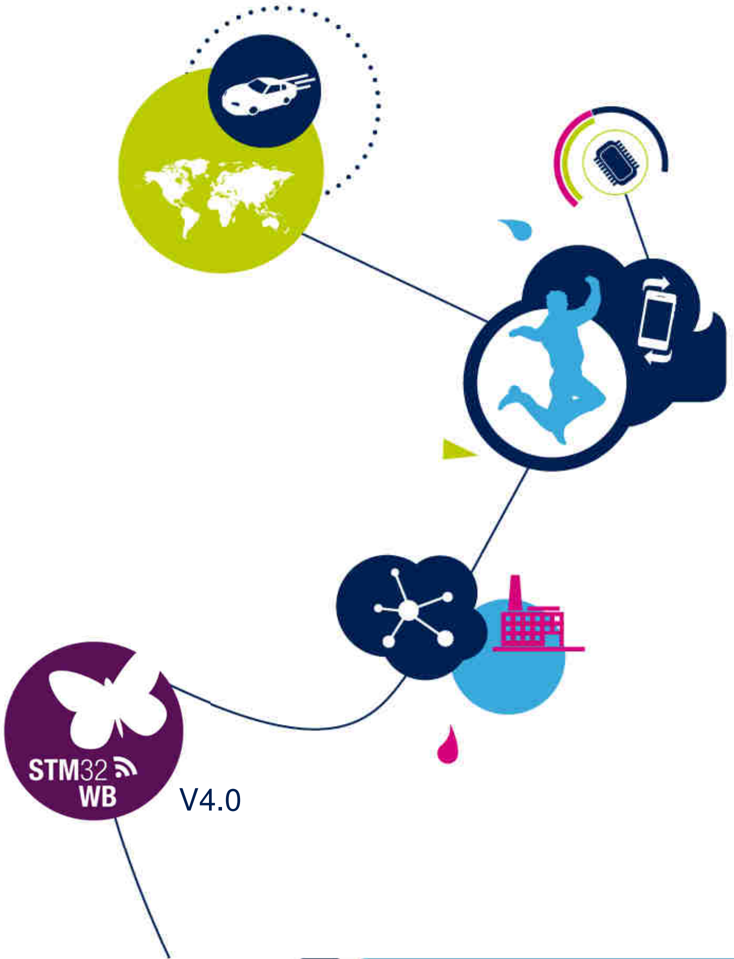
STM32WB Workshop

Alec Bath

Americas Marketing and Applications Team

Chicago, IL Technology Tour 2019



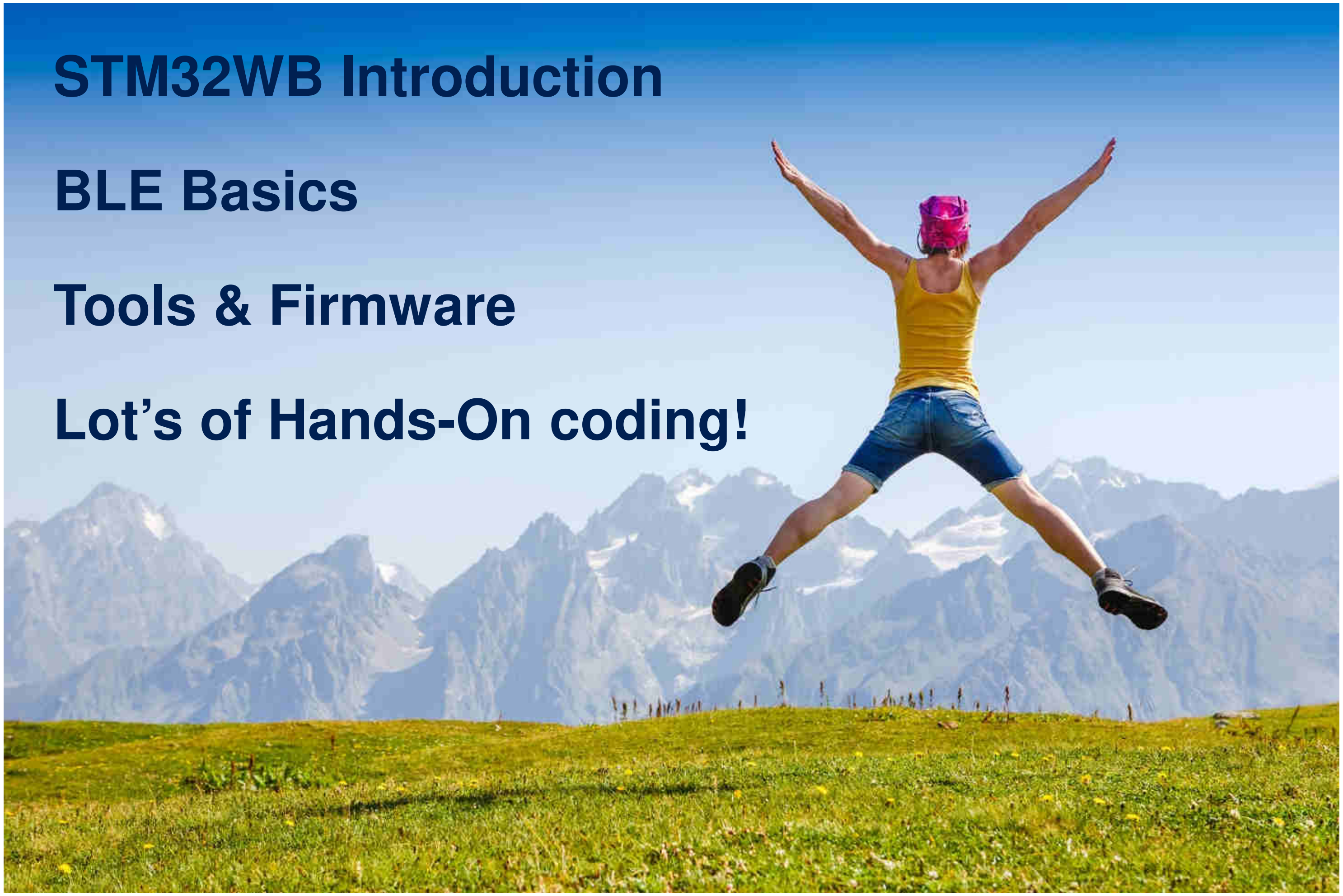
V4.0

life.augmented

Technology Tour 2019

Schaumburg, IL | April 25

Objectives



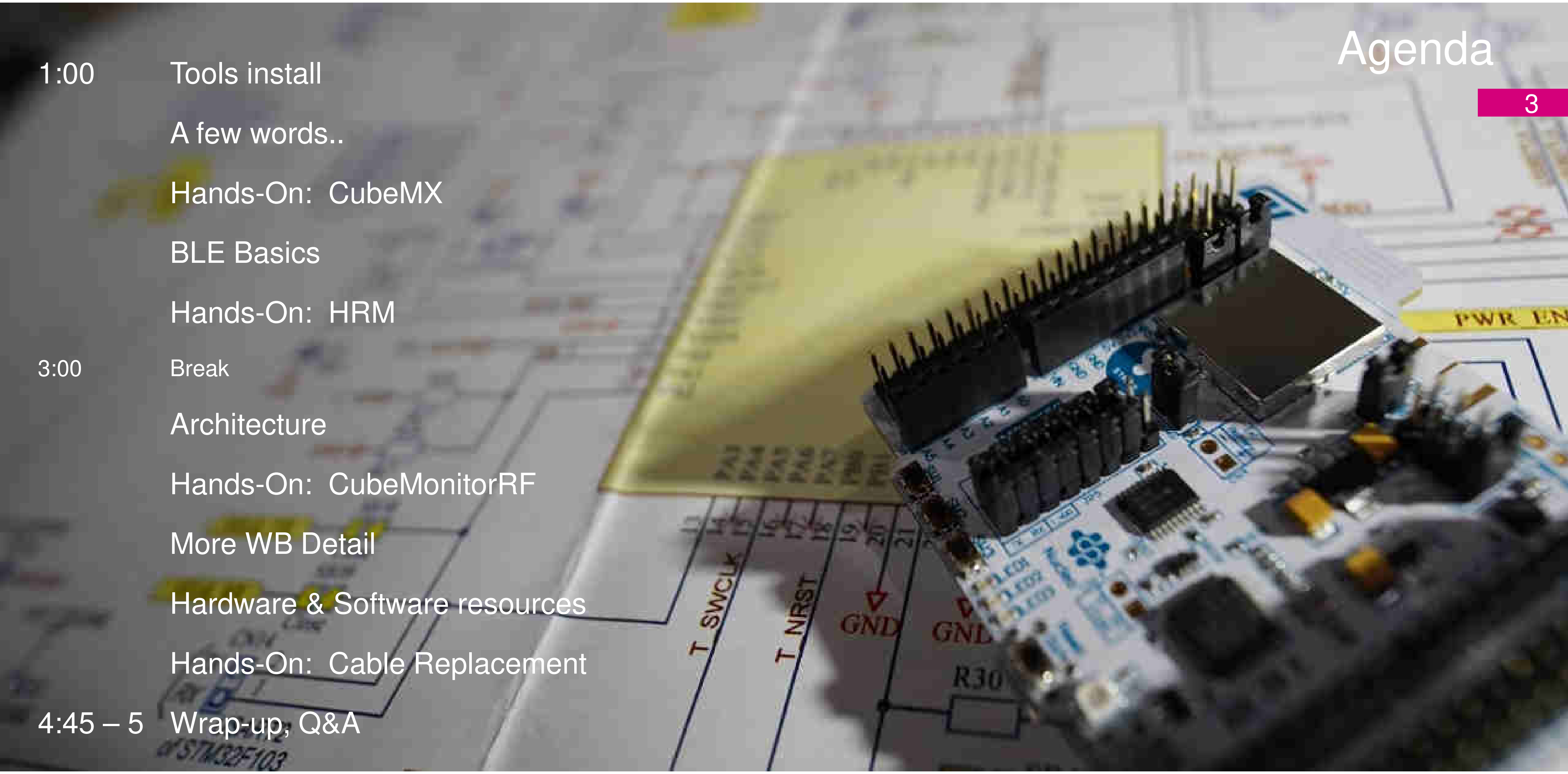
STM32WB Introduction

BLE Basics

Tools & Firmware

Lot’s of Hands-On coding!

Agenda

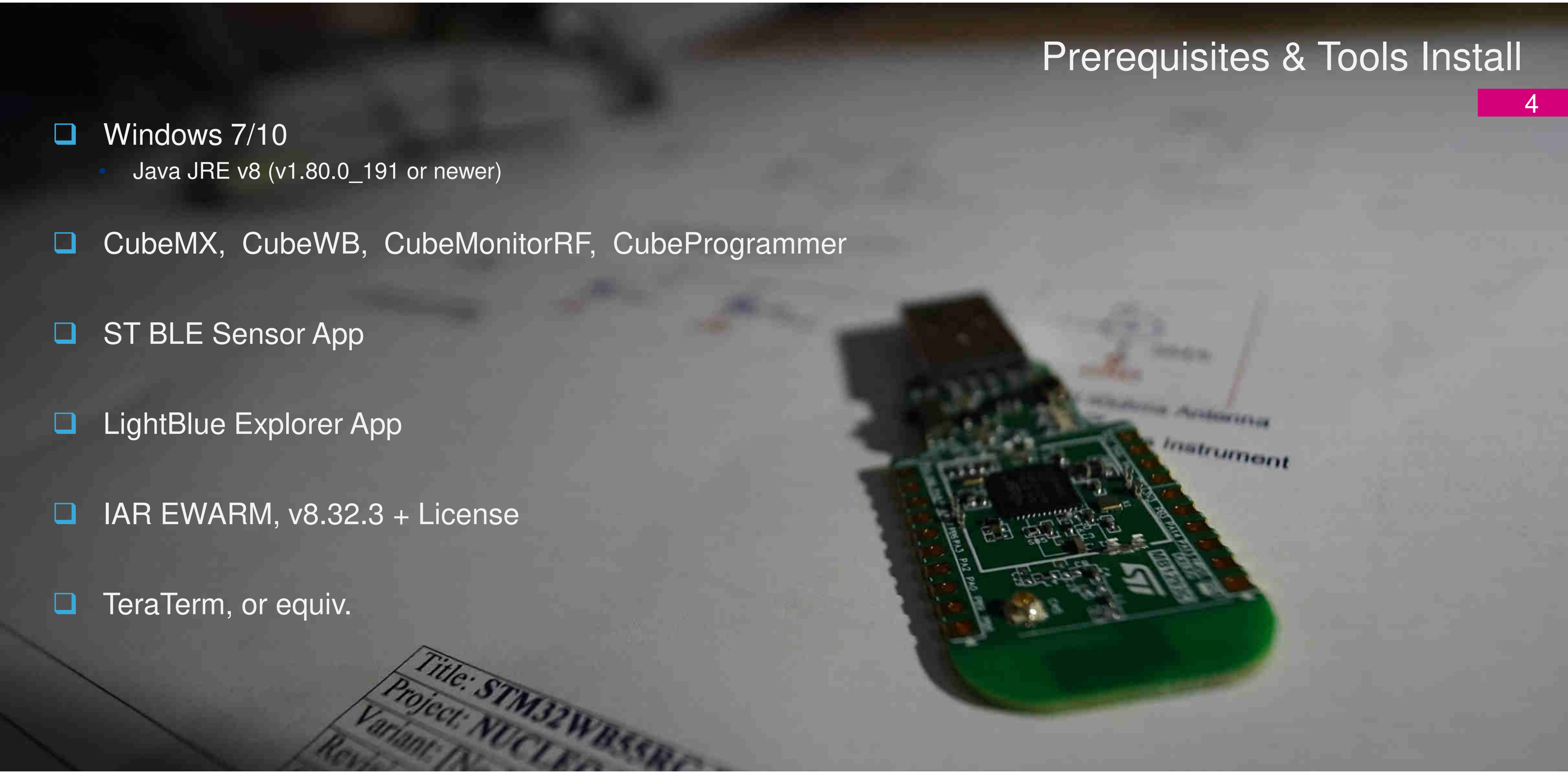


1:00	Tools install
	A few words..
	Hands-On: CubeMX
	BLE Basics
	Hands-On: HRM
3:00	Break
	Architecture
	Hands-On: CubeMonitorRF
	More WB Detail
	Hardware & Software resources
	Hands-On: Cable Replacement
4:45 – 5	Wrap-up, Q&A

life.augmented

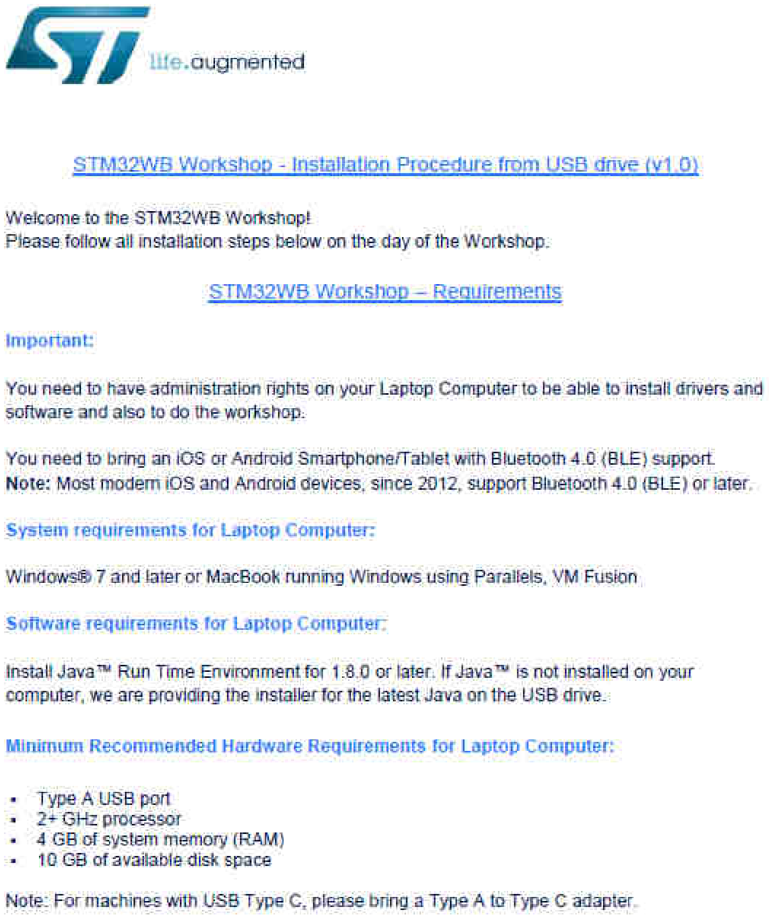
STM32 WB

Prerequisites & Tools Install



- □ Windows 7/10
  - Java JRE v8 (v1.80.0_191 or newer)
- □ CubeMX, CubeWB, CubeMonitorRF, CubeProgrammer
- □ ST BLE Sensor App
- □ LightBlue Explorer App
- □ IAR EWARM, v8.32.3 + License
- □ TeraTerm, or equiv.

Install directions on USB stick



STM32WB Workshop - Installation Procedure from USB drive (v1.0)

Welcome to the STM32WB Workshop!
Please follow all installation steps below on the day of the Workshop.

STM32WB Workshop – Requirements

Important:

You need to have administration rights on your Laptop Computer to be able to install drivers and software and also to do the workshop.

You need to bring an iOS or Android Smartphone/Tablet with Bluetooth 4.0 (BLE) support.
Note: Most modern iOS and Android devices, since 2012, support Bluetooth 4.0 (BLE) or later.

System requirements for Laptop Computer:

Windows® 7 and later or MacBook running Windows using Parallels, VM Fusion

Software requirements for Laptop Computer:

Install Java™ Run Time Environment for 1.8.0 or later. If Java™ is not installed on your computer, we are providing the installer for the latest Java on the USB drive.

Minimum Recommended Hardware Requirements for Laptop Computer:

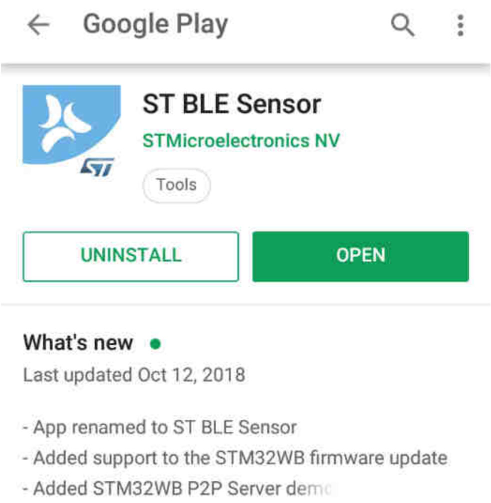
- Type A USB port
- 2+ GHz processor
- 4 GB of system memory (RAM)
- 10 GB of available disk space

Note: For machines with USB Type C, please bring a Type A to Type C adapter.

ST BLE Sensor app, v4.3

Play Store

App Store



ST BLE Sensor

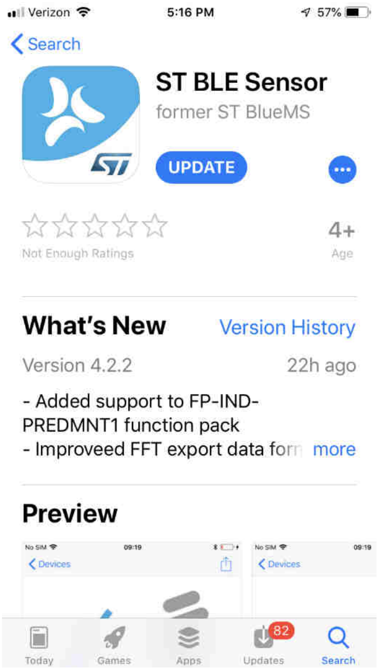
STMicroelectronics NV

Tools

What's new

Last updated Oct 12, 2018

- App renamed to ST BLE Sensor
- Added support to the STM32WB firmware update
- Added STM32WB P2P Server demo



ST BLE Sensor

former ST BlueMS

Not Enough Ratings

4+

Age

What's New

Version History

Version 4.2.2

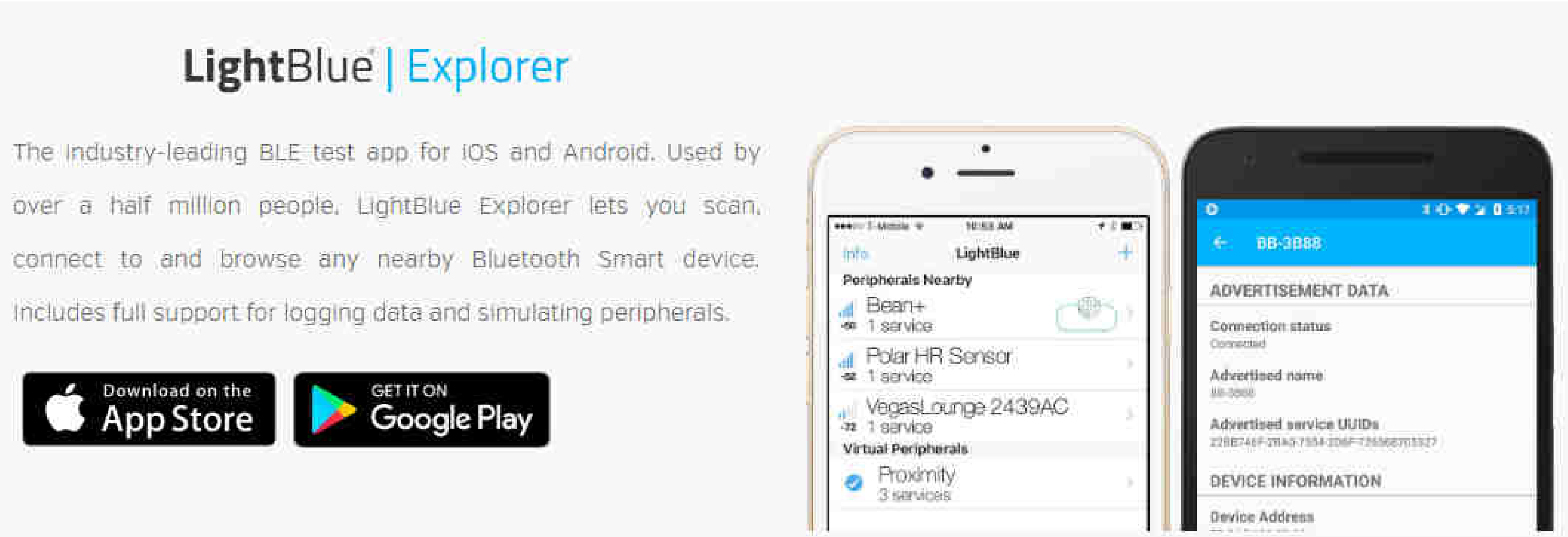
22h ago

- Added support to FP-IND-PREDMNT1 function pack
- Improveed FFT export data forr more

Preview

LightBlue Explorer

PunchThrough



LightBlue | Explorer

The industry-leading BLE test app for iOS and Android. Used by over a half million people, LightBlue Explorer lets you scan, connect to and browse any nearby Bluetooth Smart device. Includes full support for logging data and simulating peripherals.

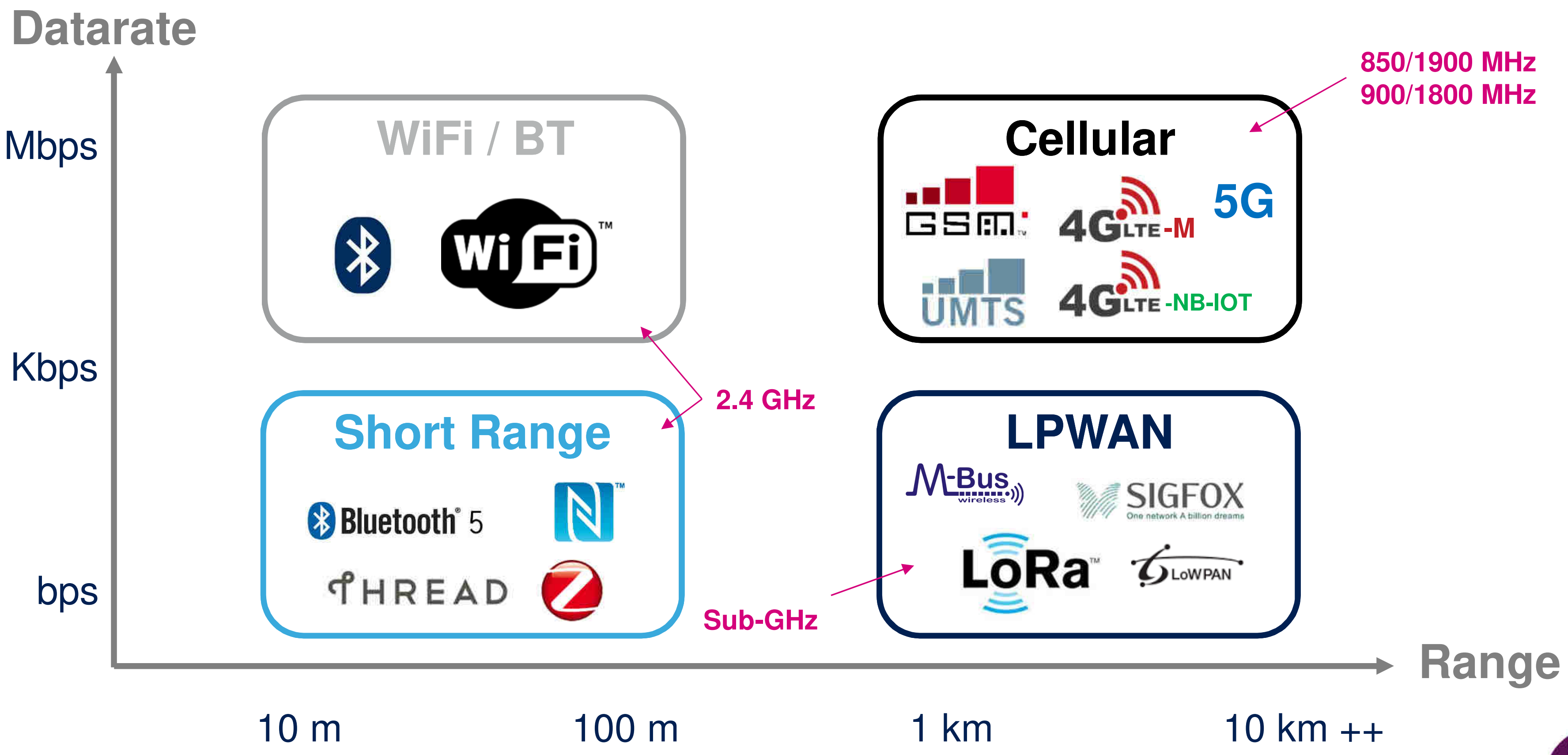
Download on the App Store

GET IT ON Google Play

life.augmented

STM32 WB

Communication Technologies



Datarate

Mbps

Kbps

bps

Range

10 m

100 m

1 km

10 km ++

WiFi / BT

Cellular

850/1900 MHz
900/1800 MHz

2.4 GHz

Short Range

Sub-GHz

LPWAN

Low-data-rate 2.4GHz connectivity

Insulin Pump

Hearing Aid

Watches

Glasses

Locator Tag

Fitness

Bluetooth Smart
Point-to-point communication with smartphones and other wireless devices

Alarm

Heating/Cooling

Door lock

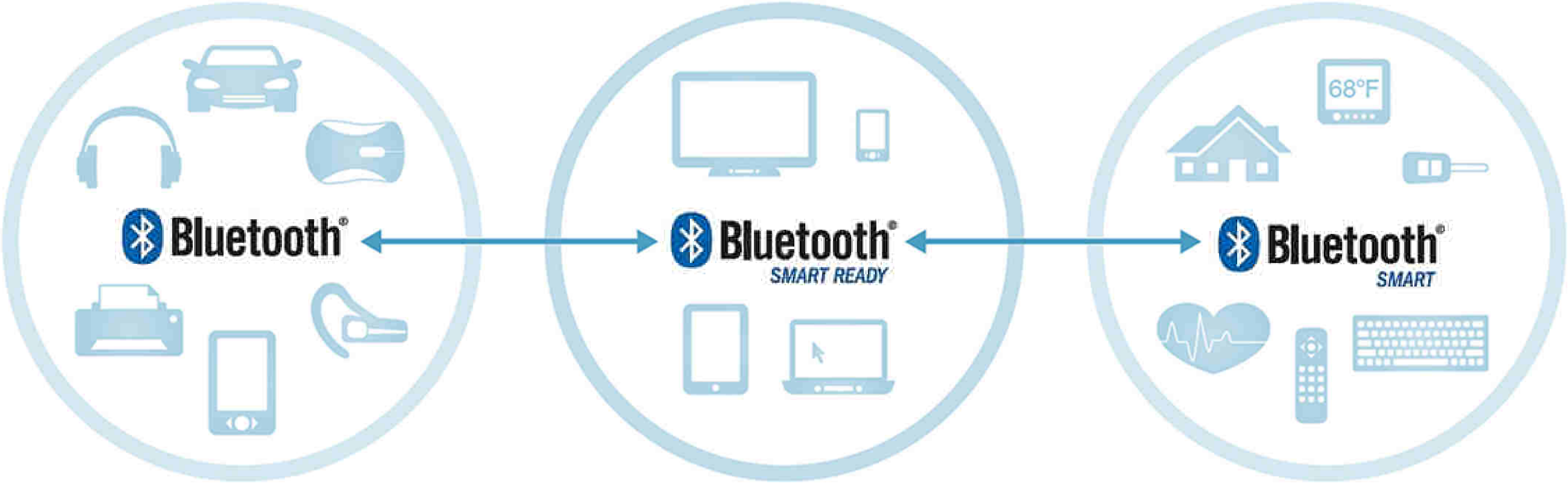
White goods

Smoke detectors

Lighting

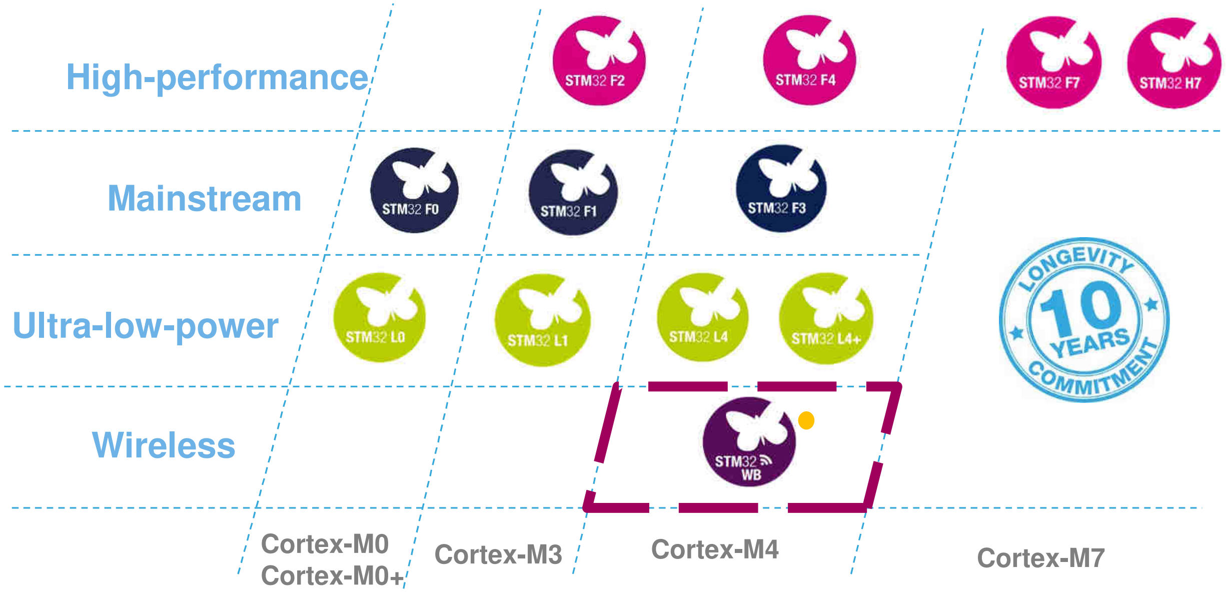
BLE Mesh / 802.15.4
Home automation with Mesh network

Classic? Smart Ready? Smart?



"SMART READY" and ***"SMART"*** are abandoned markings

STM32 Portfolio

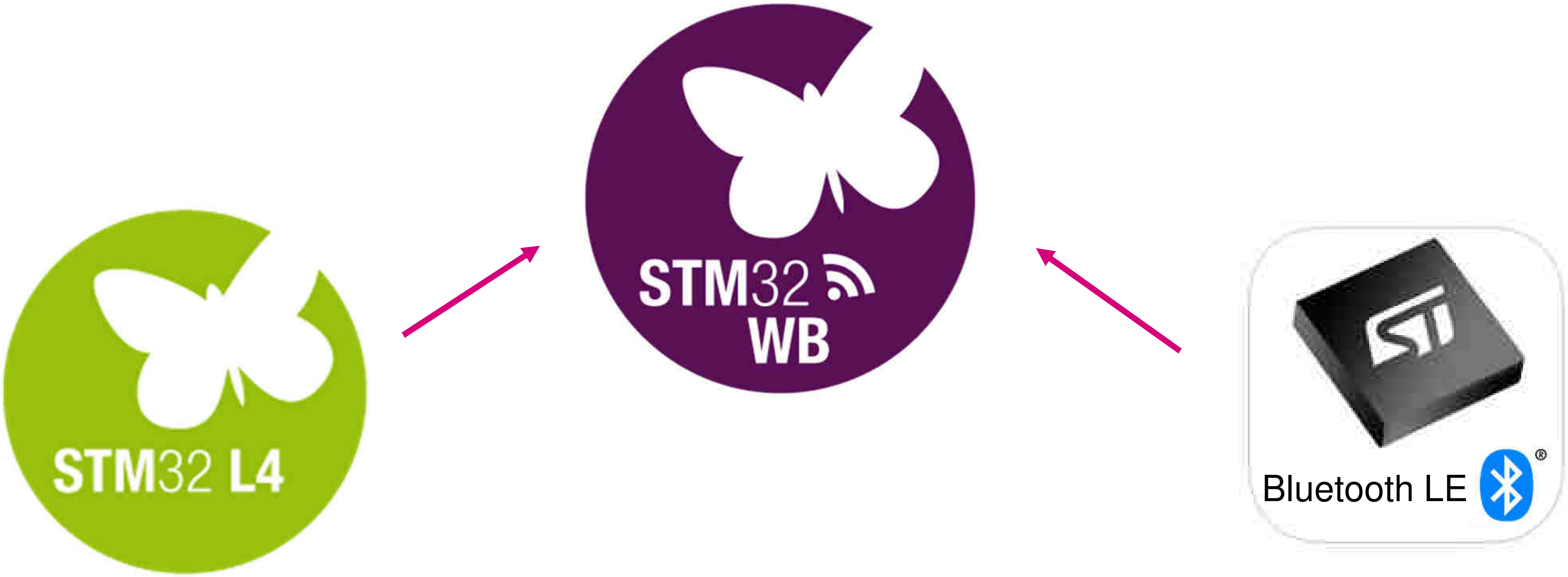


	Cortex-M0 Cortex-M0+	Cortex-M3	Cortex-M4	Cortex-M7
High-performance		STM32 F2	STM32 F4	STM32 F7 STM32 H7
Mainstream	STM32 F0	STM32 F1	STM32 F3	
Ultra-low-power	STM32 L0	STM32 L1	STM32 L4 STM32 L4+	
Wireless			STM32 WB ●	

● Legend: Cortex-M0+ Radio Co-processor

More than
40,000 customers

Ultra-low-power and RF



STM32WB Key Takeaways

Bluetooth® 5

THREAD

ZigBee®

Multi-protocol

Dual-core
Ultra-Low Power

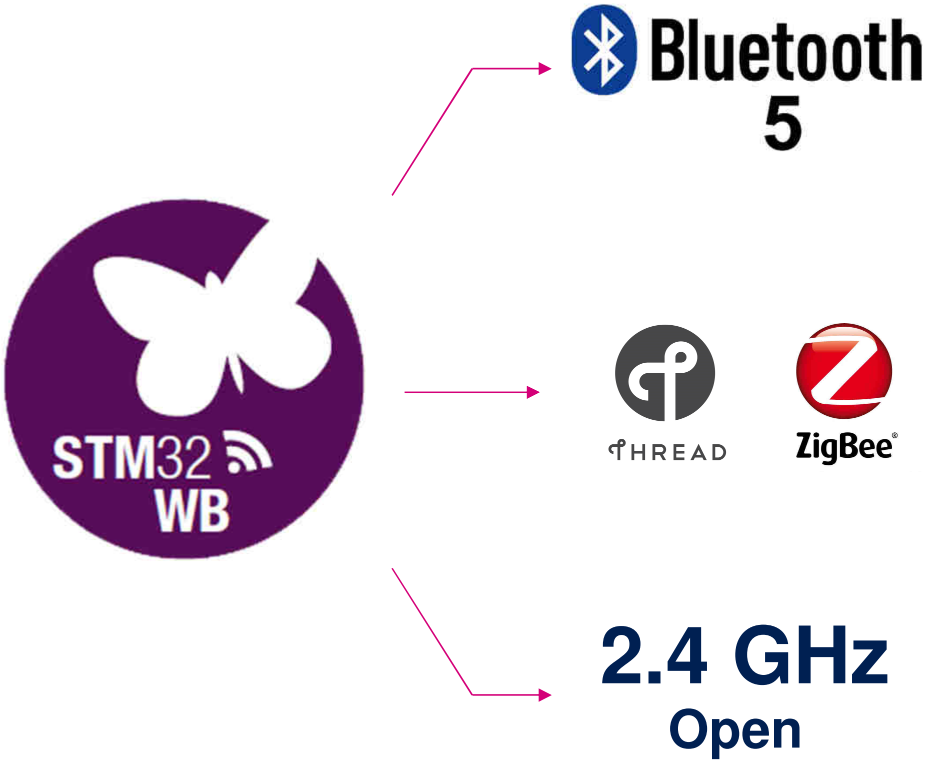
Secure

STM32 Cube

STM32 CubeMonitor-RF

Comprehensive Ecosystem

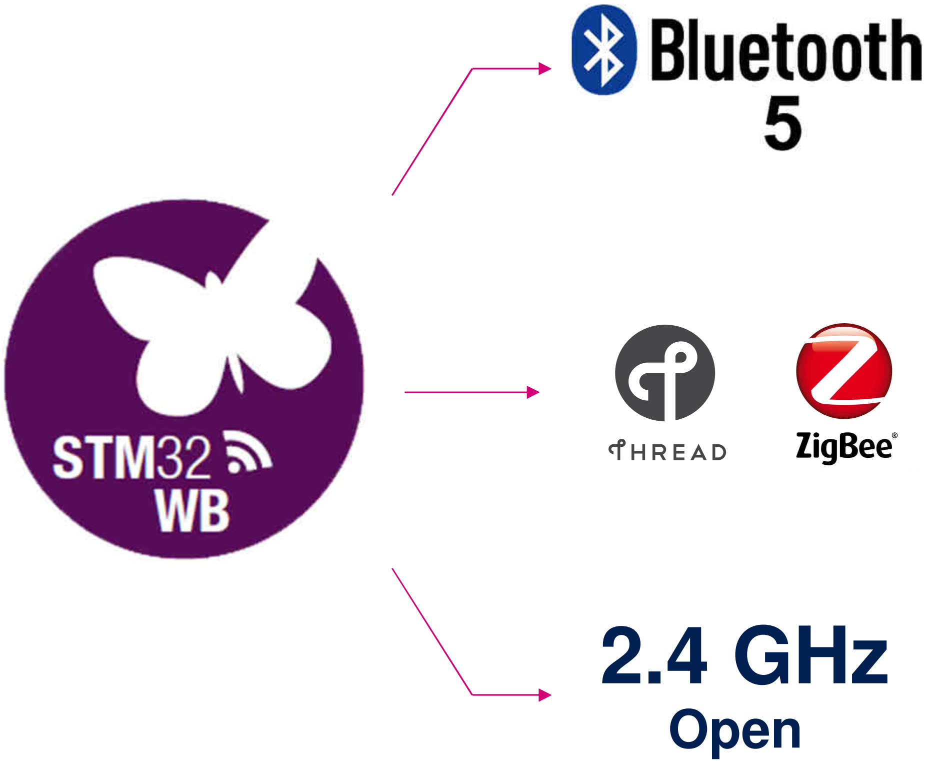
Multiprotocol



Bluetooth 5

2.4 GHz Open

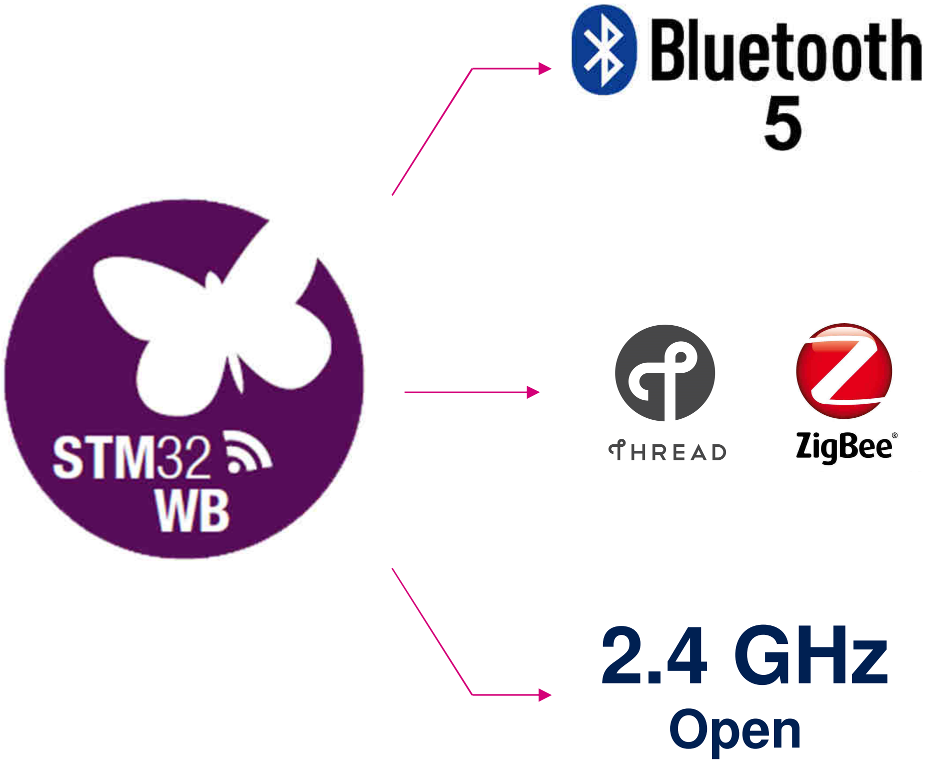
Multiprotocol



- Fully certified
- 2Mbps
- BLE Mesh

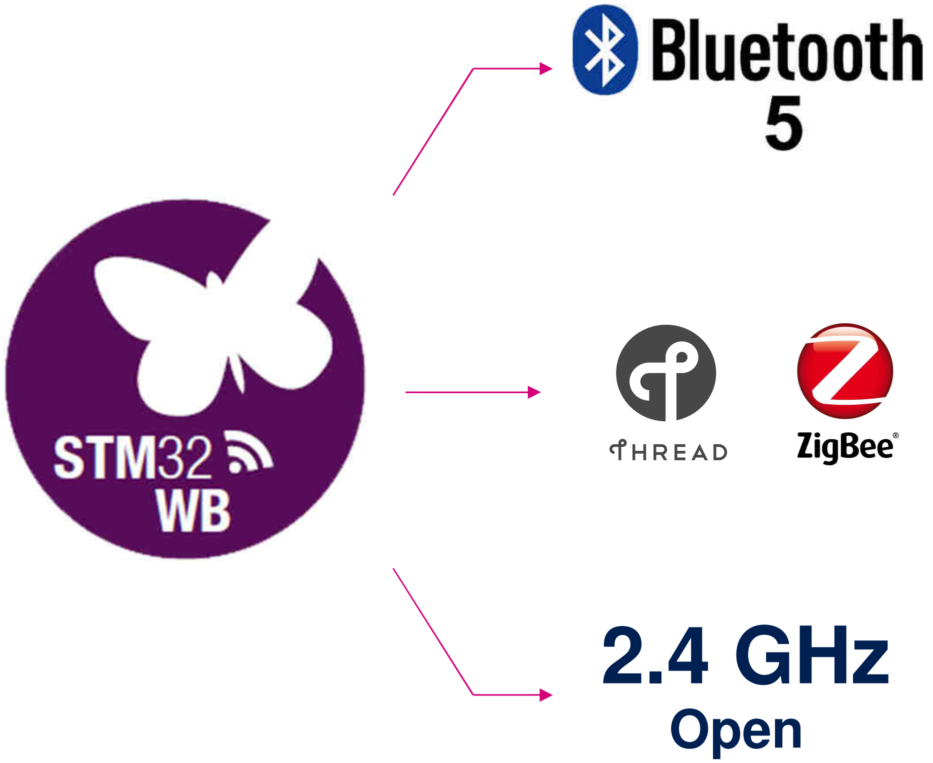
2.4 GHz
Open

Multiprotocol



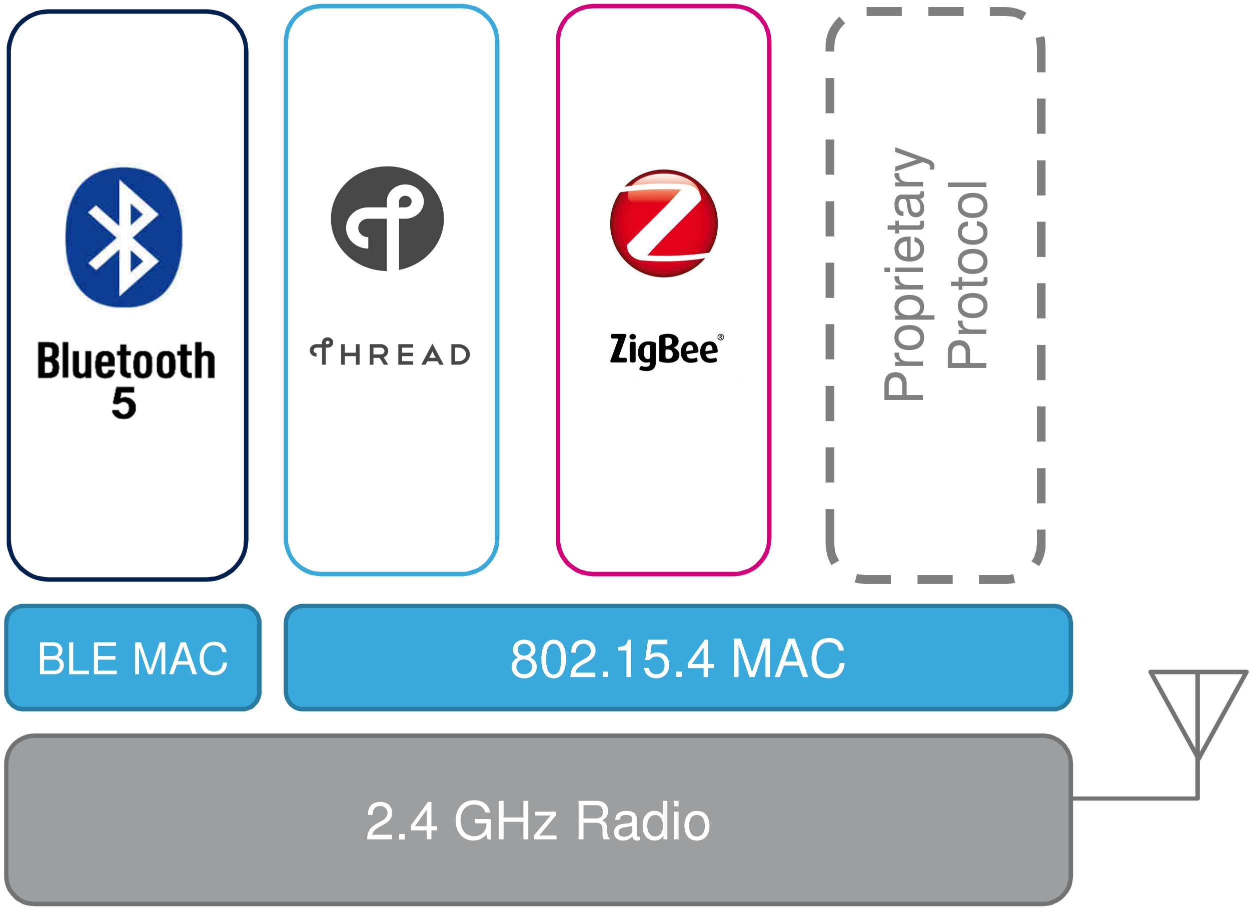
- Zigbee 3.0
- OpenThread
- Concurrent BLE + OpenThread

Multiprotocol



- OpenMAC

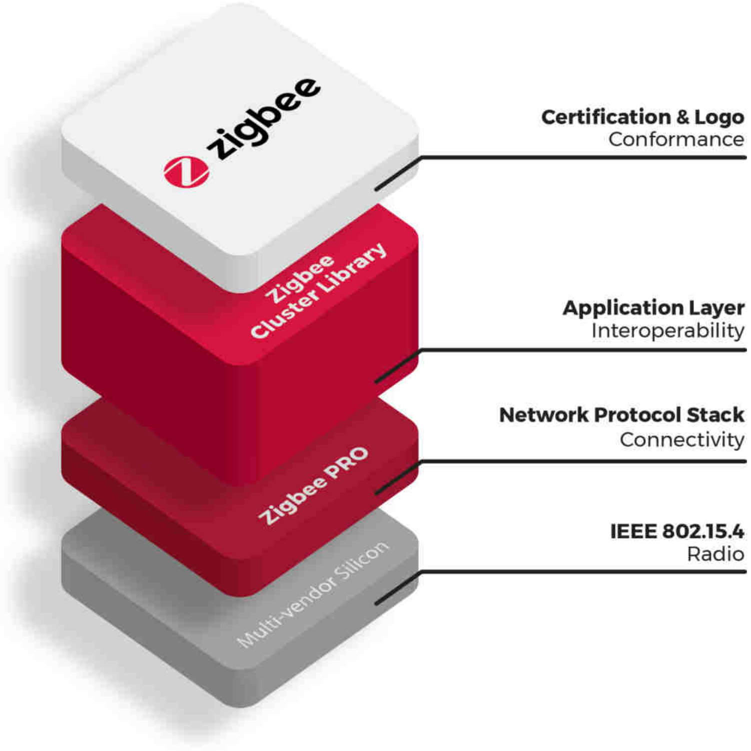
Stack + MAC + PHY



Zigbee 3.0 Stack

- Fully certified
- Legacy cluster support
- Revision R21 to R23.
- Coming in Q3

Coming soon in the ecosystem !



Certification & Logo Conformance

Application Layer Interoperability

Network Protocol Stack Connectivity

IEEE 802.15.4 Radio

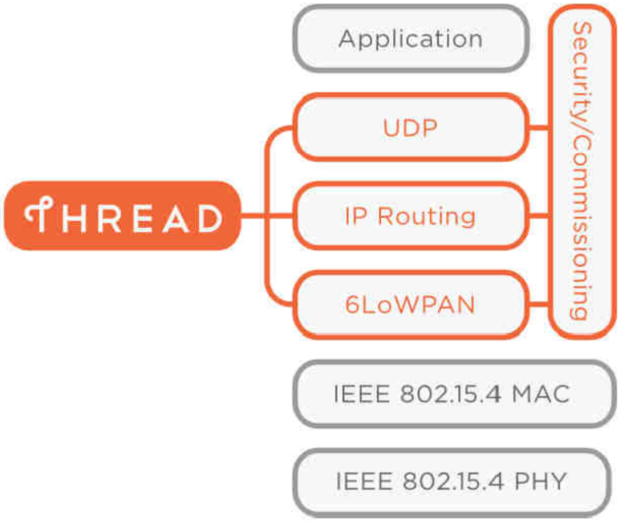
THREAD What it delivers

ST is member of Thread Group (Contributor level)

A secure wireless mesh network for your home and its connected products

- Built on well-proven, existing technologies
- Uses 6LoWPAN and carries IPv6 natively
- Runs on existing 802.15.4 silicon
- New security architecture to make it simple and secure to add / remove products
- 250+ products per network
- Designed for very low power operation
- Reliable for critical infrastructure

Can support many popular application layer protocols and platforms

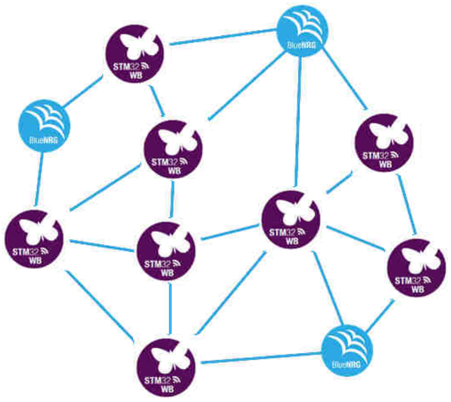


A software upgrade can add Thread to currently shipping 802.15.4 products

Bluetooth Mesh vs Thread

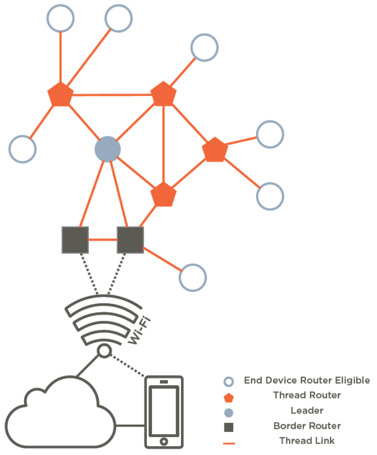
Bluetooth Mesh

- Based on Bluetooth 4.0 and later
- Broadcast type, flood the network with messages, no routing
- Shorter range, 3kbps application data rate, 1Mbps on air data rate
- High power consumption



Thread

- IPv6-based using 802.15.4 MAC
- Routing table approach with network self healing
- Medium range, 40Kbps application data rate, 250Kbps on air data rate
- Low power consumption

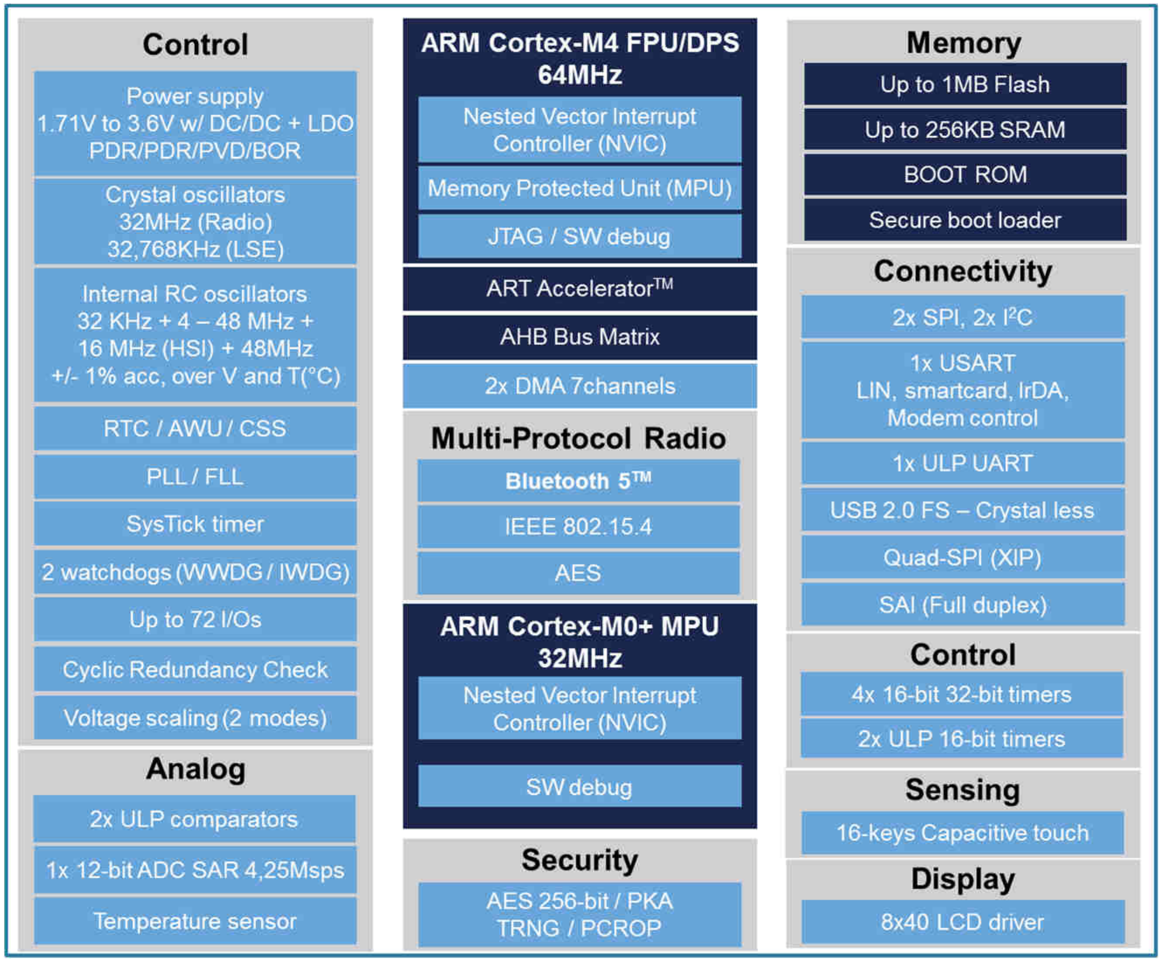


Block Diagram

- Radio with integrated balun
  - Output power: +6.0 dBm
  - BLE RX sensitivity: -96 dBm
  - 802.15.4 RX sensitivity: -100 dBm
  - RX: 4.5mA
  - TX: 5.2mA (0dBm)

- -40°C to +105°C

- Packages
  - QFN48 / 68
  - WLCSP100
  - BGA129



Control

- Power supply 1.71V to 3.6V w/ DC/DC + LDO PDR/PDR/PVD/BOR
- Crystal oscillators 32MHz (Radio) 32,768KHz (LSE)
- Internal RC oscillators 32 KHz + 4 – 48 MHz + 16 MHz (HSI) + 48MHz +/- 1% acc, over V and T(°C)
- RTC / AWU / CSS
- PLL / FLL
- SysTick timer
- 2 watchdogs (WWDG / IWDG)
- Up to 72 I/Os
- Cyclic Redundancy Check
- Voltage scaling (2 modes)

Analog

- 2x ULP comparators
- 1x 12-bit ADC SAR 4,25Msps
- Temperature sensor

ARM Cortex-M4 FPU/DPS 64MHz

- Nested Vector Interrupt Controller (NVIC)
- Memory Protected Unit (MPU)
- JTAG / SW debug

ART Accelerator^{TM}

AHB Bus Matrix

2x DMA 7channels

Multi-Protocol Radio

- **Bluetooth 5^{TM}**
- IEEE 802.15.4
- AES

ARM Cortex-M0+ MPU 32MHz

- Nested Vector Interrupt Controller (NVIC)
- SW debug

Security

- AES 256-bit / PKA TRNG / PCROP

Memory

- Up to 1MB Flash
- Up to 256KB SRAM
- BOOT ROM
- Secure boot loader

Connectivity

- 2x SPI, 2x I^{2}C
- 1x USART LIN, smartcard, IrDA, Modem control
- 1x ULP UART
- USB 2.0 FS – Crystal less
- Quad-SPI (XIP)
- SAI (Full duplex)

Control

- 4x 16-bit 32-bit timers
- 2x ULP 16-bit timers

Sensing

- 16-keys Capacitive touch

Display

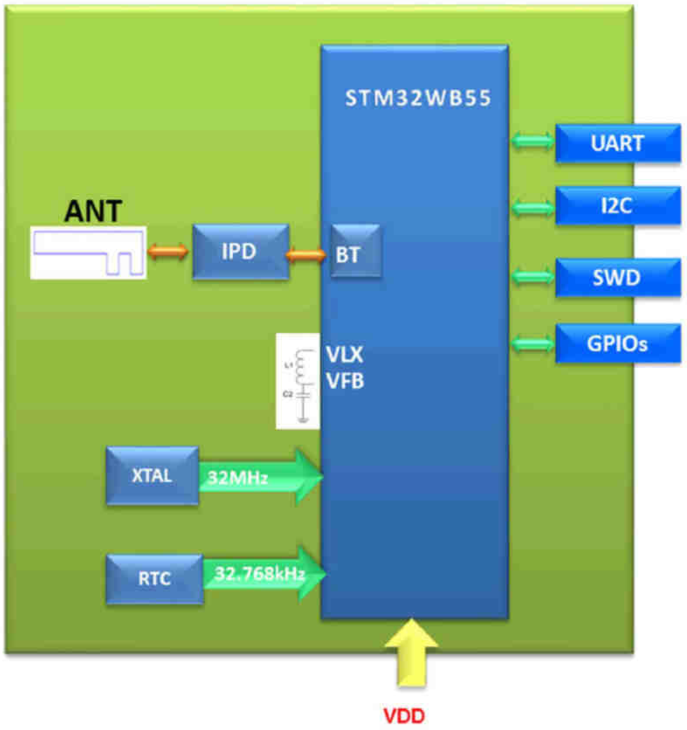
- 8x40 LCD driver

STM32WB55 Series Portfolio

<table>
<tr><th>Flash / RAM</th><th>48 pins
7x7mm
UQFN (p=0.5)</th><th>68 pins
8x8mm
VQFN (p=0.4)</th><th>100 pins
4.39x4.37mm
WLCSP (p=0.4)</th><th>129 pins
7x7mm
BGA (p=0.5)</th></tr>
<tr><td>1M / 256K</td><td>STM32WB55CG</td><td>STM32WB55RG</td><td colspan="2">STM32WB55VG</td></tr>
<tr><td>512K / 256K</td><td>STM32WB55CE</td><td>STM32WB55RE</td><td colspan="2">STM32WB55VE</td></tr>
<tr><td>256K / 128K</td><td>STM32WB55CC</td><td>STM32WB55RC</td><td colspan="2">STM32WB55VC</td></tr>
</table>

STM32WB55 Module

- ST Branded
- Pre-Certified
- Chip Antenna
- 10x10mm
- Large GPIO count
- Pin pitch = 2 layer PCB-ready
- Production in Q3



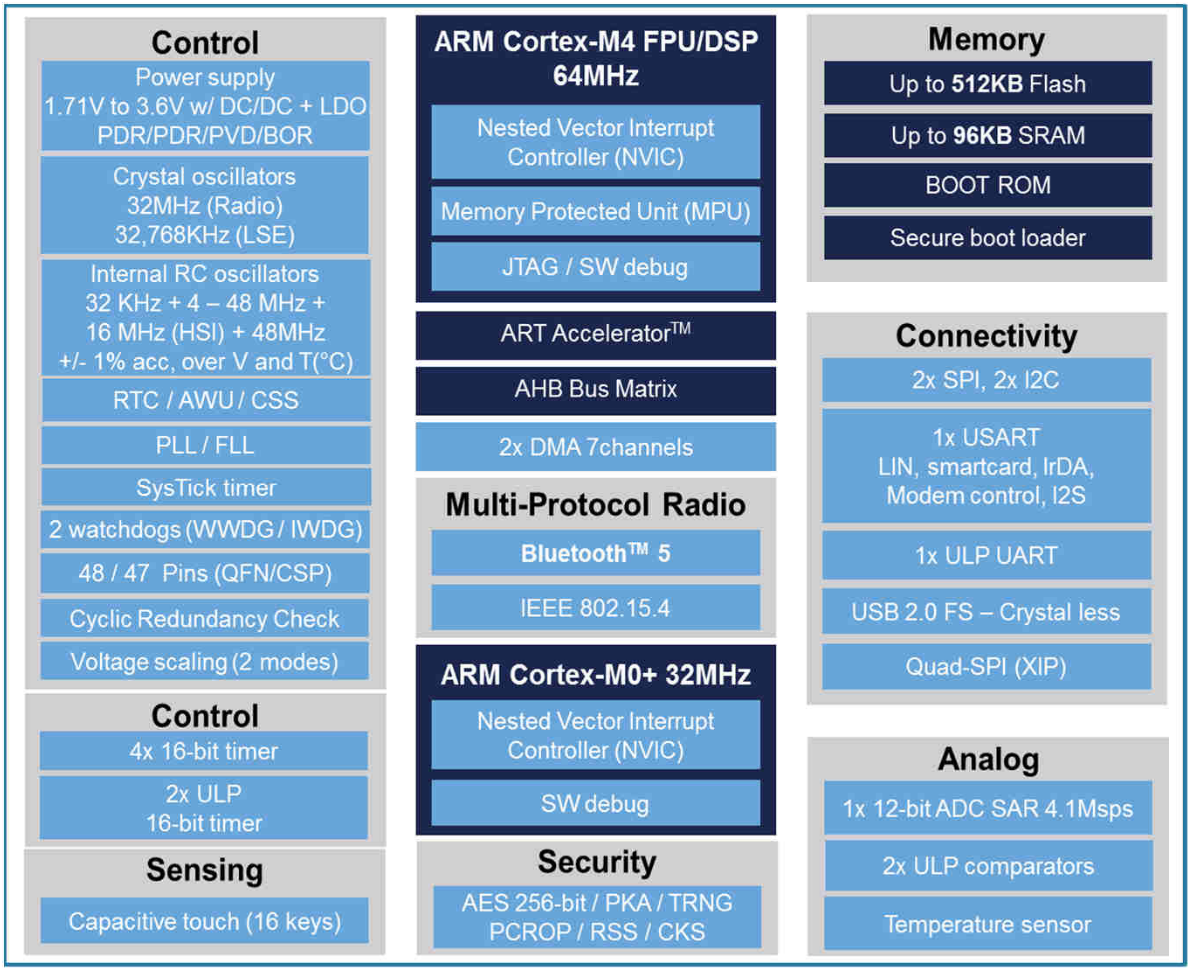
STM32WB35 – Block Diagram

256KB or 512KB Flash

96KB SRAM

- QFN48
- WLCSP47

Late 2019



Control

Power supply
1.71V to 3.6V w/ DC/DC + LDO
PDR/PDR/PVD/BOR

Crystal oscillators
32MHz (Radio)
32,768KHz (LSE)

Internal RC oscillators
32 KHz + 4 – 48 MHz +
16 MHz (HSI) + 48MHz
+/- 1% acc, over V and T(°C)

RTC / AWU / CSS

PLL / FLL

SysTick timer

2 watchdogs (WWDG / IWDG)

48 / 47 Pins (QFN/CSP)

Cyclic Redundancy Check

Voltage scaling (2 modes)

Control

4x 16-bit timer

2x ULP
16-bit timer

Sensing

Capacitive touch (16 keys)

ARM Cortex-M4 FPU/DSP 64MHz

Nested Vector Interrupt Controller (NVIC)

Memory Protected Unit (MPU)

JTAG / SW debug

ART Accelerator™

AHB Bus Matrix

2x DMA 7channels

Multi-Protocol Radio

Bluetooth™ 5

IEEE 802.15.4

ARM Cortex-M0+ 32MHz

Nested Vector Interrupt Controller (NVIC)

SW debug

Security

AES 256-bit / PKA / TRNG
PCROP / RSS / CKS

Memory

Up to **512KB** Flash

Up to **96KB** SRAM

BOOT ROM

Secure boot loader

Connectivity

2x SPI, 2x I2C

1x USART
LIN, smartcard, IrDA,
Modem control, I2S

1x ULP UART

USB 2.0 FS – Crystal less

Quad-SPI (XIP)

