO618, SIO619, SIO620, SIO621, SIO622, SIO623, SIO624, SIO625, SIO626, SIO627, SIO628, SIO629, SIO630, SIO631, SIO632, SIO633, SIO634, SIO635, SIO636, SIO637, SIO638,



Defining the HSE_VALUE in Microvision



- This is equivalent to:
 - `#define HSE_VALUE 8000000`

HAL parameters “customization”

Project

Project: rccbasic

Target 1

Source Group 1

CMSIS

Device

- stm32f4xx_hal.c (STM32Cube HAL:Common)
- stm32f4xx_hal_cortex.c (STM32Cube HAL:Cortex)
- stm32f4xx_hal_gpio.c (STM32Cube HAL:GPIO)
- stm32f4xx_hal_pwr.c (STM32Cube HAL:PWR)
- stm32f4xx_hal_pwr_ex.c (STM32Cube HAL:PWR)
- stm32f4xx_hal_rcc.c (STM32Cube HAL:RCC)
- stm32f4xx_hal_rcc_ex.c (STM32Cube HAL:RCC)
- RTE_Device.h (STM32Cube Framework)
- startup_stm32f429xx.s (Startup)
- stm32f4xx_hal_conf.h (STM32Cube HAL:RCC)
- system_stm32f4xx.c (Startup)

stm32f4xx_hal_conf.h

```
210 * (when HSE is used as system clock source, directly or through the PLL).
211 */
212 #if !defined (HSE_VALUE)
213 #define HSE_VALUE (8000000U) /*< Value of the External oscillator in Hz */
214 #endif /* HSE_VALUE */
215
216 #if !defined (HSE_STARTUP_TIMEOUT)
217 #define HSE_STARTUP_TIMEOUT (100U) /*< Time out for HSE start up, in ms */
218 #endif /* HSE_STARTUP_TIMEOUT */
219
220 /**
221 * @brief Internal High Speed oscillator (HSI) value.
222 * This value is used by the RCC HAL module to compute the system frequency
223 * (when HSI is used as system clock source, directly or through the PLL).
224 */
225 #if !defined (HSI_VALUE)
226 #define HSI_VALUE (16000000U) /*< Value of the Internal oscillator in Hz*/
227 #endif /* HSI_VALUE */
228
229 /**
230 * @brief Internal Low Speed oscillator (LSI) value.
231 */
232 #if !defined (LSI_VALUE)
233 #define LSI_VALUE (32000U)
```

