STM32-Murata LoRaWAN Pre-Training TESA Top Gun Rally 2017 KMUTNB

Nov 2017

Pisit Rojthongkham

life.augmented

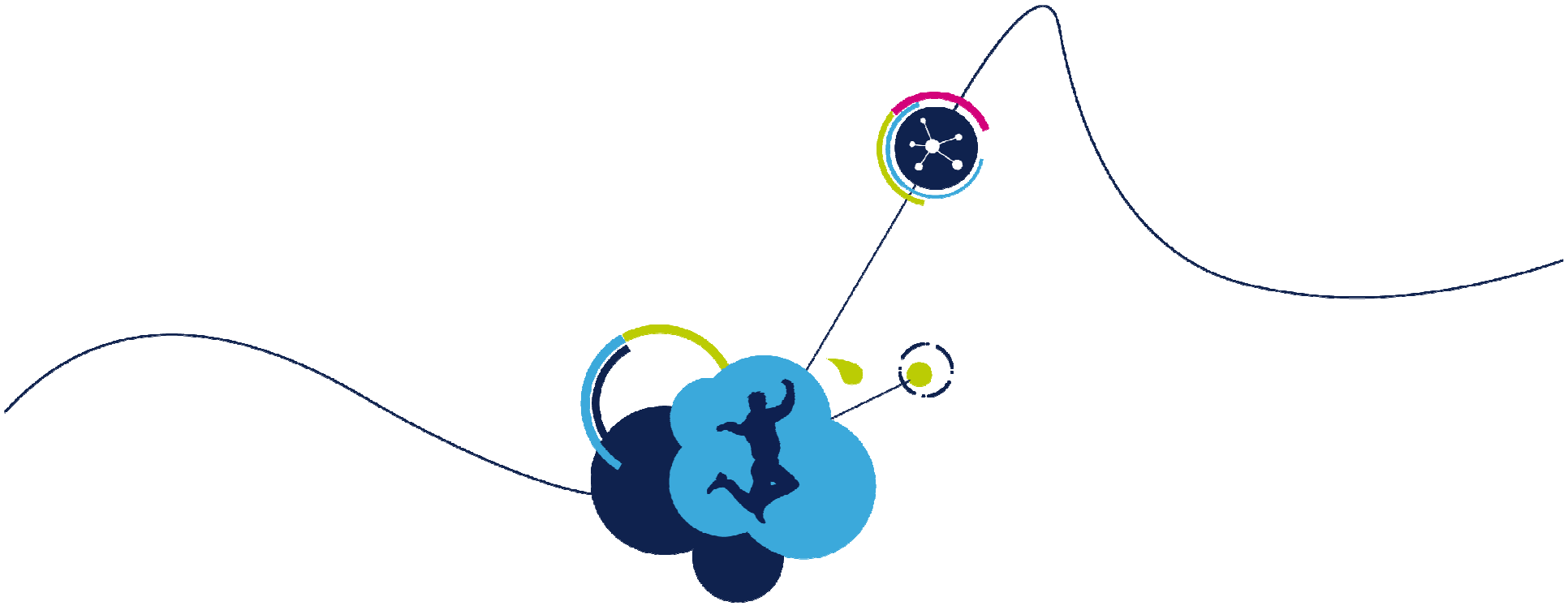
System Requirement

- Laptop
  - Administrative privileges (Admin rights) needed for driver/software install and for compiling codes.
- Windows Operating System
  - XP/Vista/Win7/Win8
- **Please bring the following:**
  - 1x USB A to mini-B cable

Installations before the workshop: Download the following files

Download the following software from the links provided. Refer to the installation procedures in the next pages.

- Software Development Environment:
  - KEIL MDK-ARM for STM32F0 and STM32L0: www2.keil.com/stmicroelectronics-stm32/mdk
    - **Important:** Requires internet access to activate the free license.
  - STMicroelectronics STM32L0 Series Device Support and Examples (Device Family Pack) from www.keil.com/dd2/Pack/
- ST-Link USB driver: www.st.com/stlinkv2
- STM32 ST-Link Utility: www.st.com/stlinkv2
- Terminal emulator software: Teraterm https://ttssh2.osdn.jp/
- LoRaWAN firmware library I-CUBE-LRWAN
  - www.st.com/i-cube-lrwan