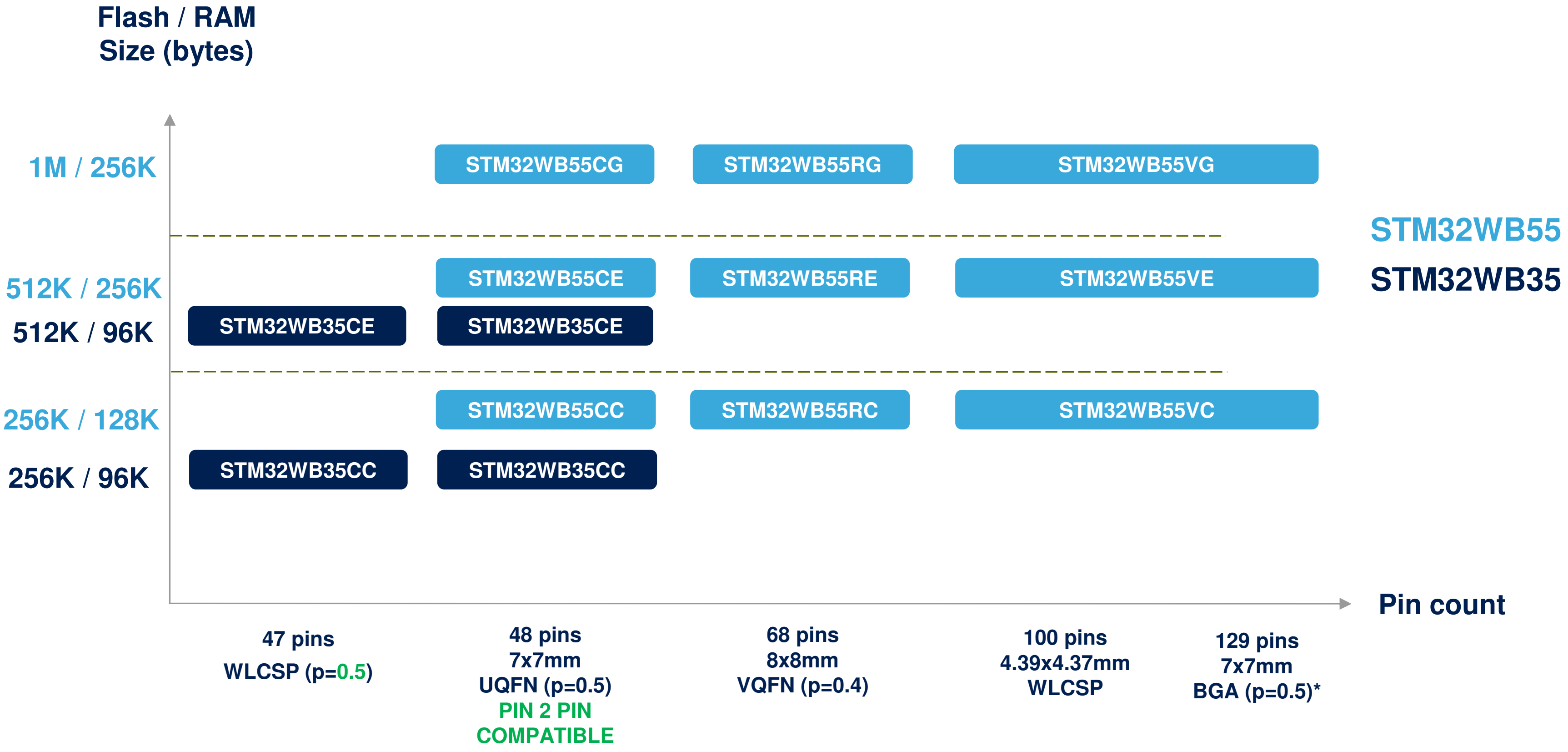
Analog

1x 12-bit ADC SAR 4.1Msps

2x ULP comparators

Temperature sensor

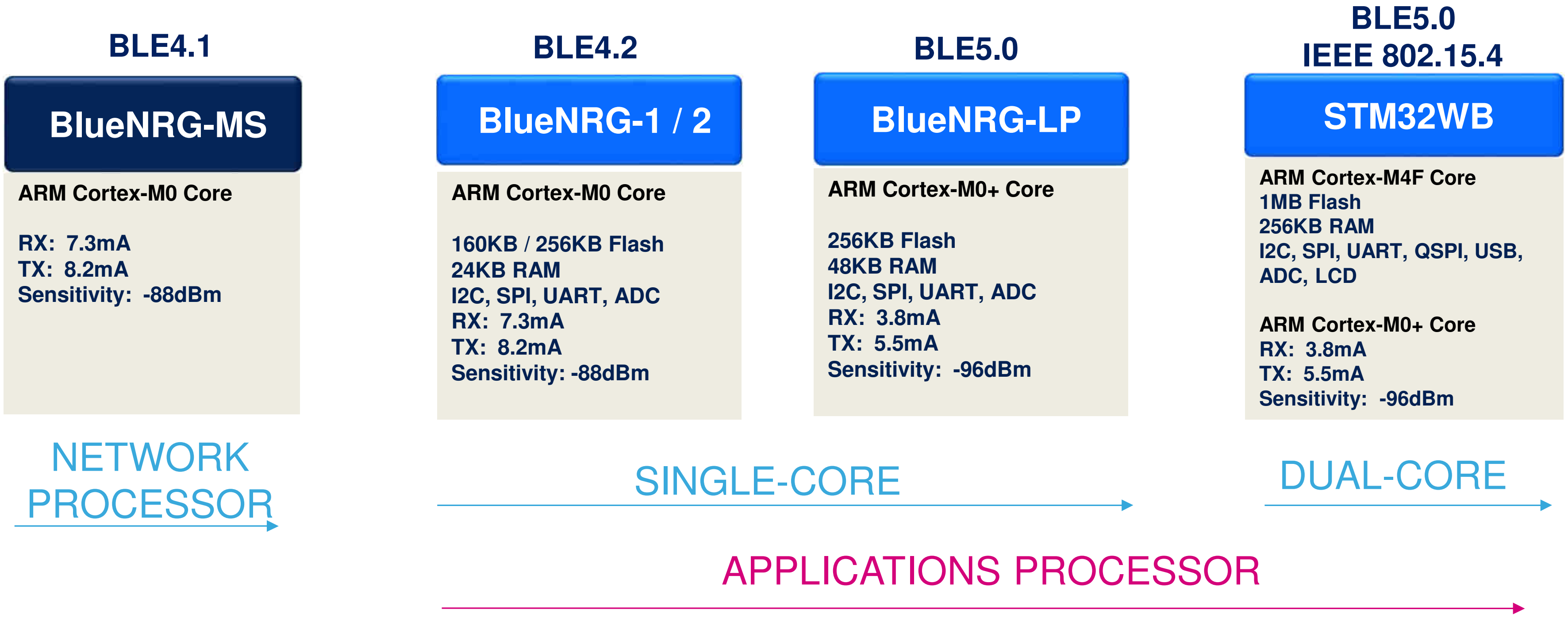
Positioning



Flash / RAM Size (bytes)	47 pins WLCSP (p=0.5)	48 pins 7x7mm UQFN (p=0.5) PIN 2 PIN COMPATIBLE	68 pins 8x8mm VQFN (p=0.4)	100 pins 4.39x4.37mm WLCSP	129 pins 7x7mm BGA (p=0.5)*
1M / 256K		STM32WB55CG	STM32WB55RG	STM32WB55VG	
512K / 256K		STM32WB55CE	STM32WB55RE	STM32WB55VE	
512K / 96K	STM32WB35CE	STM32WB35CE			
256K / 128K		STM32WB55CC	STM32WB55RC	STM32WB55VC	
256K / 96K	STM32WB35CC	STM32WB35CC			

* Exact part number on BGA 129 To be confirmed

STM32WB & BlueNRG Series



BLE4.1

BlueNRG-MS

ARM Cortex-M0 Core

RX: 7.3mA
TX: 8.2mA
Sensitivity: -88dBm

BLE4.2

BlueNRG-1 / 2

ARM Cortex-M0 Core

160KB / 256KB Flash
24KB RAM
I2C, SPI, UART, ADC
RX: 7.3mA
TX: 8.2mA
Sensitivity: -88dBm

BLE5.0

BlueNRG-LP

ARM Cortex-M0+ Core

256KB Flash
48KB RAM
I2C, SPI, UART, ADC
RX: 3.8mA
TX: 5.5mA
Sensitivity: -96dBm

BLE5.0
IEEE 802.15.4

STM32WB

ARM Cortex-M4F Core
1MB Flash
256KB RAM
I2C, SPI, UART, QSPI, USB, ADC, LCD

ARM Cortex-M0+ Core
RX: 3.8mA
TX: 5.5mA
Sensitivity: -96dBm

NETWORK PROCESSOR

SINGLE-CORE

DUAL-CORE

APPLICATIONS PROCESSOR

Cube Tools

STM32CubeMX

STM32CubeProgrammer

STM32CubeMonitorRF

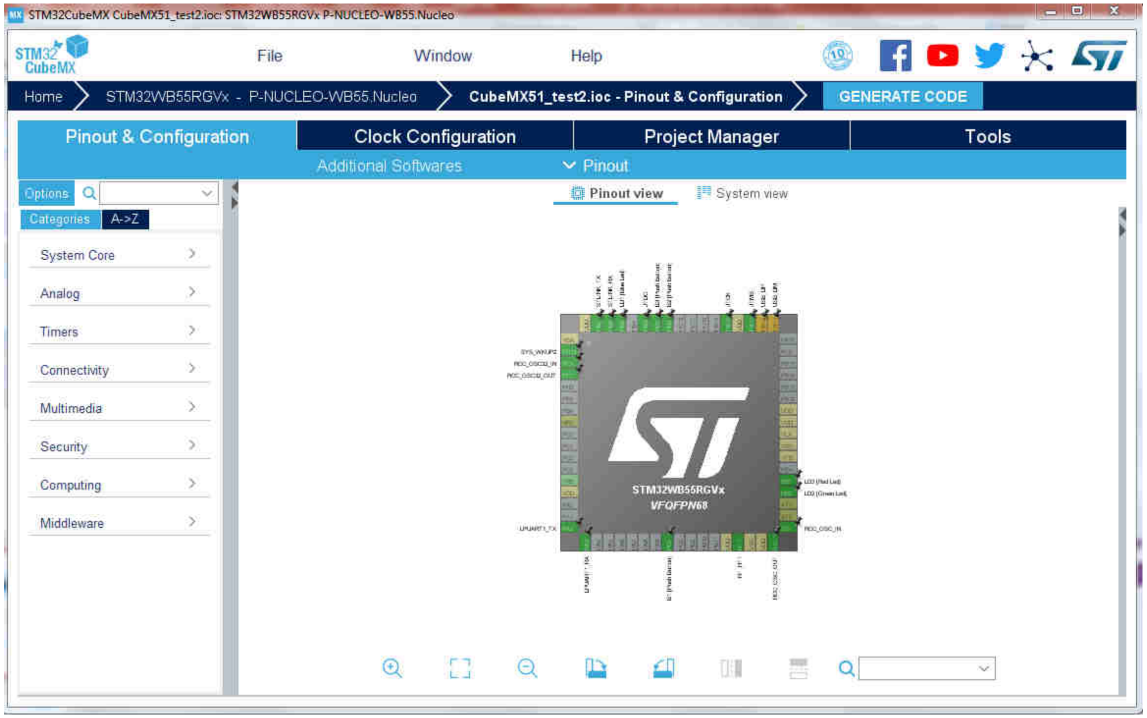
STM32CubeWB

STM32CubeMX

STM32CubeProgrammer

STM32CubeMonitorRF

CubeWB HAL Firmware

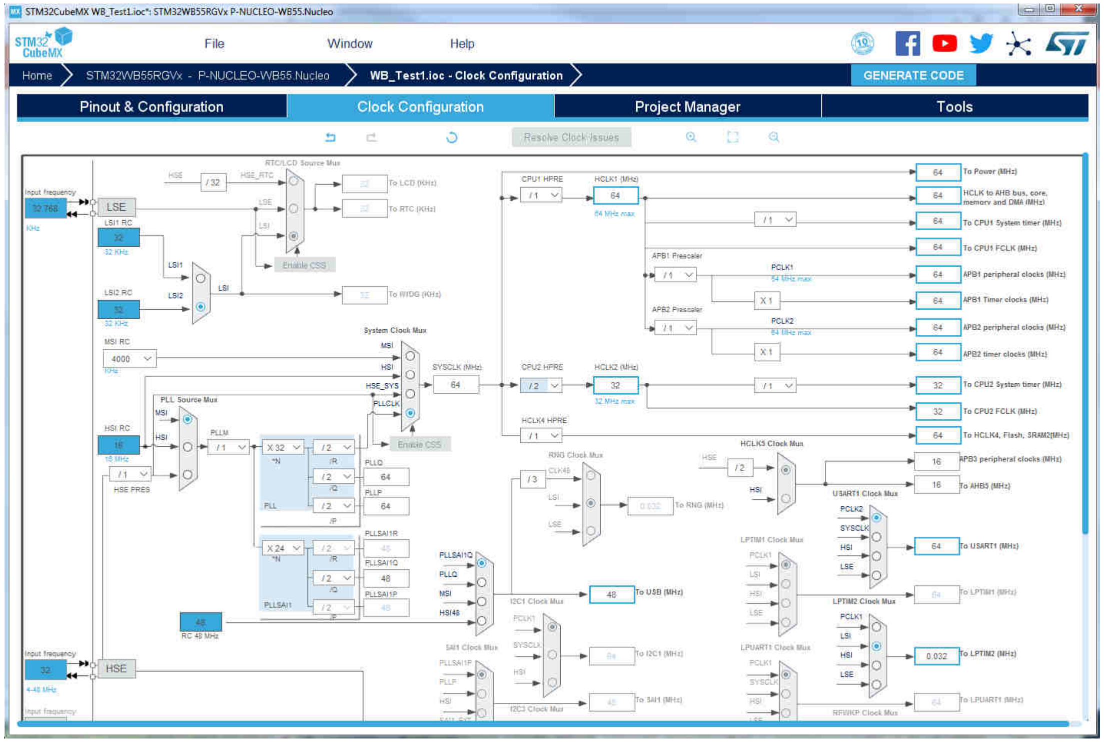


STM32CubeMX

STM32CubeProgrammer

STM32CubeMonitorRF

CubeWB HAL Firmware

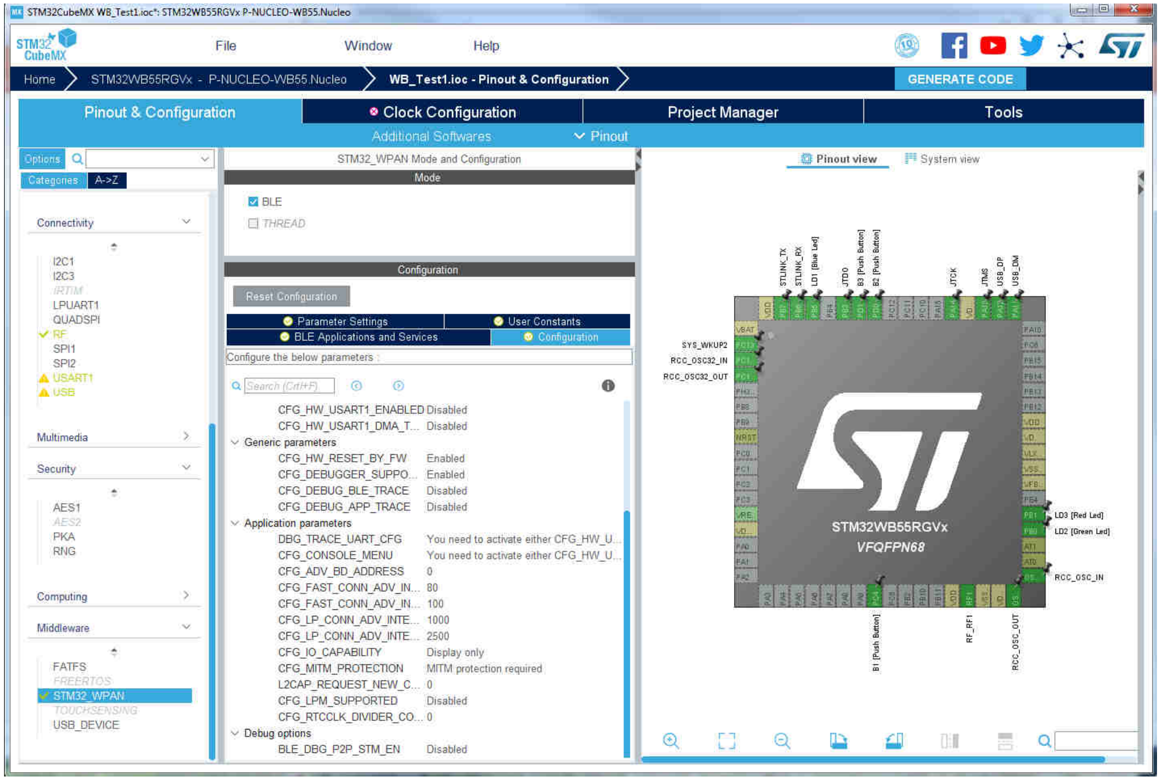


STM32CubeMX

STM32CubeProgrammer

STM32CubeMonitorRF

CubeWB HAL Firmware



☑ BLE

□ THREAD

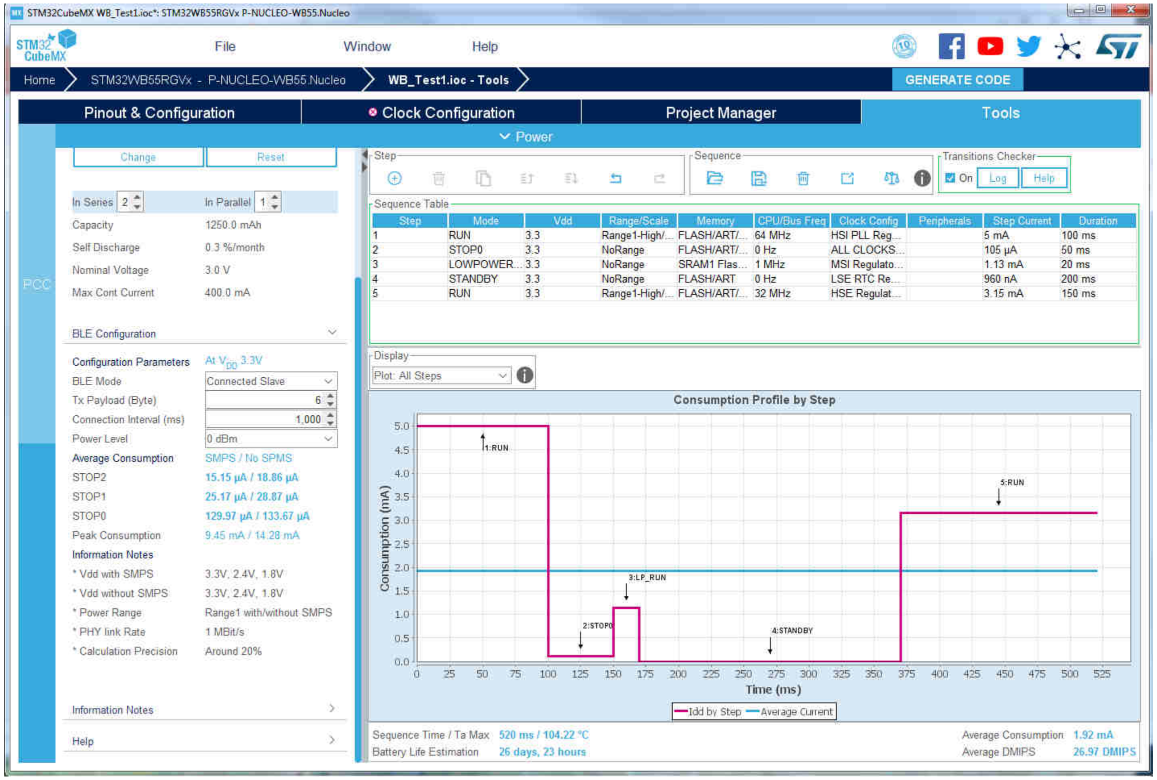
CFG_HW_USART1_ENABLED	Disabled
CFG_HW_USART1_DMA_T...	Disabled
Generic parameters	
CFG_HW_RESET_BY_FW	Enabled
CFG_DEBUGGER_SUPPO...	Enabled
CFG_DEBUG_BLE_TRACE	Disabled
CFG_DEBUG_APP_TRACE	Disabled
Application parameters	
DBG_TRACE_UART_CFG	You need to activate either CFG_HW_U...
CFG_CONSOLE_MENU	You need to activate either CFG_HW_U...
CFG_ADV_BD_ADDRESS	0
CFG_FAST_CONN_ADV_IN...	80
CFG_FAST_CONN_ADV_IN...	100
CFG_LP_CONN_ADV_INTE...	1000
CFG_LP_CONN_ADV_INTE...	2500
CFG_IO_CAPABILITY	Display only
CFG_MITM_PROTECTION	MITM protection required
L2CAP_REQUEST_NEW_C...	0
CFG_LPM_SUPPORTED	Disabled
CFG_RTCCLK_DIVIDER_CO...	0
Debug options	
BLE_DBG_P2P_STM_EN	Disabled

STM32CubeMX

STM32CubeProgrammer

STM32CubeMonitorRF

CubeWB HAL Firmware



In Series 2 In Parallel 1

Capacity 1250.0 mAh

Self Discharge 0.3 %/month

Nominal Voltage 3.0 V

Max Cont Current 400.0 mA

BLE Configuration

Configuration Parameters At V_{DD} 3.3V

BLE Mode Connected Slave

Tx Payload (Byte) 6

Connection Interval (ms) 1,000

Power Level 0 dBm

Average Consumption SMPS / No SPMS

STOP2 15.15 µA / 18.86 µA

STOP1 25.17 µA / 28.87 µA

STOP0 129.97 µA / 133.67 µA

Peak Consumption 9.45 mA / 14.28 mA

Information Notes

* Vdd with SMPS 3.3V, 2.4V, 1.8V

* Vdd without SMPS 3.3V, 2.4V, 1.8V

* Power Range Range1 with/without SMPS

* PHY link Rate 1 MBit/s

* Calculation Precision Around 20%

Sequence Table

Step	Mode	Vdd	Range/Scale	Memory	CPU/Bus Freq	Clock Config	Peripherals	Step Current	Duration
1	RUN	3.3	Range1-High/...	FLASH/ART/...	64 MHz	HSI PLL Reg...		5 mA	100 ms
2	STOP0	3.3	NoRange	FLASH/ART/...	0 Hz	ALL CLOCKS...		105 µA	50 ms
3	LOWPOWER...	3.3	NoRange	SRAM1 Flas...	1 MHz	MSI Regulato...		1.13 mA	20 ms
4	STANDBY	3.3	NoRange	FLASH/ART	0 Hz	LSE RTC Re...		960 nA	200 ms
5	RUN	3.3	Range1-High/...	FLASH/ART/...	32 MHz	HSE Regulat...		3.15 mA	150 ms

Sequence Time / Ta Max 520 ms / 104.22 °C

Battery Life Estimation 26 days, 23 hours

Average Consumption 1.92 mA

Average DMIPS 26.97 DMIPS

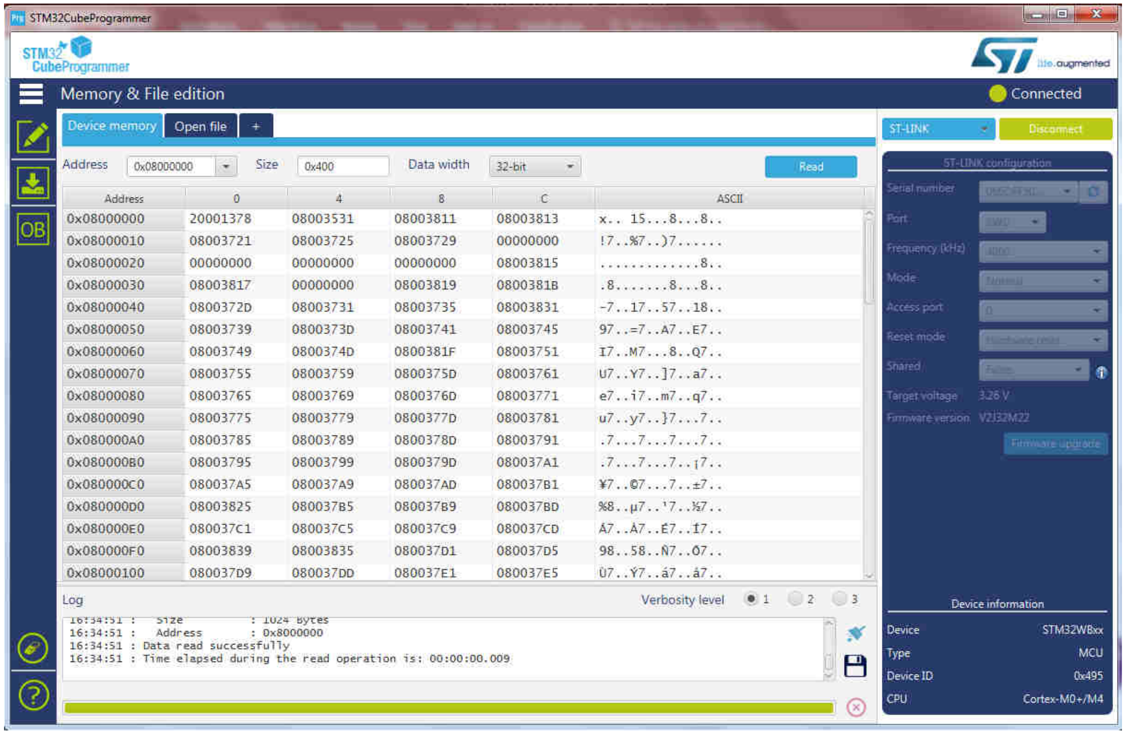
Cube Tools

STM32CubeMX

STM32CubeProgrammer

STM32CubeMonitorRF

CubeWB HAL Firmware



Address: 0x08000000 | Size: 0x400 | Data width: 32-bit

```
Address      0         4         8         C         ASCII
0x08000000   20001378  08003531  08003811  08003813  x.. 15...8...8..
0x08000010   08003721  08003725  08003729  00000000  !7..%7..)7......
0x08000020   00000000  00000000  00000000  08003815  ............8..
0x08000030   08003817  00000000  08003819  0800381B  .8.......8...8..
0x08000040   0800372D  08003731  08003735  08003831  -7..17..57..18..
0x08000050   08003739  0800373D  08003741  08003745  97..=7..A7..E7..
0x08000060   08003749  0800374D  0800381F  08003751  I7..M7...8..Q7..
0x08000070   08003755  08003759  0800375D  08003761  U7..Y7..]7..a7..
0x08000080   08003765  08003769  0800376D  08003771  e7..i7..m7..q7..
0x08000090   08003775  08003779  0800377D  08003781  u7..y7..}7...7..
0x080000A0   08003785  08003789  0800378D  08003791  .7...7...7...7..
0x080000B0   08003795  08003799  0800379D  080037A1  .7...7...7..¡7..
0x080000C0   080037A5  080037A9  080037AD  080037B1  ¥7..©7...7..±7..
0x080000D0   08003825  080037B5  080037B9  080037BD  %8..µ7..¹7..½7..
0x080000E0   080037C1  080037C5  080037C9  080037CD  Á7..Å7..É7..Í7..
0x080000F0   08003839  08003835  080037D1  080037D5  98..58..Ñ7..Õ7..
0x08000100   080037D9  080037DD  080037E1  080037E5  Ù7..Ý7..á7..å7..
```

Log — Verbosity level ● 1 ○ 2 ○ 3

```
16:34:51 :   Size          : 1024 Bytes
16:34:51 :   Address       : 0x8000000
16:34:51 : Data read successfully
16:34:51 : Time elapsed during the read operation is: 00:00:00.009
```

ST-LINK configuration

Serial number	[illegible]
Port	SWD
Frequency (kHz)	4000
Mode	Normal
Access port	0
Reset mode	Hardware reset
Shared	False
Target voltage	3.26 V
Firmware version	V2J32M22

Device information

Device	STM32WBxx
Type	MCU
Device ID	0x495
CPU	Cortex-M0+/M4

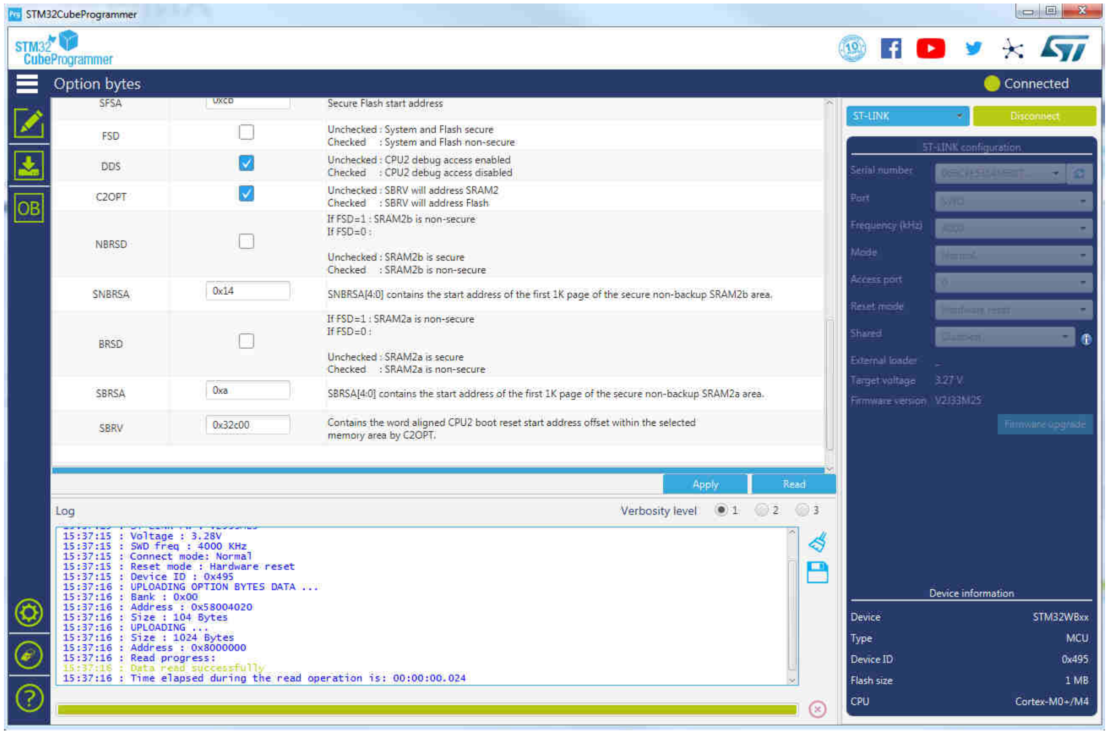
Cube Tools

STM32CubeMX

STM32CubeProgrammer

STM32CubeMonitorRF

CubeWB HAL Firmware



SFSA	0xcb	Secure Flash start address
FSD	□	Unchecked : System and Flash secure Checked : System and Flash non-secure
DDS	☑	Unchecked : CPU2 debug access enabled Checked : CPU2 debug access disabled
C2OPT	☑	Unchecked : SBRV will address SRAM2 Checked : SBRV will address Flash
NBRSD	□	If FSD=1 : SRAM2b is non-secure If FSD=0 : Unchecked : SRAM2b is secure Checked : SRAM2b is non-secure
SNBRSA	0x14	SNBRSA[4:0] contains the start address of the first 1K page of the secure non-backup SRAM2b area.
BRSD	□	If FSD=1 : SRAM2a is non-secure If FSD=0 : Unchecked : SRAM2a is secure Checked : SRAM2a is non-secure
SBRSA	0xa	SBRSA[4:0] contains the start address of the first 1K page of the secure non-backup SRAM2a area.
SBRV	0x32c00	Contains the word aligned CPU2 boot reset start address offset within the selected memory area by C2OPT.

Log

Verbosity level ● 1 ○ 2 ○ 3

```
15:37:15 : Voltage : 3.28V
15:37:15 : SWD freq : 4000 KHz
15:37:15 : Connect mode: Normal
15:37:15 : Reset mode : Hardware reset
15:37:15 : Device ID : 0x495
15:37:16 : UPLOADING OPTION BYTES DATA ...
15:37:16 : Bank : 0x00
15:37:16 : Address : 0x58004020
15:37:16 : Size : 104 Bytes
15:37:16 : UPLOADING ...
15:37:16 : Size : 1024 Bytes
15:37:16 : Address : 0x8000000
15:37:16 : Read progress:
15:37:16 : Data read successfully
15:37:16 : Time elapsed during the read operation is: 00:00:00.024
```

ST-LINK configuration

Serial number	[illegible]
Port	SWD
Frequency (kHz)	4000
Mode	Normal
Access port	0
Reset mode	Hardware reset
Shared	Disabled
External loader	_
Target voltage	3.27 V
Firmware version	V2J33M25

Device information

Device	STM32WBxx
Type	MCU
Device ID	0x495
Flash size	1 MB
CPU	Cortex-M0+/M4

life.augmented

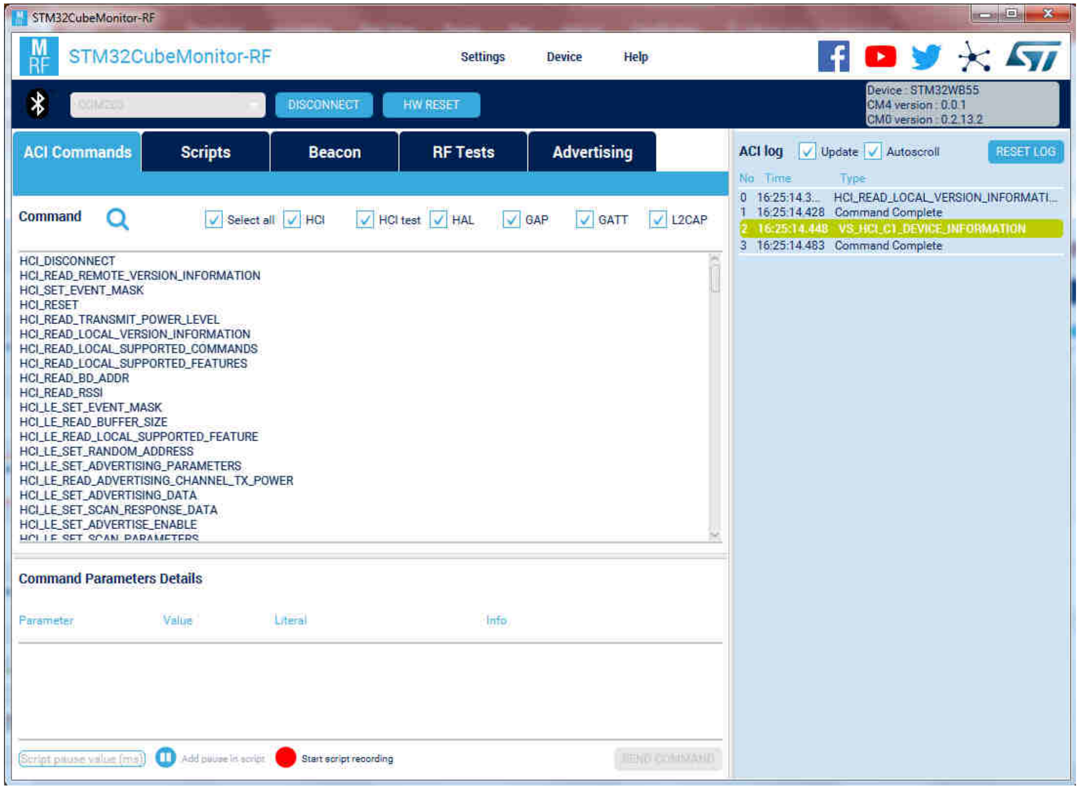
STM32 WB

STM32CubeMX

STM32CubeProgrammer

STM32CubeMonitorRF

CubeWB HAL Firmware



Device : STM32WB55
CM4 version : 0.0.1
CM0 version : 0.2.13.2

```
HCI_DISCONNECT
HCI_READ_REMOTE_VERSION_INFORMATION
HCI_SET_EVENT_MASK
HCI_RESET
HCI_READ_TRANSMIT_POWER_LEVEL
HCI_READ_LOCAL_VERSION_INFORMATION
HCI_READ_LOCAL_SUPPORTED_COMMANDS
HCI_READ_LOCAL_SUPPORTED_FEATURES
HCI_READ_BD_ADDR
HCI_READ_RSSI
HCI_LE_SET_EVENT_MASK
HCI_LE_READ_BUFFER_SIZE
HCI_LE_READ_LOCAL_SUPPORTED_FEATURE
HCI_LE_SET_RANDOM_ADDRESS
HCI_LE_SET_ADVERTISING_PARAMETERS
HCI_LE_READ_ADVERTISING_CHANNEL_TX_POWER
HCI_LE_SET_ADVERTISING_DATA
HCI_LE_SET_SCAN_RESPONSE_DATA
HCI_LE_SET_ADVERTISE_ENABLE
HCI_LE_SET_SCAN_PARAMETERS
```

Command Parameters Details

Parameter	Value	Literal	Info

```
No  Time          Type
0   16:25:14.3... HCI_READ_LOCAL_VERSION_INFORMATI...
1   16:25:14.428  Command Complete
2   16:25:14.448  VS_HCI_C1_DEVICE_INFORMATION
3   16:25:14.483  Command Complete
```

life.augmented

STM32 WB

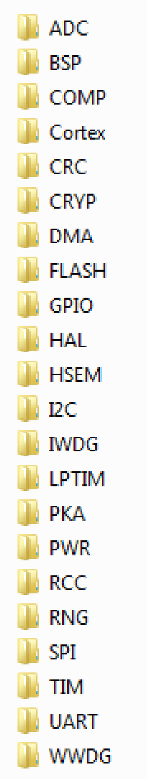
Cube Tools

STM32CubeMX

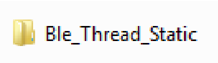
STM32CubeProgrammer

STM32CubeMonitorRF

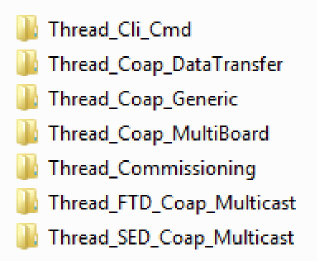
CubeWB HAL Firmware



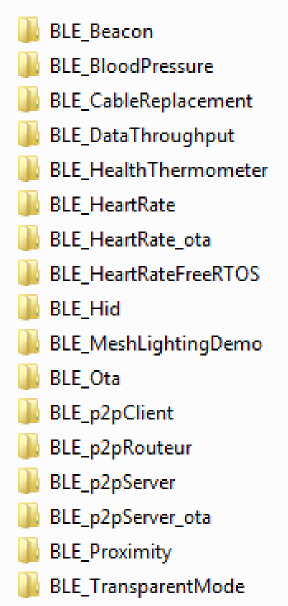
- ADC
- BSP
- COMP
- Cortex
- CRC
- CRYP
- DMA
- FLASH
- GPIO
- HAL
- HSEM
- I2C
- IWDG
- LPTIM
- PKA
- PWR
- RCC
- RNG
- SPI
- TIM
- UART
- WWDG



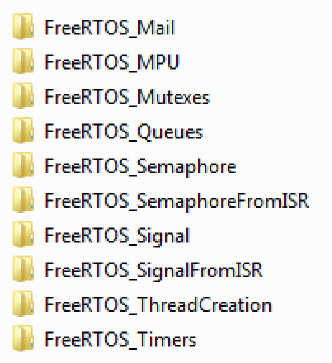
- Ble_Thread_Static



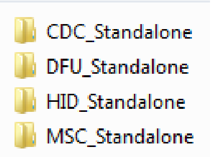
- Thread_Cli_Cmd
- Thread_Coap_DataTransfer
- Thread_Coap_Generic
- Thread_Coap_MultiBoard
- Thread_Commissioning
- Thread_FTD_Coap_Multicast
- Thread_SED_Coap_Multicast



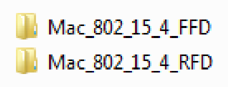
- BLE_Beacon
- BLE_BloodPressure
- BLE_CableReplacement
- BLE_DataThroughput
- BLE_HealthThermometer
- BLE_HeartRate
- BLE_HeartRate_ota
- BLE_HeartRateFreeRTOS
- BLE_Hid
- BLE_MeshLightingDemo
- BLE_Ota
- BLE_p2pClient
- BLE_p2pRouteur
- BLE_p2pServer
- BLE_p2pServer_ota
- BLE_Proximity
- BLE_TransparentMode



- FreeRTOS_Mail
- FreeRTOS_MPU
- FreeRTOS_Mutexes
- FreeRTOS_Queues
- FreeRTOS_Semaphore
- FreeRTOS_SemaphoreFromISR
- FreeRTOS_Signal
- FreeRTOS_SignalFromISR
- FreeRTOS_ThreadCreation
- FreeRTOS_Timers



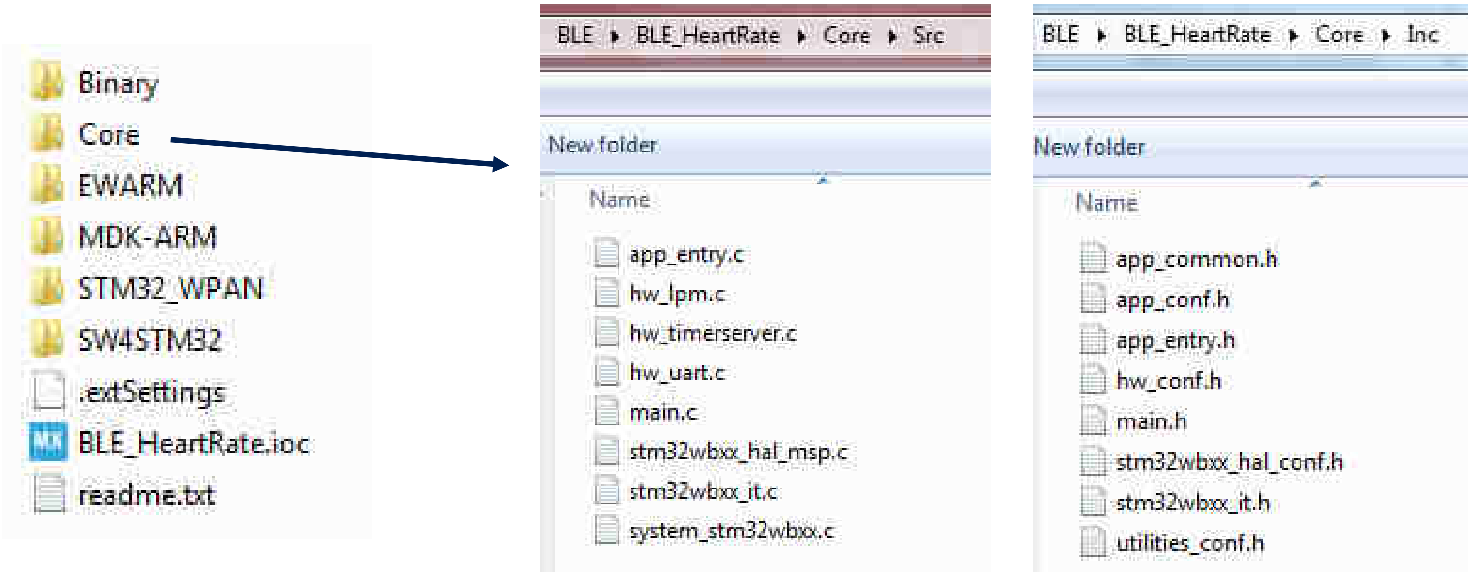
- CDC_Standalone
- DFU_Standalone
- HID_Standalone
- MSC_Standalone



- Mac_802_15_4_FFD
- Mac_802_15_4_RFD

CubeWB firmware

Core folder contains application-related source code



Binary
Core
EWARM
MDK-ARM
STM32_WPAN
SW4STM32
.extSettings
BLE_HeartRate.ioc
readme.txt

BLE ▸ BLE_HeartRate ▸ Core ▸ Src

New folder

Name

app_entry.c
hw_lpm.c
hw_timerserver.c
hw_uart.c
main.c
stm32wbxx_hal_msp.c
stm32wbxx_it.c
system_stm32wbxx.c

BLE ▸ BLE_HeartRate ▸ Core ▸ Inc

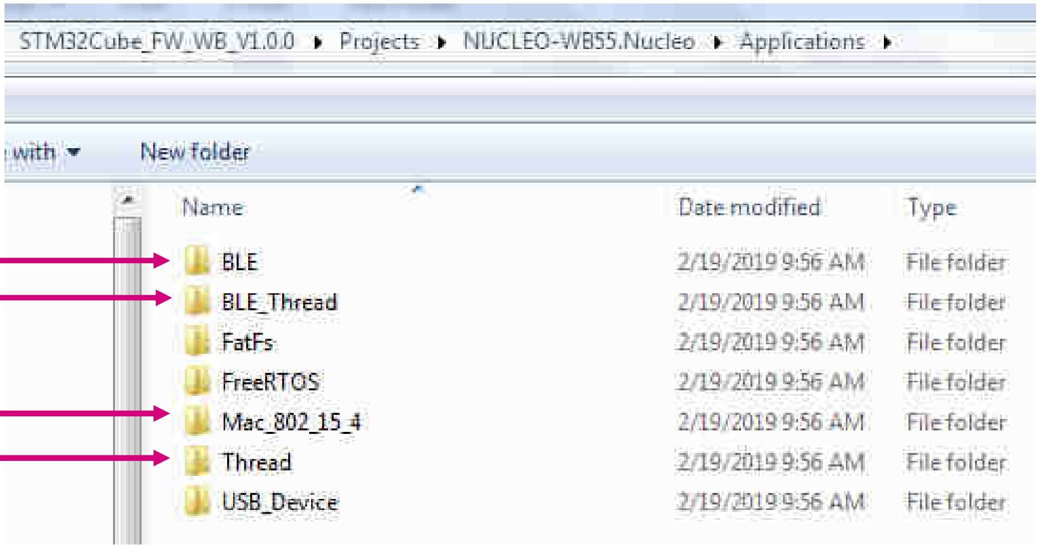
New folder

Name

app_common.h
app_conf.h
app_entry.h
hw_conf.h
main.h
stm32wbxx_hal_conf.h
stm32wbxx_it.h
utilities_conf.h

CubeWB firmware

Different stacks required for different application types



STM32Cube_FW_WB_V1.0.0 ▸ Projects ▸ NUCLEO-WB55.Nucleo ▸ Applications ▸

Name	Date modified	Type
BLE	2/19/2019 9:56 AM	File folder
BLE_Thread	2/19/2019 9:56 AM	File folder
FatFs	2/19/2019 9:56 AM	File folder
FreeRTOS	2/19/2019 9:56 AM	File folder
Mac_802_15_4	2/19/2019 9:56 AM	File folder
Thread	2/19/2019 9:56 AM	File folder
USB_Device	2/19/2019 9:56 AM	File folder

BLE projects → BLE

BLE + Thread Static Concurrent mode project → BLE_Thread

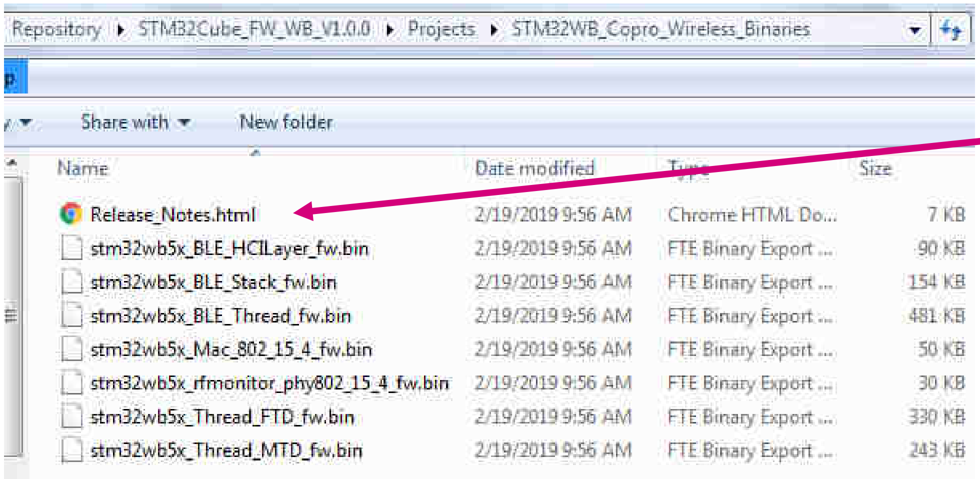
Open MAC project → Mac_802_15_4

Thread projects → Thread

Zigbee 3.0 coming soon!

CubeWB firmware

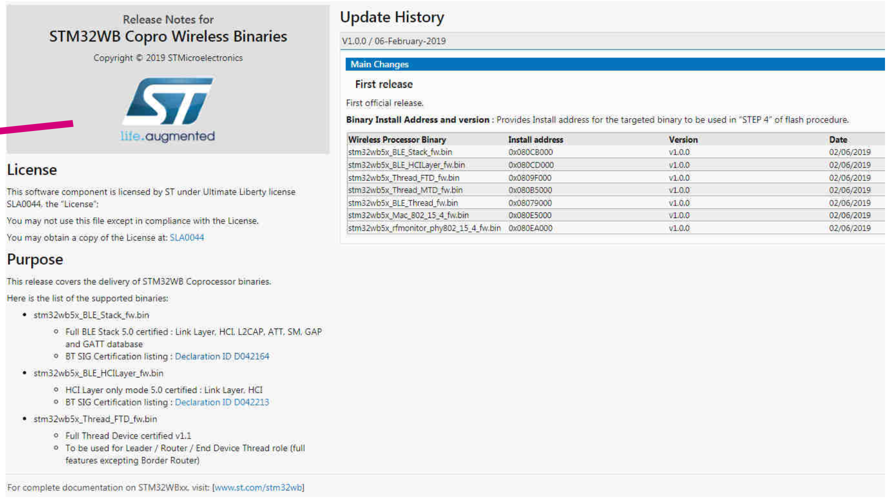
Encrypted radio stack binaries here



Repository ▸ STM32Cube_FW_WB_V1.0.0 ▸ Projects ▸ STM32WB_Copro_Wireless_Binaries

Name	Date modified	Type	Size
Release_Notes.html	2/19/2019 9:56 AM	Chrome HTML Do...	7 KB
stm32wb5x_BLE_HCILayer_fw.bin	2/19/2019 9:56 AM	FTE Binary Export ...	90 KB
stm32wb5x_BLE_Stack_fw.bin	2/19/2019 9:56 AM	FTE Binary Export ...	154 KB
stm32wb5x_BLE_Thread_fw.bin	2/19/2019 9:56 AM	FTE Binary Export ...	481 KB
stm32wb5x_Mac_802_15_4_fw.bin	2/19/2019 9:56 AM	FTE Binary Export ...	50 KB
stm32wb5x_rfmonitor_phy802_15_4_fw.bin	2/19/2019 9:56 AM	FTE Binary Export ...	30 KB
stm32wb5x_Thread_FTD_fw.bin	2/19/2019 9:56 AM	FTE Binary Export ...	330 KB
stm32wb5x_Thread_MTD_fw.bin	2/19/2019 9:56 AM	FTE Binary Export ...	243 KB

HTML file details update procedure



Release Notes for
STM32WB Copro Wireless Binaries

Copyright © 2019 STMicroelectronics

life.augmented

License

This software component is licensed by ST under Ultimate Liberty license SLA0044, the "License";

You may not use this file except in compliance with the License.

You may obtain a copy of the License at: SLA0044

Purpose

This release covers the delivery of STM32WB Coprocessor binaries.

Here is the list of the supported binaries:

- stm32wb5x_BLE_Stack_fw.bin
  - Full BLE Stack 5.0 certified : Link Layer, HCI, L2CAP, ATT, SM, GAP and GATT database
  - BT SIG Certification listing : Declaration ID D042164
- stm32wb5x_BLE_HCILayer_fw.bin
  - HCI Layer only mode 5.0 certified : Link Layer, HCI
  - BT SIG Certification listing : Declaration ID D042213
- stm32wb5x_Thread_FTD_fw.bin
  - Full Thread Device certified v1.1
  - To be used for Leader / Router / End Device Thread role (full features excepting Border Router)

For complete documentation on STM32WBxx, visit: [www.st.com/stm32wb]

Update History

V1.0.0 / 06-February-2019

Main Changes

First release

First official release.

Binary Install Address and version : Provides Install address for the targeted binary to be used in "STEP 4" of flash procedure.

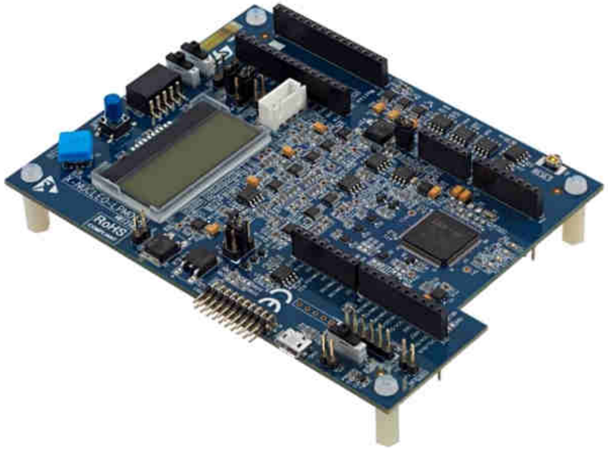
Wireless Processor Binary	Install address	Version	Date
stm32wb5x_BLE_Stack_fw.bin	0x080CB000	v1.0.0	02/06/2019
stm32wb5x_BLE_HCILayer_fw.bin	0x080CD000	v1.0.0	02/06/2019
stm32wb5x_Thread_FTD_fw.bin	0x0809F000	v1.0.0	02/06/2019
stm32wb5x_Thread_MTD_fw.bin	0x080B5000	v1.0.0	02/06/2019
stm32wb5x_BLE_Thread_fw.bin	0x08079000	v1.0.0	02/06/2019
stm32wb5x_Mac_802_15_4_fw.bin	0x080E5000	v1.0.0	02/06/2019
stm32wb5x_rfmonitor_phy802_15_4_fw.bin	0x080EA000	v1.0.0	02/06/2019

Nucleo & Dongle boards come preloaded with the BLE stack

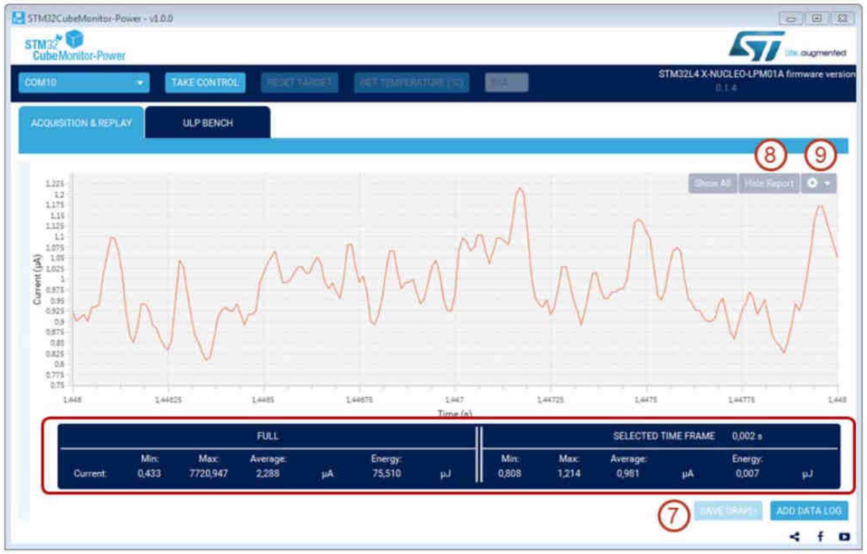
Bonus Cube Tool!

STM32CubeMonitor-Power

$70



X-NUCLEO-LPM01A

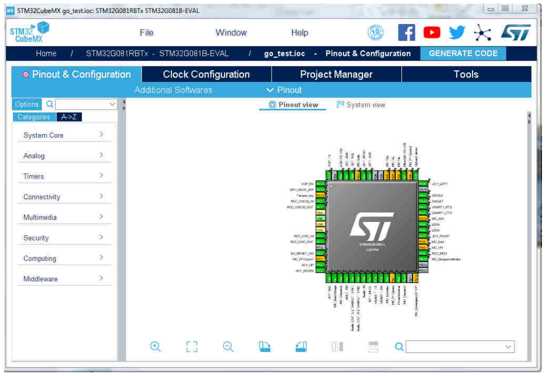


	FULL						SELECTED TIME FRAME 0,002 s					
	Min:	Max:	Average:		Energy:		Min:	Max:	Average:		Energy:	
Current:	0,433	7720,947	2,288	µA	75,510	µJ	0,808	1,214	0,981	µA	0,007	µJ

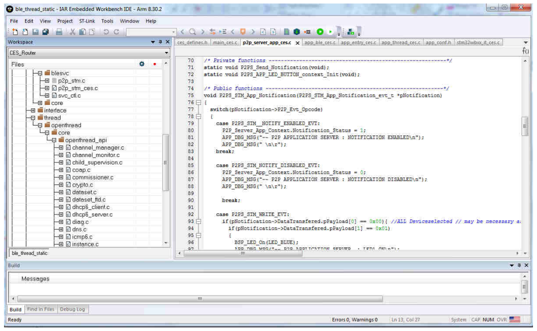
Atollic TrueSTUDIO

Iterative Design Process

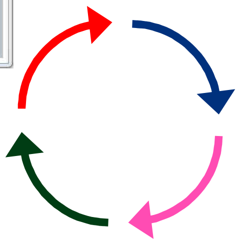
Configure



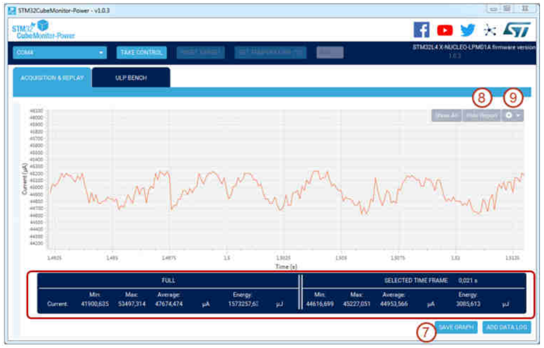
Code & Debug



```
/* Private functions ---------------------------------------------------------*/
static void P2PS_Send_Notification(void);
static void P2PS_APP_LED_BUTTON_context_Init(void);

/* Public functions ----------------------------------------------------------*/
void P2PS_STM_App_Notification(P2PS_STM_App_Notification_evt_t *pNotification)
{
  switch(pNotification->P2P_Evt_Opcode)
  {
    case P2PS_STM__NOTIFY_ENABLED_EVT:
      P2P_Server_App_Context.Notification_Status = 1;
      APP_DBG_MSG("-- P2P APPLICATION SERVER : NOTIFICATION ENABLED\n");
      APP_DBG_MSG(" \n\r");
    break;

    case P2PS_STM_NOTIFY_DISABLED_EVT:
      P2P_Server_App_Context.Notification_Status = 0;
      APP_DBG_MSG("-- P2P APPLICATION SERVER : NOTIFICATION DISABLED\n");
      APP_DBG_MSG(" \n\r");

      break;

    case P2PS_STM_WRITE_EVT:
      if(pNotification->DataTransfered.pPayload[0] == 0x00){ //ALL Deviceselected // may be necessary a
        if(pNotification->DataTransfered.pPayload[1] == 0x01)
        {
          BSP_LED_On(LED_BLUE);
```

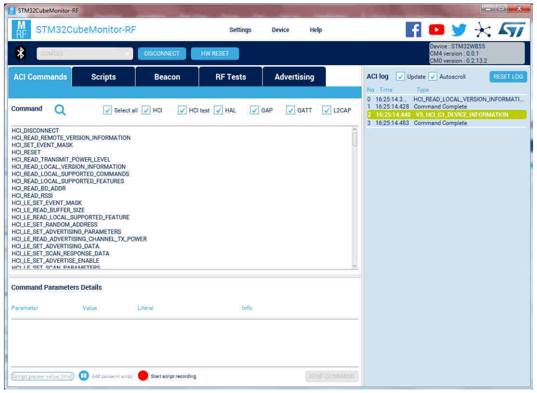


Measure



	FULL						SELECTED TIME FRAME	0,021 s				
	Min:	Max:	Average:		Energy:		Min:	Max:	Average:		Energy:	
Current:	41900,635	53497,314	47674,474	µA	1573257,6	µJ	44616,699	45227,051	44953,566	µA	3085,613	µJ

Test



Device : STM32WB55
CM4 version : 0.0.1
CM0 version : 0.2.13.2

☑ Select all ☑ HCI ☑ HCI test ☑ HAL ☑ GAP ☑ GATT ☑ L2CAP

HCI_DISCONNECT
HCI_READ_REMOTE_VERSION_INFORMATION
HCI_SET_EVENT_MASK
HCI_RESET
HCI_READ_TRANSMIT_POWER_LEVEL
HCI_READ_LOCAL_VERSION_INFORMATION
HCI_READ_LOCAL_SUPPORTED_COMMANDS
HCI_READ_LOCAL_SUPPORTED_FEATURES
HCI_READ_BD_ADDR
HCI_READ_RSSI
HCI_LE_SET_EVENT_MASK
HCI_LE_READ_BUFFER_SIZE
HCI_LE_READ_LOCAL_SUPPORTED_FEATURE
HCI_LE_SET_RANDOM_ADDRESS
HCI_LE_SET_ADVERTISING_PARAMETERS
HCI_LE_READ_ADVERTISING_CHANNEL_TX_POWER
HCI_LE_SET_ADVERTISING_DATA
HCI_LE_SET_SCAN_RESPONSE_DATA
HCI_LE_SET_ADVERTISE_ENABLE
HCI_LE_SET_SCAN_PARAMETERS

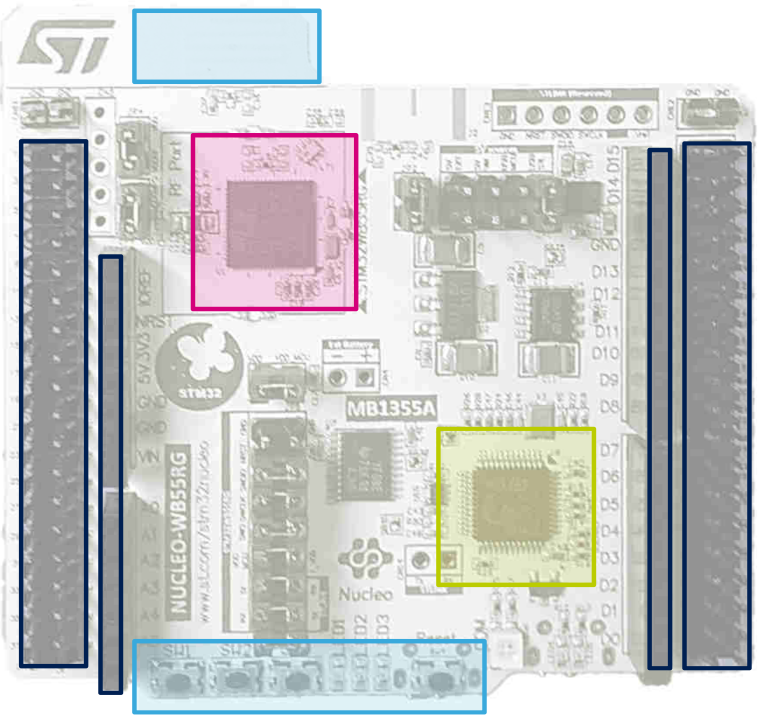
☑ Update ☑ Autoscroll

No	Time	Type
0	16:25:14.3...	HCI_READ_LOCAL_VERSION_INFORMATI...
1	16:25:14.428	Command Complete
2	16:25:14.448	VS_HCI_C1_DEVICE_INFORMATION
3	16:25:14.483	Command Complete

life.augmented

STM32 WB

Nucleo



2.4GHz PCB antenna

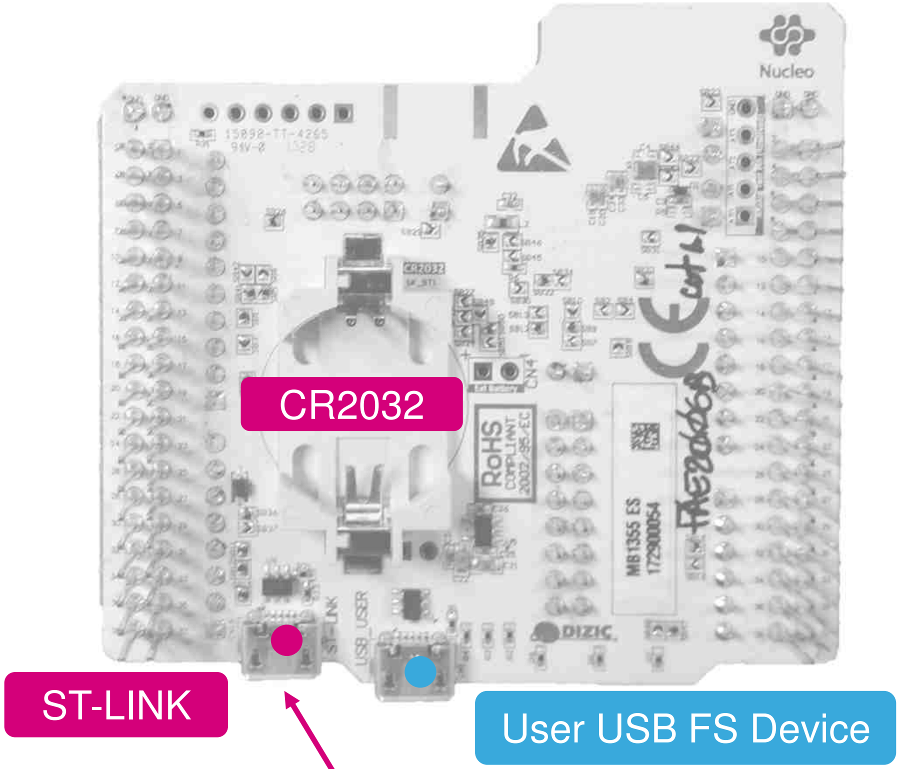
STM32WB55RGV6
(VQFPN68)

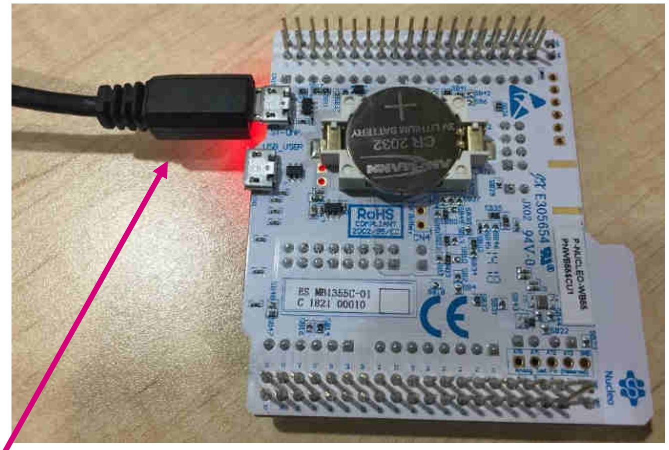
Arduino & Morpho Headers

ST-Link/V2-1

Buttons & LED’s

Nucleo

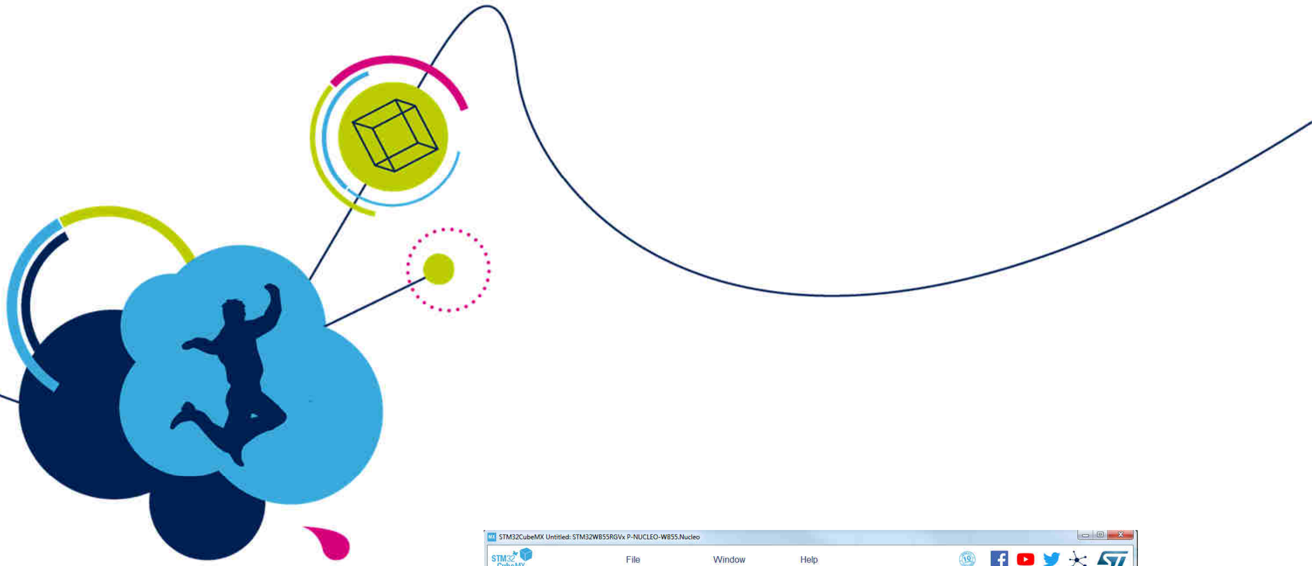




We will use this one!

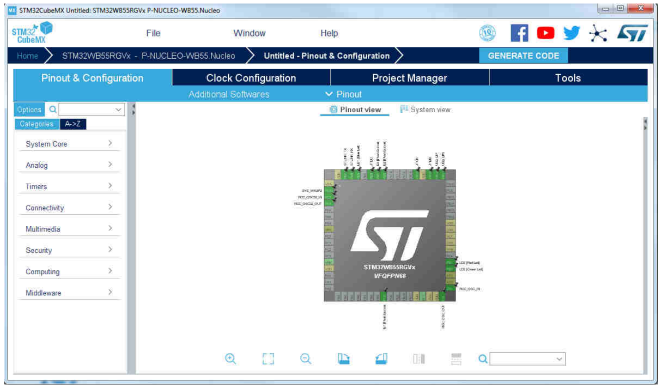
Your Magic number!