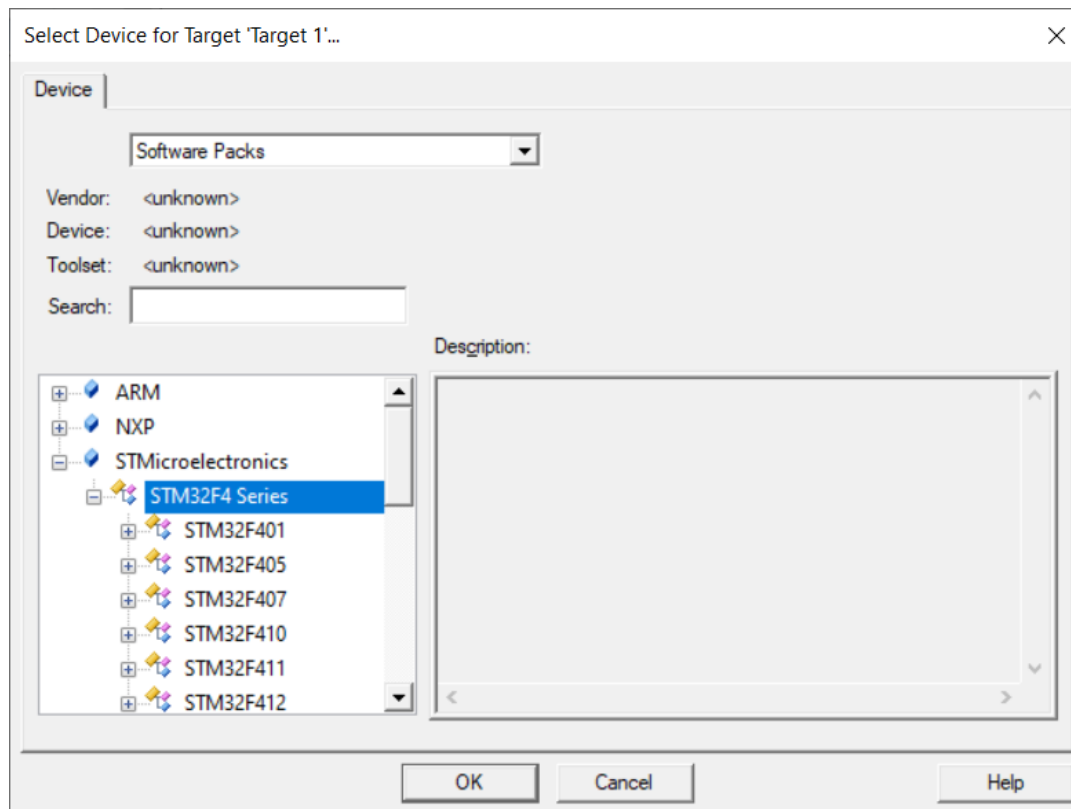
File local to your project

Keil Microvision

First Project from scratch

