



life.augmented

FOTA Develop with SPC5-STUDIO

Targeting SPC58 Chorus MCU

Neo JUNG

Agenda

1 SPC58 Chorus Family

2 SPC5-STUDIO

3 Background : FOTA

4 AFLS + FOTA

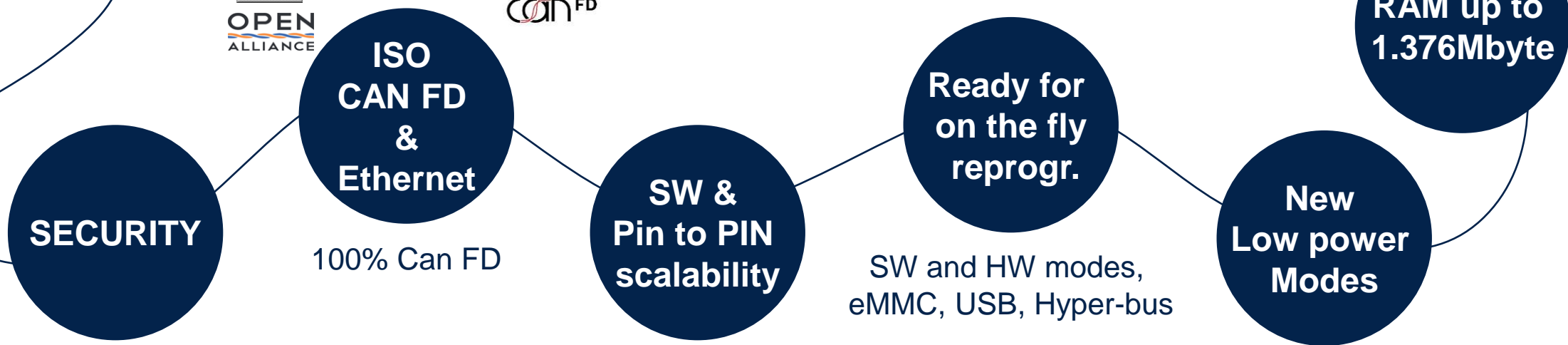
SPC58 Chorus Family



SPC58 Chorus Advantage



Ready Now



SECURITY

**ISO
CAN FD
&
Ethernet**

**SW &
Pin to PIN
scalability**

**Ready for
on the fly
reprog.**

**New
Low power
Modes**

**RAM up to
1.376Mbyte**

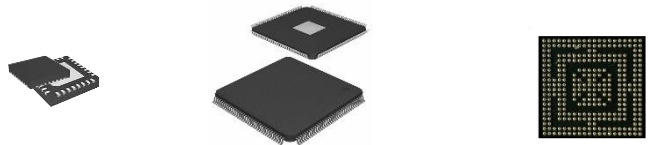
100% Can FD

SW and HW modes,
eMMC, USB, Hyper-bus

ADC monitoring
capability in Stand By

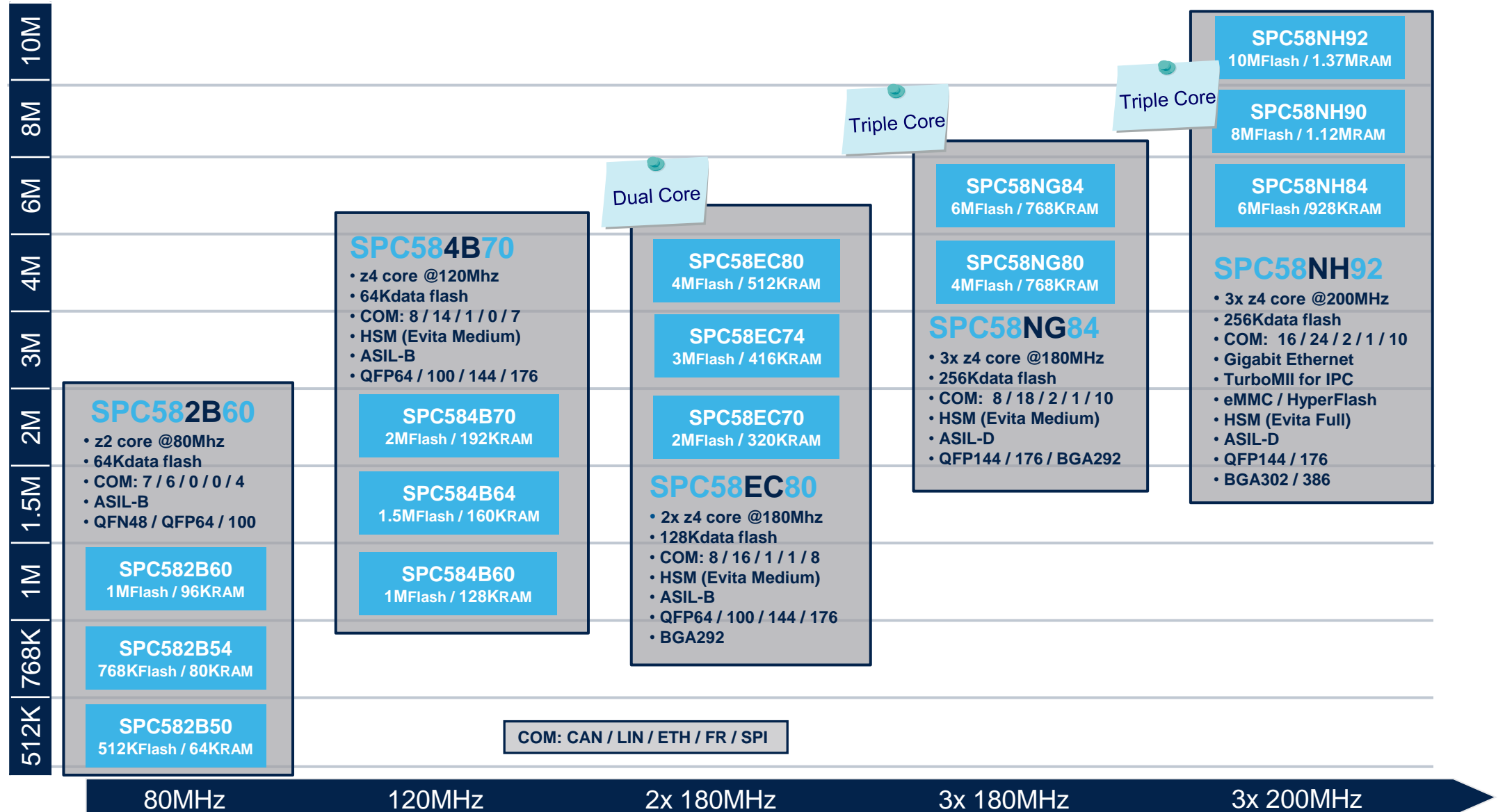
Best Subsystem Separation
Fast data decryption capability
Evita Medium and Full

Unprecedented
Scalability



QFN 7x7 eTQFP 64-176 BGA292/302/386

SPC58 Chorus Family



Chorus 2M SPC584B

Core

- 120 MHz Power Architecture™ ISA e200z4 Core (VLE)
 - Dual Issue Core with Floating Point Unit
 - 8k-Instruction Cache, 4k-Data Cache
 - 64k Local d-RAM

I/O

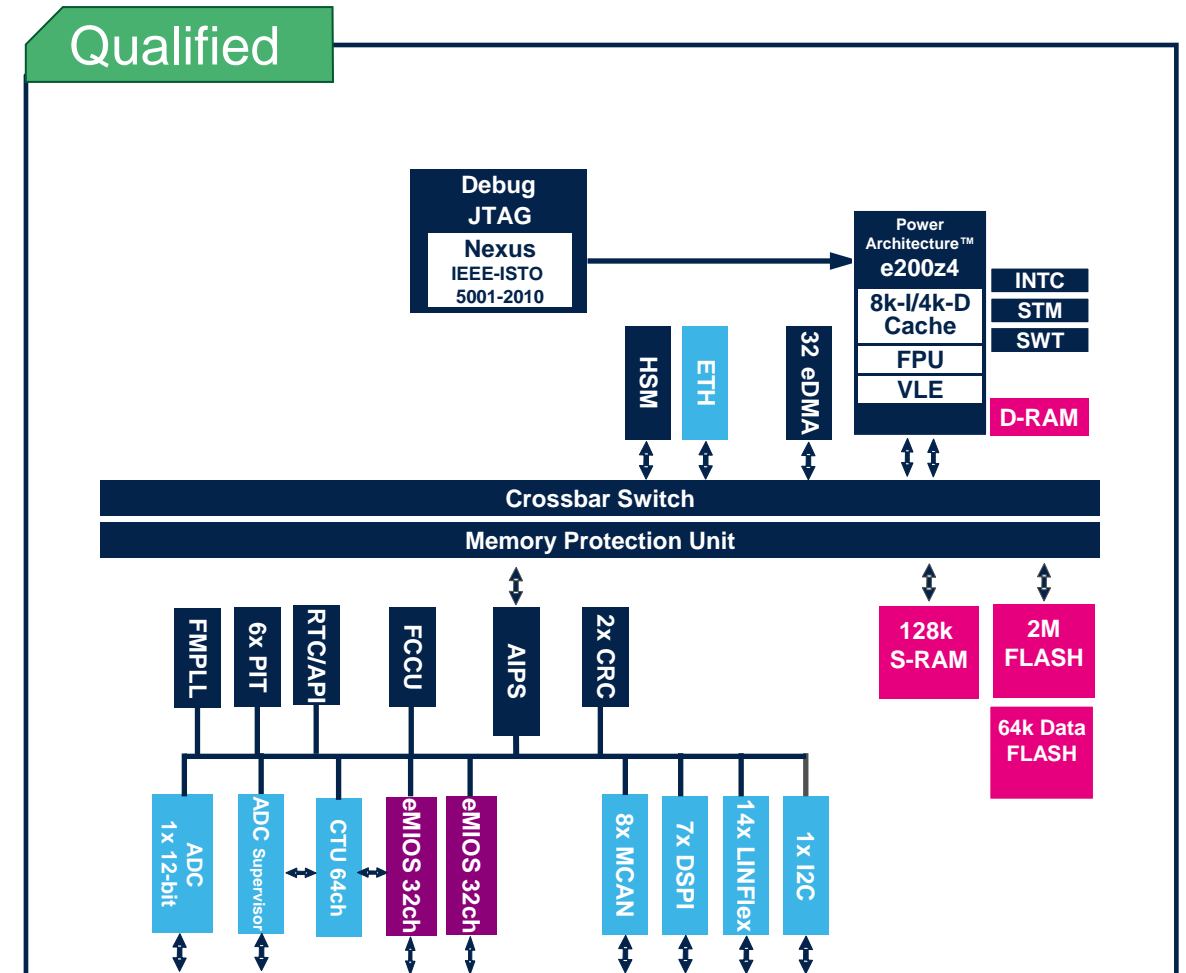
- 8 x MCAN / FD-CAN
- 14 x LINFlex
- 1 x Ethernet (100Mb/s, time stamping, AVB, IPv4 Checksum)
- 7 x DSPI
- 1 x I2C
- 2x 32ch eMIOS
- 64ch CTU (Cross Triggering Unit)
- 64 channel ADC
 - 2x 12-bit ADC
 - 1x 12-bit ADC Supervisor
 - 1x 10-bit Standby ADC

