Getting Started with the STM32L4 IoT Discovery Kit Node

David Kwak

Slim Jallouli

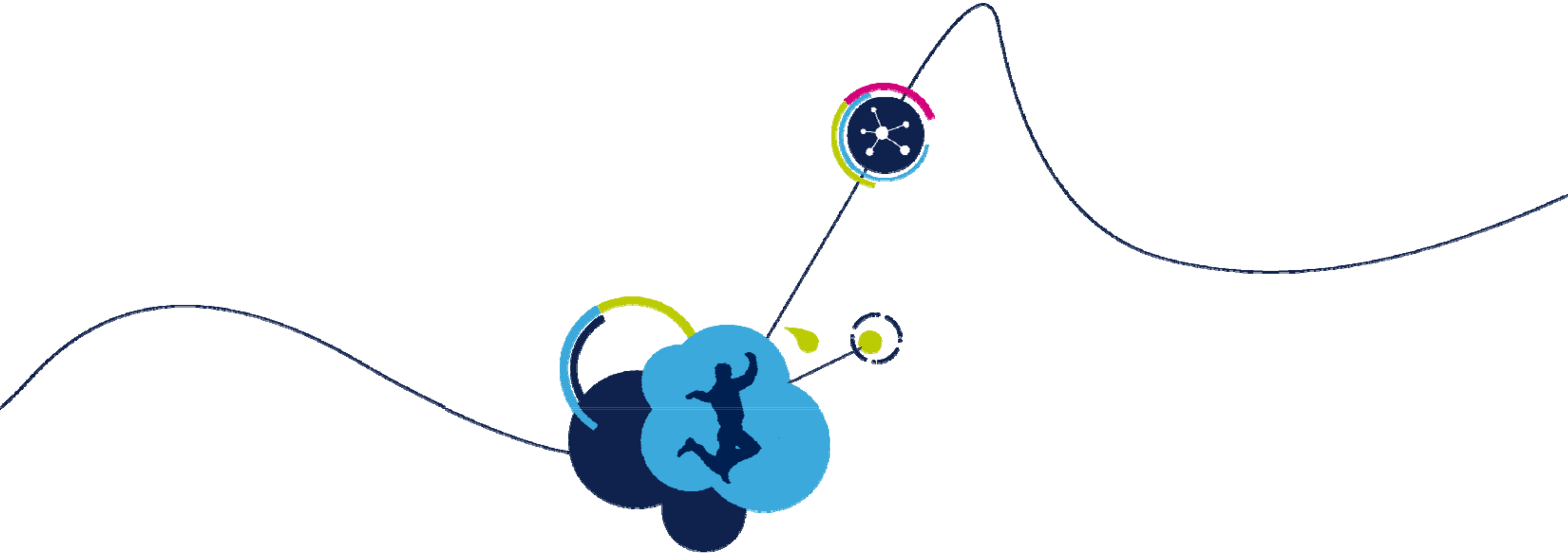
life.augmented

Technology
Tour 2017

Agenda

Presentation

- Training Material Installation
- Overview of the STM32 Portfolio
- Overview of the STM32L475
- Overview of the STM32L4 IoT Discovery Kit Node
- STM32Cube Introduction
- IAR License Installation
- ST-Link Driver Installation
- Lab 1 : Getting Started with STM32CubeMX - Blinky LED
- Bluetooth® Low Energy Overview
- Lab 2 : Bluetooth Low Energy pairing
- Wi-Fi Module Overview
- Amazon AWS IoT Overview
- Lab 3 : Creating your device (“Thing”) on AWS
- Lab 4 : Connect to AWS IoT & Send Sensor Data
- Lab 5 : Connect to a Different MQTT Topic
- Alexa Voice Demo



Tools installation

Training Materials Installation

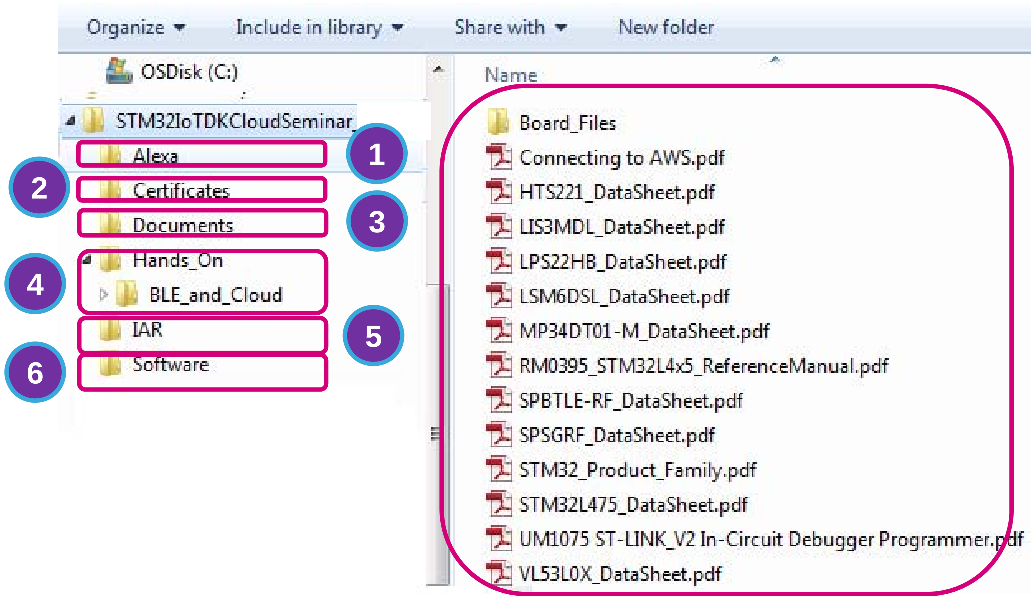
- Each participant should have received a USB Flash drive. It contains the Seminar Installer. This will install Tera Term, the latest Java, STM32CubeMX, STM32CubeL4 HAL, and extract the seminar file to `C:\STM32IoTDKCloudSeminar`.

- Please insert the USB Drive to your machine. Copy all the files to your desktop and execute the installer (<u>Run as Admin</u>) (`STM32_IoT_DK_Cloud_Training_Installer-1.07.exe`).

- At the end of the seminar material installation, we will continue with ST-Link Utility and IAR installation.

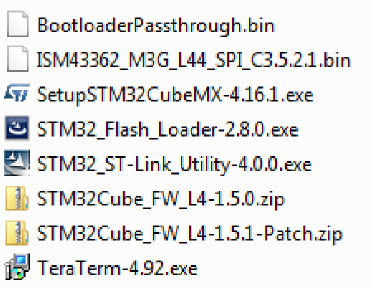
Seminar Directory Content

1. Alexa Skill Code
2. Thing Certificates
3. Documents
4. Hands on
5. IAR
6. Software



STM32IoTDKCloudSeminar
- Alexa
- Certificates
- Documents
- Hands_On
  - BLE_and_Cloud
- IAR
- Software

Board_Files
Connecting to AWS.pdf
HTS221_DataSheet.pdf
LIS3MDL_DataSheet.pdf
LPS22HB_DataSheet.pdf
LSM6DSL_DataSheet.pdf
MP34DT01-M_DataSheet.pdf
RM0395_STM32L4x5_ReferenceManual.pdf
SPBTLE-RF_DataSheet.pdf
SPSGRF_DataSheet.pdf
STM32_Product_Family.pdf
STM32L475_DataSheet.pdf
UM1075 ST-LINK_V2 In-Circuit Debugger Programmer.pdf
VL53L0X_DataSheet.pdf

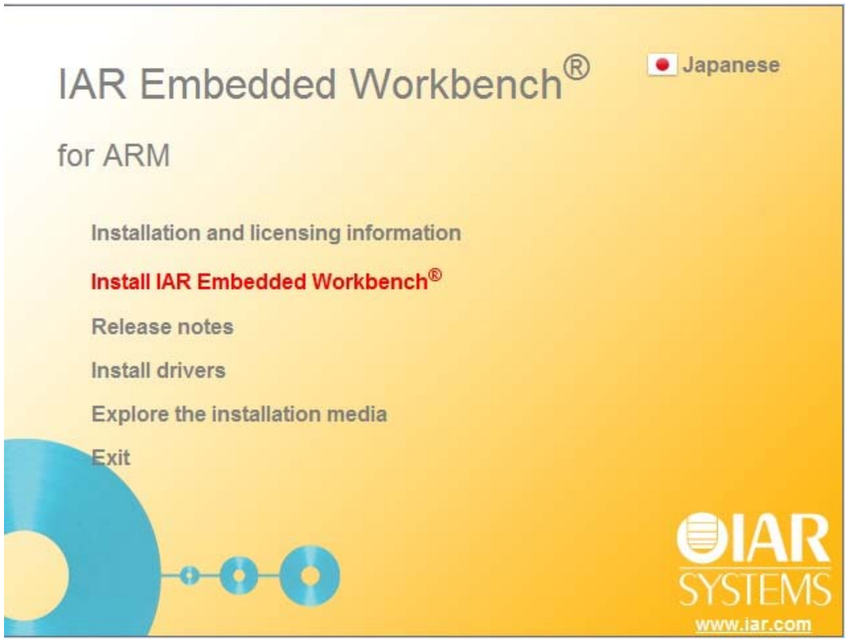


BootloaderPassthrough.bin
ISM43362_M3G_L44_SPI_C3.5.2.1.bin
SetupSTM32CubeMX-4.16.1.exe
STM32_Flash_Loader-2.8.0.exe
STM32_ST-Link_Utility-4.0.0.exe
STM32Cube_FW_L4-1.5.0.zip
STM32Cube_FW_L4-1.5.1-Patch.zip
TeraTerm-4.92.exe

IAR installation

- Run the IAR professional tool suite installer: `C:\STM32IoTDKCloudSeminar\IAR.`

- From the installer menu select Install IAR Embedded Workbench.



IAR Embedded Workbench®

for ARM

Installation and licensing information

Install IAR Embedded Workbench®

Release notes

Install drivers

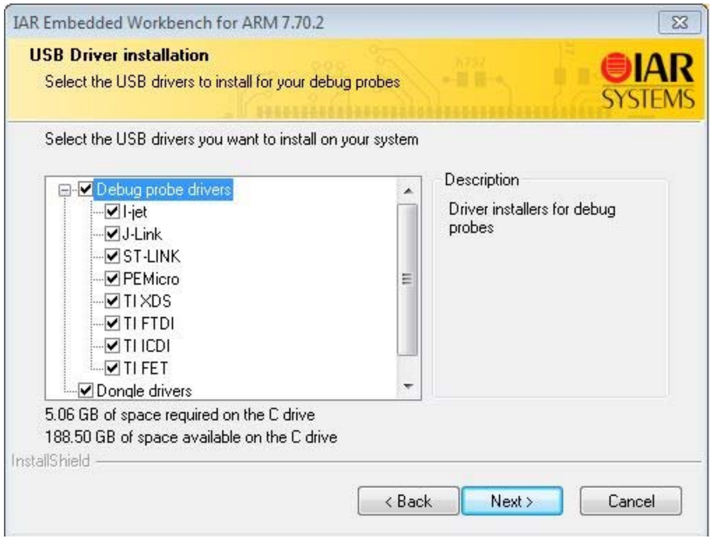
Explore the installation media

Exit

IAR USB Driver Installation

- De-select **all** the USB drivers when IAR prompts you to install the USB drivers. This will speed-up IAR installation.

- ST-Link driver will be installed later.



IAR Embedded Workbench for ARM 7.70.2

USB Driver installation

Select the USB drivers to install for your debug probes

Select the USB drivers you want to install on your system

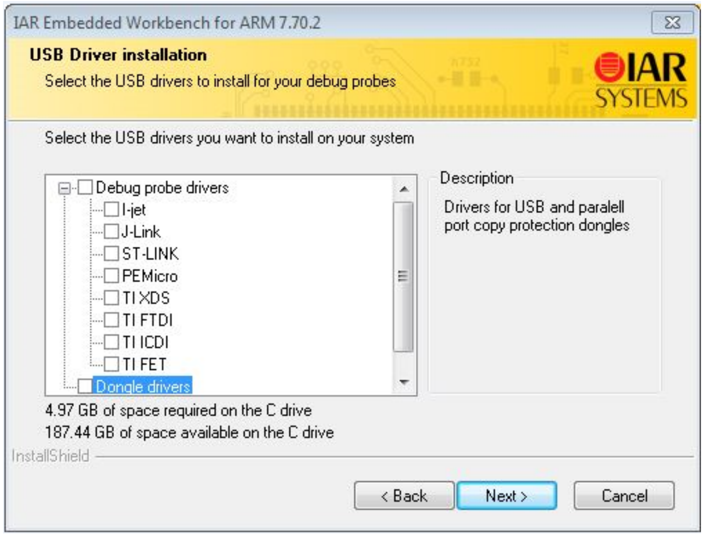
- ☑ Debug probe drivers
  - ☑ I-jet
  - ☑ J-Link
  - ☑ ST-LINK
  - ☑ PEMicro
  - ☑ TI XDS
  - ☑ TI FTDI
  - ☑ TI ICDI
  - ☑ TI FET
- ☑ Dongle drivers

Description

Driver installers for debug probes

5.06 GB of space required on the C drive

188.50 GB of space available on the C drive



IAR Embedded Workbench for ARM 7.70.2

USB Driver installation

Select the USB drivers to install for your debug probes

Select the USB drivers you want to install on your system

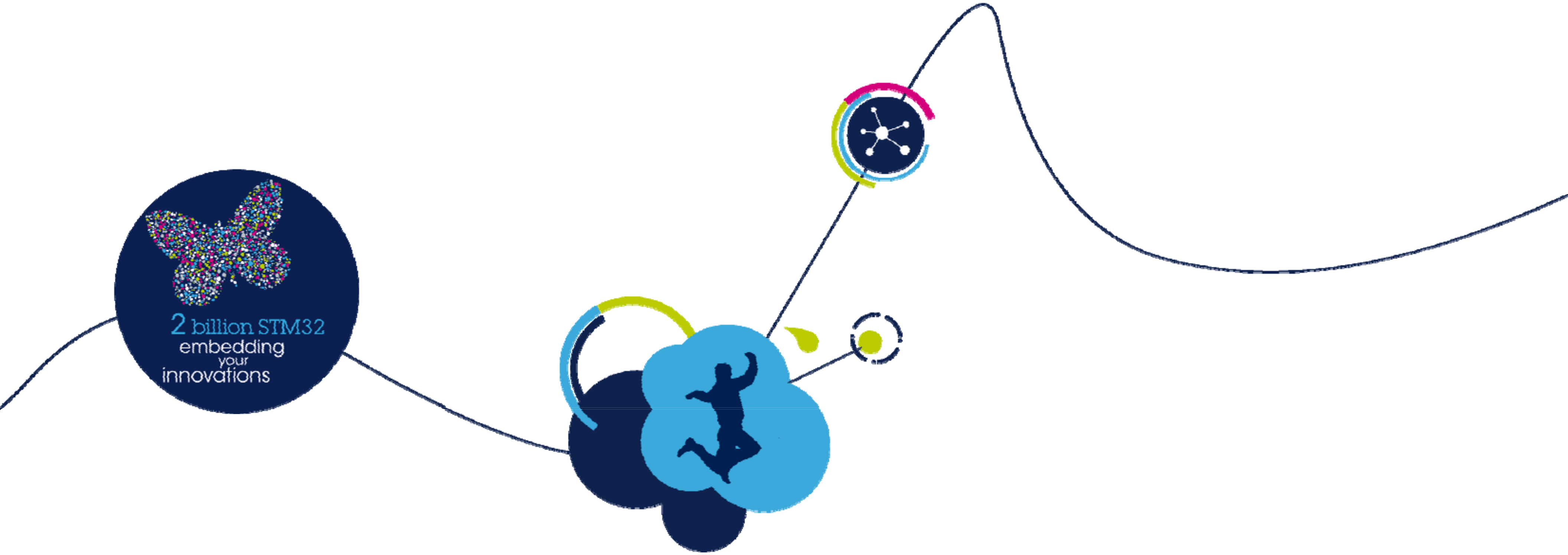
- □ Debug probe drivers
  - □ I-jet
  - □ J-Link
  - □ ST-LINK
  - □ PEMicro
  - □ TI XDS
  - □ TI FTDI
  - □ TI ICDI
  - □ TI FET
- □ Dongle drivers

Description

Drivers for USB and paralell port copy protection dongles

4.97 GB of space required on the C drive

187.44 GB of space available on the C drive



Overview of the STM32 Portfolio

life.augmented

Technology Tour 2017

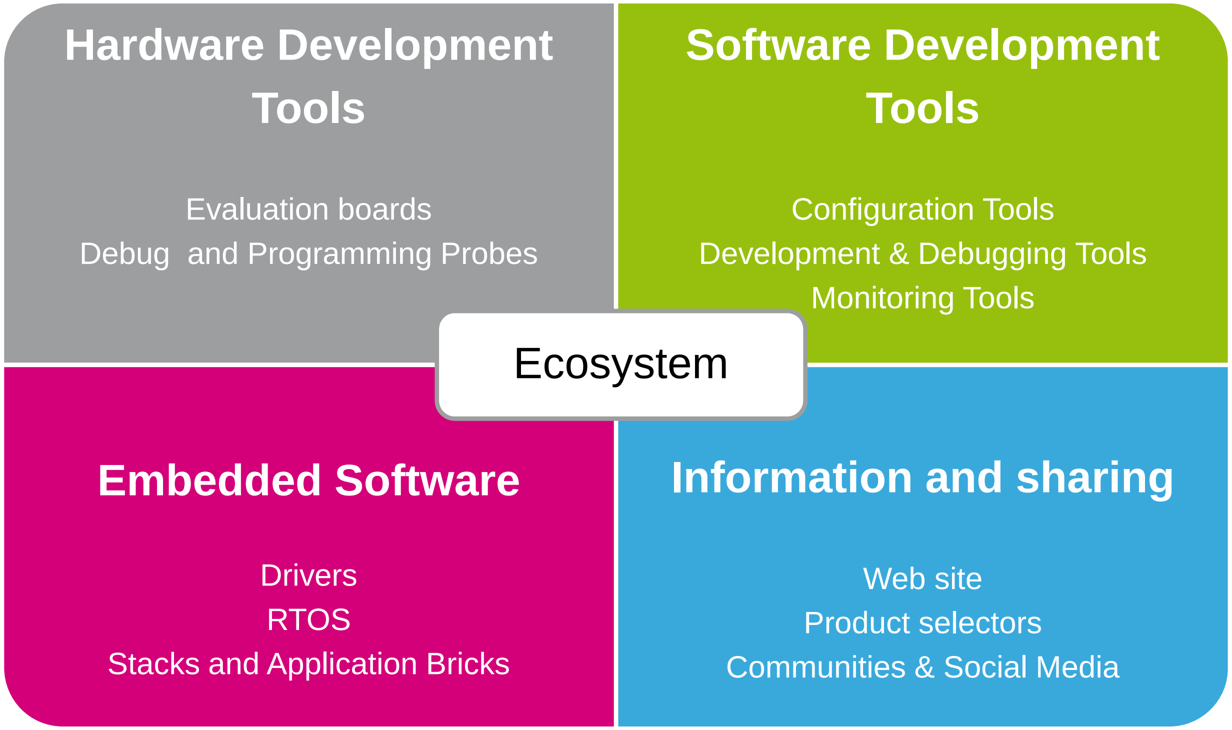
Today - STM32 Portfolio

10 product series / More than 40 product lines

High-performance		STM32 F2 398 CoreMark 120 MHz 150 DMIPS	STM32 F4 608 CoreMark 180 MHz 225 DMIPS	STM32 H7 2 010 CoreMark 400 MHz 856 DMIPS STM32 F7 1 082 CoreMark 216 MHz 462 DMIPS
Mainstream	STM32 F0 106 CoreMark 48 MHz 38 DMIPS	STM32 F1 177 CoreMark 72 MHz 61 DMIPS	STM32 F3 245 CoreMark* 72 MHz 90 DMIPS (*) from CCM-SRAM	LONGEVITY 10 YEARS COMMITMENT
Ultra-low-power	STM32 L0 75 CoreMark 32 MHz 26 DMIPS	STM32 L1 93 CoreMark 32 MHz 33 DMIPS	STM32 L4 273 CoreMark 80 MHz 100 DMIPS	
ARM CORTEX Processor Technology	Cortex-M0 Cortex-M0+	Cortex-M3	Cortex-M4	Cortex-M7

What is MCU Ecosystem?