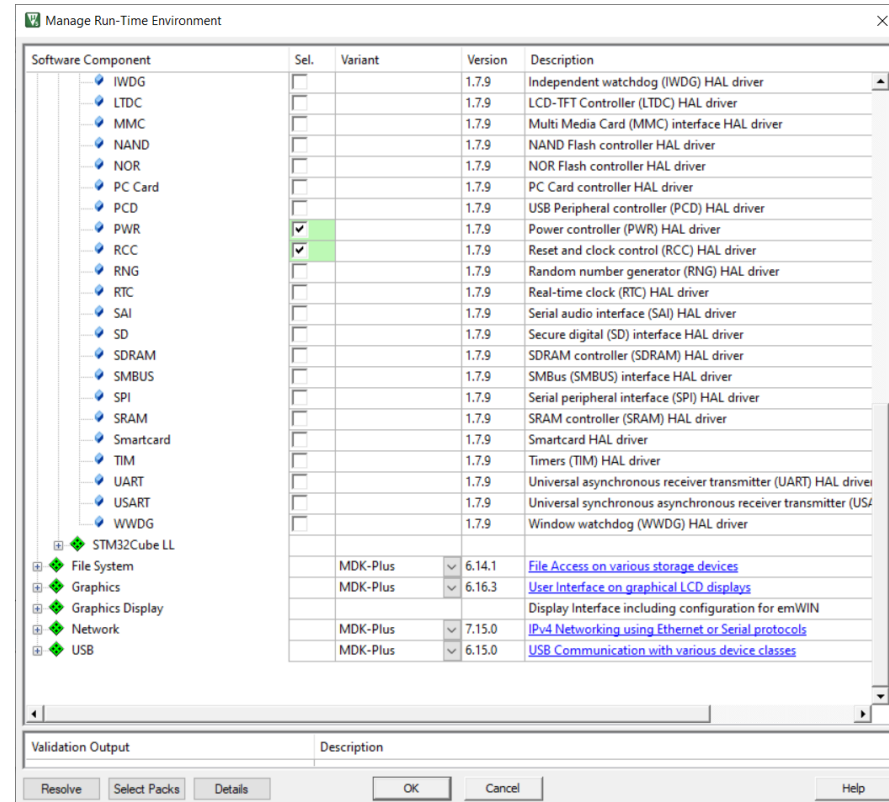
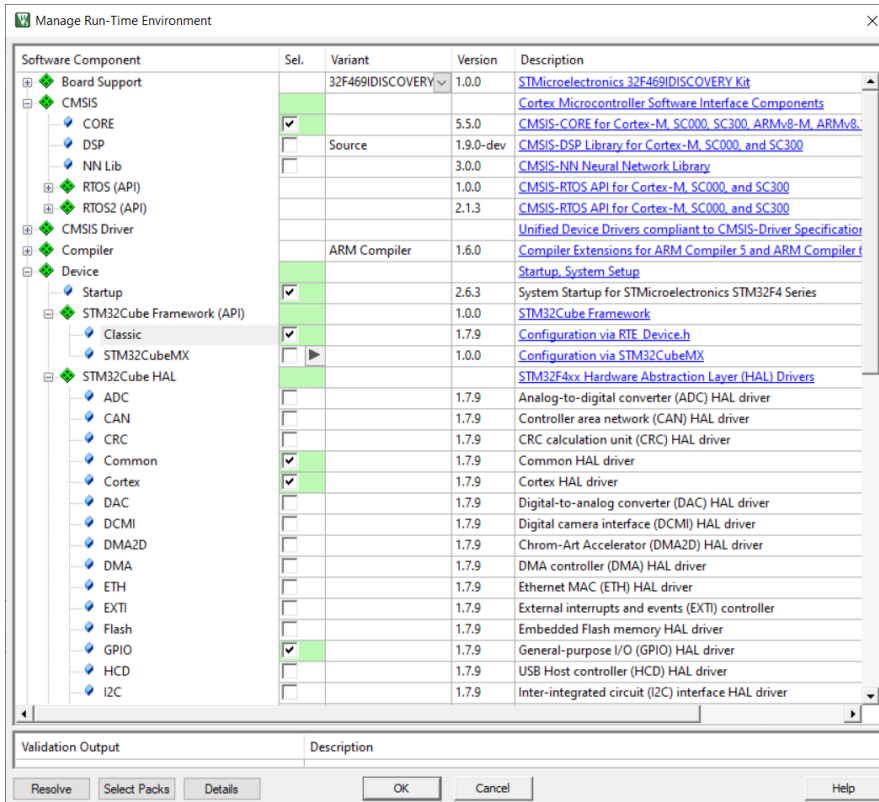
Steps I