STM32WB55 Nucleo Pack
Wireless ultra-low-power
01
STM32WB55 FEATURES
Arm® Cortex®-M4 MCU with 64 MHz/80 DMIPS and dedicated Arm® Cortex®-M0+ radio co-processor at 32 MHz
Multiprotocol: BLE 5.0, 802.15.4, concurrent mode
1 Mbyte of Flash memory
256 Kbytes of SRAM
High RF performances : RX -96 dBm/ -100 dBm; TX +6 dBm
STM32 Nucleo
Download on the App Store
GET IT ON Google Play

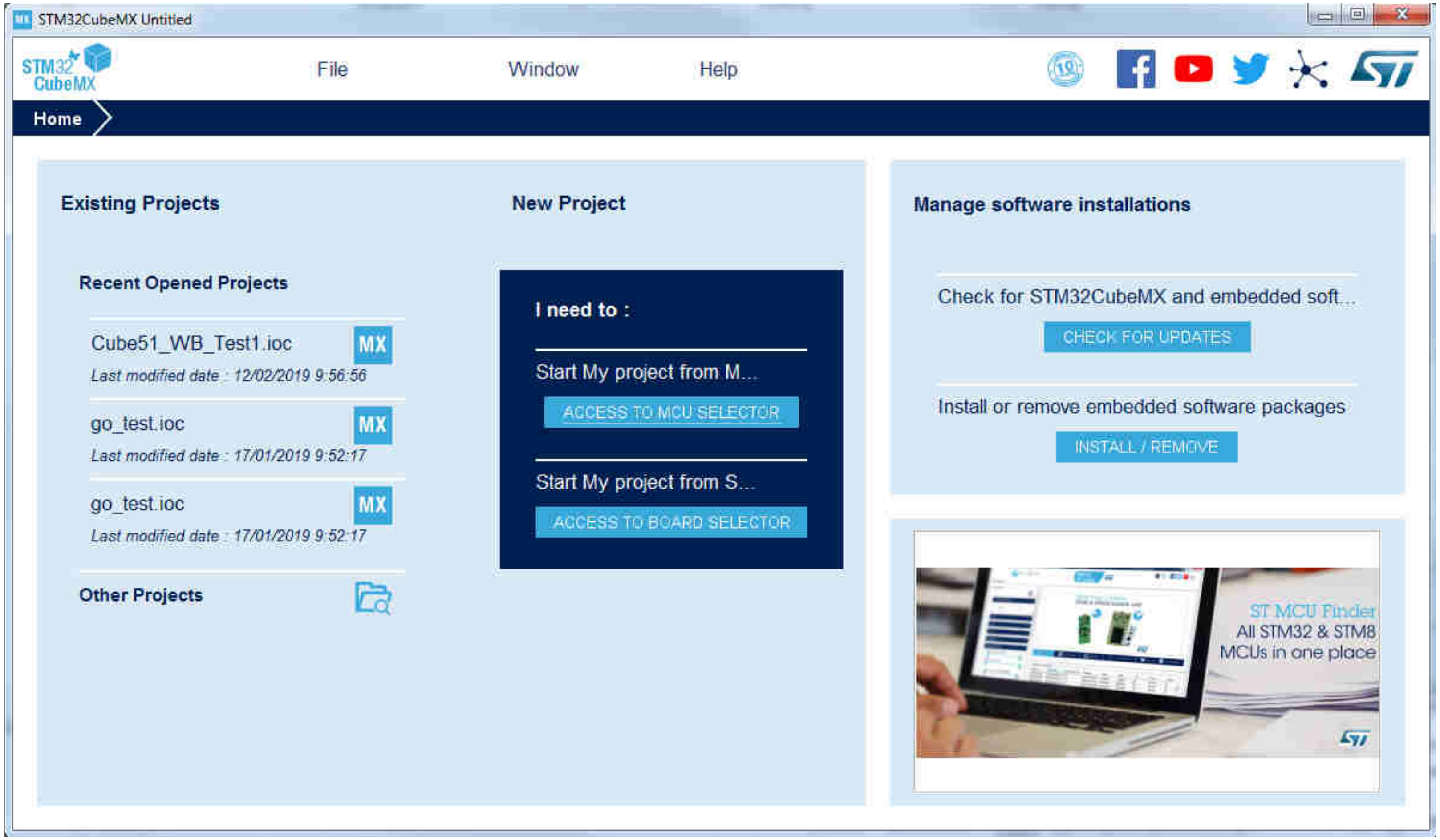


Hands-On

CubeMX



Launch CubeMX



STM32CubeMX Untitled

Existing Projects

Recent Opened Projects

Cube51_WB_Test1.ioc
Last modified date : 12/02/2019 9:56:56

go_test.ioc
Last modified date : 17/01/2019 9:52:17

go_test.ioc
Last modified date : 17/01/2019 9:52:17

Other Projects

New Project

I need to :

Start My project from M...
ACCESS TO MCU SELECTOR

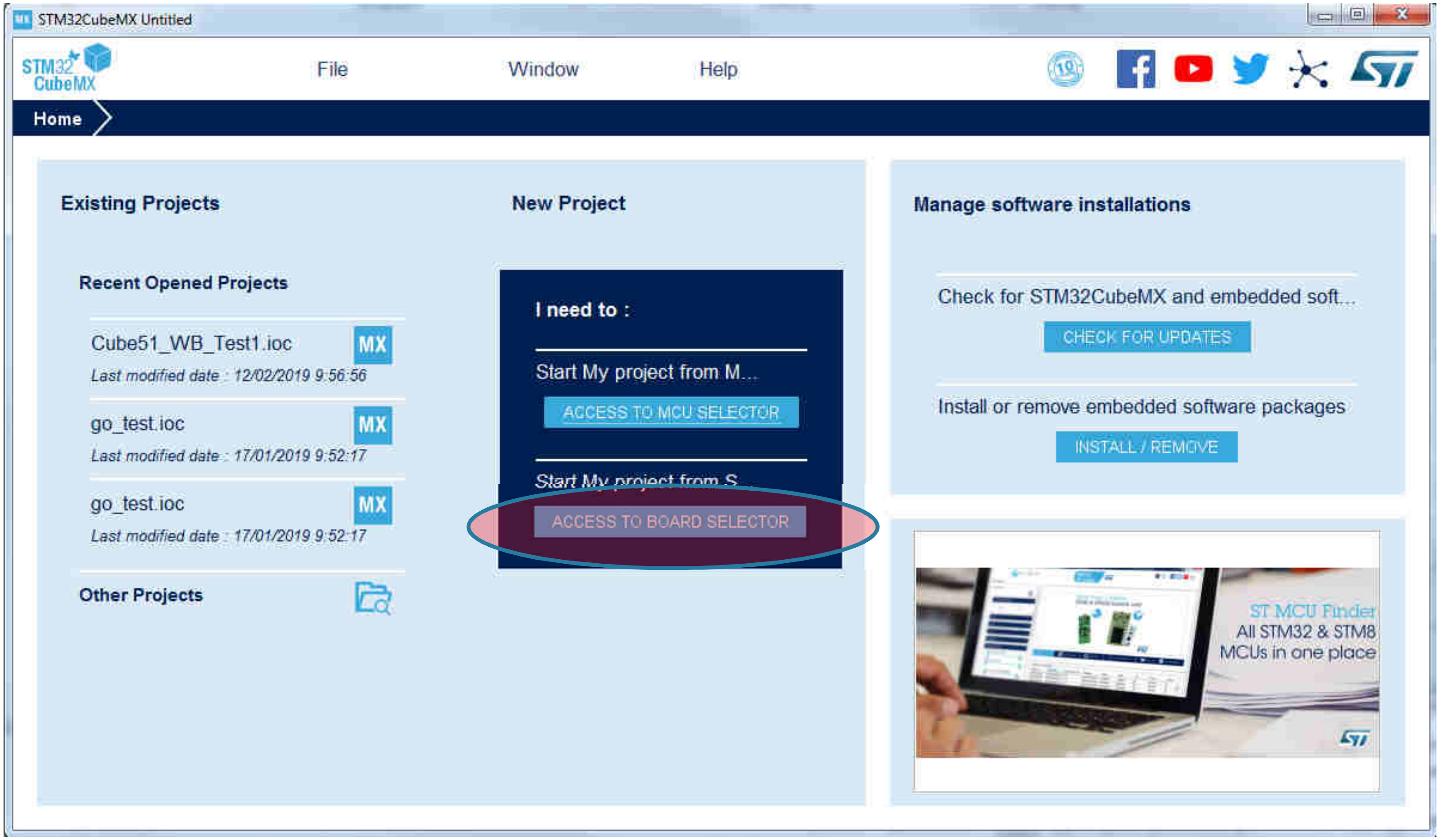
Start My project from S...
ACCESS TO BOARD SELECTOR

Manage software installations

Check for STM32CubeMX and embedded soft...
CHECK FOR UPDATES

Install or remove embedded software packages
INSTALL / REMOVE

Start project from Board Selector



STM32CubeMX Untitled

Existing Projects

Recent Opened Projects

Cube51_WB_Test1.ioc
Last modified date : 12/02/2019 9:56:56

go_test.ioc
Last modified date : 17/01/2019 9:52:17

go_test.ioc
Last modified date : 17/01/2019 9:52:17

Other Projects

New Project

I need to :

Start My project from M...

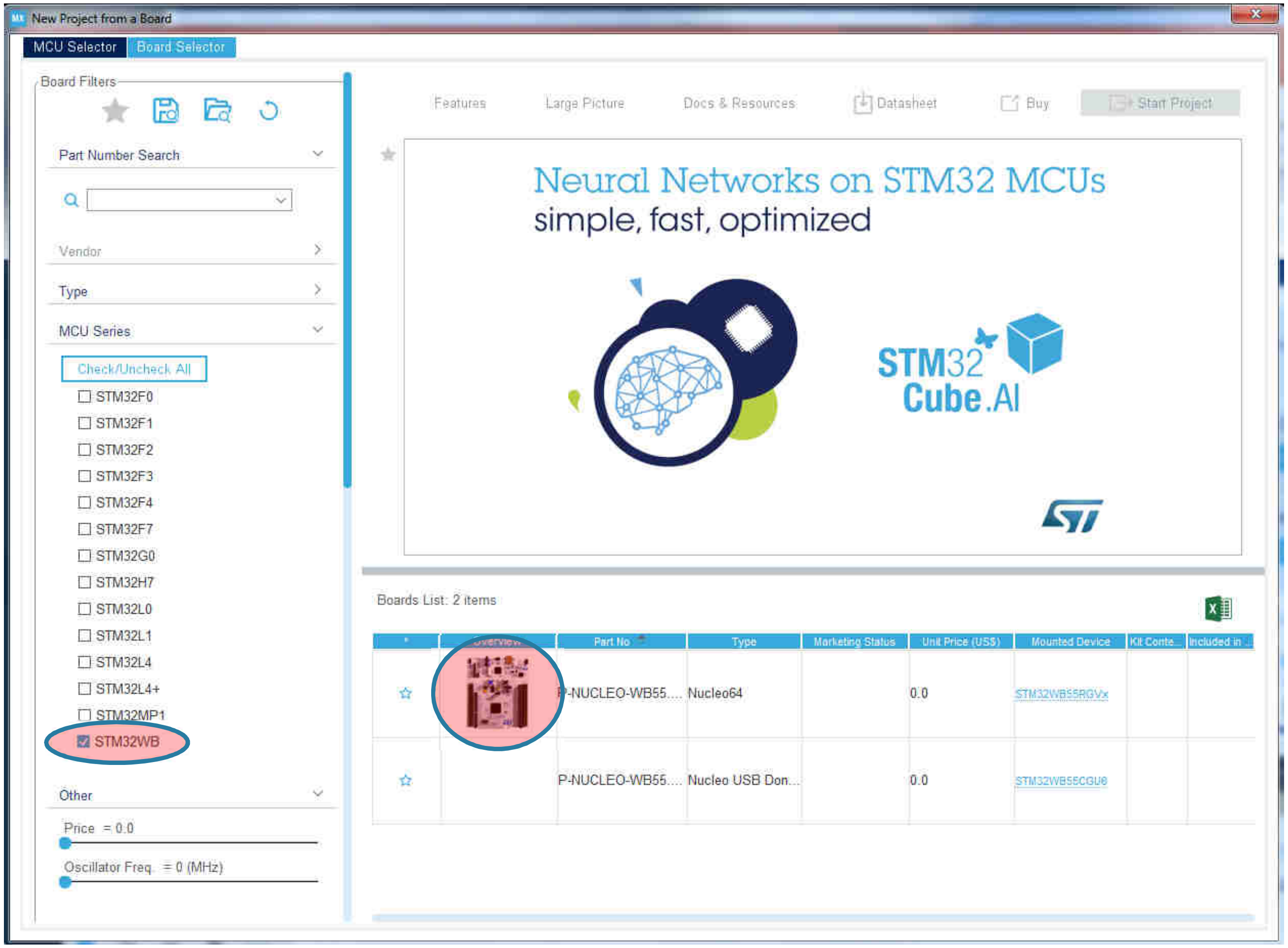
Start My project from S...

Manage software installations

Check for STM32CubeMX and embedded soft...

Install or remove embedded software packages

Filter by STM32WB and double-click on the Nucleo board!



New Project from a Board

MCU Selector | Board Selector

Board Filters

Part Number Search

Vendor

Type

MCU Series

Check/Uncheck All

- ☐ STM32F0
- ☐ STM32F1
- ☐ STM32F2
- ☐ STM32F3
- ☐ STM32F4
- ☐ STM32F7
- ☐ STM32G0
- ☐ STM32H7
- ☐ STM32L0
- ☐ STM32L1
- ☐ STM32L4
- ☐ STM32L4+
- ☐ STM32MP1
- ☑ STM32WB

Other

Price = 0.0

Oscillator Freq. = 0 (MHz)

Features | Large Picture | Docs & Resources | Datasheet | Buy | Start Project

Neural Networks on STM32 MCUs
simple, fast, optimized

STM32 Cube.AI

Boards List: 2 items

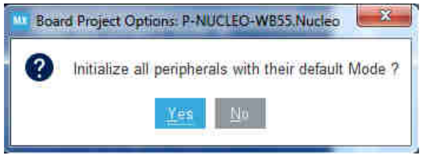
*	Overview	Part No	Type	Marketing Status	Unit Price (US$)	Mounted Device	Kit Conte...	Included in ...
☆		P-NUCLEO-WB55....	Nucleo64		0.0	STM32WB55RGVx		
☆		P-NUCLEO-WB55....	Nucleo USB Don...		0.0	STM32WB55CGU6		

life.augmented

STM32 WB

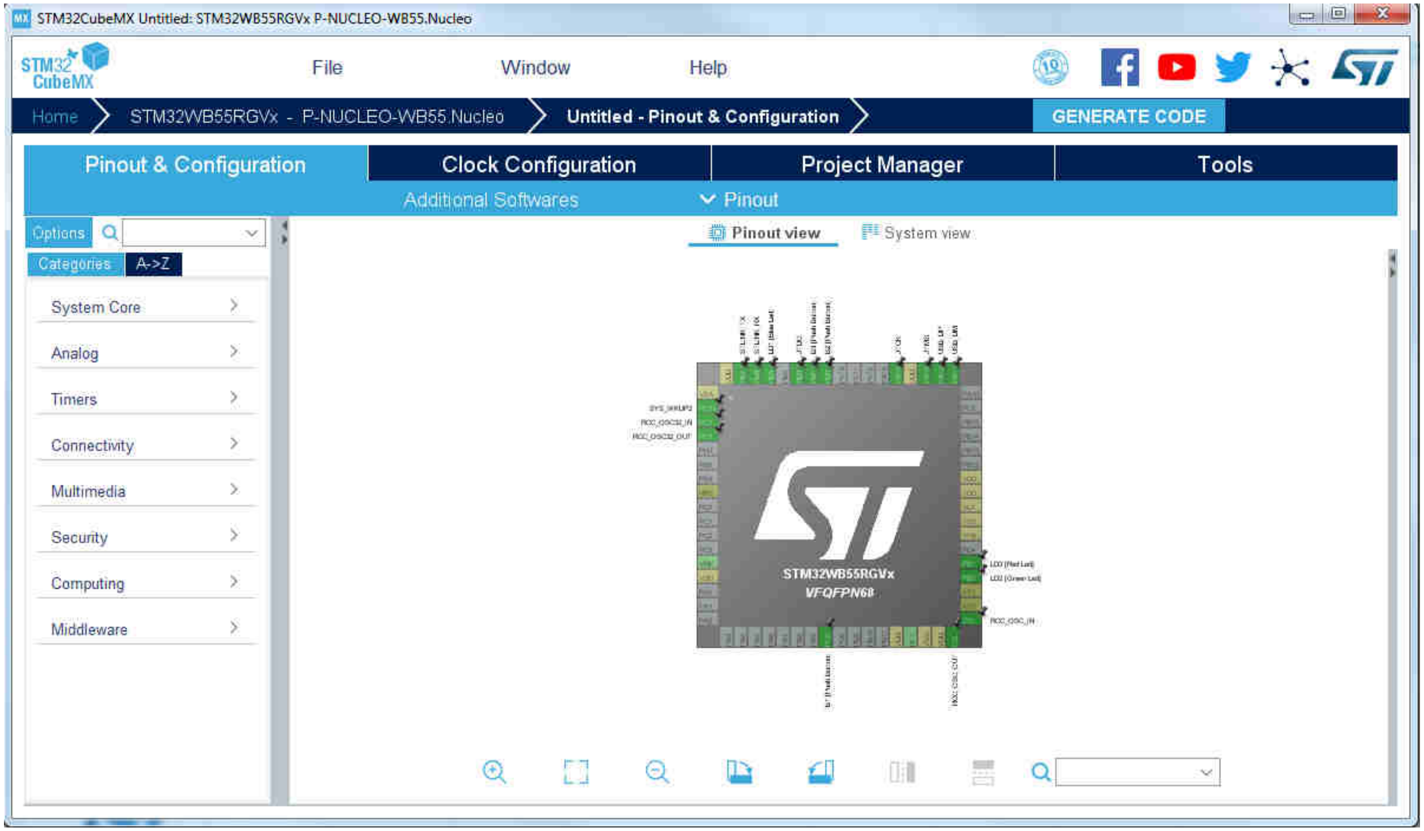
P-NUCLEO-WB55 Board Project

Yes



Board Project Options: P-NUCLEO-WB55.Nucleo

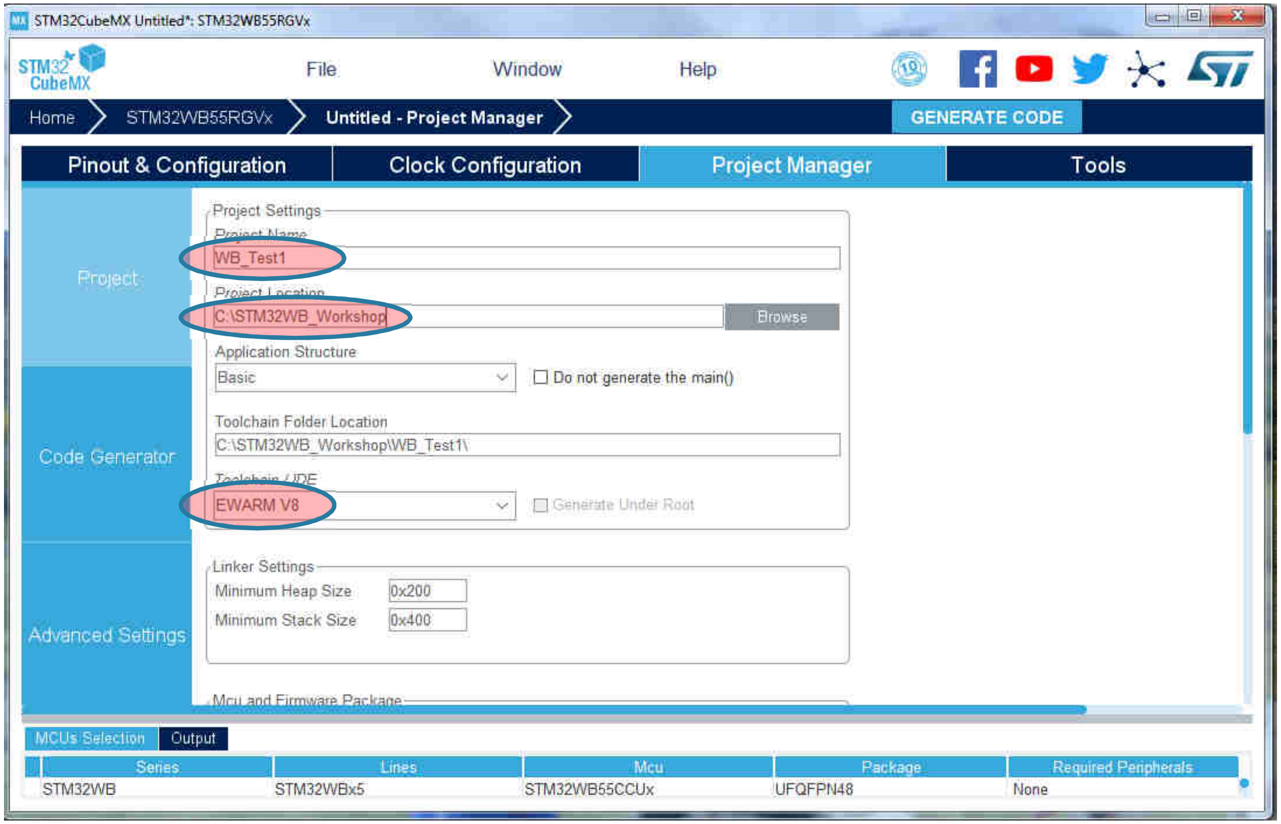
Initialize all peripherals with their default Mode ?



STM32CubeMX Untitled: STM32WB55RGVx P-NUCLEO-WB55.Nucleo

P-NUCLEO-WB55 Board Project

- Name your project
- Recommended Project location: C:\STM32WB_Workshop\
- Use EWARM V8 toolchain



Project Settings

Project Name: WB_Test1

Project Location: C:\STM32WB_Workshop

Application Structure: Basic

□ Do not generate the main()

Toolchain Folder Location: C:\STM32WB_Workshop\WB_Test1\

Toolchain / IDE: EWARM V8

□ Generate Under Root

Linker Settings

Minimum Heap Size: 0x200

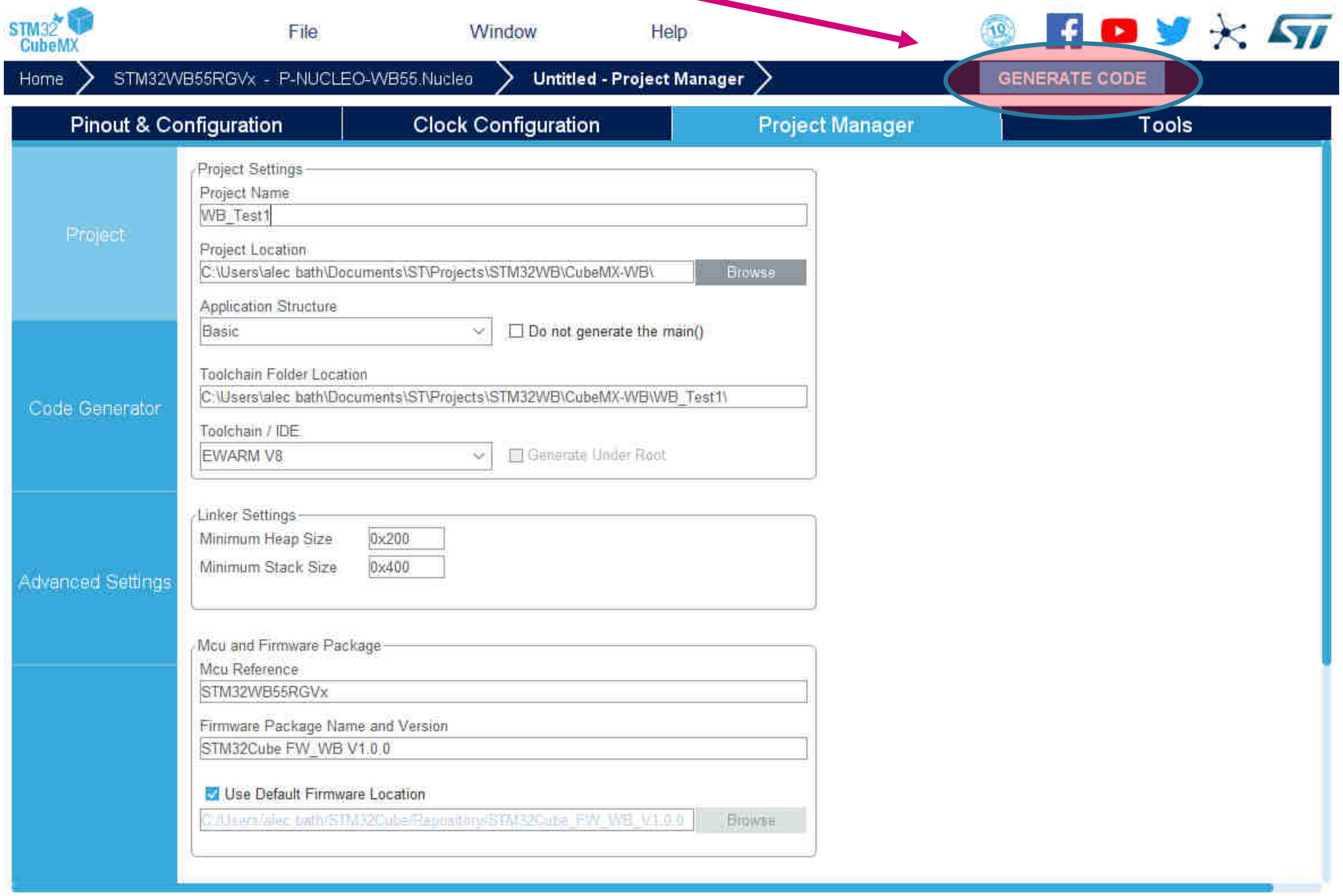
Minimum Stack Size: 0x400

Mcu and Firmware Package

Series	Lines	Mcu	Package	Required Peripherals
STM32WB	STM32WBx5	STM32WB55CCUx	UFQFPN48	None

P-NUCLEO-WB55 Board Project

GENERATE CODE



Project Settings

Project Name: WB_Test1

Project Location: C:\Users\alec bath\Documents\ST\Projects\STM32WB\CubeMX-WB\ Browse

Application Structure: Basic □ Do not generate the main()

Toolchain Folder Location: C:\Users\alec bath\Documents\ST\Projects\STM32WB\CubeMX-WB\WB_Test1\

Toolchain / IDE: EWARM V8 □ Generate Under Root

Linker Settings

Minimum Heap Size: 0x200

Minimum Stack Size: 0x400

Mcu and Firmware Package

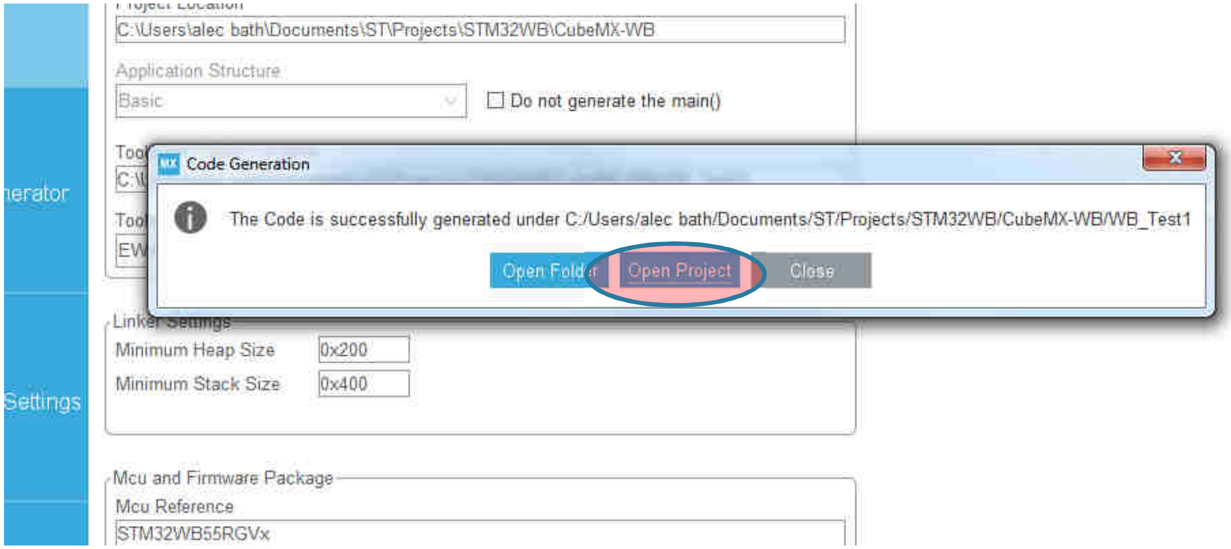
Mcu Reference: STM32WB55RGVx

Firmware Package Name and Version: STM32Cube FW_WB V1.0.0

☑ Use Default Firmware Location

C:/Users/alec bath/STM32Cube/Repository/STM32Cube_FW_WB_V1.0.0 Browse

Open Project



Project Location

C:\Users\alec bath\Documents\ST\Projects\STM32WB\CubeMX-WB

Application Structure

Basic

☐ Do not generate the main()

Code Generation

The Code is successfully generated under C:/Users/alec bath/Documents/ST/Projects/STM32WB/CubeMX-WB/WB_Test1

Linker Settings

Minimum Heap Size 0x200

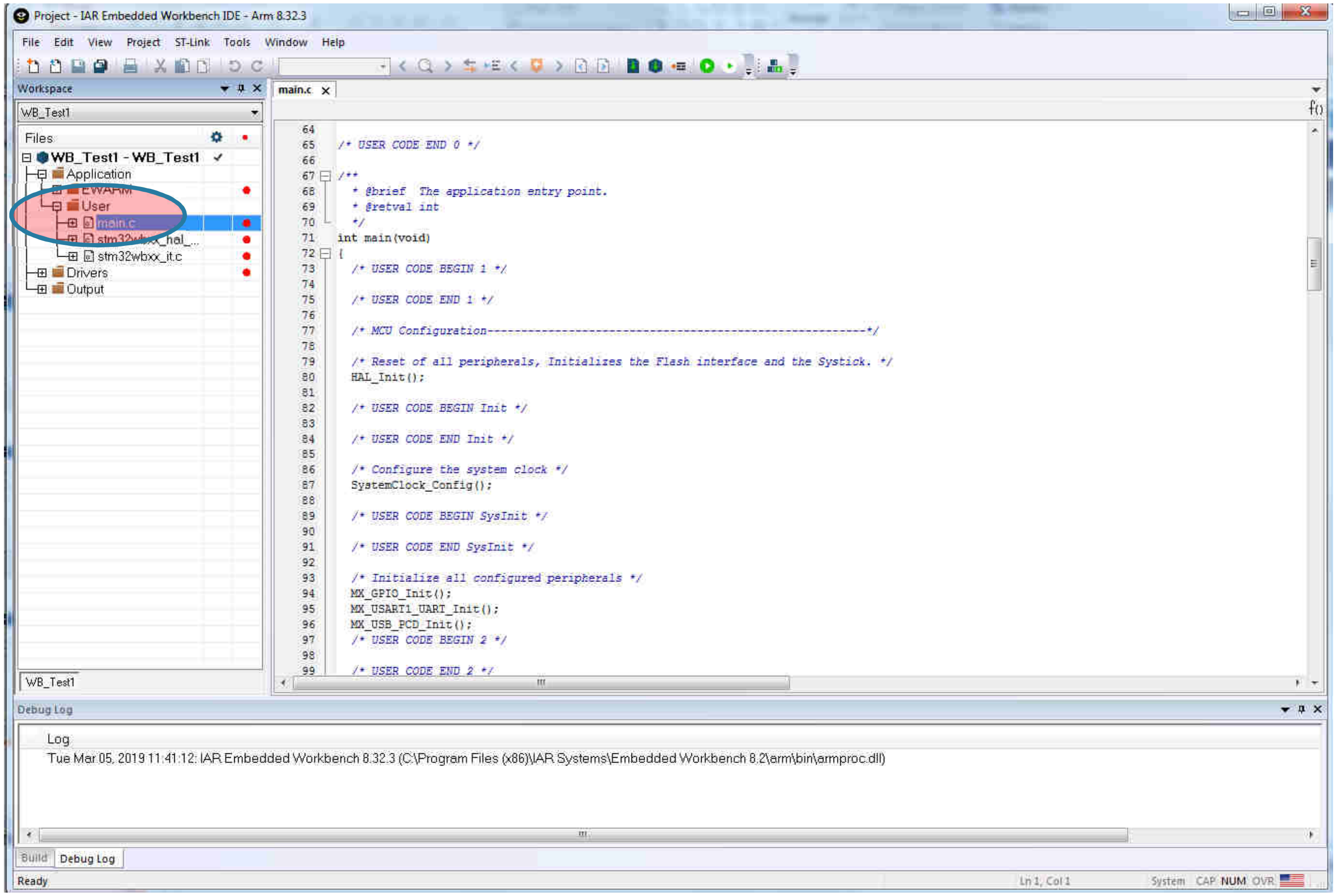
Minimum Stack Size 0x400

Mcu and Firmware Package

Mcu Reference

STM32WB55RGVx

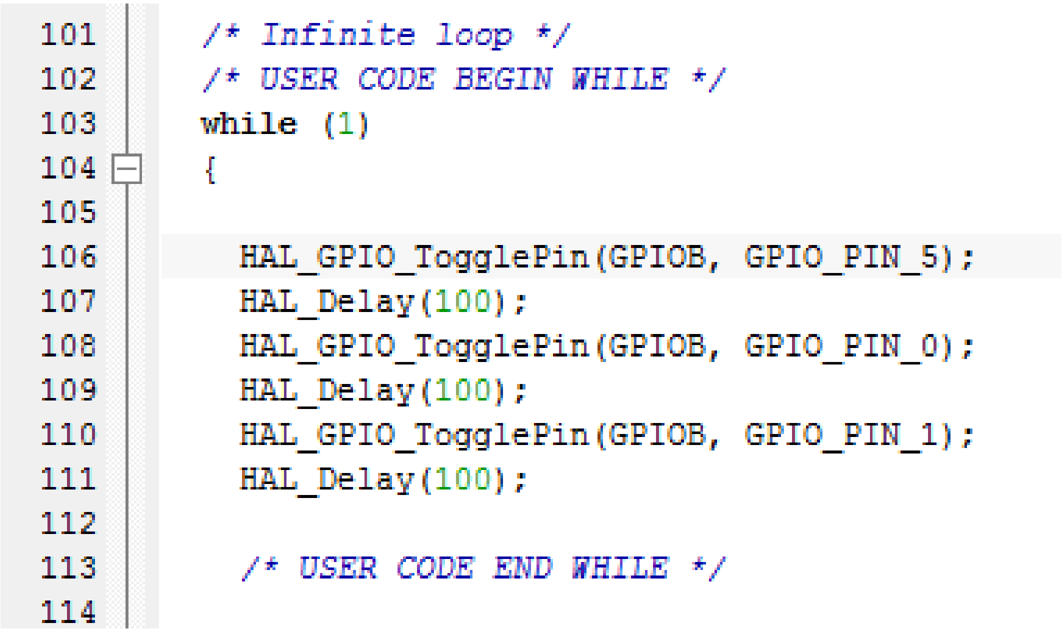
Expand the **User** file tree and Open **main.c**



```
/* USER CODE END 0 */

/**
  * @brief  The application entry point.
  * @retval int
  */
int main(void)
{
  /* USER CODE BEGIN 1 */

  /* USER CODE END 1 */

  /* MCU Configuration--------------------------------------------------------*/

  /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
  HAL_Init();

  /* USER CODE BEGIN Init */

  /* USER CODE END Init */

  /* Configure the system clock */
  SystemClock_Config();

  /* USER CODE BEGIN SysInit */

  /* USER CODE END SysInit */

  /* Initialize all configured peripherals */
  MX_GPIO_Init();
  MX_USART1_UART_Init();
  MX_USB_PCD_Init();
  /* USER CODE BEGIN 2 */

  /* USER CODE END 2 */
```

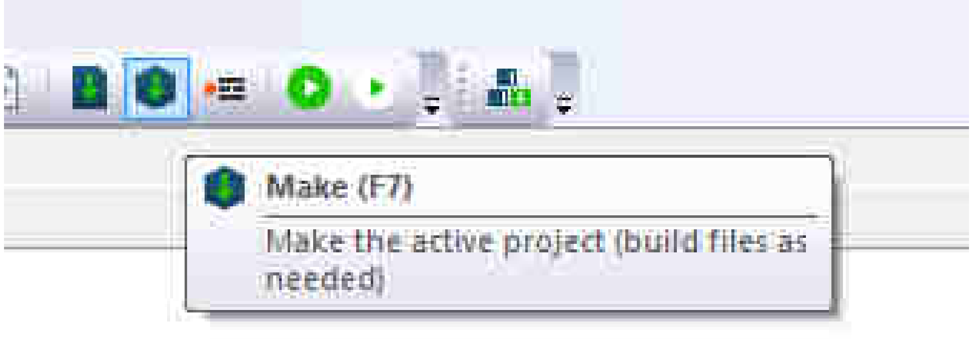
```
Tue Mar 05, 2019 11:41:12: IAR Embedded Workbench 8.32.3 (C:\Program Files (x86)\IAR Systems\Embedded Workbench 8.2\arm\bin\armproc.dll)
```

Add some code to while(1) loop:



```
/* Infinite loop */
/* USER CODE BEGIN WHILE */
while (1)
{

  HAL_GPIO_TogglePin(GPIOB, GPIO_PIN_5);
  HAL_Delay(100);
  HAL_GPIO_TogglePin(GPIOB, GPIO_PIN_0);
  HAL_Delay(100);
  HAL_GPIO_TogglePin(GPIOB, GPIO_PIN_1);
  HAL_Delay(100);

  /* USER CODE END WHILE */

```

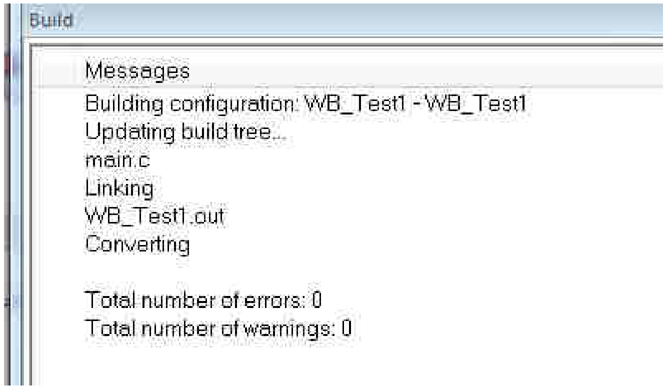
Build the project



Make (F7)

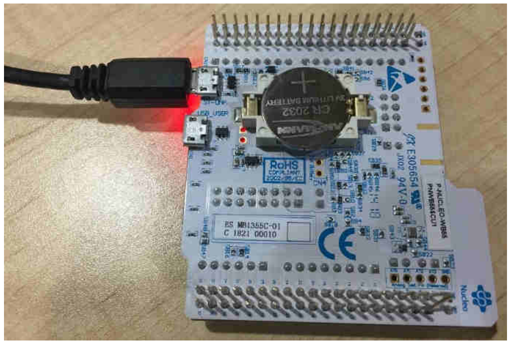
Make the active project (build files as needed)

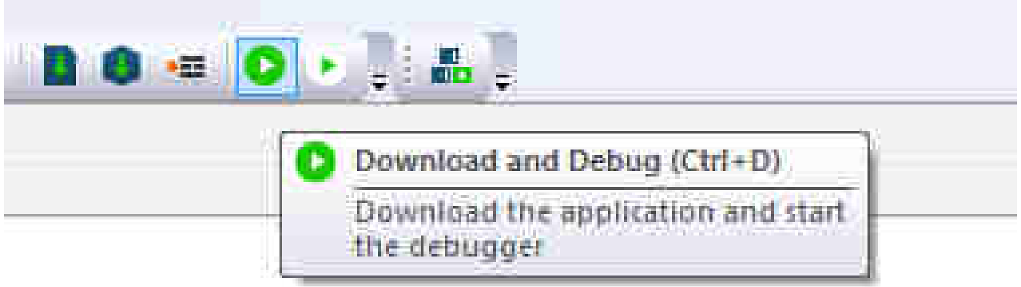
Check for errors



```
Build
Messages
Building configuration: WB_Test1 - WB_Test1
Updating build tree...
main.c
Linking
WB_Test1.out
Converting

Total number of errors: 0
Total number of warnings: 0
```

Download & Debug (attach your board) ☺

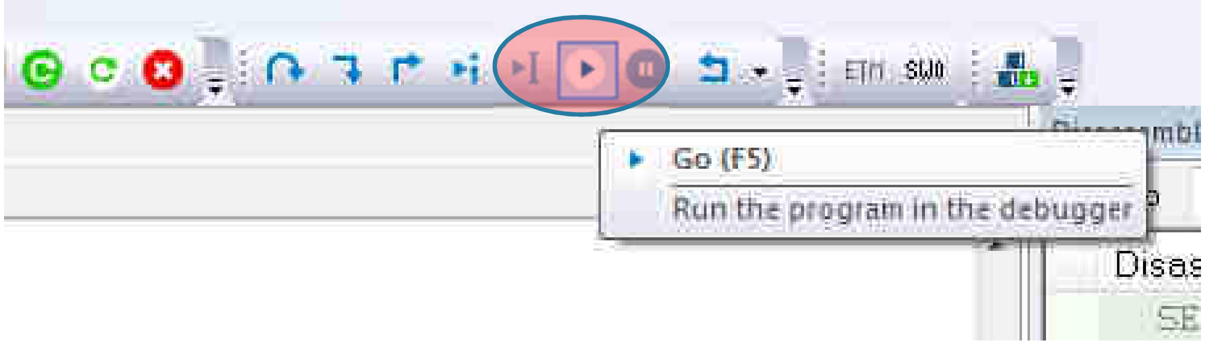




Download and Debug (Ctrl+D)

Download the application and start the debugger

GO!

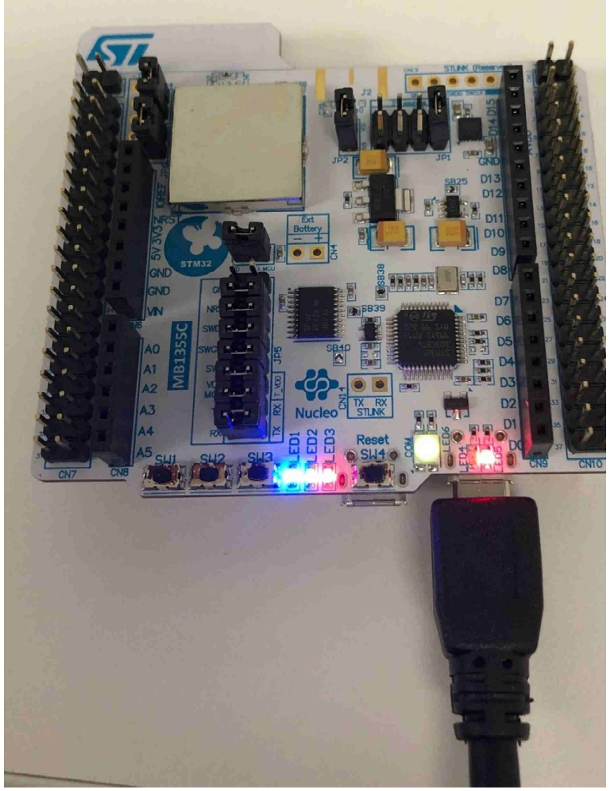


Go (F5)

Run the program in the debugger

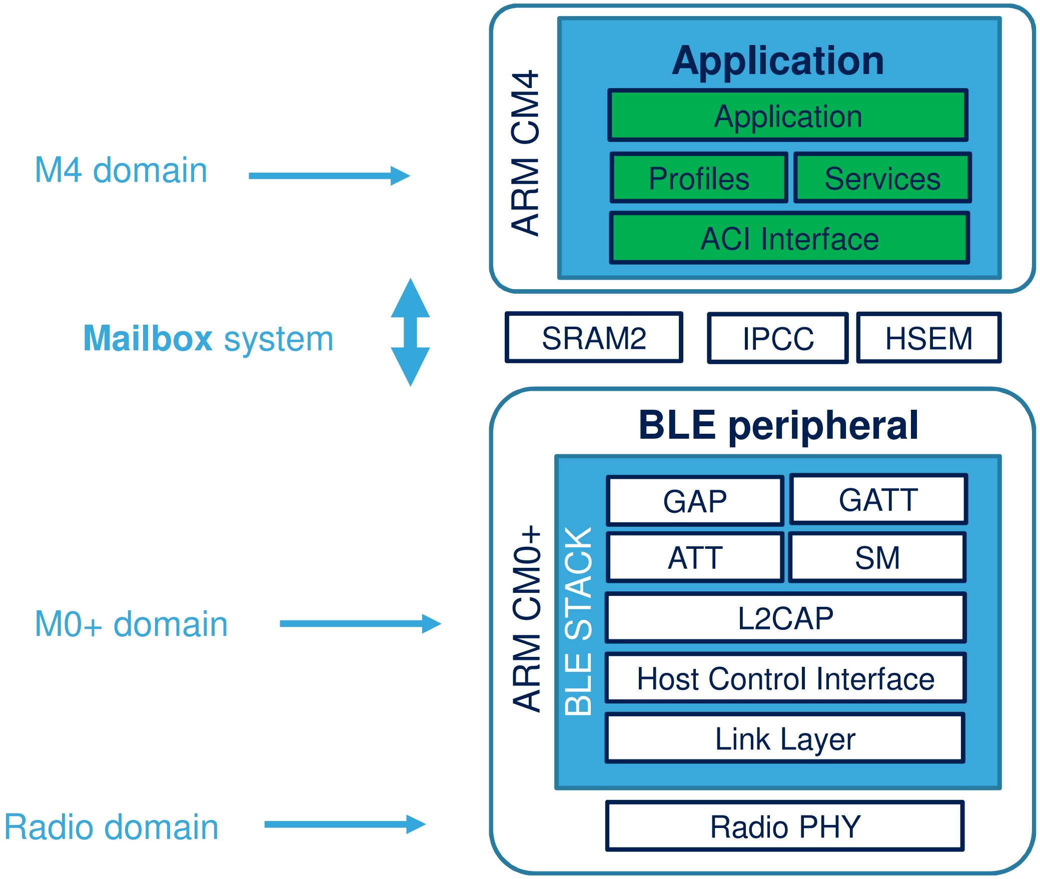
P-NUCLEO-WB55 Board Project

Enjoy the dancing LED’s! ☺

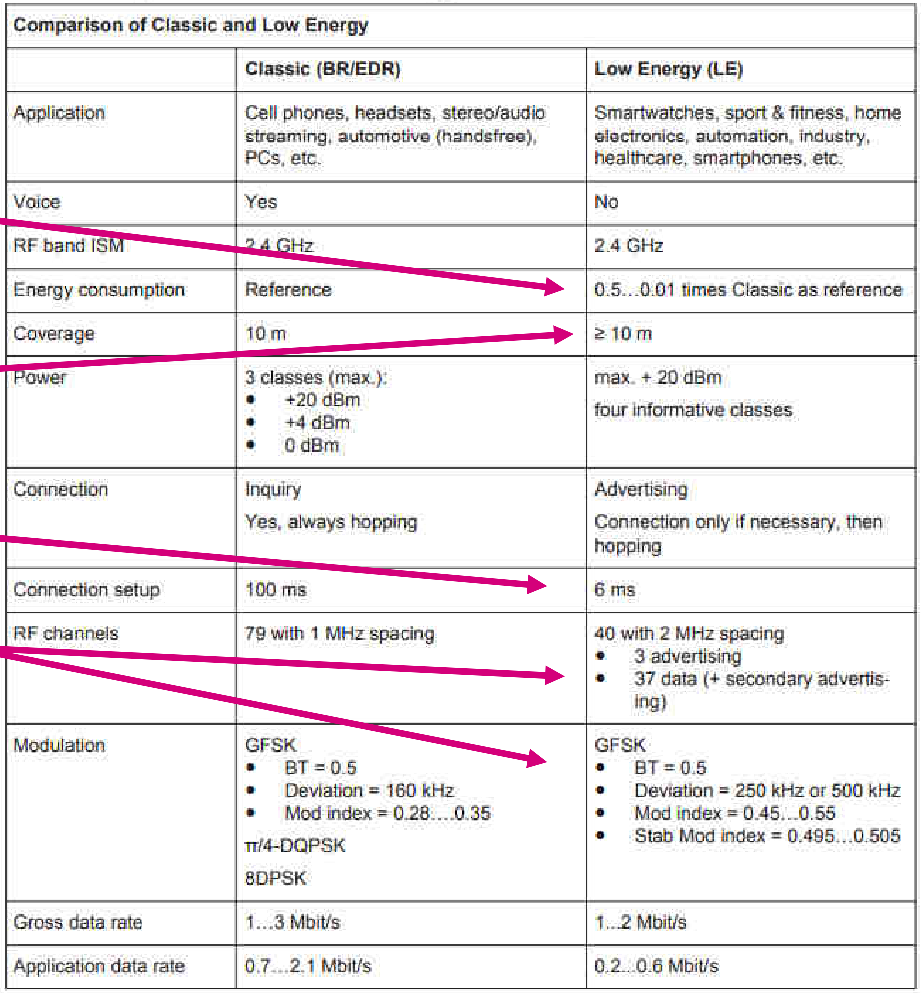


BLE Fundamentals

BLE Protocol Stack layers



Bluetooth Classic (BR/EDR) vs Low Energy (LE)



Comparison of Classic and Low Energy

	Classic (BR/EDR)	**Low Energy (LE)**
Application	Cell phones, headsets, stereo/audio streaming, automotive (handsfree), PCs, etc.	Smartwatches, sport & fitness, home electronics, automation, industry, healthcare, smartphones, etc.
Voice	Yes	No
RF band ISM	2.4 GHz	2.4 GHz
Energy consumption	Reference	0.5…0.01 times Classic as reference
Coverage	10 m	≥ 10 m
Power	3 classes (max.): • +20 dBm • +4 dBm • 0 dBm	max. + 20 dBm four informative classes
Connection	Inquiry Yes, always hopping	Advertising Connection only if necessary, then hopping
Connection setup	100 ms	6 ms
RF channels	79 with 1 MHz spacing	40 with 2 MHz spacing • 3 advertising • 37 data (+ secondary advertising)
Modulation	GFSK • BT = 0.5 • Deviation = 160 kHz • Mod index = 0.28….0.35 π/4-DQPSK 8DPSK	GFSK • BT = 0.5 • Deviation = 250 kHz or 500 kHz • Mod index = 0.45…0.55 • Stab Mod index = 0.495…0.505
Gross data rate	1…3 Mbit/s	1…2 Mbit/s
Application data rate	0.7…2.1 Mbit/s	0.2…0.6 Mbit/s

100X lower

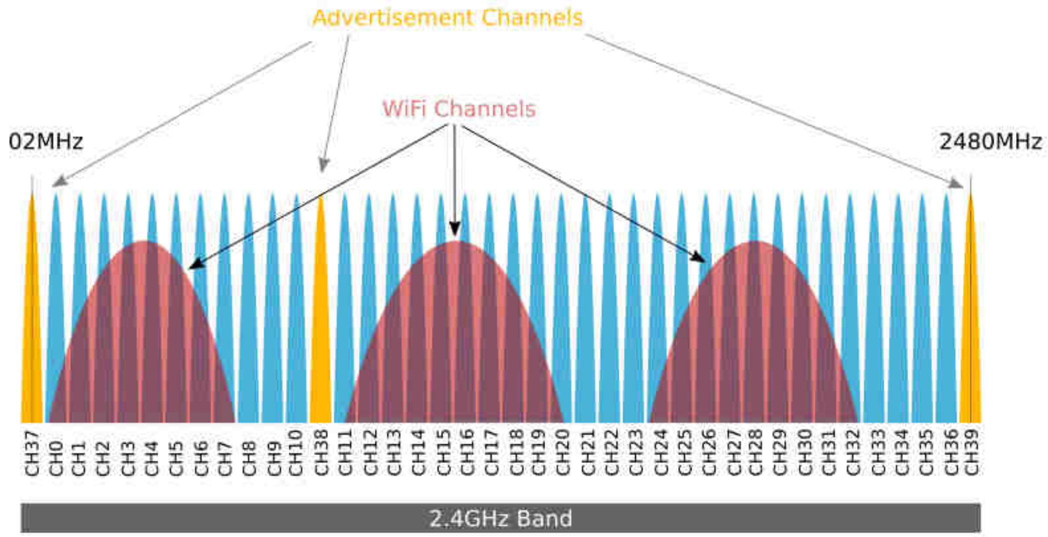
Longer range

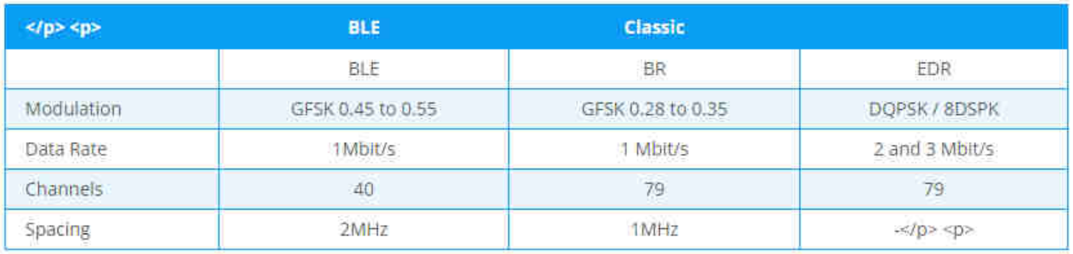
Fast connection (only 3 advertising channels to scan)

Relaxed RF requirements (lower cost silicon / passives)

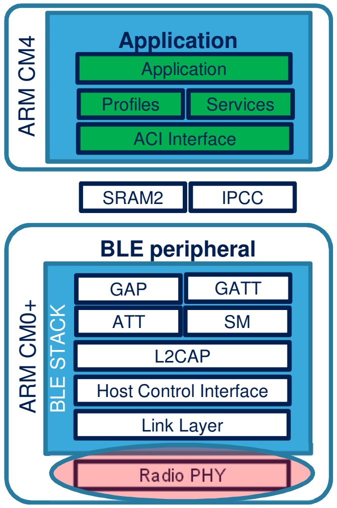
Strategically placed advertising channels

Remaining 37 channels are data channels



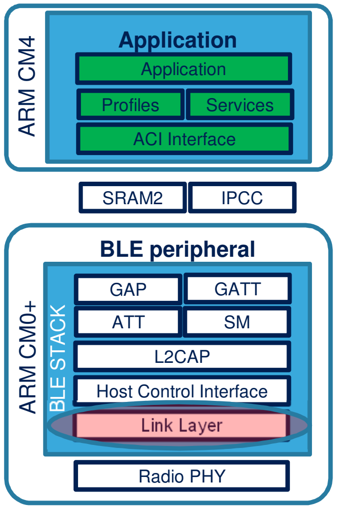


</p> <p>	BLE	Classic	
	BLE	BR	EDR
Modulation	GFSK 0.45 to 0.55	GFSK 0.28 to 0.35	DQPSK / 8DSPK
Data Rate	1Mbit/s	1 Mbit/s	2 and 3 Mbit/s
Channels	40	79	79
Spacing	2MHz	1MHz	-</p> <p>



Link Layer State Machine

- **Standby** state: Sleep, Stop, Standby
- **Advertising** is the key to initiating all BLE communications!
- An **Initiator** and **Advertiser** negotiate a **Connection**
- In a Connection
  - The Link-Layer ***Master*** is also the GAP **Central**
  - The Link-Layer ***Slave*** is also the GAP **Peripheral**



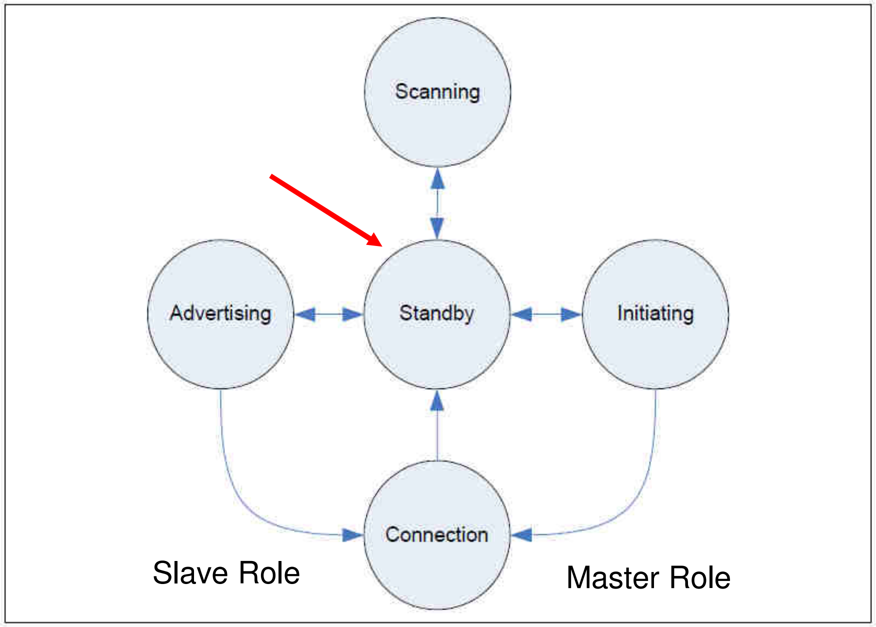
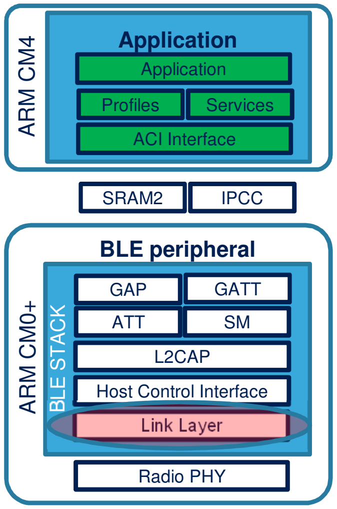


Figure 1.1: State diagram of the Link Layer state machine

Link Layer State Machine

- **Standby** state: Sleep, Stop, Standby
- **Advertising** is the key to initiating all BLE communications!
- An **Initiator** and **Advertiser** negotiate a **Connection**
- In a Connection
  - The Link-Layer ***Master*** is also the GAP **Central**
  - The Link-Layer ***Slave*** is also the GAP **Peripheral**



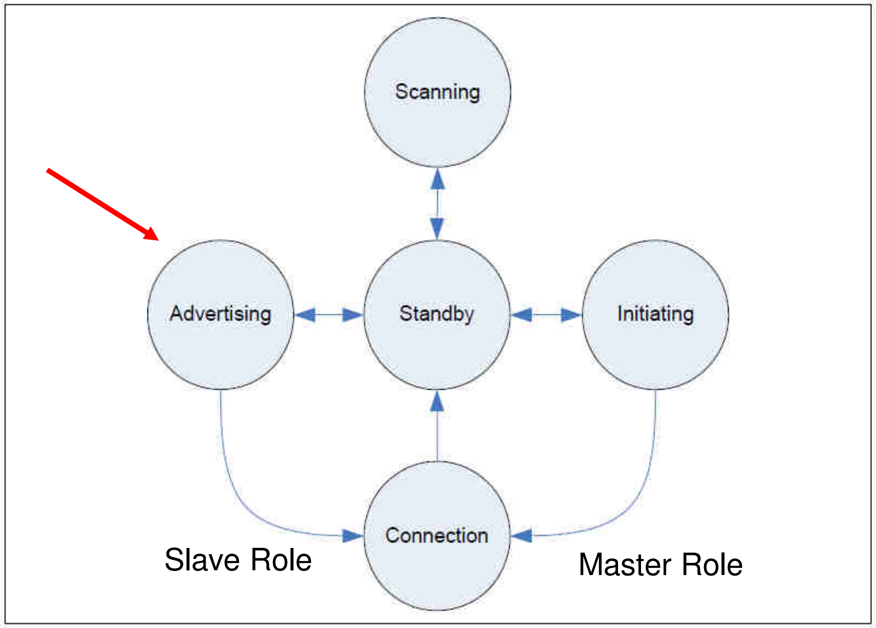
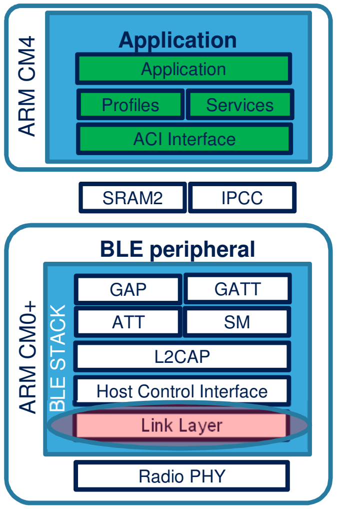


Figure 1.1: State diagram of the Link Layer state machine

Link Layer State Machine

- **Standby** state: Sleep, Stop, Standby
- **Advertising** is the key to initiating all BLE communications!
- An **Initiator** and **Advertiser** negotiate a **Connection**
- In a Connection
    - The Link-Layer ***Master*** is also the GAP **Central**
    - The Link-Layer ***Slave*** is also the GAP **Peripheral**



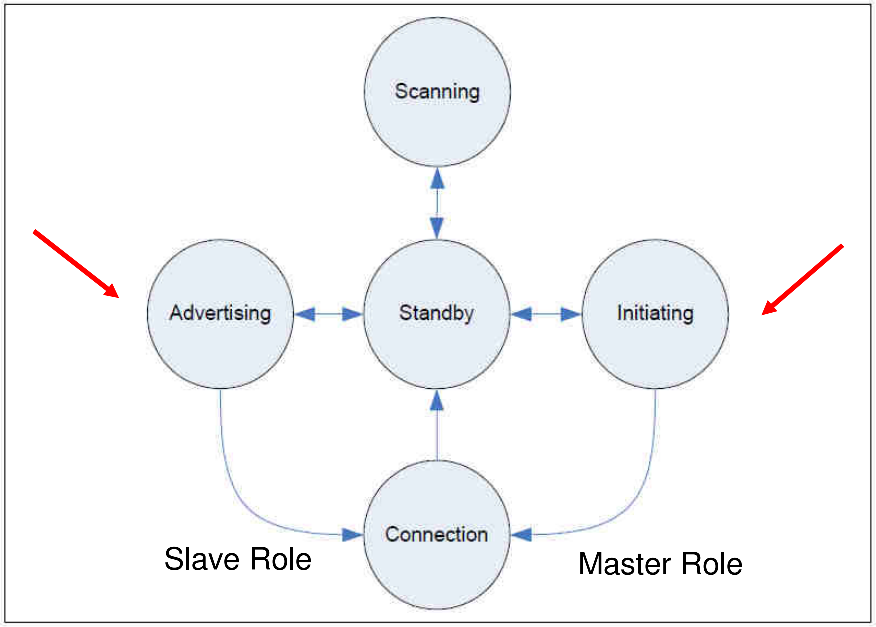
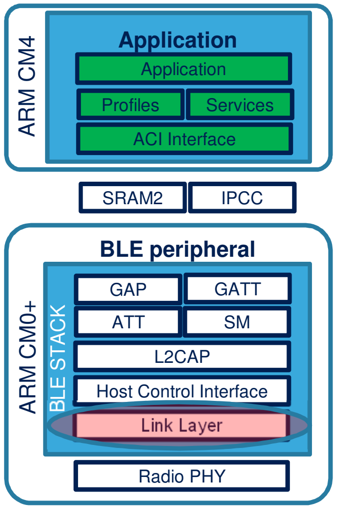


Figure 1.1: State diagram of the Link Layer state machine

Link Layer State Machine

Up to 8 simultaneous State Machines!