All collaterals required to develop with an MCU



Hardware Development Tools

Evaluation boards
Debug and Programming Probes

Software Development Tools

Configuration Tools
Development & Debugging Tools
Monitoring Tools

Ecosystem

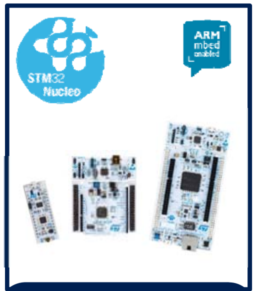
Embedded Software

Drivers
RTOS
Stacks and Application Bricks

Information and sharing

Web site
Product selectors
Communities & Social Media

Hardware Development Tools

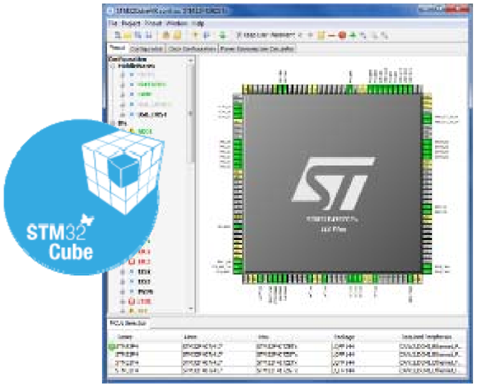


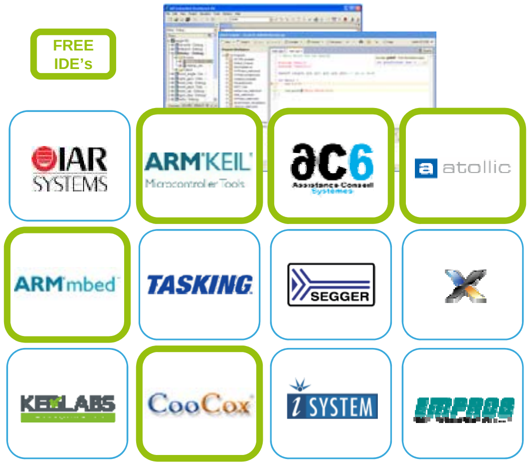
	STM32 Nucleo	Discovery kits	Evaluation boards	3rd parties
Typical use case	Flexible prototyping, Community	Prototyping, Creative demos	Full feature evaluation	From full evaluation to open hardware
Extension possibilities	+ + +	+ +	+	
Connectivity	Arduino™ ST Morpho	ST	ST	

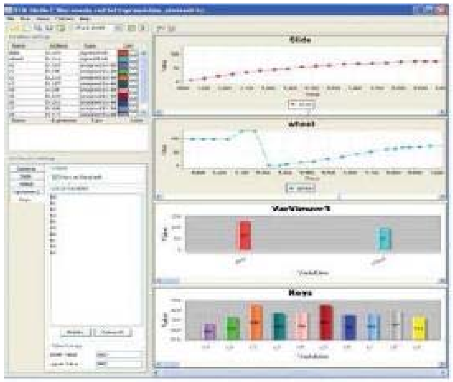
STM32 ecosystem SW development tools

C/C++ Focus

A complete flow, from configuration up to monitoring







STM32CubeMX
Configure & Generate Code

Partners IDEs
Compile and Debug

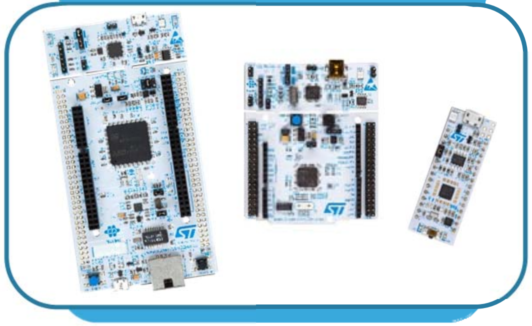
STMStudio
Monitor

STM32 ODE platform

Hardware

STM32 Nucleo

STM32 Nucleo development boards



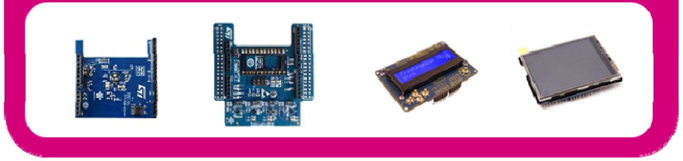
Software

STM32 Cube

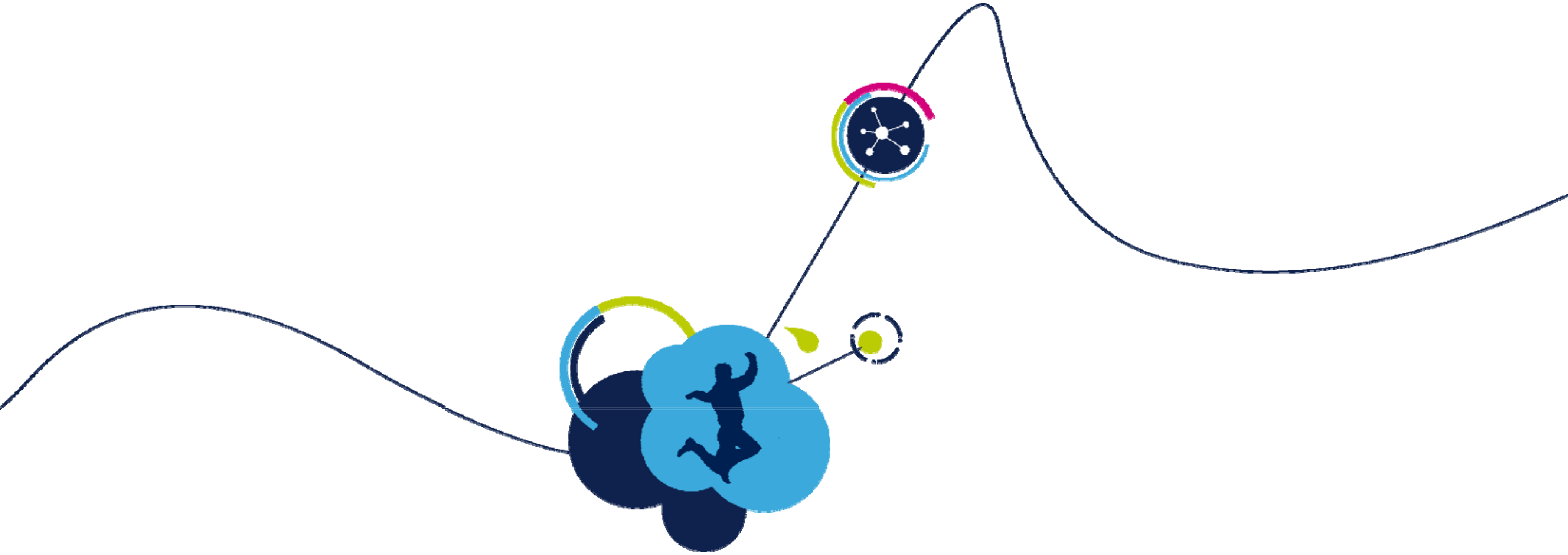
STM32Cube software library

Expansion

STM32 Nucleo expansion boards from ST and third parties



STM32Cube expansion SW

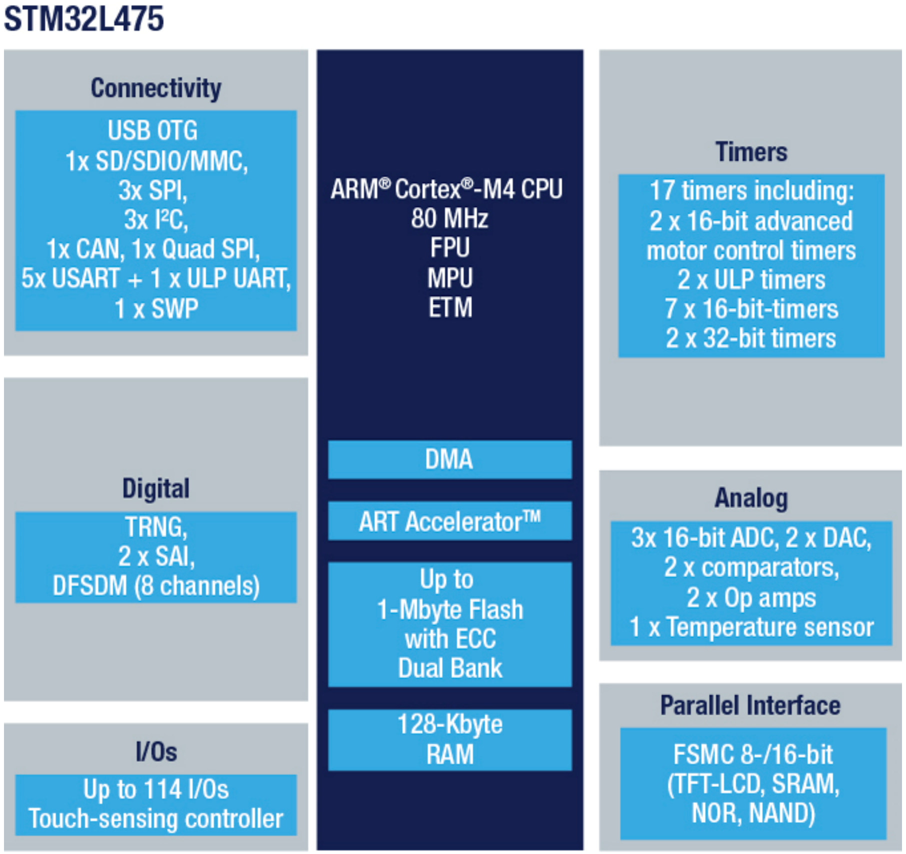


Overview of STM32L475

STM32L475 Block Diagram

Key features

- Cortex M4 with DSP, FPU @ 80MHz and ART
- 1.71V – 3.6V supply 80 MHz Full functional
- 1MB Flash dual bank/ 128KB RAM
- USB OTG FS –LPM Battery Charging Detection
- 3 x Ultra-low-power 12-bit ADC 5 MSPS
- Touch-Sensing 24 channels
- Ultra-low power
    - VBAT
    - Better Wake Up time vs. STM32L1
    - Down to 160µA/MHz dynamic
- I²C FM+
- SPI: variable data length
- USART
- LP UART & 16-bit Timer
- FSMC, Quad SPI
- CAN, SWPMI, SDMMC, 2x SAI
- Digital filter for Sigma delta modulator
- 17 x timers
- Analog: Op-Amps, comparators, DAC, VREF, temperature sensor
- RNG



STM32L475

Connectivity

USB OTG
1x SD/SDIO/MMC,
3x SPI,
3x I²C,
1x CAN, 1x Quad SPI,
5x USART + 1 x ULP UART,
1 x SWP

Digital

TRNG,
2 x SAI,
DFSDM (8 channels)

I/Os

Up to 114 I/Os
Touch-sensing controller

ARM® Cortex®-M4 CPU
80 MHz
FPU
MPU
ETM

DMA

ART Accelerator™

Up to
1-Mbyte Flash
with ECC
Dual Bank

128-Kbyte
RAM

Timers

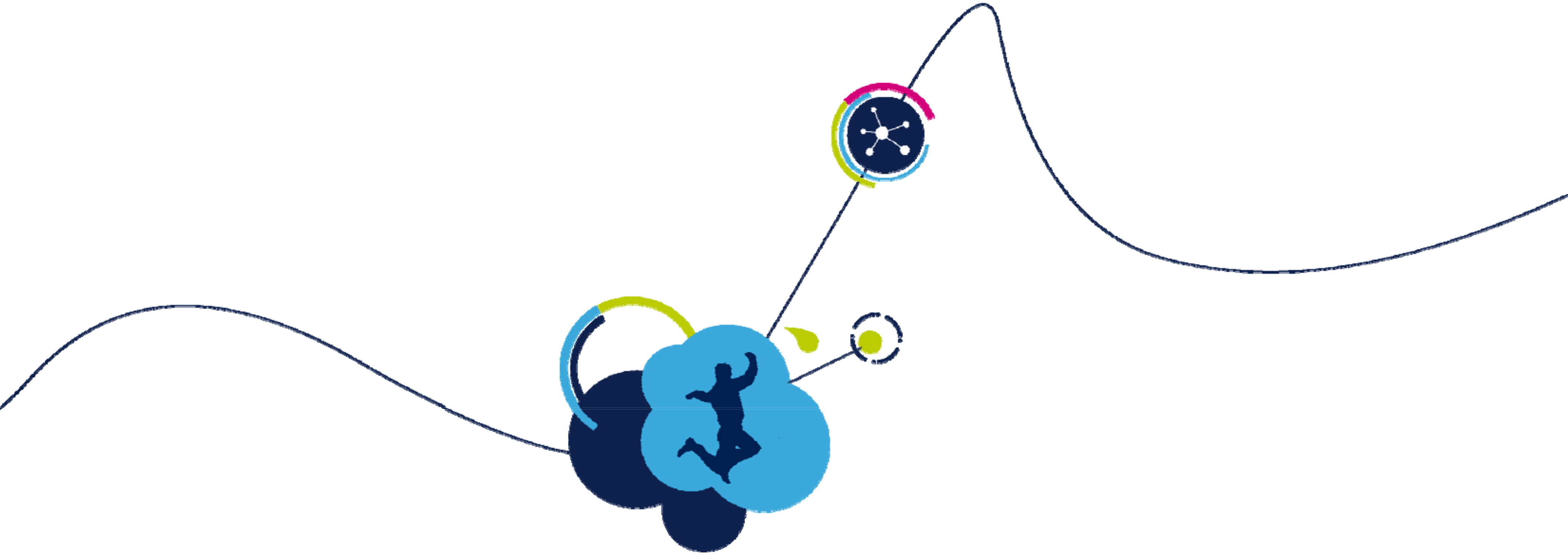
17 timers including:
2 x 16-bit advanced
motor control timers
2 x ULP timers
7 x 16-bit-timers
2 x 32-bit timers

Analog

3x 16-bit ADC, 2 x DAC,
2 x comparators,
2 x Op amps
1 x Temperature sensor

Parallel Interface

FSMC 8-/16-bit
(TFT-LCD, SRAM,
NOR, NAND)

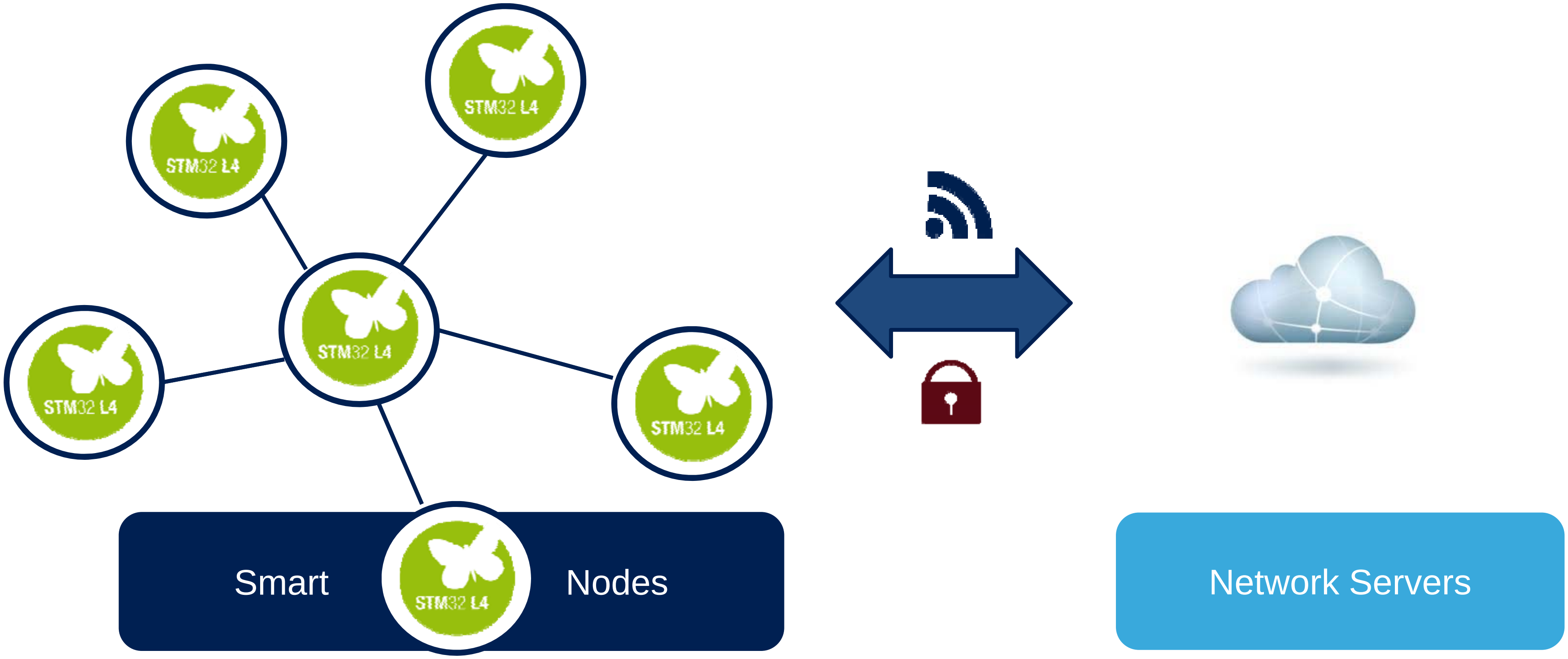


Overview of the STM32L4 IoT Discovery Kit Node

Technology Tour 2017

STM32L475 Discovery IoT Node

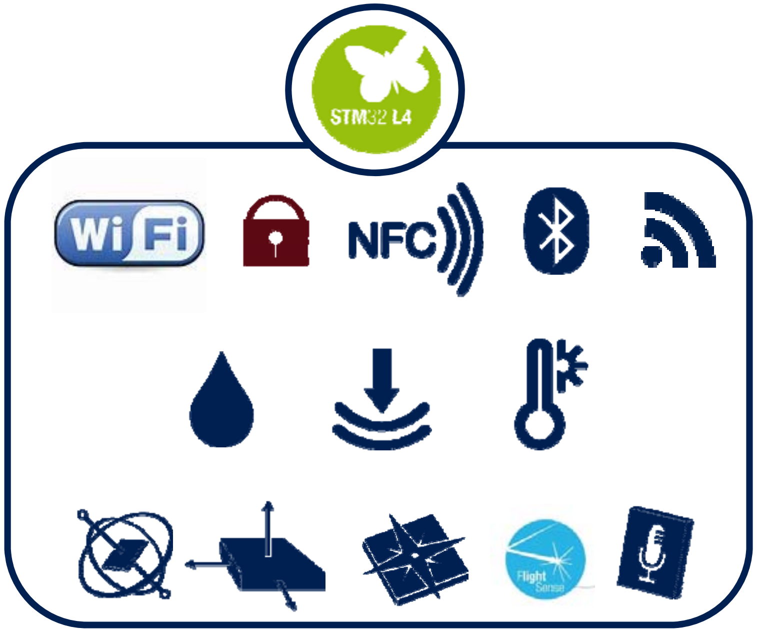
Get connected seamlessly!