- Project->New Microvision Project
- Device selection (STM32F429ZI)



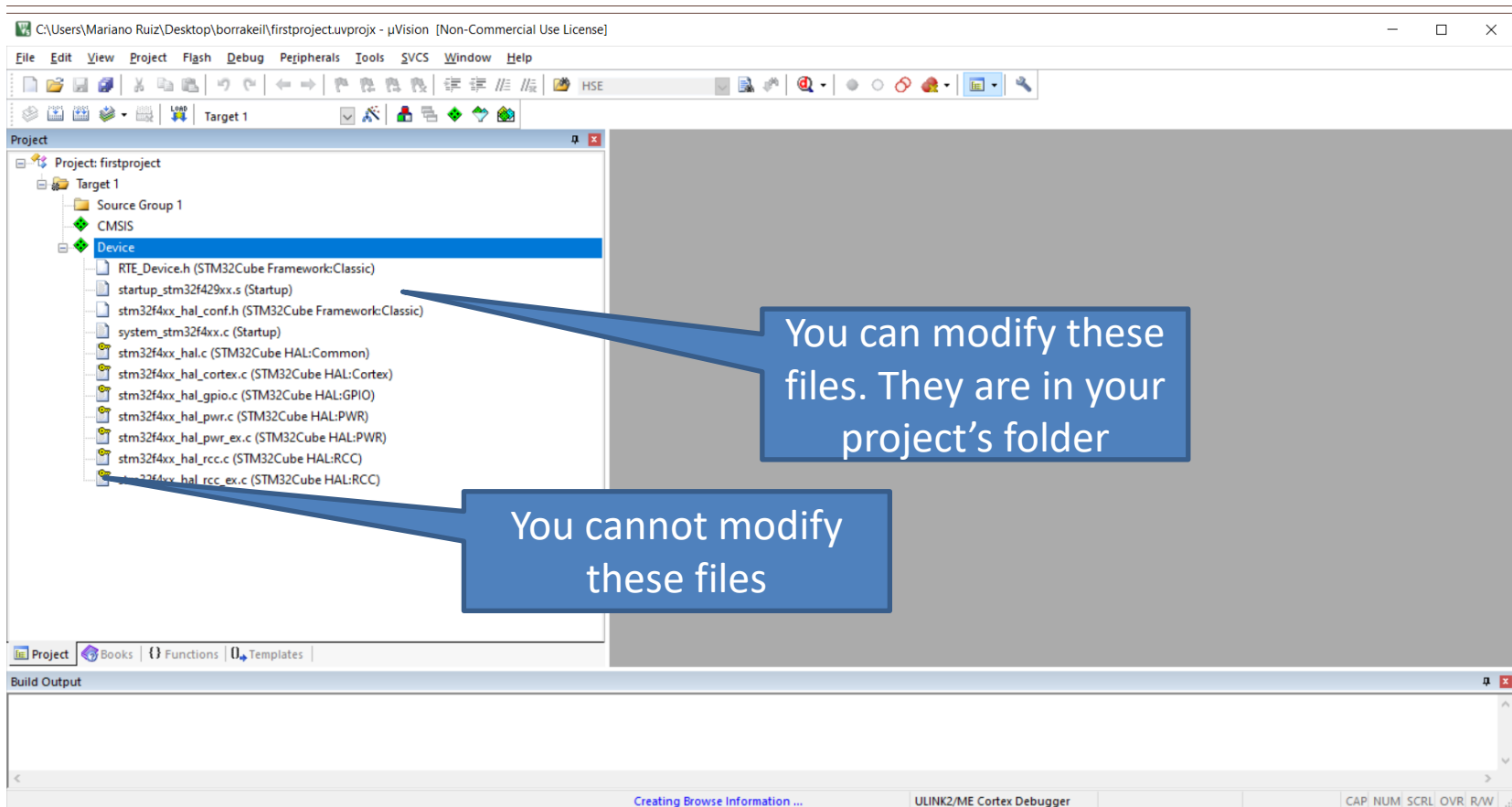
Steps II