- **Standby** state: Sleep, Stop, Standby
- **Advertising** is the key to initiating all BLE communications!
- As an **Initiator** and **Advertiser** negotiate a **Connection**
- In a Connection
  - The Link-Layer ***Master*** is also the GAP **Central**
  - The Link-Layer ***Slave*** is also the GAP **Peripheral**



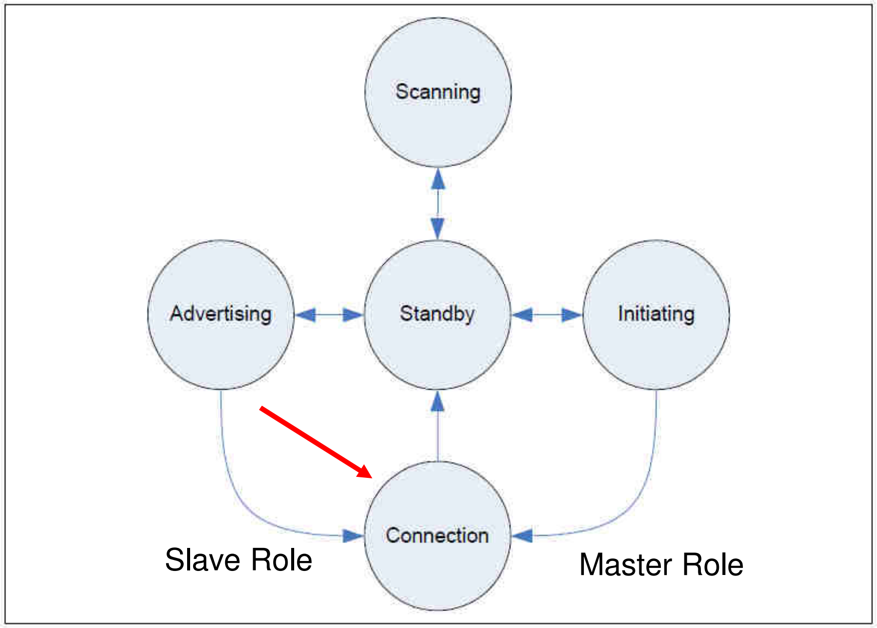
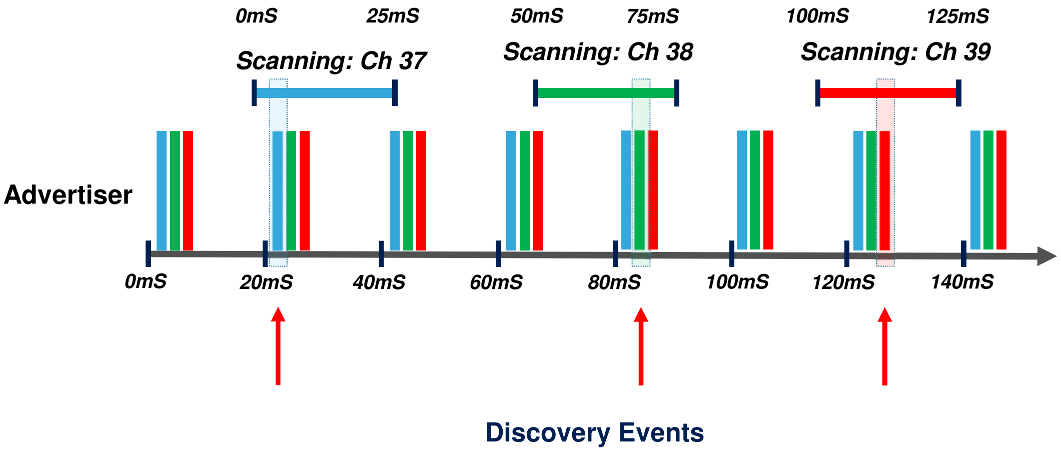
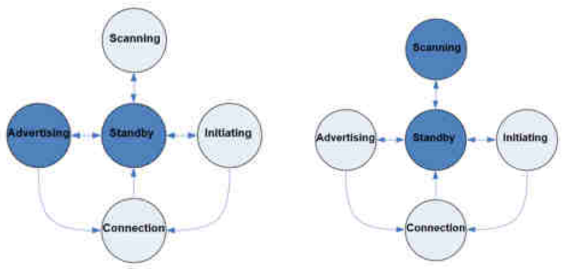


Figure 1.1: State diagram of the Link Layer state machine

Discovery: Advertising & Scanning



Advertising on Ch 37: 🟦
Advertising on Ch 38: 🟩
Advertising on Ch 39: 🟥



Advertiser Settings:

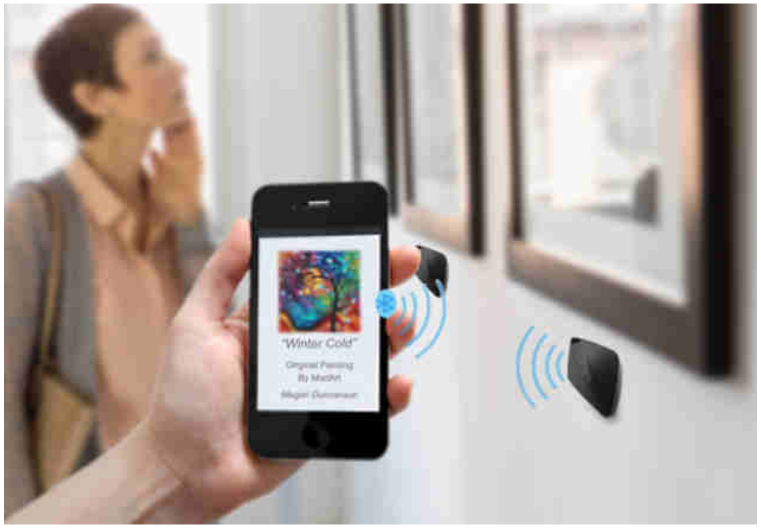
- **Advertising Interval: 20mS**

Scanner Settings:

- **Scan Interval: 50mS**
- **Scan Window: 25mS**

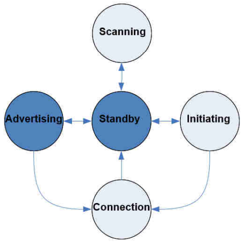
Roles and Modes

- Advertising Mode
- Connected Mode



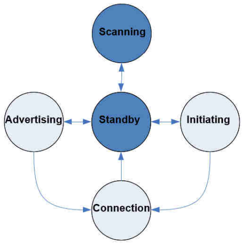
Broadcaster

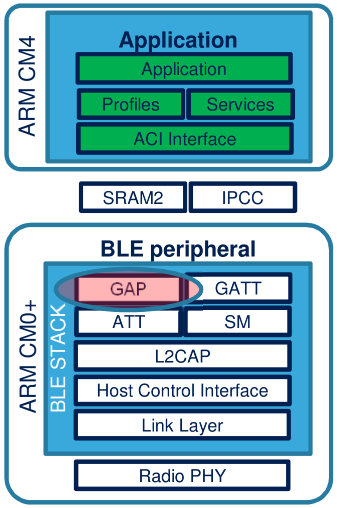
Sends advertising events
Can include characteristics and service data
Doesn't need receiver
Can be discoverable if it does have receiver



Observer

Receives advertising events
Listens for characteristics and service data
Doesn't need transmitter
Can discover devices if it does have transmitter



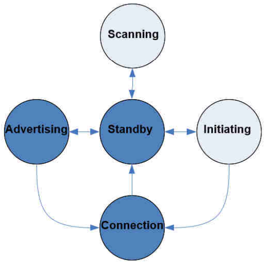


Roles and Modes

- Advertising Mode
- Connected Mode

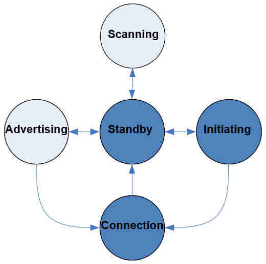
Peripheral

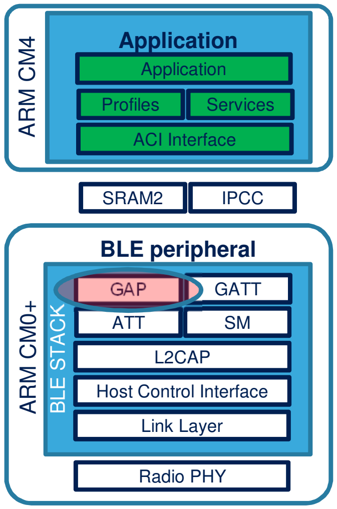
Has transmitter and receiver
Always slave
Connectable advertising



Central

Has transmitter and receiver
Always master
Never advertises

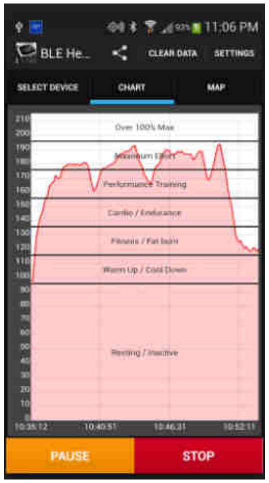


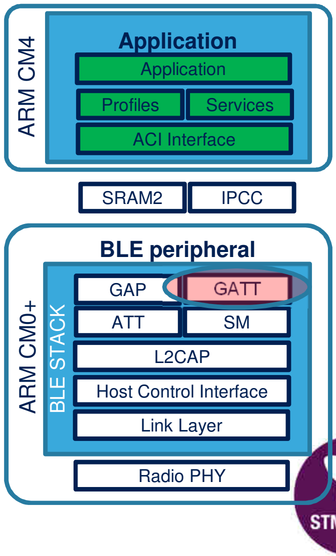


GATT (Generic Attribute Profile)

GAP Central is also a “GATT Client”

GAP Peripheral” is also a “GATT Server”




ARM CM4

Application

Application

Profiles | Services

ACI Interface

SRAM2 | IPCC

BLE peripheral

ARM CM0+

BLE STACK

GAP | GATT

ATT | SM

L2CAP

Host Control Interface

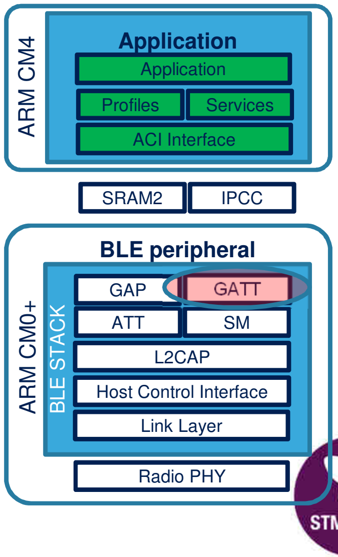
Link Layer

Radio PHY

GATT (Generic Attribute Profile)

GAP Central is also a “GATT Client”

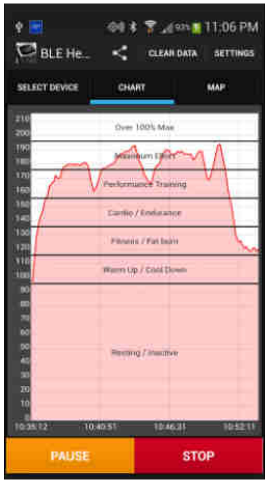
GAP Peripheral” is also a “GATT Server”



GATT (Generic Attribute Profile)

GAP Central is also a “GATT Client”
GAP Peripheral” is also a “GATT Server”

- Central queries the Services available
  - Peripheral **Services** and **Characteristics** are exposed via its’ **GATT** database

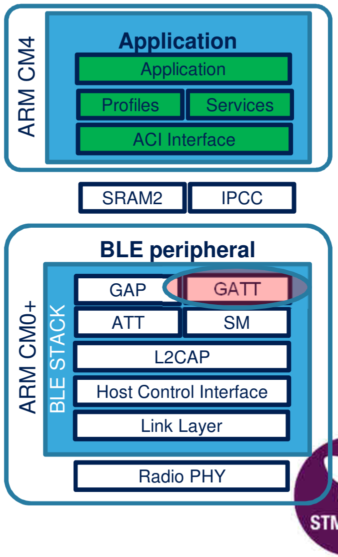


What is my heartrate? →

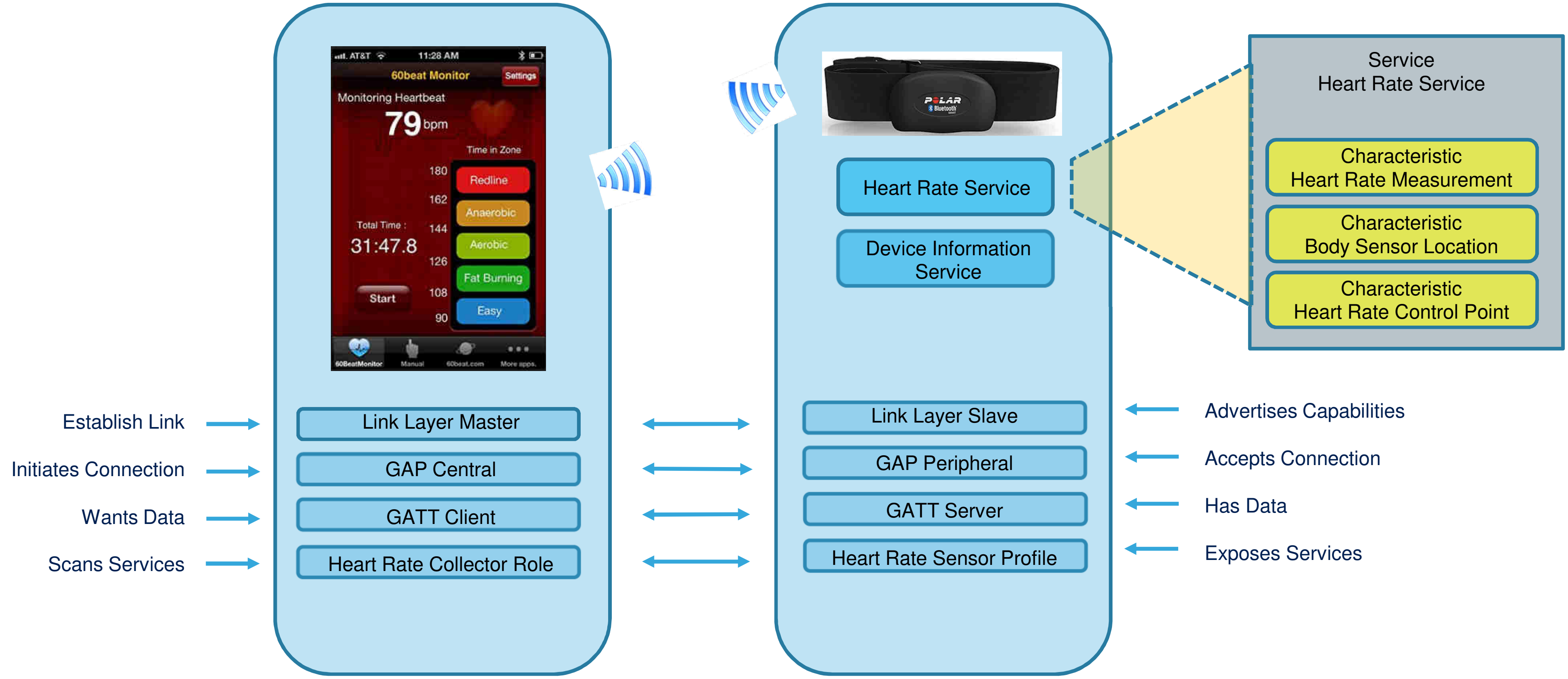
← 147 bpm

What is your Mfr ID? →

← Polar



GATT: Profiles, Services, Characteristics & Descriptors



Heart Rate Service

Device Information Service

Service
Heart Rate Service

Characteristic
Heart Rate Measurement

Characteristic
Body Sensor Location

Characteristic
Heart Rate Control Point

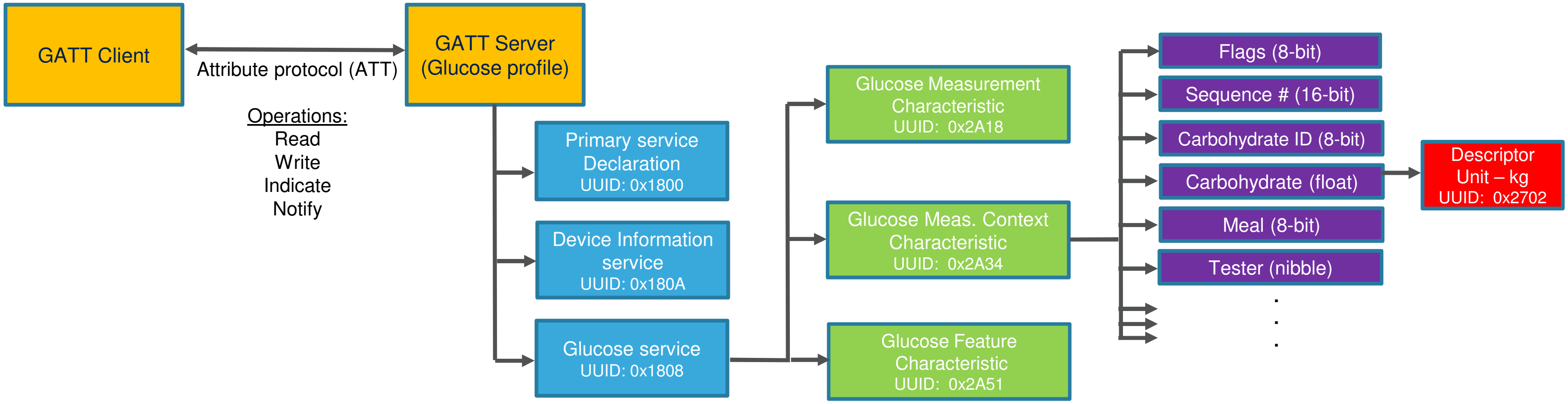
Establish Link → Link Layer Master ↔ Link Layer Slave ← Advertises Capabilities

Initiates Connection → GAP Central ↔ GAP Peripheral ← Accepts Connection

Wants Data → GATT Client ↔ GATT Server ← Has Data

Scans Services → Heart Rate Collector Role ↔ Heart Rate Sensor Profile ← Exposes Services

GLP Profile defines two roles: Collector & Glucose Sensor



GATT Client

Attribute protocol (ATT)

GATT Server (Glucose profile)

Operations:
Read
Write
Indicate
Notify

Primary service Declaration
UUID: 0x1800

Device Information service
UUID: 0x180A

Glucose service
UUID: 0x1808

Glucose Measurement Characteristic
UUID: 0x2A18

Glucose Meas. Context Characteristic
UUID: 0x2A34

Glucose Feature Characteristic
UUID: 0x2A51

Flags (8-bit)

Sequence # (16-bit)

Carbohydrate ID (8-bit)

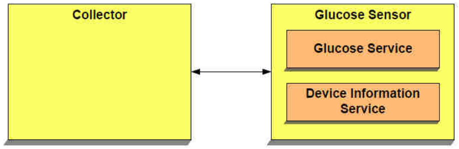
Carbohydrate (float)

Meal (8-bit)

Tester (nibble)

. . .

Descriptor
Unit – kg
UUID: 0x2702



Collector

Glucose Sensor

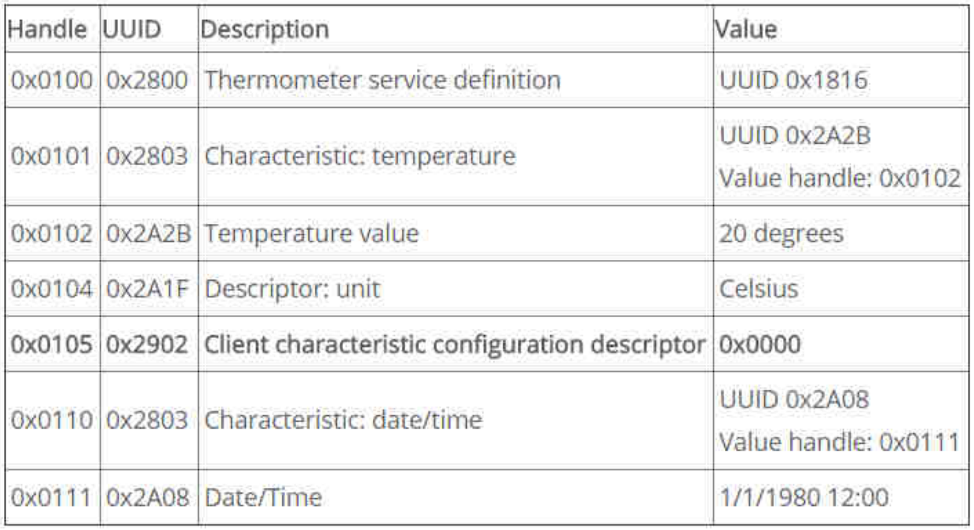
Glucose Service

Device Information Service

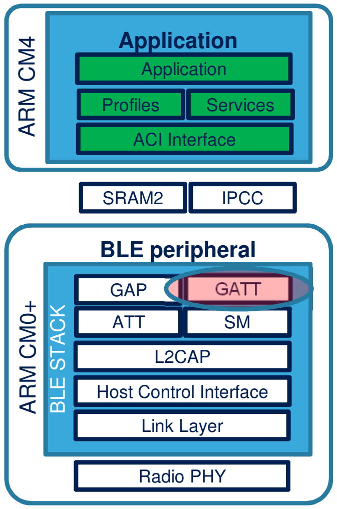
GATT Database details – Handles, UUID’s & Values

Once the GATT Server’s database information is known to the GATT Client, it can reference data via **Handles**

- ***“What is the temperature reported by the Thermometer Service?”*** ATT read command of Handle 0x0102
- ***“What are the units of temperature used?”*** ATT read command of Handle 0x0104



Handle	UUID	Description	Value
0x0100	0x2800	Thermometer service definition	UUID 0x1816
0x0101	0x2803	Characteristic: temperature	UUID 0x2A2B Value handle: 0x0102
0x0102	0x2A2B	Temperature value	20 degrees
0x0104	0x2A1F	Descriptor: unit	Celsius
0x0105	0x2902	Client characteristic configuration descriptor	0x0000
0x0110	0x2803	Characteristic: date/time	UUID 0x2A08 Value handle: 0x0111
0x0111	0x2A08	Date/Time	1/1/1980 12:00

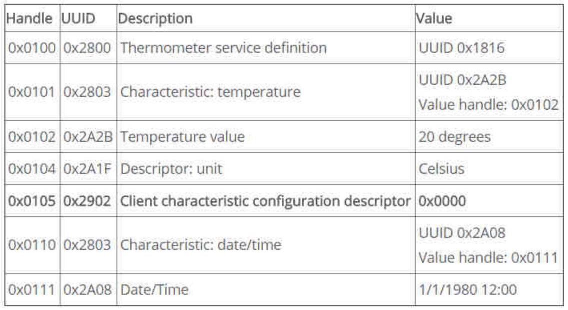


Attribute protocol details (ATT)

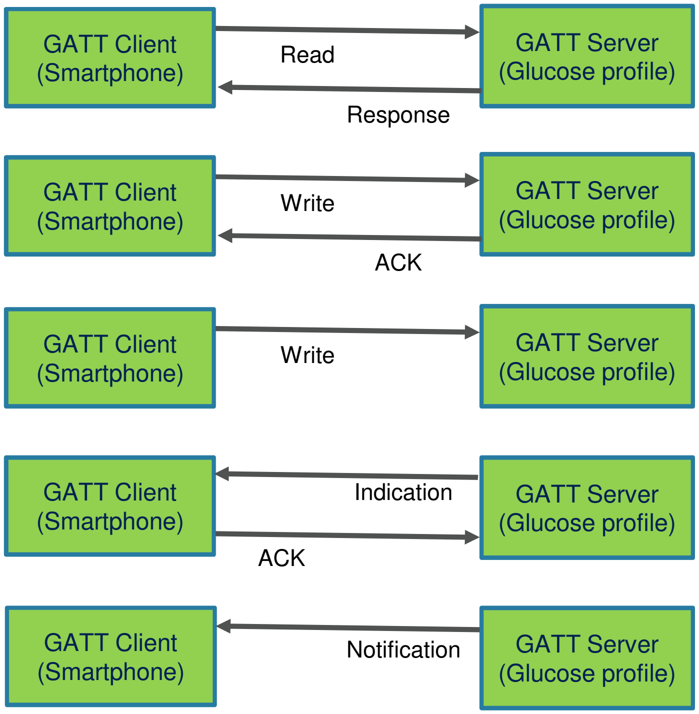
- Access GATT database information on the *Server*

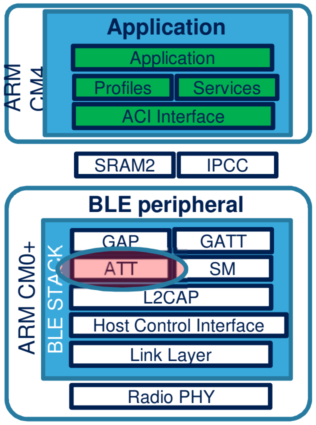
- Operations
  - Read
  - Write / Write without response
  - Indicate / Notify

- Four elements
  - 16-bit **Handle**
  - **Type** of attribute (UUID)
  - **Value**
  - Attribute **Permissions** (Read-only, etc)

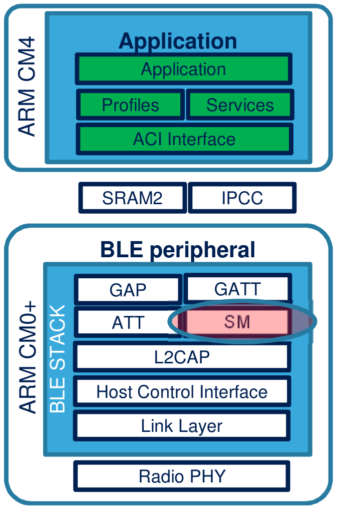


Handle	UUID	Description	Value
0x0100	0x2800	Thermometer service definition	UUID 0x1816
0x0101	0x2803	Characteristic: temperature	UUID 0x2A2B Value handle: 0x0102
0x0102	0x2A2B	Temperature value	20 degrees
0x0104	0x2A1F	Descriptor: unit	Celsius
0x0105	0x2902	Client characteristic configuration descriptor	0x0000
0x0110	0x2803	Characteristic: date/time	UUID 0x2A08 Value handle: 0x0111
0x0111	0x2A08	Date/Time	1/1/1980 12:00





- **Connection**: GAP Central connected to a GAP Peripheral (Connection interval = 7.5ms to 4 secs)
- **Pairing**: Connected devices exchange encryption keys to **encrypt** the link. There are now **paired**.
- **Bonding**: Paired devices can be bonded – Keys are stored for the next connection.
- **Whitelisting**: Restrict connections from any other than known devices.



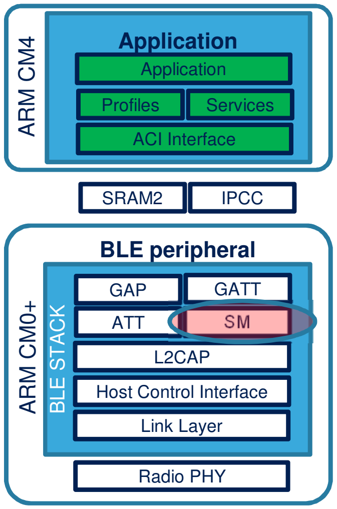
Security Topics

- **Connection**: GAP Central connected to a GAP Peripheral (Connection interval = 7.5ms to 4 secs)

- **Pairing**: Connected devices exchange encryption keys to **encrypt** the link. There are now **paired**.

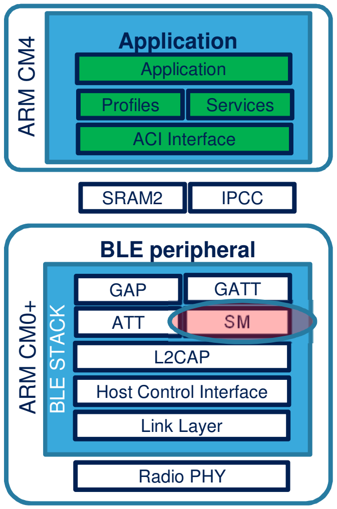
- **Bonding**: Paired devices can be bonded – Keys are stored for the next connection.

- **Whitelisting**: Restrict connections from any other than known devices.



Security Topics

- **Connection**: GAP Central connected to a GAP Peripheral (Connection interval = 7.5ms to 4 secs)
- **Pairing**: Connected devices exchange encryption keys to **encrypt** the link. There are now **paired**.
- **Bonding**: Paired devices can be bonded – Keys are stored for the next connection.
- **Whitelisting**: Restrict connections from any other than known devices.



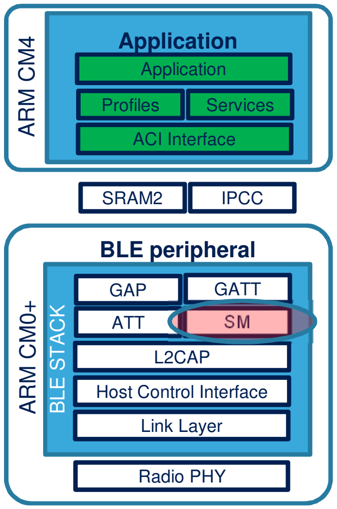
Security Topics

- **Connection:** GAP Central connected to a GAP Peripheral (Connection interval = 7.5ms to 4 secs)

- **Pairing:** Connected devices exchange encryption keys to **encrypt** the link. There are now **paired**.

- **Bonding:** Paired devices can be bonded – Keys are stored for the next connection.

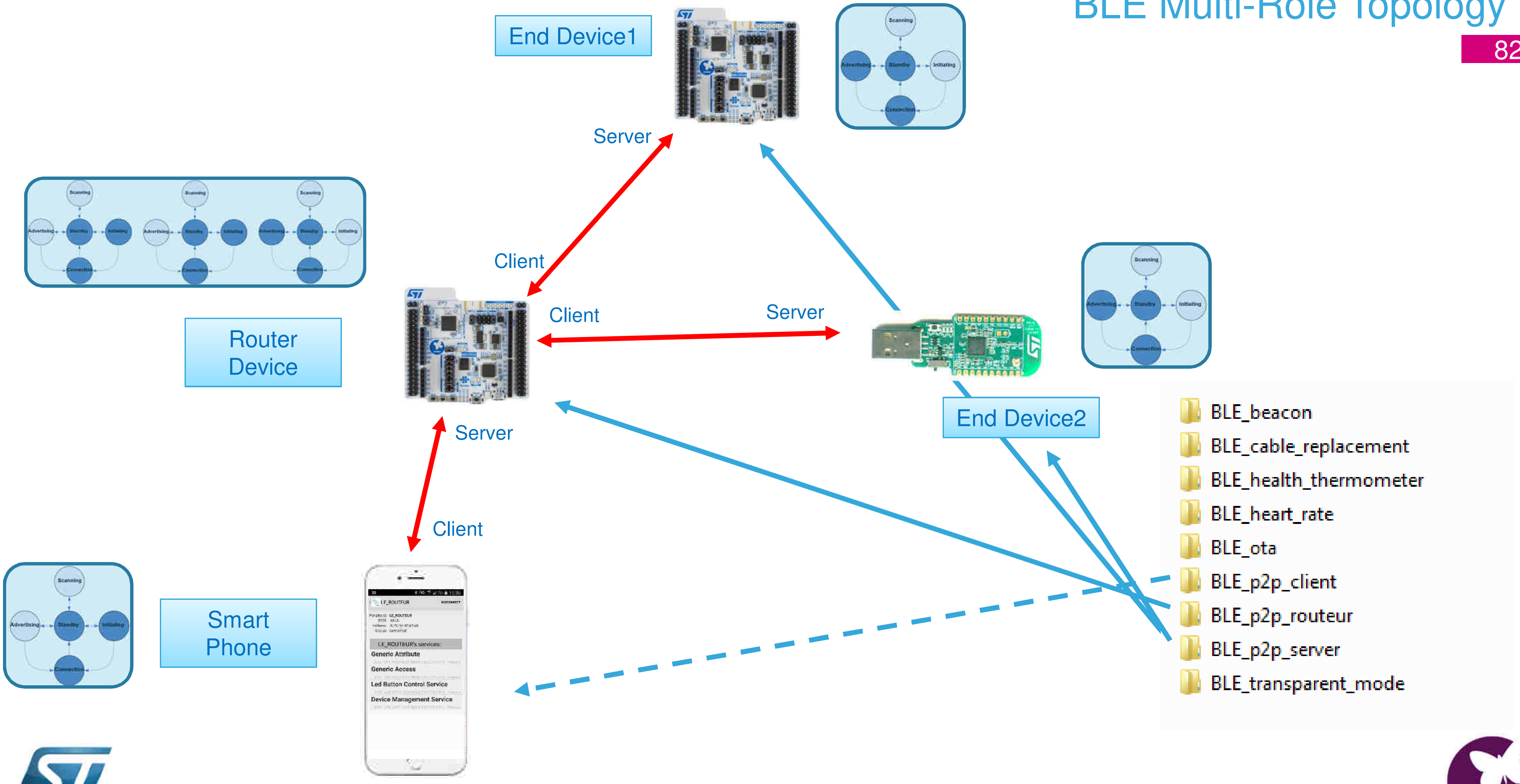
- **Whitelisting:** Restrict connections from any other than known devices.



- Security modes are deployed after a BLE connection is established

- BLE Link Layer uses AES-128 CCM mode for authenticated encryption

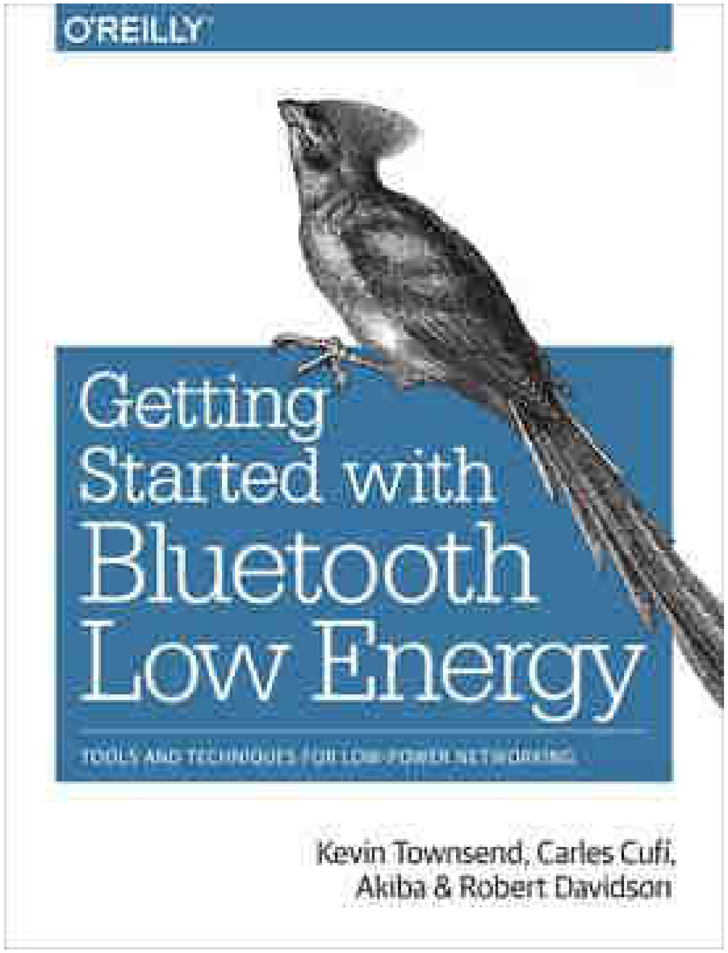
BLE Multi-Role Topology



Dig Deeper

$36 on Amazon

5.0 not covered however



O'REILLY

Getting Started with Bluetooth Low Energy

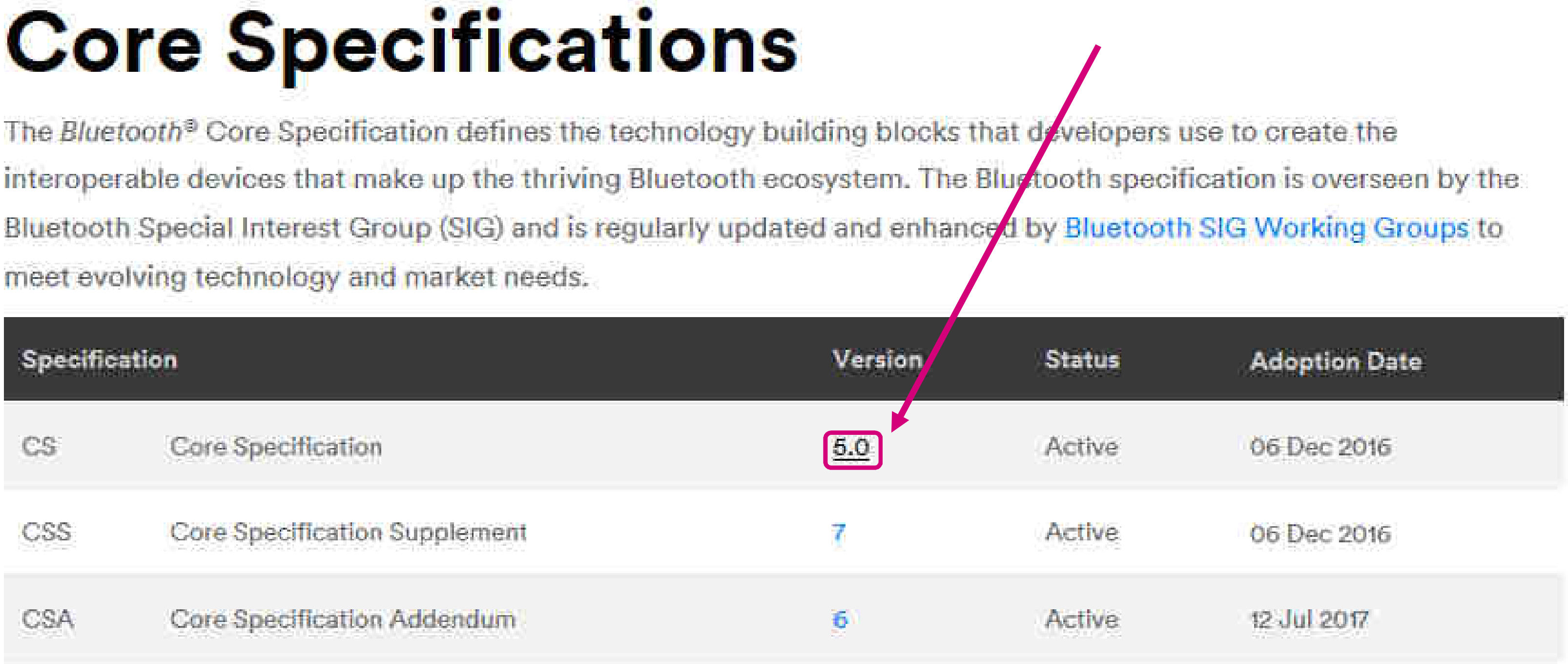
TOOLS AND TECHNIQUES FOR LOW-POWER NETWORKING

Kevin Townsend, Carles Cufí,
Akiba & Robert Davidson

2014

Dig Deeper

Bluetooth® https://www.bluetooth.com/



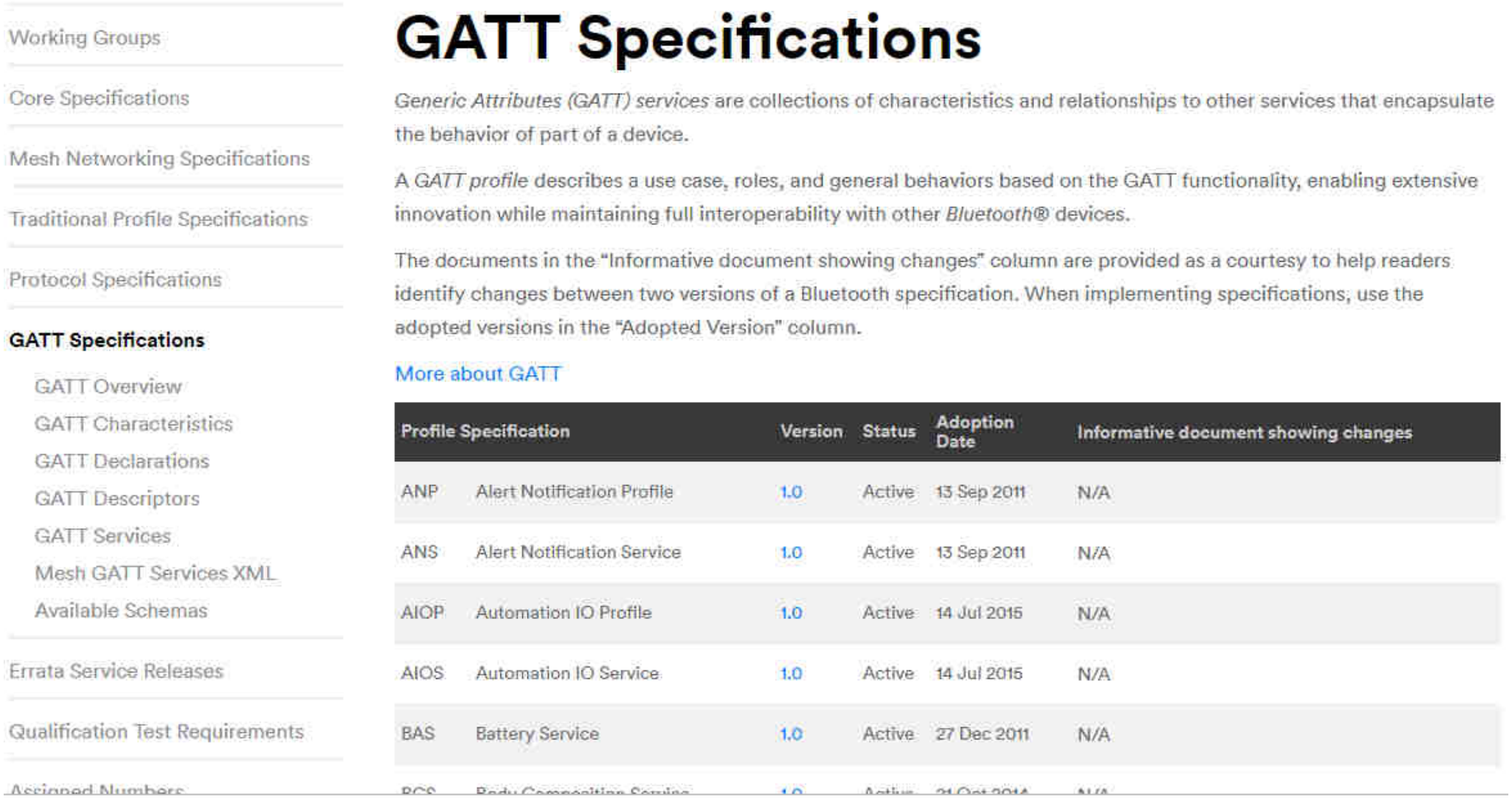
Core Specifications

The *Bluetooth*® Core Specification defines the technology building blocks that developers use to create the interoperable devices that make up the thriving Bluetooth ecosystem. The Bluetooth specification is overseen by the Bluetooth Special Interest Group (SIG) and is regularly updated and enhanced by Bluetooth SIG Working Groups to meet evolving technology and market needs.

Specification		Version	Status	Adoption Date
CS	Core Specification	5.0	Active	06 Dec 2016
CSS	Core Specification Supplement	7	Active	06 Dec 2016
CSA	Core Specification Addendum	6	Active	12 Jul 2017

Bluetooth®

https://www.bluetooth.com/specifications/gatt



- Working Groups
- Core Specifications
- Mesh Networking Specifications
- Traditional Profile Specifications
- Protocol Specifications
- **GATT Specifications**
  - GATT Overview
  - GATT Characteristics
  - GATT Declarations
  - GATT Descriptors
  - GATT Services
  - Mesh GATT Services XML
  - Available Schemas
- Errata Service Releases
- Qualification Test Requirements
- Assigned Numbers

GATT Specifications

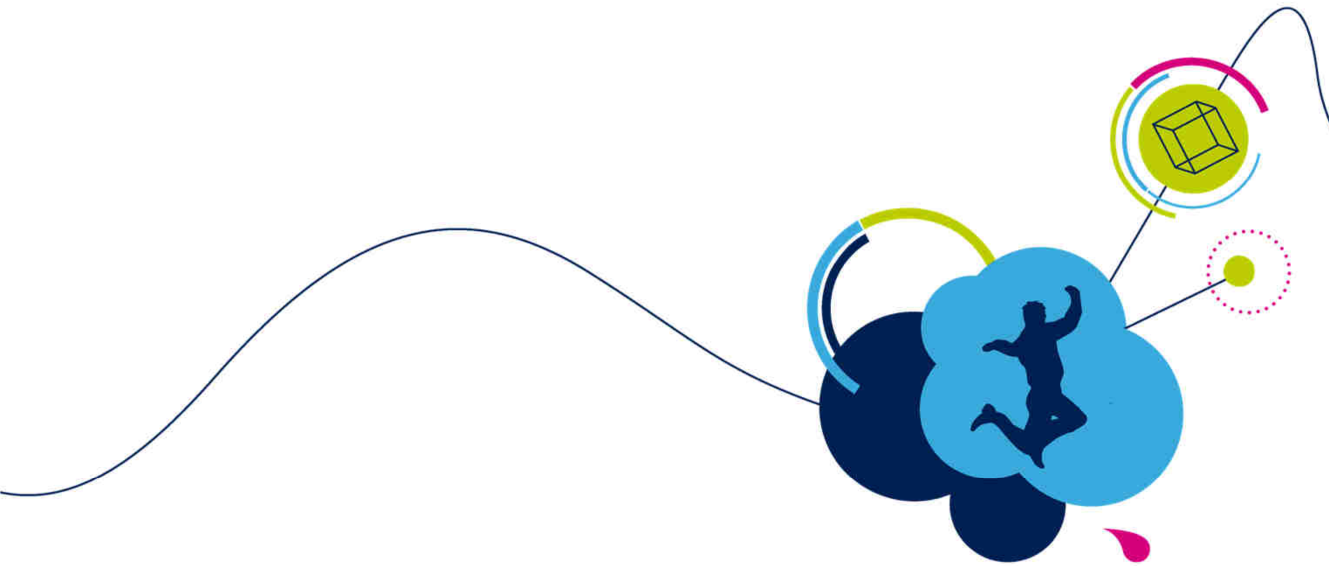
Generic Attributes (GATT) services are collections of characteristics and relationships to other services that encapsulate the behavior of part of a device.

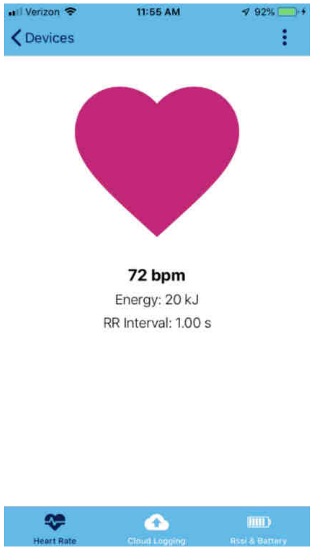
A *GATT profile* describes a use case, roles, and general behaviors based on the GATT functionality, enabling extensive innovation while maintaining full interoperability with other *Bluetooth®* devices.

The documents in the "Informative document showing changes" column are provided as a courtesy to help readers identify changes between two versions of a Bluetooth specification. When implementing specifications, use the adopted versions in the "Adopted Version" column.

More about GATT

Profile Specification		Version	Status	Adoption Date	Informative document showing changes
ANP	Alert Notification Profile	1.0	Active	13 Sep 2011	N/A
ANS	Alert Notification Service	1.0	Active	13 Sep 2011	N/A
AIOP	Automation IO Profile	1.0	Active	14 Jul 2015	N/A
AIOS	Automation IO Service	1.0	Active	14 Jul 2015	N/A
BAS	Battery Service	1.0	Active	27 Dec 2011	N/A
BCS	Body Composition Service	1.0	Active	21 Oct 2014	N/A





72 bpm

Energy: 20 kJ

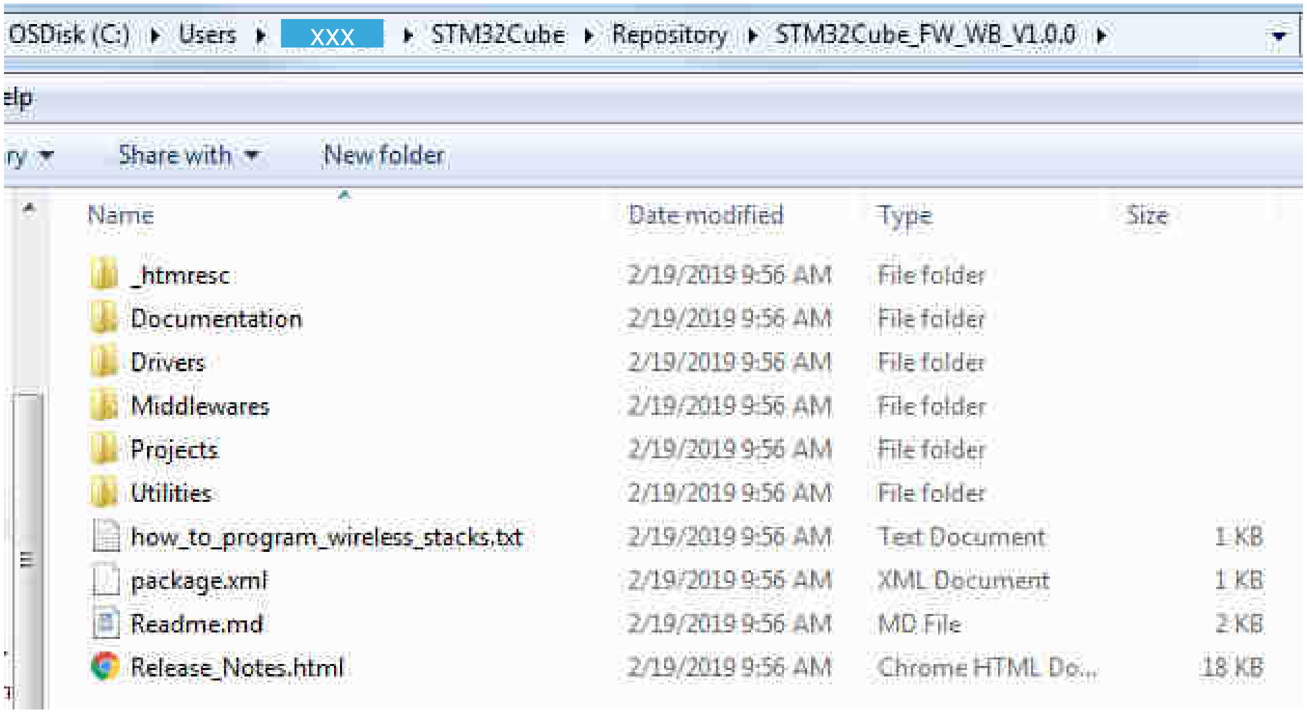
RR Interval: 1.00 s

Hands-On

Heart-Rate Monitor

Hands-On: HRM example

All CubeWB Projects referenced can be found in the CubeMX Repository folder:

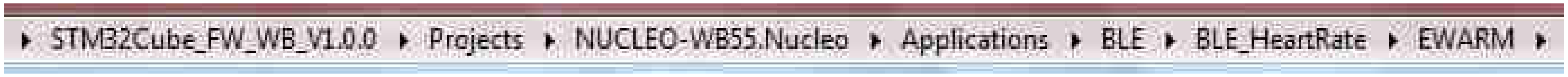


OSDisk (C:) ▸ Users ▸ xxx ▸ STM32Cube ▸ Repository ▸ STM32Cube_FW_WB_V1.0.0 ▸

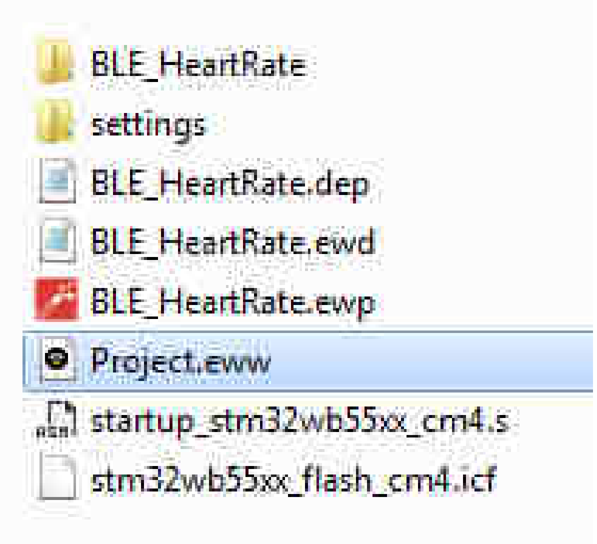
Name	Date modified	Type	Size
_htmresc	2/19/2019 9:56 AM	File folder	
Documentation	2/19/2019 9:56 AM	File folder	
Drivers	2/19/2019 9:56 AM	File folder	
Middlewares	2/19/2019 9:56 AM	File folder	
Projects	2/19/2019 9:56 AM	File folder	
Utilities	2/19/2019 9:56 AM	File folder	
how_to_program_wireless_stacks.txt	2/19/2019 9:56 AM	Text Document	1 KB
package.xml	2/19/2019 9:56 AM	XML Document	1 KB
Readme.md	2/19/2019 9:56 AM	MD File	2 KB
Release_Notes.html	2/19/2019 9:56 AM	Chrome HTML Do...	18 KB

Hands-On: HRM example

Open the BLE_HeartRate workspace

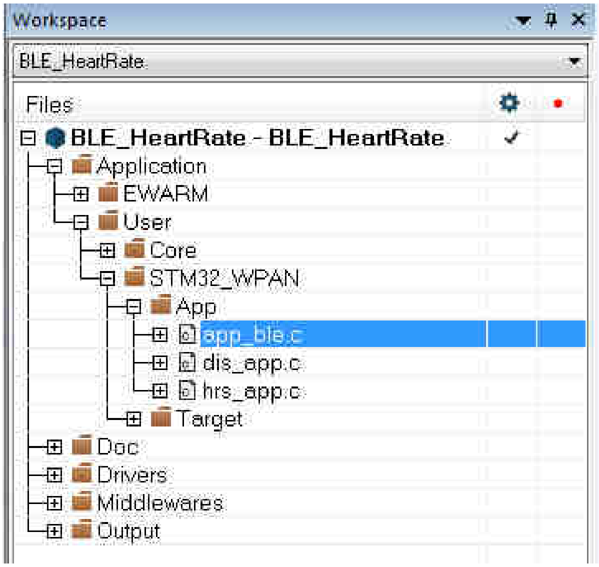


▸ STM32Cube_FW_WB_V1.0.0 ▸ Projects ▸ NUCLEO-WB55.Nucleo ▸ Applications ▸ BLE ▸ BLE_HeartRate ▸ EWARM ▸



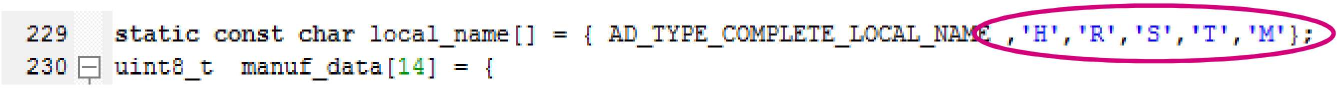
- BLE_HeartRate
- settings
- BLE_HeartRate.dep
- BLE_HeartRate.ewd
- BLE_HeartRate.ewp
- Project.eww
- startup_stm32wb55xx_cm4.s
- stm32wb55xx_flash_cm4.icf

Open **app_ble.c**

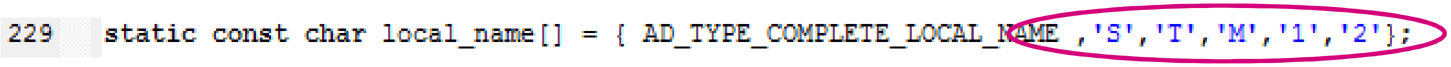


Change the **local name**, using your *Magic Number*!

*(You can change it as you wish, however keep the # of ASCII chars to **5**)*

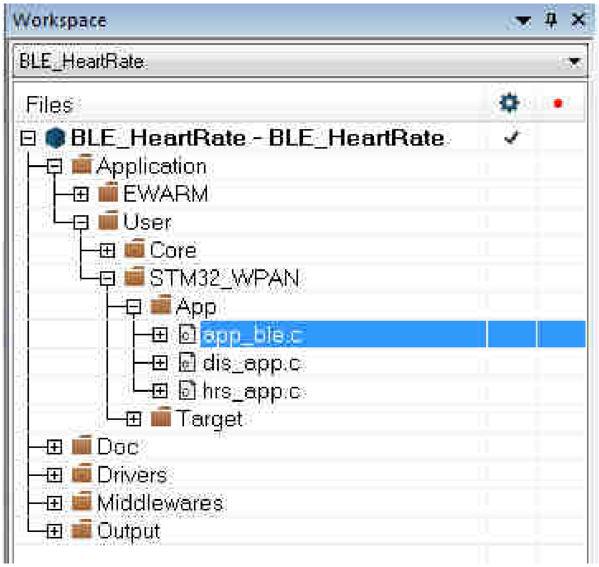


```
static const char local_name[] = { AD_TYPE_COMPLETE_LOCAL_NAME ,'H','R','S','T','M'};
uint8_t  manuf_data[14] = {
```

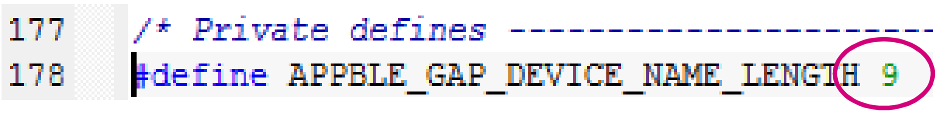


```
static const char local_name[] = { AD_TYPE_COMPLETE_LOCAL_NAME ,'S','T','M','1','2'};
```

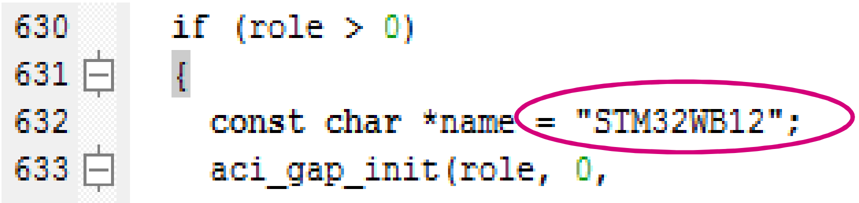
Hands-On: HRM example



Also change the advertised **name** and the **NAME_LENGTH**, using your *Magic Number*!



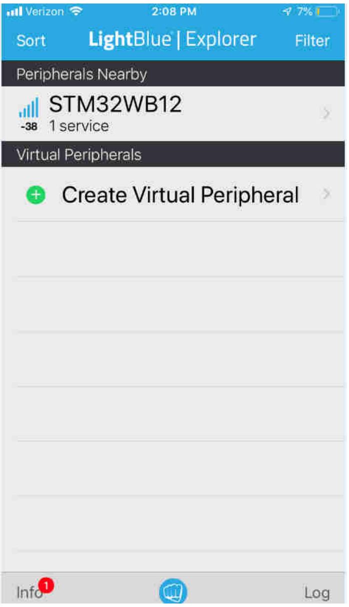
```
/* Private defines -----------------------
#define APPBLE_GAP_DEVICE_NAME_LENGTH 9
```



```
  if (role > 0)
  {
    const char *name = "STM32WB12";
    aci_gap_init(role, 0,
```

Hands-On: HRM example

- Open your LightBlue Explorer App on **iOS**
- Find your device and tap on it
- **Show** Advertisement Data
- Click on the Heart Rate Measurement and Enable Notifications

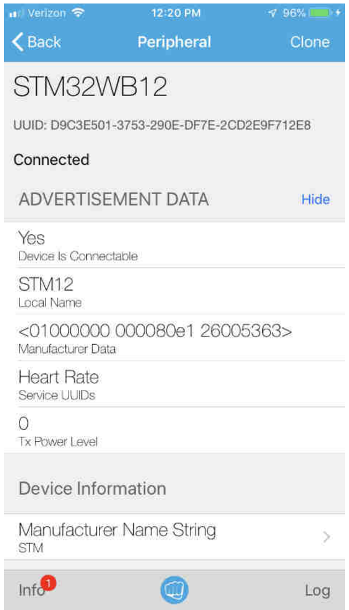


Peripherals Nearby

STM32WB12
-38 1 service

Virtual Peripherals

Create Virtual Peripheral



STM32WB12

UUID: D9C3E501-3753-290E-DF7E-2CD2E9F712E8

Connected

ADVERTISEMENT DATA Hide

Yes
Device Is Connectable

STM12
Local Name

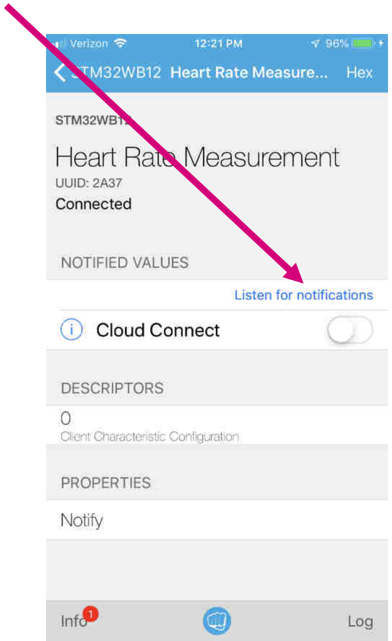
<01000000 000080e1 26005363>
Manufacturer Data

Heart Rate
Service UUIDs

0
Tx Power Level

Device Information

Manufacturer Name String
STM



STM32WB12

Heart Rate Measurement
UUID: 2A37
Connected

NOTIFIED VALUES

Listen for notifications

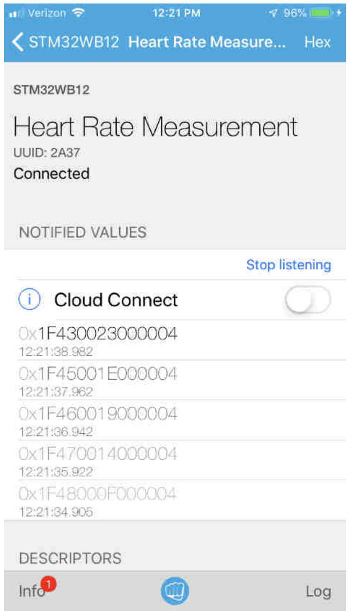
Cloud Connect

DESCRIPTORS

0
Client Characteristic Configuration

PROPERTIES

Notify



STM32WB12

Heart Rate Measurement
UUID: 2A37
Connected

NOTIFIED VALUES

Stop listening

Cloud Connect

0x1F430023000004
12:21:38.982

0x1F45001E000004
12:21:37.962

0x1F460019000004
12:21:36.942

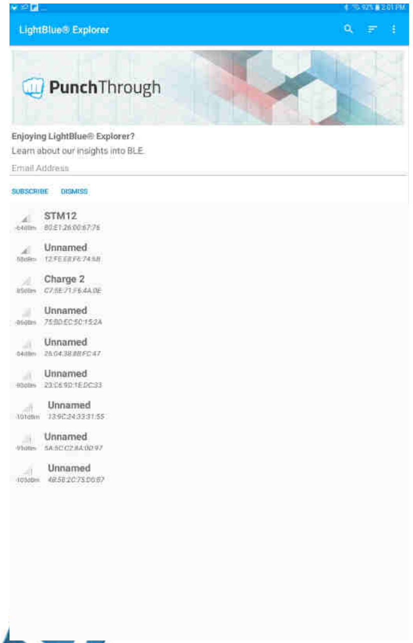
0x1F470014000004
12:21:35.922

0x1F48000F000004
12:21:34.905

DESCRIPTORS

Hands-On: HRM example

- Open your LightBlue Explorer App on **Android**
- Find your device and tap on it
- Tap on the Heart Rate section and select Heart Rate Measurement
- Tap on “SUBSCRIBE” to Enable Notifications



LightBlue® Explorer

Enjoying LightBlue® Explorer?
Learn about our insights into BLE.
Email Address

SUBSCRIBE DISMISS

STM12
-64dBm 80:E1:26:00:67:76

Unnamed
-58dBm 12:FE:E8:F6:74:58

Charge 2
-85dBm C7:5E:71:F6:4A:0E

Unnamed
-86dBm 75:BD:EC:5C:15:2A

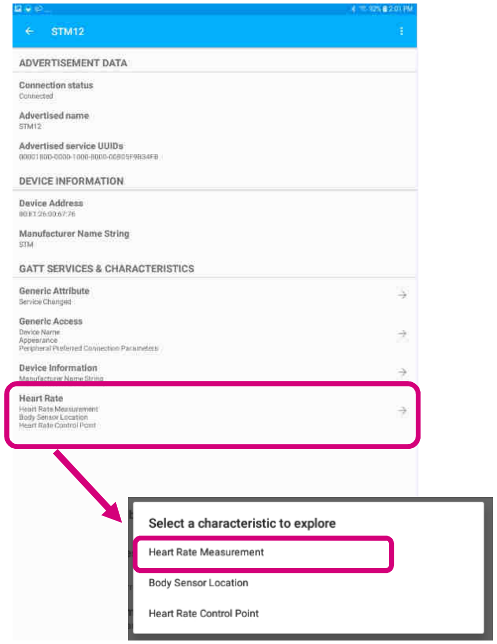
Unnamed
-84dBm 26:04:38:8B:FC:47

Unnamed
-90dBm 23:C6:9D:1E:DC:33

Unnamed
-101dBm 13:9C:24:33:31:55

Unnamed
-95dBm 5A:5C:C2:8A:00:97

Unnamed
-103dBm 48:58:2C:75:D6:87



STM12

ADVERTISEMENT DATA

Connection status
Connected

Advertised name
STM12

Advertised service UUIDs
0000180D-0000-1000-8000-00805F9B34FB

DEVICE INFORMATION

Device Address
80:E1:26:00:67:76

Manufacturer Name String
STM

GATT SERVICES & CHARACTERISTICS

Generic Attribute
Service Changed

Generic Access
Device Name
Appearance
Peripheral Preferred Connection Parameters

Device Information
Manufacturer Name String

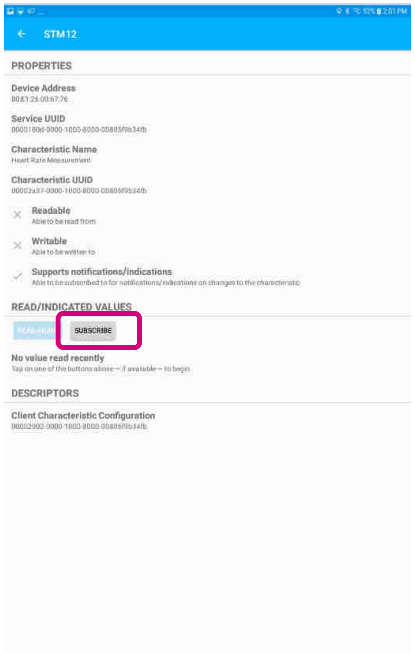
Heart Rate
Heart Rate Measurement
Body Sensor Location
Heart Rate Control Point

Select a characteristic to explore

Heart Rate Measurement

Body Sensor Location

Heart Rate Control Point



STM12

PROPERTIES

Device Address
80:E1:26:00:67:76

Service UUID
0000180d-0000-1000-8000-00805f9b34fb

Characteristic Name
Heart Rate Measurement

Characteristic UUID
00002a37-0000-1000-8000-00805f9b34fb

✗ Readable
Able to be read from

✗ Writable
Able to be written to

✓ Supports notifications/indications
Able to be subscribed to for notifications/indications on changes to the characteristic

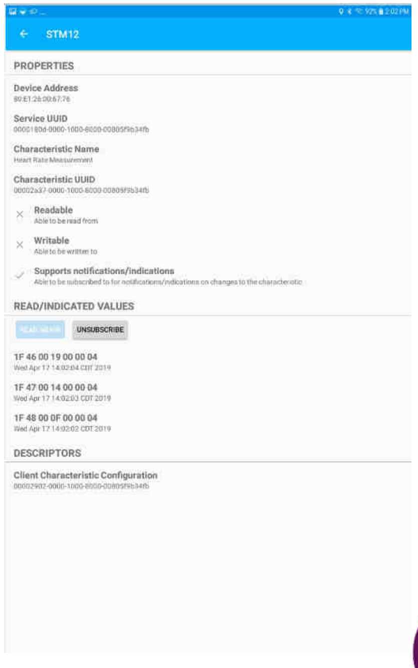
READ/INDICATED VALUES

READ AGAIN SUBSCRIBE

No value read recently
Tap on one of the buttons above — if available — to begin

DESCRIPTORS

Client Characteristic Configuration
00002902-0000-1000-8000-00805f9b34fb



STM12

PROPERTIES

Device Address
80:E1:26:00:67:76

Service UUID
0000180d-0000-1000-8000-00805f9b34fb

Characteristic Name
Heart Rate Measurement

Characteristic UUID
00002a37-0000-1000-8000-00805f9b34fb

✗ Readable
Able to be read from

✗ Writable
Able to be written to

✓ Supports notifications/indications
Able to be subscribed to for notifications/indications on changes to the characteristic

READ/INDICATED VALUES

READ AGAIN UNSUBSCRIBE

1F 46 00 19 00 00 04
Wed Apr 17 14:02:04 CDT 2019

1F 47 00 14 00 00 04
Wed Apr 17 14:02:03 CDT 2019

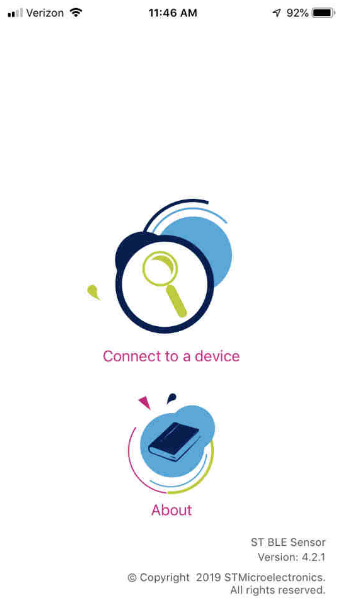
1F 48 00 0F 00 00 04
Wed Apr 17 14:02:02 CDT 2019

DESCRIPTORS

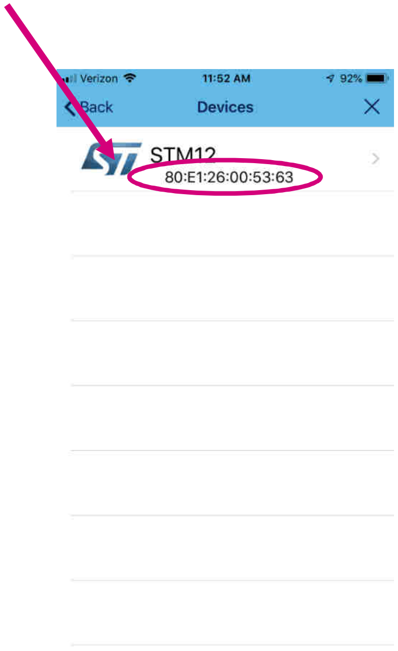
Client Characteristic Configuration
00002902-0000-1000-8000-00805f9b34fb

Hands-On: HRM example

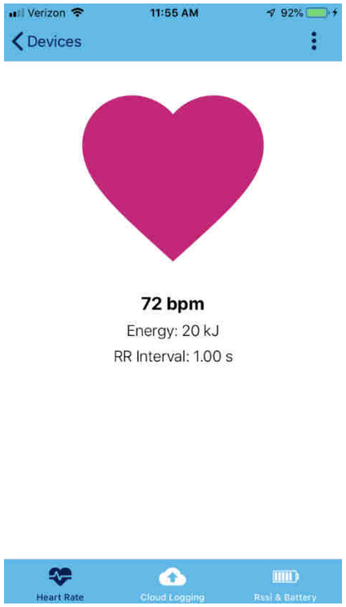
- Disconnect from the LightBlue Explorer App
- Launch the ST BLE Sensor App
- Tap on your device name
- Note your Nucleo Bluetooth Device Address.
  - ➢ *Can you find it in the Mfr-Specific advertised data via LightBlue app?*



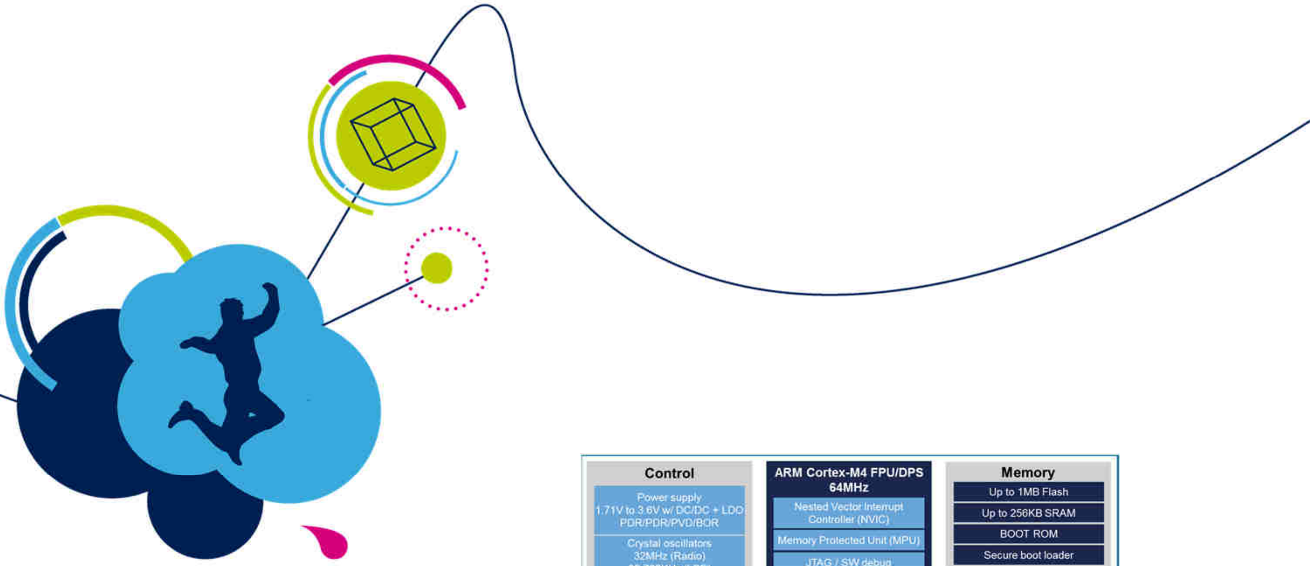
ST BLE Sensor
Version: 4.2.1
© Copyright 2019 STMicroelectronics.
All rights reserved.