Software Installation Steps

life.augmented

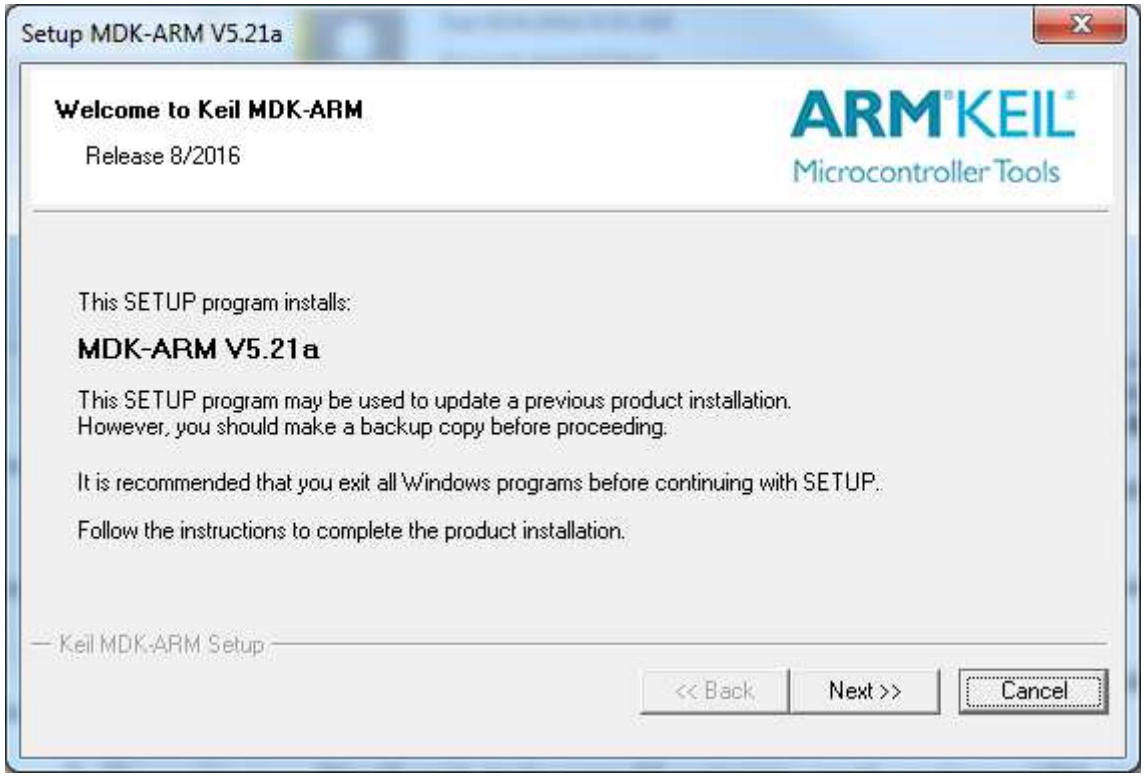
Step 1: KEIL MDK-ARM Installation

- We will be using KEIL MDK-ARM v5 in this session.
- **MDK for STMicroelectronics STM32F0 and STM32L0**
  - Keil MDK-ARM for STM32F0 and STM32L0 provides software developers working with STM32 devices based on the ARM Cortex-M0 and ARM Cortex-M0+ cores with a **FREE-TO-USE** professional tool suite.
    - Product Serial Number (PSN) to activate the MDK for STM32F0 and STM32L0:
      - ***U1E21-CM9GY-L3G4L***
- Keil MDK is the most comprehensive software development system for ARM processor-based microcontroller applications. Keil MDK includes the ARM C/C++ Compiler, the CMSIS-RTOS RTX Kernel, and the µVision IDE/Debugger.

From www2.keil.com/stmicroelectronics-stm32/mdk

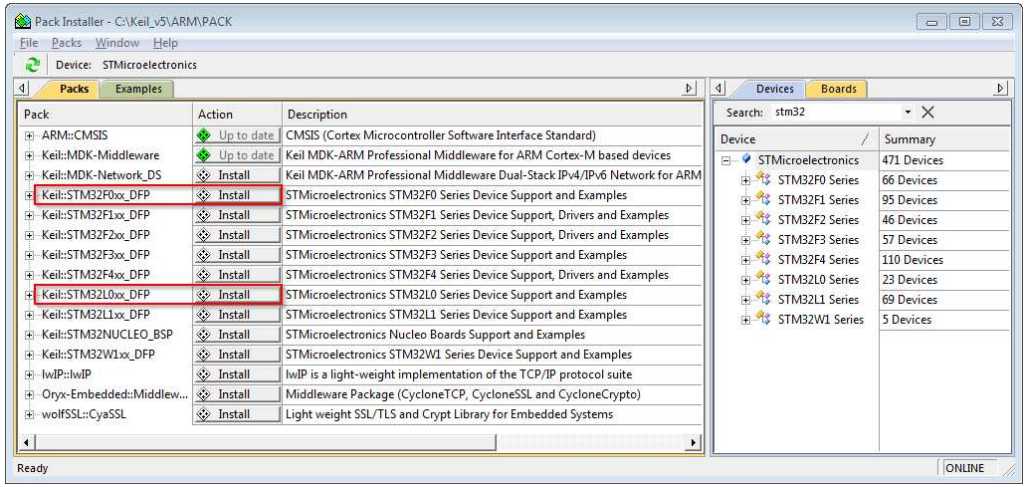
Step 1a: KEIL MDK-ARM Installation

- Download Keil MDK-ARM www2.keil.com/stmicroelectronics-stm32/mdk
  - Take note of the **Product Serial Number (PSN)** as you will use this later to activate the License.
- Double click on **MDK5xx.EXE** to begin installation.