• Configuration of the Run Time Environment



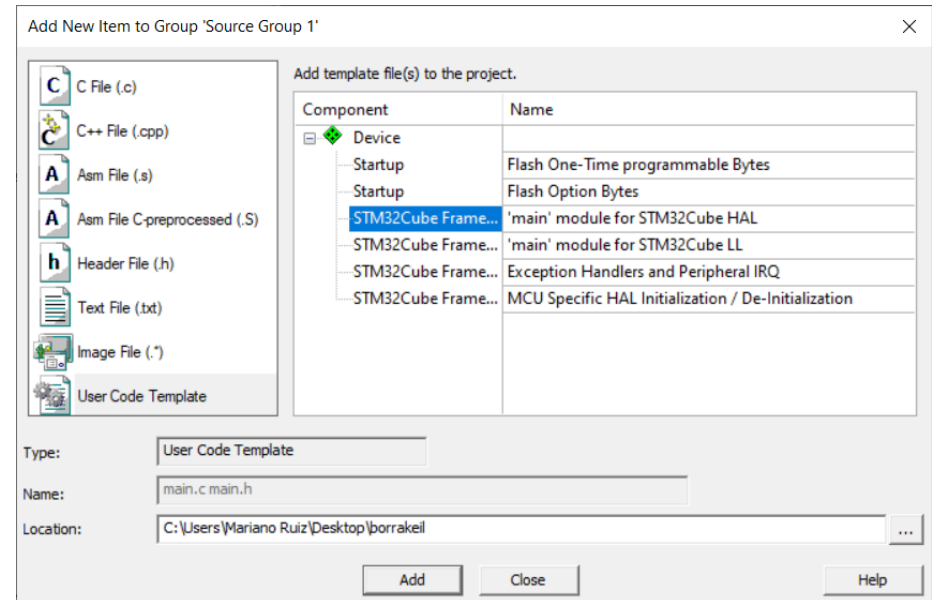
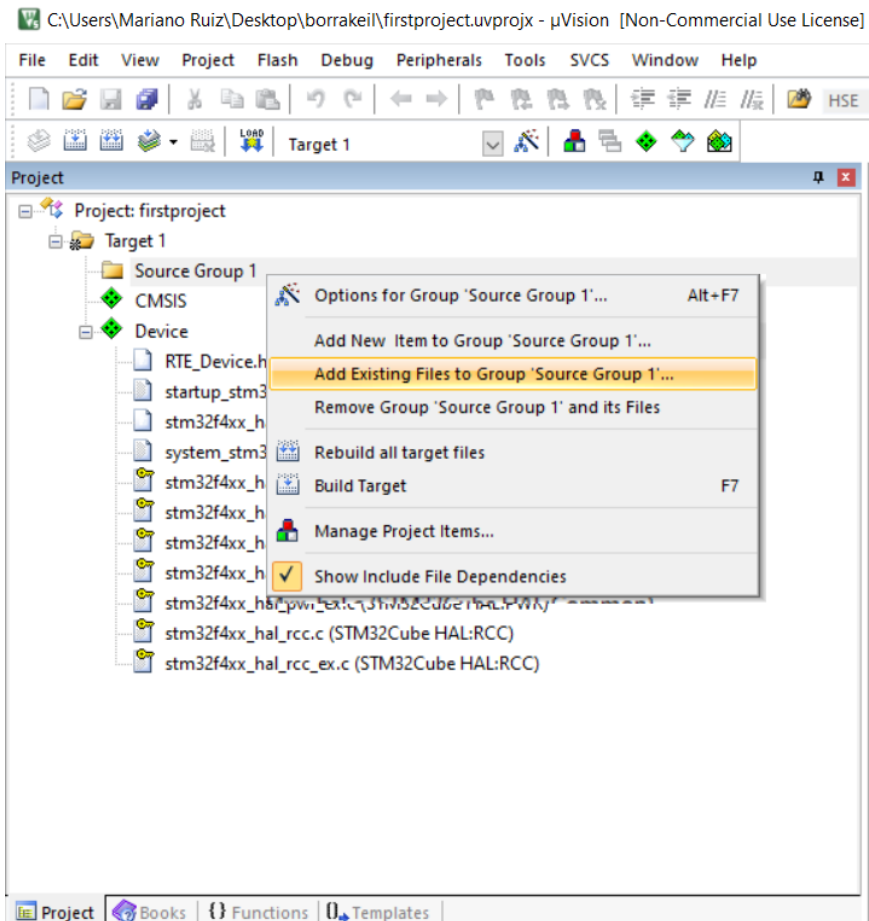
Step III