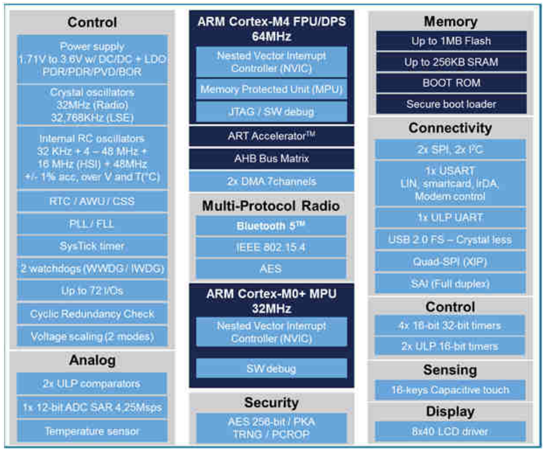
STM12
80:E1:26:00:53:63



72 bpm
Energy: 20 kJ
RR Interval: 1.00 s



WB Architecture



Control

Power supply
1.71V to 3.6V w/ DC/DC + LDO
PDR/PDR/PVD/BOR

Crystal oscillators
32MHz (Radio)
32,768KHz (LSE)

Internal RC oscillators
32 KHz + 4 – 48 MHz +
16 MHz (HSI) + 48MHz
+/- 1% acc, over V and T(°C)

RTC / AWU / CSS

PLL / FLL

SysTick timer

2 watchdogs (WWDG / IWDG)

Up to 72 I/Os

Cyclic Redundancy Check

Voltage scaling (2 modes)

Analog

2x ULP comparators

1x 12-bit ADC SAR 4,25Msps

Temperature sensor

ARM Cortex-M4 FPU/DPS 64MHz

Nested Vector Interrupt Controller (NVIC)

Memory Protected Unit (MPU)

JTAG / SW debug

ART Accelerator™

AHB Bus Matrix

2x DMA 7channels

Multi-Protocol Radio

Bluetooth 5™

IEEE 802.15.4

AES

ARM Cortex-M0+ MPU 32MHz

Nested Vector Interrupt Controller (NVIC)

SW debug

Security

AES 256-bit / PKA
TRNG / PCROP

Memory

Up to 1MB Flash

Up to 256KB SRAM

BOOT ROM

Secure boot loader

Connectivity

2x SPI, 2x I²C

1x USART
LIN, smartcard, IrDA,
Modem control

1x ULP UART

USB 2.0 FS – Crystal less

Quad-SPI (XIP)

SAI (Full duplex)

Control

4x 16-bit 32-bit timers

2x ULP 16-bit timers

Sensing

16-keys Capacitive touch

Display

8x40 LCD driver

RF System Front-End

Balun – Combine TX and RX signals

Matching Network – 50 Ω impedance transformation

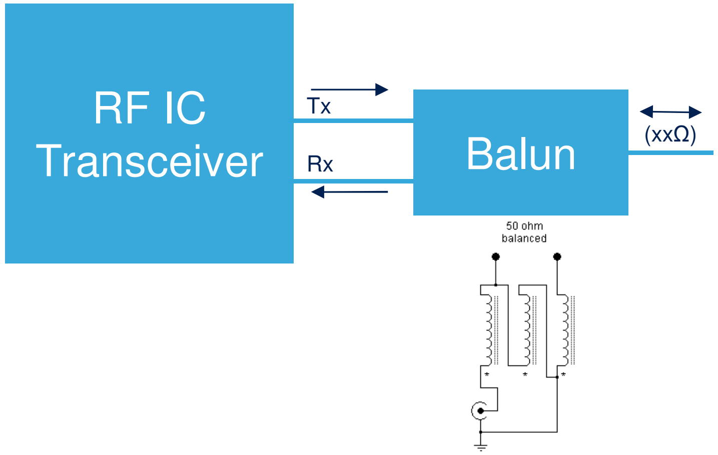
Harmonic Filter – Reduce out-of-band harmonics

RF System Front-End

Balun – Combine TX and RX signals

Matching Network – 50 Ω impedance transformation

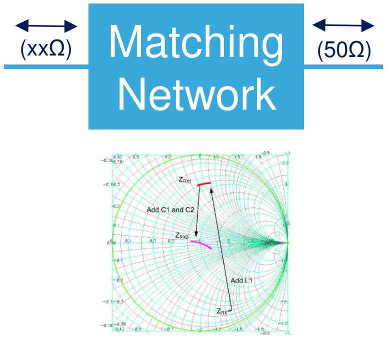
Harmonic Filter – Reduce out-of-band harmonics



Balun – Combine TX and RX signals

Matching Network – 50 Ω impedance transformation

Harmonic Filter – Reduce out-of-band harmonics

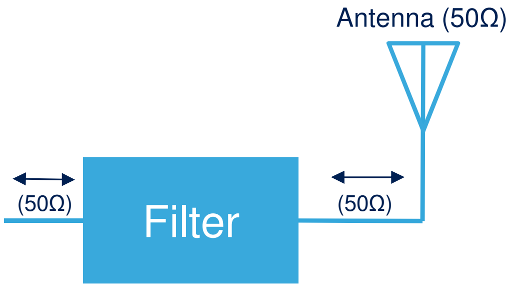


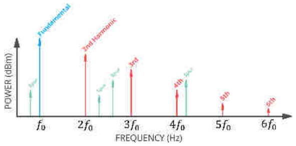
RF System Front-End

Balun – Combine TX and RX signals

Matching Network – 50 Ω impedance transformation

Harmonic Filter – Reduce out-of-band harmonics



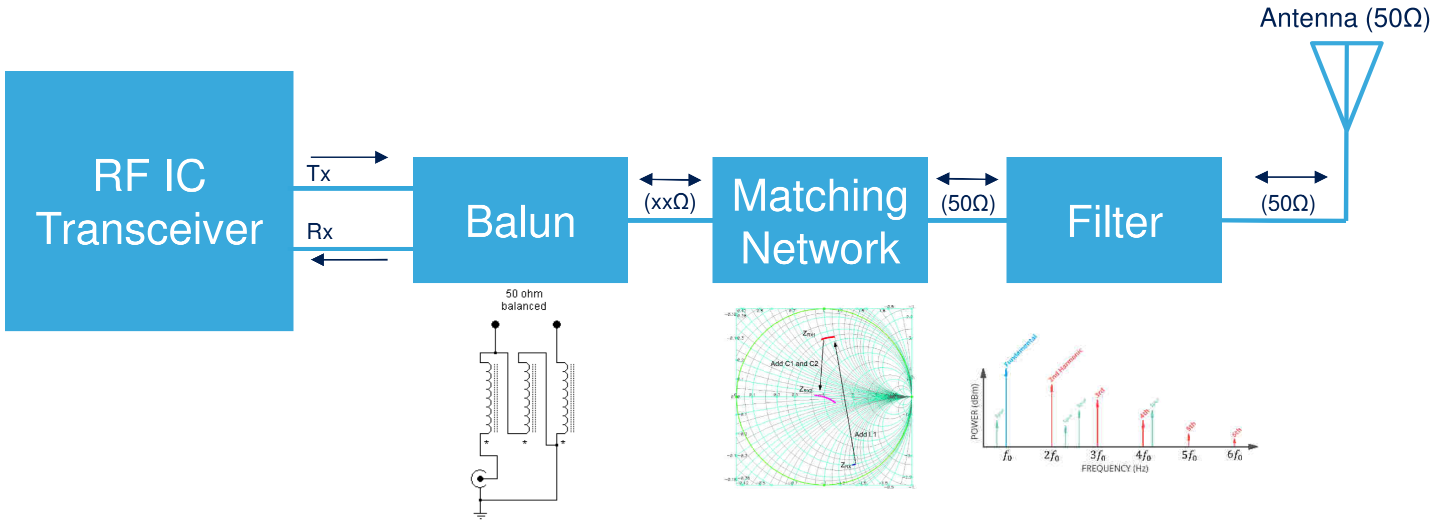


RF System Front-End

Balun – Combine TX and RX signals

Matching Network – 50 Ω impedance transformation

Harmonic Filter – Reduce out-of-band harmonics

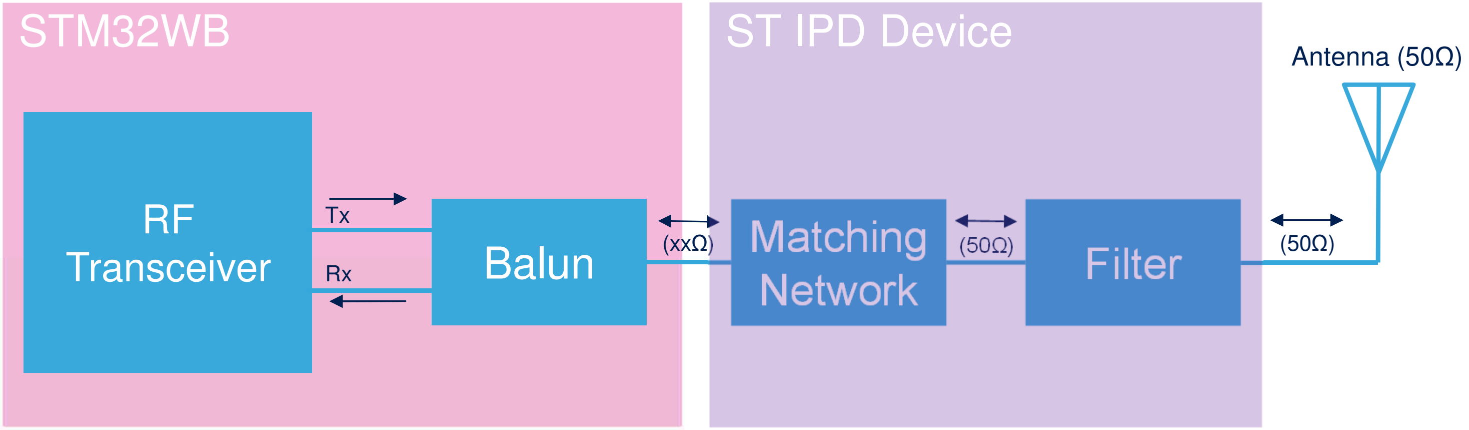


Basic RF System

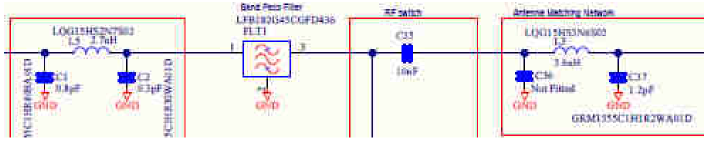
Balun – Combine TX and RX signals

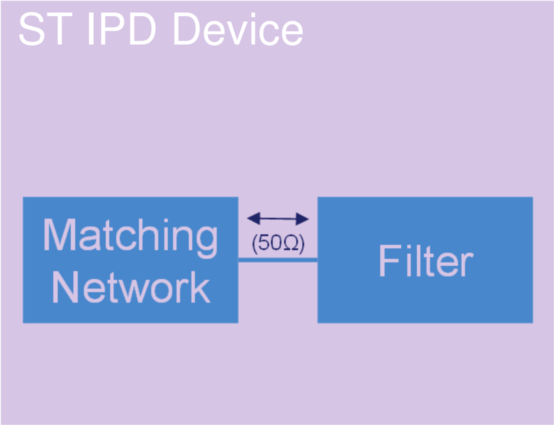
Matching Network – 50 Ω impedance transformation

Harmonic Filter – Reduce out-of-band harmonics



Matching Network + Harmonic Filter



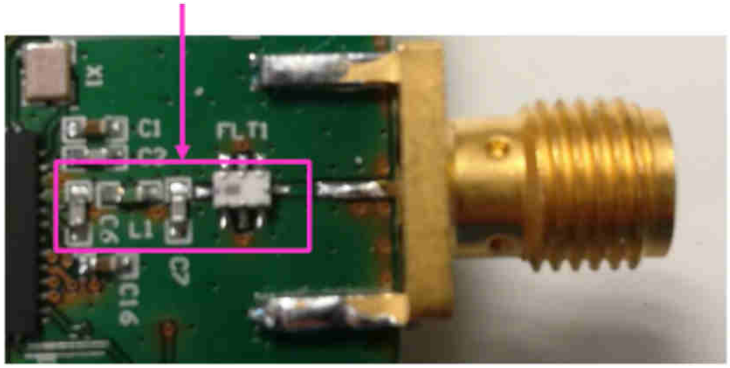

ST IPD Device

Matching Network

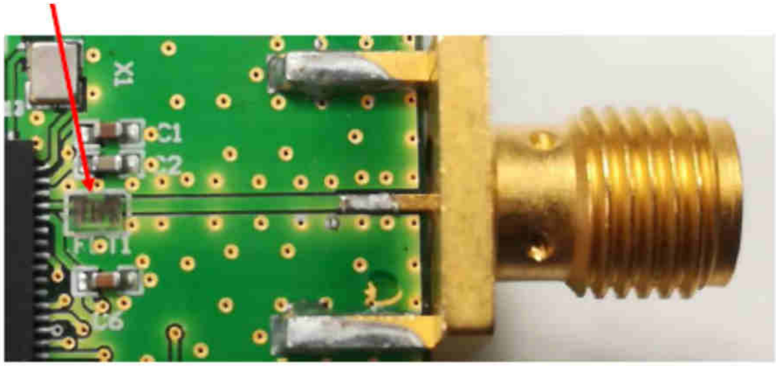
(50Ω)

Filter

Discrete solution

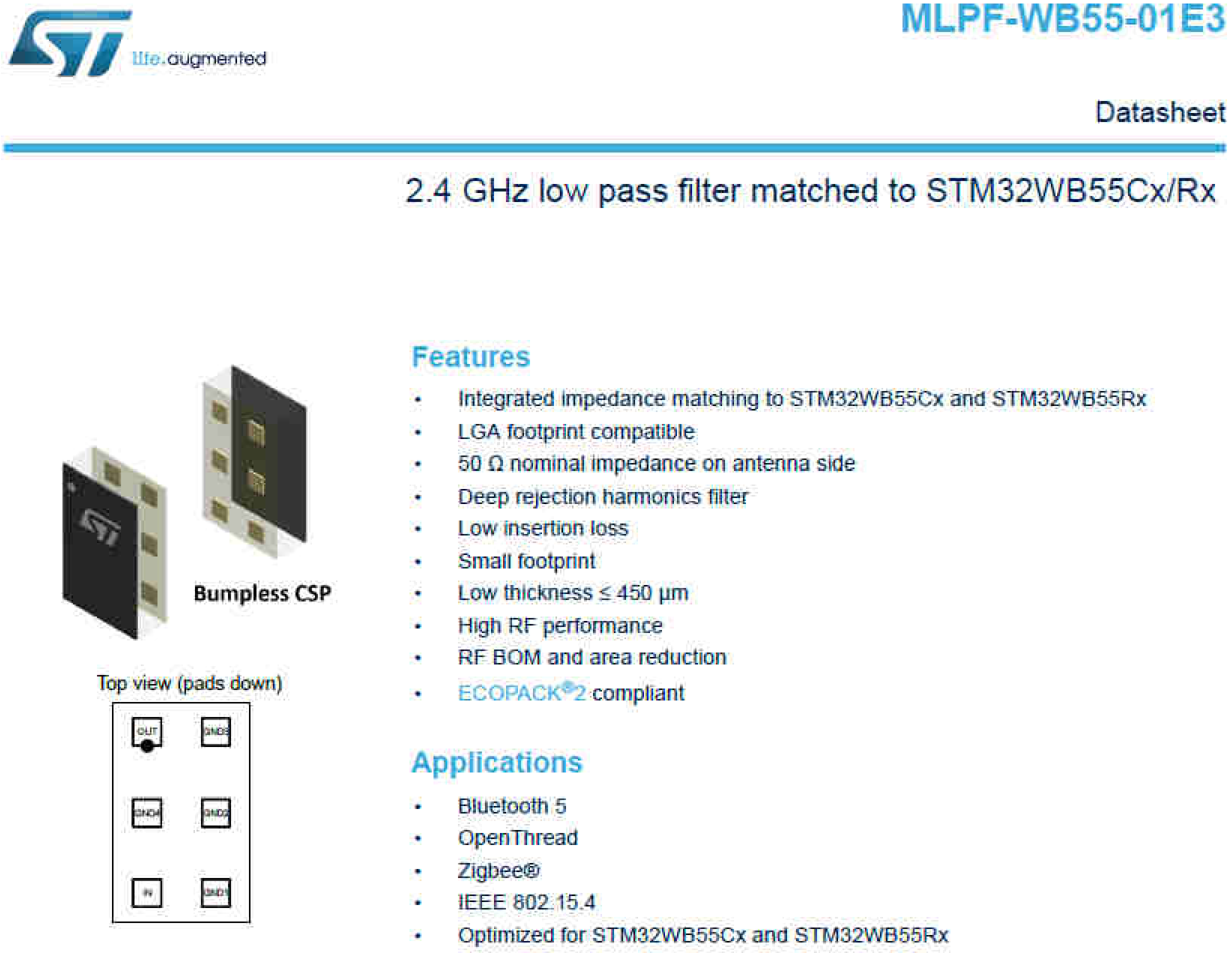


IPD device from ST



IPD Filter

Mass Production NOW



MLPF-WB55-01E3

Datasheet

2.4 GHz low pass filter matched to STM32WB55Cx/Rx

Bumpless CSP

Top view (pads down)

OUT
GND3
GND4
GND2
IN
GND1

Features

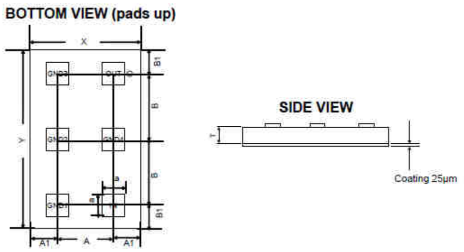
- Integrated impedance matching to STM32WB55Cx and STM32WB55Rx
- LGA footprint compatible
- 50 Ω nominal impedance on antenna side
- Deep rejection harmonics filter
- Low insertion loss
- Small footprint
- Low thickness ≤ 450 µm
- High RF performance
- RF BOM and area reduction
- ECOPACK®2 compliant

Applications

- Bluetooth 5
- OpenThread
- Zigbee®
- IEEE 802.15.4
- Optimized for STM32WB55Cx and STM32WB55Rx

IPD Filter

1mm x 1.6mm CSP



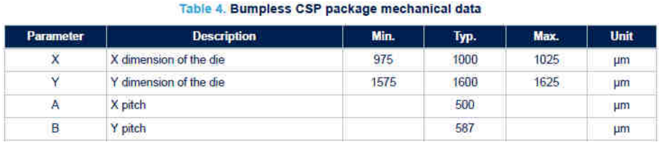
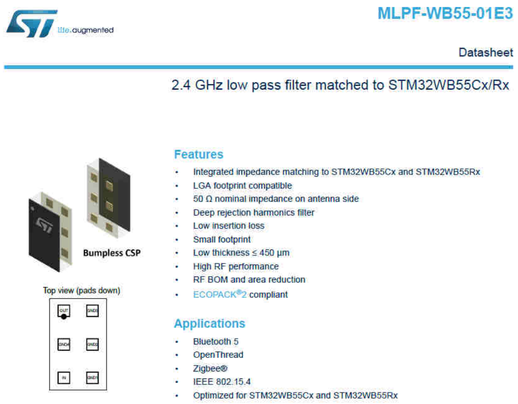


Table 4. Bumpless CSP package mechanical data

Parameter	Description	Min.	Typ.	Max.	Unit
X	X dimension of the die	975	1000	1025	µm
Y	Y dimension of the die	1575	1600	1625	µm
A	X pitch		500		µm
B	Y pitch		587		µm



MLPF-WB55-01E3

Datasheet

2.4 GHz low pass filter matched to STM32WB55Cx/Rx

Features

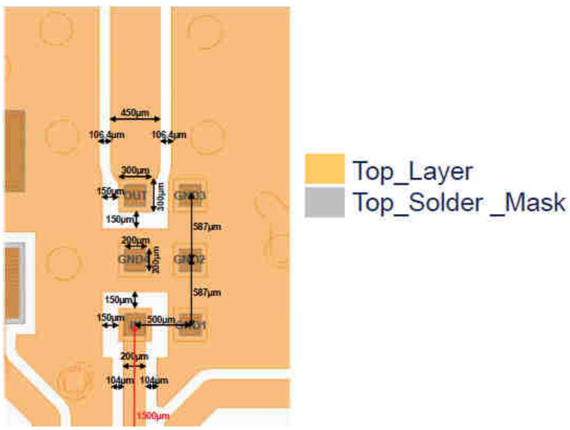
- Integrated impedance matching to STM32WB55Cx and STM32WB55Rx
- LGA footprint compatible
- 50 Ω nominal impedance on antenna side
- Deep rejection harmonics filter
- Low insertion loss
- Small footprint
- Low thickness ≤ 450 µm
- High RF performance
- RF BOM and area reduction
- ECOPACK®2 compliant

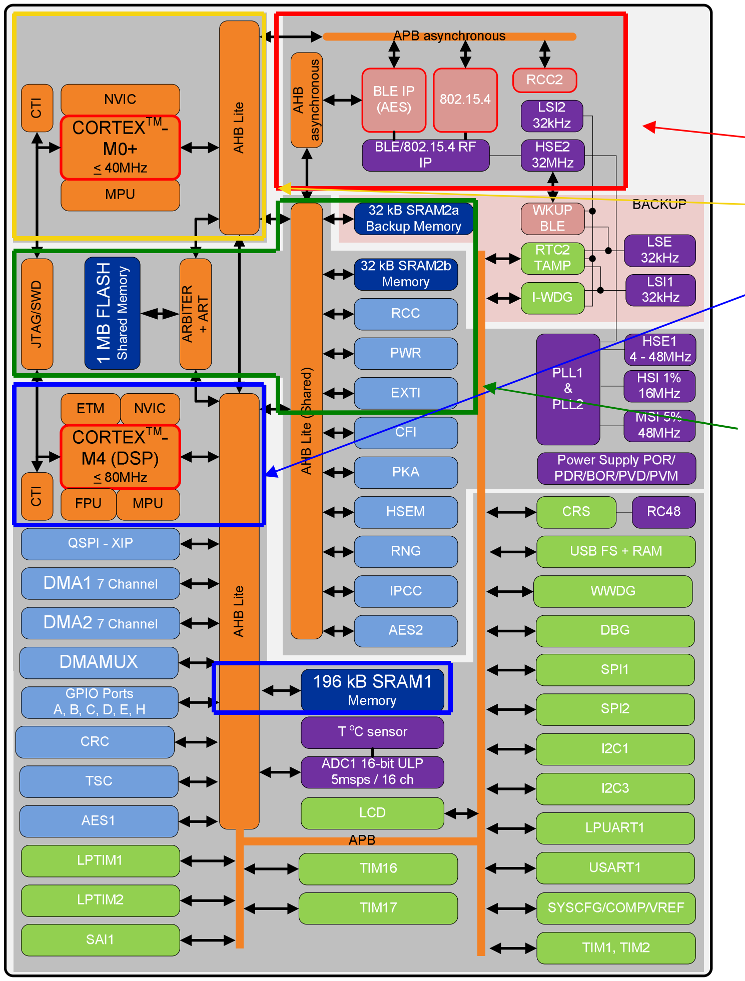
Applications

- Bluetooth 5
- OpenThread
- Zigbee®
- IEEE 802.15.4
- Optimized for STM32WB55Cx and STM32WB55Rx

PCB recommendations
included in datasheet

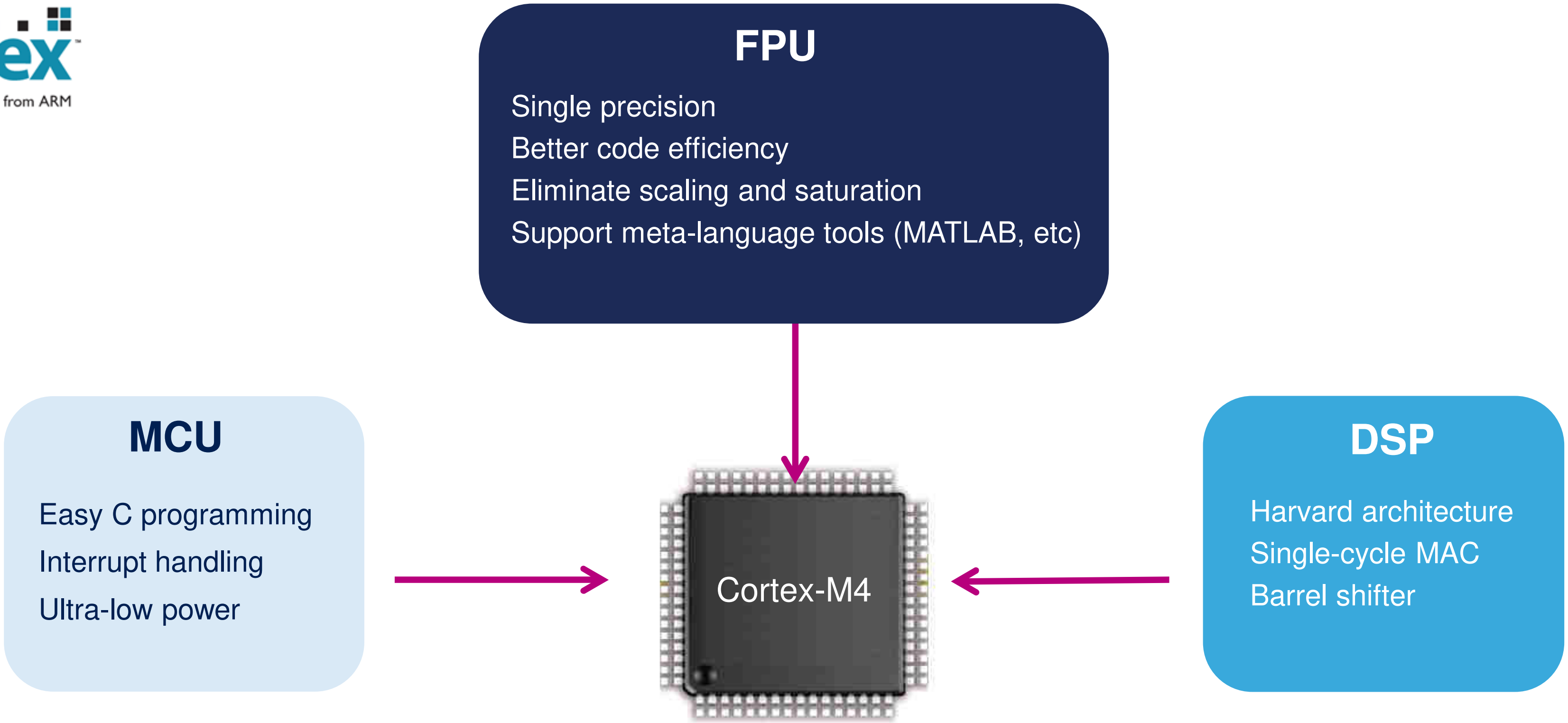
Figure 13. PCB land pattern recommendations





- 3 autonomous sub-systems
  - Radio sub-system
  - Cortex-M0+ (CPU2)
  - Cortex-M4 (CPU1)
- Common run domain
  - Flash, SRAM2, RCC, PWR, EXTI

ARM® Cortex® M4 Core



FPU

Single precision
Better code efficiency
Eliminate scaling and saturation
Support meta-language tools (MATLAB, etc)

MCU

Easy C programming
Interrupt handling
Ultra-low power

DSP

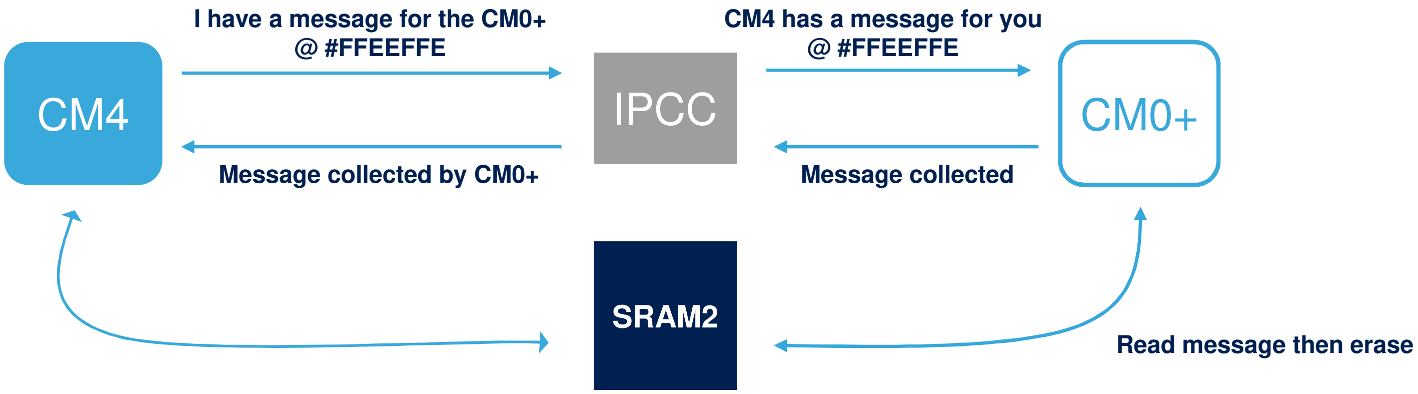
Harvard architecture
Single-cycle MAC
Barrel shifter

http://www.arm.com/products/processors/cortex-m/cortex-m4-processor.php

Dual core – How does that work?

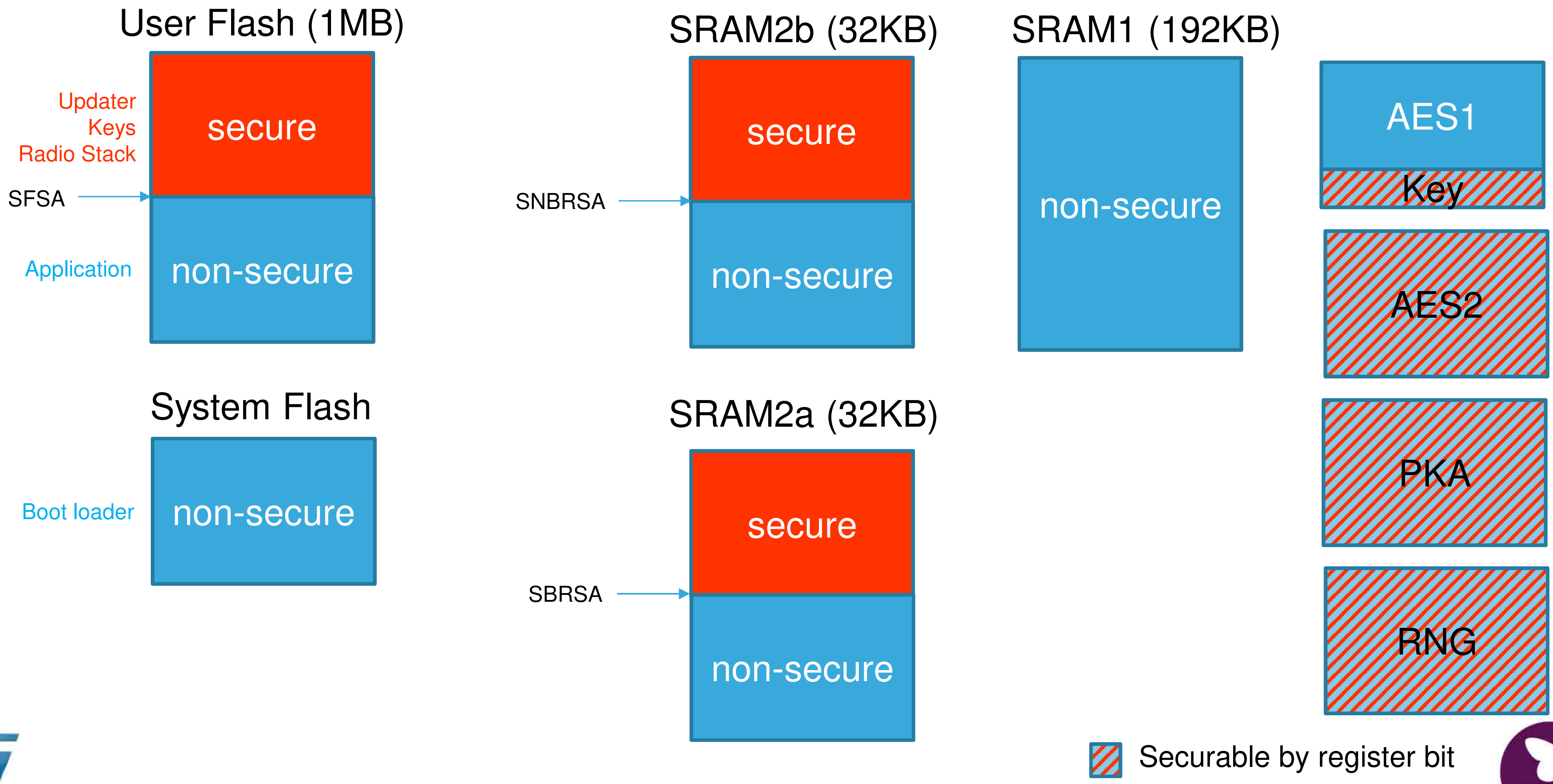
IPCC: Inter Processor Communication Controller

HSEM: Hardware Semaphore – prevent shared resource access conflicts


I have a message for the CM0+
@ #FFEEFFE
CM4 has a message for you
@ #FFEEFFE
CM4
IPCC
CM0+
Message collected by CM0+
Message collected
SRAM2
Read message then erase

IPCC works in both directions

Memory Partitioning



User Flash (1MB)

Updater
Keys
Radio Stack

secure

SFSA

Application

non-secure

System Flash

Boot loader

non-secure

SRAM2b (32KB)

secure

SNBRSA

non-secure

SRAM2a (32KB)

secure

SBRSA

non-secure

SRAM1 (192KB)

non-secure

AES1

Key

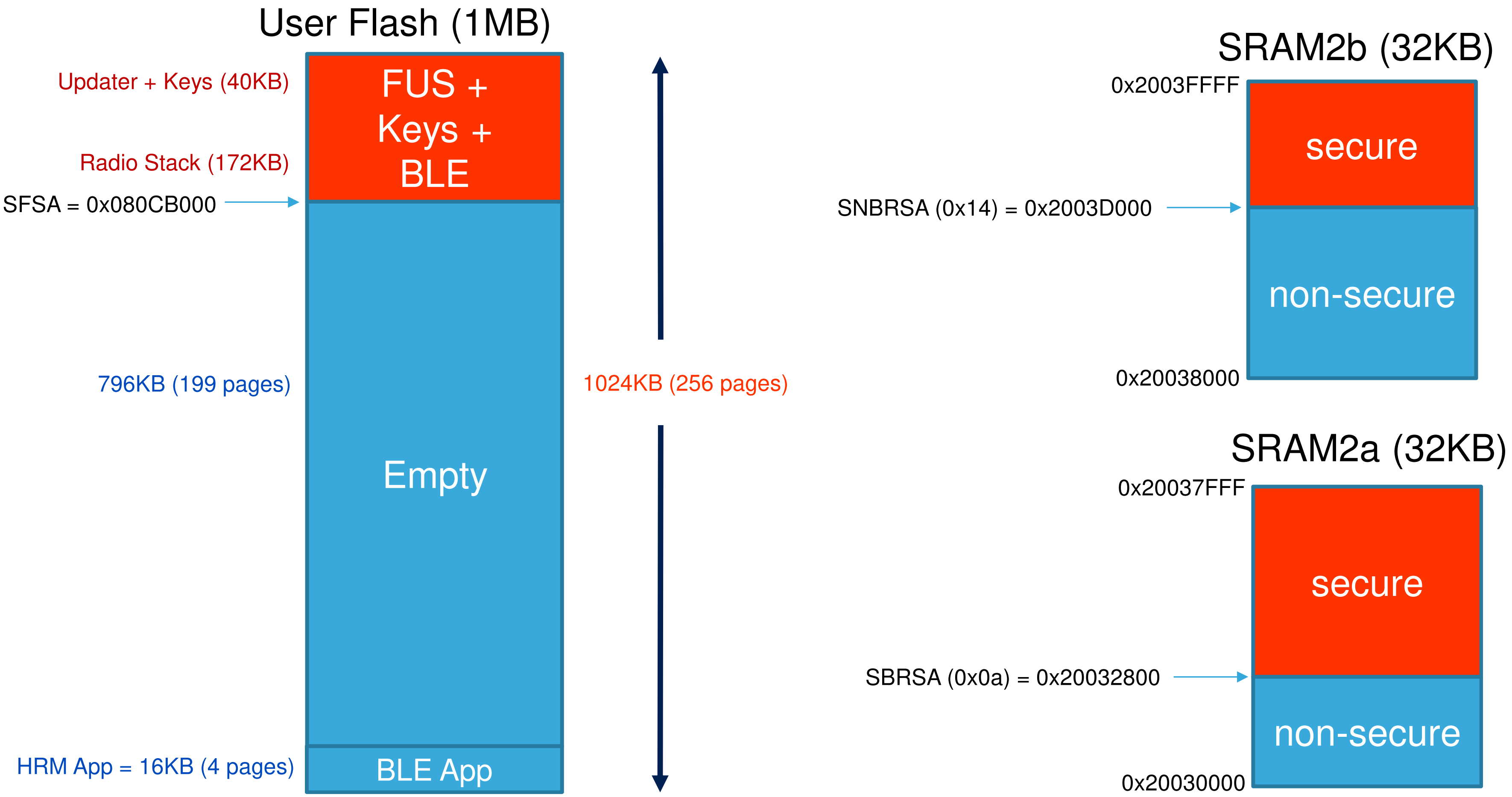
AES2

PKA

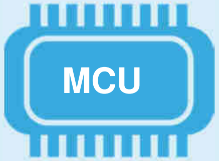
RNG

▨ Securable by register bit

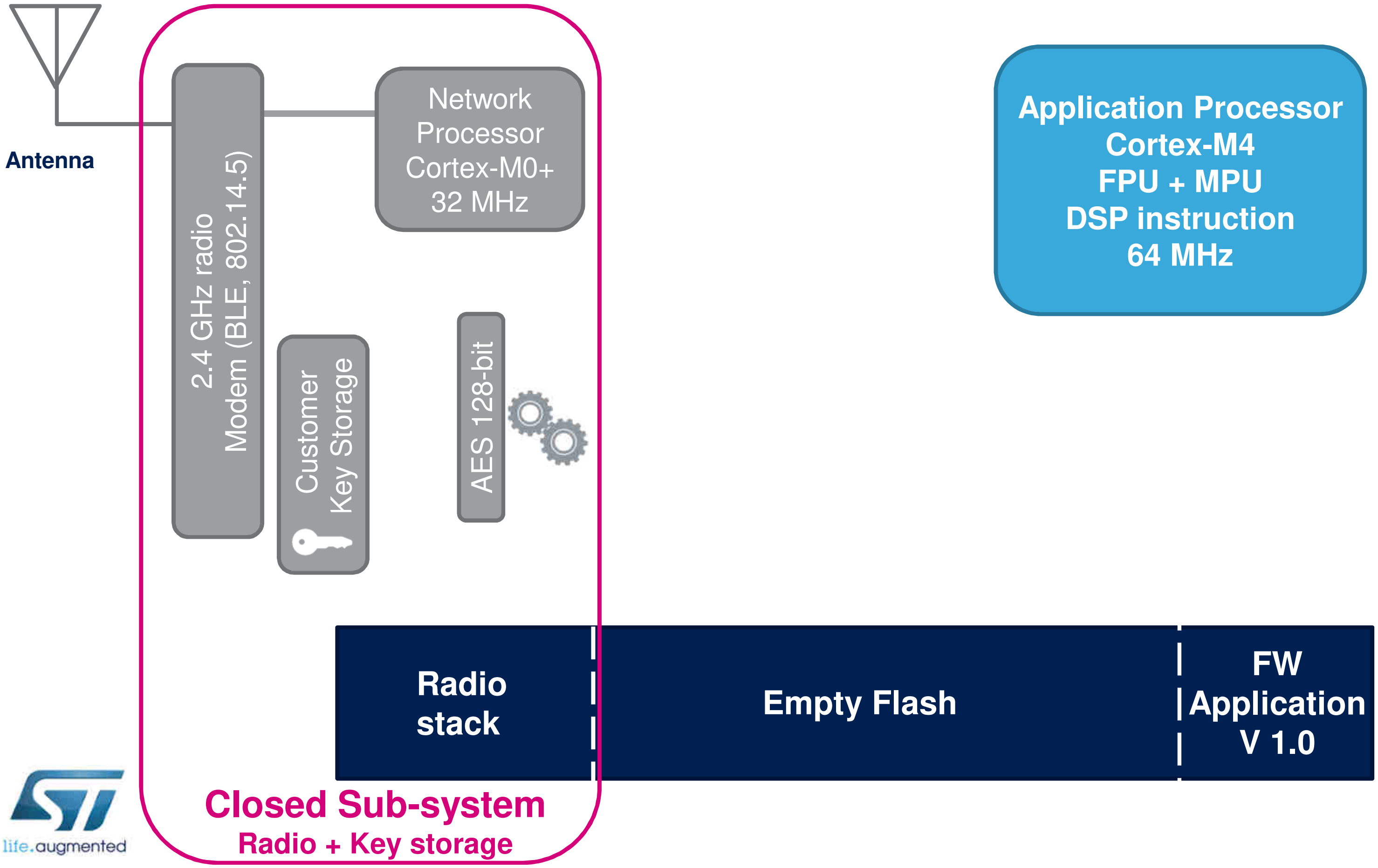
Memory Partitioning: BLE Stack



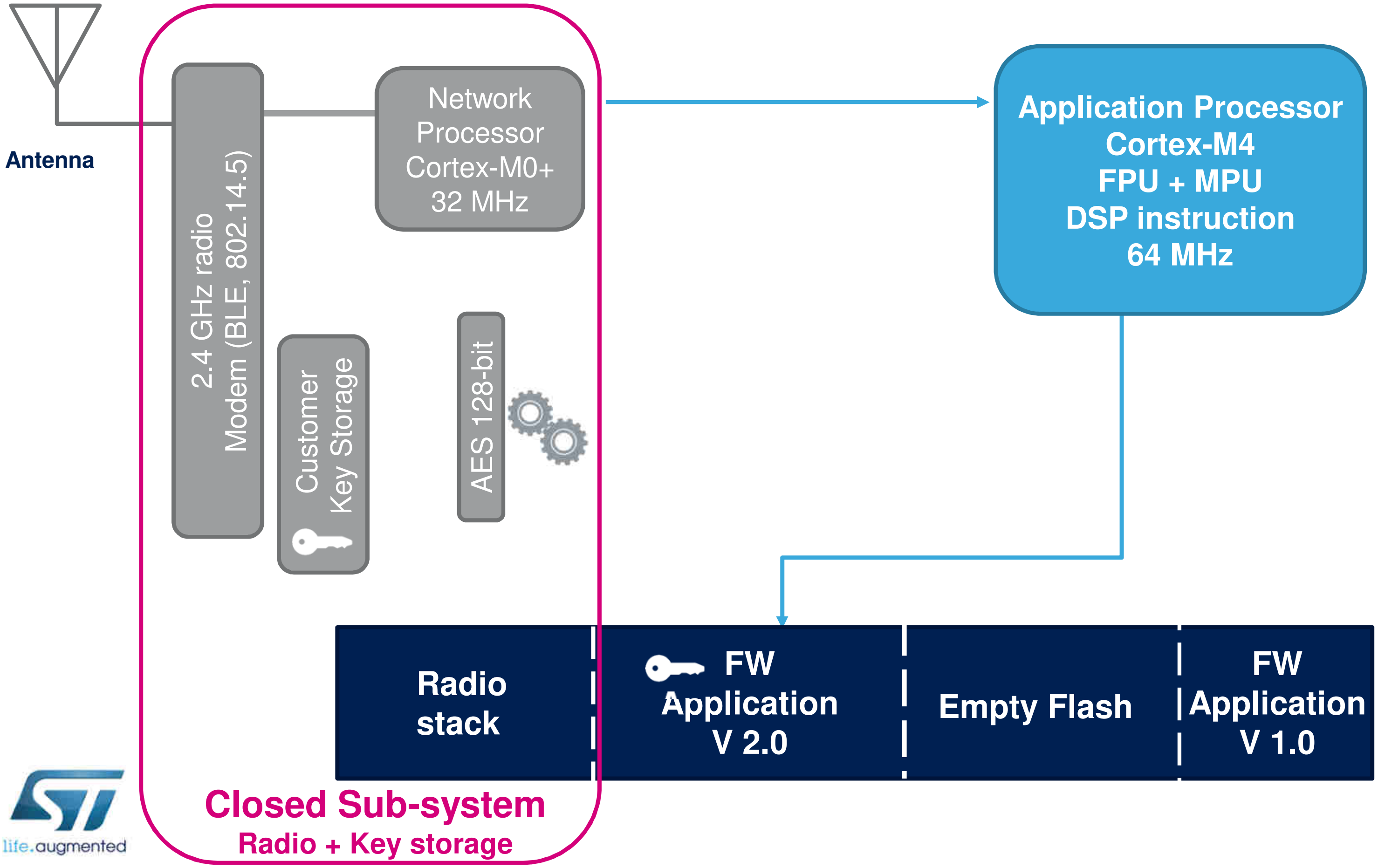
IoT Protection Ready

Attacks	Attacks description	STM32WB Countermeasures
Non Invasive  	- Environment - Temp / Voltage / Clocks - Fault injection - Exploit debugger - Side channel - Power Analysis	- Temp sensor - Power supply monitor - Clock security system - Tamper pads - ECC, Parity check - SRAM mass erase - Read out protection - Flash-only boot
Software 	- Break the encryption - Extract keys - Exploit debugger / test modes - Malware - Replay	- Customer Key Storage - RNG, Crypto accelerator, CRC - Readout / Write memory protections - Memory Protection Unit - Root Security Service - Secure Firmware Update (SFU) - 96-bit Unique ID

OTA Update Scenario

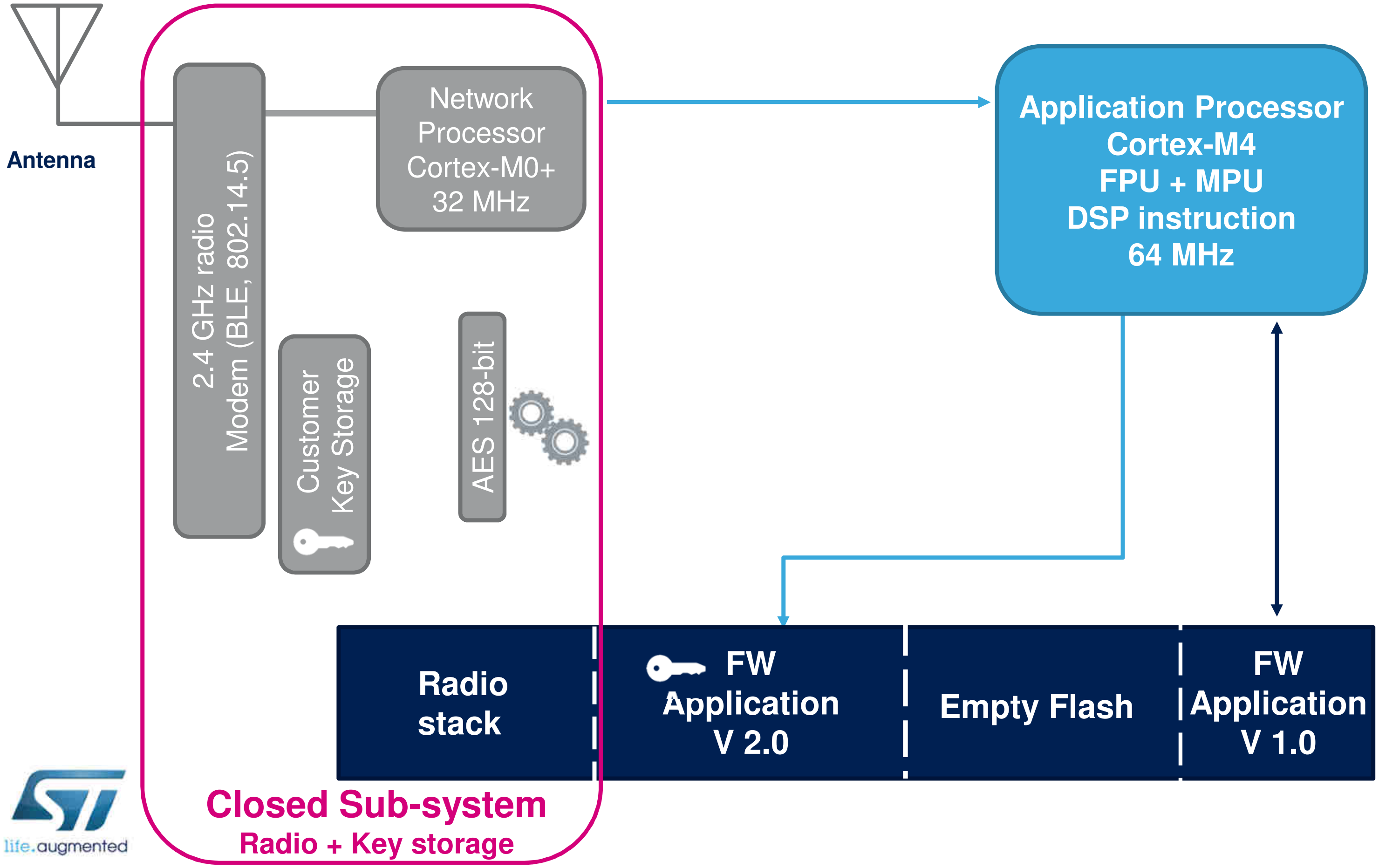


OTA Update Scenario



1 New FW package received

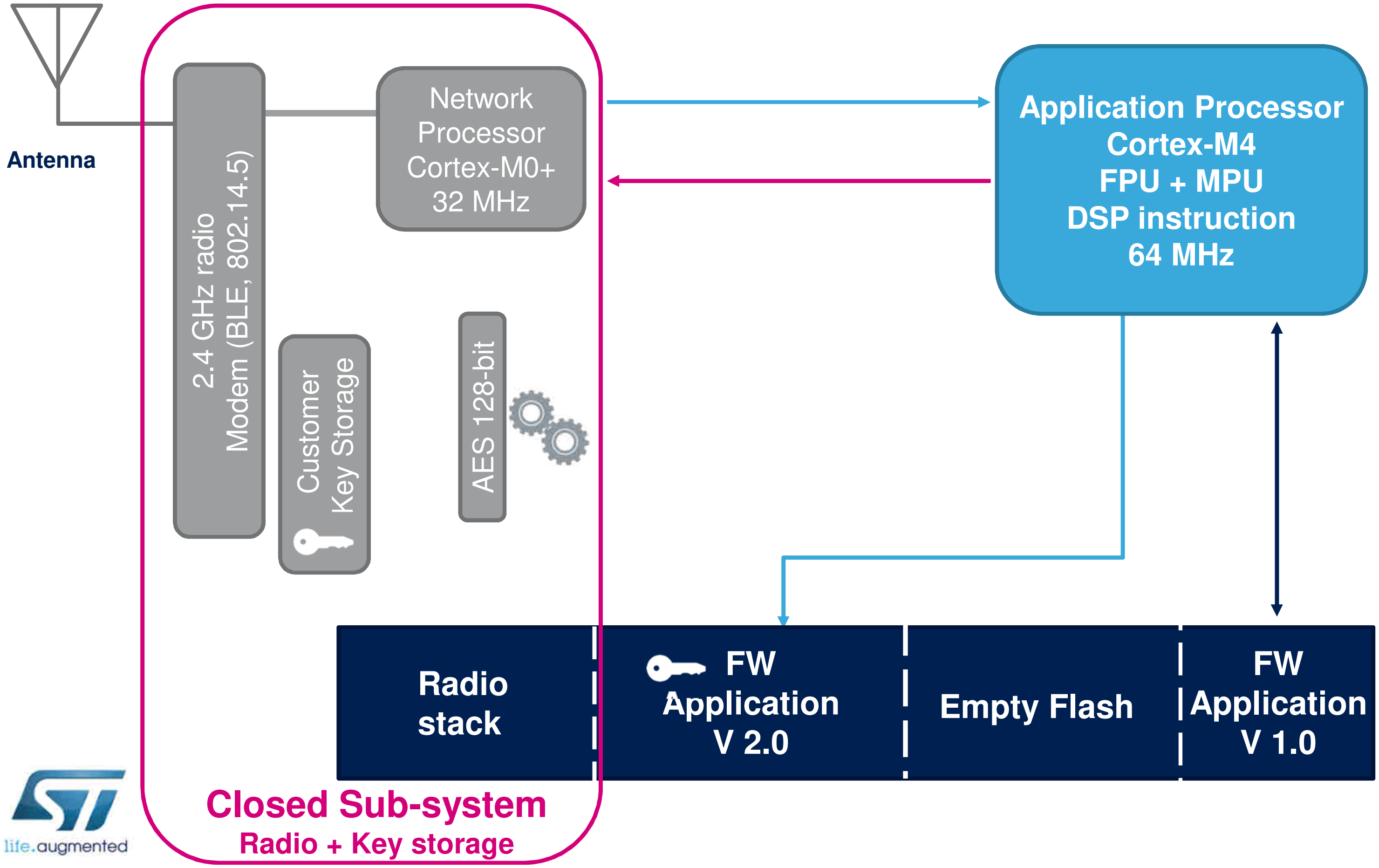
OTA Update Scenario



1 New FW package received

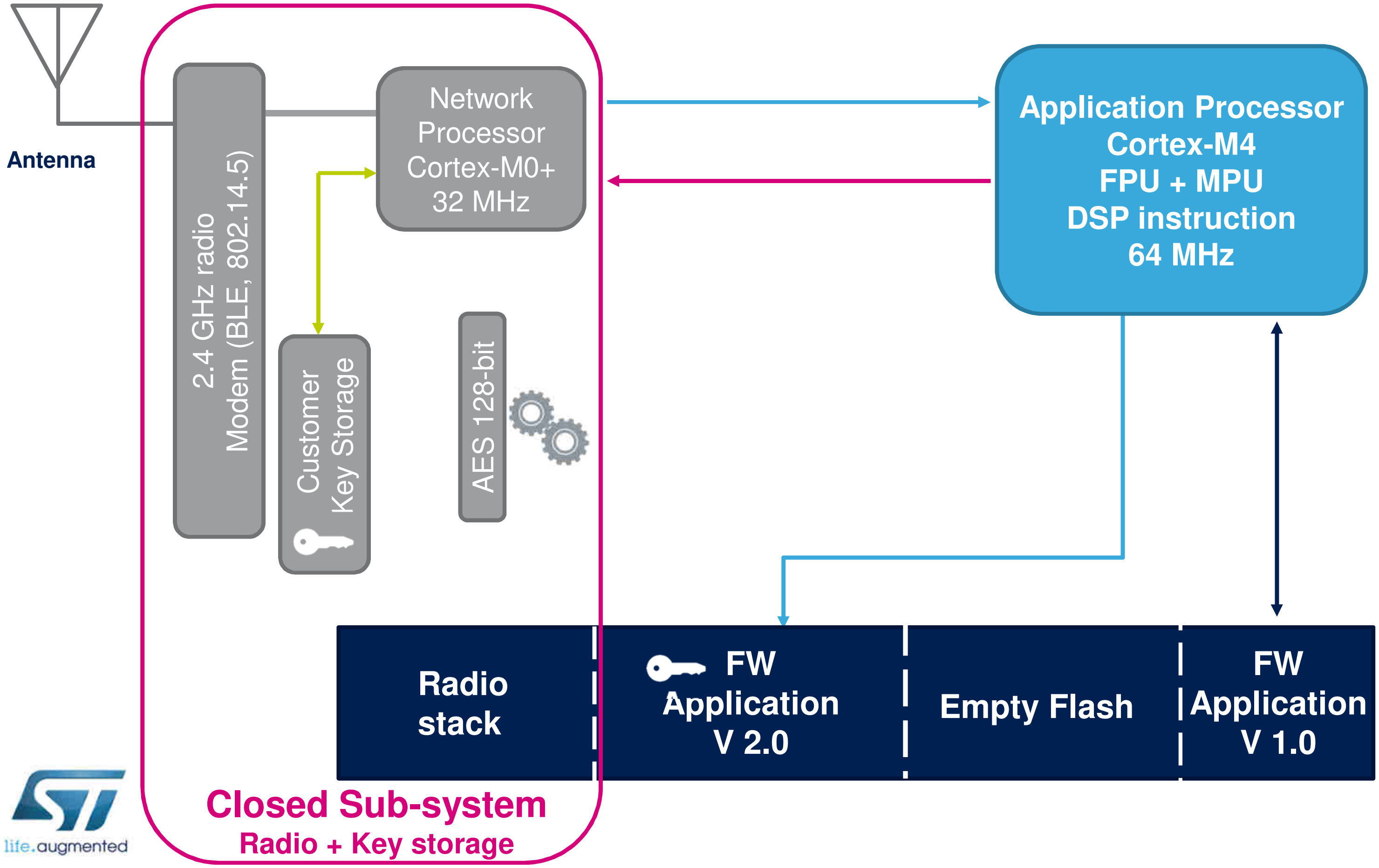
2 New FW detected
Update is launched

OTA Update Scenario



1. New FW package received
2. New FW detected
   Update is launched
3. App Processor send New
   FW package signature and
   encryption key for authentication

OTA Update Scenario



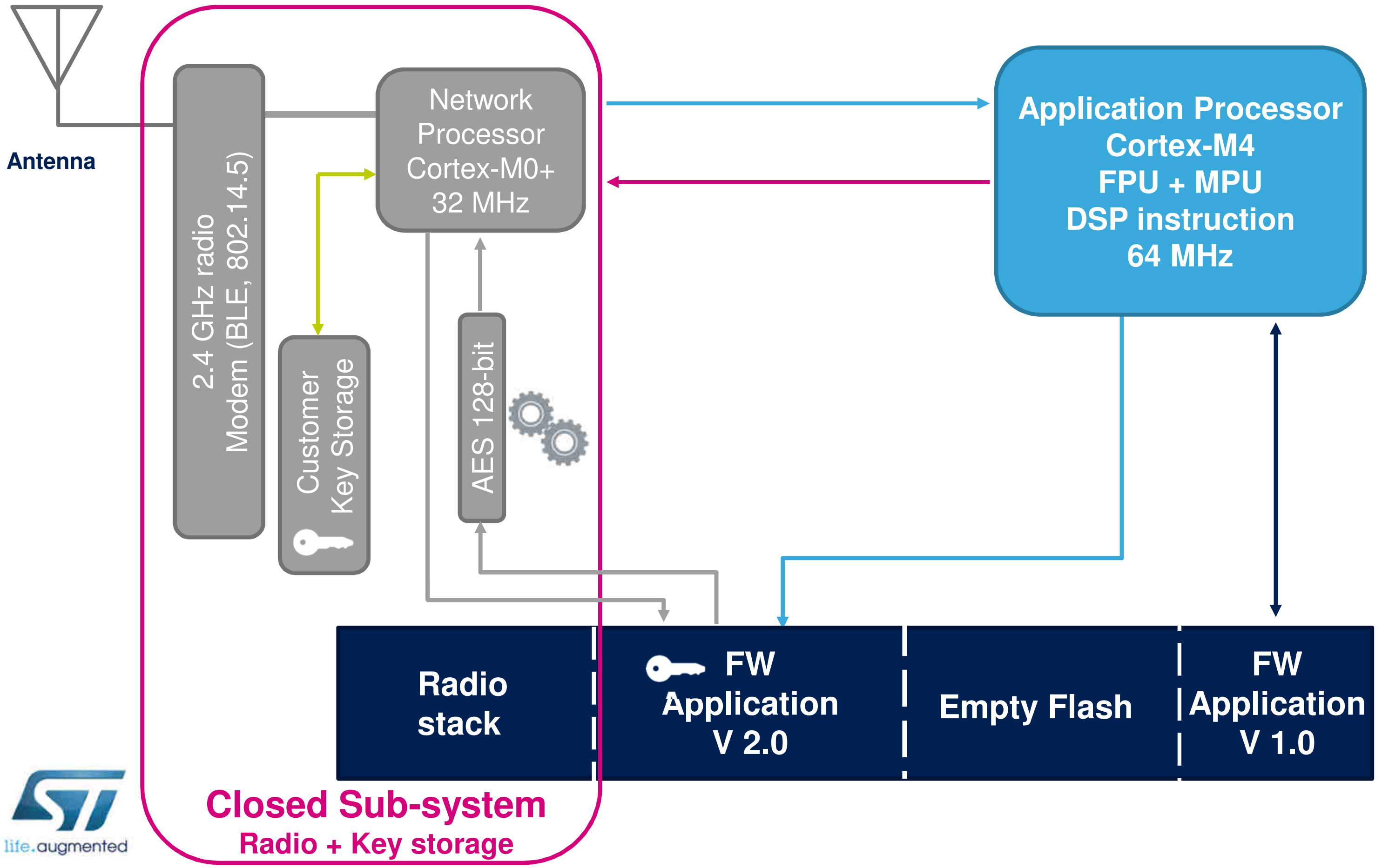
1 New FW package received

2 New FW detected
Update is launched

3 App Processor send New
FW package signature and
encryption key for authentication

4 Authentication signature
matches preprogrammed key
if not, the process is
aborted and device resets

OTA Update Scenario



1 New FW package received

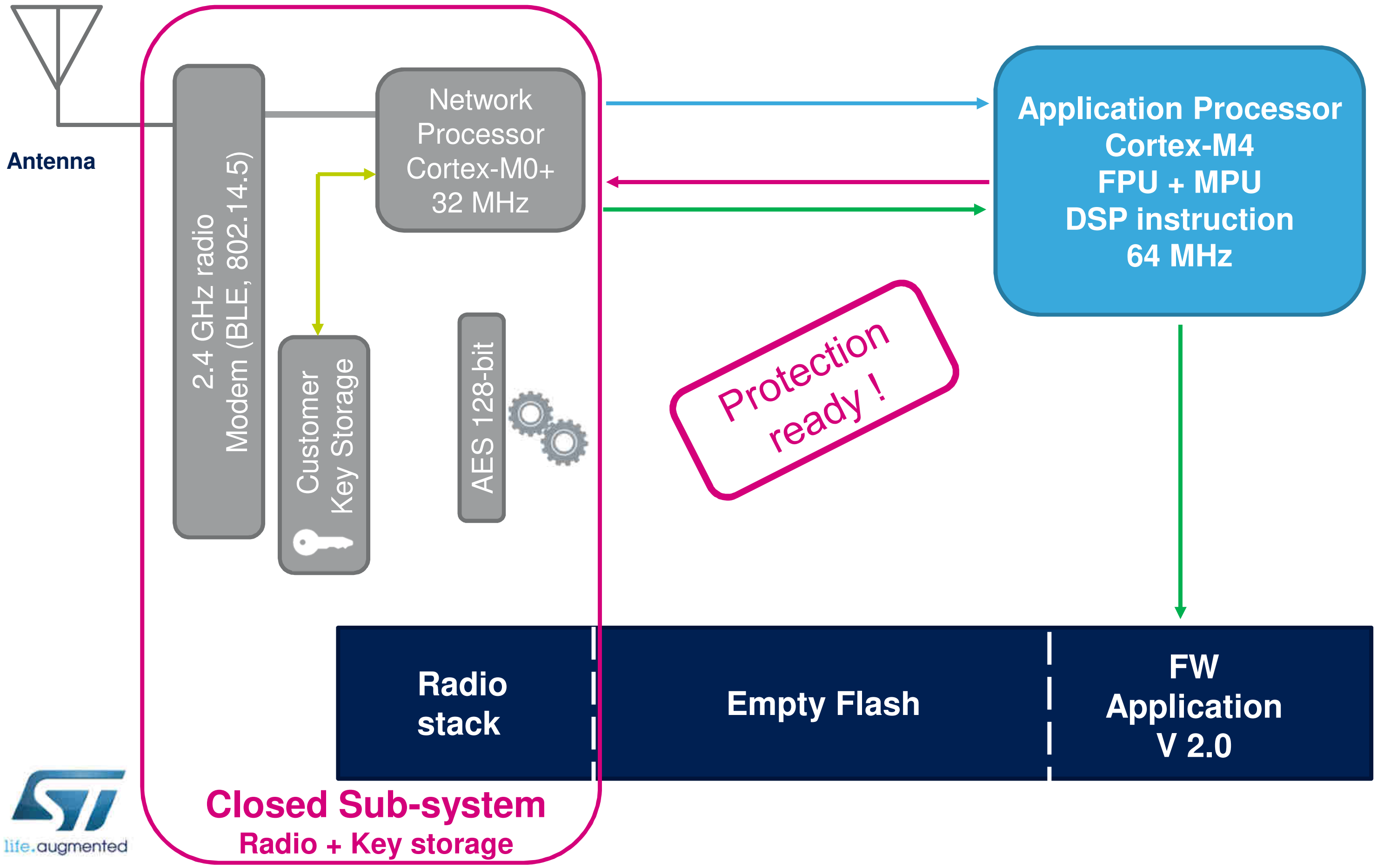
2 New FW detected
Update is launched

3 App Processor send New FW package signature and encryption key for authentication

4 Authentication signature matches preprogrammed key if not, the process is aborted and device resets

5 New FW package is decrypted with proprietary Key.

OTA Update Scenario



1. New FW package received
2. New FW detected
Update is launched
3. App Processor send New FW package signature and encryption key for authentication
4. Authentication signature matches preprogrammed key if not, the process is aborted and device resets
5. New FW package is decrypted with proprietary Key.
6. New Firmware replaces older firmware device resets.