Memory

- 2M byte RWW Flash with ECC
- 4x16k Data Flash with ECC
- 192k RAM (128k SRAM, 64k Local d-RAM) with ECC

System

- SSWU (Smart Standby Wake-up)
- Security Module: HSM (Evita Medium)
- FM-PLL
- MPU
- 32 Channel eDMA Controller
- 2 x CRC Unit
- Fault Collection & Control Unit
- 6 x PIT / 1x STM / 1x RTC/API
- Nexus IEEE-ISTO 5001-2010 Class 3+



Freq. Max	Voltage	Temp.	ASIL	I/O	Package
120MHz	5V / 3.3V	-40 / +125°C	B	150	64/100/144/176



Chorus 4M SPC58EC

Core

- **180MHz Power Architecture™ ISA e200z4 Core (VLE)**
 - Dual Issue Core with Floating Point Unit
 - 8k-Instruction Cache, 4k-Data Cache
 - 64k Local d-RAM
- **180MHz Power Architecture™ ISA e200z4 Core (VLE)**
 - Dual Issue Core with Floating Point Unit
 - 8k-Instruction Cache, 4k-Data Cache
 - 64k Local d-RAM

Memory

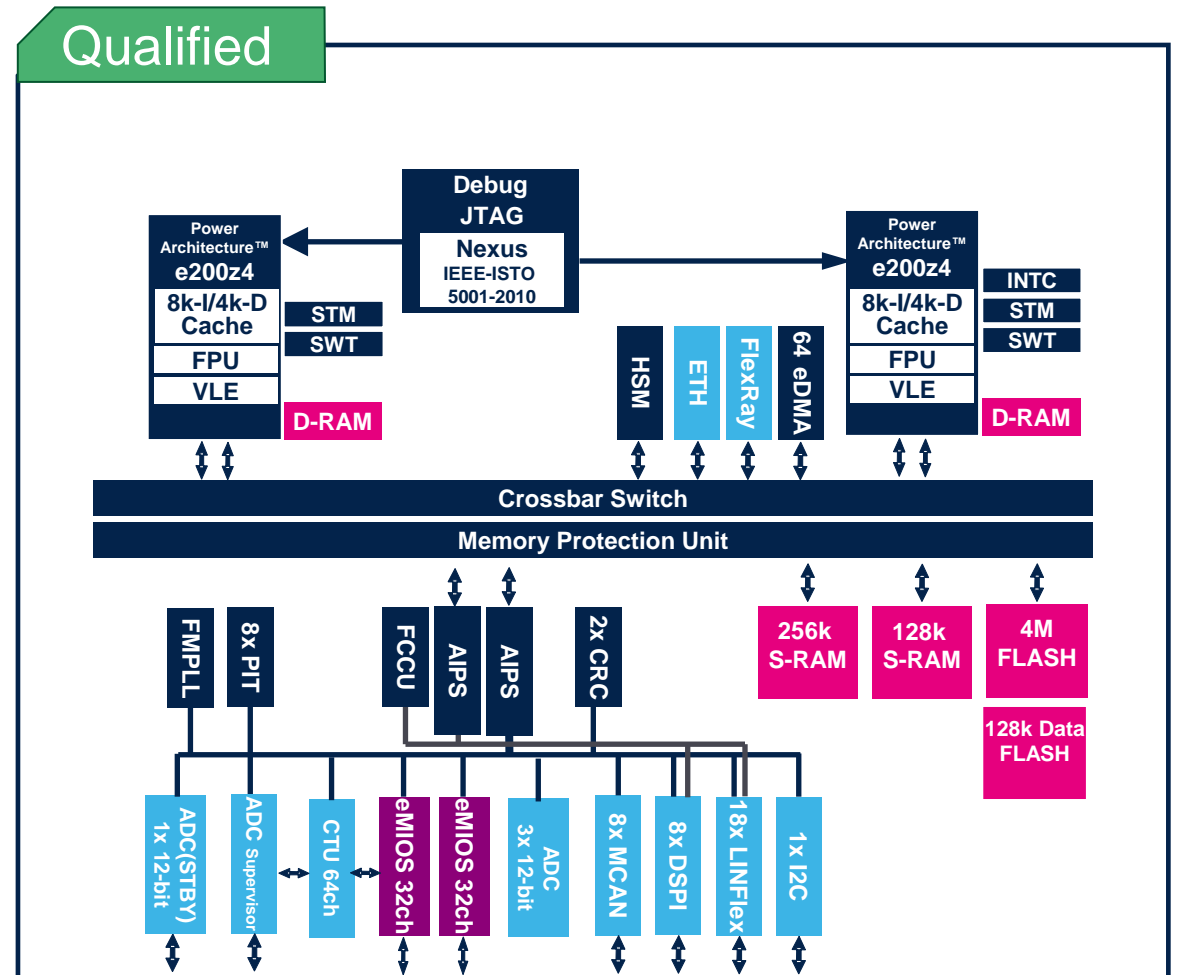
- **4M byte RWW Flash with ECC**
- **4x32k Data Flash with ECC**
- **512k RAM (384k SRAM, 2x 64k Local D-RAM) with ECC**

I/O

- **8 x MCAN / FD-CAN**
- **18 x LINFlex**
- **1 x Ethernet (100Mb/s, time stamping, AVB, IPv4 Checksum)**
- **Dual Channel FlexRay (10MB/s, 128 buffers)**
- **8 x DSPI, 1 x I2C 2x 32ch eMIOS**
- **64ch CTU (Cross Triggering Unit)**
- **95 channel ADC**
 - 3x 12-bit ADC
 - 1x 12-bit ADC Supervisor
 - 1x 10-bit Standby ADC

System

- **SSWU (Smart Standby Wake-up)**
- **Security Module: HSM (Evita Medium)**
- **FM-PLL**
- **MPU**
- **64 Channel eDMA Controller**
- **2 x CRC Unit**
- **Fault Collection & Control Unit (incl. error pin)**
- **8x PIT / 1x STM / 1x RTC/API**
- **Nexus IEEE-ISTO 5001-2010 Class 3+**



Freq. Max	Voltage	Temp.	ASIL	I/O	Package
180MHz	5V / 3.3V	-40 / +125°C	B	215	64/100/144/176/292



Chorus 6M SPC58NG

Core

- **180MHz Power Architecture™ ISA e200z4 Core (VLE)**
 - Floating Point Unit
 - 8k-Instruction Cache, 4k-Data Cache
 - 16k Local i-RAM, 64k Local d-RAM
 - Lock Step (optional)
- **180MHz Power Architecture™ ISA e200z4 Core (VLE)**
 - Floating Point Unit
 - 8k-Instruction Cache, 4k-Data Cache
 - 16k Local i-RAM, 64k Local d-RAM
- **180MHz Power Architecture™ ISA e200z4 Core (VLE)**
 - Floating Point Unit & LSP(DSP)
 - 8k-Instruction Cache
 - 16k Local i-RAM, 32k Local d-RAM