- Project created



Step IV

- Adding a basic “main.c” file

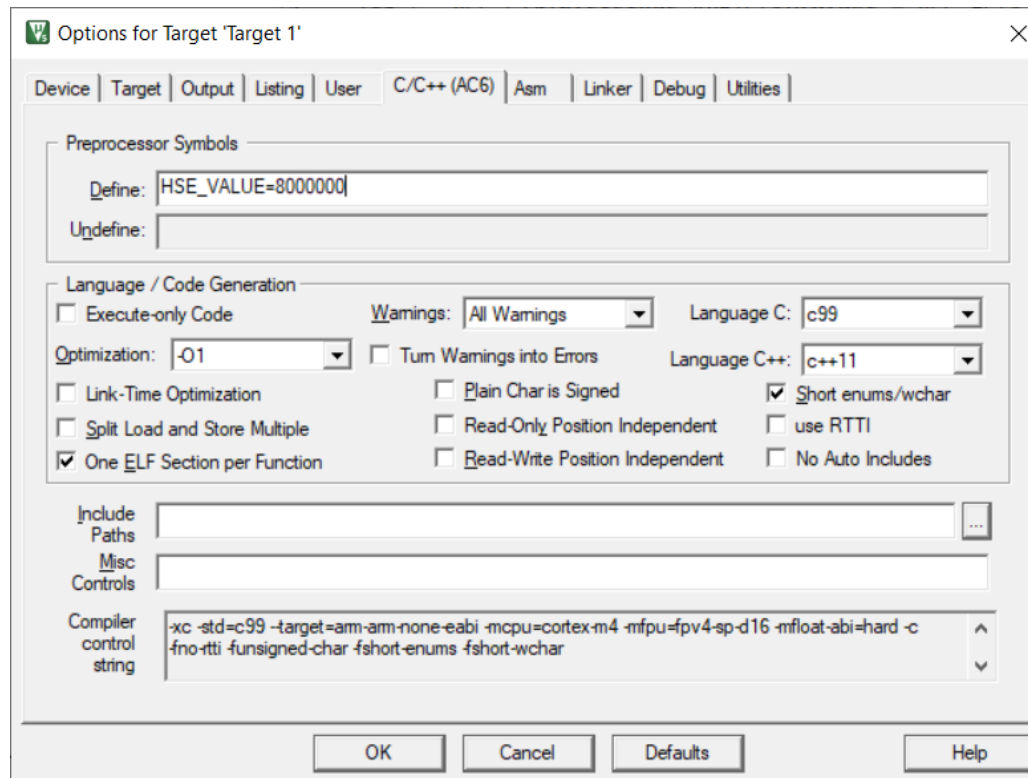


Step V

- Some details of “main.c”
 - RTE_Components.h added
 - main() calls to
 - HAL_Init()
 - SystemClock_Config();
 - SystemCoreClockUpdate();
 - SystemClock_Config() and Error_Handler functions code
 - Main code is an infinite loop