AN5156 – Security

AN5156 is a deep-dive into many security topics, some common and some WB-specific

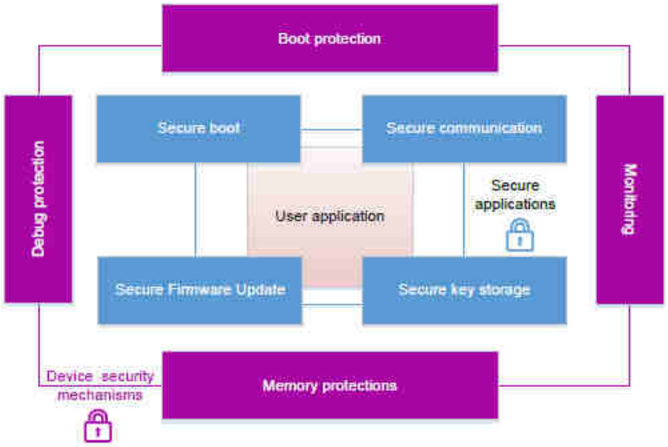


Table 4. Attacks types and costs

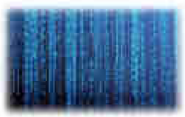
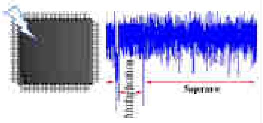
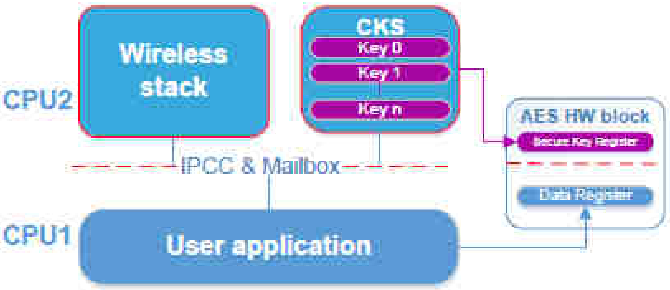
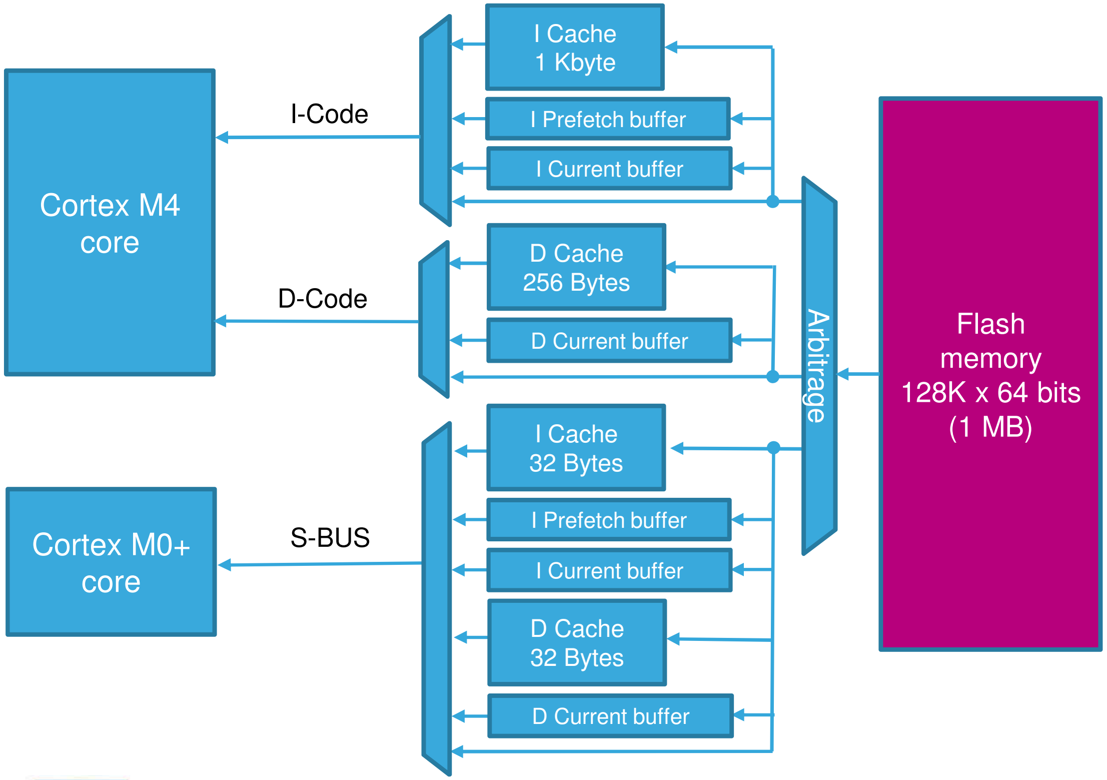
Attacks types	Software	Hardware non–invasive	Hardware invasive
			
Scope	Remote or local	Local board and device level	Local device level
Technics	Software bugs Protocol weaknesses Trojan horse Eavesdropping…	Debug port Power Glitches Fault injection Side-channels analysis…	Probing Laser FIB Reverse engineering…
Cost/ expertise	From very low to high depending on the security failure targeted	Quite low cost. Need only moderately sophisticated equipment and knowledge to implement	Very expensive. Need dedicated/ heavy equipment and very specific skills
Objectives	Access to confidential assets (code and data). Usurpation Denial of service	Access to secret data or device internal behavior (algorithm)	Reverse engineering of the device (silicon intellectual property) Access to hidden hardware and software secrets (Flash access)

Figure 13. Dual-core architecture with CKS service

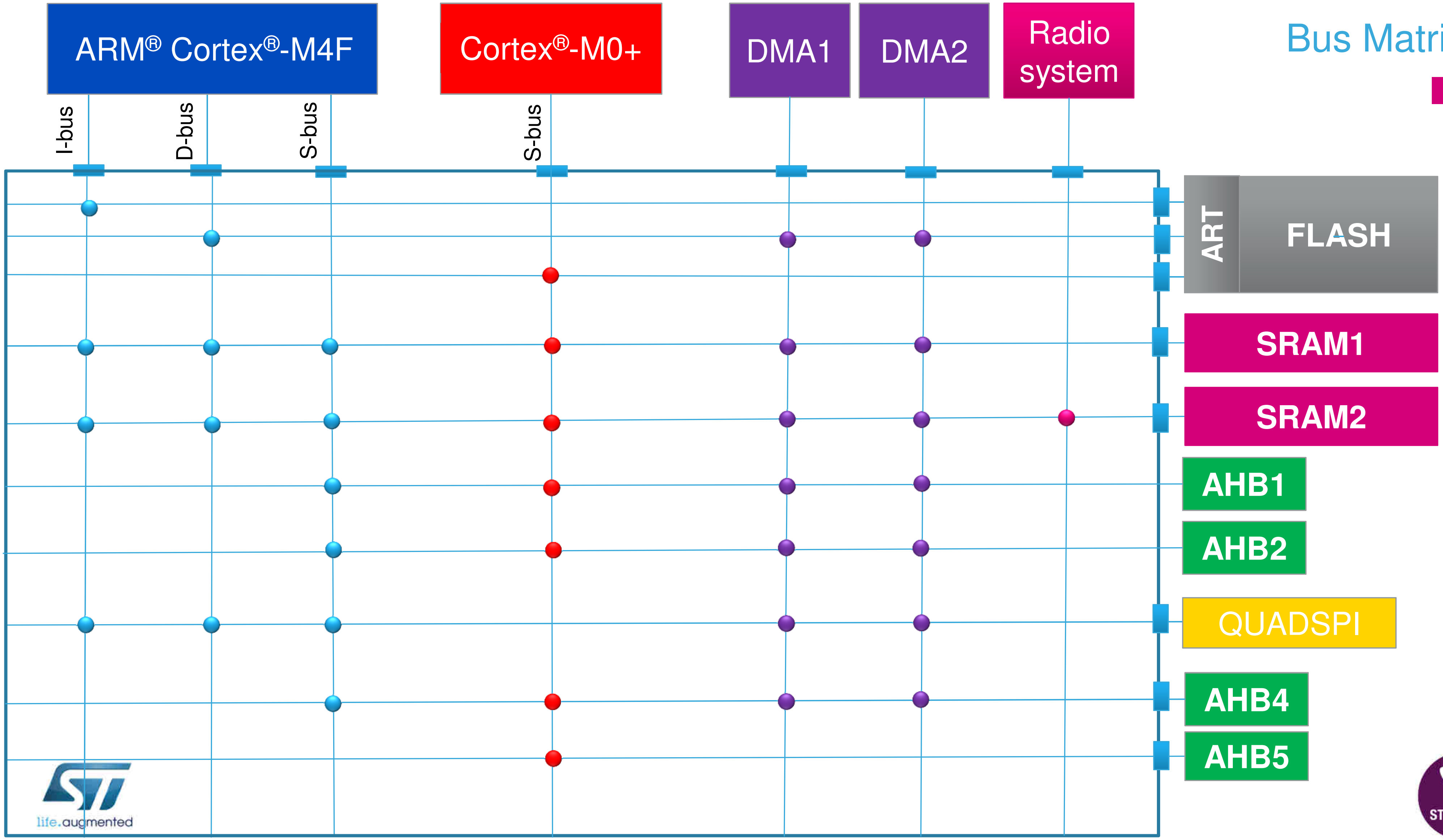


ART Accelerator™



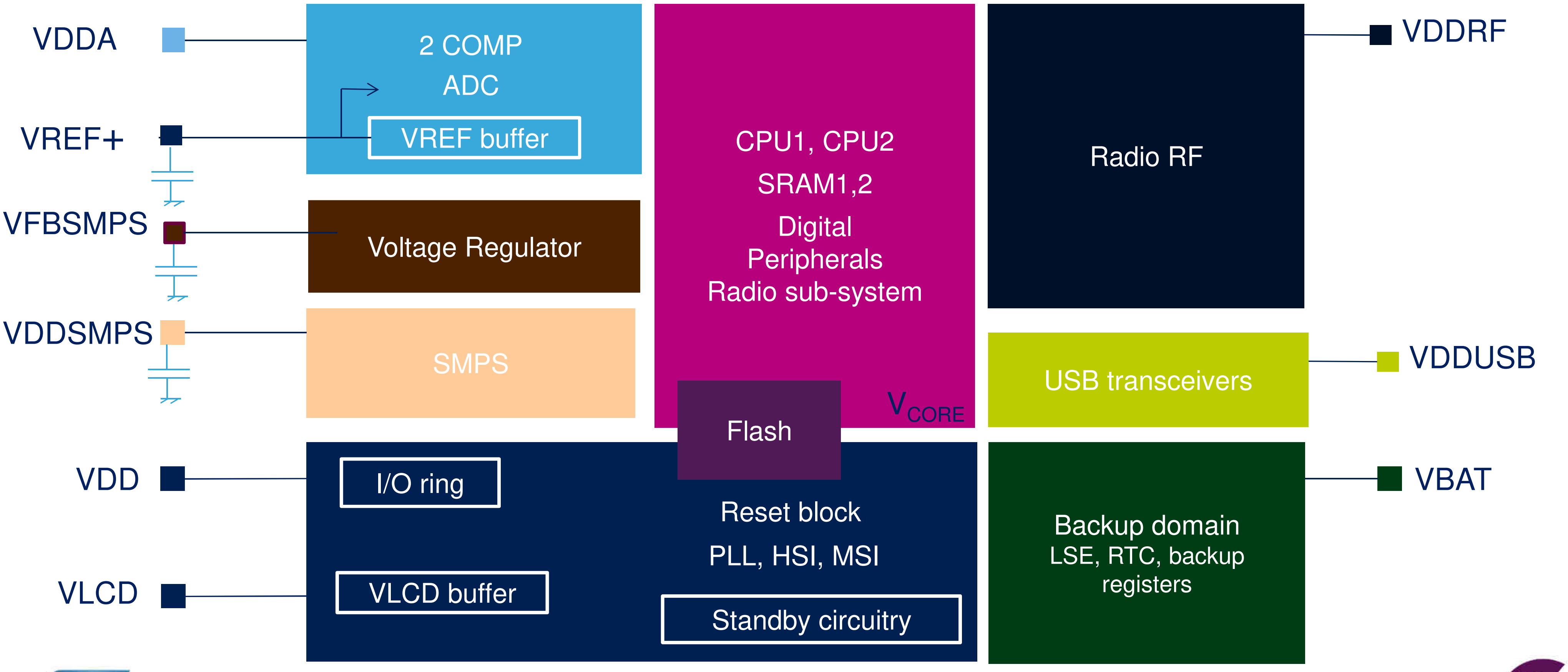
- **Cortex-M4**
  - **Instruction cache** = 32 lines of 4x64 bits
  - **Data cache** = 8 lines of 4x64 bits
  - **Pre-fetch buffer**
- **Cortex-M0+**
  - **Instruction cache** = 4 lines of 1x64 bits
  - **Data cache** = 4 lines of 1x64 bits
  - **Pre-fetch buffer**

Bus Matrix

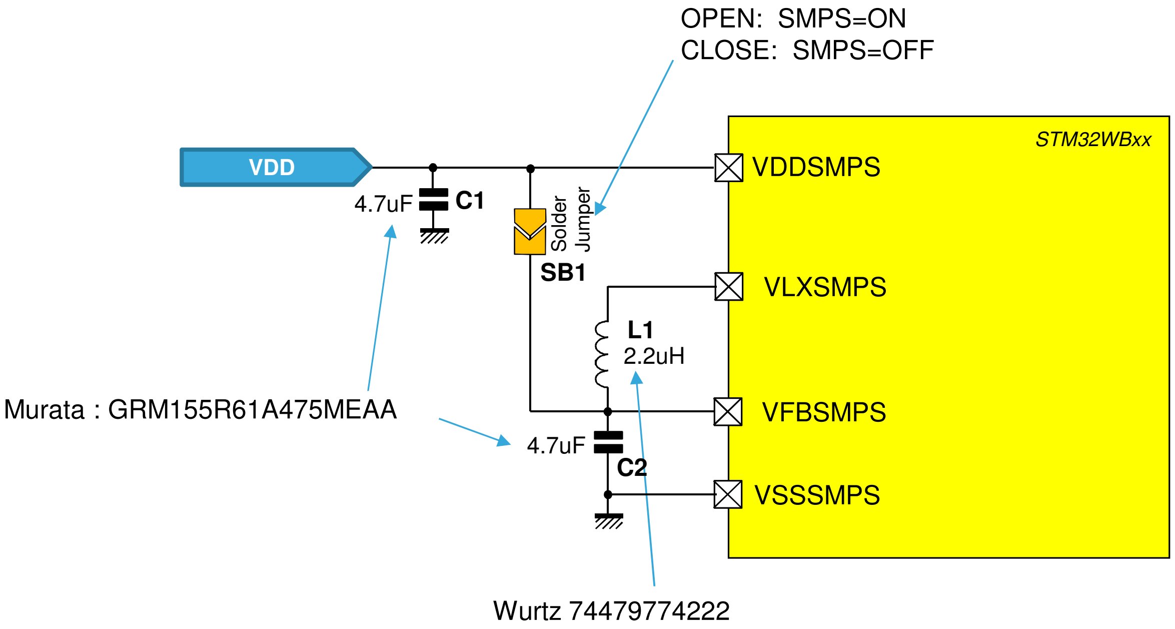


	ARM® Cortex®-M4F I-bus	ARM® Cortex®-M4F D-bus	ARM® Cortex®-M4F S-bus	Cortex®-M0+ S-bus	DMA1	DMA2	Radio system
ART FLASH	●						
ART FLASH		●			●	●	
ART FLASH				●			
SRAM1	●	●	●	●	●	●	
SRAM2	●	●	●	●	●	●	●
AHB1			●	●	●	●	
AHB2			●	●	●	●	
QUADSPI	●	●	●		●	●	
AHB4			●	●	●	●	
AHB5				●			

Power schemes



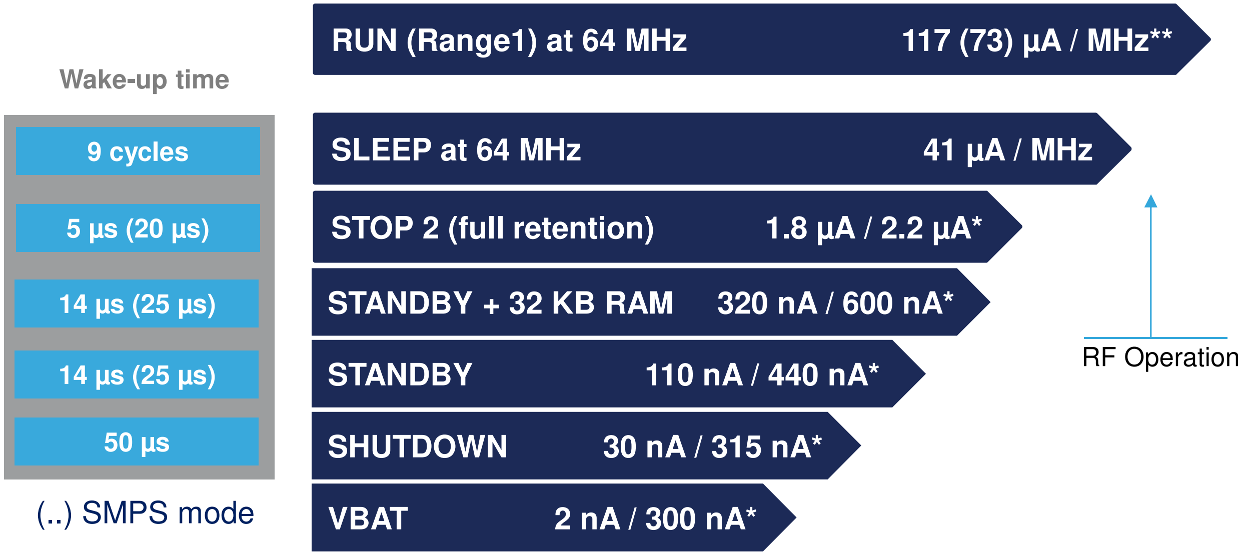
SMPS Schematic



8MHz SMPS configuration

For 4MHZ SMPS configuration change L1 = 4.7µH

FlexPowerControl



Wake-up time	Mode	Consumption
	RUN (Range1) at 64 MHz	117 (73) µA / MHz**
9 cycles	SLEEP at 64 MHz	41 µA / MHz
5 µs (20 µs)	STOP 2 (full retention)	1.8 µA / 2.2 µA*
14 µs (25 µs)	STANDBY + 32 KB RAM	320 nA / 600 nA*
14 µs (25 µs)	STANDBY	110 nA / 440 nA*
50 µs	SHUTDOWN	30 nA / 315 nA*
	VBAT	2 nA / 300 nA*

(..) SMPS mode

Typ @ VDD =1.8 V @ 25 °C

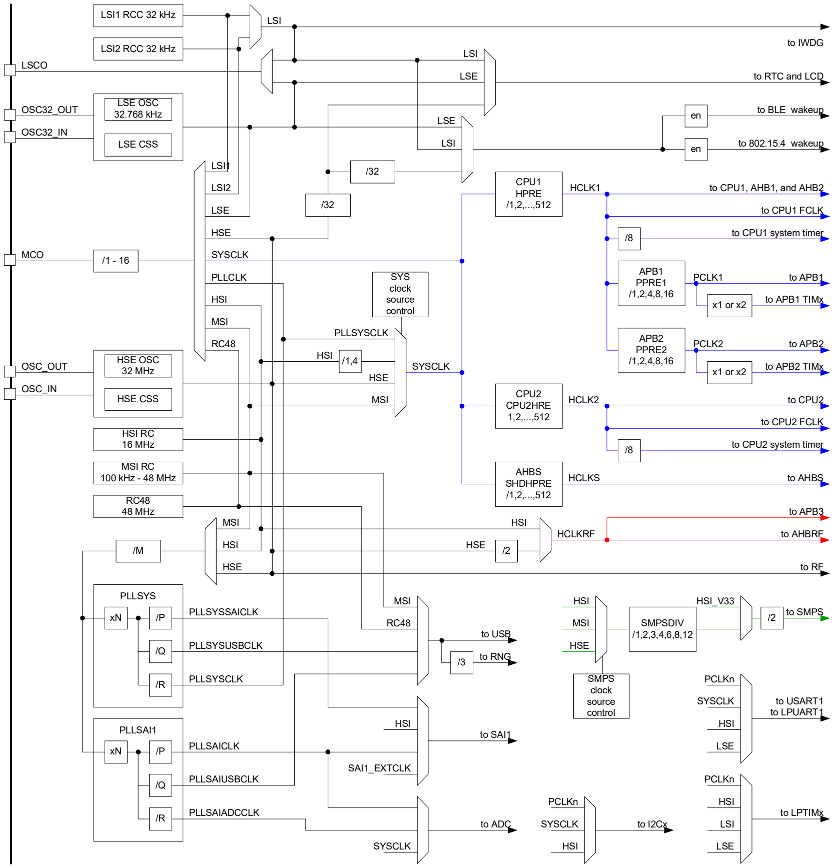
** with RTC*
*** from SRAM1*

- High performance
  - ➔ CoreMark score = 215
- Outstanding power efficiency
  - ➔ ULPBbench score = 175

Clock tree

HSE (32MHz) required for radio operation

LSE (32.768KHz) required for most BLE applications

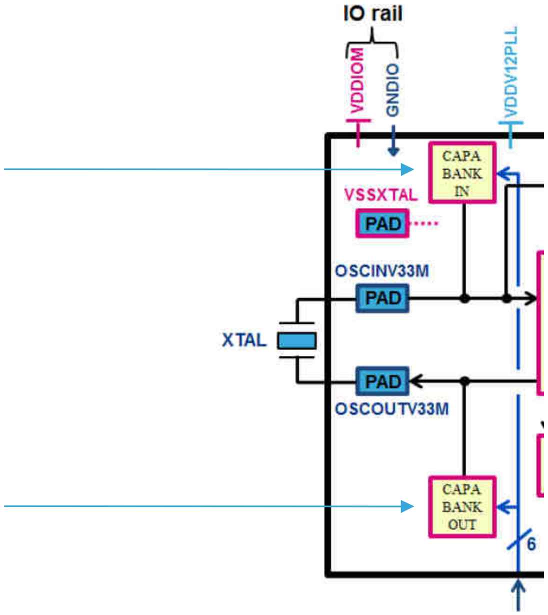


life.augmented

STM32 WB

HSE Trimming

- BLE requires very accurate 32 MHz clock
- Frequency can vary
  - Manufacturing process variations
  - Crystal used
  - PCB design
- Integrated load capacitor bank
  - 64 values for fine tuning
  - MCO clock output pin used for measurement at factory test
  - Stored in OTP
- **No need for external capacitance**
- AN5042 provides details

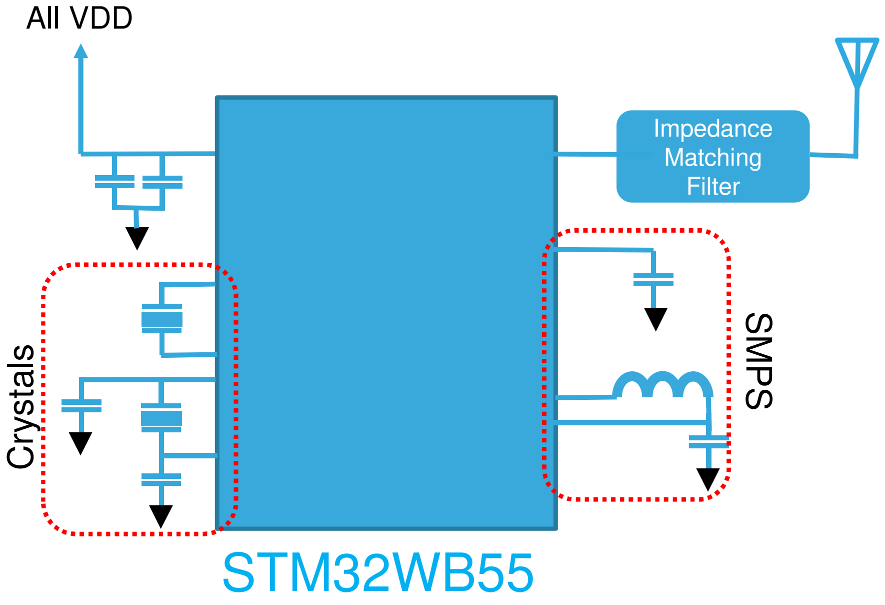


AN5042
Application note

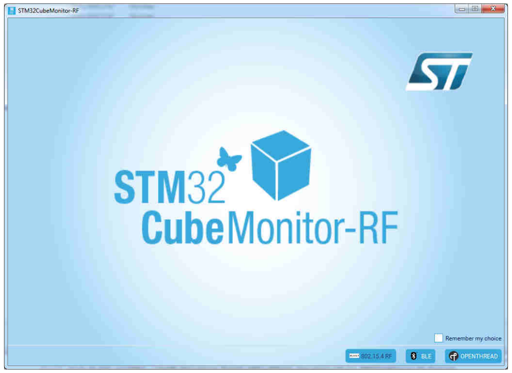
HSE trimming for RF applications using the STM32WB Series

Silicon Cost Savings

- Embedded RF balun
- Single IPD from ST
- Simple SMPS circuit
- Integrated HSE crystal tuning caps
- Minimal passives needed
- Simple 2 layer PCB design



Simplified Schematic Diagram



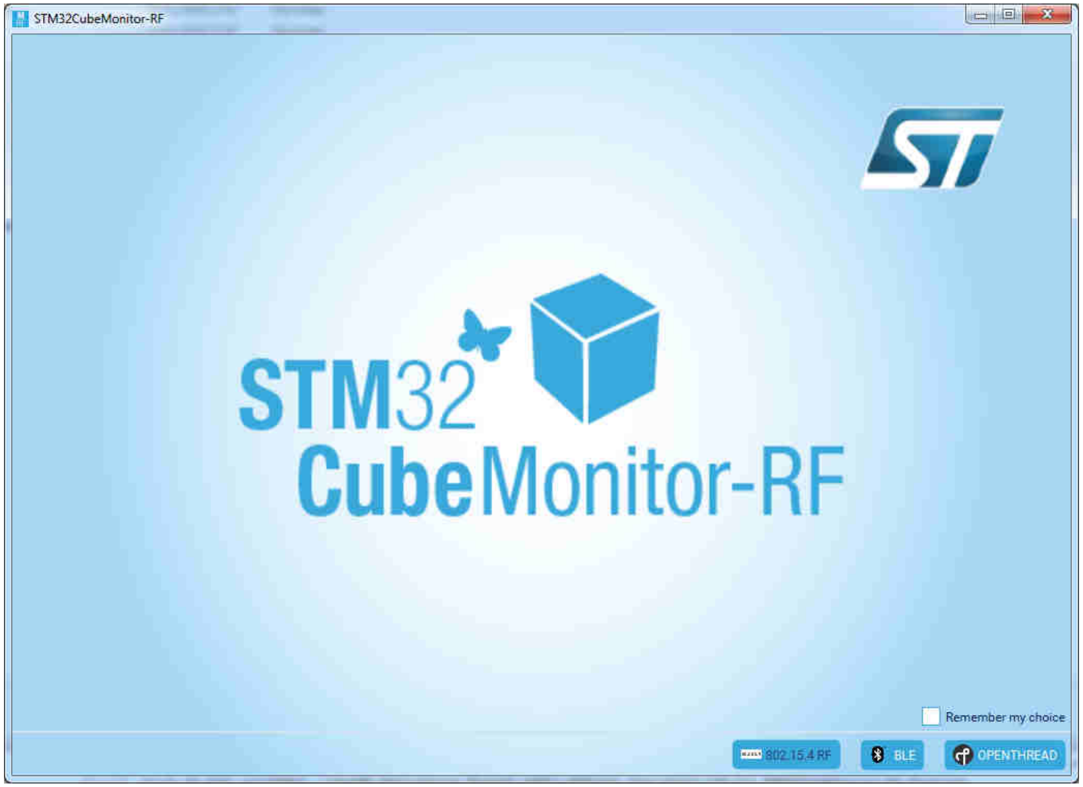
□ Remember my choice

Hands-On

CubeMonitorRF

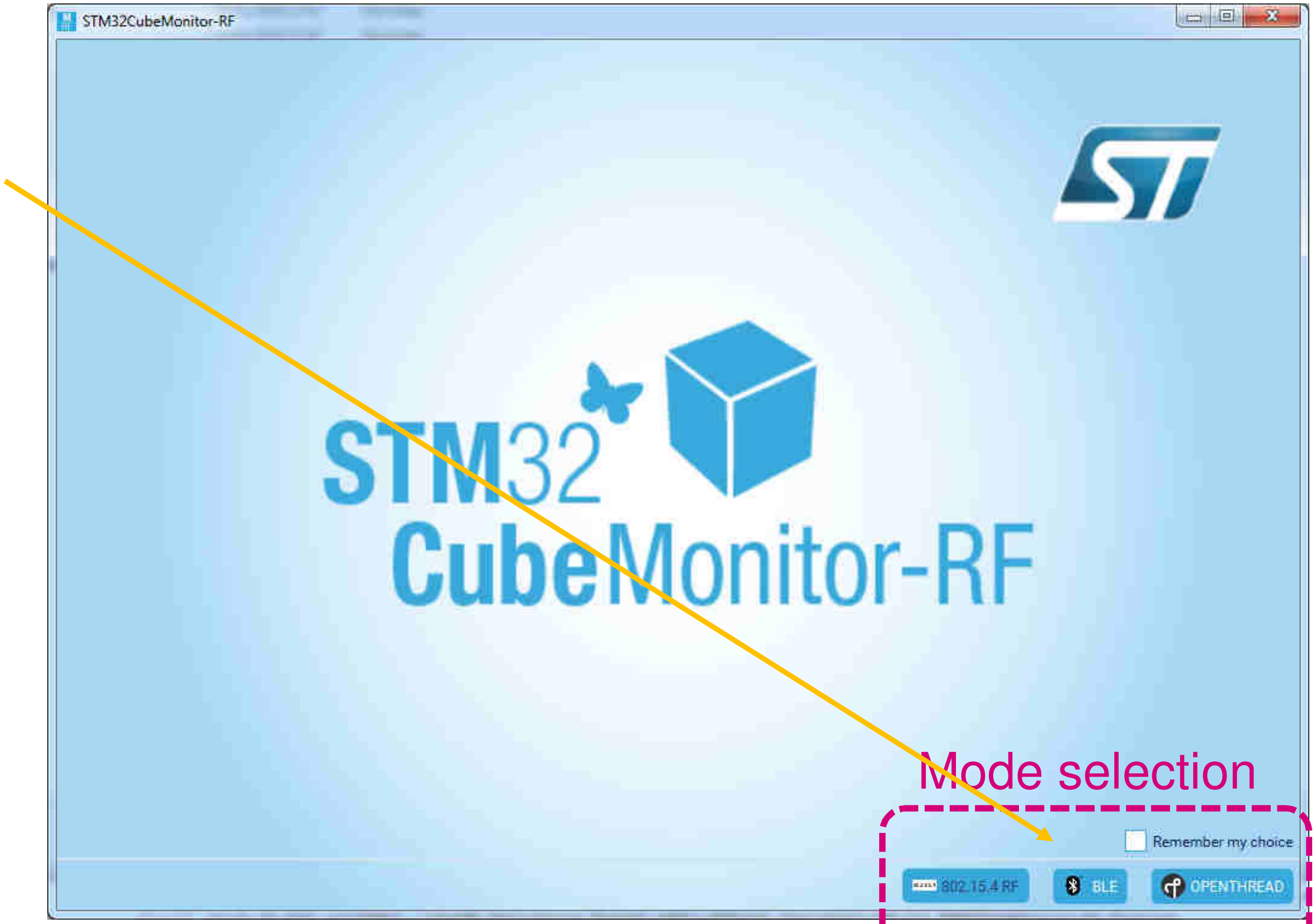
STM32CubeMonitor-RF

- BLE commands
- OpenThread commands
- BLE & 802.15.4 RF tests
- COM-port based



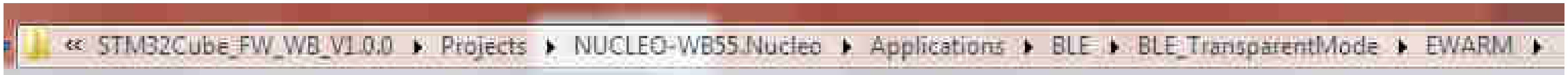
Mode selection

We will run in BLE mode

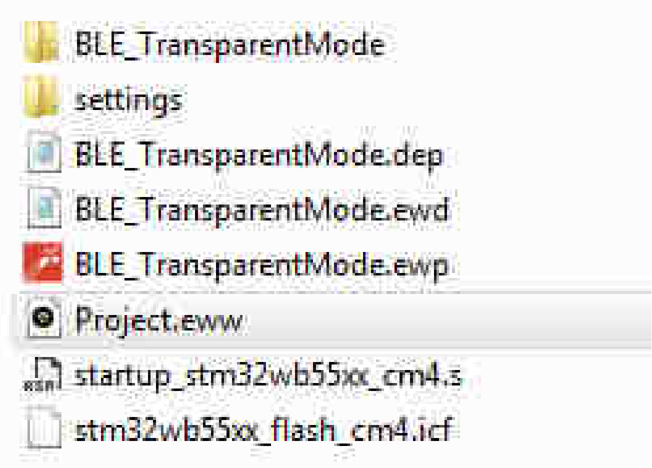


Mode selection

Open the **Transparent Mode** workspace

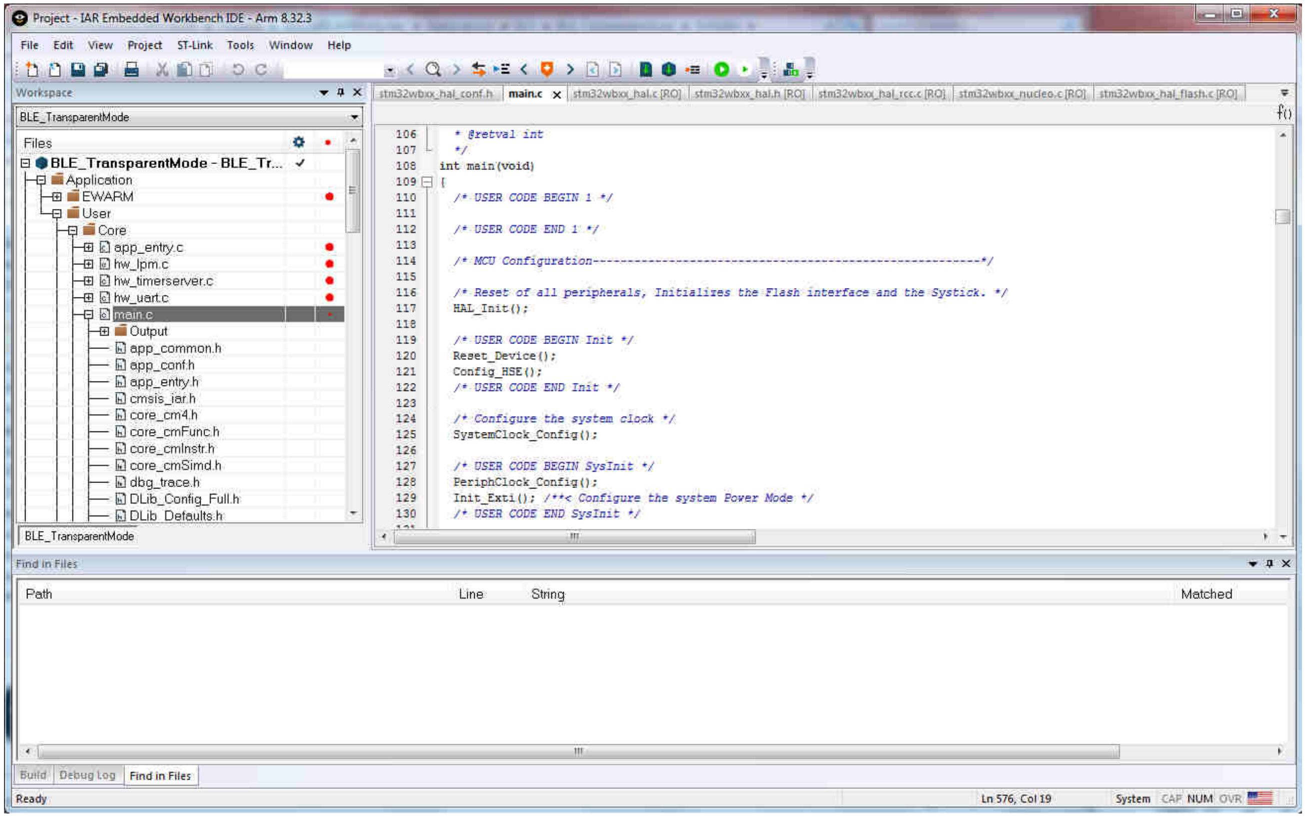


« STM32Cube_FW_WB_V1.0.0 ▸ Projects ▸ NUCLEO-WB55.Nucleo ▸ Applications ▸ BLE ▸ BLE_TransparentMode ▸ EWARM ▸



BLE_TransparentMode
settings
BLE_TransparentMode.dep
BLE_TransparentMode.ewd
BLE_TransparentMode.ewp
Project.eww
startup_stm32wb55xx_cm4.s
stm32wb55xx_flash_cm4.icf

Build, Debug & Run on your Nucleo board

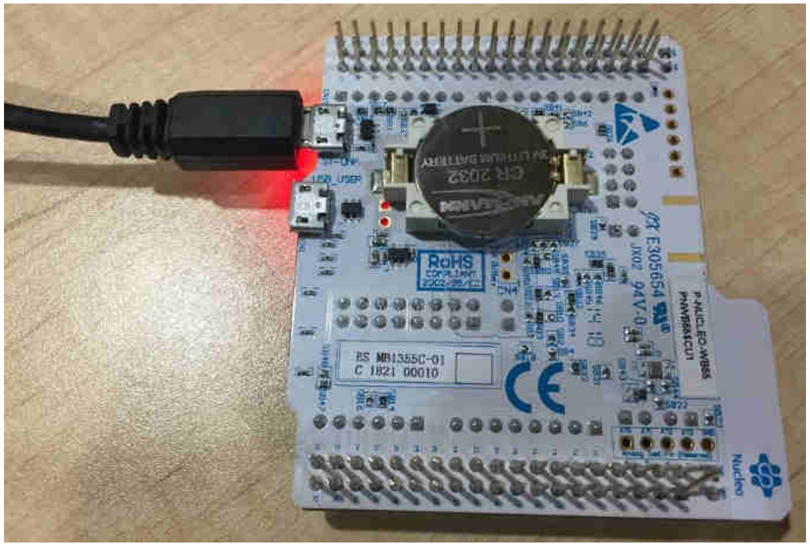


```
  * @retval int
  */
int main(void)
{
  /* USER CODE BEGIN 1 */

  /* USER CODE END 1 */

  /* MCU Configuration----------------------------------------------------------*/

  /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
  HAL_Init();

  /* USER CODE BEGIN Init */
  Reset_Device();
  Config_HSE();
  /* USER CODE END Init */

  /* Configure the system clock */
  SystemClock_Config();

  /* USER CODE BEGIN SysInit */
  PeriphClock_Config();
  Init_Exti(); /**< Configure the system Power Mode */
  /* USER CODE END SysInit */
```

Connect via USB (virtual COM) port

Select device on relevant COM port

Connect to start communication

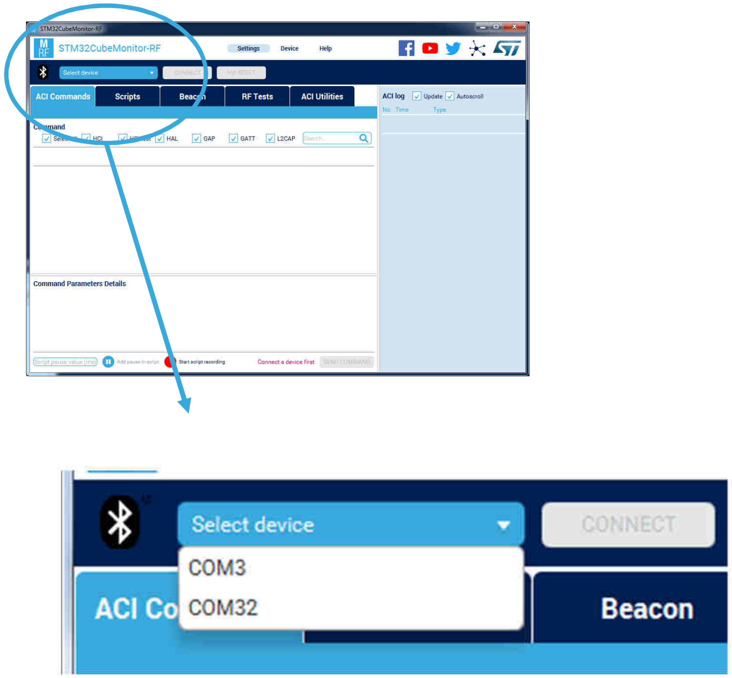


Connect

Connect via USB (virtual COM) port

Select device on relevant COM port

Connect to start communication

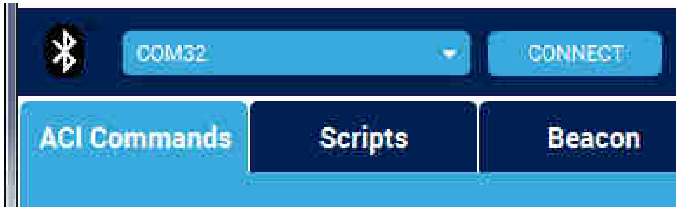


Connect

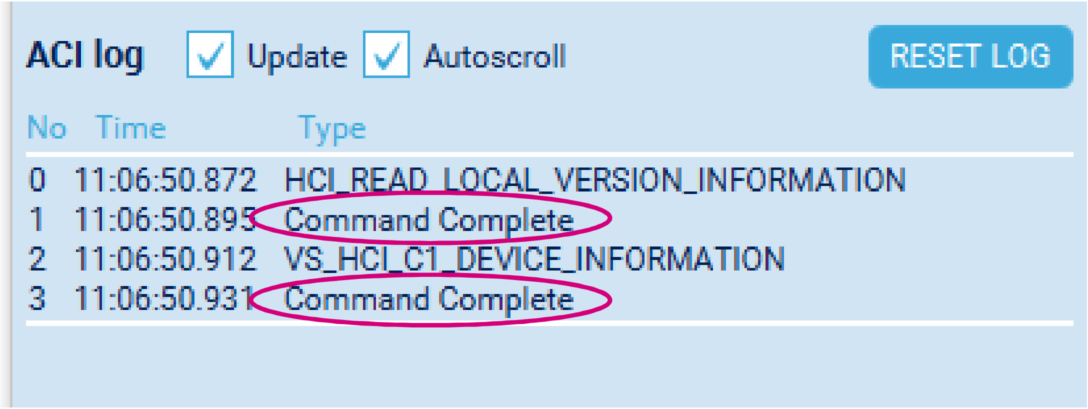
Connect via USB (virtual COM) port

Select device on relevant COM port

Connect to start communication

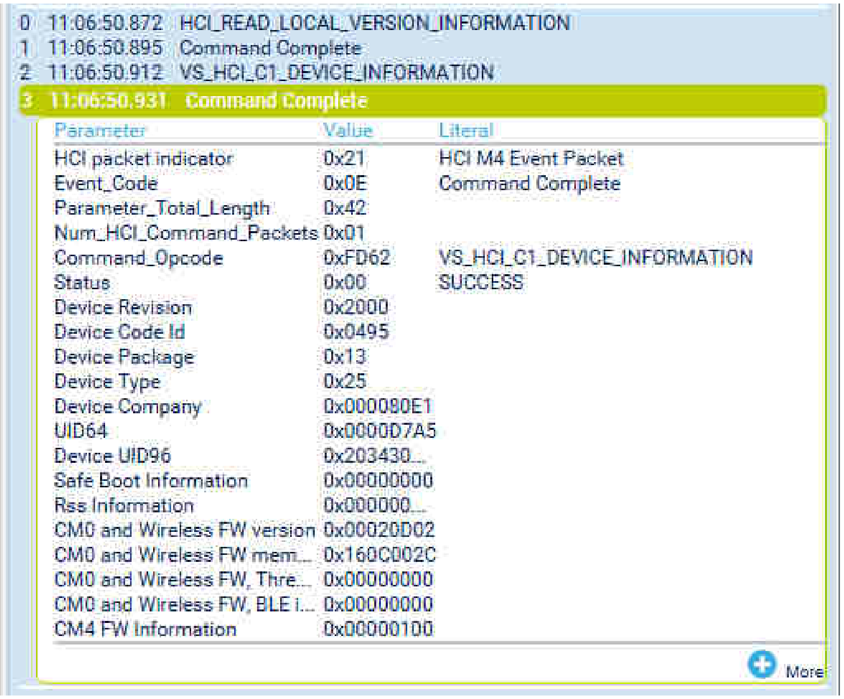


Command Complete signals successful communications



```
No  Time          Type
0   11:06:50.872  HCI_READ_LOCAL_VERSION_INFORMATION
1   11:06:50.895  Command Complete
2   11:06:50.912  VS_HCI_C1_DEVICE_INFORMATION
3   11:06:50.931  Command Complete
```

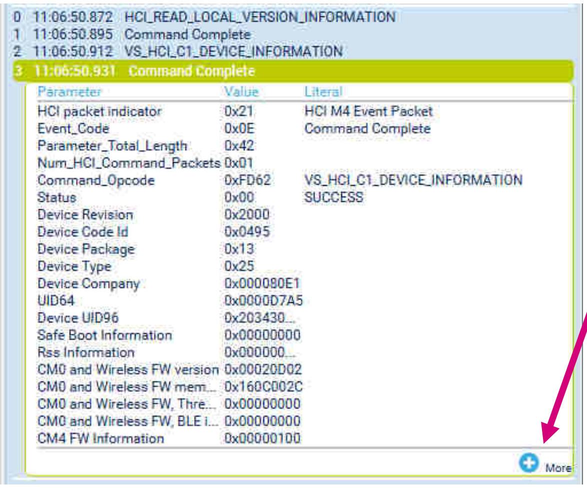
Click on the **Command Complete** line to get more information on the command sent



```
0  11:06:50.872  HCI_READ_LOCAL_VERSION_INFORMATION
1  11:06:50.895  Command Complete
2  11:06:50.912  VS_HCI_C1_DEVICE_INFORMATION
3  11:06:50.931  Command Complete
```

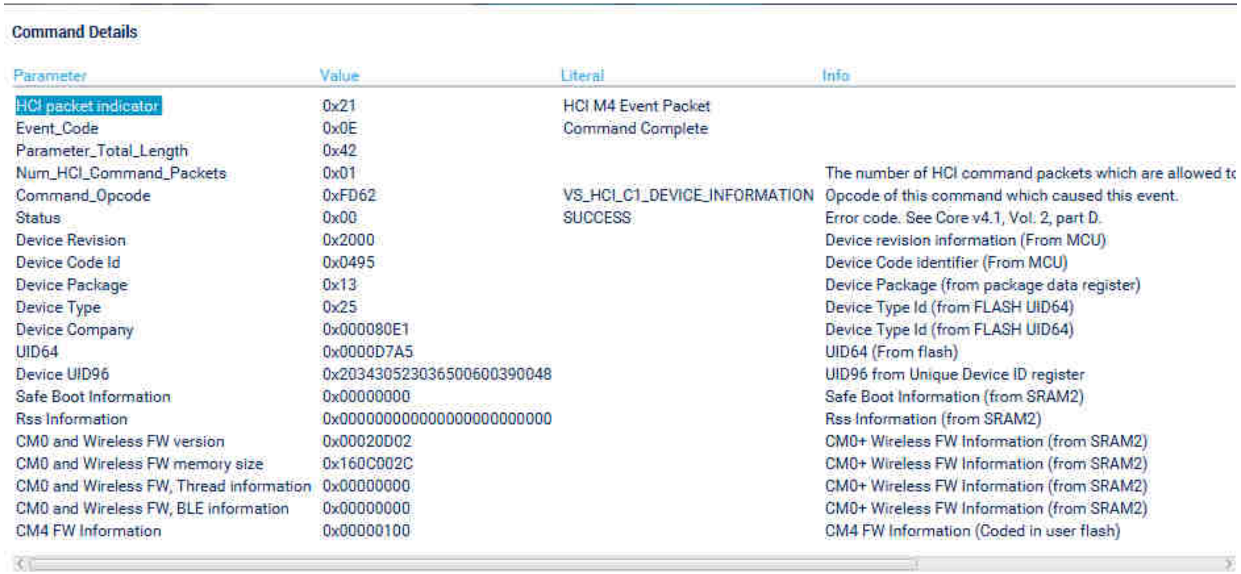
Parameter	Value	Literal
HCI packet indicator	0x21	HCI M4 Event Packet
Event_Code	0x0E	Command Complete
Parameter_Total_Length	0x42	
Num_HCI_Command_Packets	0x01	
Command_Opcode	0xFD62	VS_HCI_C1_DEVICE_INFORMATION
Status	0x00	SUCCESS
Device Revision	0x2000	
Device Code Id	0x0495	
Device Package	0x13	
Device Type	0x25	
Device Company	0x000080E1	
UID64	0x0000D7A5	
Device UID96	0x203430...	
Safe Boot Information	0x00000000	
Rss Information	0x000000...	
CM0 and Wireless FW version	0x00020D02	
CM0 and Wireless FW mem...	0x160C002C	
CM0 and Wireless FW, Thre...	0x00000000	
CM0 and Wireless FW, BLE i...	0x00000000	
CM4 FW Information	0x00000100	

Click on ✚ More for additional detail



```
0  11:06:50.872  HCI_READ_LOCAL_VERSION_INFORMATION
1  11:06:50.895  Command Complete
2  11:06:50.912  VS_HCI_C1_DEVICE_INFORMATION
3  11:06:50.931  Command Complete
```

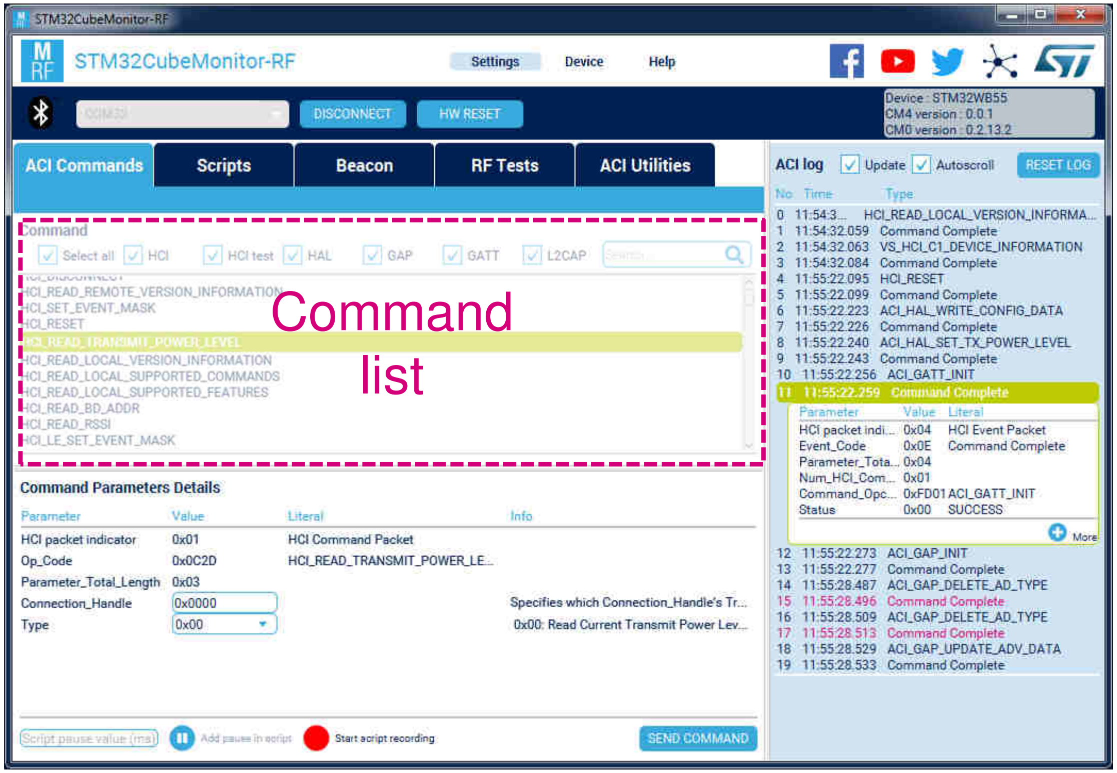
Parameter	Value	Literal
HCI packet indicator	0x21	HCI M4 Event Packet
Event_Code	0x0E	Command Complete
Parameter_Total_Length	0x42	
Num_HCI_Command_Packets	0x01	
Command_Opcode	0xFD62	VS_HCI_C1_DEVICE_INFORMATION
Status	0x00	SUCCESS
Device Revision	0x2000	
Device Code Id	0x0495	
Device Package	0x13	
Device Type	0x25	
Device Company	0x000080E1	
UID64	0x0000D7A5	
Device UID96	0x203430...	
Safe Boot Information	0x00000000	
Rss Information	0x000000...	
CM0 and Wireless FW version	0x00020D02	
CM0 and Wireless FW mem...	0x160C002C	
CM0 and Wireless FW, Thre...	0x00000000	
CM0 and Wireless FW, BLE i...	0x00000000	
CM4 FW Information	0x00000100	



Command Details

Parameter	Value	Literal	Info
HCI packet indicator	0x21	HCI M4 Event Packet	
Event_Code	0x0E	Command Complete	
Parameter_Total_Length	0x42		
Num_HCI_Command_Packets	0x01		The number of HCI command packets which are allowed tc
Command_Opcode	0xFD62	VS_HCI_C1_DEVICE_INFORMATION	Opcode of this command which caused this event.
Status	0x00	SUCCESS	Error code. See Core v4.1, Vol. 2, part D.
Device Revision	0x2000		Device revision information (From MCU)
Device Code Id	0x0495		Device Code identifier (From MCU)
Device Package	0x13		Device Package (from package data register)
Device Type	0x25		Device Type Id (from FLASH UID64)
Device Company	0x000080E1		Device Type Id (from FLASH UID64)
UID64	0x0000D7A5		UID64 (From flash)
Device UID96	0x203430523036500600390048		UID96 from Unique Device ID register
Safe Boot Information	0x00000000		Safe Boot Information (from SRAM2)
Rss Information	0x000000000000000000000000		Rss Information (from SRAM2)
CM0 and Wireless FW version	0x00020D02		CM0+ Wireless FW Information (from SRAM2)
CM0 and Wireless FW memory size	0x160C002C		CM0+ Wireless FW Information (from SRAM2)
CM0 and Wireless FW, Thread information	0x00000000		CM0+ Wireless FW Information (from SRAM2)
CM0 and Wireless FW, BLE information	0x00000000		CM0+ Wireless FW Information (from SRAM2)
CM4 FW Information	0x00000100		CM4 FW Information (Coded in user flash)

Lots of categories to choose and filter from



Device : STM32WB55
CM4 version : 0.0.1
CM0 version : 0.2.13.2

Command

☑ Select all ☑ HCI ☑ HCI test ☑ HAL ☑ GAP ☑ GATT ☑ L2CAP Search...

```
HCI_DISCONNECT
HCI_READ_REMOTE_VERSION_INFORMATION
HCI_SET_EVENT_MASK
HCI_RESET
HCI_READ_TRANSMIT_POWER_LEVEL
HCI_READ_LOCAL_VERSION_INFORMATION
HCI_READ_LOCAL_SUPPORTED_COMMANDS
HCI_READ_LOCAL_SUPPORTED_FEATURES
HCI_READ_BD_ADDR
HCI_READ_RSSI
HCI_LE_SET_EVENT_MASK
```

Command Parameters Details

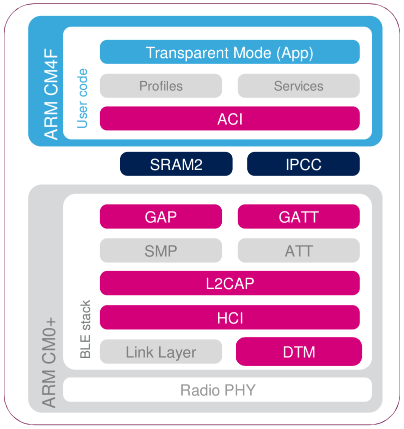
Parameter	Value	Literal	Info
HCI packet indicator	0x01	HCI Command Packet	
Op_Code	0x0C2D	HCI_READ_TRANSMIT_POWER_LE...	
Parameter_Total_Length	0x03		
Connection_Handle	0x0000		Specifies which Connection_Handle's Tr...
Type	0x00		0x00: Read Current Transmit Power Lev...

ACI log ☑ Update ☑ Autoscroll

```
No  Time          Type
0   11:54:3...    HCI_READ_LOCAL_VERSION_INFORMA...
1   11:54:32.059  Command Complete
2   11:54:32.063  VS_HCI_C1_DEVICE_INFORMATION
3   11:54:32.084  Command Complete
4   11:55:22.095  HCI_RESET
5   11:55:22.099  Command Complete
6   11:55:22.223  ACI_HAL_WRITE_CONFIG_DATA
7   11:55:22.226  Command Complete
8   11:55:22.240  ACI_HAL_SET_TX_POWER_LEVEL
9   11:55:22.243  Command Complete
10  11:55:22.256  ACI_GATT_INIT
11  11:55:22.259  Command Complete
      Parameter           Value   Literal
      HCI packet indi...  0x04    HCI Event Packet
      Event_Code          0x0E    Command Complete
      Parameter_Tota...   0x04
      Num_HCI_Com...      0x01
      Command_Opc...      0xFD01  ACI_GATT_INIT
      Status              0x00    SUCCESS
12  11:55:22.273  ACI_GAP_INIT
13  11:55:22.277  Command Complete
14  11:55:28.487  ACI_GAP_DELETE_AD_TYPE
15  11:55:28.496  Command Complete
16  11:55:28.509  ACI_GAP_DELETE_AD_TYPE
17  11:55:28.513  Command Complete
18  11:55:28.529  ACI_GAP_UPDATE_ADV_DATA
19  11:55:28.533  Command Complete
```

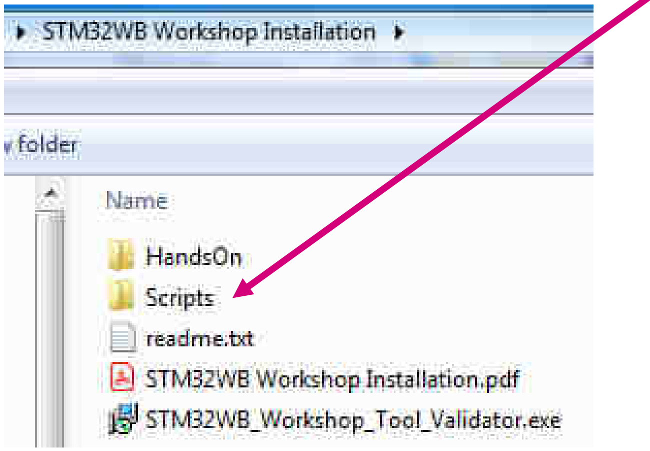
Command categories:

- HCI
- HCI test
- HAL
- GAP
- GATT
- L2CAP

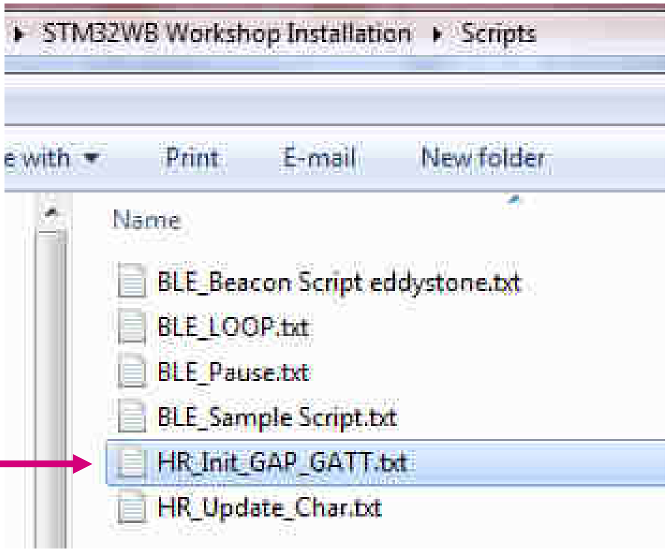


Open and Edit the HR_Init_GAP_GATT.txt Script file

(In your installation zip file, ***Scripts*** folder)

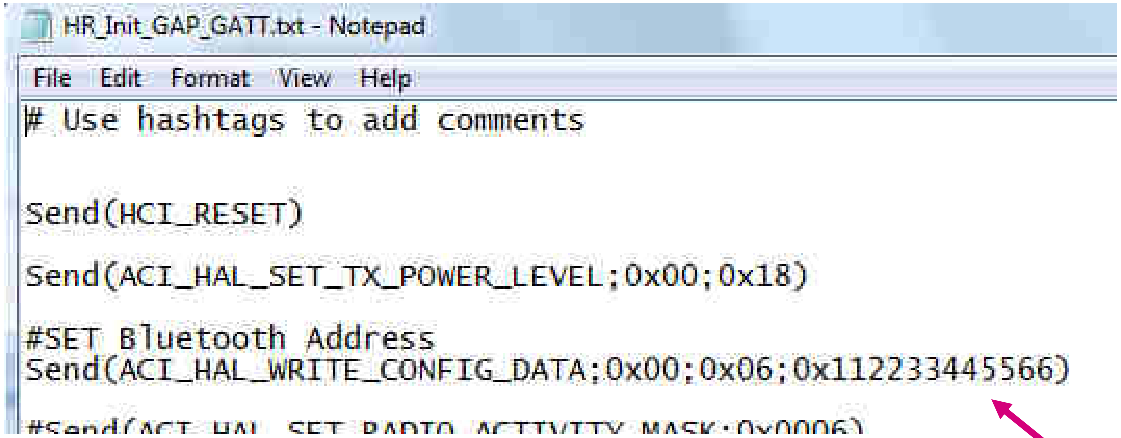


HandsOn
Scripts
readme.txt
STM32WB Workshop Installation.pdf
STM32WB_Workshop_Tool_Validator.exe



BLE_Beacon Script eddystone.txt
BLE_LOOP.txt
BLE_Pause.txt
BLE_Sample Script.txt
HR_Init_GAP_GATT.txt
HR_Update_Char.txt

Scripts



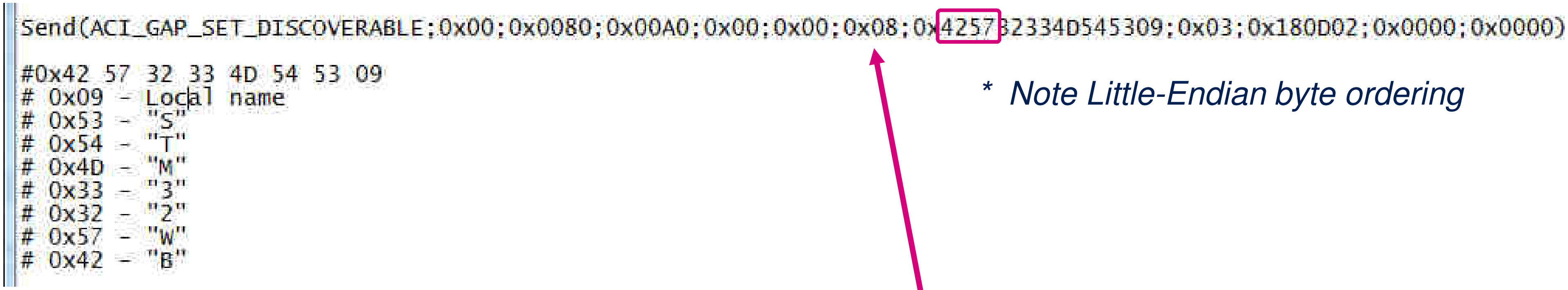
```
# Use hashtags to add comments


Send(HCI_RESET)

Send(ACI_HAL_SET_TX_POWER_LEVEL;0x00;0x18)

#SET Bluetooth Address
Send(ACI_HAL_WRITE_CONFIG_DATA;0x00;0x06;0x112233445566)

#Send(ACI_HAL_SET_RADIO_ACTIVITY_MASK;0x0006)
```

Set the Bluetooth Address

Modify this value as you wish

Send(ACI_HAL_WRITE_CONFIG_DATA;0x00;0x06;0x112233445566)

- Change the two characters of the Local Name with your Magic number (e.g. change 0x4257 to 0x3130 for magic number “01”.



```
Send(ACI_GAP_SET_DISCOVERABLE;0x00;0x0080;0x00A0;0x00;0x00;0x08;0x425732334D545309;0x03;0x180D02;0x0000;0x0000)

#0x42 57 32 33 4D 54 53 09
# 0x09 - Local name
# 0x53 - "S"
# 0x54 - "T"
# 0x4D - "M"
# 0x33 - "3"
# 0x32 - "2"
# 0x57 - "W"
# 0x42 - "B"
```

* *Note Little-Endian byte ordering*

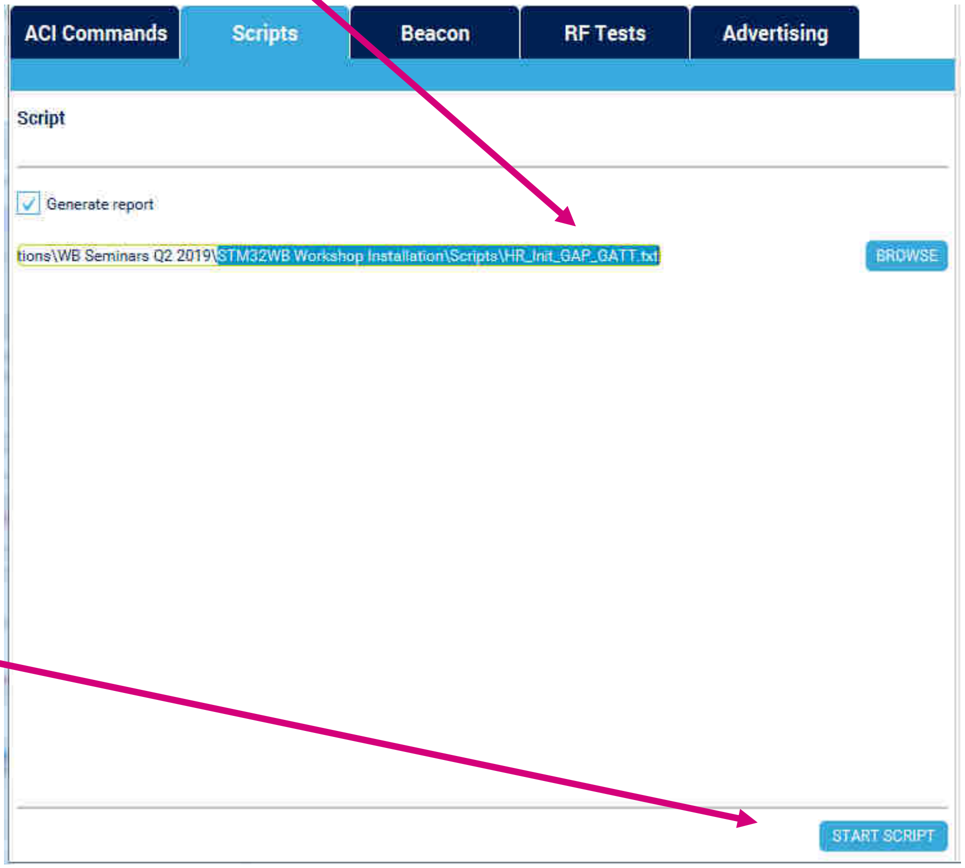
7 ASCII chars + 0x09 = 0x08. To add characters, also change the LENGTH parameter (x+1)

Hex	Char
30	0
31	1
32	2
33	3
34	4
35	5
36	6
37	7
38	8
39	9

** ASCII Character Set for Magic numbers*

Scripts

Save and Load your script



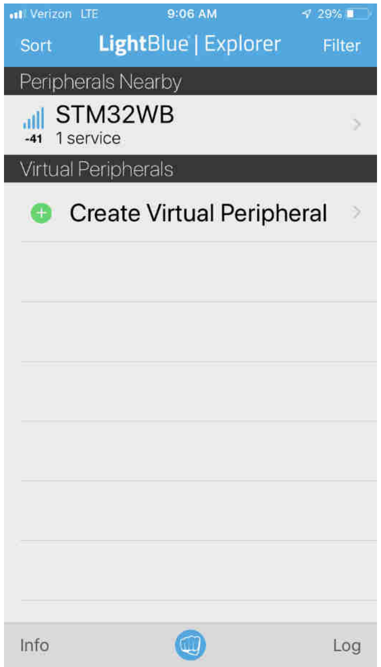
☑ Generate report

tions\WB Seminars Q2 2019\STM32WB Workshop Installation\Scripts\HR_Init_GAP_GATT.txt

Start Script

LightBlue | Explorer App

Find your device



Verizon LTE 9:06 AM 29%

Sort LightBlue | Explorer Filter

Peripherals Nearby

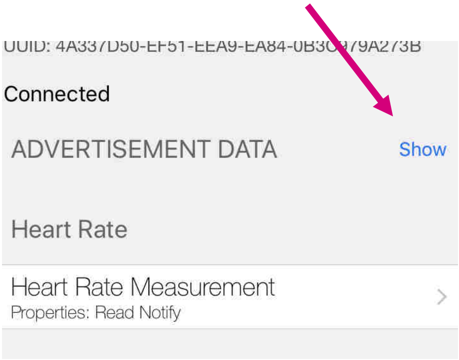
STM32WB
-41 1 service

Virtual Peripherals

Create Virtual Peripheral

Info Log

Show ADV data



UUID: 4A337D50-EF51-EEA9-EA84-0B3C979A273B

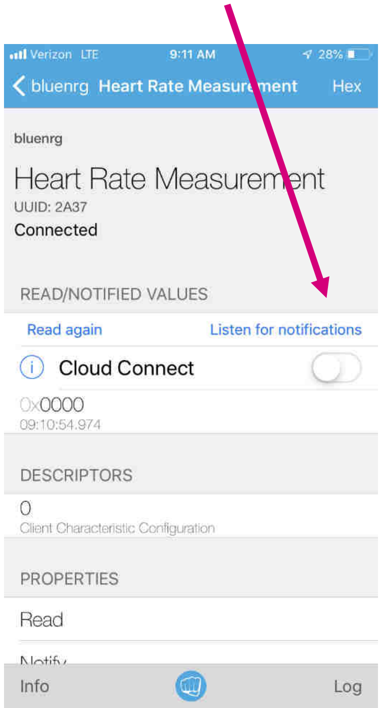
Connected

ADVERTISEMENT DATA Show

Heart Rate

Heart Rate Measurement
Properties: Read Notify

Enable Notifications



Verizon LTE 9:11 AM 28%

bluenrg Heart Rate Measurement Hex

bluenrg

Heart Rate Measurement
UUID: 2A37
Connected

READ/NOTIFIED VALUES

Read again Listen for notifications

Cloud Connect

0x0000
09:10:54.974

DESCRIPTORS

0
Client Characteristic Configuration

PROPERTIES

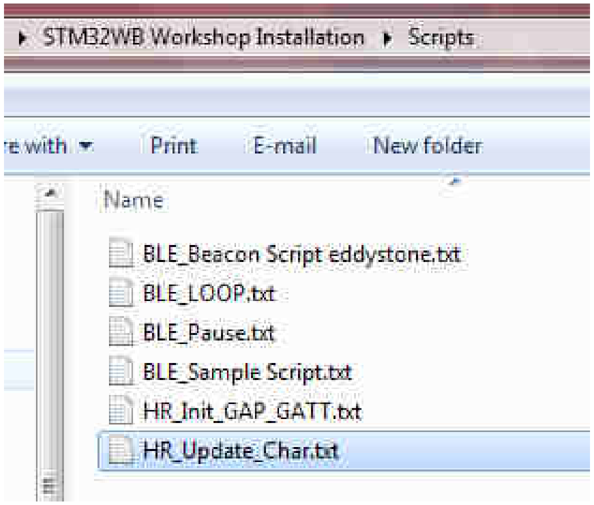
Read

Notify

Info Log

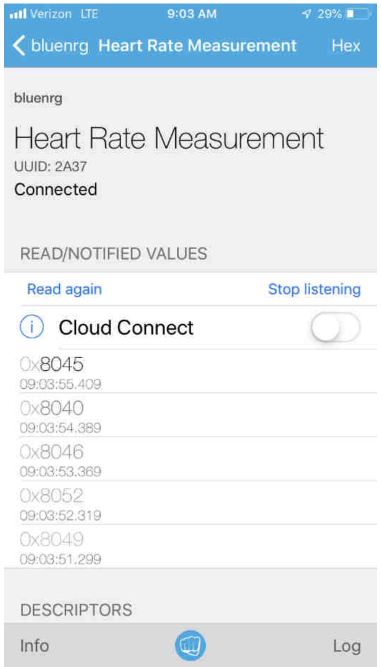
STM32CubeMonitorRF

Now load the ***HR_Update_Char.txt*** script to send Notification Updates



BLE_Beacon Script eddystone.txt
BLE_LOOP.txt
BLE_Pause.txt
BLE_Sample Script.txt
HR_Init_GAP_GATT.txt
HR_Update_Char.txt

Dummy Heart Rate Values are sent

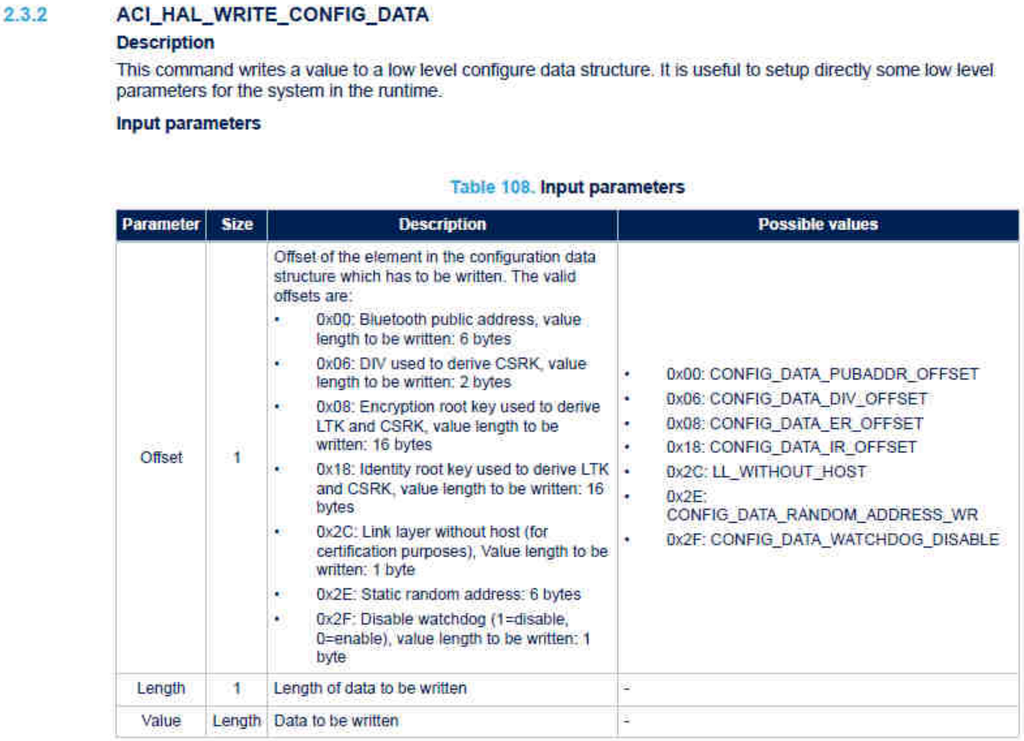


bluenrg
Heart Rate Measurement
UUID: 2A37
Connected

0x8045
09:03:55.409
0x8040
09:03:54.389
0x8046
09:03:53.369
0x8052
09:03:52.319
0x8049
09:03:51.299

Application Note AN5270 describes the ACI/HCI commands available

- via CubeMonitorRF
- via Application API's



2.3.2 ACI_HAL_WRITE_CONFIG_DATA

Description

This command writes a value to a low level configure data structure. It is useful to setup directly some low level parameters for the system in the runtime.