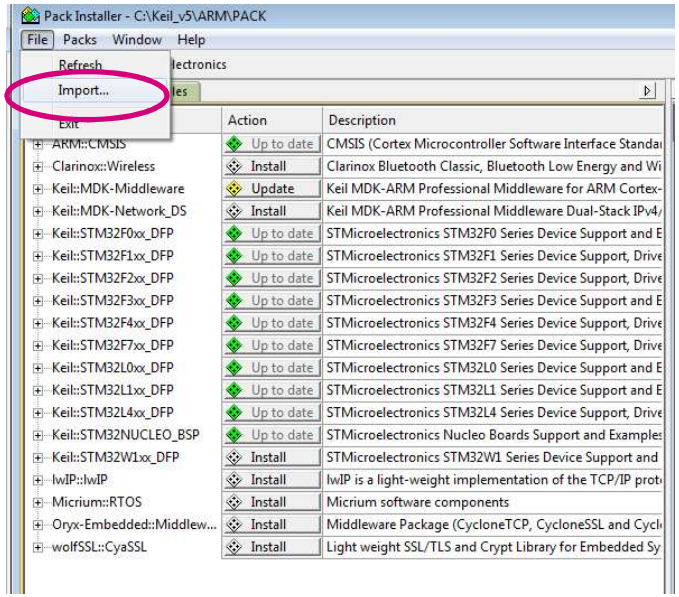
Step 1b: STM32L0 Device Family Pack (DFP) Installation

- **Device Family Pack (DFP)** contains CMSIS system/startup files, drivers, and flash algorithms for a microcontroller device family.
- The STM32L0 DFP is required to be able to compile STM32L0 Keil projects.
- After installation, KEIL Pack Installer should automatically launch. Click Install to download and install the STM32L0 Device Family Packs supplied by KEIL.
  - If not, launch it manually by clicking ***Project menu>Manage>Pack Installer***



Alternatively,

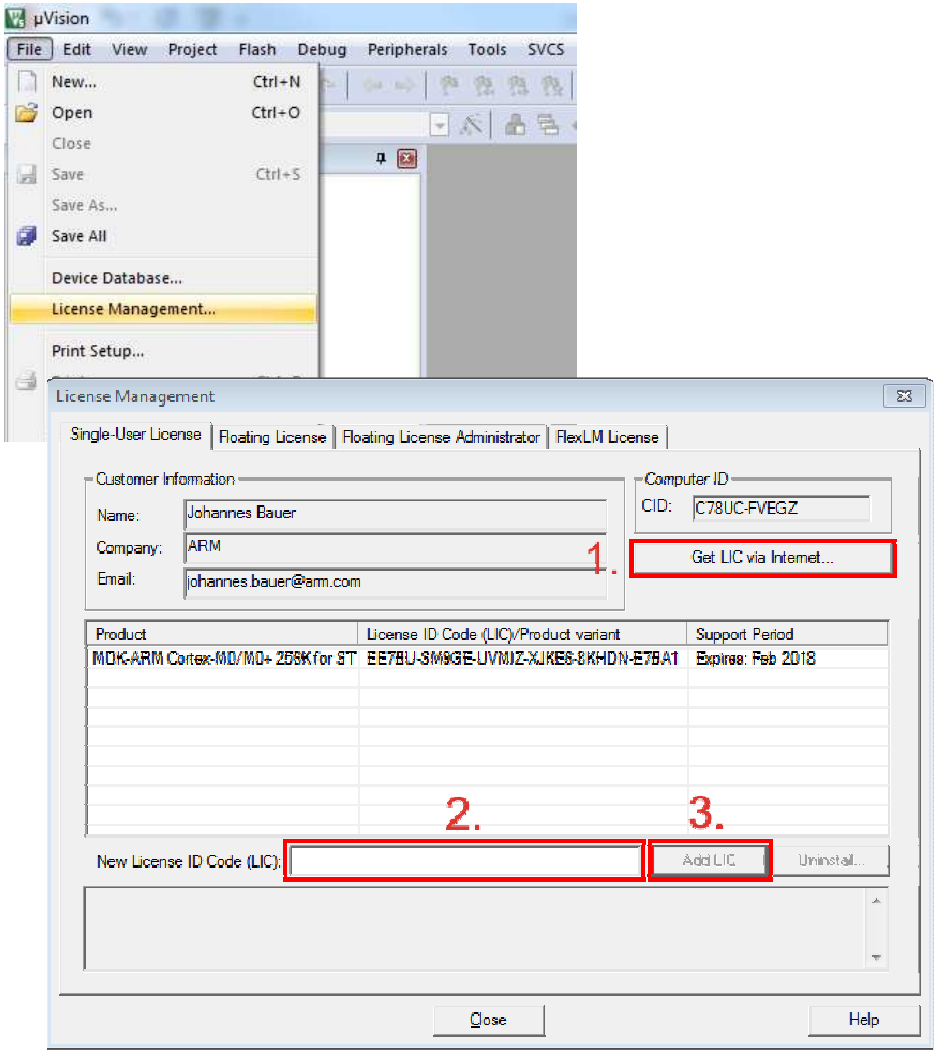
- An alternative way to install the STM32L0 DFP, is to manually import it.
- Download the STM32L0 DFP www.keil.com/dd2/Pack/
- Import the downloaded file (Keil.STM32L0xx_DFP.1.6.0.zip) through *File menu>Import*