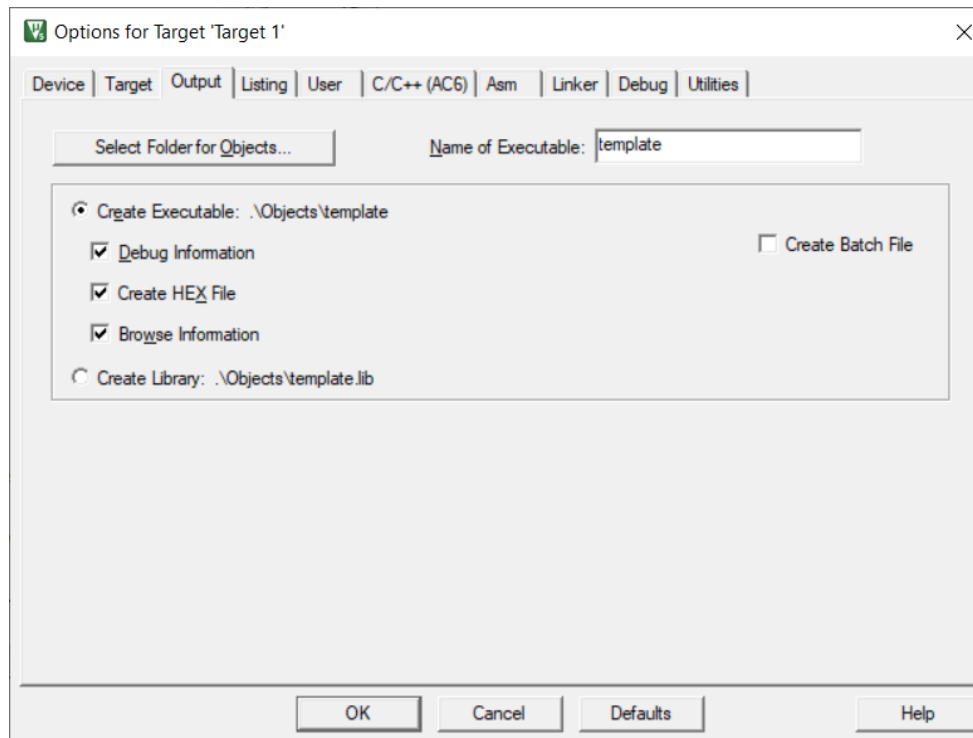
Step VI

- Add definition of HSE_VALUE to the Keil project
- Compile and debug



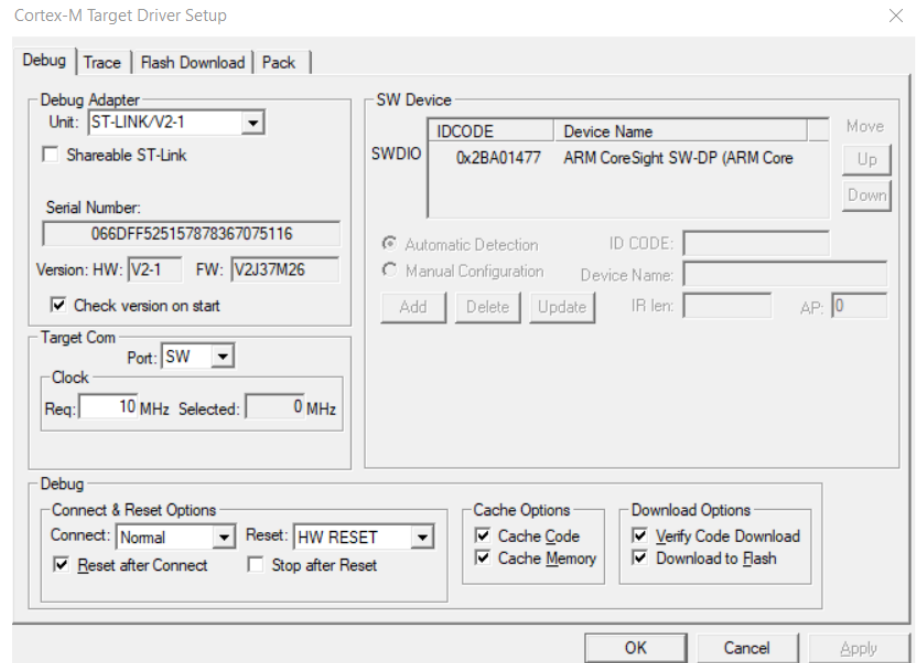
Activating the debug

- Select “Debug Information”
- For release version unselect it!



(Keil) Debug Connect Options

- Normal: stops CPU at the current executed instructions after connecting
- With Pre-reset: applies a hardware reset before connecting the device
- Under Reset: holds the hardware reset active while connecting to the device
- Without stop: connects and disconnects without explicitly stopping the CPU