Memory

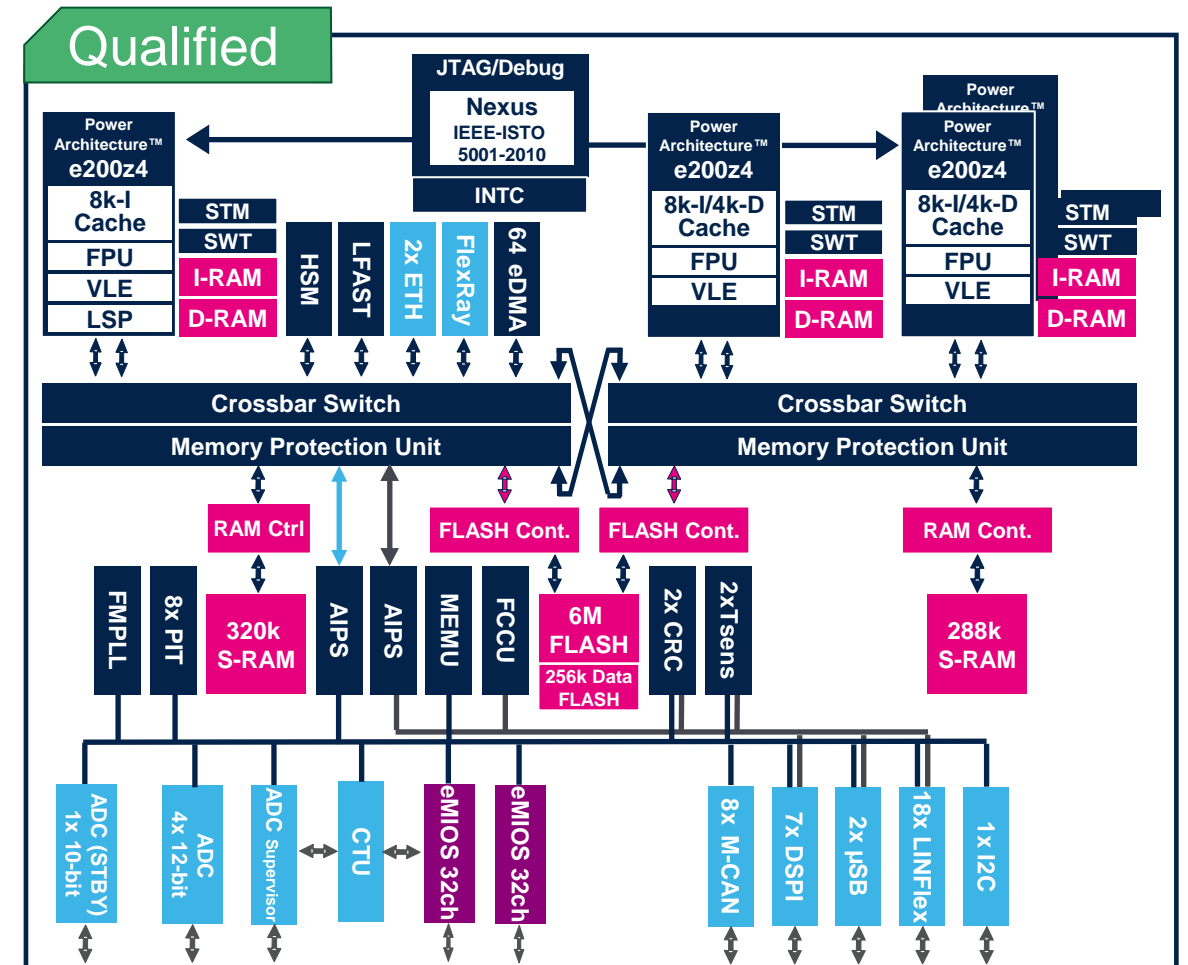
- **6M byte RWW Flash with ECC**
 - 4x64k Data Flash with ECC
- **768kRAM** (608k SRAM + 160k d-RAM) **with ECC**
 - Including 256k Standby

I/O

- **8 x MCAN / FD-CAN**
- **18 x LINFlex**
- **2 x Ethernet** (100Mb/s, time stamping, AVB, IPv4 Checksum)
- **Dual Channel FlexRay** (10MB/s, 128 buffers)
- **10 x DSPI, 1 x I2C**
- **2x 32ch eMIOS**
- **64ch CTU** (Cross Triggering Unit)
- **86 channel ADC**
 - 4x 12-bit ADC
 - 1x 12-bit ADC Supervisor
 - 1x 10-bit Standby ADC

System

- **SSWU** (Smart Standby Wake-up)
- **Security Module: HSM** (Evita Medium)
- **FM-PLL**
- **MPU**
- **64 Channel eDMA Controller**
- **2 x CRC Unit**
- **Fault Collection & Control Unit** (incl. error pin)
- **8 x PIT / 1x STM / 1x RTC/API**
- **1x LFAST** (Interprocessor bus)
- **Nexus IEEE-ISTO 5001-2010 Class 3+** (Aurora interface)



Freq. Max	Voltage	Temp.	ASIL	I/O	Package
180MHz	5V / 3.3V	-40 / +125°C	D	205	144/176/292

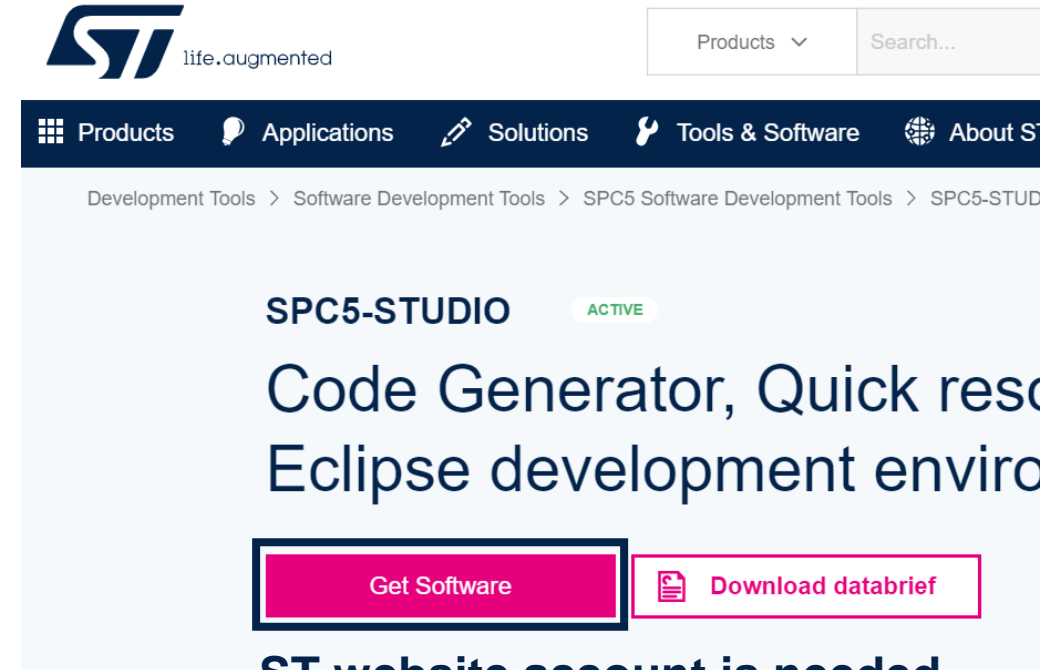
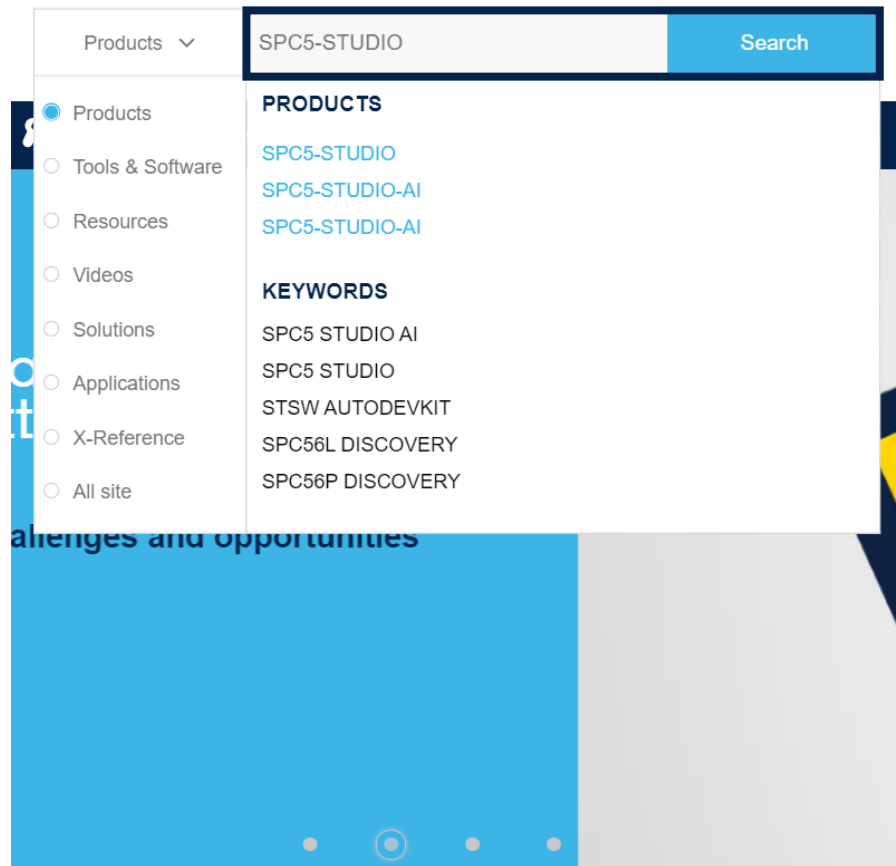
SPC5-STUDIO



- Features
 - OpenSource Integrated Development Environment (IDE) based on ECLIPSE
 - Code, Makefiles, ELF, and Doc Generation based on a template structure
 - Full MISRA 2012 compliant Register Level Access (RLA) low level drivers
 - Easy to use MCU's Pins & Run modes & Clock tree configuration
 - Software Examples for Discovery kits and premium evaluation boards
 - Free GCC / GHS / HighTec compiler support
- SPC5-STUDIO LLD is targeting evaluation purpose only.
- Further integration work should be needed for mass production.