life.augmented

Step 1c: KEIL MDK-ARM License Activation

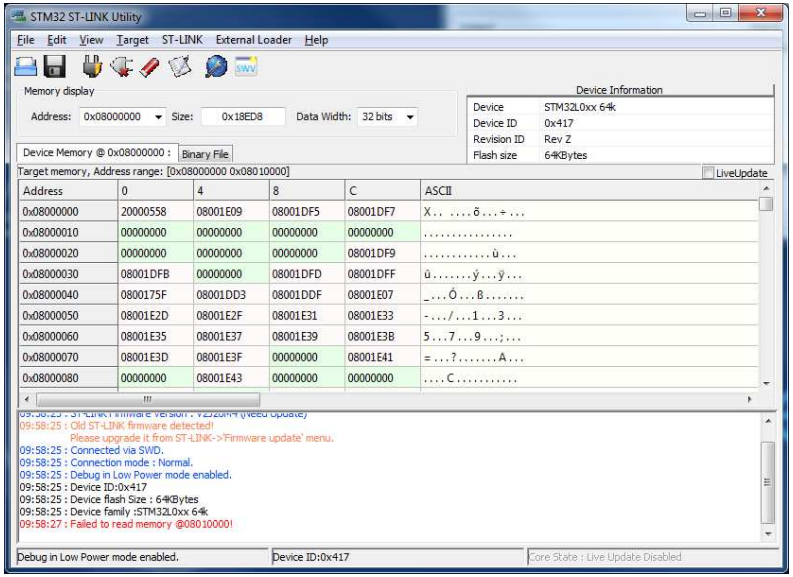
- Login with an account that has administration rights.
- Right-click the µVision icon and select **Run as Administrator...** from the context menu.
- Open the dialog **File — License Management...** and select the **Single-User License** tab.
- Click the button **Get LIC via Internet...**, then click the button **OK** to register the product. This action opens the License Management page on the Keil web site.
- Enter the **Product Serial Number U1E21-CM9GY-L3G4L** along with your contact information and click the button **Submit**. An e-mail is sent back with the **License ID Code (LIC)** within a few minutes.
- To activate the Software Product, enter the **LIC** in the field **New License ID Code (LIC)** of the dialog **License Management...** and click **Add LIC**.



Step 2: ST-Link USB driver And ST-Link Utility

- Download the ST-Link USB drivers from www.st.com/stlinkv2 and install the drivers.

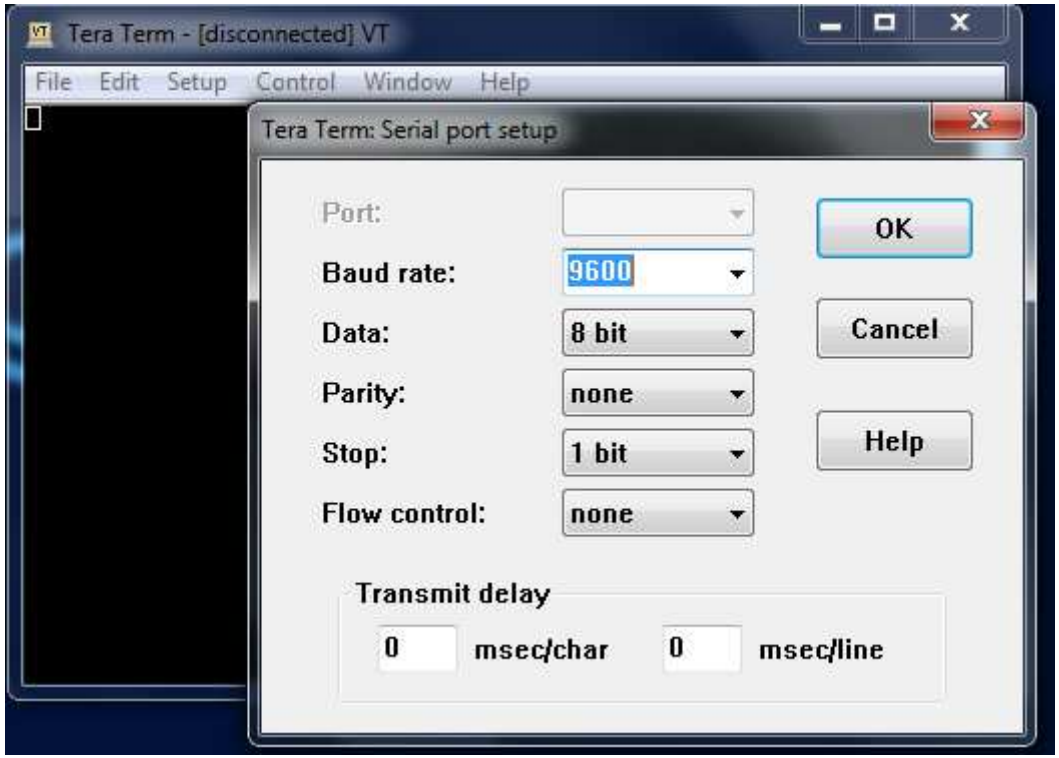
- Also download the STM32 ST-Link Utility from www.st.com/stlinkv2 and install the utility.
  - The STM32 ST-Link Utility is an STM32 programming software using the ST-Link programmer/debugger tool.
  - Besides programming capability, the tool can also be used to verify the ST-Link driver installation and connection to the target MCU.



