

## Creating & Porting Cube-MX USB CDC Project


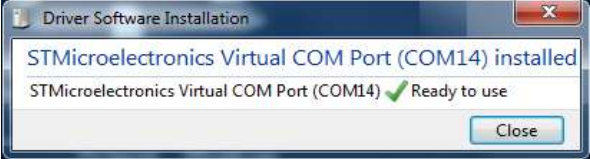
This section creates initialization code for a simple USB CDC project made with STM32-Cube-MX and **ports** it to a new SW4STM32 C/C++ perspective project. The MX generated code, builds without errors, flashes the orange LED3 on the F4Discovery board and allows a host (PC) to enumerate a virtual com port VCP when the F4Discovery is connected to a host via connector CN5.

Workspace: SW4stm32      C:\stm32\my\workspace\OfficeAc6\_2016.01\_Cube1.11.0\F4\discovery\UseMX  
 Stm32CubeMX      C:\stm32\my\workspace\ATSA54\STM32CubeMx-test1\USB-MX\_01

In	Intent	Method	Result expected
SW 1.	In SW4STM32 C/C++ perspective; create a new project for the relevant target board with NO HAL firmware or device drivers since these will come from MX	<p>[SW] <b>File→New→C Project</b>            Name: <b>USB-MX-01</b>            Project type: AC6...            Project Tools: AC6... <input type="button" value="Next"/> <input type="button" value="Next"/>            Series: STM32F4            Board: STM32F4DISCOVERY <input type="button" value="Next"/>            Firmware: <b>(o) No firmware</b> <input type="button" value="Finish"/>  <b>WAIT for task done!</b>            Watch lower right of window.</p>	<p>Project created and opened in SW4STM32</p> <p>The next step may fail if you don't wait and you will later have restart project creation from step1 all over again</p>
MX 2.	Create a project for your target board  (this text assumes that target is an F4Discovery)	<p>[MX] <b>File→New Project</b>            /MCU Selector(tab)\            Series: STM32F4            Lines: STM32F407/417            Package: LQFP100            MCUs List: <b>STM32F407VGTx</b>            /Board selector(tab)\            Type board: <b>Discovery</b>            MCU Series: STM32F4            Boards List: <b>STM32F4Discovery</b> <input type="button" value="OK"/> Button.</p>	<p>Project Created in MX</p> <p><b>In the chip-view CHECK F4Discovery pins LD6, LD5, LD3, LD4 are allocated to PD15,14,13,12.</b></p>
MX 3.	Set project settings	<p>[MX] <b>Project→Settings..</b>            /Project(tab)\            Name: <b>USB-MX-01</b>            Toolchain: <b>SW4STM32</b>            Minimum Heap Size: <b>0x800</b> (from 0x400)            Minimum Stack Size: <b>0x400</b> (from 0x200)            /Code Generator(tab)\  <b>(o) Copy only the necessary library files....</b> <input type="button" value="OK"/></p>	<p>Project named, and aimed at the SW4STM32 IDE</p>
MX 4.	Configure clock	<p>[MX] / pinout tab\            RCC(node)            High Speed Clock( HSE)= <b>Crystal/Ceramic</b>            /Clock Configuration(tab)\            Proceed as described above in Steps 5, 6 of section <b>Clock and Systick Configuration</b> p35</p>	<p>SYSCLK upped to maximum of 168MHz from HSE via PLL, instead of only 16MHz from HSI</p>
MX 5	Enable USB FS Device	<p>[MX] /Pinout(tab)\            Peripherals→USB_OTG_FS (node)            Mode: <b>Device only</b>            Middlewares→<b>USB_Device</b>  <b>Class for FS IP: Communication..</b></p>	<p>USB enabled</p> <p>As CDC</p>
MX 6	Save – so far	[MX]File→Save	-