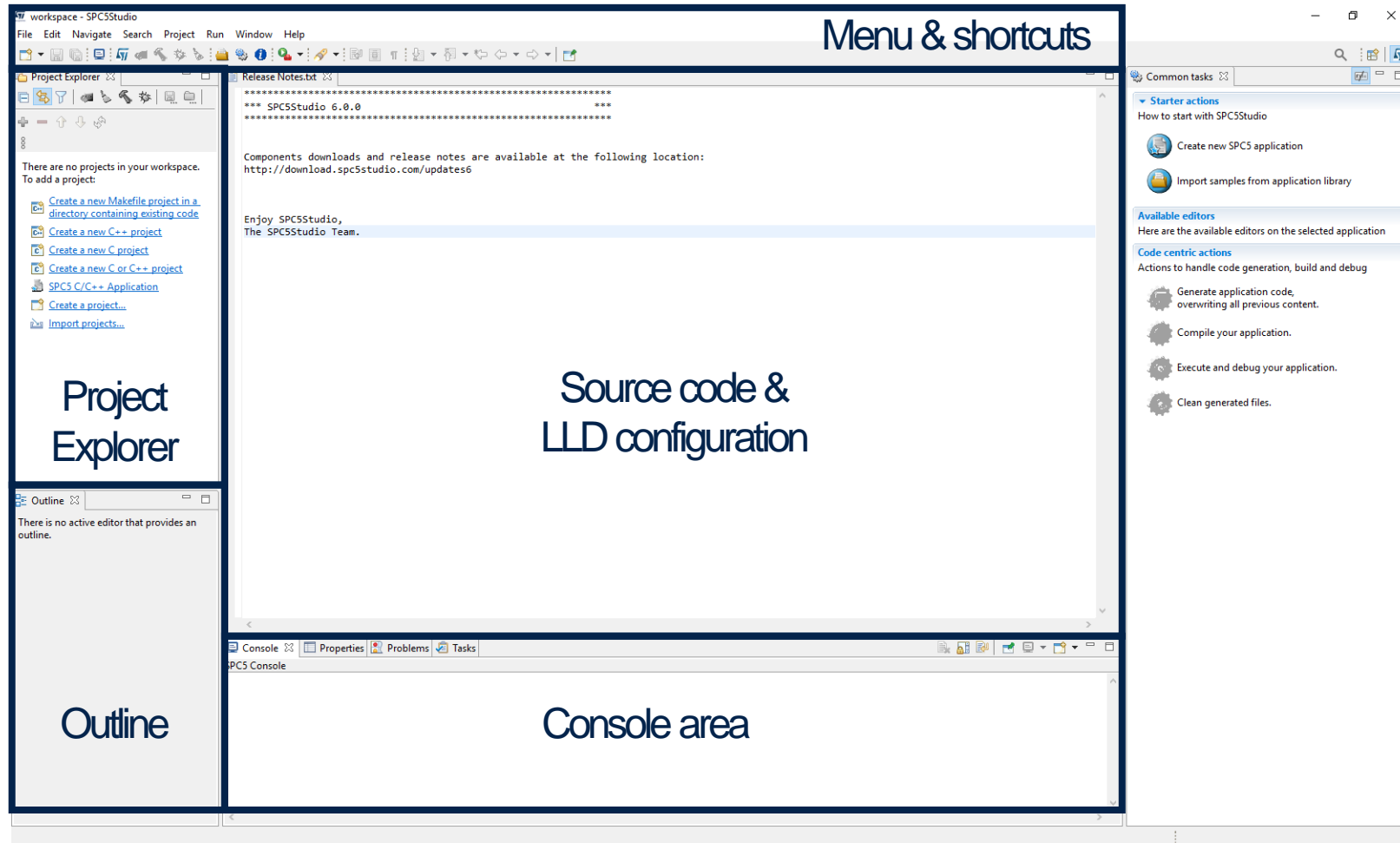
How to Get SPC5-STUDIO

- <https://st.com>



ST website account is needed

SPC5-STUDIO



SPC5-STUDIO

- Studio & plugin version

The screenshot displays three overlapping windows from the SPC5 Studio application:

- Help Menu:** A dropdown menu with options: Welcome, Help Contents, Search, Show Contextual Help, Show Active Keybindings... (Ctrl+Shift+L), Tips and Tricks..., Cheat Sheets..., Eclipse User Storage, Check for Updates, Install New Software..., and Eclipse Marketplace... The 'About SPC5Studio' option is highlighted.
- About SPC5Studio Dialog:** A window titled 'About SPC5Studio' showing the version (6.0), copyright (© 2020 STMicroelectronics), and a link to the website (<http://www.st.com/spc5studio>). It also provides a link to download components (<http://download.spc5studio.com/updates6/>). The 'Installation Details' button is highlighted.
- SPC5Studio Installation Details Dialog:** A window showing a list of installed software components. The 'Installation Details' tab is active. The list includes various features and their versions.

Name	Version	Id
> Firmware Updater components	1.4.2.20200626121133	com.st.spc5.componen
> Gnu GCC feature	4.9.4.20200626121336	com.st.tools.spc5.tools.
> GNU Tools Feature	1.1.0.20200626121250	com.st.tools.spc5.tools.
> Hightec Free Tools Feature	4.9.4.201809141314	com.st.tools.spc5.tools.
> PinMap Editor for SPC5x	4.4.1.20200626120943	com.st.tools.spc5.wizar
> SPC5 Flash driver components	1.10.1.202002250928	com.st.spc5.componen
> SPC5-Benchmarks_Support_Feature	1.4.0.20200626121141	com.st.spc5.componen
> SPC5-RLA_AI_Support_Feature	1.0.0.20200626121232	com.st.spc5.componen
> SPC5-RLA_FreeRTOS_Support_Feature	1.6.4.20200626121050	com.st.spc5.componen
> SPC5-RLA_FreeRTOS_TCPIP_Support_Featu	1.9.0.20200626121113	com.st.spc5.componen
> SPC5-RLA_GTM_Support_Feature	1.13.1.202006261211...	com.st.spc5.componen
> SPC5-RLA_Network_Support_Feature	1.8.1.20200626121110	com.st.spc5.componen
> SPC5-RLA_Runtime-IO_Support_Feature	1.3.6.20200626121056	com.st.spc5.componen
> SPC5-RLA_SPC560Bxx_Support_Feature	1.13.1.202006261210...	com.st.spc5.componen
> SPC5-RLA_SPC560Bxx_Support_Feature	1.13.1.202006261210...	com.st.spc5.componen

SPC5-STUDIO

- Short cut icons

The image shows a screenshot of the SPC5Studio application window. The title bar reads "workspace - C:\SPC5Studio-6.0\workspace\Release Notes.txt - SPC5Studio". The menu bar includes "File", "Edit", "Navigate", "Search", "Project", "Run", "Window", and "Help". The toolbar contains various icons, with a dark blue box highlighting five specific icons. Arrows point from these icons to the following text labels:

- Generate SPC5 application
- Build SPC5 application
- Launch UDE debugger
- Clean SPC5 application
- Example application wizard

Example code wizard

- User-friendly wizard



SPC5Studio Wizard

SPC5Studio Wizard

Import Application(s) from SPC5Studio Application template library.

SPC5Studio Import application Wizard

Step 1:

Select a family:
SPC58


Select a product line:
CHORUS4M-Line

Select a device:
SPC58EC80E5

Select an evaluation board:

SPC58EC-DISP AEK-MCU-C4MULT1



Next > Finish Cancel

SPC5Studio Wizard

SPC5Studio Wizard

Filter through the available SPC5 application templates in SPC5Studio library.

SPC5Studio Import application Wizard

Step 2:

Template library for selected lines / evaluation boards.

Select your search parameters:

Board	Drivers	RTOS
<input type="radio"/> SPC58ECXX_DIS	<input type="radio"/> PAL	<input type="radio"/> OSLess
	<input type="radio"/> PIT	<input type="radio"/> FreeRTOS
	<input type="radio"/> Serial	
	<input type="radio"/> CAN	

Choose your sample application:

Application Name	Description	Device
<input checked="" type="checkbox"/> SPC58ECxx_RLA PIT Test Ap...	Test application for the SPC58ECxx create...	SPC58E...
<input type="checkbox"/> SPC58ECxx_RLA PIT Lifetime...	Test application for the SPC58ECxx create...	SPC58E...
<input type="checkbox"/> SPC58ECxx_RLA Network Pin...	Test application for the SPC58ECxx create...	SPC58E...
<input type="checkbox"/> SPC58ECxx_RLA Network IPE...	Test application for the SPC58ECxx create...	SPC58E...
<input type="checkbox"/> SPC58ECxx_RLA Network Ga...	Test application for the SPC58ECxx create...	SPC58E...

Finish Cancel

Platform component

- Build Settings

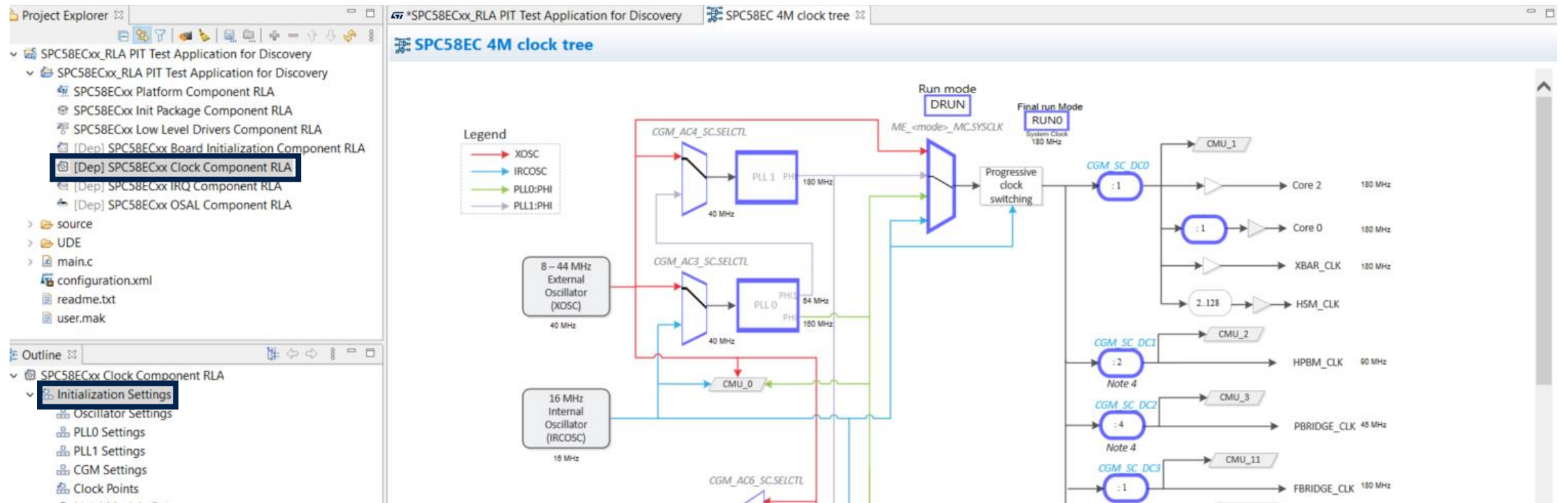
The screenshot displays the IDE interface for configuring the SPC58ECxx Platform Component RLA. The Project Explorer on the left shows the project structure, with 'SPC58ECxx Platform Component RLA' selected. The Outline view shows the 'Build Settings' section expanded. The main window shows the 'Application Configuration' for 'SPC58ECxx Platform Component RLA', with the 'Build Settings' section expanded to show compiler options and switches.