life.augmented

Step 3: Teraterm

- Tera Term is a free software terminal emulator (communication program) which supports multiple communication including Serial port connections which will be used during this workshop.
- Download Teraterm from https://ttssh2.osdn.jp/
- Run the teraterm-4.92.exe and follow the installation wizard.

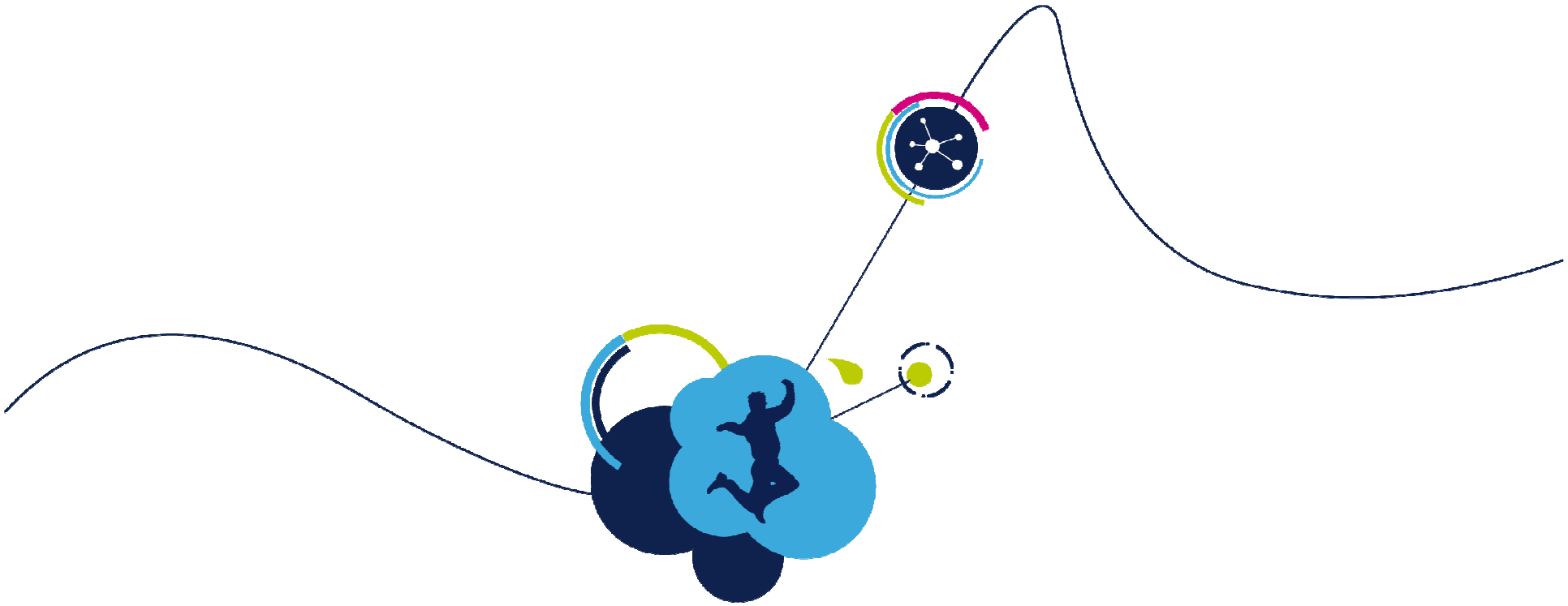


Hands-on Outline

- STM32L0 Overview
- STM32LoRa Discovery kit Overview
- LoRa Device Firmware Library Overview
- LoRa Sensor Device Setup and Reconfiguration
- LoRaWAN Gateway setup
- Actility Network Setup
- myDevices Cayenne Application Server Setup