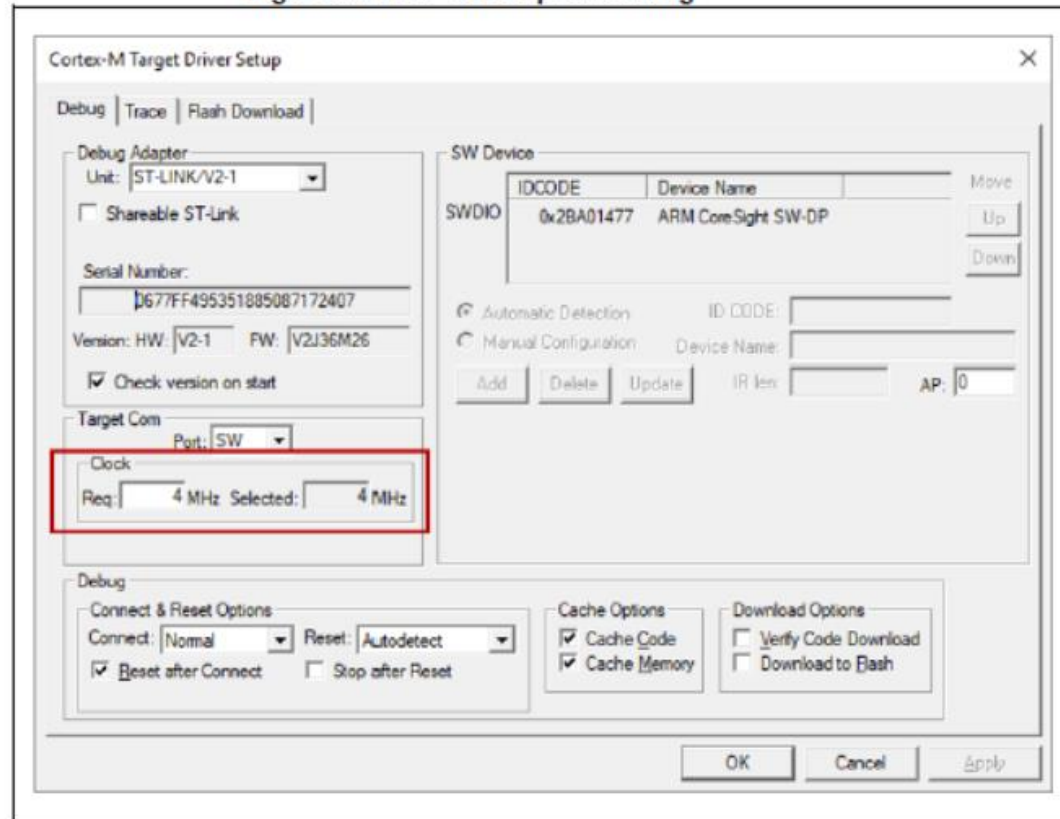
(Keil) Debug Reset options

- Reset after connect: enables or disables the operation selected in the Reset drop-down list
- HW Reset: asserts the hardware reset signal
- SYSRESETREQ: performs a software reset (Cortex M and peripherals are reset)
- VECTRESET: Set the VECTRESET bit and only the cortex M is reset

Serial Wire Debug (Speed configuration)

- Set to 4MHz maximum

Figure 36. Keil® SWD Speed Setting



The image shows the 'Cortex-M Target Driver Setup' dialog box in Keil, with the 'Debug' tab selected. The 'SW Device' section is highlighted with a red rectangle. In this section, the 'Clock' field is set to '4 MHz' and the 'Selected' field is also set to '4 MHz'. The 'SWDIO' field is set to '0x2BA01477' and the 'Device Name' is 'ARM CoreSight SW-DP'. The 'Automatic Detection' radio button is selected. The 'ID CODE' field is empty. The 'Device Name' field is empty. The 'IR len' field is empty. The 'AP' field is set to '0'. The 'Add', 'Delete', and 'Update' buttons are visible. The 'Port' is set to 'SW'. The 'Clock' field is highlighted with a red rectangle. The 'Selected' field is also highlighted with a red rectangle. The 'Debug' section at the bottom has 'Connect' set to 'Normal', 'Reset' set to 'Autodetect', and 'Reset after Connect' checked. The 'Cache Options' section has 'Cache Code' and 'Cache Memory' checked. The 'Download Options' section has 'Verify Code Download' and 'Download to Flash' unchecked. The 'OK', 'Cancel', and 'Apply' buttons are at the bottom right.

IDCODE	Device Name	Move
SWDIO 0x2BA01477	ARM CoreSight SW-DP	Up Down

Automatic Detection ID CODE:

Manual Configuration Device Name:

Add Delete Update IR len: AP: 0

Port: SW

Clock
Req: 4 MHz Selected: 4 MHz

Debug
Connect & Reset Options
Connect: Normal Reset: Autodetect
☒ Reset after Connect ☐ Stop after Reset


Cache Options
☒ Cache Code
☒ Cache Memory

Download Options
☐ Verify Code Download
☐ Download to Flash

OK Cancel Apply

Debug capabilities

Table 7. STM32 Series vs. debug capabilities

STM32 Series 	Cortex type	SWD	JTAG	ETM	SWO	Hardware breakpoints	Core Reset	MCO ⁽¹⁾
L0/F0	M0/0+	Yes	No	No	No	4	No	1
F1/L1/F2	M3	Yes	Yes	Yes ⁽²⁾	Yes	6	Yes	1
F3/F4/L4	M4	Yes	Yes	Yes ⁽²⁾	Yes	6	Yes	2 ⁽²⁾
F7/H7	M7	Yes	Yes	Yes ⁽²⁾	Yes	8	Yes	2 ⁽²⁾

1. Microcontroller Clock Output (refer to [Section 8.2: Microcontroller clock output \(MCO\) on page 87](#))
2. Depends on package size. Check availability in the Pin Allocation Table in the related datasheet.