STM32L4 smart node

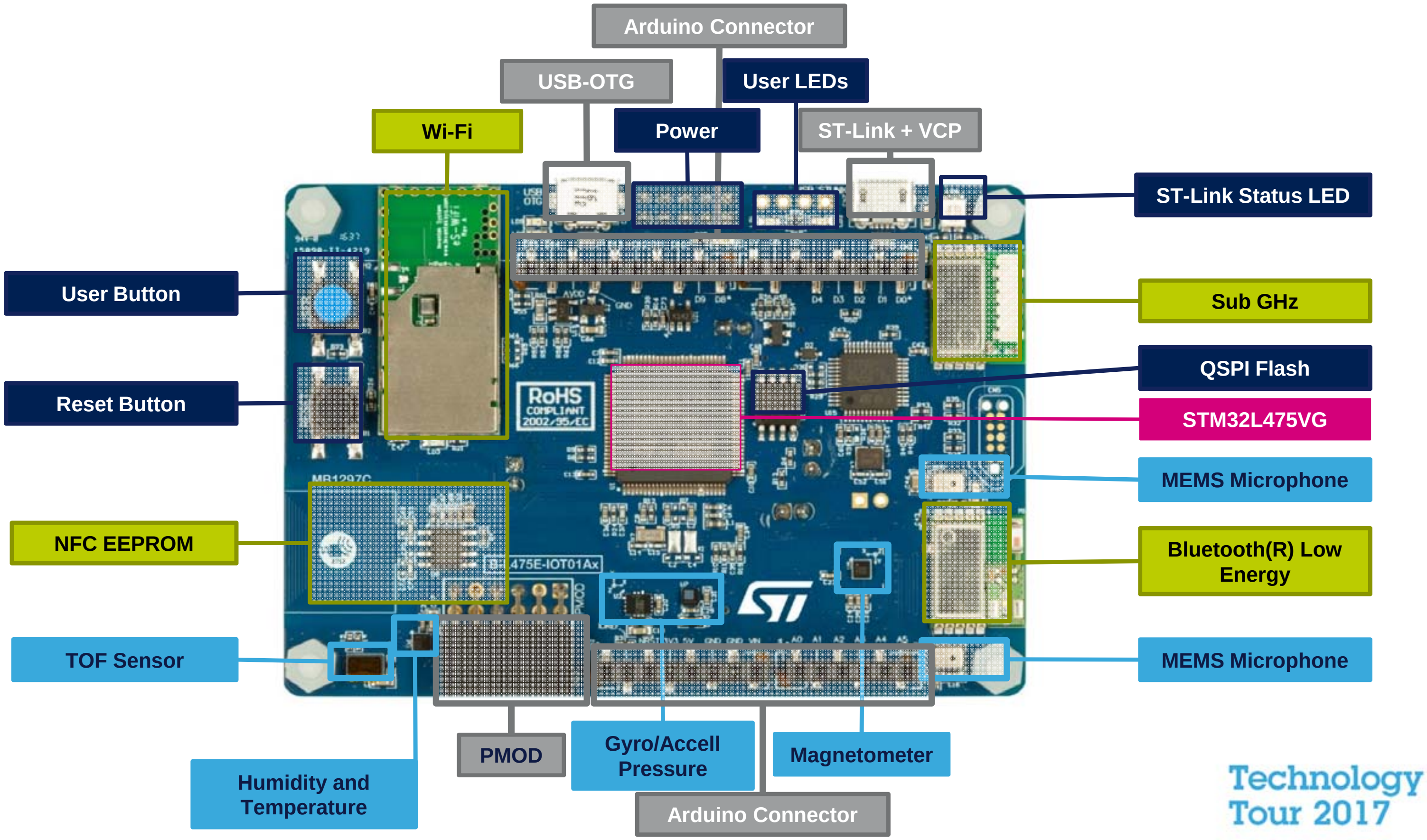
Open the door to remote services

- Direct connection to cloud servers
- Low-power long-range communication
- Environmental awareness: humidity, pressure, temp
- Detection hub: motion, proximity, audio



STM32L475 Discovery IoT Node

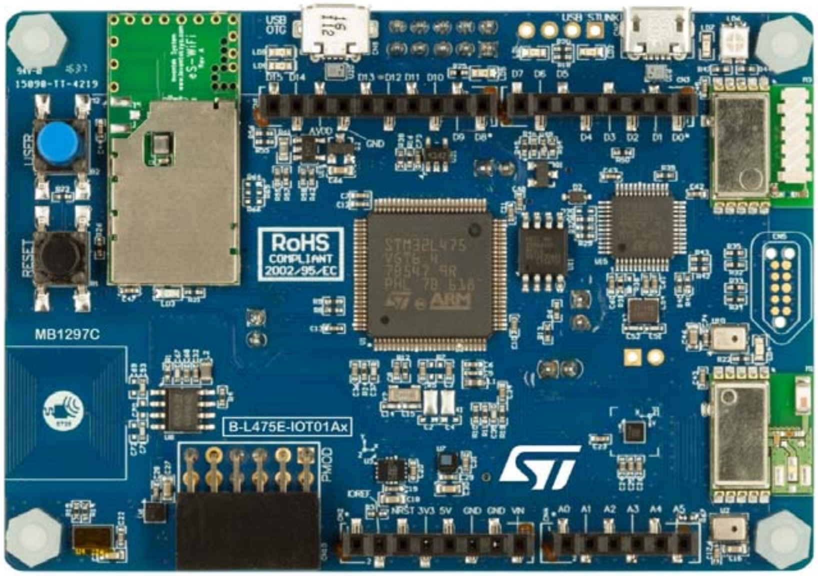
Multi-link communication, multiway sensing



Comprehensive software libraries

Instant showcase

- SW Libraries for STM32L4 mcu & sensors
- Connectivity SW protocol stacks
- Cloud service connectors (AWS)
- Demo examples (X-CUBE-AWS)

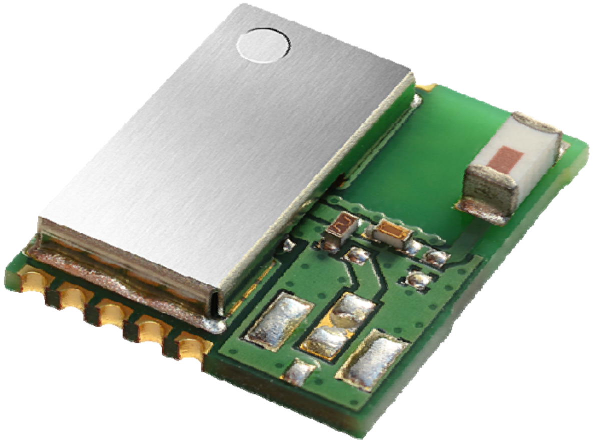


Wireless Connectivity – Wi-Fi

- Inventek ISM43362 Wi-Fi Module
    - 802.11 b/g/n compliant module based on a Broadcom MAC/Baseband/Radio device
    - Fully contained TCP/IP stack minimizing host CPU requirements
    - FCC and CE certified
    - Secure Wi-Fi authentication supporting WEP-128, WPA-PSK (TKIP), WPA2-PSK

Wireless Connectivity - Bluetooth

- ST SPBTLE-RF Bluetooth Low Energy Module
  - Based on our ST BlueNRG-MS Wireless Network Processor
  - Bluetooth Low Energy 4.1 compliant
  - FCC and BQ certified module with integrated balun & antenna



Wireless Connectivity - SubGHz

- ST SPSGRF-915 Sub-GHz Module (915 MHz - US)
  - FCC and IC certified module with integrated balun & antenna
  - Supports 2-FSK, GFSK, MSK, GMSK, OOK and ASK modulation schemes
  - Long range (100s of meters) with an air data rate from 1 to 500 kbps

Wireless Connectivity - NFC

- ST M24SR64-Y Dynamic NFC/RFID Tag
  - NFC Forum Type 4 Tag
  - ISO/IEC 14443 Type A
  - 106 Kbps Data Rate

Wired Connectivity Features

- ST-Link V2
  - Programming and Debug Interface
- USB OTG FS
  - Full Speed USB On-The-Go Communication Interface
- PMOD
  - Peripheral Module Interface Supporting GSM, GPS, etc…
- Arduino Connectors
  - Arduino Compatible Connectors to Interface with Additional ST X-NUCLEO or 3rd Party Expansion Board (eg: LoRa)

ST Sensors

- Full Range of Motion & Environmental MEMS Sensors
  - LSM6DSL Accelerometer + Gyroscope Sensor
  - LIS3MDL Magnetometer Sensor
  - HTS221 Humidity + Temperature Sensor
  - LPS22HB Pressure Sensor
- Integrated High Accuracy Proximity/Range Sensor
  - VL53L0X Time-of-Flight Range Sensor
- Digital Microphones
  - MP34DT01 MEMS Digital Microphones
    - Voice & Audio Recognition Functions
    - Acoustic Beam Forming with OSX_AcousticBF_Library

User Resource Features

- Reset and User Buttons
  - Board Reset and Programmable Application Buttons
- User LEDs
  - Programmable Application LEDs
- QSPI Flash
  - 64Mbit for Data Storage and Program Execution
- Selectable Power Supply
  - ST-Link, USB-OTG, Arduino or External Power

Summary

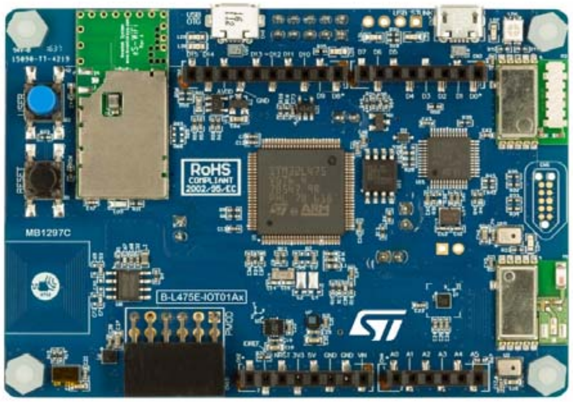
Advantages of Single Board

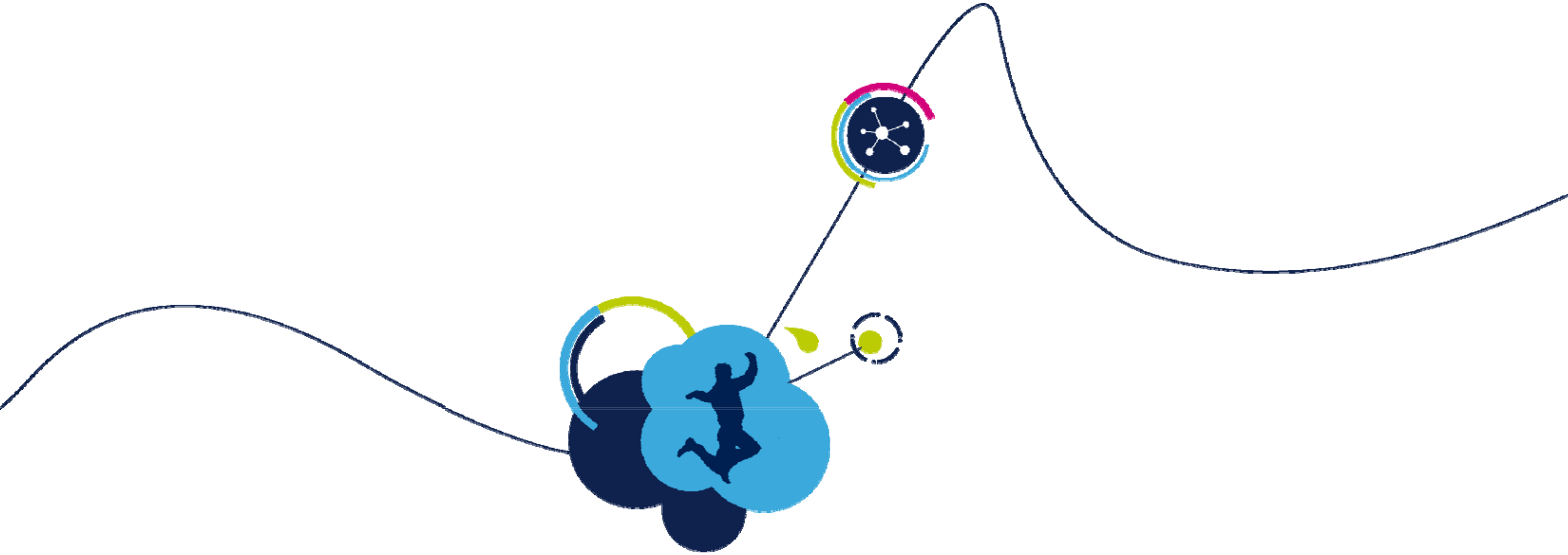
- Easily Debug Hardware Issues on a Single Board.
- Collateral Includes Tightly Coupled Firmware
  - BSP Included for All Board Components
  - Cloud Connectivity Reference Solutions Included
- Represents a Cost Effective Development Solution (~$60)
- No Need to Manage & Order Multiple Board SKUs.

Availability

Early access during the ST Tech Tour

Part number	Samples	Mass Market Availability	SubGHz frequency band	Regions with authorized use
B-L475E-IOT01A1	NOW	June 2017	915 MHz	US
B-L475E-IOT01A2	NOW	June 2017	868 MHz	Rest of the World





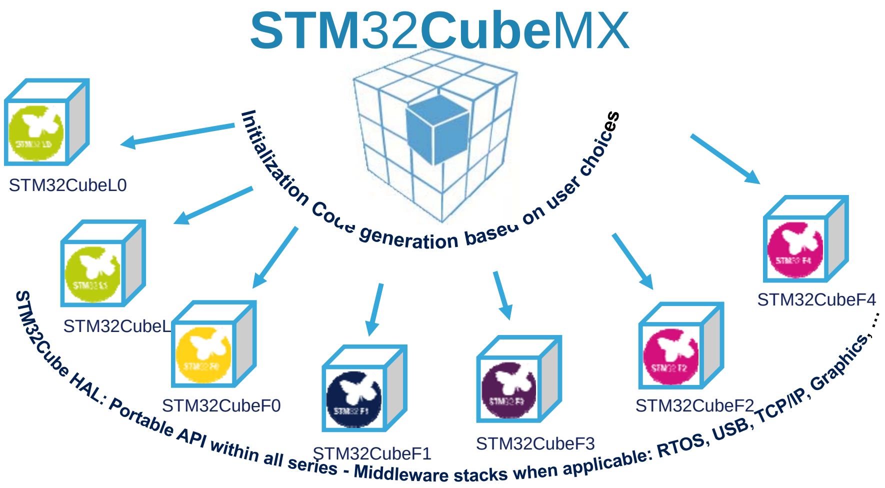
STM32Cube^{TM} Introduction

Technology Tour 2017

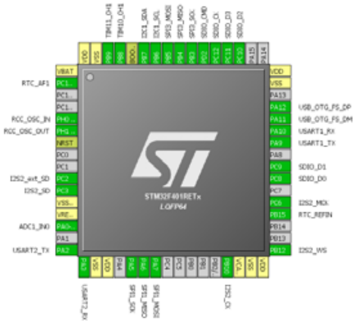
STM32Cube^{TM}

Introduction

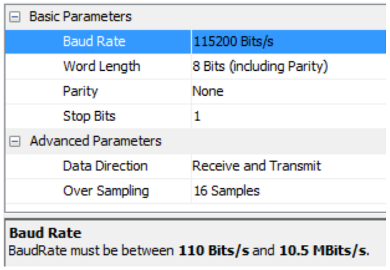
- STM32Cube^{TM} includes:
  - A configuration tool, STM32CubeMX generating initialization code from user choices
  - Firmware offering, delivered per series (like STM32CubeF4) with:
    - An STM32 Abstraction Layer embedded software: STM32Cube HAL
    - A consistent set of Middleware: RTOS, USB, TCP/IP, Graphics, …


STM32CubeMX
Initialization Code generation based on user choices
STM32CubeL0
STM32CubeL1
STM32CubeF0
STM32CubeF1
STM32CubeF3
STM32CubeF2
STM32CubeF4
STM32Cube HAL: Portable API within all series - Middleware stacks when applicable: RTOS, USB, TCP/IP, Graphics, …

STM32CubeMX



Pinout Wizard

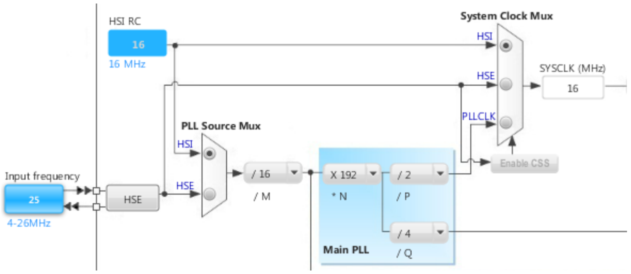


⊟ Basic Parameters	
Baud Rate	115200 Bits/s
Word Length	8 Bits (including Parity)
Parity	None
Stop Bits	1
⊟ Advanced Parameters	
Data Direction	Receive and Transmit
Over Sampling	16 Samples

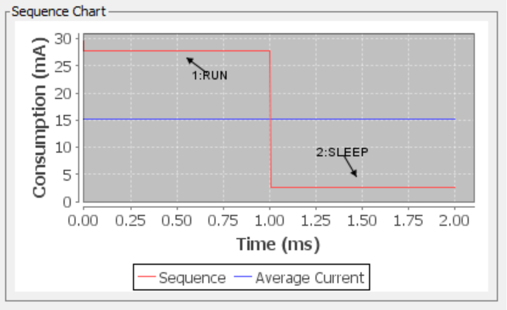
Baud Rate
BaudRate must be between **110 Bits/s** and **10.5 MBits/s**.

Peripherals & Middleware Wizard

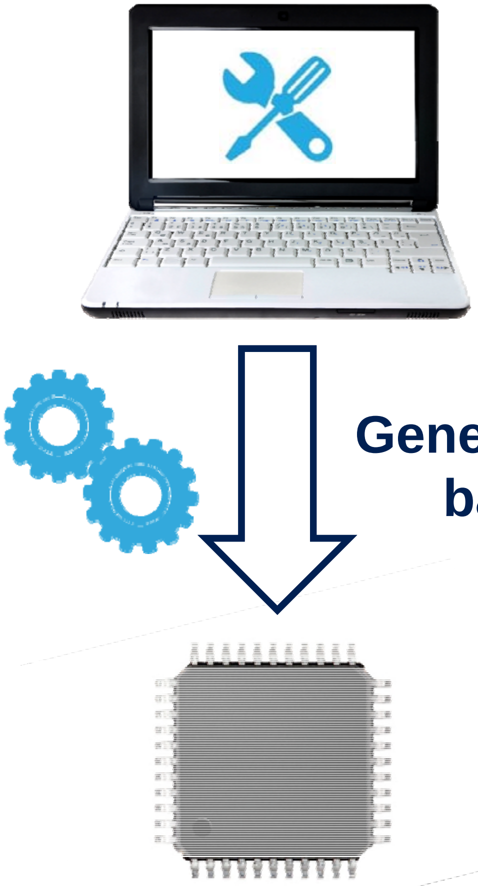
Clock Tree wizard



Power Consumption Wizard



STM32CubeMX

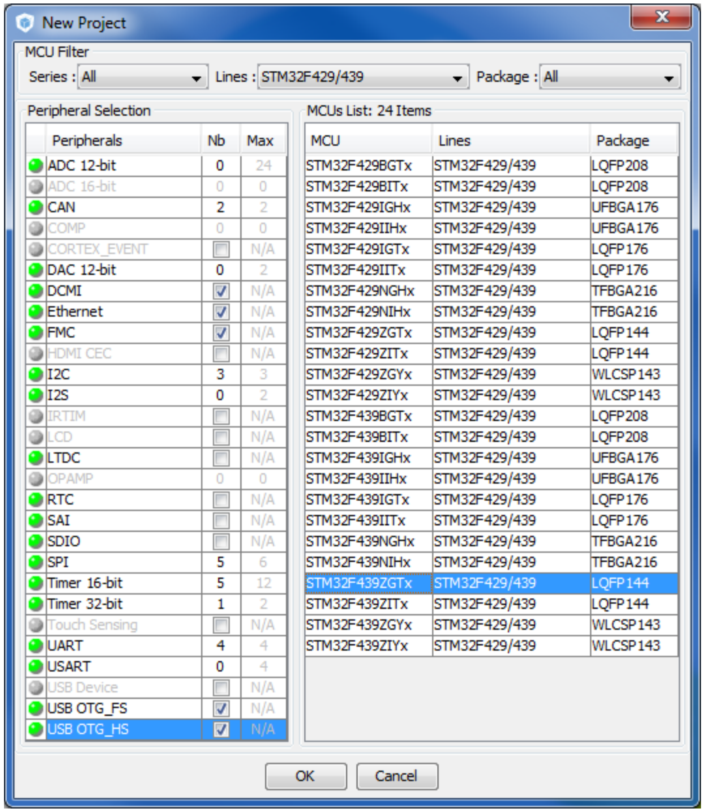


Generates Initialization C Code based on user choices !