MX 7.	Generate code and then open the MX project folder in the filesystem	Project→Generate code. In the success message that follows, click <b>Open Folder</b> to open the MX project folder in the filesystem, ready for next step...	Code available at: (for example) C:\stm32\my\workspace\ATSA54\STM32CubeMx-test1\MX-04
SW 8.	Switch perspective from STM32CubeMx to C/C++	Click <b>C/C++</b> Perspective tab	
FS 9.	Drag-Copy MX-generated inc/*.h <b>files</b> from MX into project folders of the same name in SW4STM32	From the filesystem window opened in the previous 2 steps, select all the <b>Inc/*.h</b> →drag→ <b>inc</b> folder in the SW4STM32 C/C++ perspective Project Explorer.	inc contains; mxconstants.h, stm32f4xx_hal_conf.h, stm32f4xx_it.h
FS 10.	Drag-Copy MX generated src/*.c <b>files</b> from MX into project folders of the same name in SW4STM32	From the filesystem window opened in a previous step, select all the <b>Src/*.c</b> →-drag-to→ <b>src</b> folder in the SW4STM32 C/C++ perspective Project Explorer.	src contains; main.c, stm32f4_hal_msp.c, stm32f4xx_it.c
FS 11.	Drag-Copy MX generated <b>Drivers folder</b> from MX into SW4STM32 project root folder	From the filesystem window opened in a previous step, select the <b>folder Drivers</b> →-drag-to→ <b>USB-MX-01</b> folder in the SW4STM32 C/C++ perspective Project Explorer.	Whole Drivers tree copied. (excludes unused folders like CMSIS/RTOS & DSP_Lib)
FS 12.	Drag-Copy MX generated <b>Middlewares folder</b> from MX to SW4STM32 project root folder	From the filesystem window opened earlier, select the <b>folder Middlewares</b> →-drag-to→ <b>USB-MX-01</b> folder in the SW4STM32 C/C++ perspective Project Explorer.	Middleware tree copied, but only relevant CDC and core files. Nice.
SW 13.	Add include paths to all folders containing .h files	[SW]right-click <b>USB-MX-01</b> →Properties→C/C++ General→Paths and Symbols→/Includes(tab)\--> <b>Add</b> button→ [x] Add to all configs, [x] Add to all languages [x] Is a workspace path <b>Workspace..</b> button--> <b>USB-MX-01/Drivers/STM32F4xx_HAL_Driver/Inc</b> <b>OK</b> <b>OK</b> repeat for... <b>USB-MX-01/Drivers/ CMSIS/Include</b> <b>USB-MX-01/Drivers/CMSIS/Device/ST/STM32F4xx/Include</b> <b>USB-MX-01/Middlewares/ST/STM32_USB_Device_Library/Class/CDC/Inc</b> <b>USB-MX-01/Middlewares/ST/ ST/STM32_USB_Device_Library/Core/Inc</b> <b>OK</b> <b>OK</b> <b>OK</b>	
SW 14.	In project properties, define the <b>symbol STM32F407xx</b> - for all languages and configs	[SW]right-click on project <b>USB-MX-01</b> →Properties→C/C++ General→Paths and Symbols→/Symbols(tab)\ → <b>Add..</b> button Name: <b>STM32F407xx</b> <b>[x]</b> Add to all configs <b>[x]</b> Add to all languages <b>OK</b> <b>OK</b>	
SW 15	In project properties ensure <b>source</b> locations include the Drivers <b>source</b> folders	[SW]right-click on <b>USB-MX-01</b> →Properties→C/C++ General→Paths and Symbols→/Source Location(tab)\ → <b>Add Folder..</b> → <b>Drivers/STM32F4xx_HAL_Driver/Src</b> <b>OK</b> <b>Add Folder..</b> → <b>Drivers/CMSIS/Device/ ST/STM32F4xx/Source</b> <b>OK</b>	