Build Settings
Compiler related options and switches.

Application Name	out	Compiler Variant	FreeGCC with VLE support
Optimization Level	-O2 speed	Use FPU	<input type="checkbox"/>
Common Options	-gdwarf-2 -fomit-frame-pointer -falign-functions=16 -fno-gcse -std=gnu99	C Options	
CPP Options	-fno-rtti -fno-exceptions	Use Verbose Compile	no
Create Static Library	<input type="checkbox"/>		

Clock component

- Clock Settings



LLD component

- LLD Settings

The screenshot displays the STM32CubeIDE interface for configuring the SPC58ECxx Low Level Drivers Component RLA. The Project Explorer on the left shows the component selected under the PIT Settings. The Application Configuration window on the right shows the PIT0 settings for eight channels. Channel 1 is the only channel that is enabled, with a frequency of 1 and a callback function of cb_pit0_ch1. All other channels (0, 2, 3, 4, 5, 6, 7) are disabled with a frequency of 0.

Channel	Enabled	Frequency	CallBack
Channel 0	<input type="checkbox"/>	0	
Channel 1	<input checked="" type="checkbox"/>	1	cb_pit0_ch1
Channel 2	<input type="checkbox"/>	0	
Channel 3	<input type="checkbox"/>	0	
Channel 4	<input type="checkbox"/>	0	
Channel 5	<input type="checkbox"/>	0	
Channel 6	<input type="checkbox"/>	0	
Channel 7	<input type="checkbox"/>	0	

Generate

- Generate
 - After generate, LLD driver code will be present

The screenshot displays two windows from an IDE. The left window, titled 'SPC5 Console', shows the output of a generation process. The output is as follows:

```
-----  
Application Generation Summary  
-----  
SPC58ECxx Init Package Component RLA ...DONE!  
SPC58ECxx IRQ Component RLA ...DONE!  
SPC58ECxx Board Initialization Component RLA ...DONE!  
SPC58ECxx Clock Component RLA ...DONE!  
SPC58ECxx Platform Component RLA ...DONE!  
SPC58ECxx OSAL Component RLA ...DONE!  
SPC58ECxx Low Level Drivers Component RLA ...DONE!
```

The right window, titled 'Project Explorer', shows the project structure for 'SPC58ECxx_RLA PIT Test Application for Discovery'. The structure is as follows:

- SPC58ECxx_RLA PIT Test Application for Discovery
 - SPC58ECxx Platform Component RLA
 - SPC58ECxx Init Package Component RLA
 - SPC58ECxx Low Level Drivers Component RLA
 - [Dep] SPC58ECxx Board Initialization Component RLA
 - [Dep] SPC58ECxx Clock Component RLA
 - [Dep] SPC58ECxx IRQ Component RLA
 - [Dep] SPC58ECxx OSAL Component RLA
 - components
 - spc58ecxx_board_initialization_component_ria
 - spc58ecxx_clock_component_ria
 - spc58ecxx_init_package_component_ria
 - spc58ecxx_irq_component_ria
 - spc58ecxx_low_level_drivers_component_ria
 - cfg
 - doc
 - SPC58ECxx RLA LLD Component User Manual.c
 - lib
 - mak

Help document

LLD code

- Build
 - After build, bin / hex / elf / map will be present



```
CDT Build Console [SPC58ECxx_RLA PIT Test Application for Discovery]
Compiling eirq.c
Compiling pit_lld_cfg.c
Compiling osal.c
Compiling systimer.c
Compiling board.c
Compiling irq_cfg.c
Compiling eirq_cfg.c
Compiling main.c
Compiling components.c
Linking build/out.elf
Creating build/out.mot
Creating build/out.bin
Creating build/out.dmp
Creating build/out.hex

text  data  bss   dec   hex filename
17970  120   4376  22466  57c2 build/out.elf

Done

13:09:14 Build Finished. 0 errors, 0 warnings. (took 1
```

6.0 > workspace > SPC58ECxx_RLA PIT Test Application for Discovery > build

Name	Date modified	Type
lst	8/21/2020 1:08 PM	File folder
mcs	8/21/2020 1:07 PM	File folder
obj	8/21/2020 1:08 PM	File folder
out.bin	8/21/2020 1:08 PM	BIN File
out.dmp	8/21/2020 1:09 PM	DMP File
out.elf	8/21/2020 1:08 PM	ELF File
out.hex	8/21/2020 1:08 PM	HEX File
out.map	8/21/2020 1:08 PM	MAP File
out.mot	8/21/2020 1:08 PM	MOT File

Flash download / debug

- Depends on your taste

LAUTERBACH
DEVELOPMENT TOOLS



pls
Development Tools



Green Hills
SOFTWARE



PE
micro



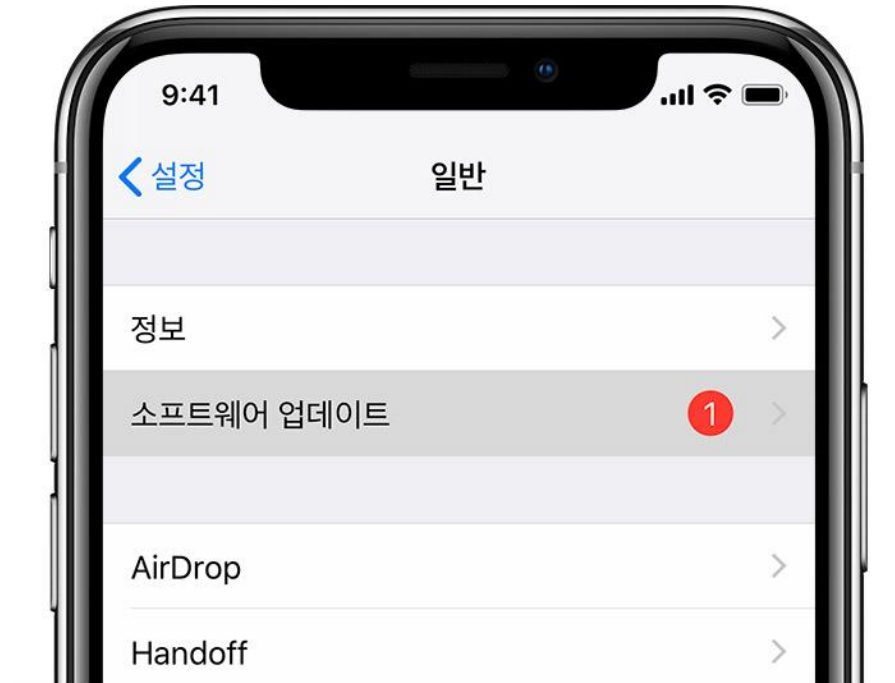
Background : FOTA



Software updates in consumer electronics

- **OS update for smartphones & set-top boxes (STB)**

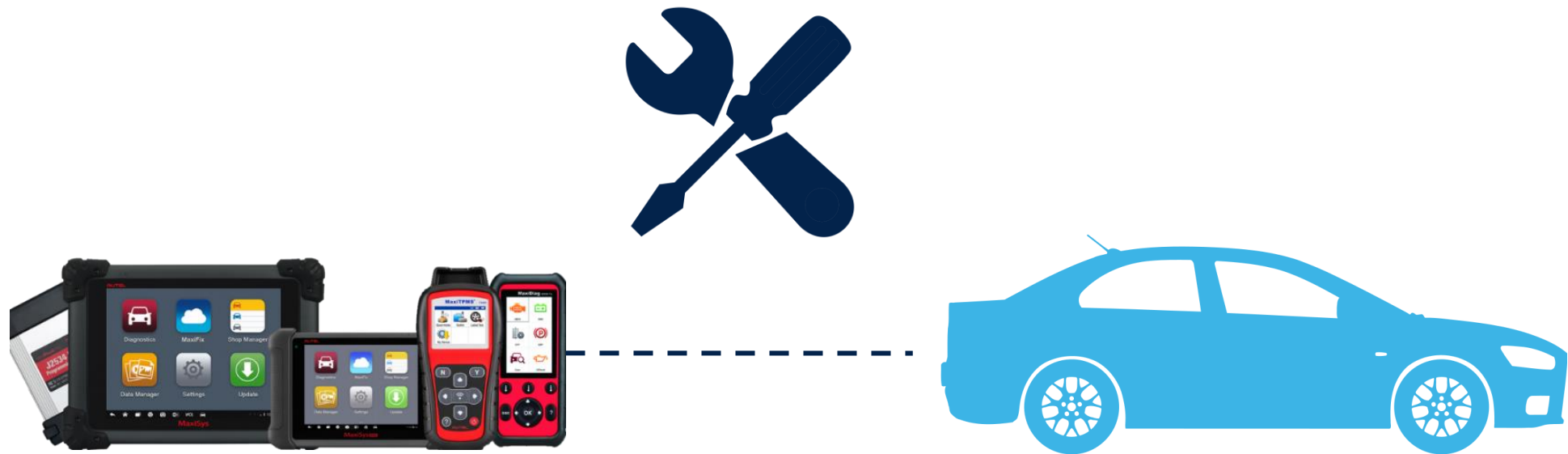
- Remote Software Updates exist for more than 10 years in consumer electronics.
- Background downloading for new firmware via Wi-Fi / Ethernet



Software updates for cars (classic)

- **Manual Update in car repair shop**

- Every customer needs to visit a car repair shop to update software via an on-board diagnostics (OBD) tool.