STM32CubeMX

MCU Selector

- Filter by:
  - Series
  - Line
  - Package
  - Peripherals



Series : All
Lines : STM32F429/439
Package : All

Peripheral Selection

	Peripherals	Nb	Max
🟢	ADC 12-bit	0	24
⚪	ADC 16-bit	0	0
🟢	CAN	2	2
⚪	COMP	0	0
⚪	CORTEX_EVENT	☐	N/A
🟢	DAC 12-bit	0	2
🟢	DCMI	☑	N/A
🟢	Ethernet	☑	N/A
🟢	FMC	☑	N/A
⚪	HDMI CEC	☐	N/A
🟢	I2C	3	3
🟢	I2S	0	2
⚪	IRTIM	☐	N/A
⚪	LCD	☐	N/A
🟢	LTDC	☐	N/A
⚪	OPAMP	0	0
🟢	RTC	☐	N/A
🟢	SAI	☐	N/A
🟢	SDIO	☐	N/A
🟢	SPI	5	6
🟢	Timer 16-bit	5	12
🟢	Timer 32-bit	1	2
⚪	Touch Sensing	☐	N/A
🟢	UART	4	4
🟢	USART	0	4
⚪	USB Device	☐	N/A
🟢	USB OTG_FS	☑	N/A
🟢	USB OTG_HS	☑	N/A

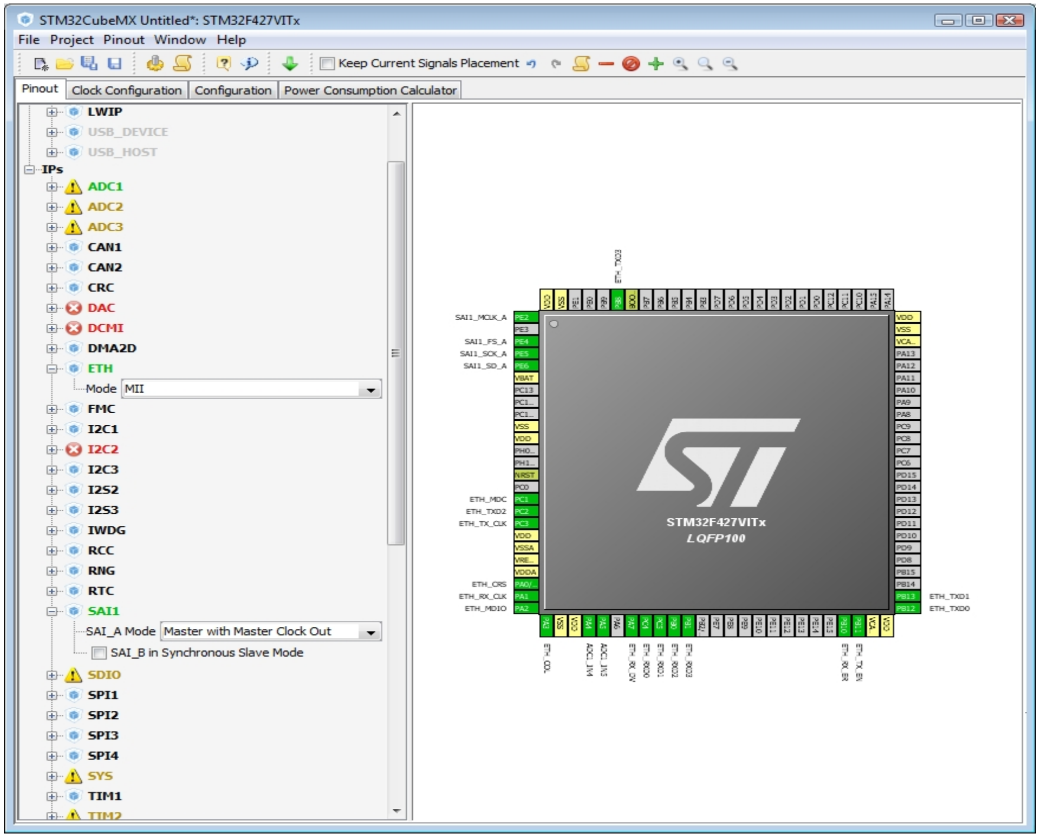
MCUs List: 24 Items

MCU	Lines	Package
STM32F429BGTx	STM32F429/439	LQFP208
STM32F429BITx	STM32F429/439	LQFP208
STM32F429IGHx	STM32F429/439	UFBGA176
STM32F429IIHx	STM32F429/439	UFBGA176
STM32F429IGTx	STM32F429/439	LQFP176
STM32F429IITx	STM32F429/439	LQFP176
STM32F429NGHx	STM32F429/439	TFBGA216
STM32F429NIHx	STM32F429/439	TFBGA216
STM32F429ZGTx	STM32F429/439	LQFP144
STM32F429ZITx	STM32F429/439	LQFP144
STM32F429ZGYx	STM32F429/439	WLCSP143
STM32F429ZIYx	STM32F429/439	WLCSP143
STM32F439BGTx	STM32F429/439	LQFP208
STM32F439BITx	STM32F429/439	LQFP208
STM32F439IGHx	STM32F429/439	UFBGA176
STM32F439IIHx	STM32F429/439	UFBGA176
STM32F439IGTx	STM32F429/439	LQFP176
STM32F439IITx	STM32F429/439	LQFP176
STM32F439NGHx	STM32F429/439	TFBGA216
STM32F439NIHx	STM32F429/439	TFBGA216
STM32F439ZGTx	STM32F429/439	LQFP144
STM32F439ZITx	STM32F429/439	LQFP144
STM32F439ZGYx	STM32F429/439	WLCSP143
STM32F439ZIYx	STM32F429/439	WLCSP143

STM32CubeMX

Pin-out configuration

- Pinout from:
  - Peripheral tree
  - Manually
- Automatic signal remapping
- Management of dependencies between peripherals

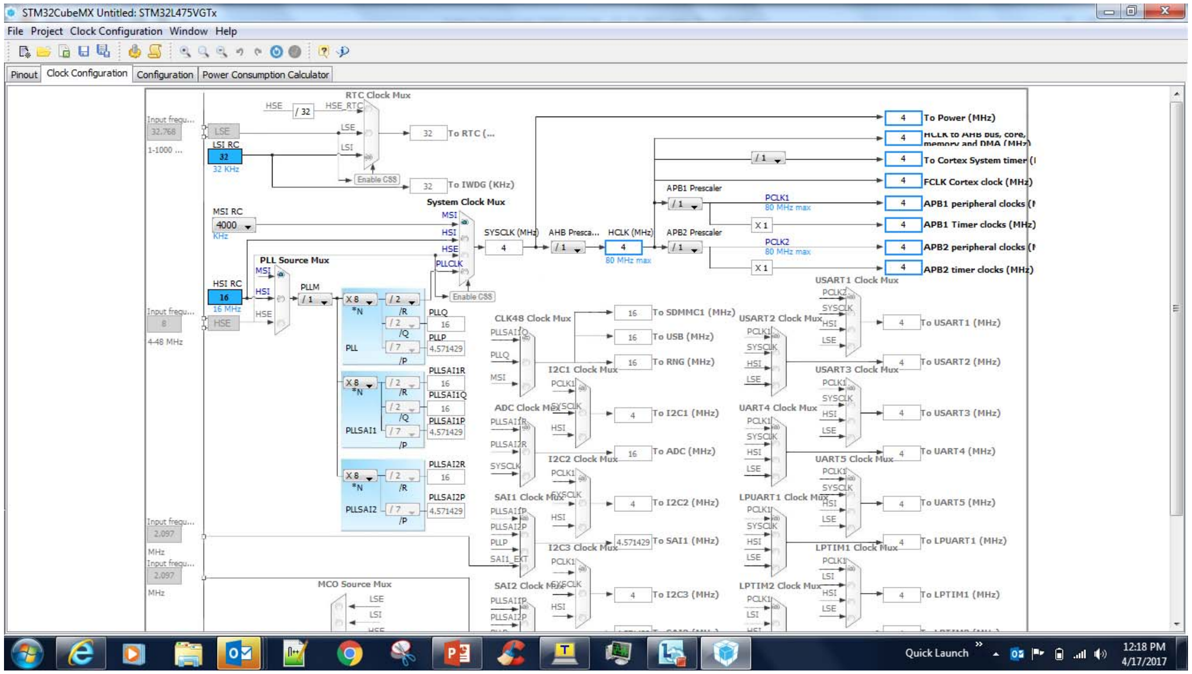


- LWIP
- USB_DEVICE
- USB_HOST
- IPs
  - ADC1
  - ADC2
  - ADC3
  - CAN1
  - CAN2
  - CRC
  - DAC
  - DCMI
  - DMA2D
  - ETH
    - Mode MII
  - FMC
  - I2C1
  - I2C2
  - I2C3
  - I2S2
  - I2S3
  - IWDG
  - RCC
  - RNG
  - RTC
  - SAI1
    - SAI_A Mode Master with Master Clock Out
    - □ SAI_B in Synchronous Slave Mode
  - SDIO
  - SPI1
  - SPI2
  - SPI3
  - SPI4
  - SYS
  - TIM1
  - TIM2

STM32CubeMX

Clock tree

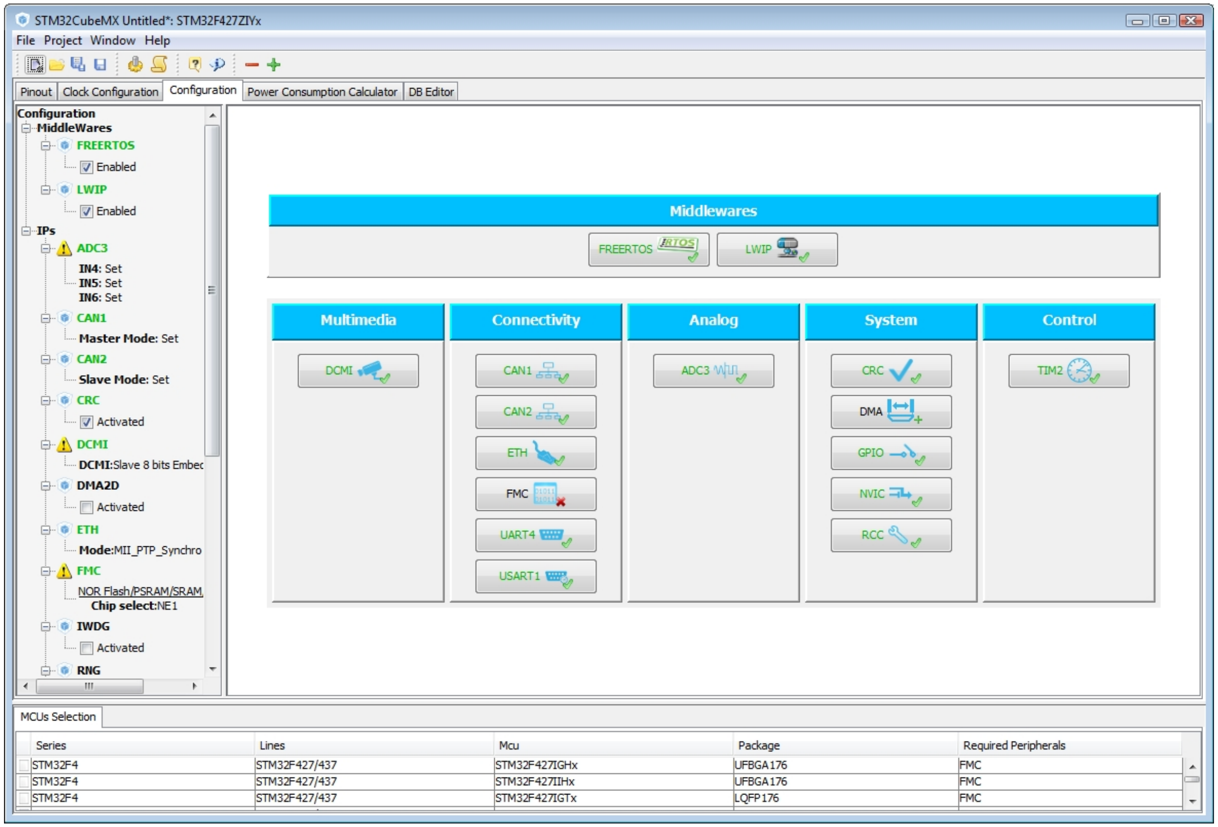
- Immediate display of all clock values
- Management of all clock constraints
- Highlight of errors



STM32CubeMX

Peripheral configuration

- Global view of used peripherals and middleware
- Highlight of configuration errors
- Manage:
  - GPIO
  - Interrupts
  - DMA

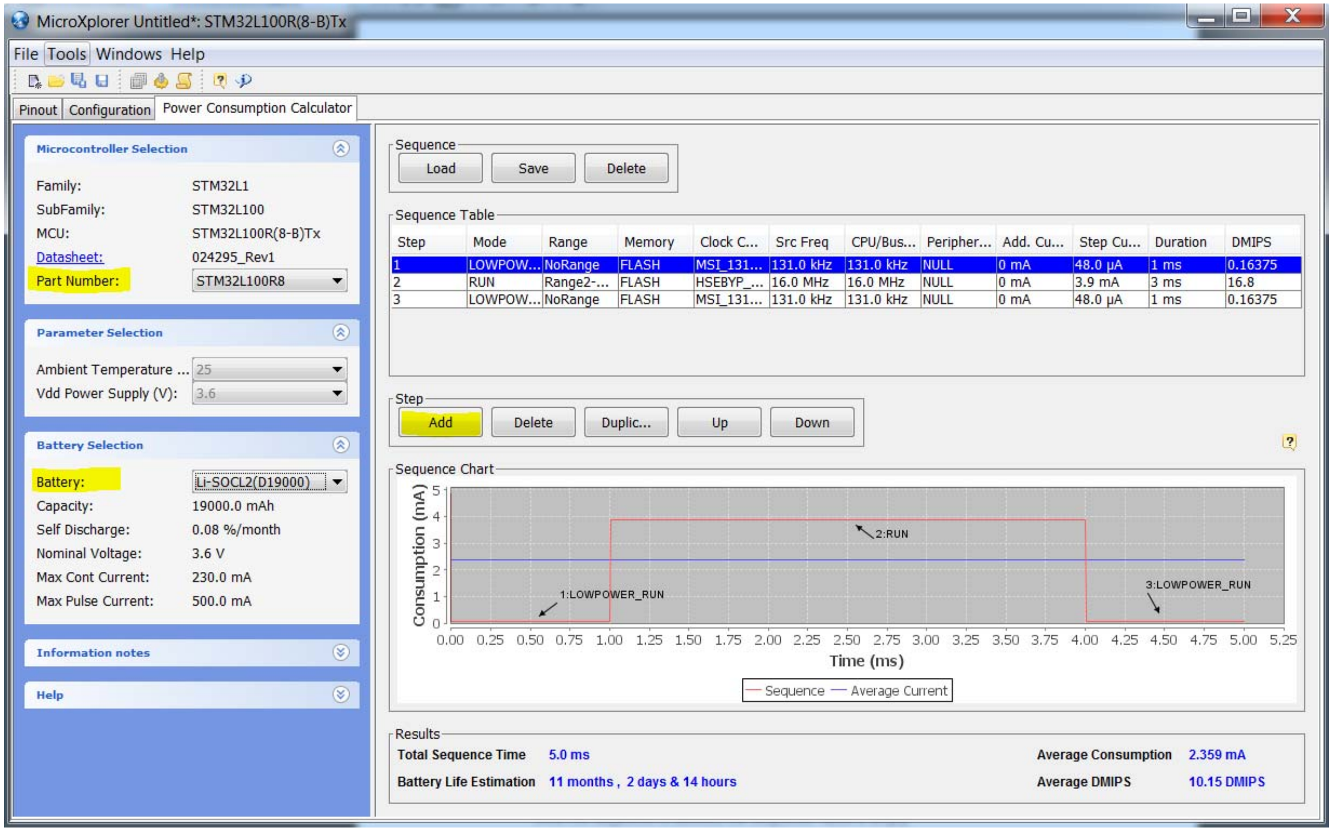


Configuration
- MiddleWares
  - FREERTOS
    - ☑ Enabled
  - LWIP
    - ☑ Enabled
- IPs
  - ⚠ ADC3
    - **IN4**: Set
    - **IN5**: Set
    - **IN6**: Set
  - CAN1
    - **Master Mode**: Set
  - CAN2
    - **Slave Mode**: Set
  - CRC
    - ☑ Activated
  - ⚠ DCMI
    - **DCMI:**Slave 8 bits Embec
  - DMA2D
    - ☐ Activated
  - ETH
    - **Mode:**MII_PTP_Synchro
  - ⚠ FMC
    - NOR Flash/PSRAM/SRAM,
      Chip select:NE1
  - IWDG
    - ☐ Activated
  - RNG

Series	Lines	Mcu	Package	Required Peripherals
STM32F4	STM32F427/437	STM32F427IGHx	UFBGA176	FMC
STM32F4	STM32F427/437	STM32F427IIHx	UFBGA176	FMC
STM32F4	STM32F427/437	STM32F427IGTx	LQFP176	FMC

Power consumption calculator

- Power step definitions
- Battery selection
- Creation of consumption graph
- Display of
  - Average consumption
  - Average DMIPS
  - Battery lifetime



Microcontroller Selection

Family: STM32L1
SubFamily: STM32L100
MCU: STM32L100R(8-B)Tx
Datasheet: 024295_Rev1
Part Number: STM32L100R8

Parameter Selection

Ambient Temperature ...: 25
Vdd Power Supply (V): 3.6

Battery Selection

Battery: Li-SOCL2(D19000)
Capacity: 19000.0 mAh
Self Discharge: 0.08 %/month
Nominal Voltage: 3.6 V
Max Cont Current: 230.0 mA
Max Pulse Current: 500.0 mA

Information notes

Help

Sequence

Sequence Table

Step	Mode	Range	Memory	Clock C...	Src Freq	CPU/Bus...	Peripher...	Add. Cu...	Step Cu...	Duration	DMIPS
1	LOWPOW...	NoRange	FLASH	MSI_131...	131.0 kHz	131.0 kHz	NULL	0 mA	48.0 µA	1 ms	0.16375
2	RUN	Range2-...	FLASH	HSEBYP_...	16.0 MHz	16.0 MHz	NULL	0 mA	3.9 mA	3 ms	16.8
3	LOWPOW...	NoRange	FLASH	MSI_131...	131.0 kHz	131.0 kHz	NULL	0 mA	48.0 µA	1 ms	0.16375