Agenda

- **09:30** Session Start – Welcome
- **09:40 – 09:55** STM32 Overview (15mins)
- **09:55 – 10:10** STM32L0 Core and System Architecture Overview (15mins)
- **10:10 – 10:25** STM32LoRa discovery kit Overview (15mins)
- **10:25 – 10:45** Break
- **10:45 – 11:00** LoRa Device Firmware Library Overview (15mins)
- **11:00 – 11:15** LoRa Sensor Device Setup and Reconfiguration (15mins)
- **11:15 – 12:00** Hands-on GPIO (45mins)



STM32L0 Core and System Overview

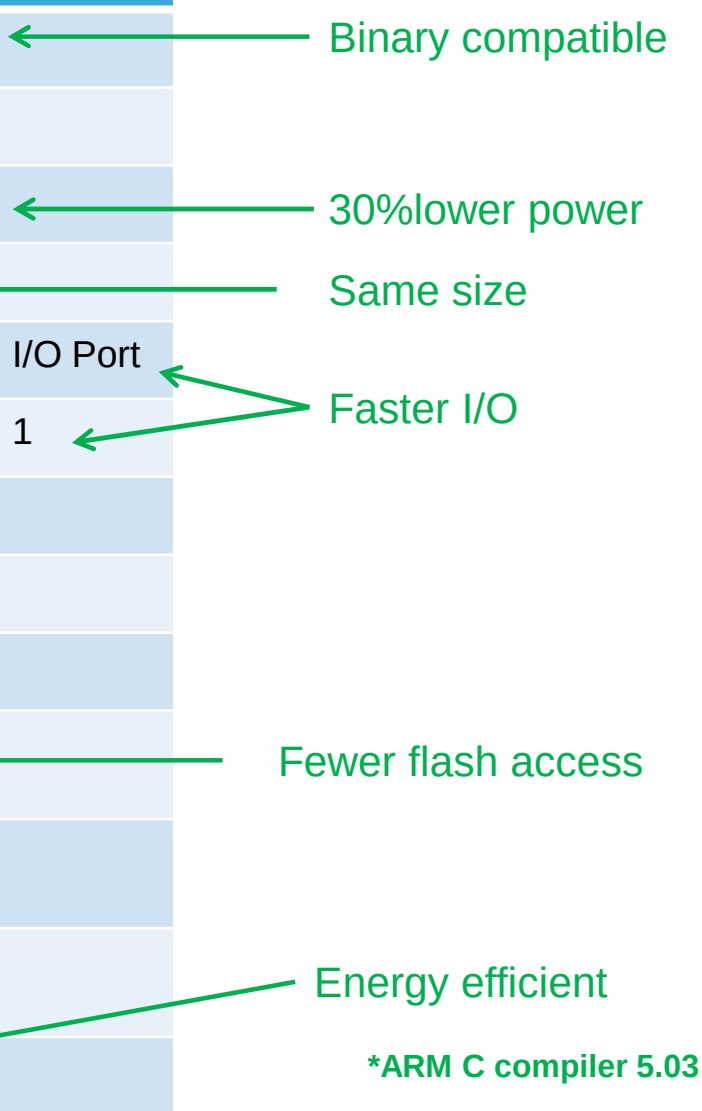
life.augmented

Cortex-M0/0+/3 feature set comparison

	Cortex-M0	Cortex-M0+	Cortex-M3
Architecture Version	V6M	V6M	v7M
Instruction set architecture	Thumb, Thumb-2 (System Instructions only)	Thumb, Thumb-2 (System Instructions only)	Thumb + Thumb-2
DMIPS/MHz	0.84/0.9	**0.95**	1.25
Bus interfaces	1	**1+ I/0 port**	3
Integrated NVIC	Yes	Yes	Yes
Number interrupts	1-32 + NMI	**1-32 + NMI**	1-240 + NMI
Re-locatable vector table	No	**Yes**	Yes
Interrupt priorities	4	4	8-256
Breakpoints, Watchpoints	4/2/0, 2/1/0	**4**/2/0, **2**/1/0	8/4/0, 2/1/0
Memory Protection Unit (MPU)	No	**Yes (Option)**	Yes (Option)
Integrated trace option (ETM)	No	**MTB (Option)**	ETM (Option)
Privilege/unprivileged	No	**Yes (Option)**	Yes (Option)
Single Cycle Multiply	Yes (Option)	**Yes (Option)**	Yes
Hardware Divide	No	No	Yes
Bit banding support	No	No	Yes
Bus protocol	AHB Lite	AHB Lite, I/O	AHB Lite, APB
CMSIS Support	Yes	Yes	Yes