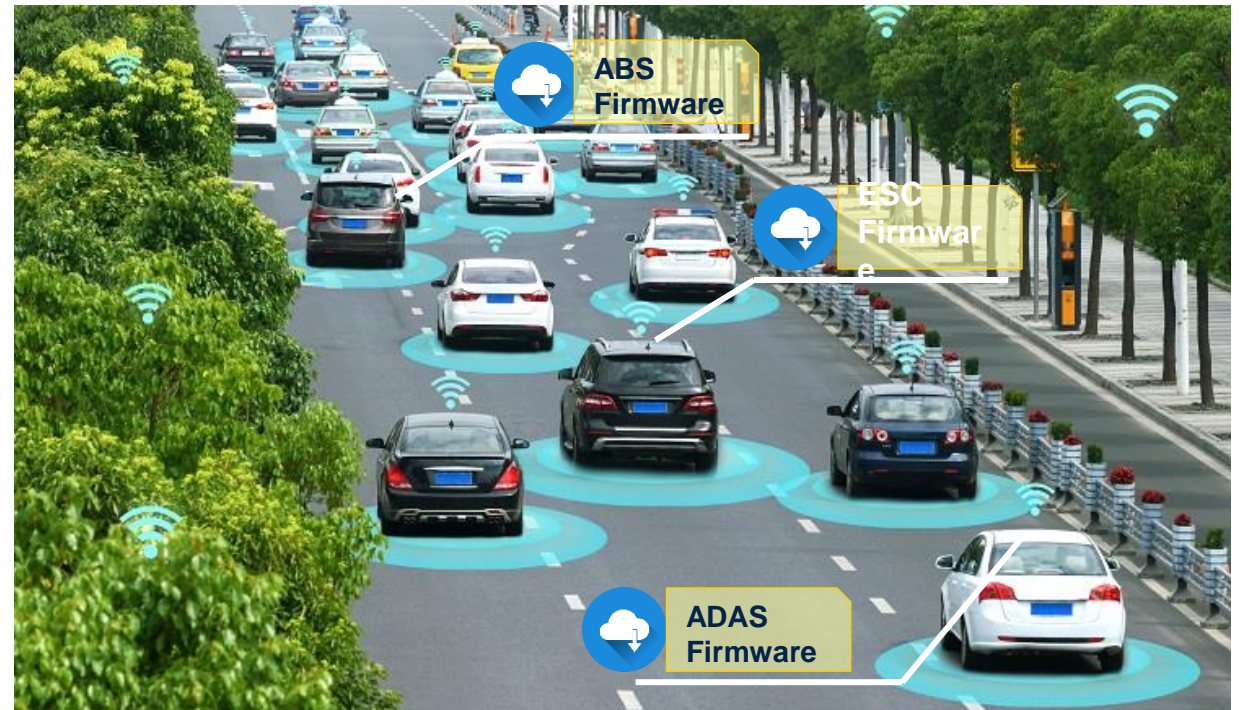
Why FOTA concepts and basic requirements

- **Why Firmware Over The Air (FOTA) updates?**
 - On an average, an automotive vehicle today has 100 ECUs and over 100 million lines of SW code. By 2020, there are expected to be 300 ECUs in a car.
 - In such a complex automotive electronics and software set-up, the need to remotely manage and update the vehicle ECU software becomes all the more important.
 - A security attack on any Electronic Control Unit (ECU) of a vehicle may prove to be dangerous for the passengers and cause a bad reputation for the automotive OEM.
 - To avoid such a crisis, Firmware Over The Air (FOTA) updates have been identified as a robust, reliable and cost-effective method for remotely managing the software updates of connected car systems.

Benefits of FOTA

Concepts and basic requirements

- Firmware-over-the-air (FOTA) services enable firmware downloads and updates for any of the specific electronic control units (ECUs) inside a car.
- Cost-efficient and better managed firmware updates
- Upgrade the firmware safely, anytime & anywhere
- Lowers the time to market



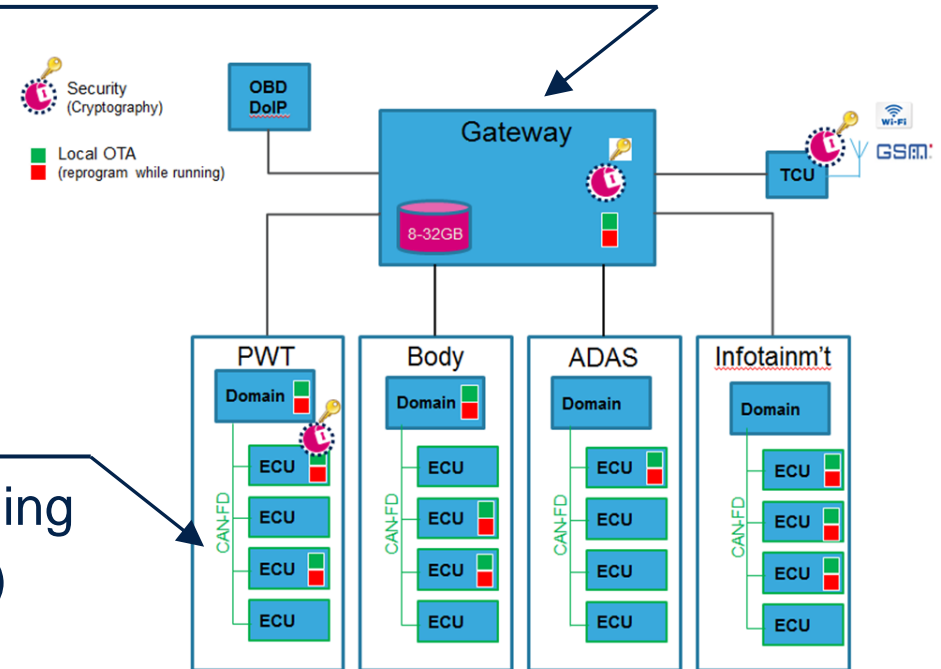
FOTA Requirements Application Scheme

- **Central OTA management application, like gateway and/or storage**

- Flash context management by HW
- Interface for external memory
- Ultra-fast communication interface
- Advanced security features

- **Local OTA available application**

- Flash erasing/program while application is running
- Security features (authentication, cryptography)
- Fast communication interface



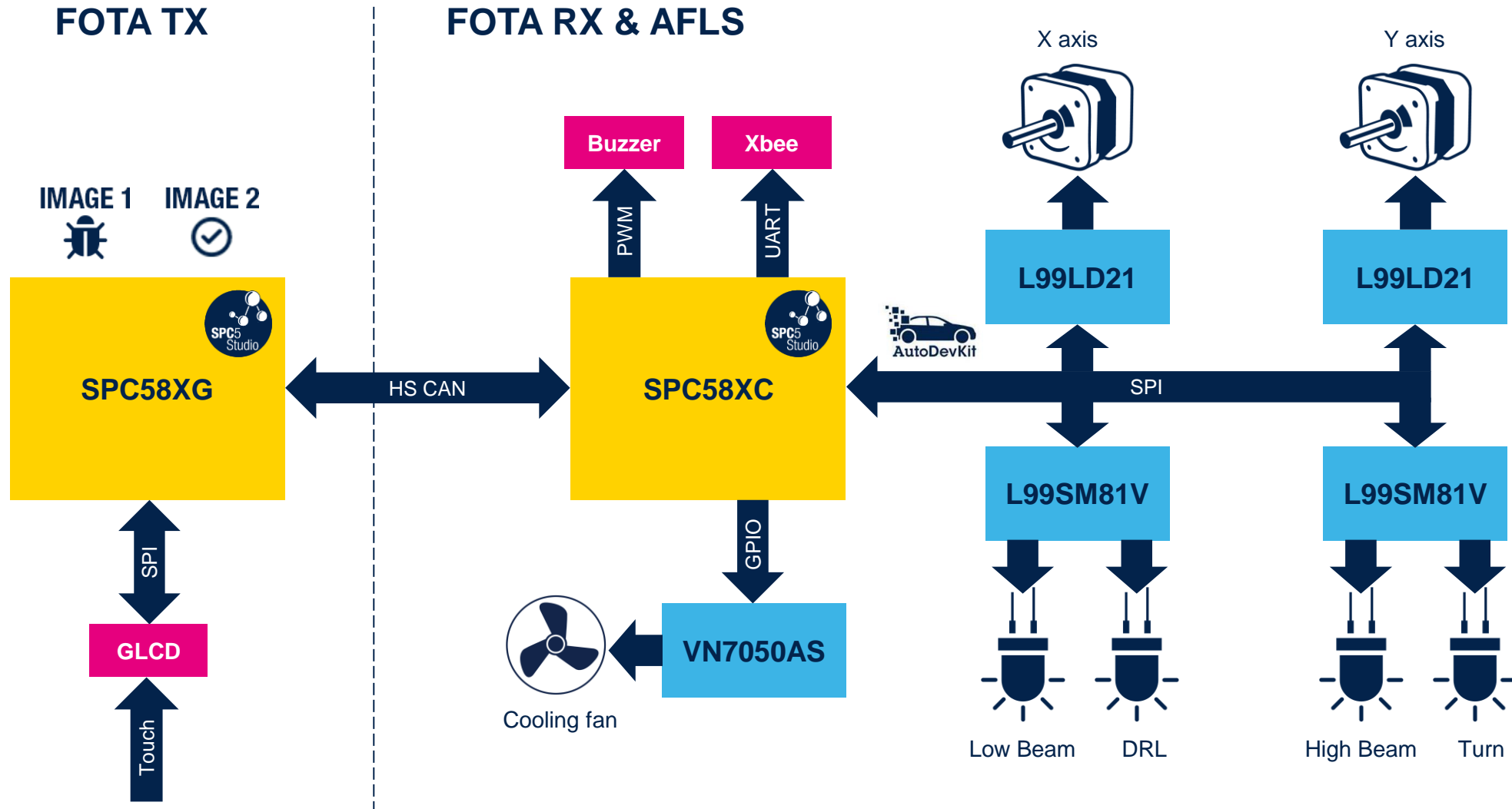
SPC5 Chorus family fully supports both kinds of applications