Step

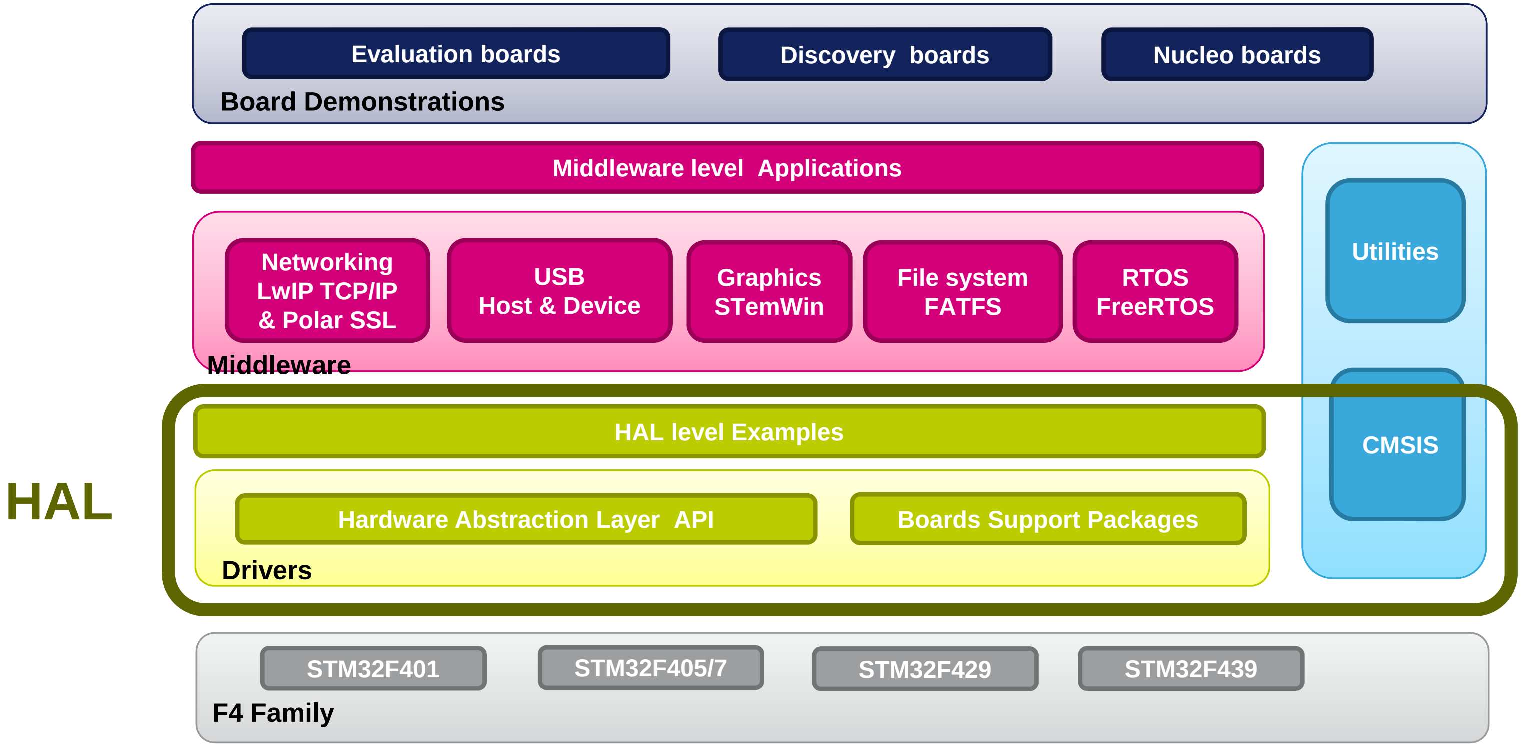
Sequence Chart

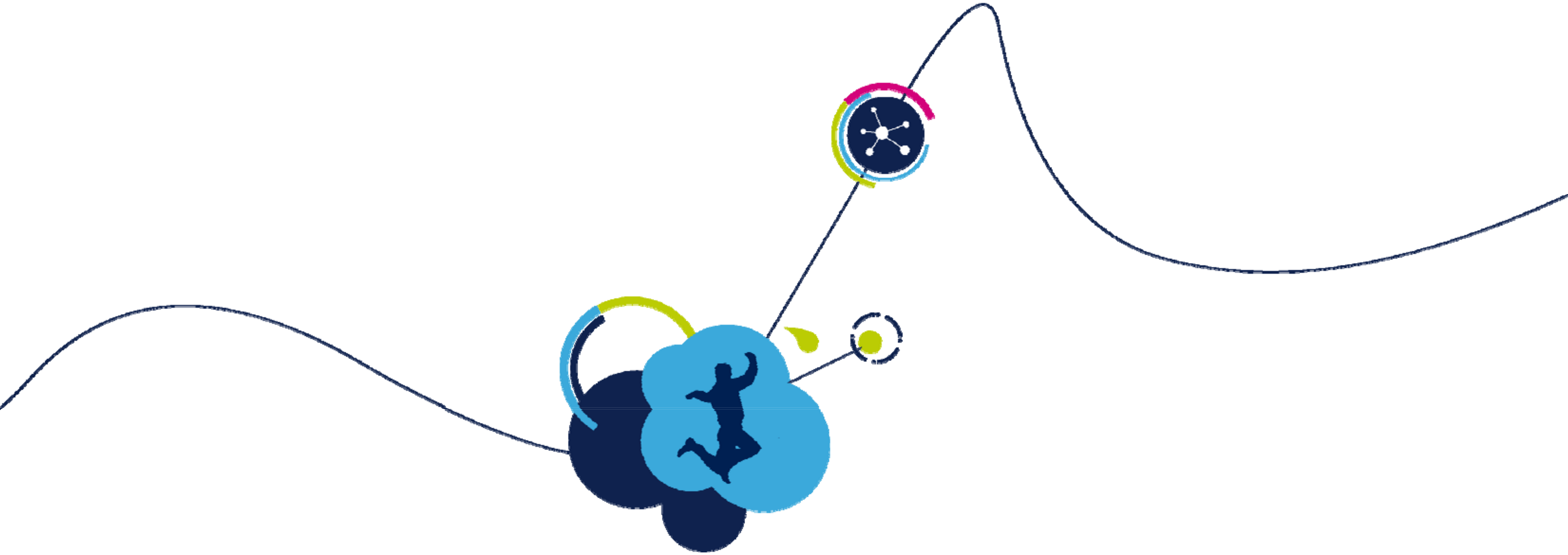
Results

Total Sequence Time 5.0 ms
Battery Life Estimation 11 months , 2 days & 14 hours
Average Consumption 2.359 mA
Average DMIPS 10.15 DMIPS

Technology Tour 2017

STM32Cube Firmware Components



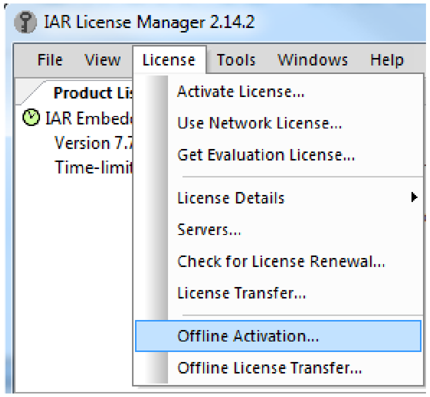


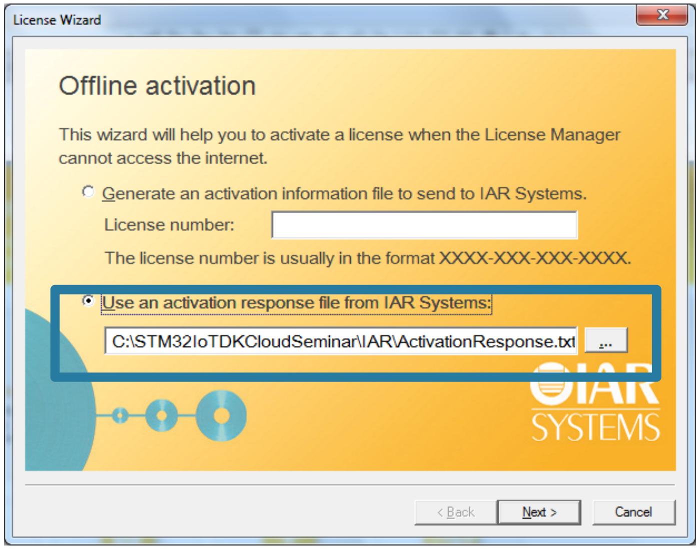
IAR License Installation

Technology Tour 2017

IAR License Installation

- Open IAR
- Go to `Help->License Manager`
- Go to `License->Offline Activation…`
- Use `C:\STM32IoTDKCloudSeminar\IAR\ActivationResponse.txt` for the activation response.





Offline activation

This wizard will help you to activate a license when the License Manager cannot access the internet.

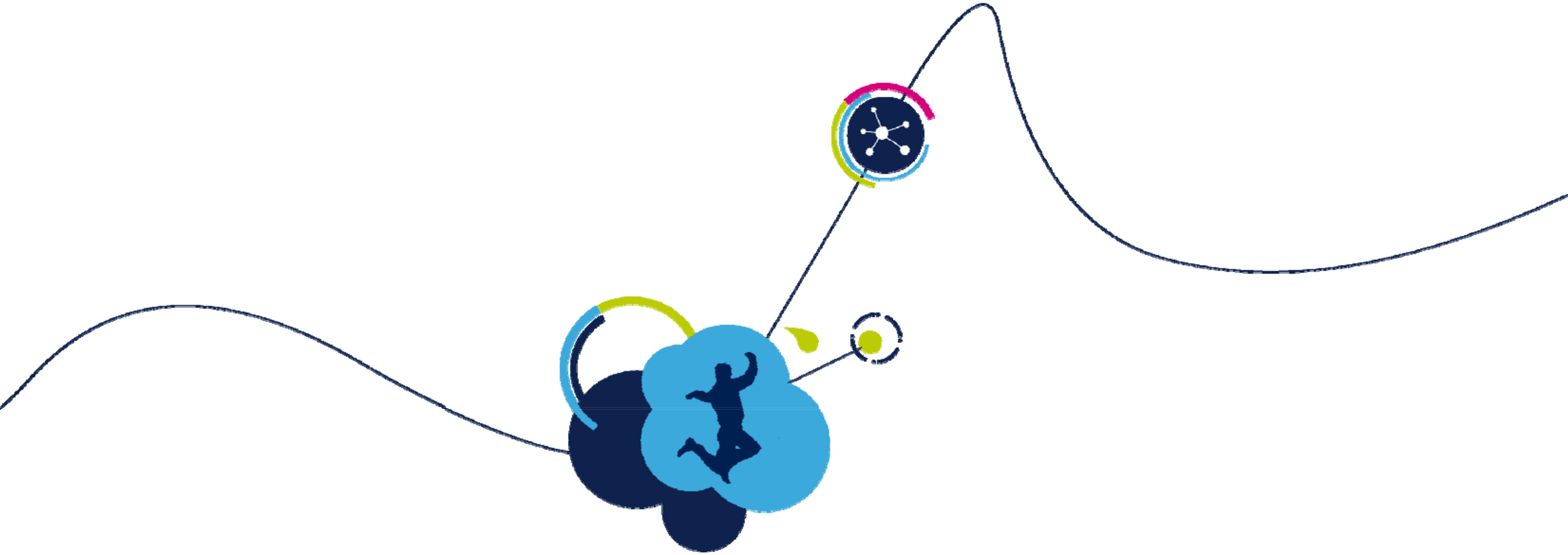
○ Generate an activation information file to send to IAR Systems.

License number:

The license number is usually in the format XXXX-XXX-XXX-XXXX.

● Use an activation response file from IAR Systems:

C:\STM32IoTDKCloudSeminar\IAR\ActivationResponse.txt



ST-Link Installation

life.augmented

Technology Tour 2017

ST-Link Utility Installation

- The ST-Link Utility allows typical flash program / erase / upload / download functions via the ST-LINK/V2 debugger, onboard the STM32L475 Discovery IoT node Board. It also installs the Windows device drivers necessary for the ST-LINK/V2 debugger.

- Run the installer that can be found at: `C:\STM32IoTDKCloudSeminar\Software\STM32 ST-LINK Utility_v4.0.0.exe`

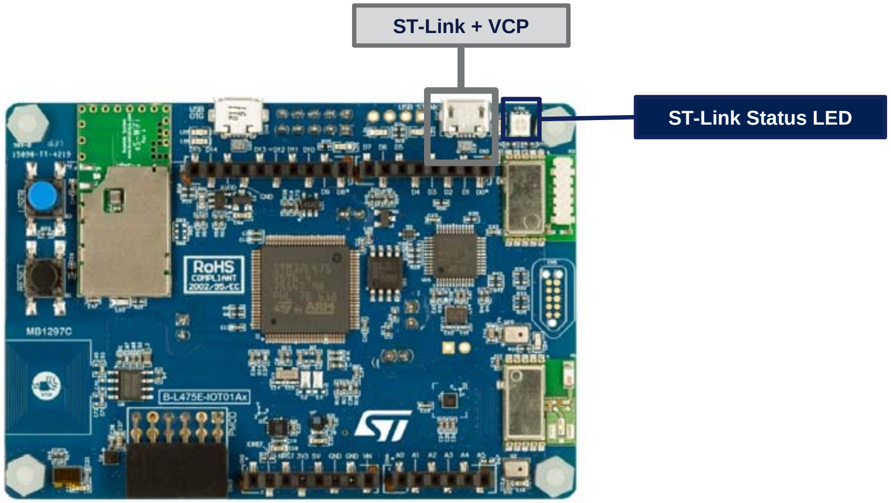
Board Distribution

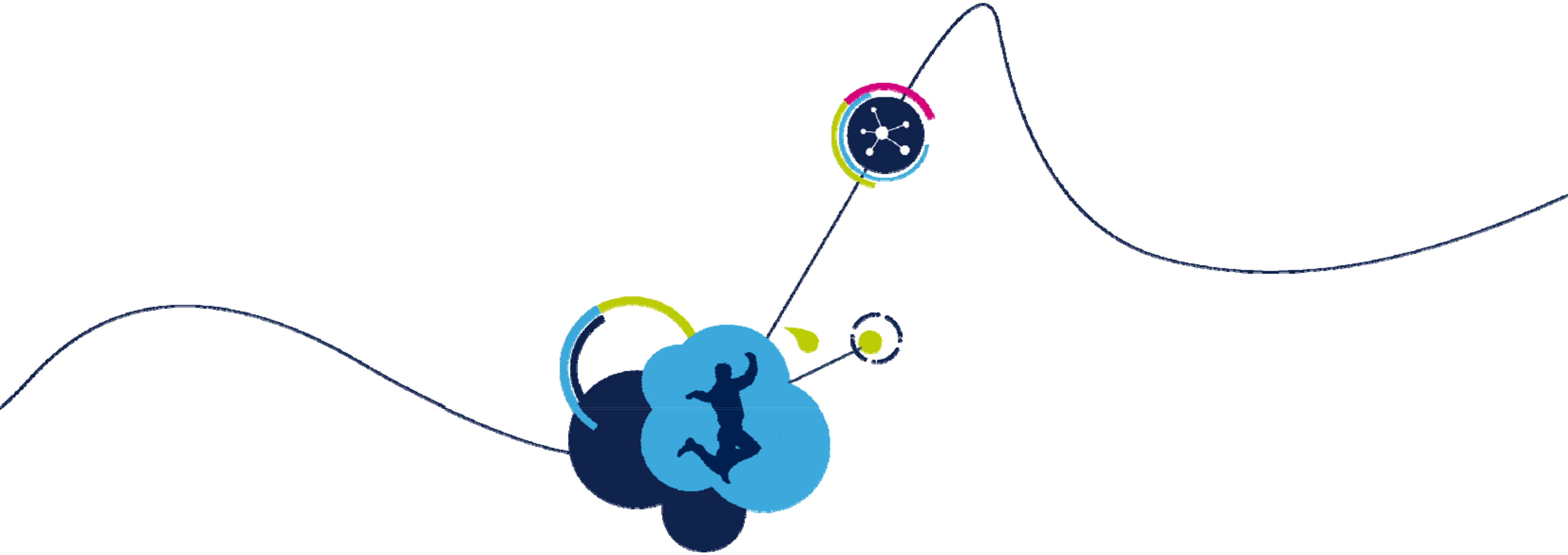
- Each board will have a label with a unique number. During the Lab sessions, this number will be referred to as your Participant Number. (Always use decimal, 2 digits)

Your Participant Number

ST-Link Driver installation

- Connect USB ST-LINK to your PC.
- The board is powered thorough ST-LINK.
- The ST-Link Status LED will be steady when ST-Link is recognized.





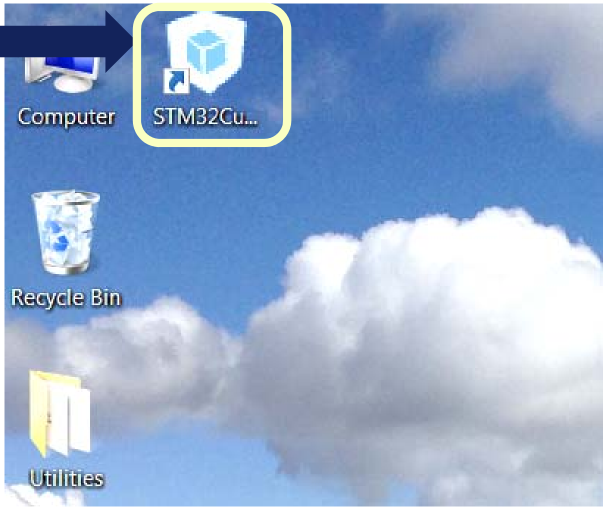
Lab1: Getting Started with STM32CubeMX

life.augmented

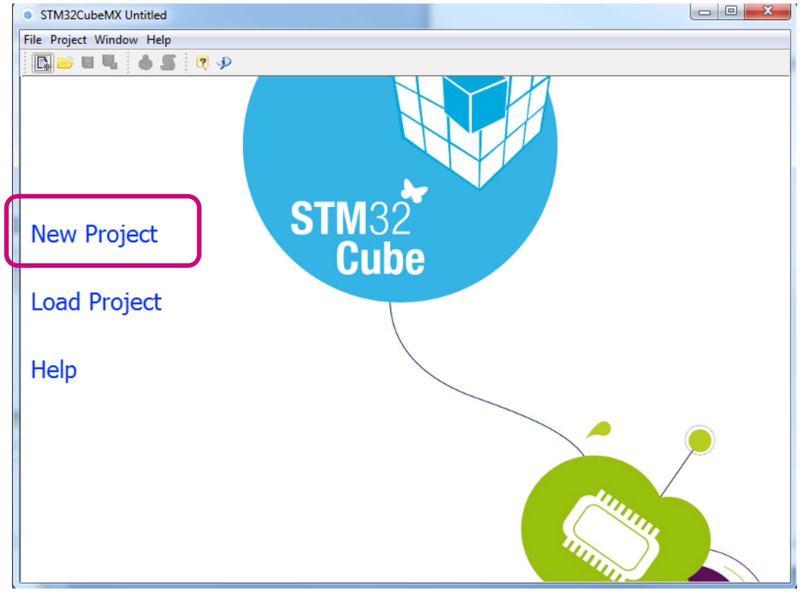
Technology Tour 2017

Create New Project

1. From your desktop open STM32CubeMX software.



2. Click New Project


STM32CubeMX Untitled

File Project Window Help

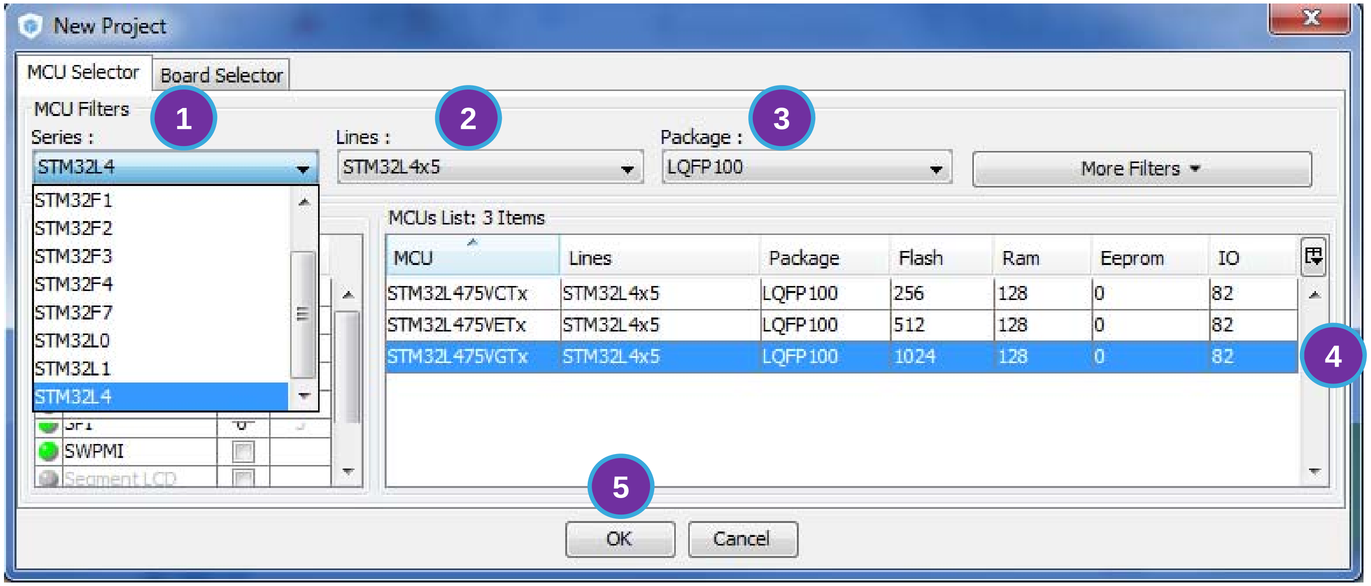
New Project

Load Project

Help

Select the Microcontroller

1. Under Series select **STM32L4**
2. Under Lines select **STM32L4x5**
3. Under Package select **LQFP100**
4. Select **STM32L475VGTx**
5. Click “**OK**”



Series : STM32L4

Lines : STM32L4x5

Package : LQFP100

MCUs List: 3 Items

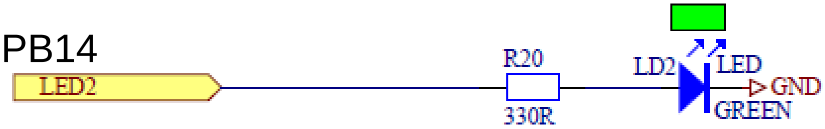
MCU	Lines	Package	Flash	Ram	Eeprom	IO
STM32L475VCTx	STM32L4x5	LQFP100	256	128	0	82
STM32L475VETx	STM32L4x5	LQFP100	512	128	0	82
STM32L475VGTx	STM32L4x5	LQFP100	1024	128	0	82

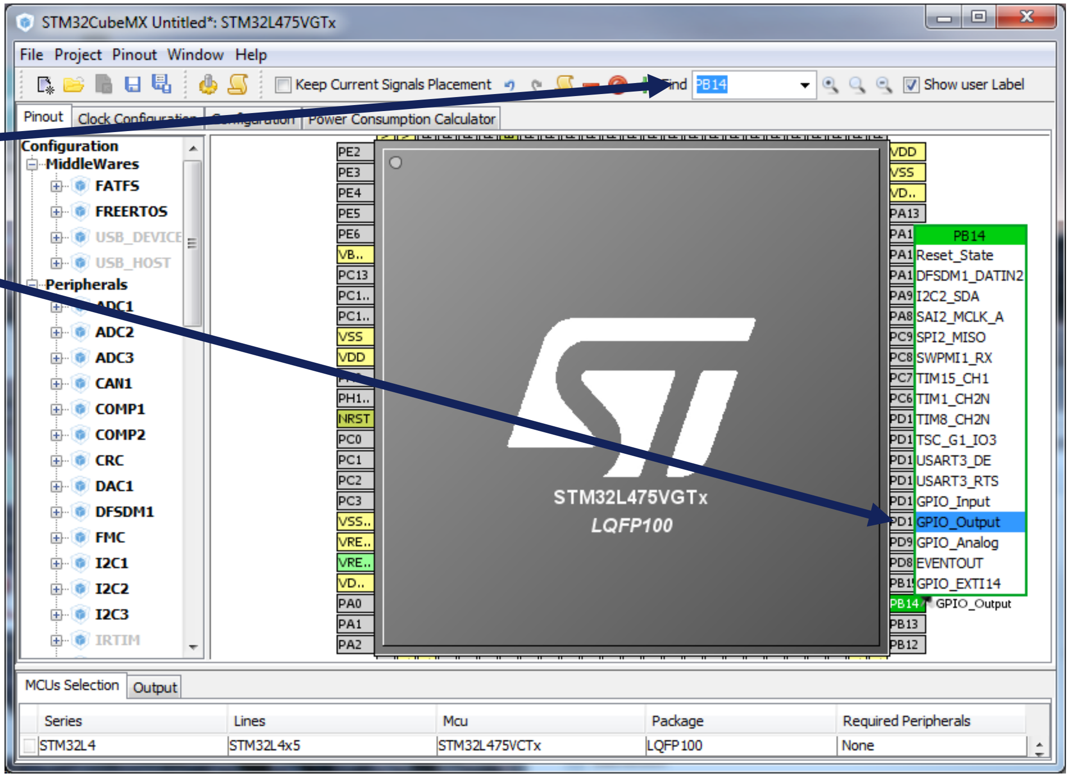
GPIO selection

- In this example we are going to use LED2 present on the IoT DK board.