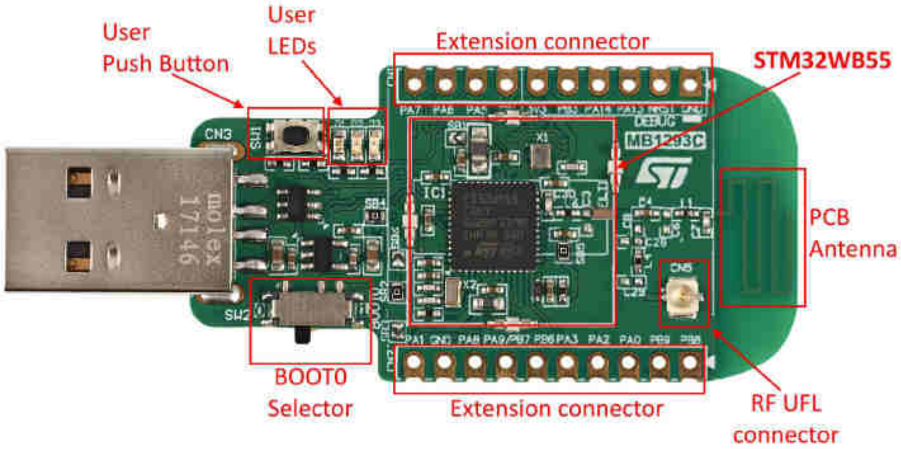
Input parameters

Table 108. **Input parameters**

Parameter	Size	Description	Possible values
Offset	1	Offset of the element in the configuration data structure which has to be written. The valid offsets are: • 0x00: Bluetooth public address, value length to be written: 6 bytes • 0x06: DIV used to derive CSRK, value length to be written: 2 bytes • 0x08: Encryption root key used to derive LTK and CSRK, value length to be written: 16 bytes • 0x18: Identity root key used to derive LTK and CSRK, value length to be written: 16 bytes • 0x2C: Link layer without host (for certification purposes), Value length to be written: 1 byte • 0x2E: Static random address: 6 bytes • 0x2F: Disable watchdog (1=disable, 0=enable), value length to be written: 1 byte	• 0x00: CONFIG_DATA_PUBADDR_OFFSET • 0x06: CONFIG_DATA_DIV_OFFSET • 0x08: CONFIG_DATA_ER_OFFSET • 0x18: CONFIG_DATA_IR_OFFSET • 0x2C: LL_WITHOUT_HOST • 0x2E: CONFIG_DATA_RANDOM_ADDRESS_WR • 0x2F: CONFIG_DATA_WATCHDOG_DISABLE
Length	1	Length of data to be written	-
Value	Length	Data to be written	-

WB55 Dongle Board

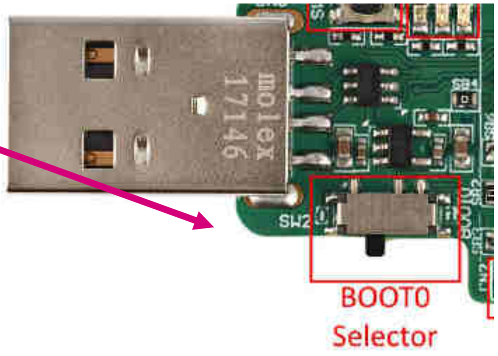
- The USB Dongle is quite useful as the CubeMonitorRF sniffer
- Uses the USB CDC class directly to parse commands
- USB bootloader invoked via **BOOT0 switch** & CubeProgrammer, and the binary can be programmed



Dongle config

- ❑ Move Dongle Switch to Bootloader mode

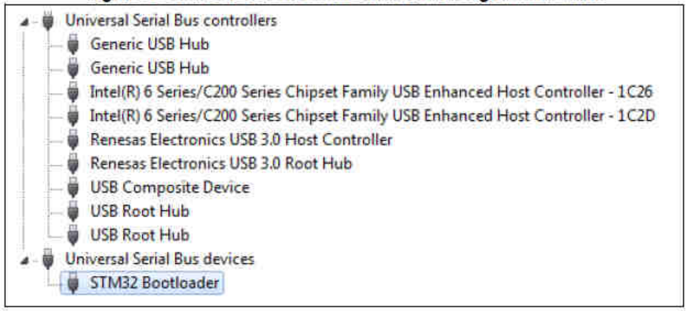
- ❑ Plug in Dongle



Bootloader active to the right

- ❑ Ensure the driver has enumerated **“STM32 Bootloader”**

Figure 4. STM32 DFU device with STM32CubeProgrammer driver

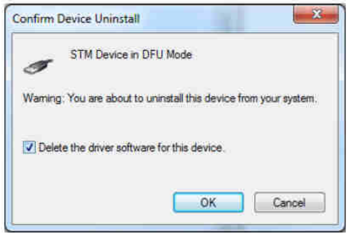


```
Universal Serial Bus controllers
    Generic USB Hub
    Generic USB Hub
    Intel(R) 6 Series/C200 Series Chipset Family USB Enhanced Host Controller - 1C26
    Intel(R) 6 Series/C200 Series Chipset Family USB Enhanced Host Controller - 1C2D
    Renesas Electronics USB 3.0 Host Controller
    Renesas Electronics USB 3.0 Root Hub
    USB Composite Device
    USB Root Hub
    USB Root Hub
Universal Serial Bus devices
    STM32 Bootloader
```

Chapter 1.2.4 details the DFU driver install / update procedure

Old or Native MS drivers must be replaced to properly access the bootloader

Figure 2. Deleting the old driver software



Confirm Device Uninstall

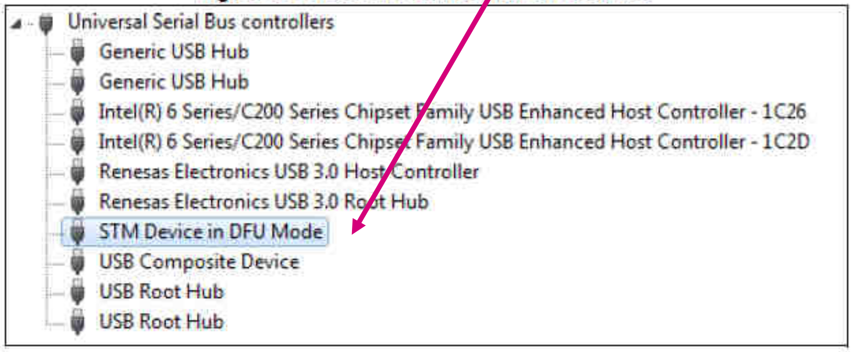
STM Device in DFU Mode

Warning: You are about to uninstall this device from your system.

☑ Delete the driver software for this device.

Old Driver

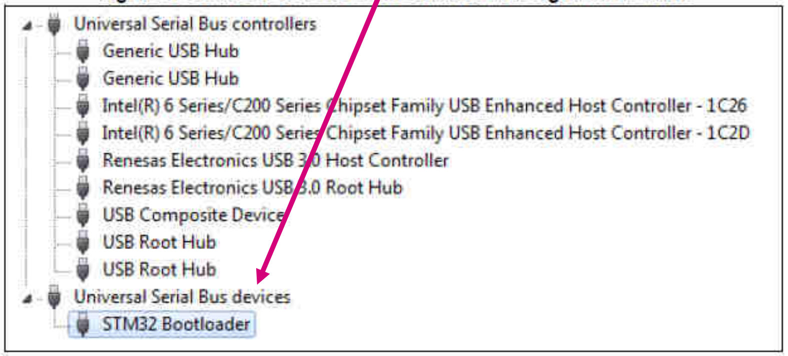
Figure 3. STM32 DFU device with DfuSe driver



- Universal Serial Bus controllers
  - Generic USB Hub
  - Generic USB Hub
  - Intel(R) 6 Series/C200 Series Chipset Family USB Enhanced Host Controller - 1C26
  - Intel(R) 6 Series/C200 Series Chipset Family USB Enhanced Host Controller - 1C2D
  - Renesas Electronics USB 3.0 Host Controller
  - Renesas Electronics USB 3.0 Root Hub
  - STM Device in DFU Mode
  - USB Composite Device
  - USB Root Hub
  - USB Root Hub

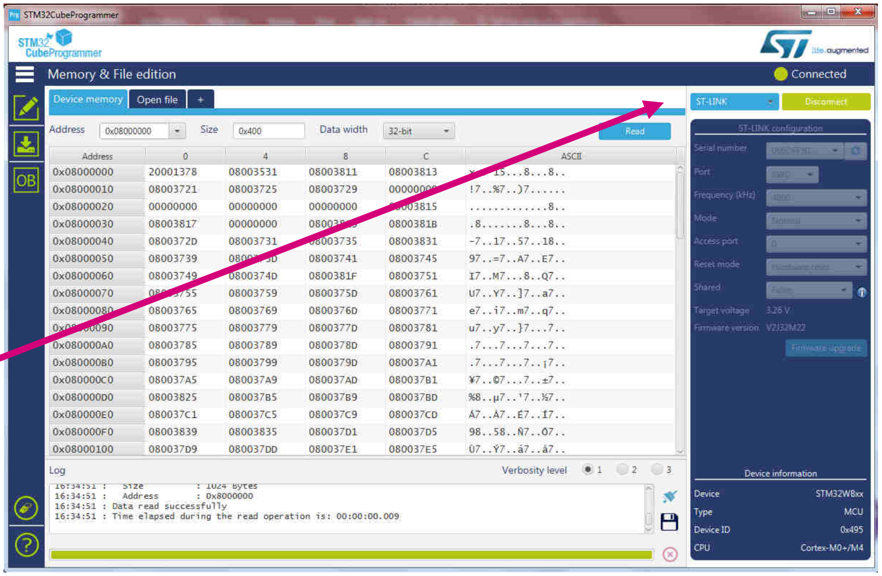
New Driver!

Figure 4. STM32 DFU device with STM32CubeProgrammer driver



- Universal Serial Bus controllers
  - Generic USB Hub
  - Generic USB Hub
  - Intel(R) 6 Series/C200 Series Chipset Family USB Enhanced Host Controller - 1C26
  - Intel(R) 6 Series/C200 Series Chipset Family USB Enhanced Host Controller - 1C2D
  - Renesas Electronics USB 3.0 Host Controller
  - Renesas Electronics USB 3.0 Root Hub
  - USB Composite Device
  - USB Root Hub
  - USB Root Hub
- Universal Serial Bus devices
  - STM32 Bootloader

- ❑ Open STM32 CubeProgrammer



Address 0x08000000 Size 0x400 Data width 32-bit

Address	0	4	8	C	ASCII
0x08000000	20001378	08003531	08003811	08003813	x...15...8...8..
0x08000010	08003721	08003725	08003729	00000000	!7..%7..)7......
0x08000020	00000000	00000000	00000000	08003815	............8..
0x08000030	08003817	00000000	0800381[illegible]	0800381B	.8.......8...8..
0x08000040	0800372D	08003731	08003735	08003831	-7..17..57..18..
0x08000050	08003739	0800373D	08003741	08003745	97..=7..A7..E7..
0x08000060	08003749	0800374D	0800381F	08003751	I7..M7...8..Q7..
0x08000070	08003755	08003759	0800375D	08003761	U7..Y7..]7..a7..
0x08000080	08003765	08003769	0800376D	08003771	e7..i7..m7..q7..
0x08000090	08003775	08003779	0800377D	08003781	u7..y7..}7....7..
0x080000A0	08003785	08003789	0800378D	08003791	.7...7...7...7..
0x080000B0	08003795	08003799	0800379D	080037A1	.7...7...7..¡7..
0x080000C0	080037A5	080037A9	080037AD	080037B1	¥7..©7...7..±7..
0x080000D0	08003825	080037B5	080037B9	080037BD	%8..µ7..¹7..½7..
0x080000E0	080037C1	080037C5	080037C9	080037CD	Á7..Å7..É7..Í7..
0x080000F0	08003839	08003835	080037D1	080037D5	98..58..Ñ7..Õ7..
0x08000100	080037D9	080037DD	080037E1	080037E5	Ù7..Ý7..á7..å7..

Log Verbosity level ● 1 ○ 2 ○ 3

```
16:34:51 :   Size           : 1024 Bytes
16:34:51 :   Address        : 0x8000000
16:34:51 : Data read successfully
16:34:51 : Time elapsed during the read operation is: 00:00:00.009
```

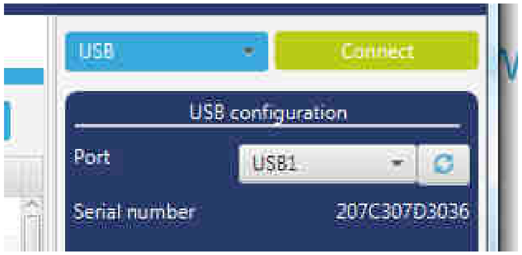
ST-LINK configuration

Serial number
Port
Frequency (kHz)
Mode
Access port
Reset mode
Shared
Target voltage 3.26 V
Firmware version V2J32M22

Device information

Device	STM32WBxx
Type	MCU
Device ID	0x495
CPU	Cortex-M0+/M4

- ❑ Select **USB** mode and **Connect**

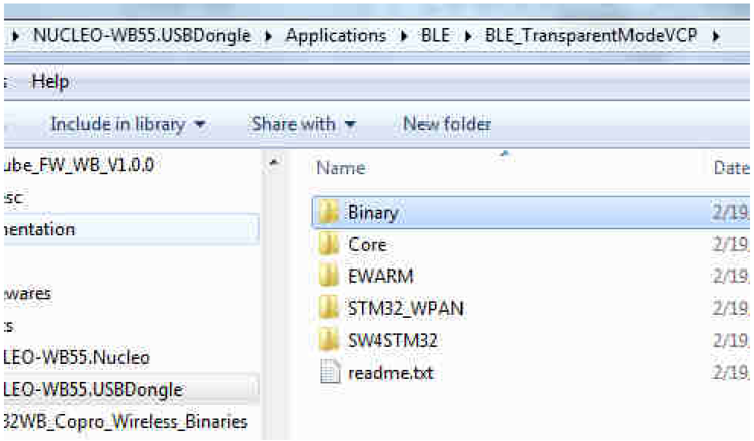


USB configuration

Port USB1

Serial number 207C307D3036

Open the BLE_TransparentModeVCP.hex file for Dongle



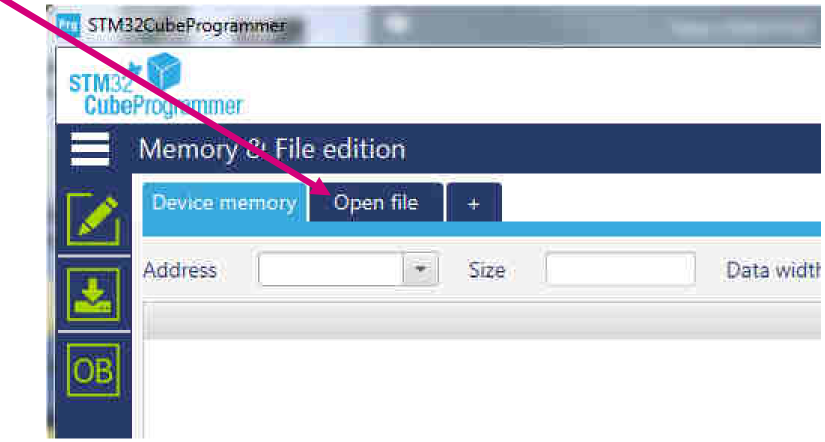
▸ NUCLEO-WB55.USBDongle ▸ Applications ▸ BLE ▸ BLE_TransparentModeVCP ▸

Help

Include in library ▾ Share with ▾ New folder

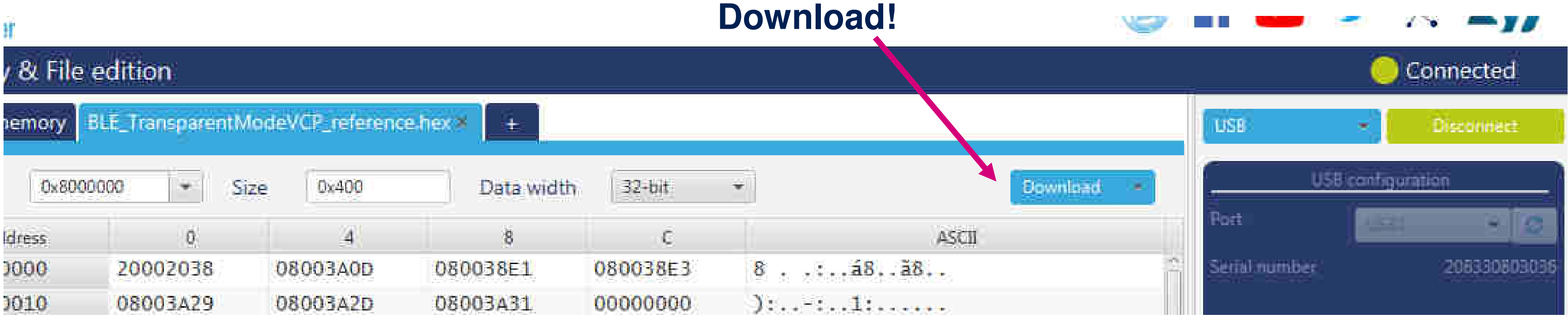
ube_FW_WB_V1.0.0
sc
ıentation
wares
s
LEO-WB55.Nucleo
LEO-WB55.USBDongle
32WB_Copro_Wireless_Binaries

Name	Date
Binary	2/19
Core	2/19
EWARM	2/19
STM32_WPAN	2/19
SW4STM32	2/19
readme.txt	2/19



STM32CubeProgrammer

Download!



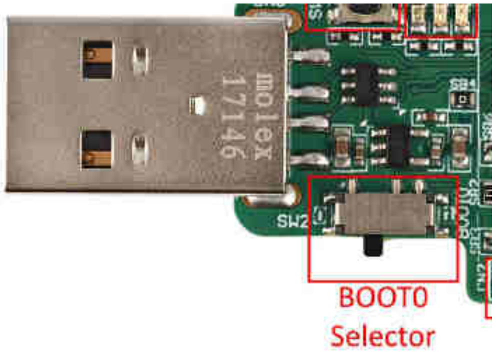
Address 0x8000000 Size 0x400 Data width 32-bit

Serial number 208330803036

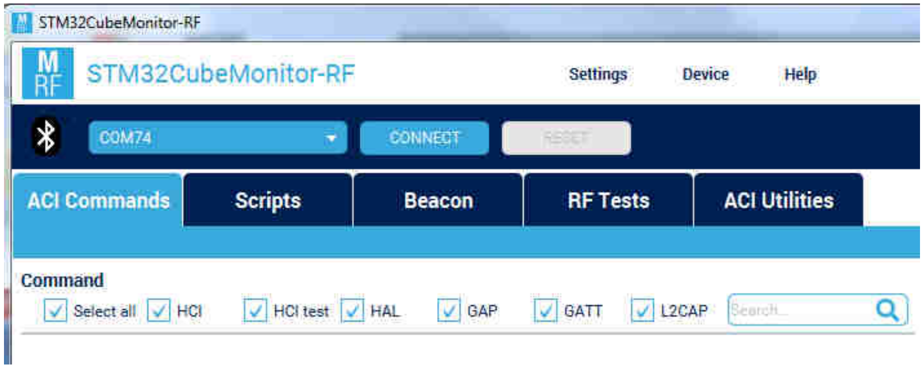
```
ldress   0         4         8         C         ASCII
0000     20002038  08003A0D  080038E1  080038E3  8 . .:..á8..ã8..
0010     08003A29  08003A2D  08003A31  00000000  ):..-:..1:......
```

Dongle config

- ❑ Disconnect from CubeProgrammer
- ❑ Unplug Dongle
- ❑ Move Dongle Switch back to normal boot mode
- ❑ Plug Dongle back in for normal boot startup
- ❑ Now you should be able to use **COMxx** in CubeMonitorRF
  - ❑ *(may differ from COM74)*
- ❑ **CONNECT**



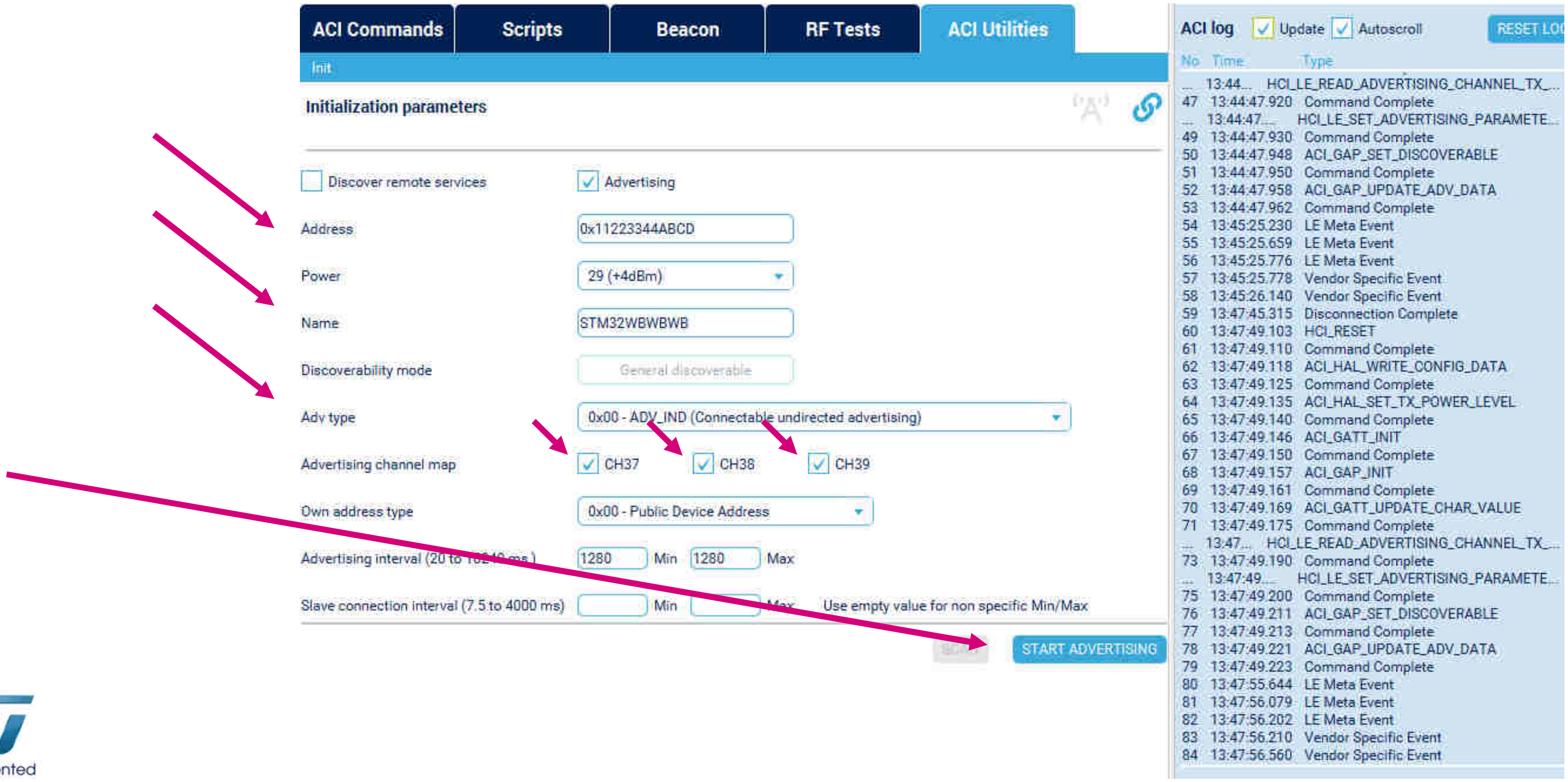
Normal Boot to the left



Change the Bluetooth Address and Name

Use Connectable advertising on all channels (37/38/39)

Start



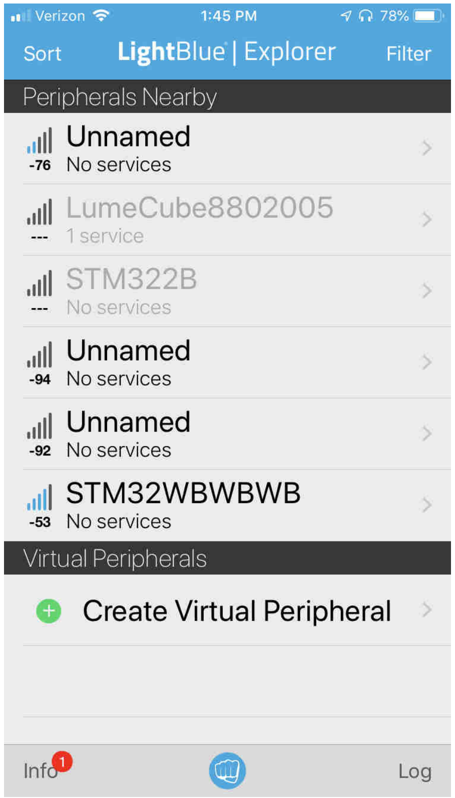
Initialization parameters

- ☐ Discover remote services
- ☑ Advertising
- Address: 0x11223344ABCD
- Power: 29 (+4dBm)
- Name: STM32WBWBWB
- Discoverability mode: General discoverable
- Adv type: 0x00 - ADV_IND (Connectable undirected advertising)
- Advertising channel map: ☑ CH37 ☑ CH38 ☑ CH39
- Own address type: 0x00 - Public Device Address
- Advertising interval (20 to 10240 ms): 1280 Min 1280 Max
- Slave connection interval (7.5 to 4000 ms): Min Max Use empty value for non specific Min/Max

ACI log ☑ Update ☑ Autoscroll

```
No  Time          Type
...  13:44...     HCI_LE_READ_ADVERTISING_CHANNEL_TX_...
47  13:44:47.920  Command Complete
...  13:44:47...  HCI_LE_SET_ADVERTISING_PARAMETE...
49  13:44:47.930  Command Complete
50  13:44:47.948  ACI_GAP_SET_DISCOVERABLE
51  13:44:47.950  Command Complete
52  13:44:47.958  ACI_GAP_UPDATE_ADV_DATA
53  13:44:47.962  Command Complete
54  13:45:25.230  LE Meta Event
55  13:45:25.659  LE Meta Event
56  13:45:25.776  LE Meta Event
57  13:45:25.778  Vendor Specific Event
58  13:45:26.140  Vendor Specific Event
59  13:47:45.315  Disconnection Complete
60  13:47:49.103  HCI_RESET
61  13:47:49.110  Command Complete
62  13:47:49.118  ACI_HAL_WRITE_CONFIG_DATA
63  13:47:49.125  Command Complete
64  13:47:49.135  ACI_HAL_SET_TX_POWER_LEVEL
65  13:47:49.140  Command Complete
66  13:47:49.146  ACI_GATT_INIT
67  13:47:49.150  Command Complete
68  13:47:49.157  ACI_GAP_INIT
69  13:47:49.161  Command Complete
70  13:47:49.169  ACI_GATT_UPDATE_CHAR_VALUE
71  13:47:49.175  Command Complete
...  13:47...     HCI_LE_READ_ADVERTISING_CHANNEL_TX_...
73  13:47:49.190  Command Complete
...  13:47:49...  HCI_LE_SET_ADVERTISING_PARAMETE...
75  13:47:49.200  Command Complete
76  13:47:49.211  ACI_GAP_SET_DISCOVERABLE
77  13:47:49.213  Command Complete
78  13:47:49.221  ACI_GAP_UPDATE_ADV_DATA
79  13:47:49.223  Command Complete
80  13:47:55.644  LE Meta Event
81  13:47:56.079  LE Meta Event
82  13:47:56.202  LE Meta Event
83  13:47:56.210  Vendor Specific Event
84  13:47:56.560  Vendor Specific Event
```

Use LightBlue Explorer to connect to and interrogate your GAP peripheral



Verizon 1:45 PM 78%

Sort LightBlue | Explorer Filter

Peripherals Nearby

Unnamed
-76 No services

LumeCube8802005
--- 1 service

STM322B
--- No services

Unnamed
-94 No services

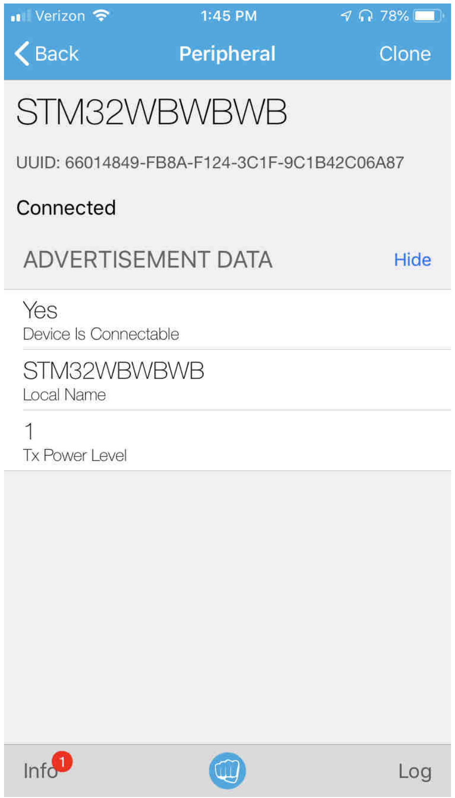
Unnamed
-92 No services

STM32WBWBWB
-53 No services

Virtual Peripherals

Create Virtual Peripheral

Info 1 Log



Verizon 1:45 PM 78%

Back Peripheral Clone

STM32WBWBWB

UUID: 66014849-FB8A-F124-3C1F-9C1B42C06A87

Connected

ADVERTISEMENT DATA Hide

Yes
Device Is Connectable

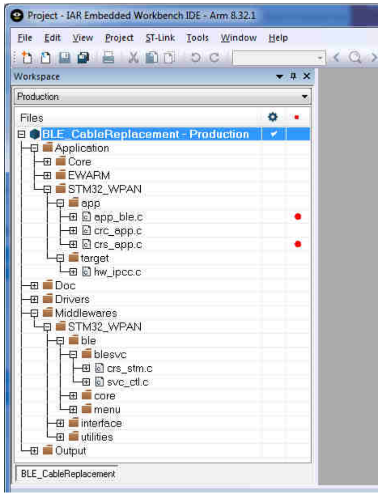
STM32WBWBWB
Local Name

1
Tx Power Level

Info 1 Log

Hands-On

Custom GATT & Cable Replacement



Project - IAR Embedded Workbench IDE - Arm 8.32.1

```
⊟ BLE_CableReplacement - Production
├⊟ Application
│ ├⊞ Core
│ ├⊞ EWARM
│ └⊟ STM32_WPAN
│   ├⊟ app
│   │ ├⊞ app_ble.c
│   │ ├⊞ crc_app.c
│   │ └⊞ crs_app.c
│   └⊟ target
│     └⊞ hw_ipcc.c
├⊞ Doc
├⊞ Drivers
├⊟ Middlewares
│ └⊟ STM32_WPAN
│   ├⊟ ble
│   │ ├⊟ blesvc
│   │ │ ├⊞ crs_stm.c
│   │ │ └⊞ svc_ctl.c
│   │ ├⊞ core
│   │ └⊞ menu
│   ├⊞ interface
│   └⊞ utilities
└⊞ Output
```

Hands On Time! Custom GATT

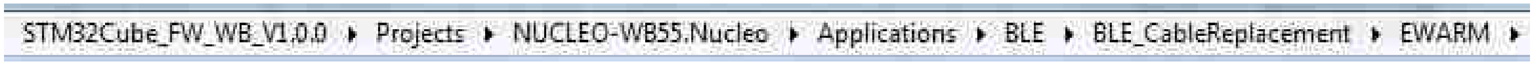
Open, Compile, Program & Run the Nucleo Board ***CableReplacement*** example

Add a custom GATT Characteristic for LED control

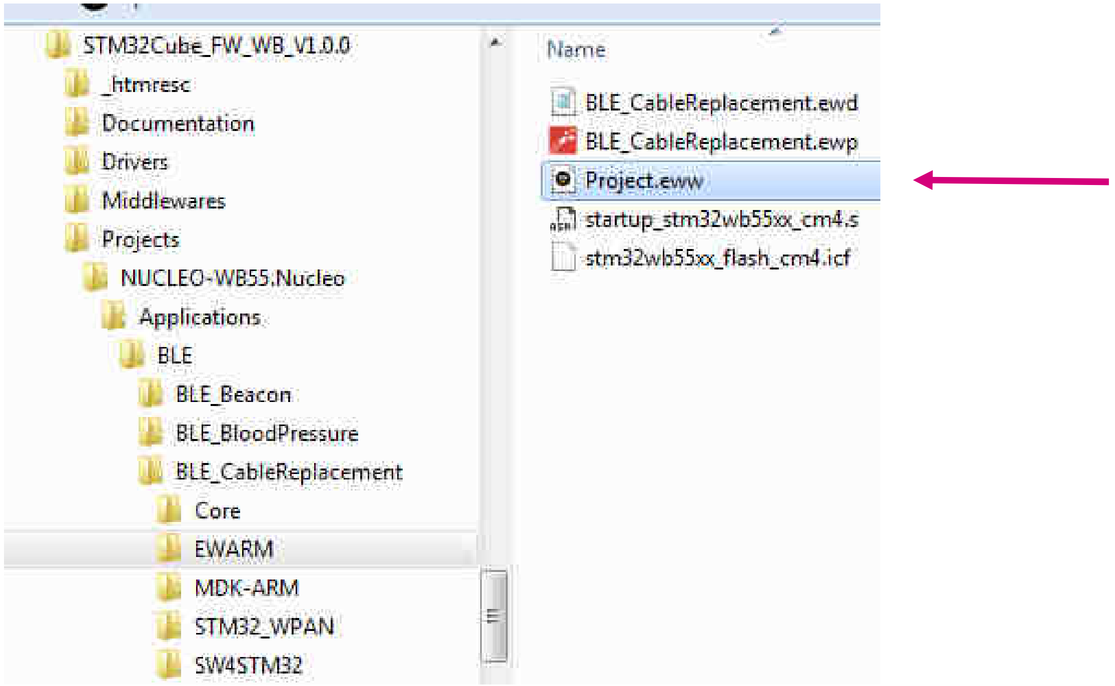
You can copy/paste the code bits from **CableReplacement_Lab.txt** *file from your install files* **Labs** *folder*

BLE_CableReplacement Project

- Open the workspace



STM32Cube_FW_WB_V1.0.0 ▸ Projects ▸ NUCLEO-WB55.Nucleo ▸ Applications ▸ BLE ▸ BLE_CableReplacement ▸ EWARM ▸



STM32Cube_FW_WB_V1.0.0
_htmresc
Documentation
Drivers
Middlewares
Projects
NUCLEO-WB55.Nucleo
Applications
BLE
BLE_Beacon
BLE_BloodPressure
BLE_CableReplacement
Core
EWARM
MDK-ARM
STM32_WPAN
SW4STM32

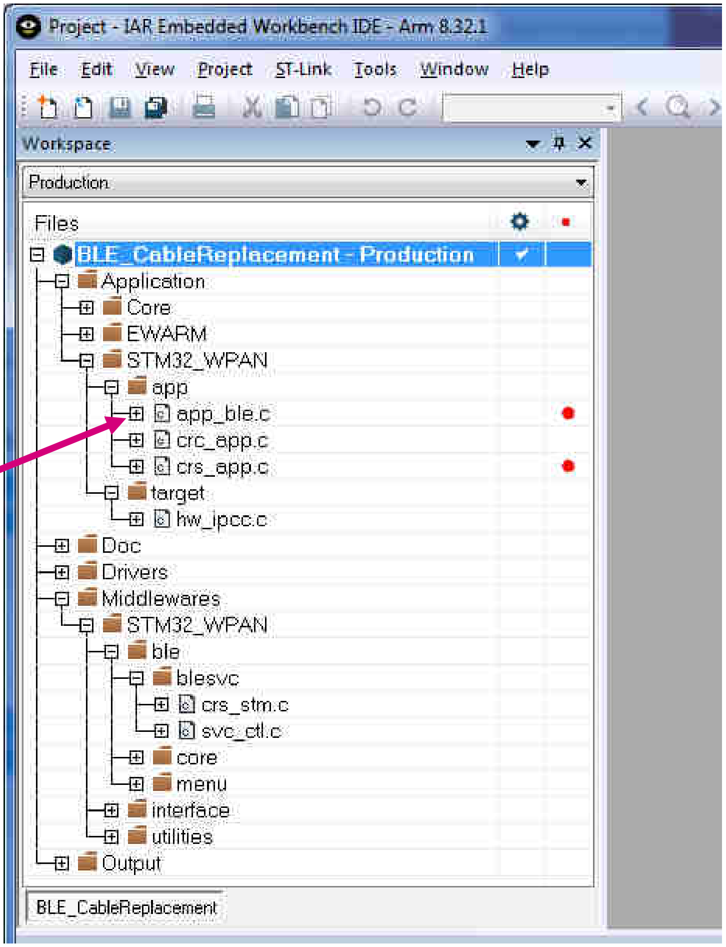
Name
BLE_CableReplacement.ewd
BLE_CableReplacement.ewp
Project.eww
startup_stm32wb55xx_cm4.s
stm32wb55xx_flash_cm4.icf

Build the Project

Open the following files

- app_conf.h
- app_ble.c – Under Application/STM32_WPAN/app
- ble_conf.h
- crs_stm.h
- crs_stm.c
- crs_app.c

To see the header files, expand the C source files



BLE Role Configuration

STM32WB is the GAP Peripheral / GATT server

Smartphone is the GAP Central / GATT client.

- Compile for GATT Server
  - #define @ line# 100 of **app_conf.h**

```
#define GATT_CLIENT         0    /* 1 = Device is GATT Client, 0 = Device is GATT Server */
```

Identify your unique device with your magic number

- Modify your local name (line# 204 of app_ble.c)

```
static const char local_name[] = { AD_TYPE_COMPLETE_LOCAL_NAME, 'C', 'R', 'S', '0', '1' };
```

- Modify your BLE device name (line# 819 of app_ble.c)

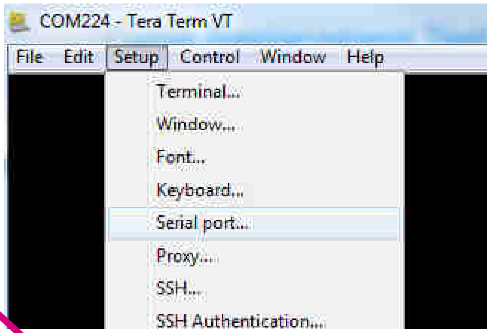
```
const char *name = “BLE-CRS-01";
```

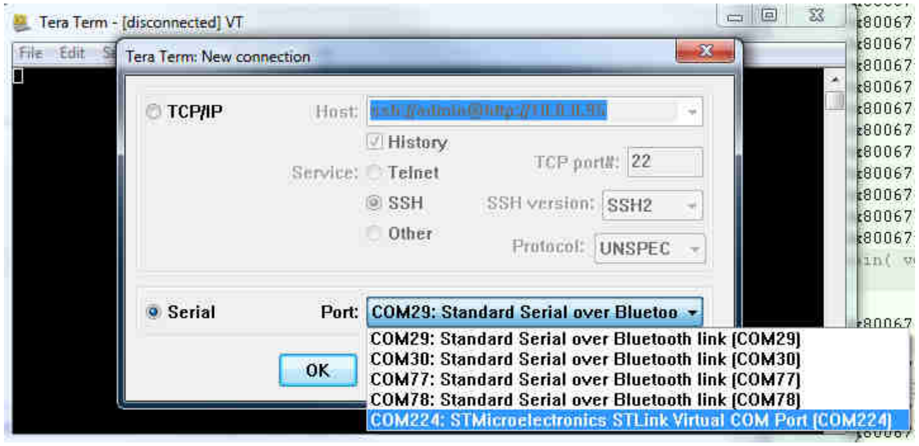
- Ensure that the BLE device name length in ASCII chars matches (line# 165 of app_ble.c)

```
#define APPBLE_GAP_DEVICE_NAME_LENGTH 10
```

Cable Replacement Test

- Build and Run the project
- Connect your TeraTerm to the Nucleo’s STLink Virtual COM port
- Configure your Serial port for 115,200bps / N / 8 / 1


COM224 - Tera Term VT


Tera Term - [disconnected] VT

Tera Term: New connection

○ TCP/IP Host: ssh://admin@http://10.0.0.96

☑ History

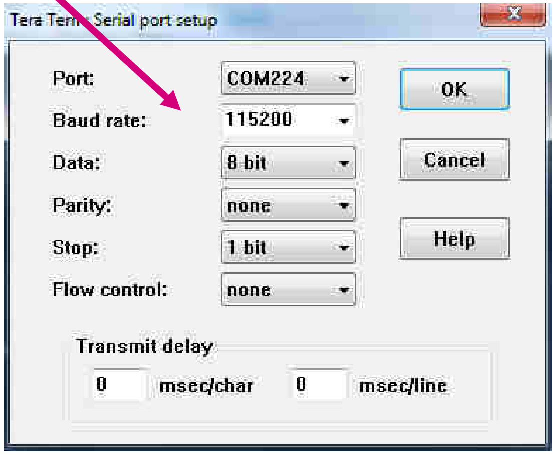
Service: ○ Telnet TCP port#: 22

● SSH SSH version: SSH2

○ Other Protocol: UNSPEC

● Serial Port: COM29: Standard Serial over Bluetoo

COM29: Standard Serial over Bluetooth link (COM29)
COM30: Standard Serial over Bluetooth link (COM30)
COM77: Standard Serial over Bluetooth link (COM77)
COM78: Standard Serial over Bluetooth link (COM78)
COM224: STMicroelectronics STLink Virtual COM Port (COM224)


Tera Term: Serial port setup

Port: COM224

Baud rate: 115200

Data: 8 bit

Parity: none

Stop: 1 bit

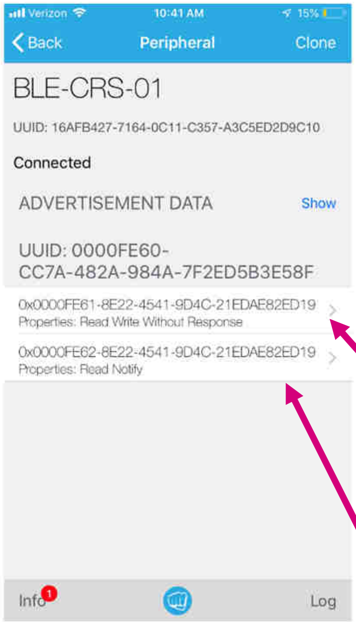
Flow control: none

Transmit delay

0 msec/char 0 msec/line

Cable Replacement Test

- Connect to your device with LightBlue Explorer
- Send and receive ASCII-based messages using the different characteristics



BLE-CRS-01

UUID: 16AFB427-7164-0C11-C357-A3C5ED2D9C10

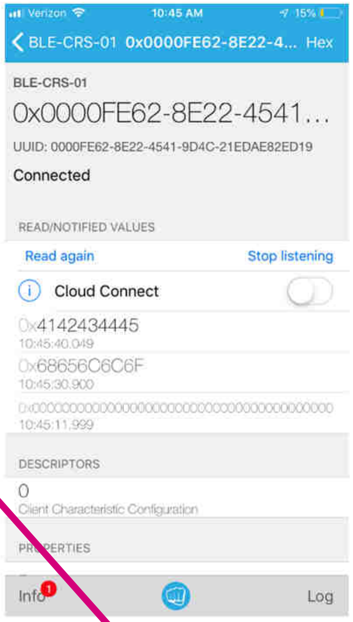
Connected

ADVERTISEMENT DATA Show

UUID: 0000FE60-CC7A-482A-984A-7F2ED5B3E58F

0x0000FE61-8E22-4541-9D4C-21EDAE82ED19
Properties: Read Write Without Response

0x0000FE62-8E22-4541-9D4C-21EDAE82ED19
Properties: Read Notify



BLE-CRS-01

0x0000FE62-8E22-4541...

UUID: 0000FE62-8E22-4541-9D4C-21EDAE82ED19

Connected

READ/NOTIFIED VALUES

Read again Stop listening

Cloud Connect

0x4142434445
10:45:40.049

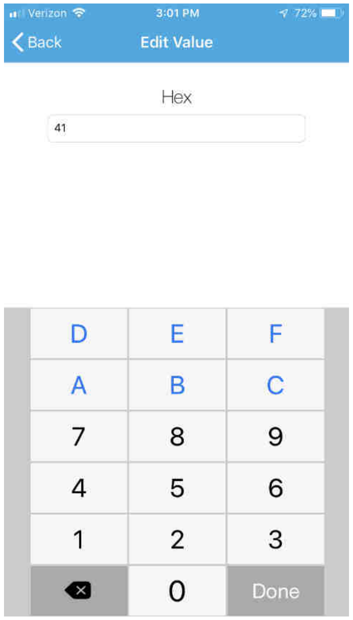
0x68656C6C6F
10:45:30.900

0x0000000000000000000000000000000000000000
10:45:11.999

DESCRIPTORS

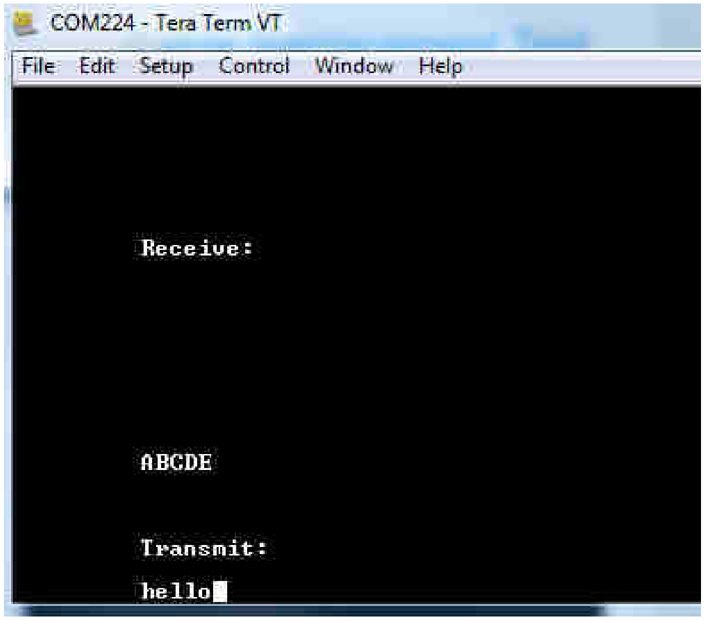
0
Client Characteristic Configuration

PROPERTIES



Hex

41



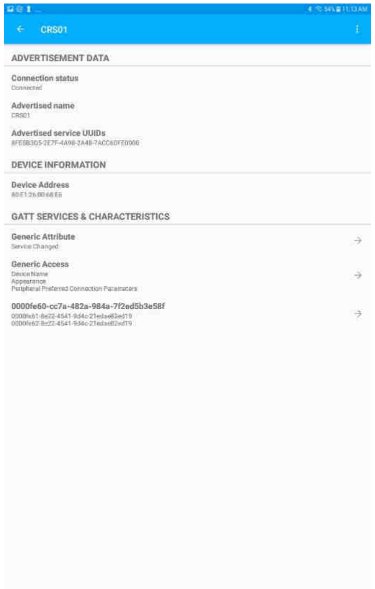
```
Receive:

ABCDE

Transmit:
hello
```

Write values to Nucleo

Enable Notifications to receive messages from Nucleo

Cable Replacement Test

- Here is the LightBlue Explorer on Android



CRS01

ADVERTISEMENT DATA

Connection status
Connected

Advertised name
CRS01

Advertised service UUIDs
8FE5B3D5-2E7F-4A98-2A48-7ACC60FE0000

DEVICE INFORMATION

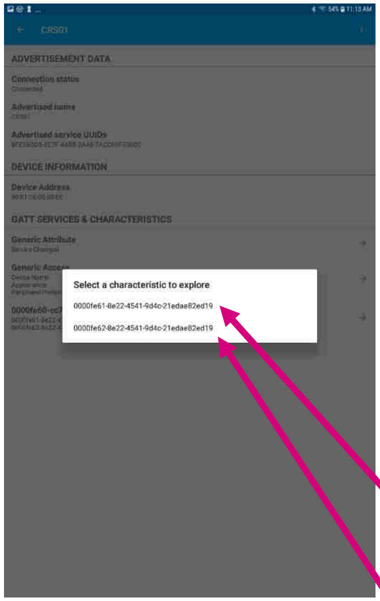
Device Address
80:E1:26:00:68:E6

GATT SERVICES & CHARACTERISTICS

Generic Attribute
Service Changed

Generic Access
Device Name
Appearance
Peripheral Preferred Connection Parameters

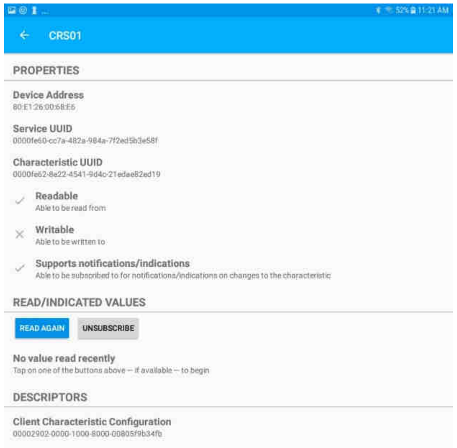
0000fe60-cc7a-482a-984a-7f2ed5b3e58f
0000fe61-8e22-4541-9d4c-21edae82ed19
0000fe62-8e22-4541-9d4c-21edae82ed19



Select a characteristic to explore

0000fe61-8e22-4541-9d4c-21edae82ed19

0000fe62-8e22-4541-9d4c-21edae82ed19



CRS01

PROPERTIES

Device Address
80:E1:26:00:68:E6

Service UUID
0000fe60-cc7a-482a-984a-7f2ed5b3e58f

Characteristic UUID
0000fe62-8e22-4541-9d4c-21edae82ed19

✓ Readable
Able to be read from

✗ Writable
Able to be written to

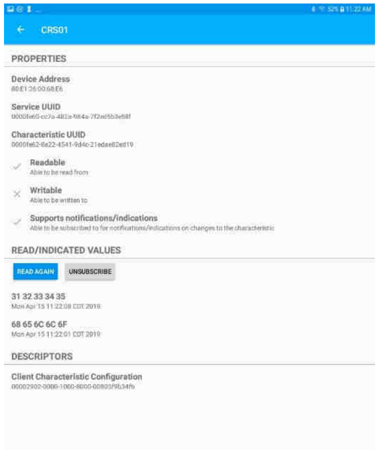
✓ Supports notifications/indications
Able to be subscribed to for notifications/indications on changes to the characteristic

READ/INDICATED VALUES

No value read recently
Tap on one of the buttons above — if available — to begin

DESCRIPTORS

Client Characteristic Configuration
00002902-0000-1000-8000-00805f9b34fb



CRS01

PROPERTIES

Device Address
80:E1:26:00:68:E6

Service UUID
0000fe60-cc7a-482a-984a-7f2ed5b3e58f

Characteristic UUID
0000fe62-8e22-4541-9d4c-21edae82ed19

✓ Readable
Able to be read from

✗ Writable
Able to be written to

✓ Supports notifications/indications
Able to be subscribed to for notifications/indications on changes to the characteristic

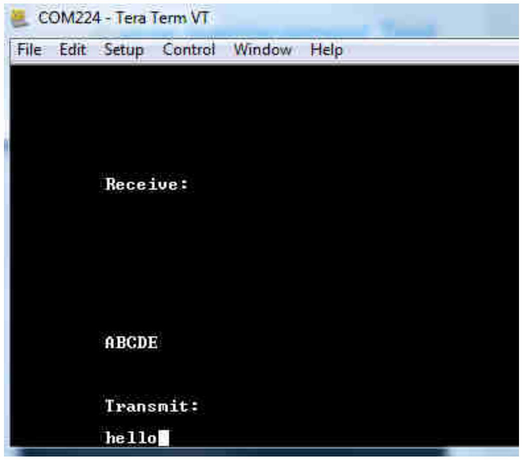
READ/INDICATED VALUES

31 32 33 34 35
Mon Apr 15 11:22:08 CDT 2019

68 65 6C 6C 6F
Mon Apr 15 11:22:01 CDT 2019

DESCRIPTORS

Client Characteristic Configuration
00002902-0000-1000-8000-00805f9b34fb



COM224 - Tera Term VT

```
Receive:


ABCDE

Transmit:
hello
```

Write values to Nucleo

Enable Notifications to receive messages from Nucleo

Add a custom characteristic to an existing Service

- Add the UUID definition (line# 74 of ble_conf.h)

```
#define STM_LED_UUDI128    0x00, 0x00, 0xfe, 0x64, 0x8e, 0x22, 0x45, 0x41, 0x9d, 0x4c, 0x21, 0xed,
0xae, 0x82, 0xed, 0x19
```

- Add event element (line# 37 of crs_stm.h)

```
typedef enum {
   STM_LED_WRITE_EVT,
   CRS_NOTIFY_ENABLED_EVT,
…
} CRS_Opcode_evt_t;
```

Note that these files may be read-only and may need permissions changed

Custom Characteristic

- Add characteristic handle (line# 32 of crs_stm.c)

```
typdef struct {
  …
  uint16_t CRSRXCharHdle;
  uint16_t LedWriteClientToServerCharHdle;
} CRSContext_t;
```

Note that these files may be read-only and may need permissions changed

- Check for the handle (line# 122 of crs_stm.c)

```
case EVT_BLUE_GATT_ATTRIBUTE_MODIFIED:
{
  attribute_modified = (aci_gatt_attribute_modified_event_rp0*)blue_evt->data;
  if(attribute_modified->Attr_Handle == (CRSContext.LedWriteClientToServerCharHdle + 1))
  {
    Notification.CRS_Evt_Opcode = STM_LED_WRITE_EVT;
    Notification.DataTransfered.Length = attribute_modified->Attr_Data_Length;
    Notification.DataTransfered.pPayload = attribute_modified->Attr_Data;
    CRSAPP_Notification(&Notification);
  }
```

Custom Characteristic

- Add uuid array (line# 193 of crs_stm.c)

```
uint8_t led_uuid[]     = { STM_LED_UUDI128 };
```

- Change the Max_Attribute_Records parameter (line# 215 of crs_stm.c)

```
hciCmdResult = aci_gatt_add_service(
                                UUID_TYPE_128,
                                (Service_UUID_t *) &uuid,
                                PRIMARY_SERVICE,
                                8,
                                &(CRSContext.SvcHdle));
```

Custom Characteristic

- Add LED characteristic (line# 281 of crs_stm.c)

```
COPY_CRS_UUID(uuid.Char_UUID_128, led_uuid);
hciCmdResult = aci_gatt_add_char(CRSContext.SvcHdle,
                                 UUID_TYPE_128,
                                 &uuid,
                                 2,   /* Char_Value_Length */
                                 CHAR_PROP_WRITE_WITHOUT_RESP,
                                 ATTR_PERMISSION_NONE,
                                 GATT_NOTIFY_ATTRIBUTE_WRITE,/* gattEvtMask*/
                                 10, /* encryKeySize */
                                 1,  /* isVariable */
                                 &(CRSContext.LedWriteClientToServerCharHdle));
```

Custom Characteristic

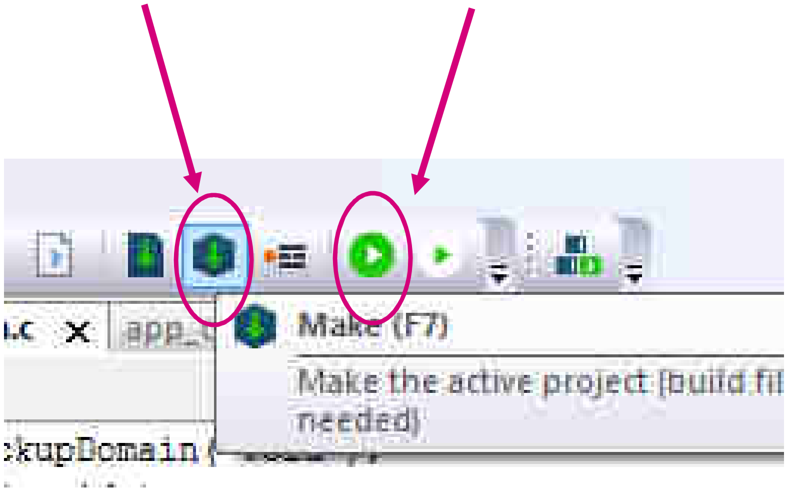
- Add event action (line# 194 of crs_app.c)

```
case STM_LED_WRITE_EVT:
      if(pNotification->DataTransfered.pPayload[0] == 0x01)
      {
        BSP_LED_On(LED_BLUE);
      }
      if(pNotification->DataTransfered.pPayload[0] == 0x00)
      {
        BSP_LED_Off(LED_BLUE);
      }
      break;
```

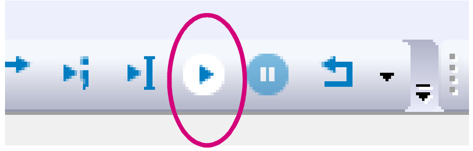
Compile and Program

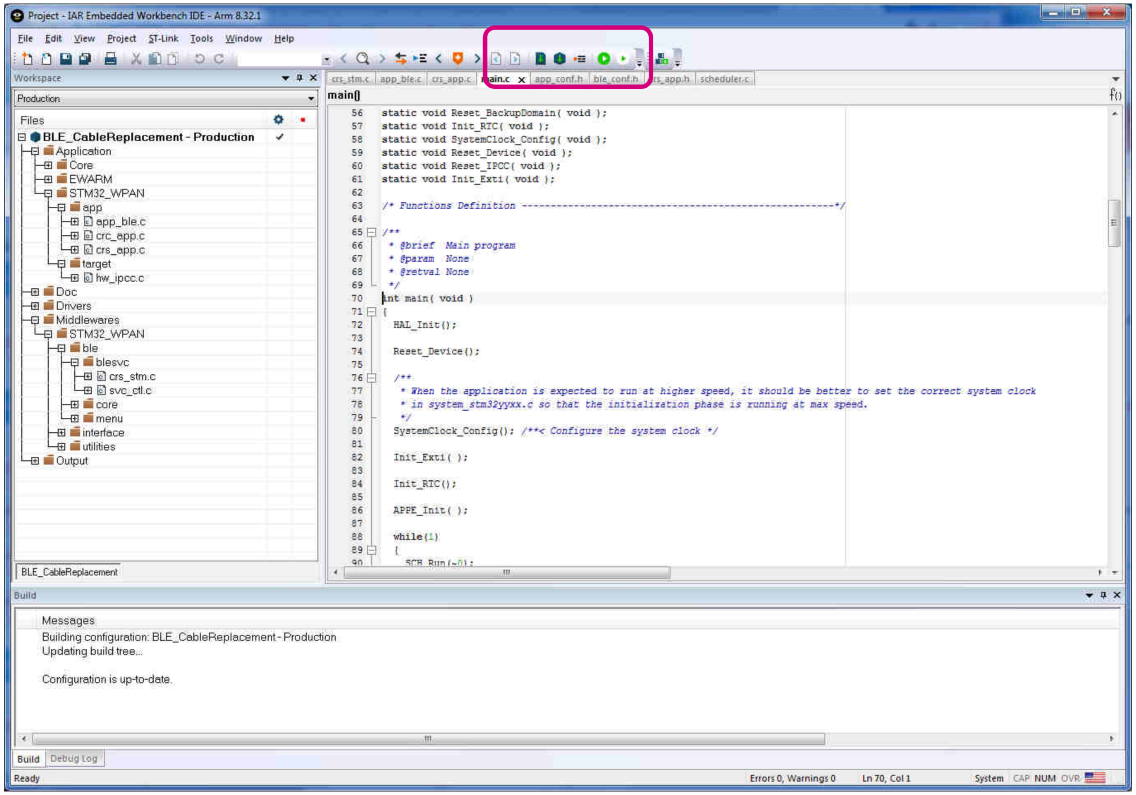
Build Project

Download and Debug



Run





```c
static void Reset_BackupDomain( void );
static void Init_RTC( void );
static void SystemClock_Config( void );
static void Reset_Device( void );
static void Reset_IPCC( void );
static void Init_Exti( void );

/* Functions Definition ------------------------------------------------------*/

/**
 * @brief  Main program
 * @param  None
 * @retval None
 */
int main( void )
{
  HAL_Init();

  Reset_Device();

  /**
   * When the application is expected to run at higher speed, it should be better to set the correct system clock
   * in system_stm32yyxx.c so that the initialization phase is running at max speed.
   */
  SystemClock_Config(); /**< Configure the system clock */

  Init_Exti( );

  Init_RTC();

  APPE_Init( );

  while(1)
  {
    SCH_Run(~0);
```

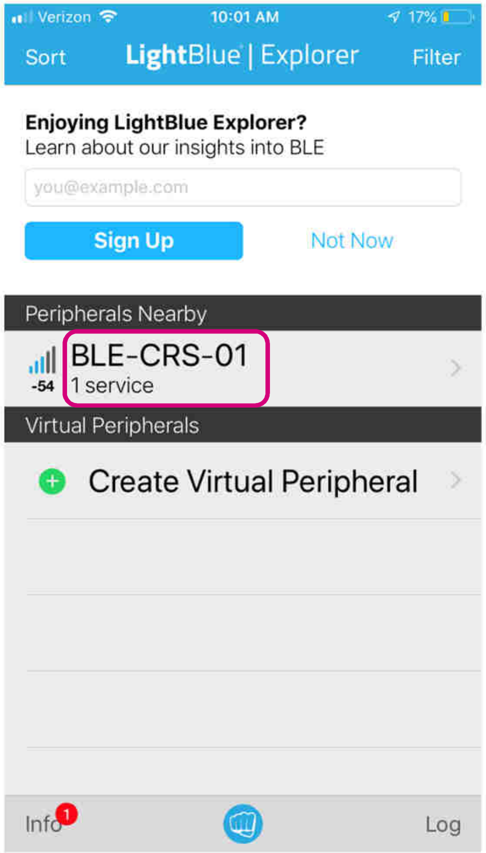
```
Messages
Building configuration: BLE_CableReplacement - Production
Updating build tree...

Configuration is up-to-date.
```

Start LightBlue App

- Launch the LightBlue app

- Find your device

- Select your device



Verizon 10:01 AM 17%

Enjoying LightBlue Explorer?
Learn about our insights into BLE

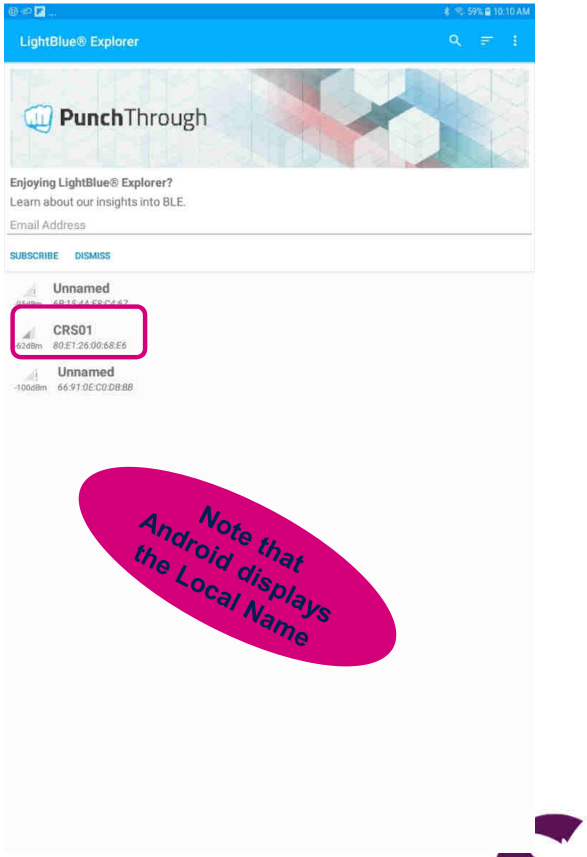
you@example.com

Peripherals Nearby

BLE-CRS-01
1 service
-54

Virtual Peripherals

Create Virtual Peripheral



59% 10:10 AM

Enjoying LightBlue® Explorer?
Learn about our insights into BLE.

Email Address

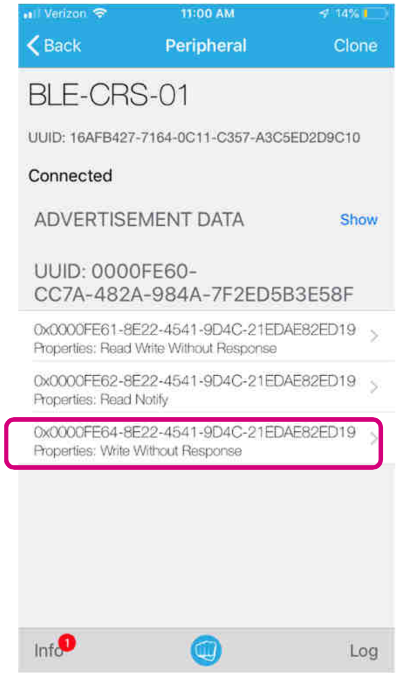
Unnamed
-95dBm 6B:15:4A:E8:C4:67

CRS01
-62dBm 80:E1:26:00:68:E6

Unnamed
-100dBm 66:91:0E:C0:DB:BB

Note that Android displays the Local Name

Find your LED characteristic UUID



BLE-CRS-01

UUID: 16AFB427-7164-0C11-C357-A3C5ED2D9C10

Connected

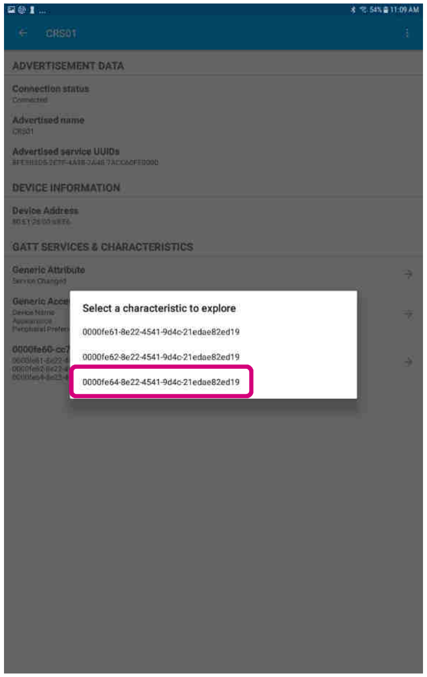
ADVERTISEMENT DATA

UUID: 0000FE60-CC7A-482A-984A-7F2ED5B3E58F

0x0000FE61-8E22-4541-9D4C-21EDAE82ED19
Properties: Read Write Without Response

0x0000FE62-8E22-4541-9D4C-21EDAE82ED19
Properties: Read Notify

0x0000FE64-8E22-4541-9D4C-21EDAE82ED19
Properties: Write Without Response



ADVERTISEMENT DATA

Connection status
Connected

Advertised name
CRS01

Advertised service UUIDs

DEVICE INFORMATION

Device Address

GATT SERVICES & CHARACTERISTICS

Generic Attribute
Service Changed

Generic Acce
Device Name
Appearance
Peripheral Prefer

0000fe60-cc7

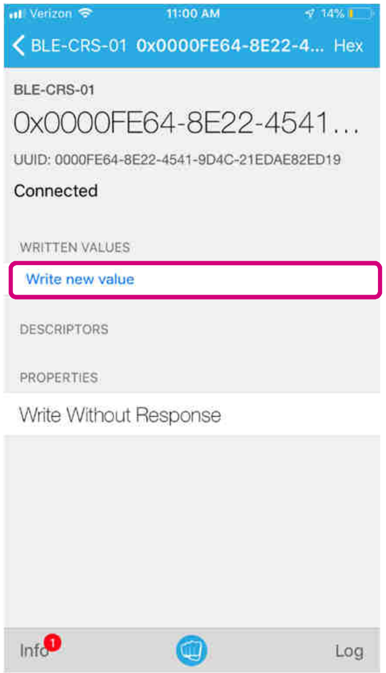
Select a characteristic to explore

0000fe61-8e22-4541-9d4c-21edae82ed19

0000fe62-8e22-4541-9d4c-21edae82ed19

0000fe64-8e22-4541-9d4c-21edae82ed19

Write new value



BLE-CRS-01

0x0000FE64-8E22-4541...