AFLS + FOTA Demo



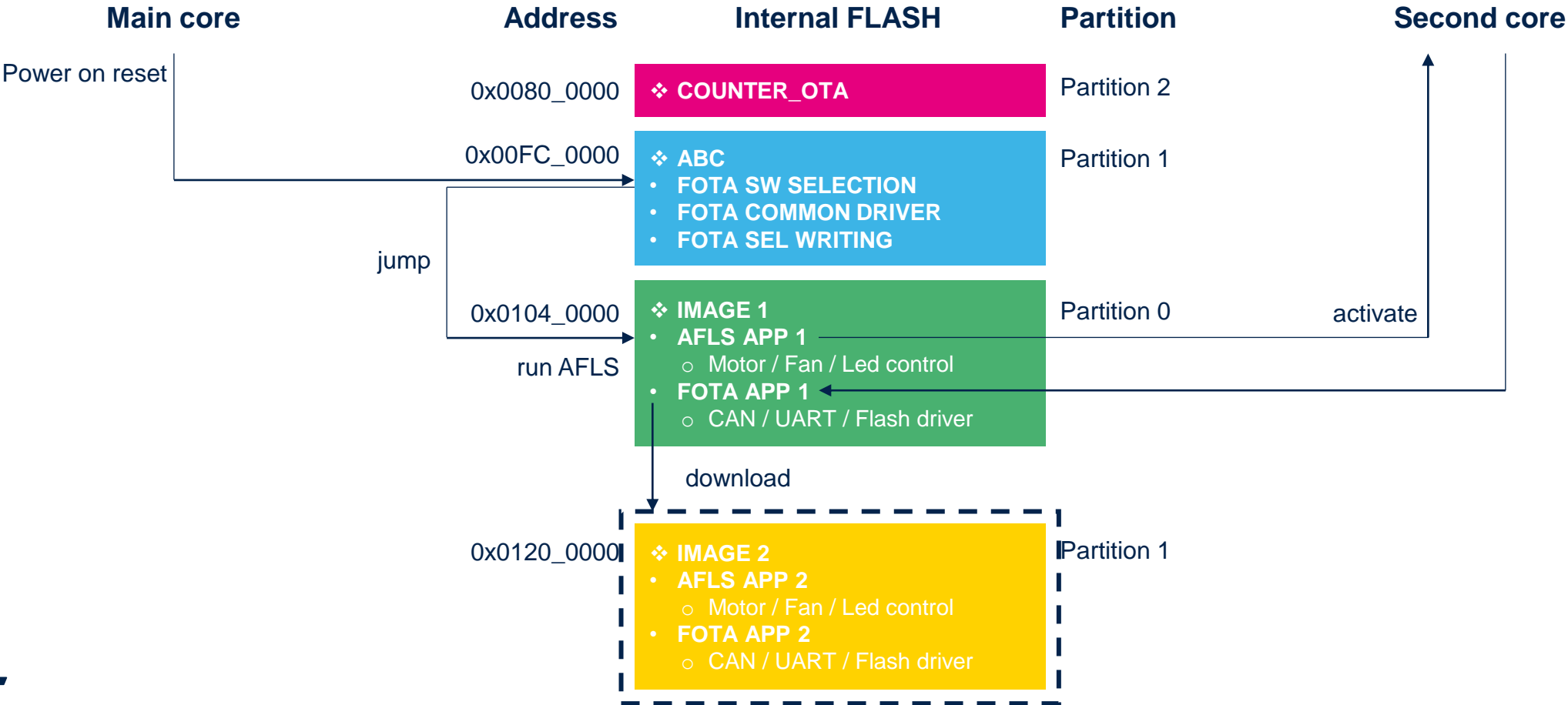
- This demo shows FOTA¹⁾ functionality during AFLS²⁾ application running.
- User can see the demo progress through LED front lamp, stepper motor, cooling fan, GLCD, serial log and buzzer.
- Key ST product
 - L99LD21 : LED driver
 - VN7050AS : IPS
 - L99SM81 : Motor driver
 - SPC58XC : Chorus 4M MCU
 - SPC58XG : Chorus 6M MCU

Block diagram (HW)



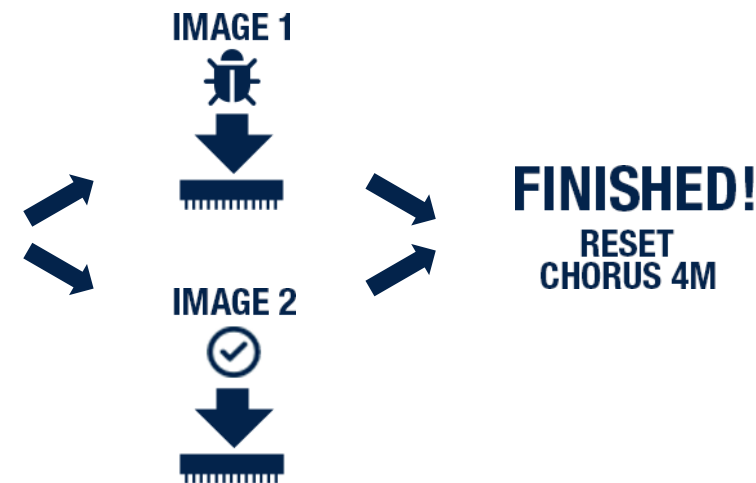
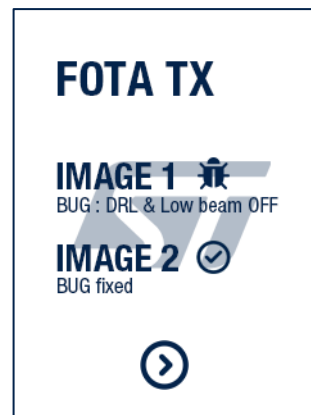
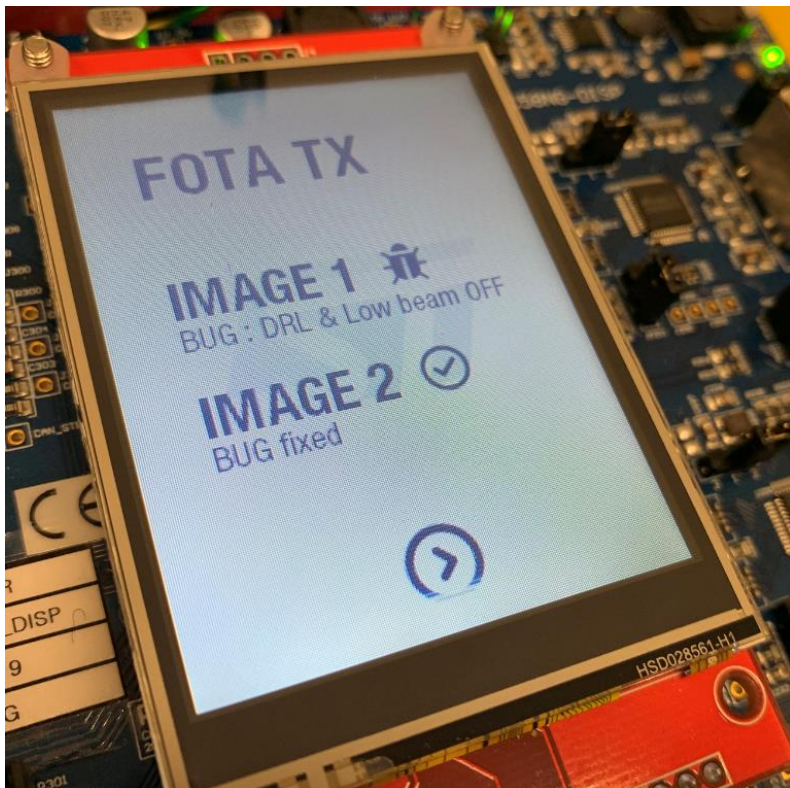
Block diagram (SW)

- How it works?



Appearance

- GLCD



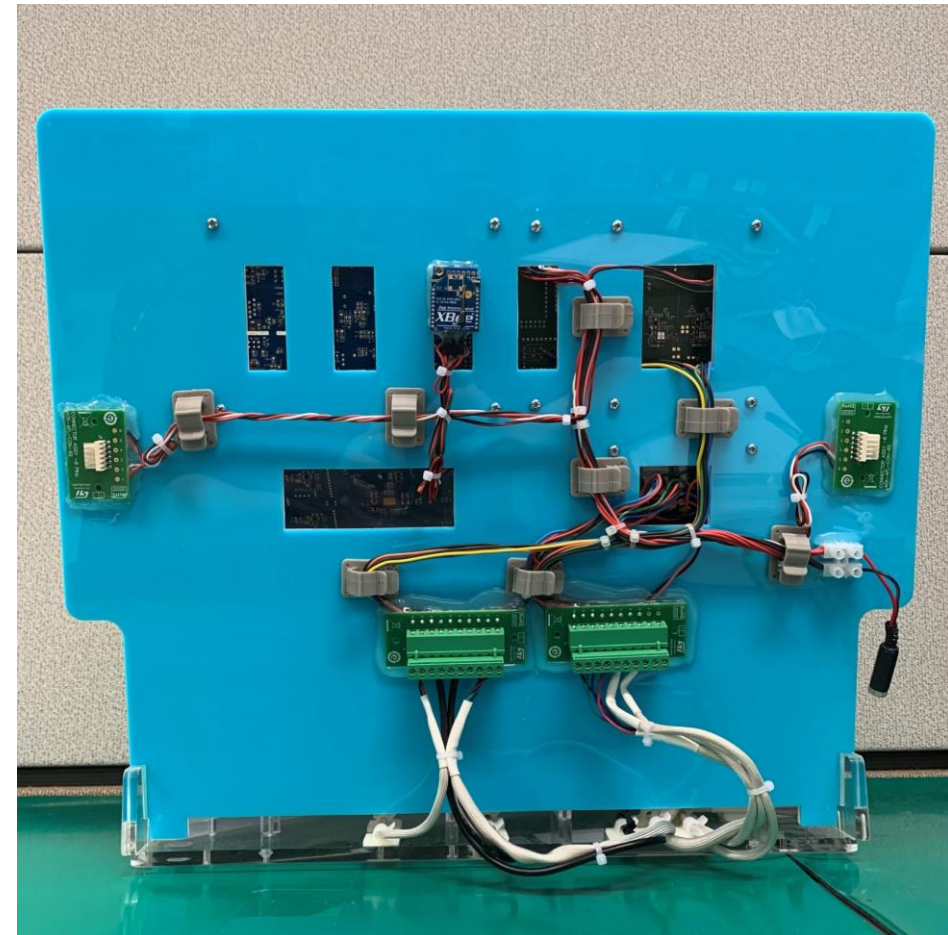
- Serial Log

Console log	Console log
<pre>===== STMicroelectronics ADG KOREA ===== AFLS + FOTA Adaptive Front Lighting System Firmware Over The Air ===== # DEMO information FOTA TX : Chorus6M (SPC58XG) FOTA RX : Chorus4M (SPC58XC) COMM. : HS CAN (250Kbps) ===== # Log from FOTA RX AFLS : AFLS core ACTIVATED... AFLS : Boot loader executing... AFLS : OTA counter reading... AFLS : Code jump area -> [IMAGE 1] AFLS : IMAGE 1 executing... AFLS : AFLS Initializing... AFLS : FOTA core ACTIVATED... AFLS : IMAGE 1 running ! FOTA RX : Ready to receive IMAGE 2...</pre>	<pre># Log from FOTA RX AFLS : AFLS core ACTIVATED... AFLS : Boot loader executing... AFLS : OTA counter reading... AFLS : Code jump area -> [IMAGE 2] AFLS : IMAGE 2 executing... AFLS : AFLS Initializing... AFLS : FOTA core ACTIVATED... AFLS : IMAGE 2 running ! FOTA RX : Ready to receive IMAGE 1... FOTA RX : IMAGE 1 downloading... FOTA RX : IMAGE 1 download complete! FOTA RX : Next Code jump area -> [IMAGE 1] ===== DEMO END RESET Chorus 4M =====</pre>