1. Use the find toolbar and type **PB14**.

2. Select **PB14** and set it to **GPIO_Output** mode.





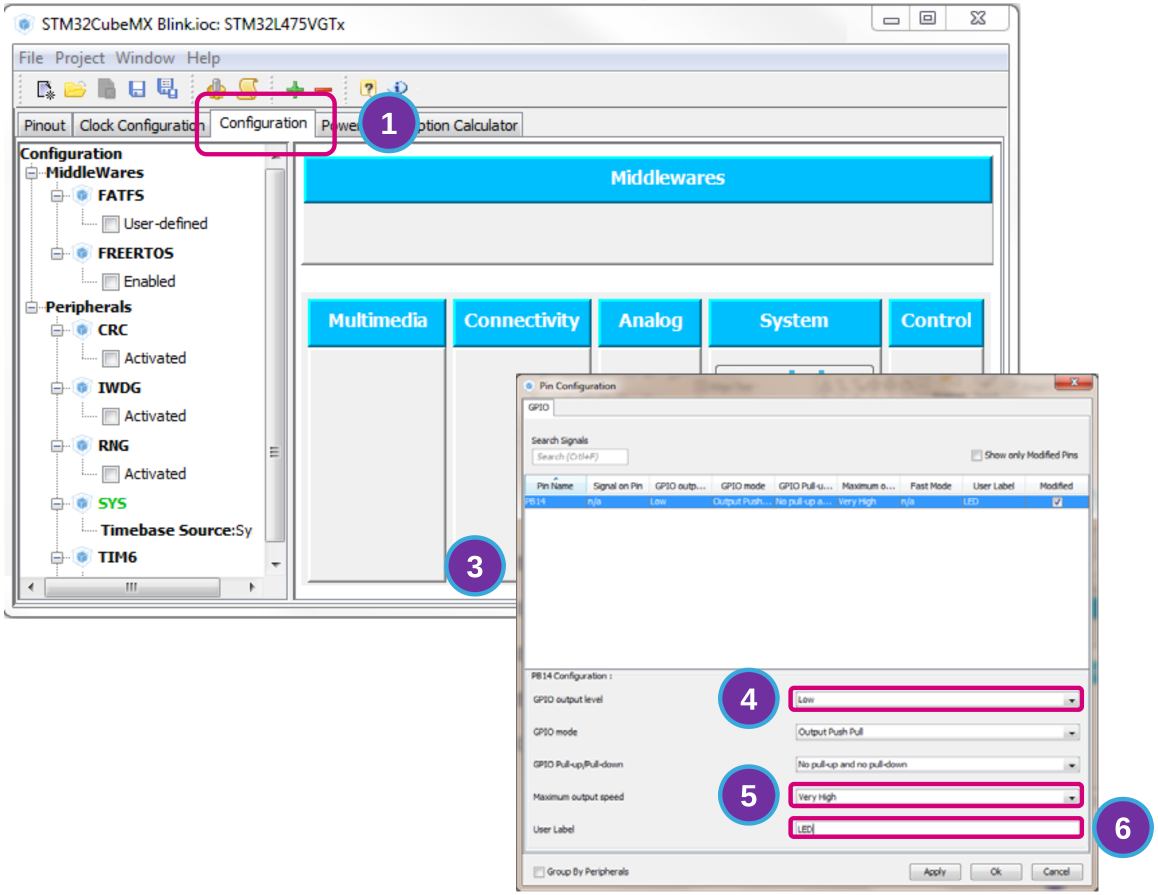
Series	Lines	Mcu	Package	Required Peripherals
STM32L4	STM32L4x5	STM32L475VCTx	LQFP100	None

life.augmented

Technology Tour 2017

GPIO Configuration

1. Select the **Configuration** tab
2. Select **GPIO** under System.
3. Select **PB14**.
4. Set the GPIO output level to **High**.
5. Set the Maximum output speed to **Very High.**
6. Set the User Label to **LED.**
7. Click Ok.



Pin Name	Signal on Pin	GPIO outp...	GPIO mode	GPIO Pull-u...	Maximum o...	Fast Mode	User Label	Modified
PB14	n/a	Low	Output Push...	No pull-up a...	Very High	n/a	LED	☑

PB14 Configuration :

- GPIO output level: Low
- GPIO mode: Output Push Pull
- GPIO Pull-up/Pull-down: No pull-up and no pull-down
- Maximum output speed: Very High
- User Label: LED

Project Settings

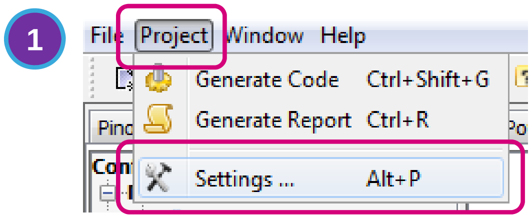
1. Open the project Settings (Alt + P).

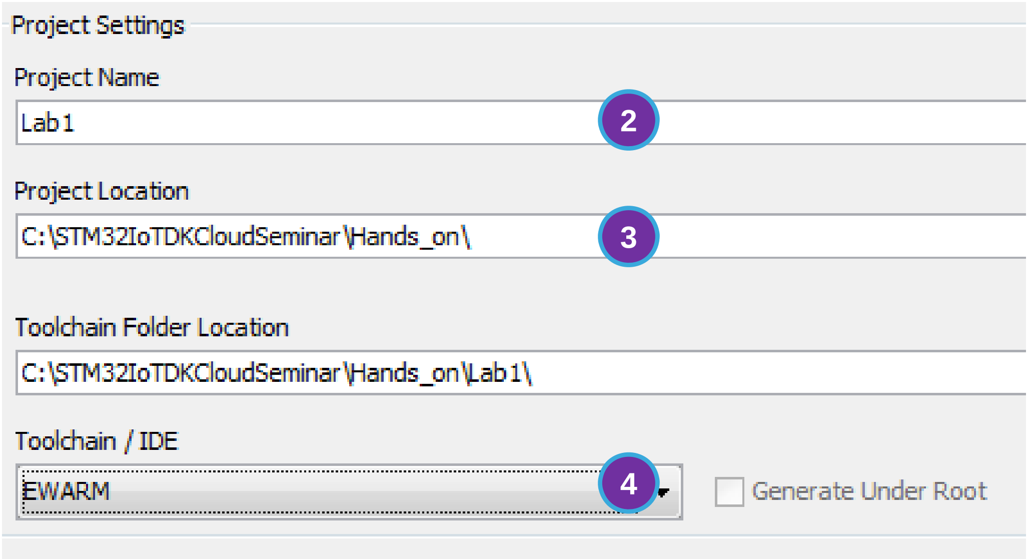
2. Set the project name to `Lab1`.

3. Set the project location `C:\STM32IoTDKCloudSeminar\Hands_on\`

4. Set the IDE Toolchain to **EWARM**.

5. Click **OK**.





Project Settings

Project Name

Lab1

Project Location

C:\STM32IoTDKCloudSeminar\Hands_on\

Toolchain Folder Location

C:\STM32IoTDKCloudSeminar\Hands_on\Lab1\

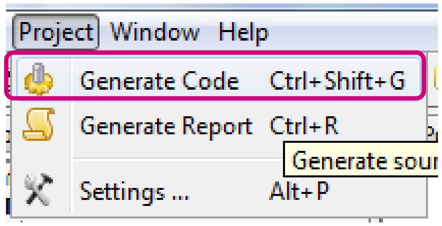
Toolchain / IDE

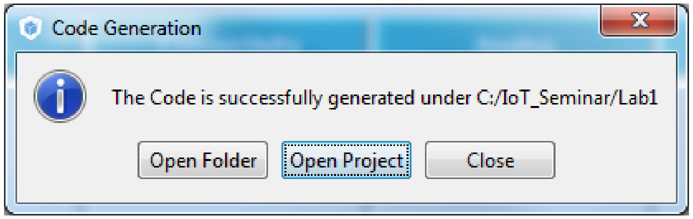
EWARM

□ Generate Under Root

Generate and Open the Project

- **Generate Code** (Ctrl + Shift + G)





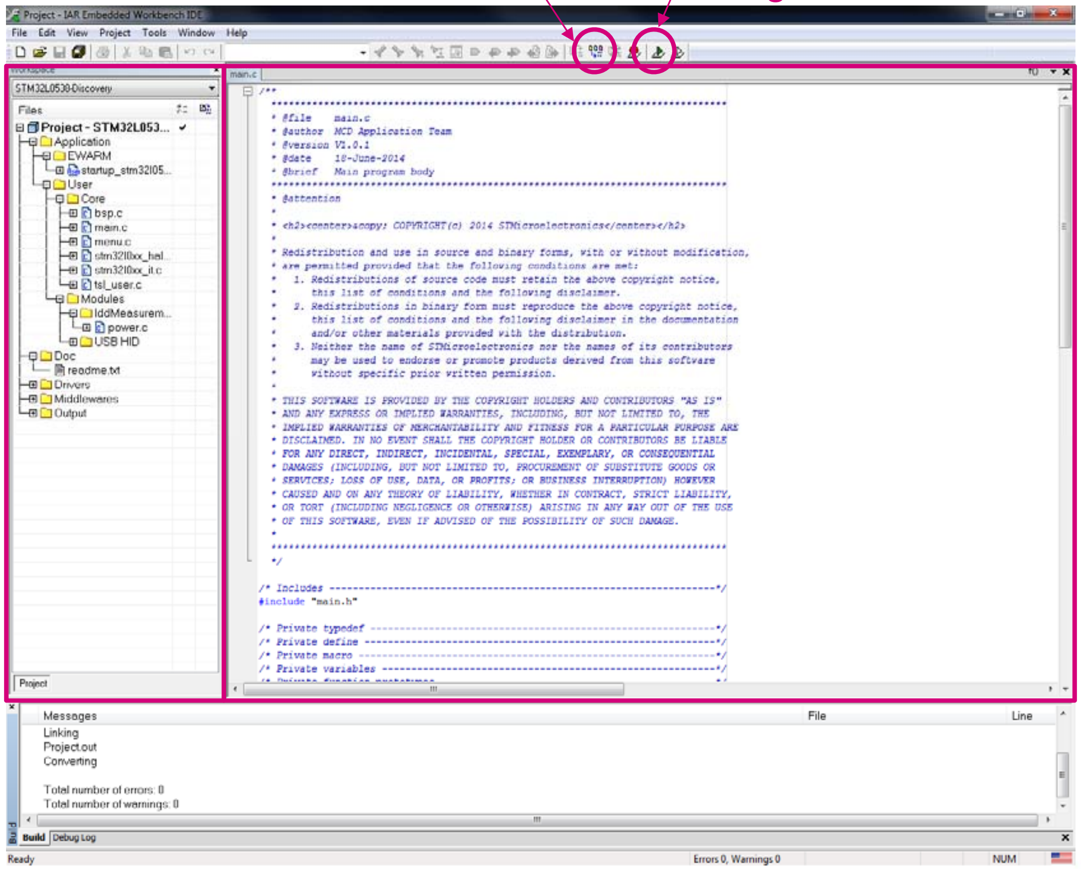
The Code is successfully generated under C:/IoT_Seminar/Lab1

- Click **Open Project.**

Inside IAR EWARM

Build Button

Debug Button



```
/**
  ******************************************************************************
  * @file    main.c
  * @author  MCD Application Team
  * @version V1.0.1
  * @date    18-June-2014
  * @brief   Main program body
  ******************************************************************************
  * @attention
  *
  * <h2><center>&copy; COPYRIGHT(c) 2014 STMicroelectronics</center></h2>
  *
  * Redistribution and use in source and binary forms, with or without modification,
  * are permitted provided that the following conditions are met:
  *   1. Redistributions of source code must retain the above copyright notice,
  *      this list of conditions and the following disclaimer.
  *   2. Redistributions in binary form must reproduce the above copyright notice,
  *      this list of conditions and the following disclaimer in the documentation
  *      and/or other materials provided with the distribution.
  *   3. Neither the name of STMicroelectronics nor the names of its contributors
  *      may be used to endorse or promote products derived from this software
  *      without specific prior written permission.
  *
  * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
  * AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
  * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
  * DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE
  * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
  * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
  * SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
  * CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
  * OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
  * OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
  *
  ******************************************************************************
  */

/* Includes ------------------------------------------------------------------*/
#include "main.h"

/* Private typedef -----------------------------------------------------------*/
/* Private define ------------------------------------------------------------*/
/* Private macro -------------------------------------------------------------*/
/* Private variables ---------------------------------------------------------*/
```

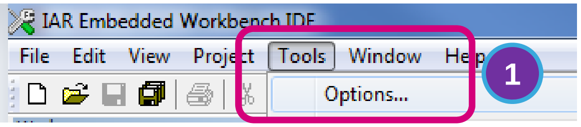
```
Linking
Project.out
Converting

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

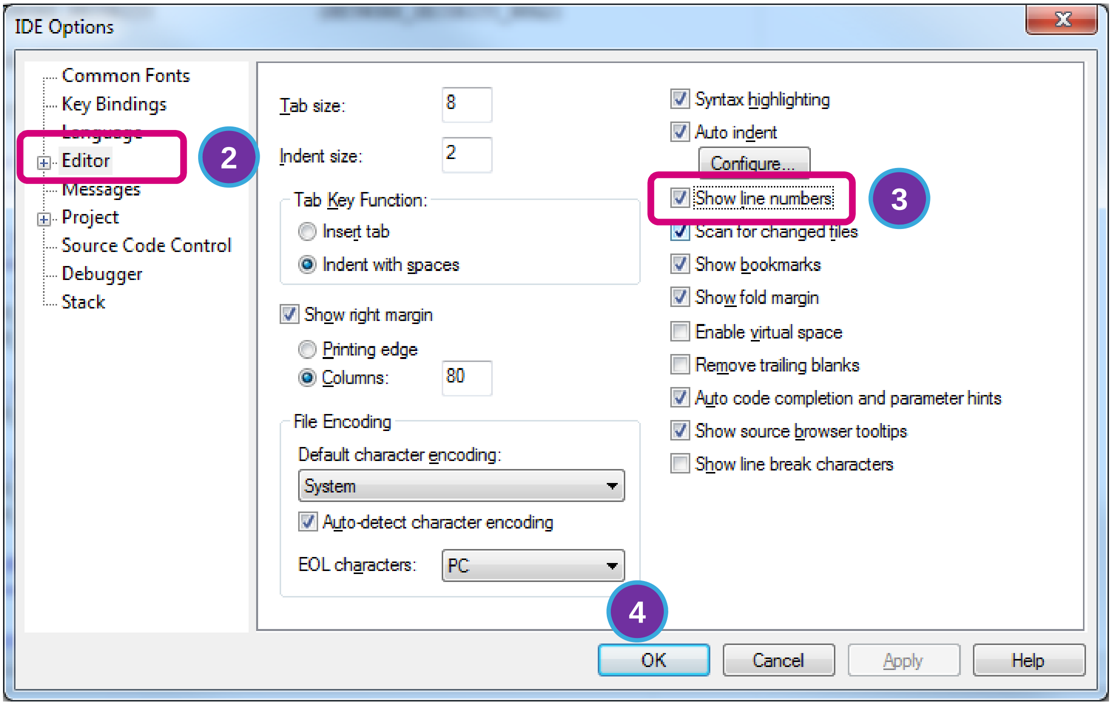
Project Window

Files Window

Configure IAR to Show Line Numbers

1. Go to Tools→Options
2. Select Editor
3. Check the Show line numbe
4. Click OK





Tab size: 8

Indent size: 2

Tab Key Function:
- ○ Insert tab
- ● Indent with spaces

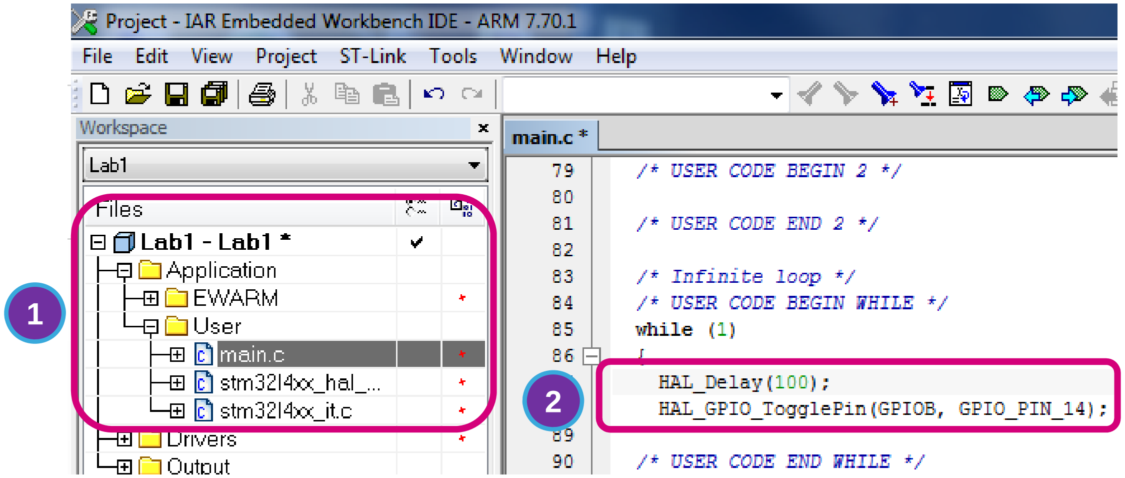
☑ Show right margin
- ○ Printing edge
- ● Columns: 80

File Encoding
- Default character encoding: System
- ☑ Auto-detect character encoding
- EOL characters: PC

- ☑ Syntax highlighting
- ☑ Auto indent
- ☑ Show line numbers
- ☑ Scan for changed files
- ☑ Show bookmarks
- ☑ Show fold margin
- □ Enable virtual space
- □ Remove trailing blanks
- ☑ Auto code completion and parameter hints
- ☑ Show source browser tooltips
- □ Show line break characters

Edit main.c

1. Expand the file tree and open **main.c** file



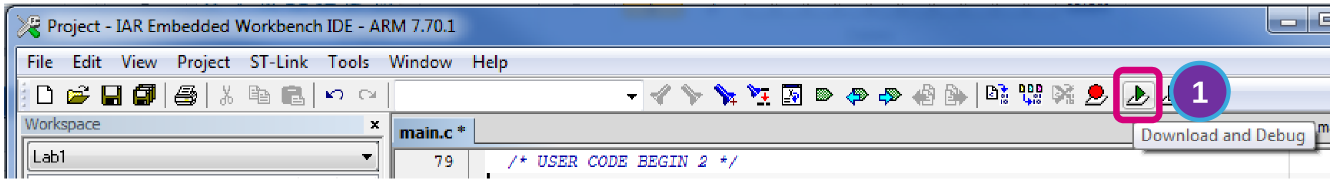
```c
  /* USER CODE BEGIN 2 */