		Add Folder.. → <a href="#">Middlewares/ST/STM32_USB_Device_Library/Class/CDC/Src</a> OK Add Folder.. → <a href="#">Middlewares/ST/STM32_USB_Device_Library/Core/Src</a> OK OK	
SW 16.	Exclude duplicate startup from build	[SW]right-click on subfolder: <b>USB-MX-01/startup/startup_stm32.s</b> → Properties → [x] <b>Exclude from build.</b> OK	Includes usb startup startup_stm32f407xx.s in /drivers/CMSIS/Device/ ST/STM32F4xx/Source/ Templates/gcc/
SW 17.	Optional. Add code to make LED LD3(orange) flash, and output "Hello " from the USB CDC device to the host, every 20ms.	Edit /src/main.c , replace the empty infinite while(1) loop with this //inspired from <a href="#">here</a> <pre> while (1) {     HAL_Delay(20);     HAL_GPIO_TogglePin(GPIOD, LD3_Pin);     CDC_Transmit_FS("Hello ", 6); } </pre>	LD3orange LD4 green LD5 red LD6 blue //
SW 18	Optional. Add code to make LED LD6(blue) toggle when the device receives the character '2' (0x32) from the host.	Edit /src/usbd_conf.c , replace HAL_PCD_DataOutStageCallback() with this;- <pre> void HAL_PCD_DataOutStageCallback(PCD_HandleTypeDef *hpcd, uint8_t epnum) {     int len=0;     uint8_t* p=0;     if ((epnum==1) &amp;&amp; (hpcd-&gt;OUT_ep[epnum].xfer_count&gt;0))         len = hpcd-&gt;OUT_ep[epnum].xfer_count;      USB_LL_DataOutStage(hpcd-&gt;pData, epnum, hpcd-&gt;OUT_ep[epnum].xfer_buff);      if (len&gt;0)     {         p = hpcd-&gt;OUT_ep[epnum].xfer_buff;         if (p[0]=='2')             HAL_GPIO_TogglePin(GPIOD, LD6_Pin); //blue     } } </pre>	
SW 19.	Optional. to eliminate warnings; Cast USB strings as unsigned char pointers.	Edit /src/usbd_desc.c, replace <pre>#define USBD_MANUFACTURER_STRING "STMicroelectronics"</pre> With <pre>#define USBD_MANUFACTURER_STRING ((uint8_t*)"STMicroelectronics")</pre> Cast the other four nearby strings similarly.	
SW 20	Optional. Eliminate a warning about missing break statement	Edit <a href="#">Middlewares/ST/STM32_USB_Device_Library/Class/CDC/src/usbd_cdc.c</a> , function USBDCDC_Setup() Insert "break;" before default: near end of function.	One less warnings
SW 21	Optional. Eliminate warning about unused function	Edit /src/usbd_conf.c and /inc/usbd_conf.h. Move from .c to .h file the forward reference; <pre>void SystemClock_Config(void);</pre>	One less warnings.
SW 22.	Save	File->Save	Whole project saved
SW 23.	build	[SW]right-click: <b>USB-MX-01</b> → <b>Index</b> → <b>Rebuild.</b> Wait for index to rebuild.. in lower right of window [SW]right-click: <b>USB-MX-01</b> → <b>Clean Project.</b> Wait for clean to complete.. [SW]right-click: <b>USB-MX-01</b> → <b>Build Project.</b> Wait for build to complete	0 Errors. SysTick_IRQn undefined may be reported but if .elf has been created then delete these problems.

SW 24 1.	Run	Run→Debug As→AC6 STM32 C/C++ Application click the run  toolbar button.	Orange LED flashes or flickers. LD7 (green - near CN5 USB) should light.
FS 25	Test enumeration	Connect CN5(device) to PC(Host), and wait For first-time device <i>enumeration</i> .	 <p>The virtual com port successfully enumerates and filesystem [Windows Device Manager] shows [in the ports node] “STMicroelectronics Virtual COM Port (COM14)” (or other com port number).</p>
FS 26	Test data flows from device to host	Run a serial terminal program like puTTY (serial) and OPEN the relevant port e.g. COM14 (any settings work)	<p>The terminal should be able to connect to the COMx port that was made available in the previous step.</p> <p>“Hello “ should be seen arriving repeatedly from the F4 on the hosts terminal.</p>
FS 27	Test data flows from host to device	At the terminal type 1,2,3,4 to changes LED lit.	Not yet demonstrated...

\***PCD** means **USB Peripheral Controller Driver** – see [UM1725](#) section 1 Acronyms and Abbreviations.

The above procedure works – but not properly. Sending several bytes from the USB-Host to the CDC Device crashes the interface fatally. Using MX with HAL for USB Device CDC is not yet demonstrated as working.