UUID: 0000FE64-8E22-4541-9D4C-21EDAE82ED19

Connected

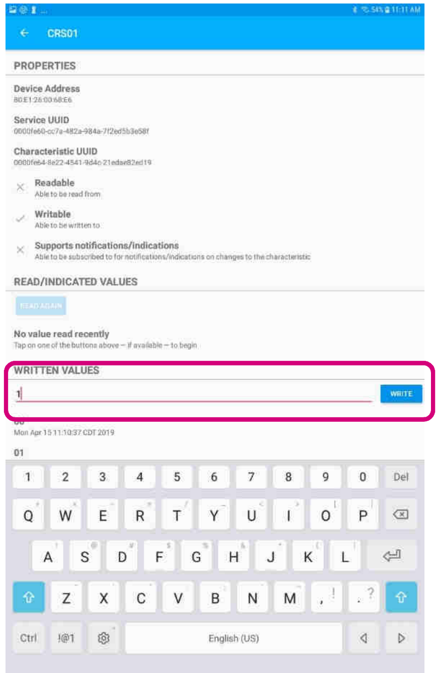
WRITTEN VALUES

Write new value

DESCRIPTORS

PROPERTIES

Write Without Response



PROPERTIES

Device Address
80:E1:26:00:68:E6

Service UUID
0000fe60-cc7a-482a-984a-7f2ed5b3e58f

Characteristic UUID
0000fe64-8e22-4541-9d4c-21edae82ed19

✗ Readable
Able to be read from

✓ Writable
Able to be written to

✗ Supports notifications/indications
Able to be subscribed to for notifications/indications on changes to the characteristic

READ/INDICATED VALUES

No value read recently
Tap on one of the buttons above – if available – to begin

WRITTEN VALUES

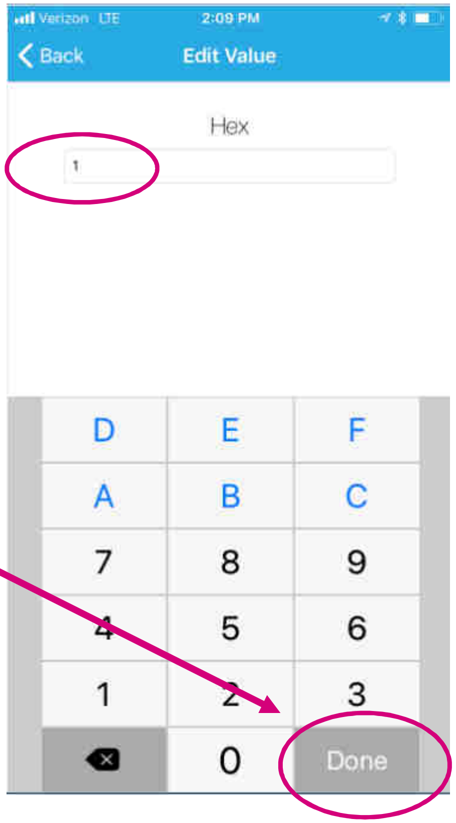
1

Mon Apr 15 11:10:37 CDT 2019

01

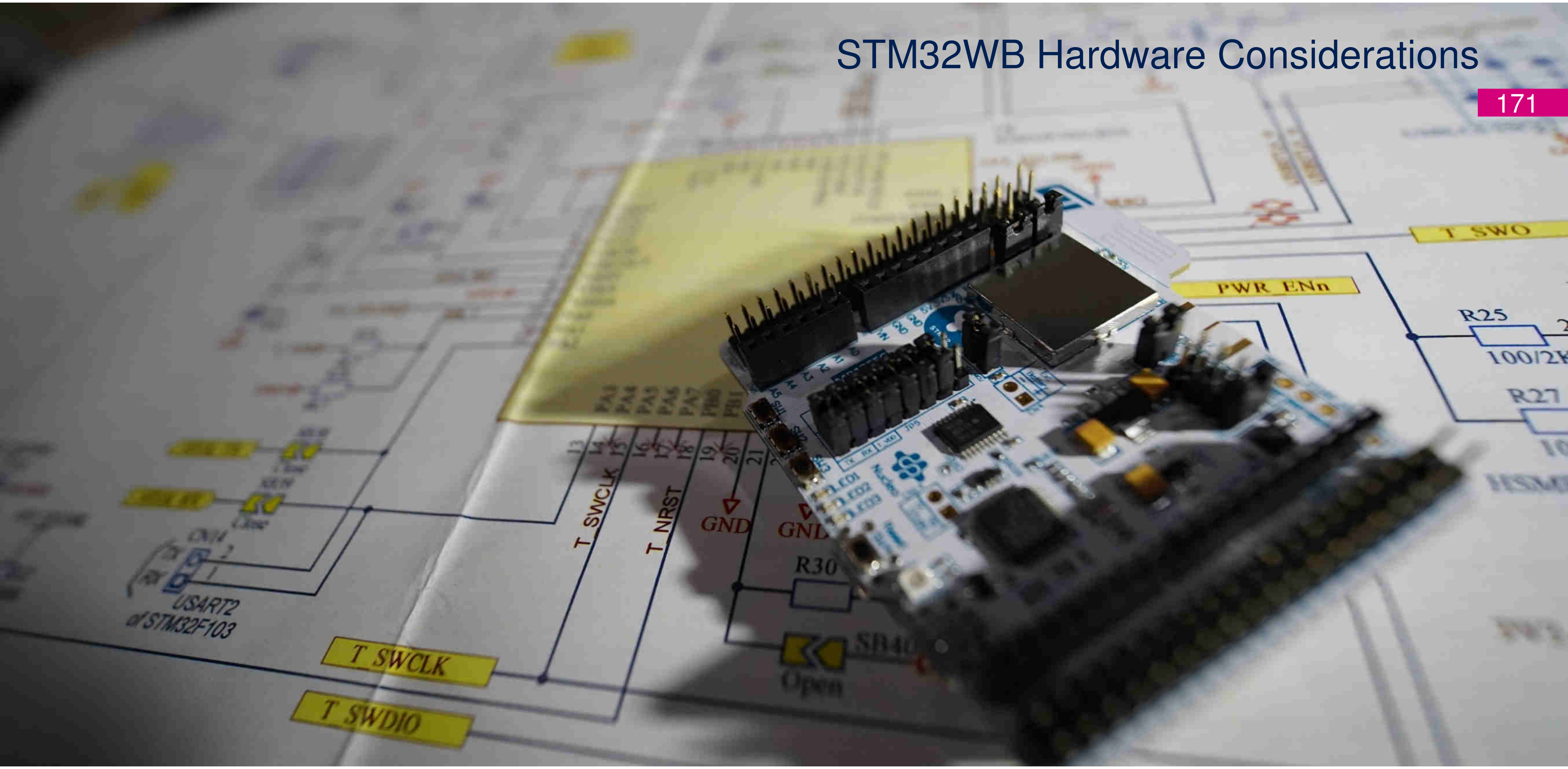
Write a value

- LED ON = 1
- LED OFF = 0



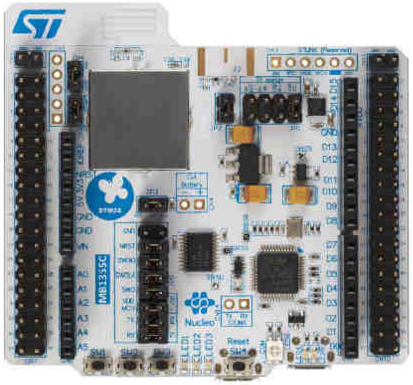
Concurrently, the CableReplacement characteristics can also be used

STM32WB Hardware Considerations

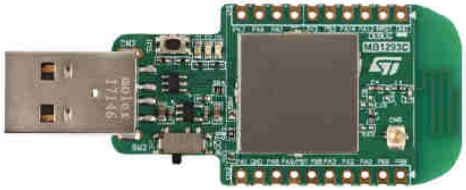


Delivery State

RSS + BLE Stack

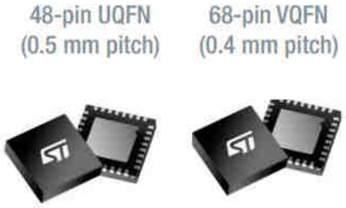


RSS + BLE Stack



RSS only

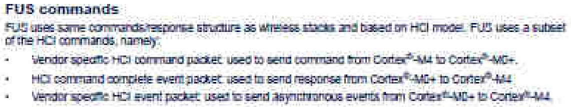
Stack must be loaded



AN5185: Firmware Update Services

AN5185 details the sequence to create your own secure stack loader project, running on the M4

- Command / Response HCI event transactions to the M0+ similar to BLE



FUS commands

FUS uses same commands/response structure as wireless stacks and based on HCI model. FUS uses a subset of the HCI commands, namely:

- Vendor specific HCI command packet: used to send command from Cortex®-M4 to Cortex®-M0+.
- HCI command complete event packet: used to send response from Cortex®-M0+ to Cortex®-M4
- Vendor specific HCI event packet: used to send asynchronous events from Cortex®-M0+ to Cortex®-M4.

Figure 6. FUS HCI subset

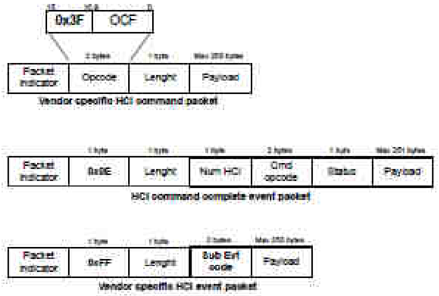
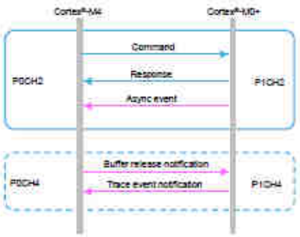
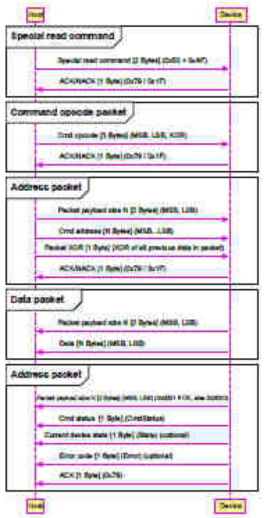


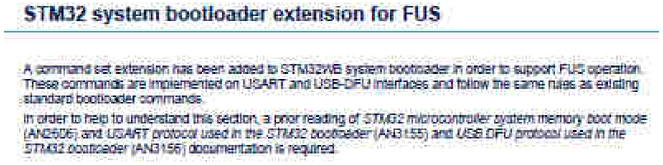
Figure 5. IPCC channels used by FUS



Also details on the bootloader sequences used

Figure 10. USART special read command


Host
Device
Special read command
Special read command [2 Bytes] (0x50 + 0xAF)
ACK/NACK [1 Byte] (0x79 / 0x1F)
Command opcode packet
Cmd opcode [3 Bytes] (MSB, LSB, XOR)
ACK/NACK [1 Byte] (0x79 / 0x1F)
Address packet
Packet payload size N [2 Bytes] (MSB, LSB)
Cmd address [N Bytes] (MSB...LSB)
Packet XOR [1 Byte] (XOR of all previous data in packet)
ACK/NACK [1 Byte] (0x79 / 0x1F)
Data packet
Packet payload size N [2 Bytes] (MSB, LSB)
Data [N Bytes] (MSB, LSB)
Address packet
Packet payload size N [2 Bytes] (MSB, LSB) (0x0001 if OK, else 0x0003)
Cmd status [1 Byte] (CmdStatus)
Current device state [1 Byte] (State) (optional)
Error code [1 Byte] (Error) (optional)
ACK [1 Byte] (0x79)
Host
Device



STM32 system bootloader extension for FUS

A command set extension has been added to STM32WB system bootloader in order to support FUS operation. These commands are implemented on USART and USB-DFU interfaces and follow the same rules as existing standard bootloader commands.

In order to help to understand this section, a prior reading of *STM32 microcontroller system memory boot mode (AN2606)* and *USART protocol used in the STM32 bootloader* (AN3155) and *USB DFU protocol used in the STM32 bootloader* (AN3156) documentation is required.

AN5165: Hardware design

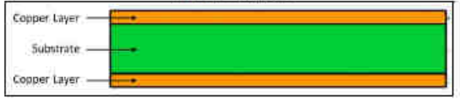
AN51659 details RF hardware considerations

- PCB stackup recommendations
- RF Front-end (discrete or IPD-based)
- SMPS passives selection
- Clocks

2-layer PCB

With the 2-layer PCB (see *Figure 21*), the RF signals and routing are on the top layer while the bottom layer is used for grounding under the RF zones, and for routing in others parts. The ground plane must be continuous under the RF zones, otherwise the return path current can increase and degrade the RF performance.

Figure 21. 2-layer PCB

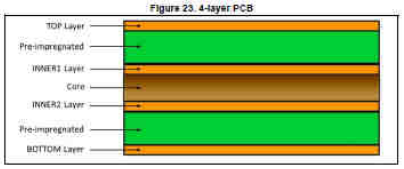


4-layer PCB

With the 4-layer PCB shown in *Figure 23*, it is recommended to have the following distribution:

- TOP layer: RF signal and routing on the top layer.
- INNER1 layer: grounding under the RF zones, routing in the others parts.
- INNER2 layer: power and low frequency routing.
- BOTTOM layer: low frequency routing.

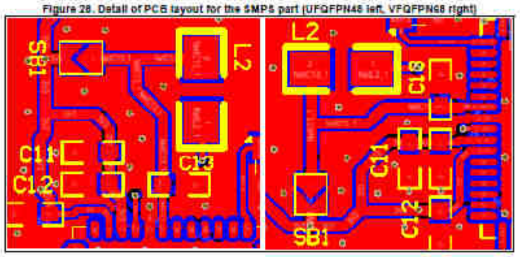
Figure 23. 4-layer PCB



6.1.2 SMPS

In addition to the recommendations given in *Section 4.3: SMPS*, to avoid important current loop when the STM32WB is in SMPS mode, it is recommended to place C11, C12 and C13 as close as possible to their respective pins on STM32WB. Do not forget to connect the solder pad to ground to have a strong current return path.

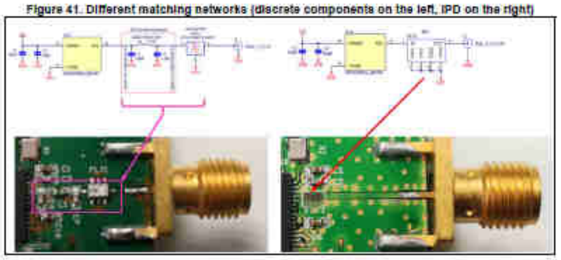
Figure 28. Detail of PCB layout for the SMPS part (UFQFPN48 left, VFQFPN68 right)



8 UFQFN48/VFQFN68 reference boards with IPD

The goal of the IPD (integrated passive device) is to replace the discrete matching network plus the integrated low-pass filter keeping equivalent TX/RX performance. *Figure 41* shows the differences between the two approaches.

Figure 41. Different matching networks (discrete components on the left, IPD on the right)



Layout recommendations for the 2-layer PCB

Figure 25. PCB layout for UFQFPN48 (left to right: all, top and bottom layers)

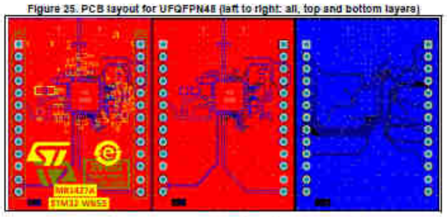


Figure 26. PCB layout for VFQFPN68 (left to right: all, top and bottom layers)

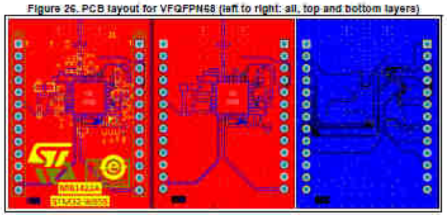
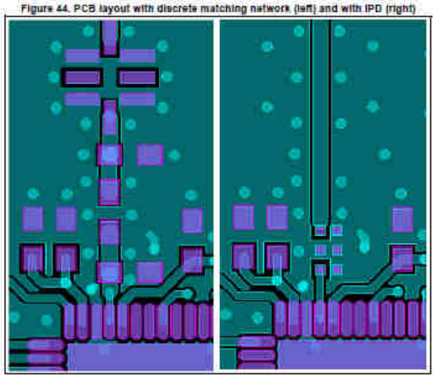


Figure 44. PCB layout with discrete matching network (left) and with IPD (right)



AN5290: Minimal BOM

AN5290 details the minimal Bill-of-Materials needed for various scenarios

Figure 5. Optimized solution with IPD

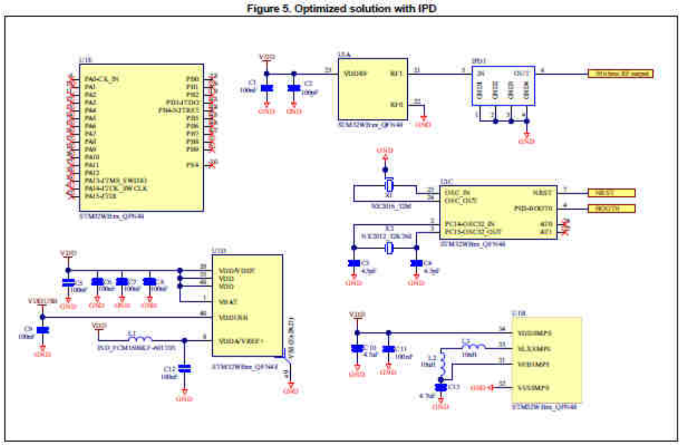
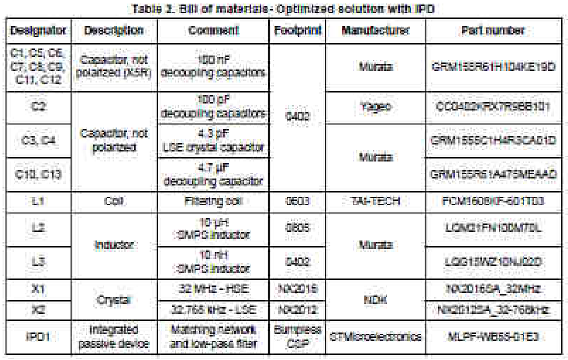


Table 2. Bill of materials - Optimized solution with IPD



Designator	Description	Comment	Footprint	Manufacturer	Part number
C1, C5, C6, C7, C8, C9, C11, C12	Capacitor, not polarized (X5R)	100 nF decoupling capacitors	0402	Murata	GRM155R61H104KE19D
C2	Capacitor, not polarized	100 pF decoupling capacitors		Yageo	CC0402KRX7R9BB101
C3, C4		4.3 pF LSE crystal capacitor		Murata	GRM1555C1H4R3CA01D
C10, C13		4.7 µF decoupling capacitor			GRM155R61A475MEAAD
L1	Coil	Filtering coil	0603	TAI-TECH	FCM1608KF-601T03
L2	Inductor	10 µH SMPS inductor	0805	Murata	LQM21FN100M70L
L3		10 nH SMPS inductor	0402		LQG15WZ10NJ02D
X1	Crystal	32 MHz - HSE	NX2016	NDK	NX2016SA_32MHz
X2		32.768 kHz - LSE	NX2012		NX2012SA_32-768kHz
IPD1	Integrated passive device	Matching network and low-pass filter	Bumpless CSP	STMicroelectronics	MLPF-WB55-01E3

Figure 4. Optimized solution with discrete components

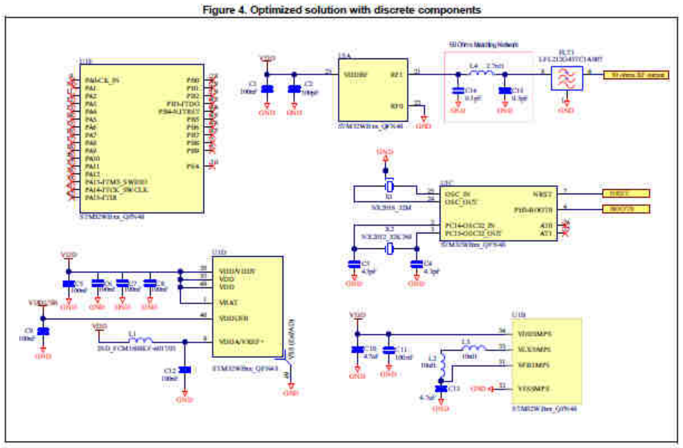
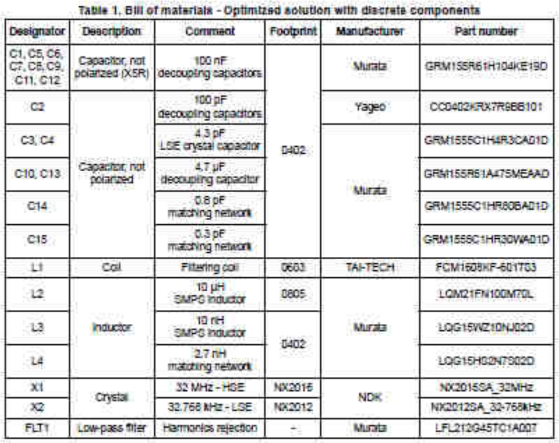


Table 1. Bill of materials - Optimized solution with discrete components



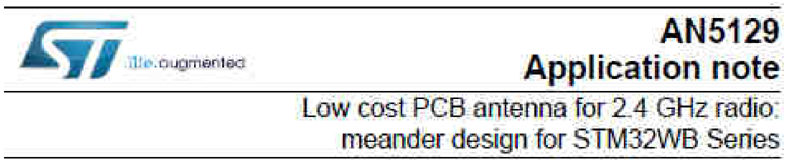
Designator	Description	Comment	Footprint	Manufacturer	Part number
C1, C5, C6, C7, C8, C9, C11, C12	Capacitor, not polarized (X5R)	100 nF decoupling capacitors	0402	Murata	GRM155R61H104KE19D
C2	Capacitor, not polarized	100 pF decoupling capacitors		Yageo	CC0402KRX7R9BB101
C3, C4		4.3 pF LSE crystal capacitor		Murata	GRM1555C1H4R3CA01D
C10, C13		4.7 µF decoupling capacitor			GRM155R61A475MEAAD
C14		0.8 pF matching network			GRM1555C1HR80BA01D
C15		0.3 pF matching network			GRM1555C1HR30WA01D
L1	Coil	Filtering coil	0603	TAI-TECH	FCM1608KF-601T03
L2	Inductor	10 µH SMPS inductor	0805	Murata	LQM21FN100M70L
L3		10 nH SMPS inductor	0402		LQG15WZ10NJ02D
L4		2.7 nH matching network			LQG15HS2N7S02D
X1	Crystal	32 MHz - HSE	NX2016	NDK	NX2016SA_32MHz
X2		32.768 kHz - LSE	NX2012		NX2012SA_32-768kHz
FLT1	Low-pass filter	Harmonics rejection	-	Murata	LFL212G45TC1A007

life.augmented

STM32 WB

AN5129: PCB Antenna design

AN5129 details a “meander-style” PCB antenna design



AN5129
Application note

Low cost PCB antenna for 2.4 GHz radio:
meander design for STM32WB Series

life.augmented

Figure 3. PCB cross section at antennae area

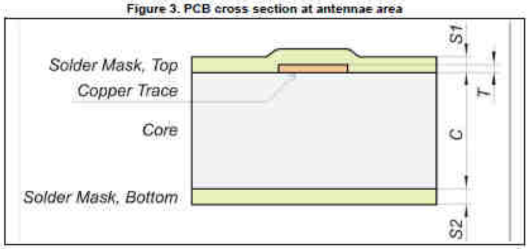
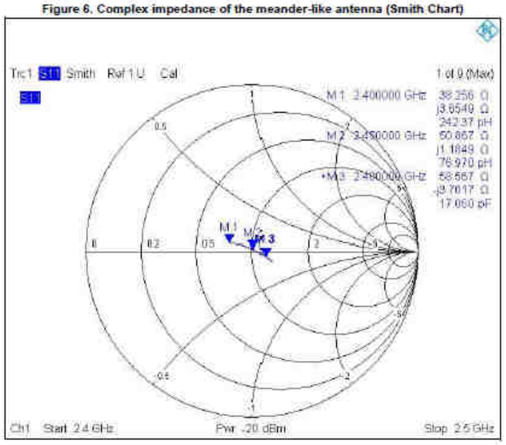
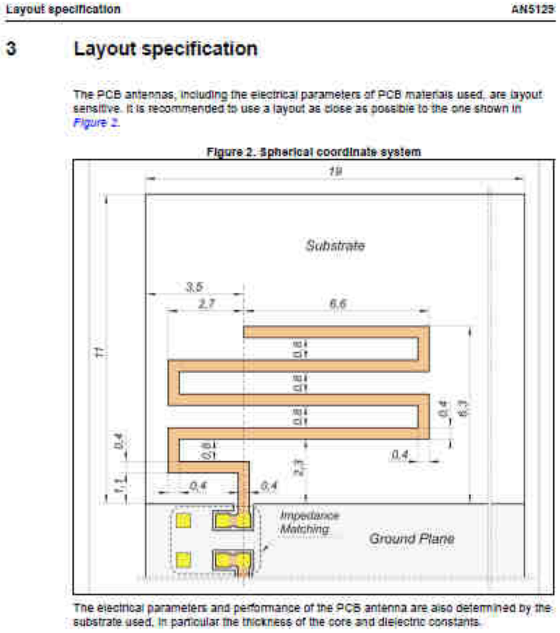


Figure 6. Complex impedance of the meander-like antenna (Smith Chart)





Layout specification AN5129

3 Layout specification

The PCB antennas, including the electrical parameters of PCB materials used, are layout sensitive. It is recommended to use a layout as close as possible to the one shown in *Figure 2*.

Figure 2. Spherical coordinate system

The electrical parameters and performance of the PCB antenna are also determined by the substrate used, in particular the thickness of the core and dielectric constants.

AN5246: SMPS

AN5246 details SMPS use cases, component selection, and various typical operating parametrics

AN5246
Application note

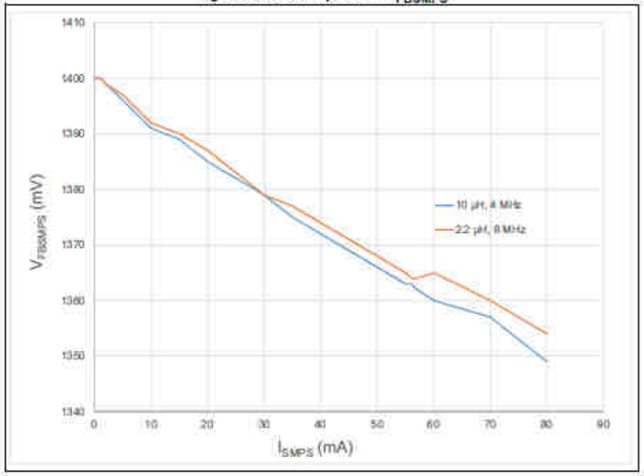
Usage of SMPS on STM32WB Series microcontrollers

Introduction

This document describes how the use the SMPS (switched mode power supply) integrated in microcontrollers of the STM32WB Series. It is intended to be used by system architects and by HW and board-level SW developers.

The patented implementation detailed in this document differs from the standard ones because it is able to maintain the RF transceiver full performance while, at the same time, providing the best power figure in burst application like those generally used by Bluetooth® Low Energy and IEEE 802.15.4 protocols.

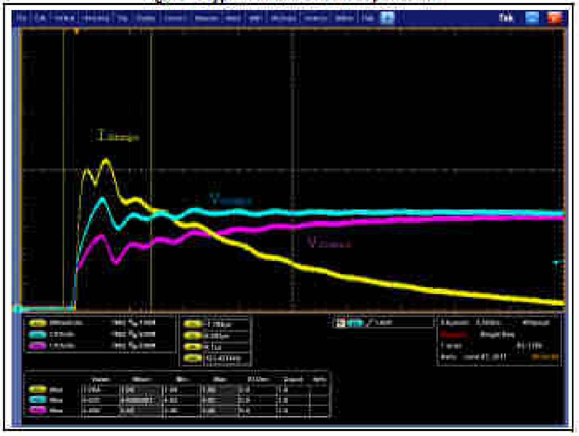
Figure 2. Load impact on V_{FBSMPS}



Inrush current at power ON

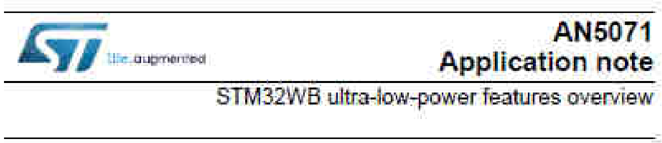
As the SMPS starts in BYPASS mode when powering up, the bulk capacitance needs to be powered when V_{DD} rises. At start up, when the V_{DD} voltage enters the 0.7 to 1 V range, the SMPS PMOS starts to conduce and V_{FBSMPS} follows V_{DDSMPS}. This leads to a temporary inrush current that can be as high as 1.1 A if the power supply is strong enough.

Figure 4. Typical inrush current at power-on



AN5071: Ultra-Low Power Design

AN5071 details the multitude of low-power options available on the WB



life.augmented

AN5071

Application note

STM32WB ultra-low-power features overview

Figure 5. Low-power modes possible transitions

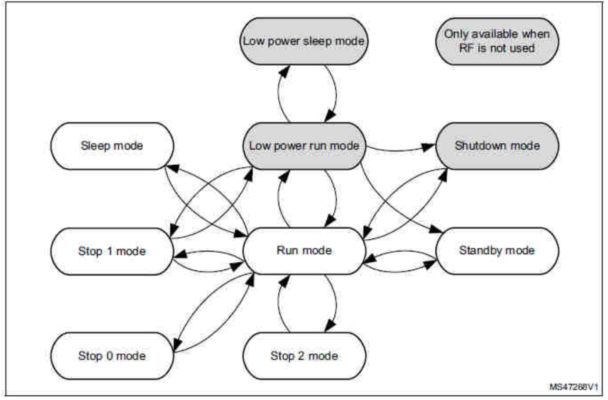


Figure 3. STM32WB55 - Current consumption for different memory configurations

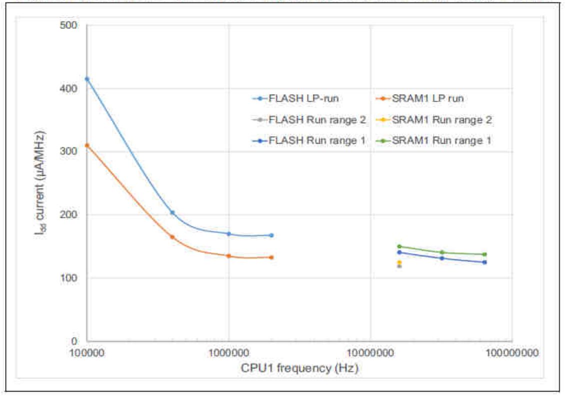


Figure 2. Power distribution architecture

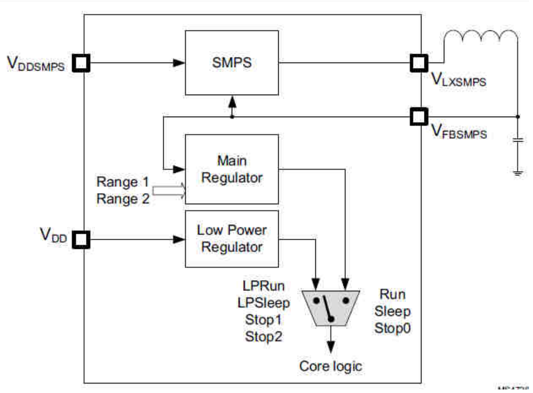
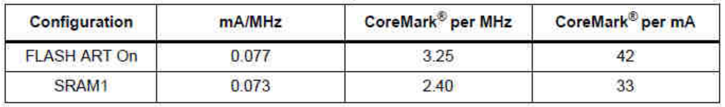


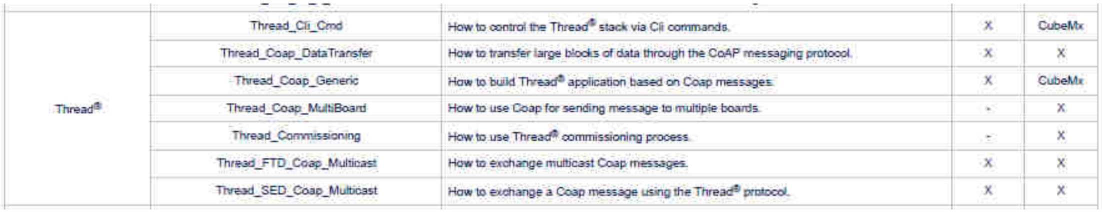
Table 2. STM32WB55 performance with SMPS



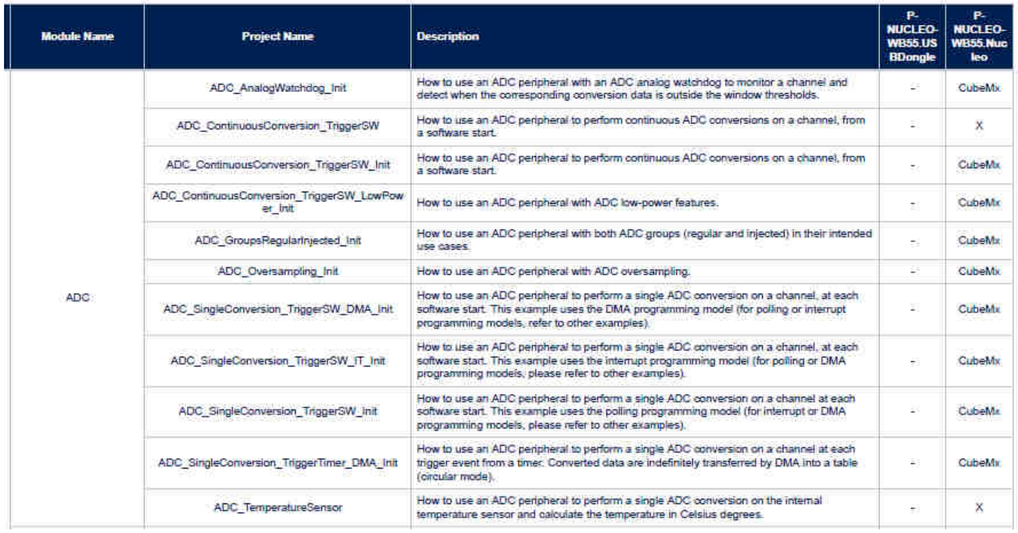
Configuration	mA/MHz	CoreMark® per MHz	CoreMark® per mA
FLASH ART On	0.077	3.25	42
SRAM1	0.073	2.40	33

AN5155: CubeWB Examples

AN5155 is an exhaustive list of all firmware examples and descriptions



Thread®	Thread_Cli_Cmd	How to control the Thread® stack via Cli commands.	X	CubeMx
	Thread_Coap_DataTransfer	How to transfer large blocks of data through the CoAP messaging protocol.	X	X
	Thread_Coap_Generic	How to build Thread® application based on Coap messages.	X	CubeMx
	Thread_Coap_MultiBoard	How to use Coap for sending message to multiple boards.	-	X
	Thread_Commissioning	How to use Thread® commissioning process.	-	X
	Thread_FTD_Coap_Multicast	How to exchange multicast Coap messages.	X	X
	Thread_SED_Coap_Multicast	How to exchange a Coap message using the Thread® protocol.	X	X



Module Name	Project Name	Description	P-NUCLEO-WB55.USBDongle	P-NUCLEO-WB55.Nucleo
ADC	ADC_AnalogWatchdog_Init	How to use an ADC peripheral with an ADC analog watchdog to monitor a channel and detect when the corresponding conversion data is outside the window thresholds.	-	CubeMx
	ADC_ContinuousConversion_TriggerSW	How to use an ADC peripheral to perform continuous ADC conversions on a channel, from a software start.	-	X
	ADC_ContinuousConversion_TriggerSW_Init	How to use an ADC peripheral to perform continuous ADC conversions on a channel, from a software start.	-	CubeMx
	ADC_ContinuousConversion_TriggerSW_LowPower_Init	How to use an ADC peripheral with ADC low-power features.	-	CubeMx
	ADC_GroupsRegularInjected_Init	How to use an ADC peripheral with both ADC groups (regular and injected) in their intended use cases.	-	CubeMx
	ADC_Oversampling_Init	How to use an ADC peripheral with ADC oversampling.	-	CubeMx
	ADC_SingleConversion_TriggerSW_DMA_Init	How to use an ADC peripheral to perform a single ADC conversion on a channel, at each software start. This example uses the DMA programming model (for polling or interrupt programming models, refer to other examples).	-	CubeMx
	ADC_SingleConversion_TriggerSW_IT_Init	How to use an ADC peripheral to perform a single ADC conversion on a channel, at each software start. This example uses the interrupt programming model (for polling or DMA programming models, please refer to other examples).	-	CubeMx
	ADC_SingleConversion_TriggerSW_Init	How to use an ADC peripheral to perform a single ADC conversion on a channel at each software start. This example uses the polling programming model (for interrupt or DMA programming models, please refer to other examples).	-	CubeMx
	ADC_SingleConversion_TriggerTimer_DMA_Init	How to use an ADC peripheral to perform a single ADC conversion on a channel at each trigger event from a timer. Converted data are indefinitely transferred by DMA into a table (circular mode).	-	CubeMx
	ADC_TemperatureSensor	How to use an ADC peripheral to perform a single ADC conversion on the internal temperature sensor and calculate the temperature in Celsius degrees.	-	X

CubeMx denotes that there is an "ioc" CubeMX project file also

AN5292: BLE Mesh

AN5292 shows how to get started using BLE Mesh

Figure 7. Internal project folder

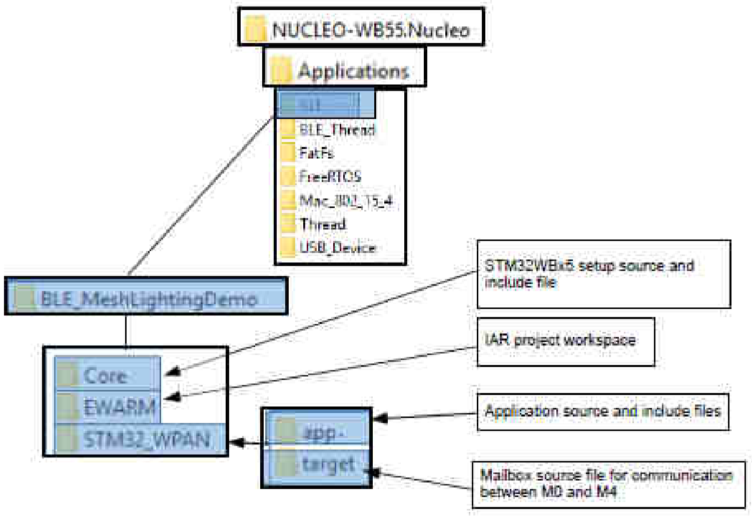
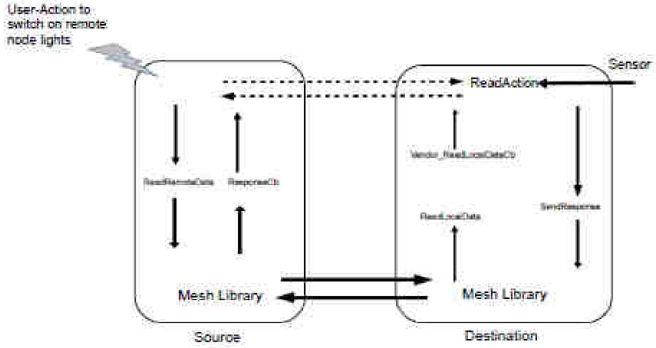
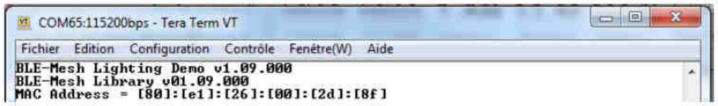


Figure 14. Read command from a remote node



The response data from the node is sent via the `BLEMesh_SendResponse` function.

Figure 10. VCOM window



```
BLE-Mesh Lighting Demo v1.09.000
BLE-Mesh Library v01.09.000
MAC Address = [80]:[e1]:[26]:[00]:[2d]:[8f]
```

MAC address management

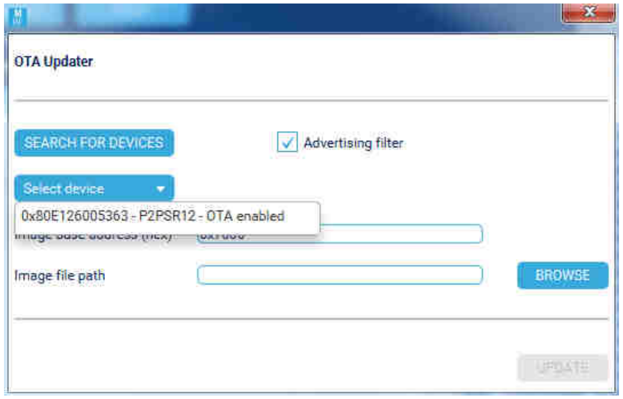
Each node in the mesh network requise a unic MAC address. The following table describes the available options to configure the MAC addresses for a node

Table 2. MAC address management

Number	MAC address Management	Comments
1	Using external MAC address	User can program the nodes with desired unique MAC address. This is stored at specific location in the flash. It is the user's responsibility to make sure that the programmed MAC address in the device is compliant with the Bluetooth communication requirements.
2	Using the unique device serial number	It is possible to configure the MAC address of the device using the unique serial number available in each device. This is the default setting.
3	Using static random MAC address	It is possible to configure the MAC address of devices using the static random MAC address

Bonus Hands-On!

Over-the-Air Firmware Update



OTA Updater

SEARCH FOR DEVICES

☑ Advertising filter

Select device ▾

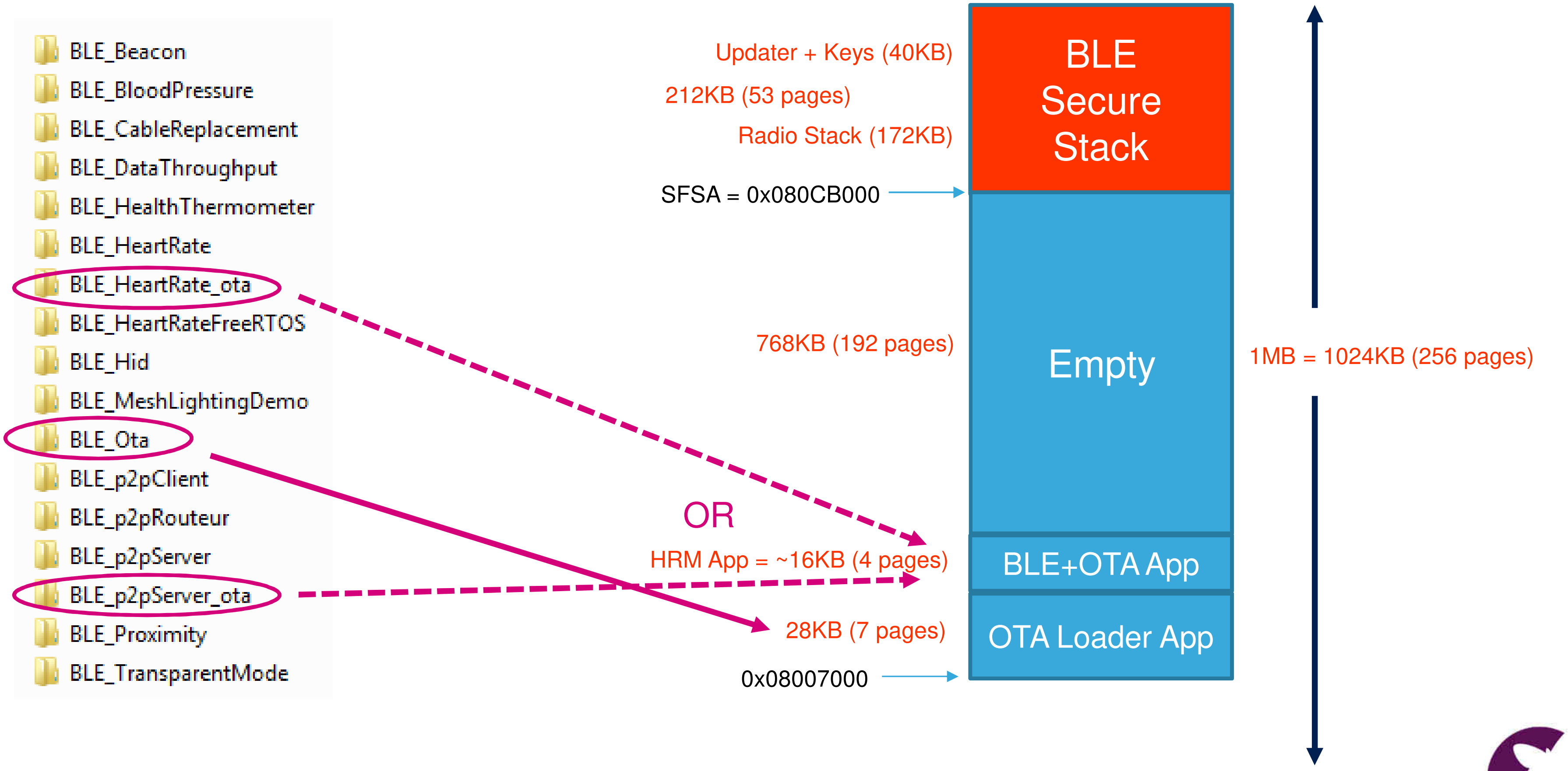
0x80E126005363 - P2PSR12 - OTA enabled

Image file path

BROWSE

UPDATE

Over-The-Air Firmware Updates



- BLE_Beacon
- BLE_BloodPressure
- BLE_CableReplacement
- BLE_DataThroughput
- BLE_HealthThermometer
- BLE_HeartRate
- BLE_HeartRate_ota
- BLE_HeartRateFreeRTOS
- BLE_Hid
- BLE_MeshLightingDemo
- BLE_Ota
- BLE_p2pClient
- BLE_p2pRouteur
- BLE_p2pServer
- BLE_p2pServer_ota
- BLE_Proximity
- BLE_TransparentMode

User Flash (1MB)

Updater + Keys (40KB)

212KB (53 pages)

Radio Stack (172KB)

BLE Secure Stack

SFSA = 0x080CB000

768KB (192 pages)

Empty

1MB = 1024KB (256 pages)

OR

HRM App = ~16KB (4 pages)

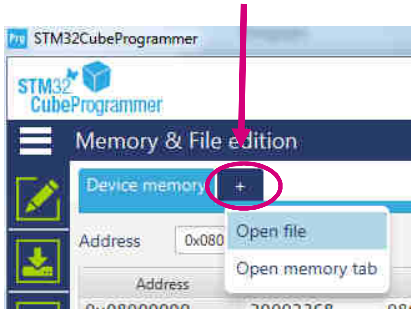
BLE+OTA App

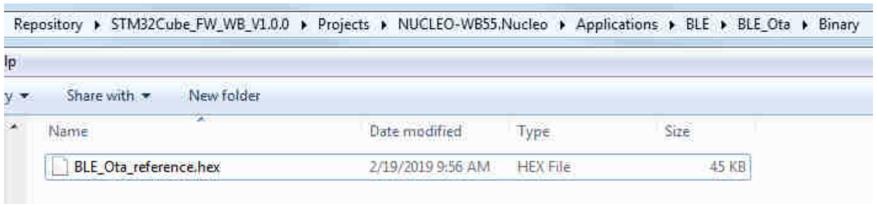
28KB (7 pages)

OTA Loader App

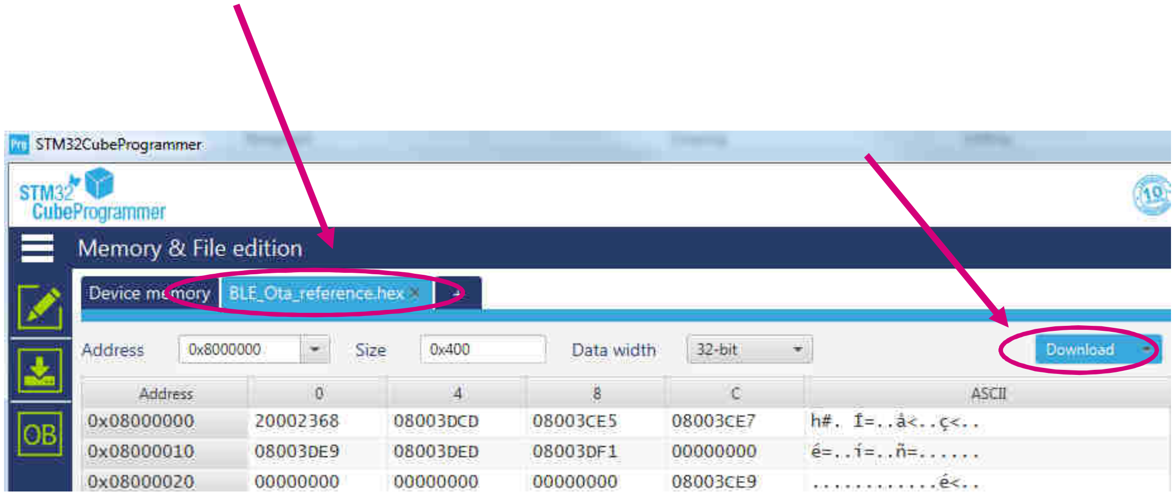
0x08007000

Flash **BLE_Ota_reference.hex** using CubeProgrammer




Repository ▸ STM32Cube_FW_WB_V1.0.0 ▸ Projects ▸ NUCLEO-WB55.Nucleo ▸ Applications ▸ BLE ▸ BLE_Ota ▸ Binary
Share with ▾
New folder

Name	Date modified	Type	Size
BLE_Ota_reference.hex	2/19/2019 9:56 AM	HEX File	45 KB

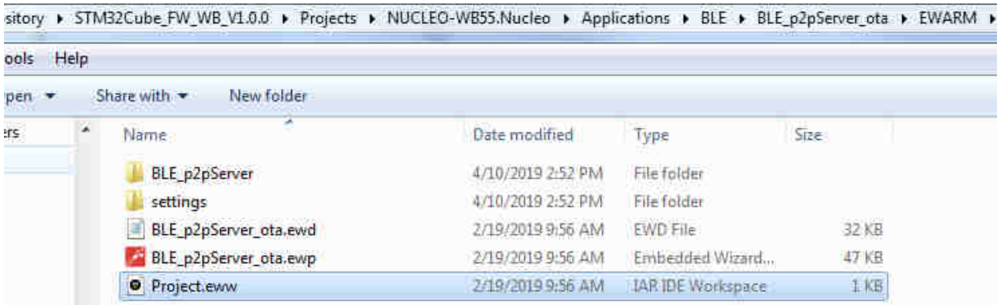


Address 0x8000000 Size 0x400 Data width 32-bit

Address	0	4	8	C	ASCII
0x08000000	20002368	08003DCD	08003CE5	08003CE7	h#. Í=..å<..ç<..
0x08000010	08003DE9	08003DED	08003DF1	00000000	é=..í=..ñ=......
0x08000020	00000000	00000000	00000000	08003CE9	............é<..

P2P Server + OTA project

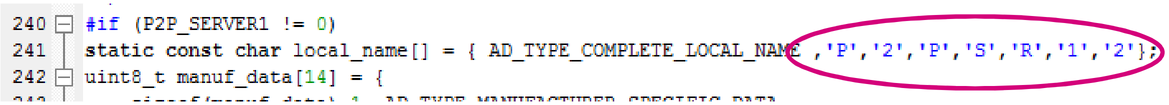
Load and personalize your **BLE_p2pServer_ota.eww** project



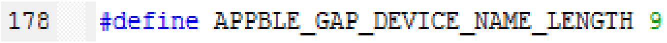
sitory ▸ STM32Cube_FW_WB_V1.0.0 ▸ Projects ▸ NUCLEO-WB55.Nucleo ▸ Applications ▸ BLE ▸ BLE_p2pServer_ota ▸ EWARM ▸

Name	Date modified	Type	Size
BLE_p2pServer	4/10/2019 2:52 PM	File folder	
settings	4/10/2019 2:52 PM	File folder	
BLE_p2pServer_ota.ewd	2/19/2019 9:56 AM	EWD File	32 KB
BLE_p2pServer_ota.ewp	2/19/2019 9:56 AM	Embedded Wizard...	47 KB
Project.eww	2/19/2019 9:56 AM	IAR IDE Workspace	1 KB

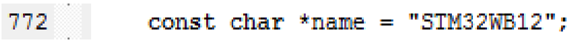
In **app_ble.c**



```
#if (P2P_SERVER1 != 0)
static const char local_name[] = { AD_TYPE_COMPLETE_LOCAL_NAME ,'P','2','P','S','R','1','2'};
uint8_t manuf_data[14] = {
    sizeof(manuf_data)-1, AD_TYPE_MANUFACTURER_SPECIFIC_DATA,
```

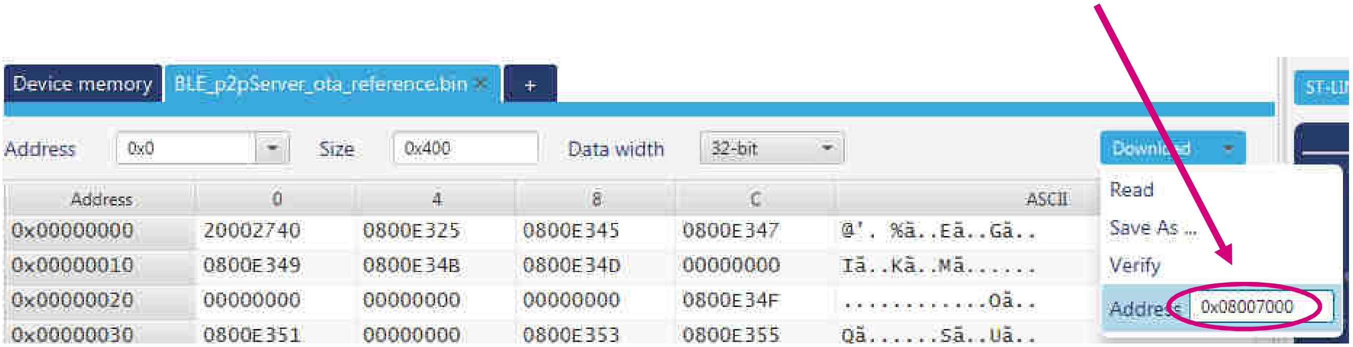


```
#define APPBLE_GAP_DEVICE_NAME_LENGTH 9
```



```
    const char *name = "STM32WB12";
```

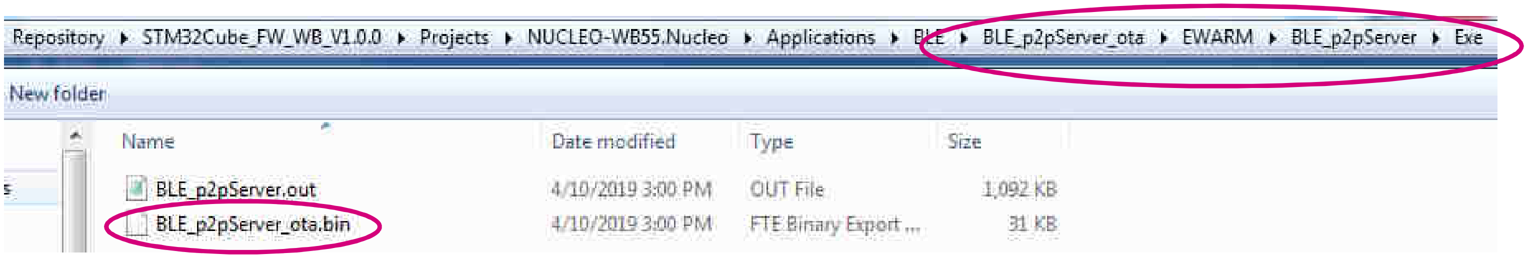
Flash your newly created **BLE_p2pServer_ota.bin** to 0x08007000



Address 0x0 Size 0x400 Data width 32-bit

Address	0	4	8	C	ASCII
0x00000000	20002740	0800E325	0800E345	0800E347	@'. %ã..Eã..Gã..
0x00000010	0800E349	0800E34B	0800E34D	00000000	Iã..Kã..Mã......
0x00000020	00000000	00000000	00000000	0800E34F	............Oã..
0x00000030	0800E351	00000000	0800E353	0800E355	Qã......Sã..Uã..

Address 0x08007000



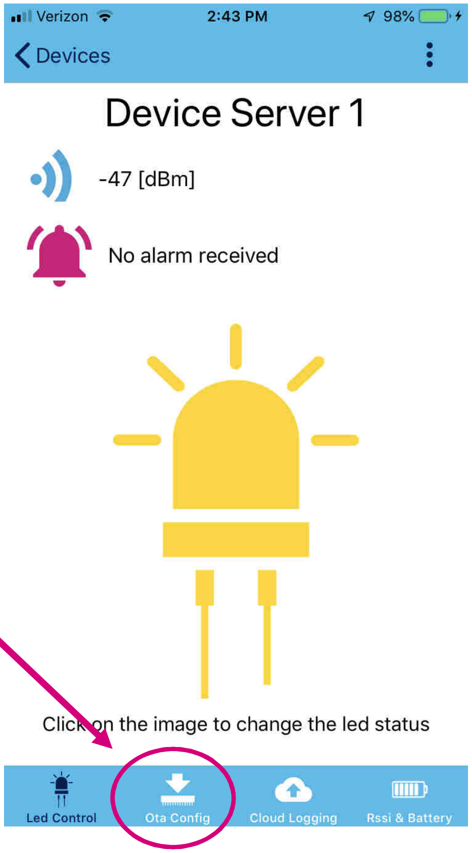
Repository ▸ STM32Cube_FW_WB_V1.0.0 ▸ Projects ▸ NUCLEO-WB55.Nucleo ▸ Applications ▸ BLE ▸ BLE_p2pServer_ota ▸ EWARM ▸ BLE_p2pServer ▸ Exe

Name	Date modified	Type	Size
BLE_p2pServer.out	4/10/2019 3:00 PM	OUT File	1,092 KB
BLE_p2pServer_ota.bin	4/10/2019 3:00 PM	FTE Binary Export ...	31 KB

Server + OTA App

Verify functionality on the ST BLE Sensor app

You should also see OTA capability



Verizon 2:43 PM 98%

Devices

Device Server 1

-47 [dBm]

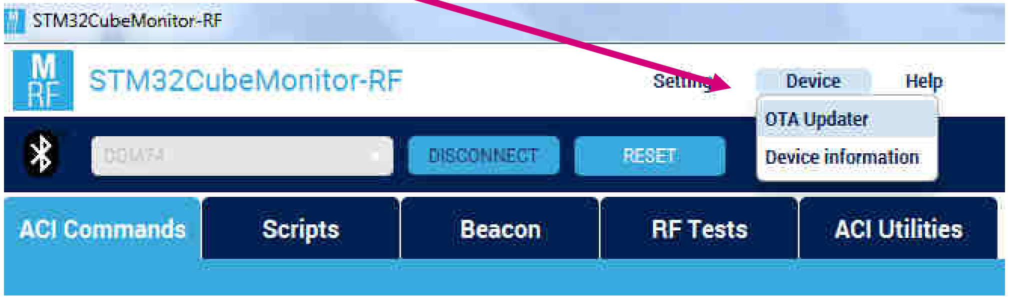
No alarm received

Click on the image to change the led status

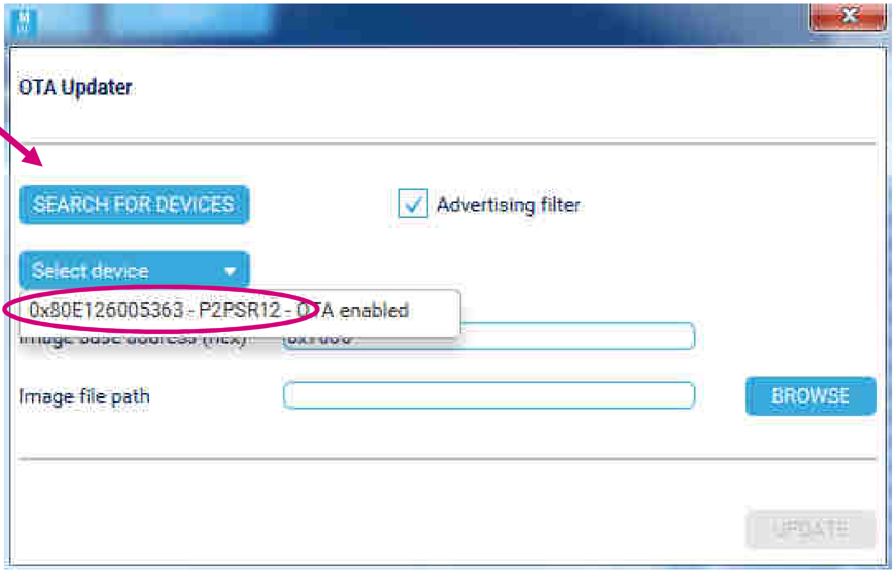
Once seen, disconnect from your device

Connect to OTA-enabled device

Connect the Dongle and select **OTA Updater**



Search and Select your Device
(you can see your local name & BLE Address)



☑ Advertising filter

0x80E126005363 - P2PSR12 - OTA enabled

Update BLE Application

Browse for the other OTA binary

- BLE_HeartRate_ota_reference.bin

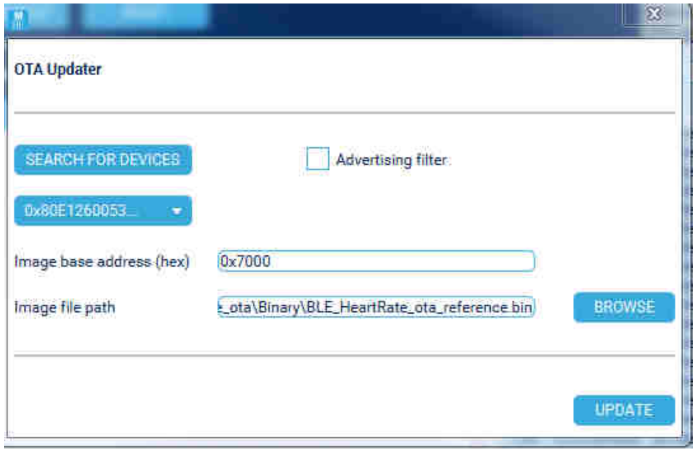
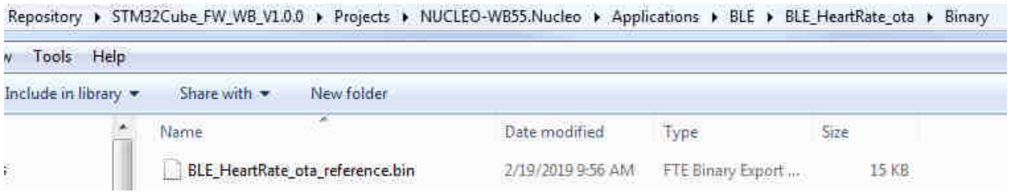

OTA Updater

Image base address (hex): 0x7000

Image file path: _ota\Binary\BLE_HeartRate_ota_reference.bin


Repository ▸ STM32Cube_FW_WB_V1.0.0 ▸ Projects ▸ NUCLEO-WB55.Nucleo ▸ Applications ▸ BLE ▸ BLE_HeartRate_ota ▸ Binary

Name	Date modified	Type	Size
BLE_HeartRate_ota_reference.bin	2/19/2019 9:56 AM	FTE Binary Export ...	15 KB

Update image

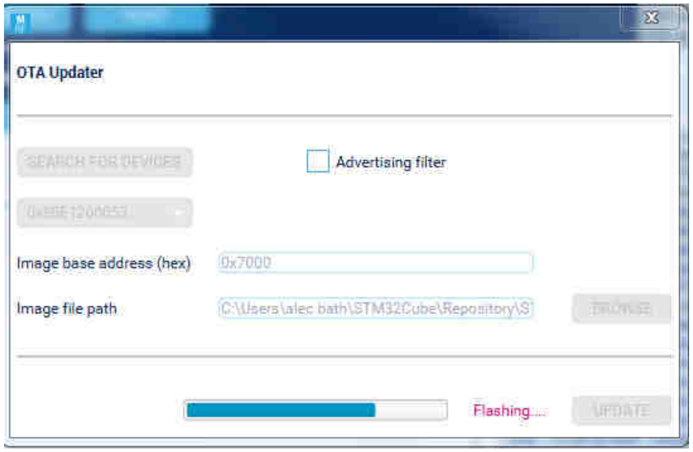

OTA Updater

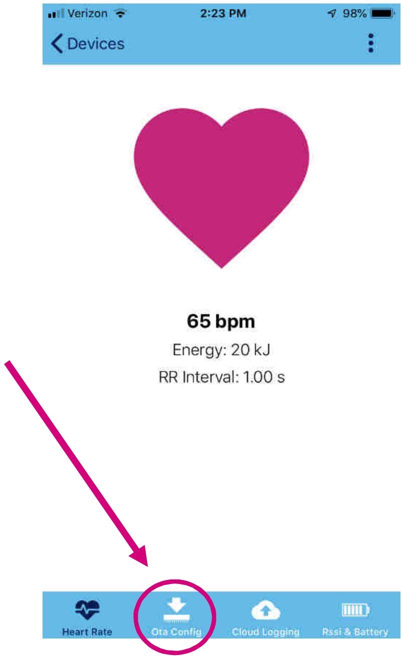
Image base address (hex): 0x7000

Image file path: C:\Users\alec bath\STM32Cube\Repository\S

Flashing....

Update via ST BLE Sensor app

OTA capability detected
Click to start

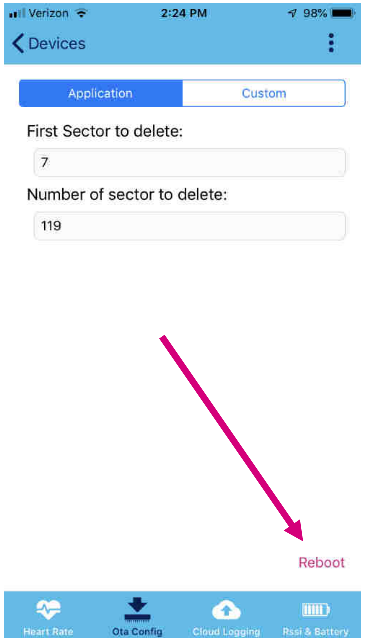


65 bpm

Energy: 20 kJ

RR Interval: 1.00 s

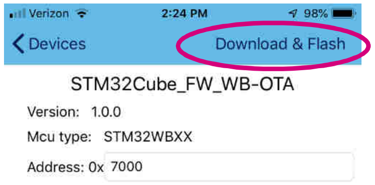
Erase & Reboot



First Sector to delete: 7

Number of sector to delete: 119

Select Smartphone file (iCloud, etc)

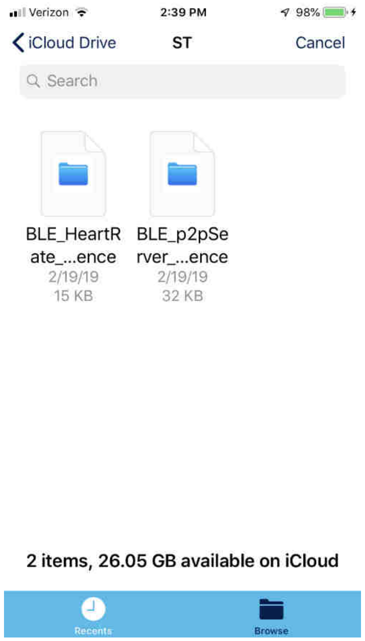


STM32Cube_FW_WB-OTA

Version: 1.0.0

Mcu type: STM32WBXX

Address: 0x 7000

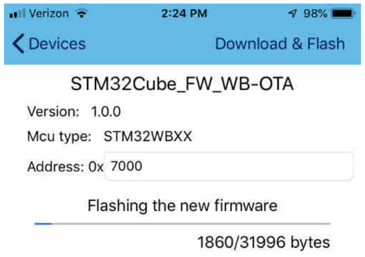


BLE_HeartRate_...ence 2/19/19 15 KB

BLE_p2pServer_...ence 2/19/19 32 KB

2 items, 26.05 GB available on iCloud

Flashing begins



STM32Cube_FW_WB-OTA

Version: 1.0.0

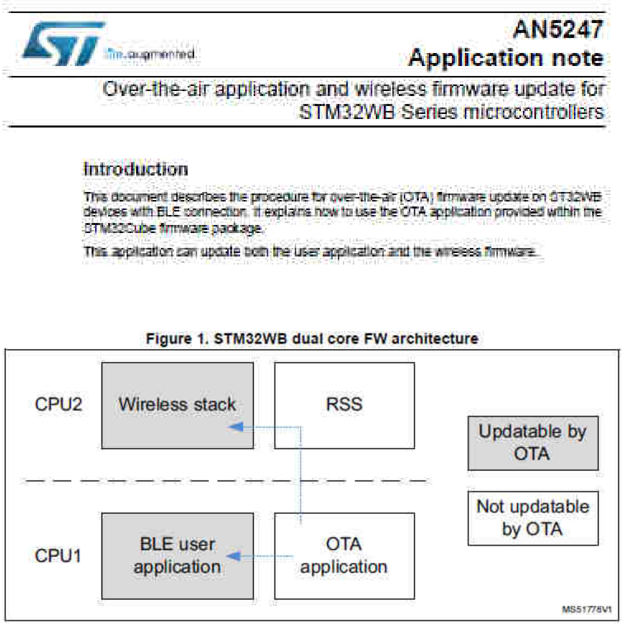
Mcu type: STM32WBXX

Address: 0x 7000

Flashing the new firmware

1860/31996 bytes

AN5247 details the OTA application in further detail.



AN5247
Application note

Over-the-air application and wireless firmware update for STM32WB Series microcontrollers

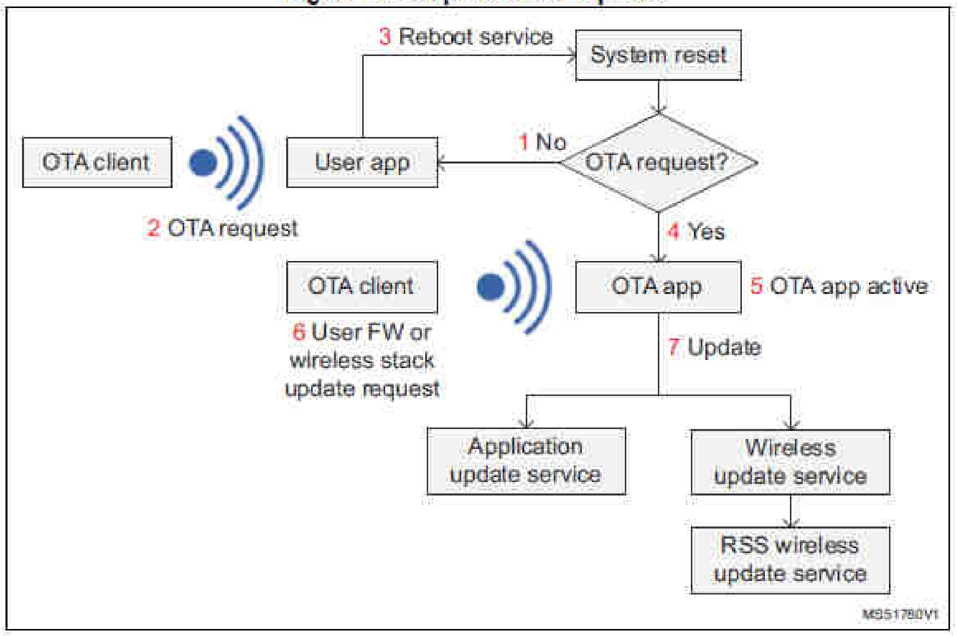
Introduction

This document describes the procedure for over-the-air (OTA) firmware update on STM32WB devices with BLE connection. It explains how to use the OTA application provided within the STM32Cube firmware package.

This application can update both the user application and the wireless firmware.

Figure 1. STM32WB dual core FW architecture

Figure 3. OTA procedure sequence



Releasing Your Creativity

/STM32 @ST_World ST community

life.augmented

www.st.com/stm32wb