  /* USER CODE END 2 */

  /* Infinite loop */
  /* USER CODE BEGIN WHILE */
  while (1)
  {
    HAL_Delay(100);
    HAL_GPIO_TogglePin(GPIOB, GPIO_PIN_14);

  /* USER CODE END WHILE */
```

2. Add the following code inside the **while(1)** loop:

```c
HAL_Delay(100);

HAL_GPIO_TogglePin(LED_GPIO_Port, LED_Pin);
```

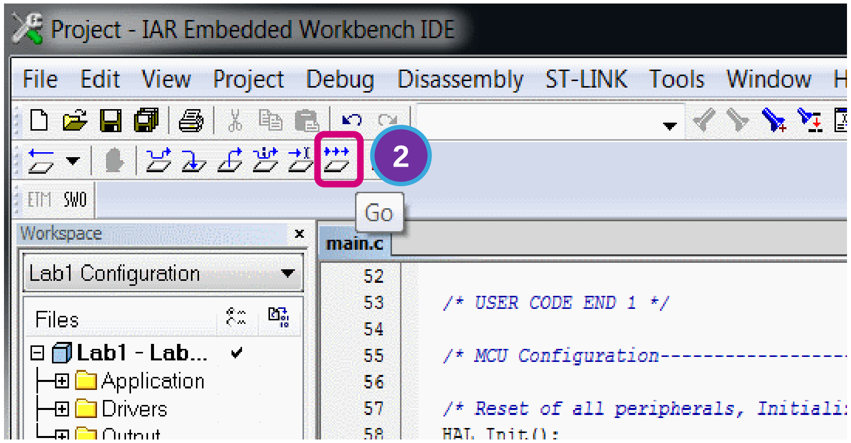
Load and Run

1. Click the GREEN ARROW to Build the Project, **Download** and start the **debugger**. (Ctrl + D)


Project - IAR Embedded Workbench IDE - ARM 7.70.1
File Edit View Project ST-Link Tools Window Help
Workspace
Lab1
main.c *
1
Download and Debug

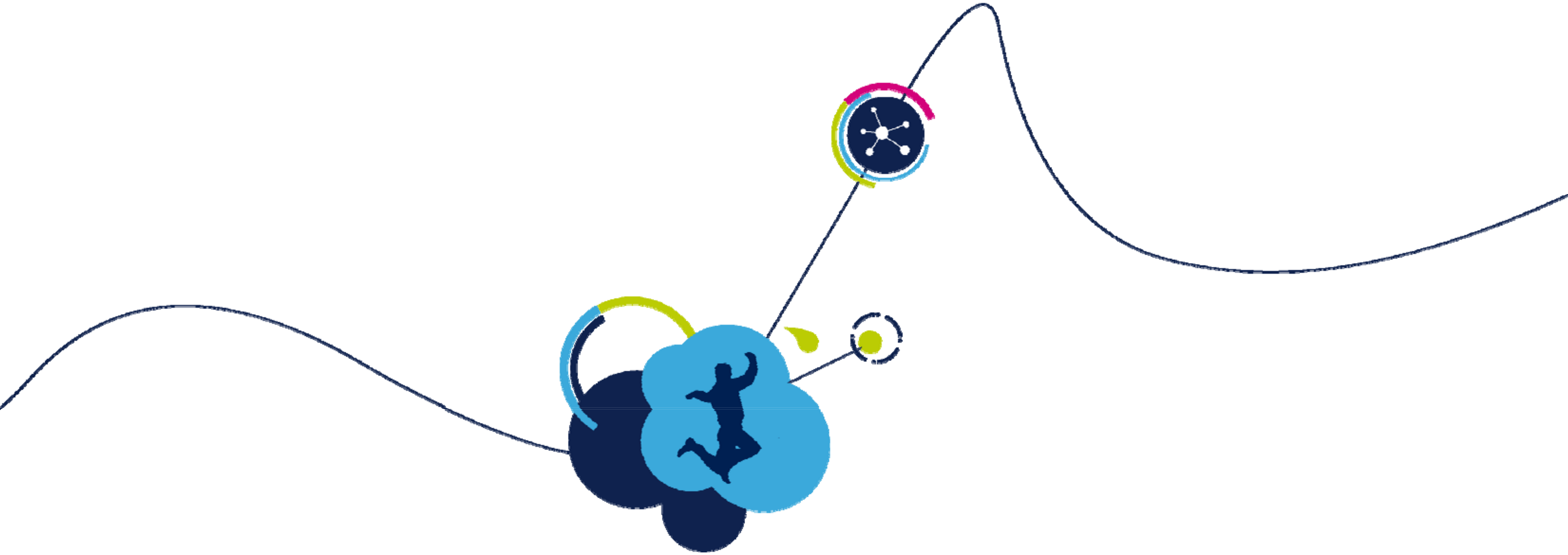
```
79    /* USER CODE BEGIN 2 */
```

2. Click the triple-arrow **GO** button! (F5)


Project - IAR Embedded Workbench IDE
File Edit View Project Debug Disassembly ST-LINK Tools Window H
ETM SWO
2
Go
Workspace
Lab1 Configuration
Files
Lab1 - Lab...
Application
Drivers
Output
main.c

```
52
53    /* USER CODE END 1 */
54
55    /* MCU Configuration-----------------
56
57    /* Reset of all peripherals, Initiali
58    HAL_Init();
```

3. Enjoy the LED!



Bluetooth® Low Energy Overview

life.augmented

Technology Tour 2017

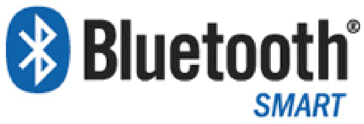
What is Bluetooth Low Energy?

- Bluetooth Low Energy technology
  - Short range wireless ISM 2.4 GHz
  - Optimized for ultra low power
    - <15 mA peak current
    - <50 uA average current
  - Fast connection procedure
  - Client server architecture
  - Low data throughput application

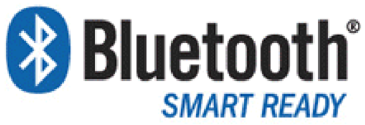
  - Security including privacy/authentication/authorization
    - Based on encryption AES128
  - Master Role : Central Device (Scanning, Initiating Connection)
  - Slave Role : Peripheral Device (Advertising)

Bluetooth Low Energy Branding

2011 **Two flavors**

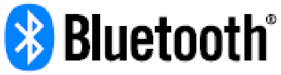


- Ultra low power consumption being a pure low energy implementation
- Months to years of lifetime on a standard coin cell battery



- Classic Bluetooth + Bluetooth low energy on a single chip
- These are the hub devices of the Bluetooth ecosystem

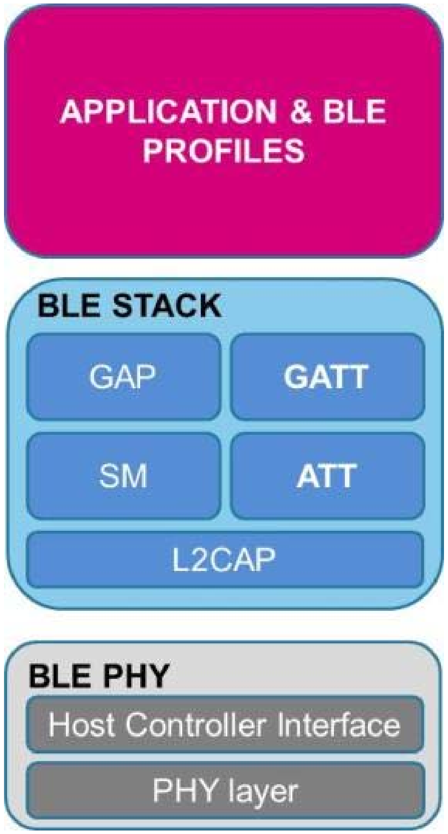
2017 **Back to one flavor**



- An implementation of the Bluetooth core system has only one Primary Controller which may be one of the following configurations:
  - BR/EDR Controller (3.0 and earlier)
  - LE (low energy) Controller (4.0 and newer)
  - Combined BR/EDR Controller portion and LE controller portion into a single Controller (4.0 and newer)

Source: Bluetooth SIG

Bluetooth Low Energy stack partitioning



- The application collects & computes the data to be transmitted over Bluetooth Low Energy.
- To transmit data , application use Bluetooth Low Energy stack **services and characteristics capabilities thanks to standard or proprietary application profile.**
- All communication in low energy takes place over the **G**eneric **A**ttribute **P**rofile **(GAP)**.
- PHY layer insures transmission over the air

BlueNRG-MS Solution - Available from ST

Integration

Flexibility

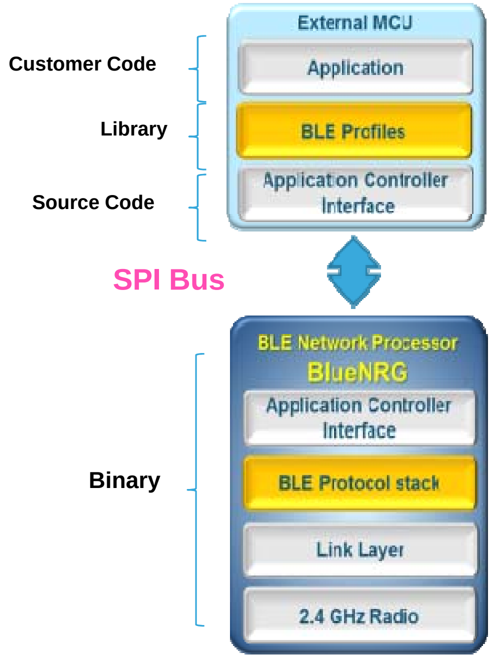
Best In Class

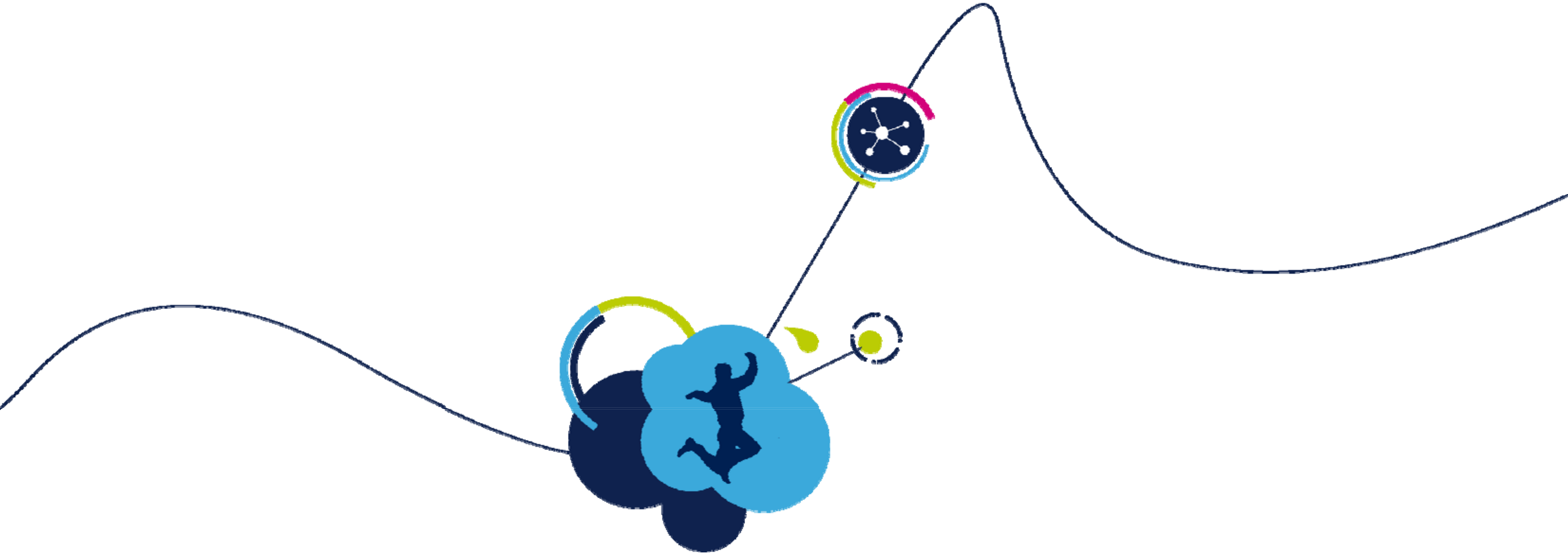
Low power

Small size

- **Single mode Bluetooth® Low Energy wireless network processor**
- 2.4GHz RF transceiver
- Cortex-M0 microcontroller (running the BT MS stack)
- AES 128-bit co-processor
- Master and Slave Mode Bluetooth Low Energy (4.1) Network Processor.
- On chip non-volatile Flash memory allows OTA stack upgrade.
- I_{CC}RX 7.3mA
- I_{CC}TX 8.2mA @ 0 dBm
- I_{CC}Sleep 1.7µA
- I_{CC}Shutdown 2.5nA

- **+ STM32 Consumption & Size**





Lab2: Bluetooth Low Energy pairing

life.augmented

Technology
Tour 2017

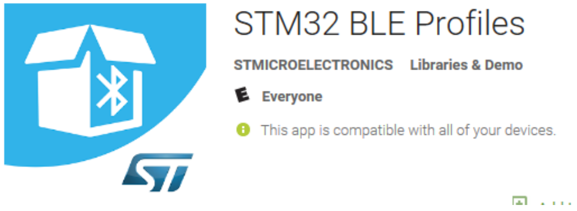
Goal

- This lab is to make sure that your BlueNRG device has a unique name and MAC address.

- This lab demonstrates a way to drive a BlueNRG device and communicate with a smartphone and display HeartRate data.

- The IoT DK will be used as server while the applet is a client.

- You need to download the **STM32 BLE Profiles** application available on App store and google play.

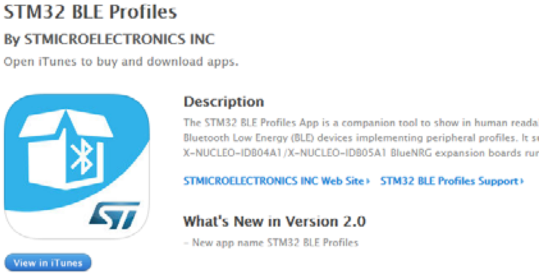


STM32 BLE Profiles

STMICROELECTRONICS Libraries & Demo

Everyone

This app is compatible with all of your devices.



STM32 BLE Profiles

By STMICROELECTRONICS INC

Open iTunes to buy and download apps.

Description

The STM32 BLE Profiles App is a companion tool to show in human reada
Bluetooth Low Energy (BLE) devices implementing peripheral profiles. It s
X-NUCLEO-IDB04A1/X-NUCLEO-IDB05A1 BlueNRG expansion boards rur

STMICROELECTRONICS INC Web Site › STM32 BLE Profiles Support ›

What's New in Version 2.0

- New app name STM32 BLE Profiles

View in iTunes

Download on the App Store

GET IT ON Google Play

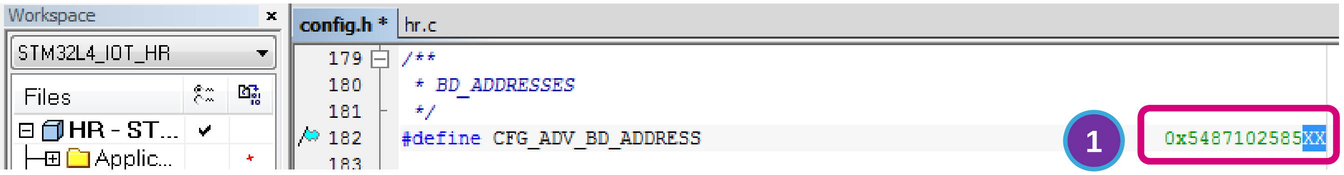
Open BlueNRG_HandsOn Project

- Now we are going to configure the `BlueNRG_HandsOn` program to give each BlueNRG module a unique MAC address and Unique device name. The device name will be used later to identify your board within the **ST BLE Profiles** app.

1. Close the previous IAR project.
2. Double click on HR.eww file located under `C:\STM32IoTDKCloudSeminar\Hands_On\BLE_and_Cloud\Projects\B-L475E-IOT01\Applications\BLE\HeartRate\EWARM`

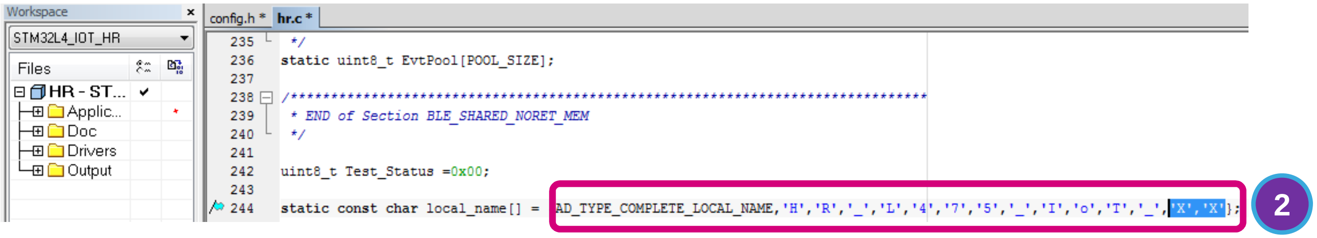
BlueNRG Module configuration

1. Open `config.h` file and replace the `‘X’, ‘X’` in the `CFG_ADV_BD_ADDRESS` with your participant number found on your box (Use decimal, 2 digits).



```
/**
 * BD_ADDRESSES
 */
#define CFG_ADV_BD_ADDRESS                                     0x5487102585XX
```

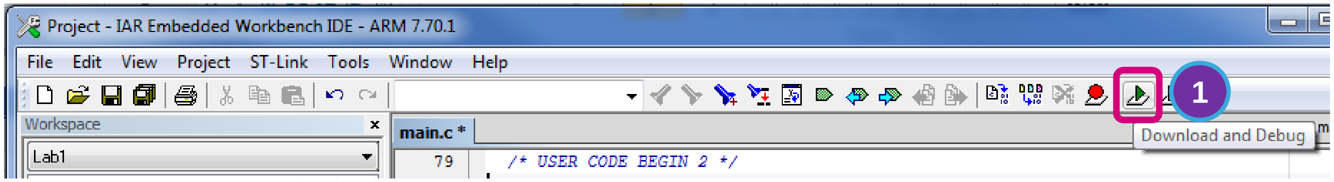
2. Open `hr.c` file and replace the XX in the `local_name` (line 244) table with your participant number (Use decimal, 2 digits).