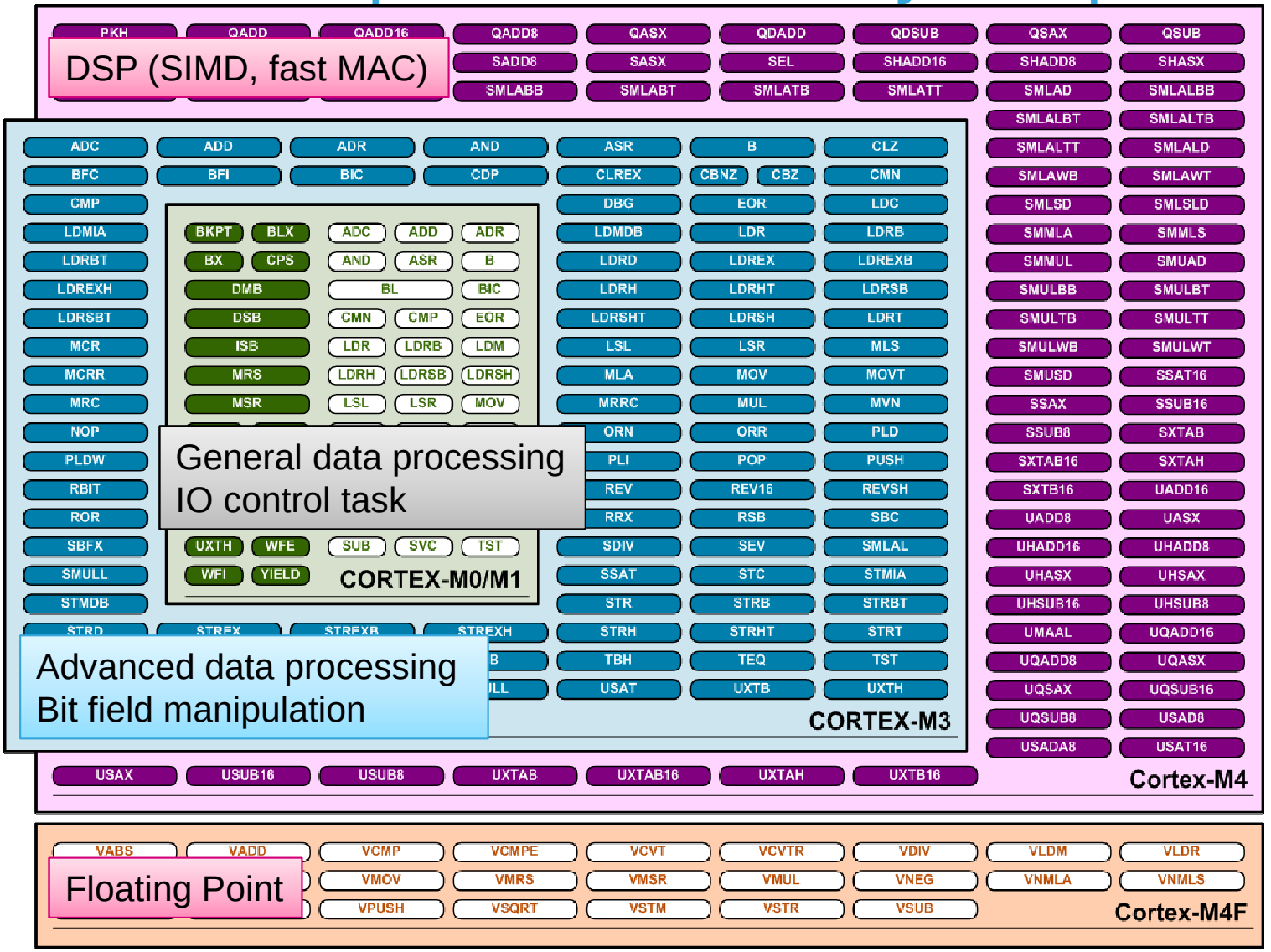
CORE M0 and M0+

Feature	Cortex-M0	Cortex-M0+	
Architecture	ARMv6-M	ARMv6-M	
Pipeline	3-stage	2-stage	
Dynamic Power (180nm*)	73 µW/MHz	52 µW/MHz	
Area (gate count)	12K	12K	
Bus interface	AHB-lite	AHB-Lite	I/O Port
Data access (cycles)	2	2	1
Privileged/Unprivileged	Privileged only	Two levels	
MPU option	No	Yes	
Re-locatable vector table	No	Yes	
Instruction fetch activity* (Relative for Dhrystone)	1	0.85	
Performance efficiency* (DMIPS/MHz)	0.84/0.9	0.93	
Performance efficiency* (coremark/MHz)	2.33	2.42	
Energy efficiency* (DMIPS/mW)	11.5	17.9	