Appearance

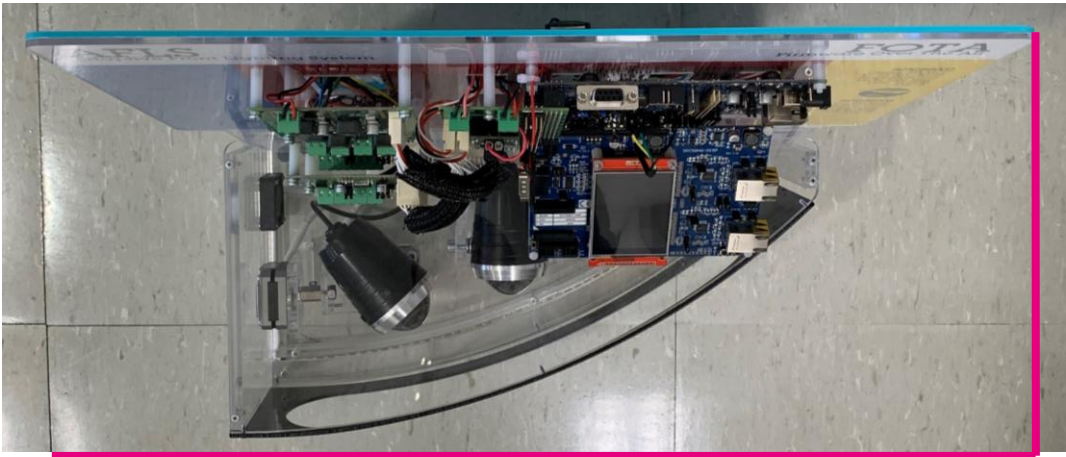


Front

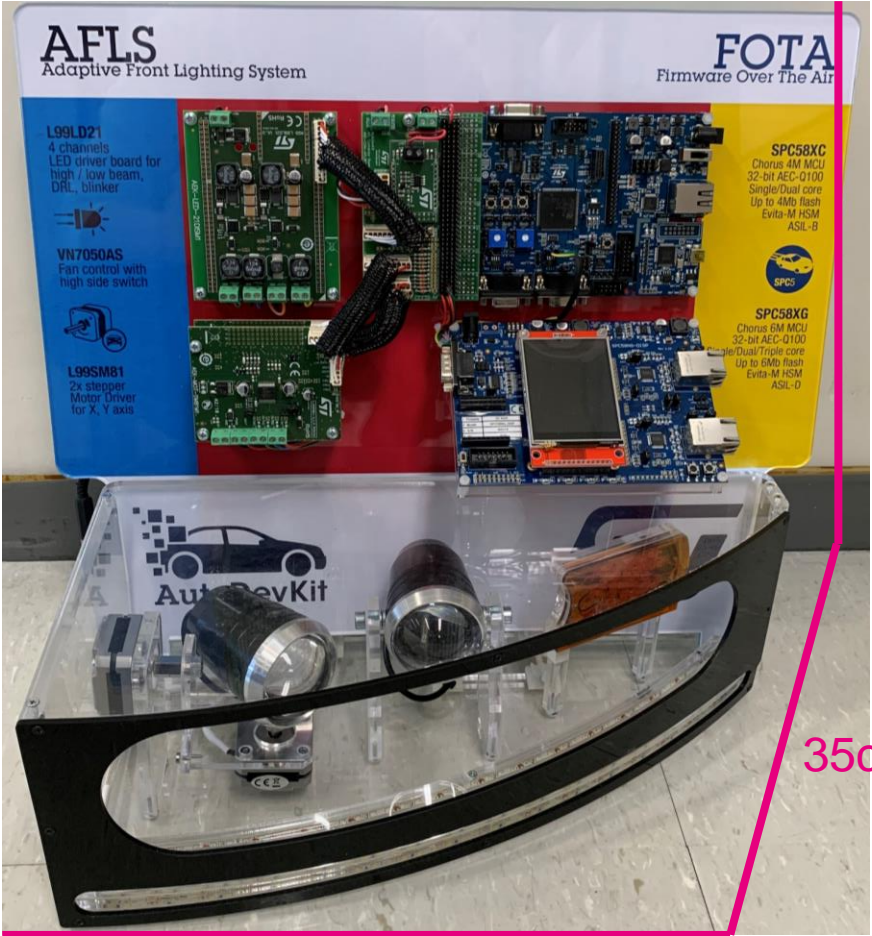


Back

Appearance



50cm



40cm

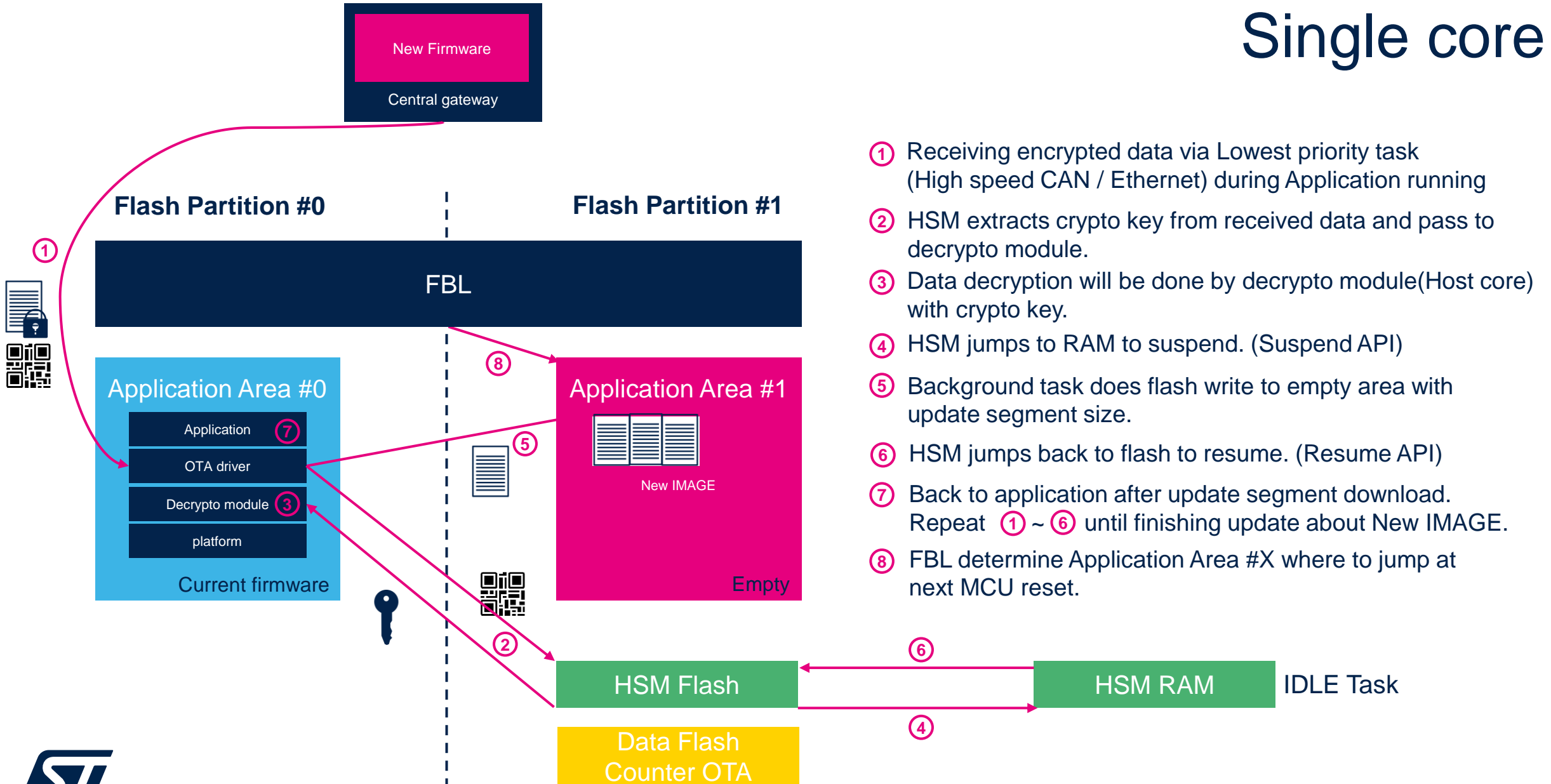
35cm

50cm

- Demo video

TO BE UPDATED

Chorus 4M OTA scenario Single core



Thank you

© STMicroelectronics - All rights reserved.

ST logo is a trademark or a registered trademark of STMicroelectronics International NV or its affiliates in the EU and/or other countries.

For additional information about ST trademarks, please refer to www.st.com/trademarks.

All other product or service names are the property of their respective owners.



life.augmented