```
235   */
236  static uint8_t EvtPool[POOL_SIZE];
237
238  /*************************************************************************
239   * END of Section BLE_SHARED_NORET_MEM
240   */
241
242  uint8_t Test_Status =0x00;
243
244  static const char local_name[] = AD_TYPE_COMPLETE_LOCAL_NAME,'H','R','_','L','4','7','5','_','I','o','T','_','X','X'};
```

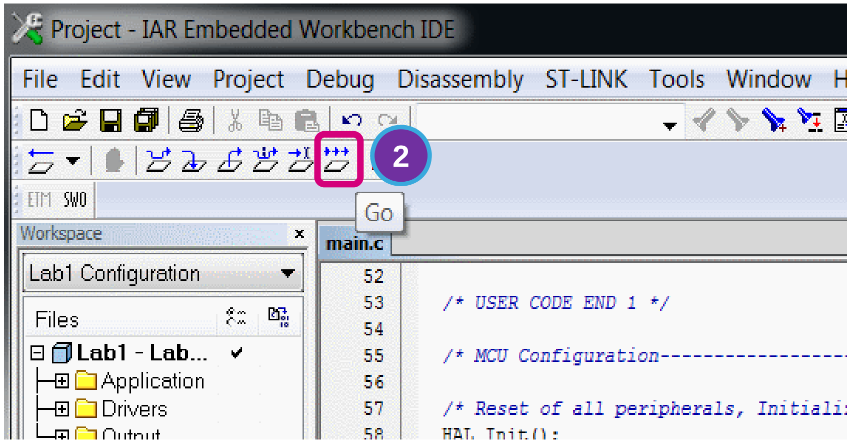
Load and Run

1. Click the GREEN ARROW to **Build** the Project, **Download** and start the **debugger**. (Ctrl + D)


Project - IAR Embedded Workbench IDE - ARM 7.70.1

```
79    /* USER CODE BEGIN 2 */
```

2. Click the triple-arrow **GO** button! (F5)


Project - IAR Embedded Workbench IDE

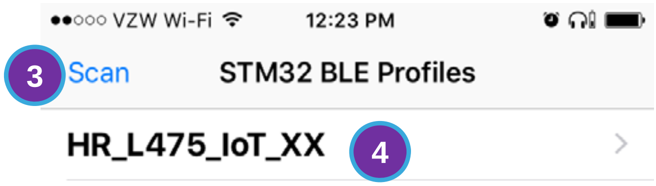
```
52
53    /* USER CODE END 1 */
54
55    /* MCU Configuration-----------------
56
57    /* Reset of all peripherals, Initiali:
58    HAL_Init();
```

Technology Tour 2017

Pair with **STM32 BLE Profiles** App

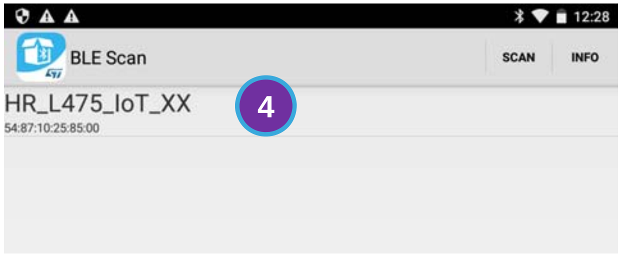
1. Make sure Bluetooth is active on your phone
2. Using your phone open the **STM32 BLE Profiles** app.
3. For iOS users click on **Scan**.
4. Identify your device using the Device name **HR_L475_IoT_XX** with **XX** is the number you have entered during the board configuration. Click on your device name,

iOS



HR_L475_IoT_XX

Android

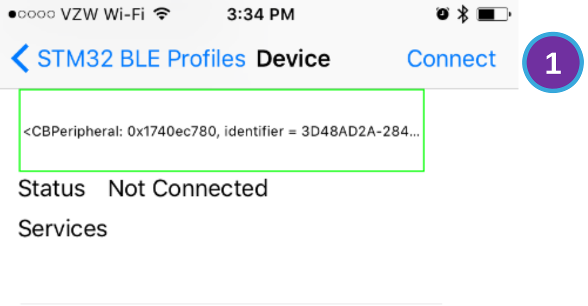


HR_L475_IoT_XX
54:87:10:25:85:00

Select the Heart Rate Profile

1. Click on **Connect** on the next screen (iOS)
2. Click on **Heart Rate** under **Services** (iOS) or **Profiles** (Android)
3. Click **Heart Rate Measurement** (iOS)

iOS



●○○○○ VZW Wi-Fi 3:34 PM

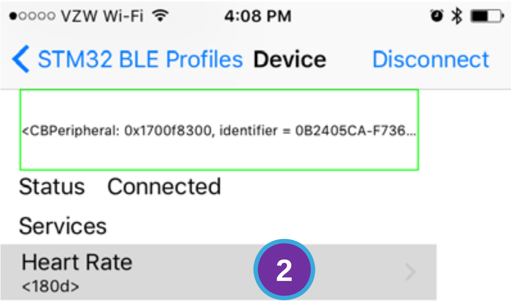
< STM32 BLE Profiles **Device** Connect

1

<CBPeripheral: 0x1740ec780, identifier = 3D48AD2A-284...

Status Not Connected

Services



●○○○○ VZW Wi-Fi 4:08 PM

< STM32 BLE Profiles **Device** Disconnect

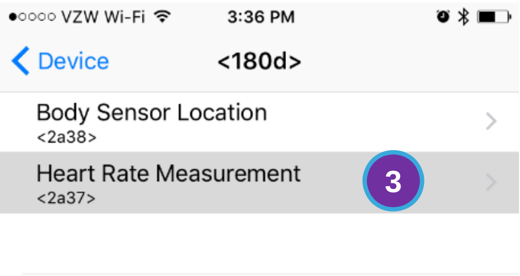
<CBPeripheral: 0x1700f8300, identifier = 0B2405CA-F736...

Status Connected

Services

Heart Rate
<180d>

2



●○○○○ VZW Wi-Fi 3:36 PM

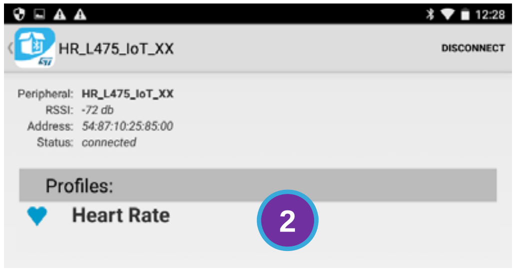
< Device <180d>

Body Sensor Location
<2a38>

Heart Rate Measurement
<2a37>

3

Android



12:28

HR_L475_IoT_XX DISCONNECT

Peripheral: HR_L475_IoT_XX
RSSI: -72 db
Address: 54:87:10:25:85:00
Status: connected

Profiles:

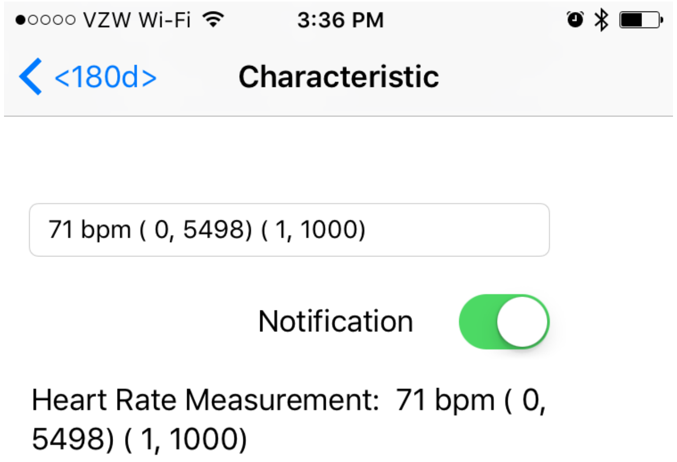
♥ Heart Rate

2

Display HR Data

- You should see the simulated heart rate.

iOS



●○○○○ VZW Wi-Fi 3:36 PM

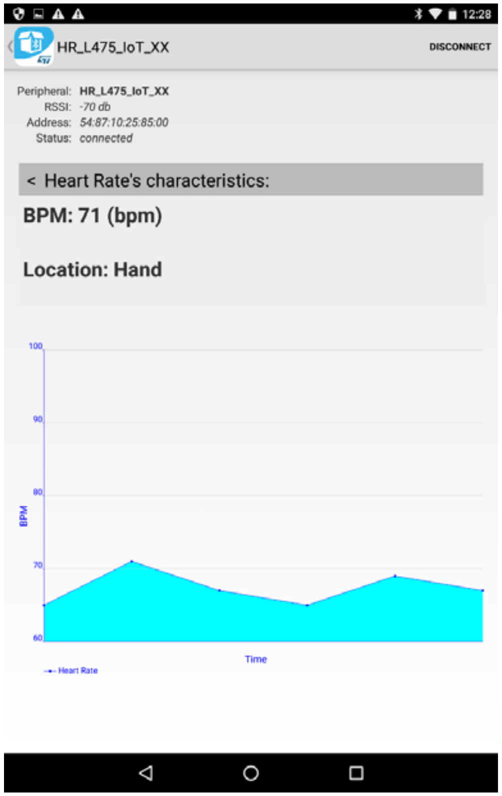
< <180d> Characteristic

71 bpm (0, 5498) (1, 1000)

Notification

Heart Rate Measurement: 71 bpm (0, 5498) (1, 1000)

Android



12:28

HR_L475_IoT_XX DISCONNECT

Peripheral: HR_L475_IoT_XX
RSSI: -70 db
Address: 54:87:10:25:85:00
Status: connected

< Heart Rate's characteristics:

BPM: 71 (bpm)

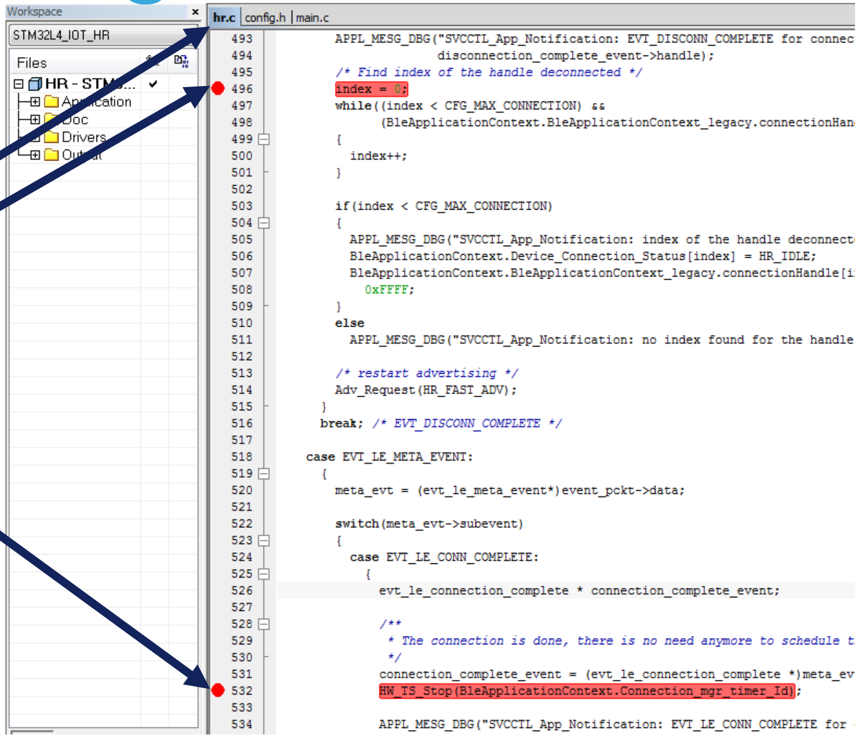
Location: Hand

Debug the firmware

- Now we are going to set beak point to stop the program execution when a client is connected/disconnected to the device.

1. Keep the program running and open the `hr.c` file.
2. Set a break point at line **496** and line **532**.

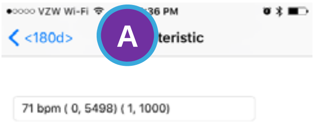
To set break point, left click on the left side of the line where you want to set it.



```
493          APPL_MESG_DBG("SVCCTL_App_Notification: EVT_DISCONN_COMPLETE for connec
494                        disconnection_complete_event->handle);
495          /* Find index of the handle deconnected */
496          index = 0;
497          while((index < CFG_MAX_CONNECTION) &&
498                (BleApplicationContext.BleApplicationContext_legacy.connectionHan
499          {
500            index++;
501          }
502
503          if(index < CFG_MAX_CONNECTION)
504          {
505            APPL_MESG_DBG("SVCCTL_App_Notification: index of the handle deconnect
506            BleApplicationContext.Device_Connection_Status[index] = HR_IDLE;
507            BleApplicationContext.BleApplicationContext_legacy.connectionHandle[i
508              0xFFFF;
509          }
510          else
511            APPL_MESG_DBG("SVCCTL_App_Notification: no index found for the handle
512
513          /* restart advertising */
514          Adv_Request(HR_FAST_ADV);
515        }
516        break; /* EVT_DISCONN_COMPLETE */
517
518      case EVT_LE_META_EVENT:
519        {
520          meta_evt = (evt_le_meta_event*)event_pckt->data;
521
522          switch(meta_evt->subevent)
523          {
524            case EVT_LE_CONN_COMPLETE:
525              {
526                evt_le_connection_complete * connection_complete_event;
527
528                /**
529                 * The connection is done, there is no need anymore to schedule t
530                 */
531                connection_complete_event = (evt_le_connection_complete *)meta_ev
532                HW_TS_Stop(BleApplicationContext.Connection_mgr_timer_Id);
533
534                APPL_MESG_DBG("SVCCTL_App_Notification: EVT_LE_CONN_COMPLETE for
```

Debug the firmware (Disconnect)

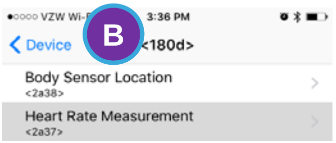
1. Now from your phone disconnect from the device:
   A. Click on **<180d>** (iOS)
   B. Click **< Device** (iOS)
   C. Click **Disconnect** (iOS and Android)



●○○○○ VZW Wi-Fi 36 PM

< <180d> teristic

71 bpm (0, 5498) (1, 1000)

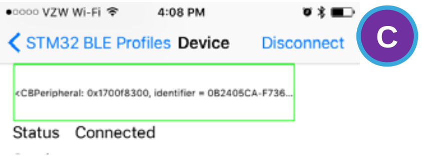


●○○○○ VZW Wi-F 3:36 PM

< Device <180d>

Body Sensor Location <2a38>

Heart Rate Measurement <2a37>

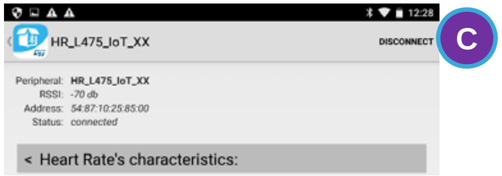


●○○○○ VZW Wi-Fi 4:08 PM

< STM32 BLE Profiles Device Disconnect

<CBPeripheral: 0x1700f8300, identifier = 0B2405CA-F736...

Status Connected



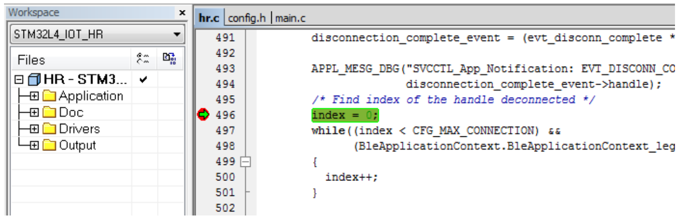
12:28

HR_L475_IoT_XX DISCONNECT

Peripheral: HR_L475_IoT_XX
RSSI: -70 db
Address: 54:87:10:25:85:00
Status: connected

< Heart Rate's characteristics:

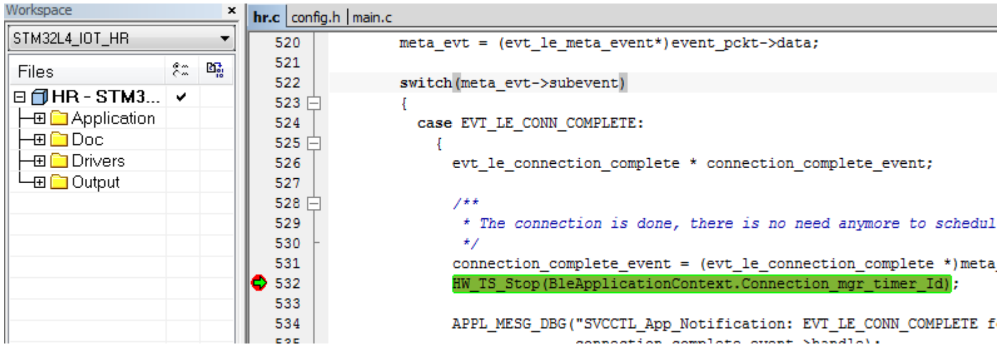
2. Once you disconnect the program will hit the break point at line **496** and stop execution.



```
491            disconnection_complete_event = (evt_disconn_complete *
492
493            APPL_MESG_DBG("SVCCTL_App_Notification: EVT_DISCONN_CO
494                          disconnection_complete_event->handle);
495            /* Find index of the handle deconnected */
496            index = 0;
497            while((index < CFG_MAX_CONNECTION) &&
498                  (BleApplicationContext.BleApplicationContext_leg
499            {
500              index++;
501            }
502
```

Debug the firmware (Connect)

1. Resume the execution by pressing the **Go** button (**F5**) on IAR.
2. Connect to the device from your phone.
3. Now the program will hit the break point at line **532**.



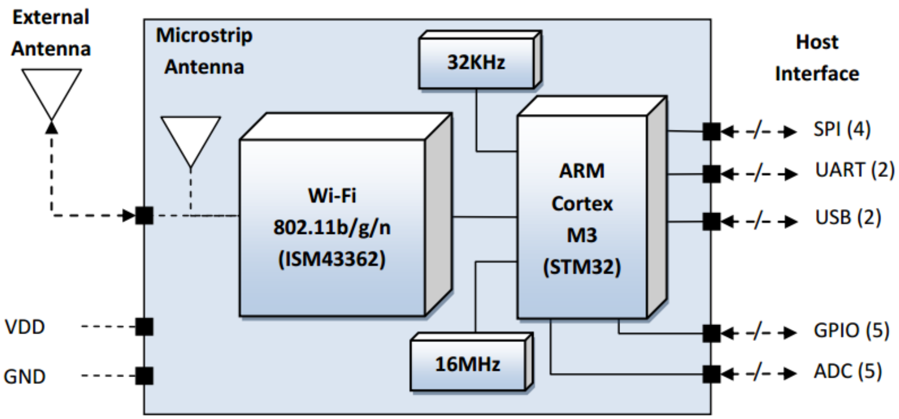
```
520         meta_evt = (evt_le_meta_event*)event_pckt->data;
521
522         switch(meta_evt->subevent)
523         {
524           case EVT_LE_CONN_COMPLETE:
525             {
526               evt_le_connection_complete * connection_complete_event;
527
528               /**
529                * The connection is done, there is no need anymore to schedul
530                */
531               connection_complete_event = (evt_le_connection_complete *)meta
532               HW_TS_Stop(BleApplicationContext.Connection_mgr_timer_Id);
533
534               APPL_MESG_DBG("SVCCTL_App_Notification: EVT_LE_CONN_COMPLETE f
```

Wi-Fi Module Overview

ISM43362-M3G-L44-E/U

- The ISM43362-M3G-L44-E/U is an embedded 2.4 GHz Wi-Fi module from Inventek. The Wi-Fi module hardware consists of a Broadcom BCM43362, an integrated antenna or optional external antenna, and a STM32F205 host processor that has a standard USB, SPI or UART interface capability.

- The Wi-Fi has an integrated TCP/IP stack that only requires a simple AT command set to establish connectivity for your wireless product.

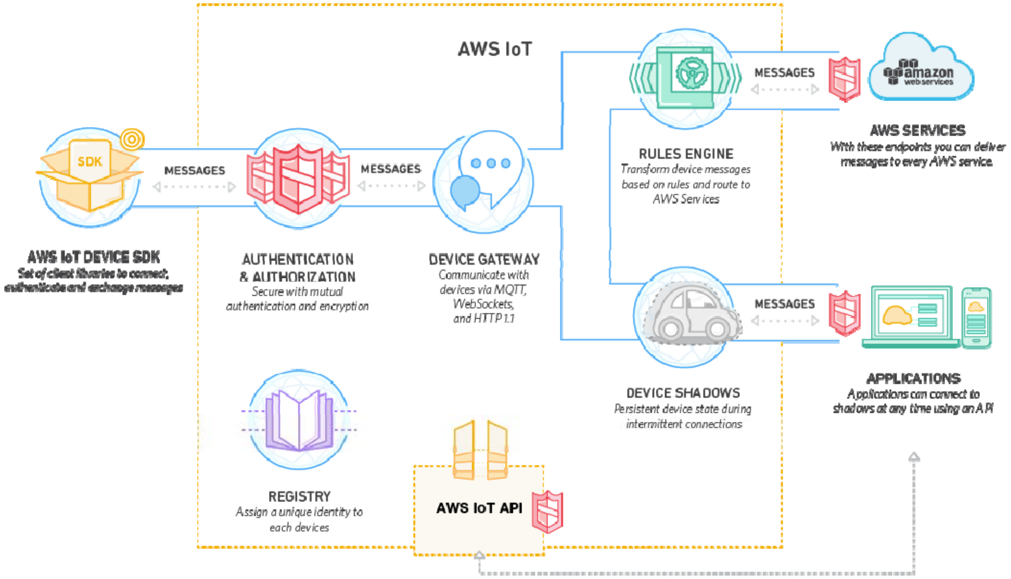


AWS IoT Overview

What is AWS IoT

- The Amazon AWS IoT service enables secure, bidirectional communication between IoT devices and the cloud over MQTT, HTTP and WebSockets.

- IoT devices are authenticated using AWS IoT service-provided X.509 certificates. Once a certificate is provisioned and activated it can be installed on a device. The device will then use that certificate to send all requests to AWS MQTT.


AWS IoT
SDK
MESSAGES
MESSAGES
MESSAGES
amazon web services
AWS SERVICES
With these endpoints you can deliver messages to every AWS service
AWS IoT DEVICE SDK
Set of client libraries to connect, authenticate and exchange messages
AUTHENTICATION & AUTHORIZATION
Secure with mutual authentication and encryption
DEVICE GATEWAY
Communicate with devices via MQTT, WebSockets, and HTTP 1.1
RULES ENGINE
Transform device messages based on rules and route to AWS Services
MESSAGES
DEVICE SHADOWS
Persistent device state during intermittent connections
APPLICATIONS
Applications can connect to shadows at any time using an API
REGISTRY
Assign a unique identity to each devices
AWS IoT API