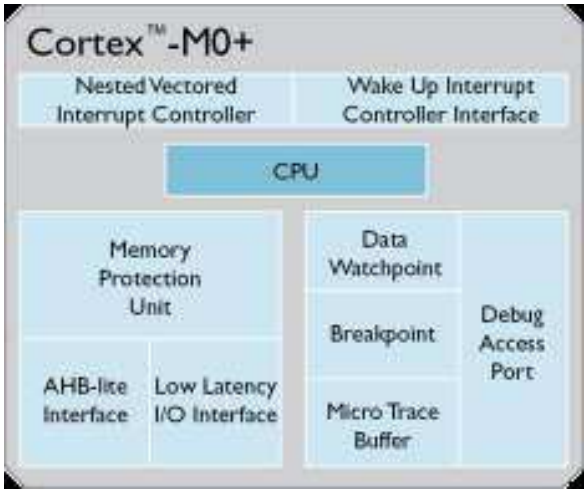


***ARM C compiler 5.03**

Cortex-M processors binary compatible

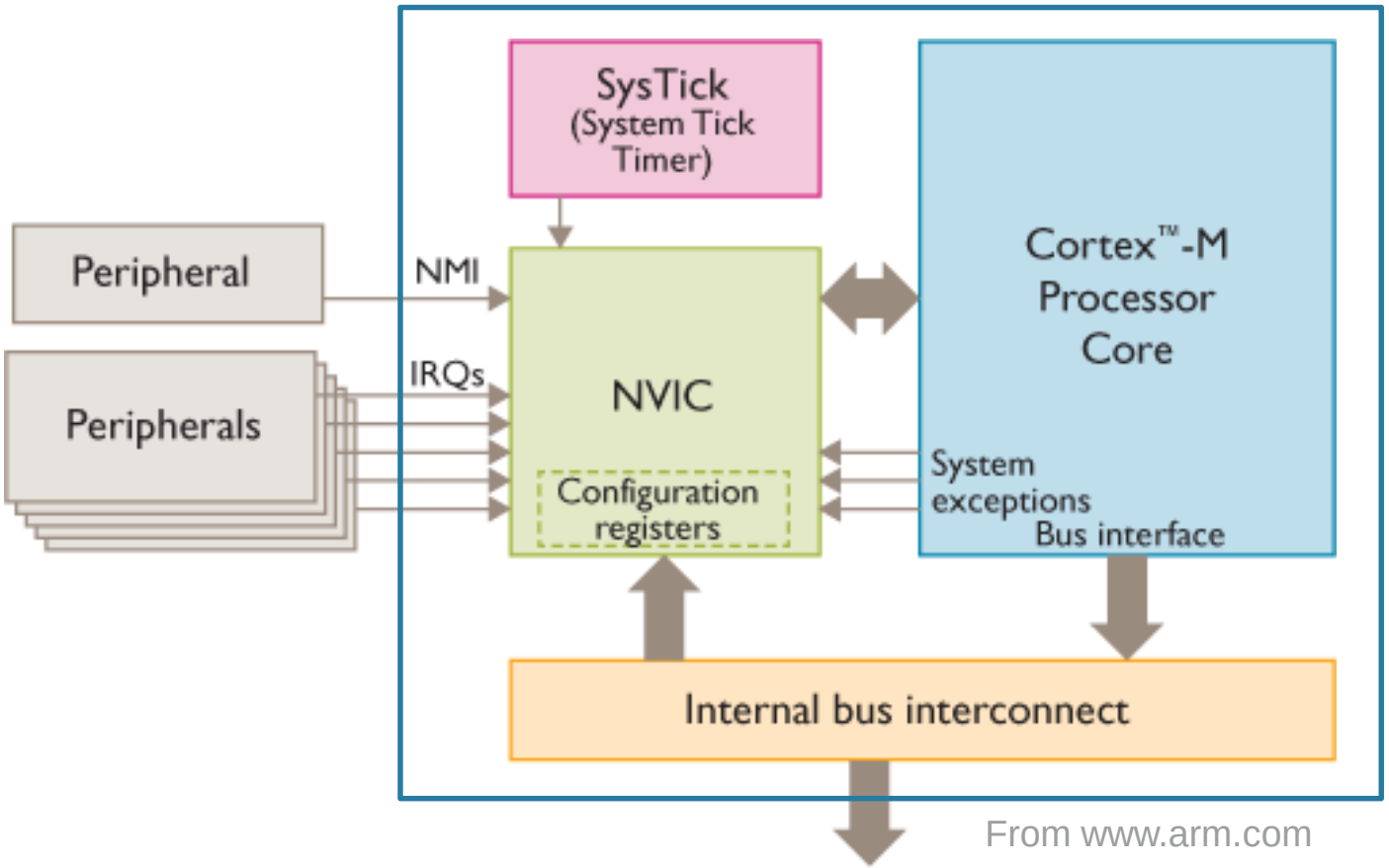


Cortex-M0+ Processor Microarchitecture



- ARMv6-M Architecture (Von Neumann)
  - Very low Processor consumption
  - Outstanding result of 2.15 CoreMark/MHz
  - Thumb/Thumb-2 subset Technology
  - Integrated configurable NVIC
  - Compatible with Cortex-M3
- Microarchitecture
  - 2-stage pipeline
  - 1x AHB-Lite Bus Interface, Fast I/O Interface for single cycle access
- Configurable for ultra low power
  - Deep Sleep Mode, Opt. Wakeup Interrupt Controller
- Flexible configurations for wider applicability
  - Configurable Interrupt Controller (1-32 Interrupts and Priorities)
  - Re-locatable vector table
  - Memory Protection Unit
  - SWD debug interface, Micro Trace Buffer for faster debug

Cortex-M NVIC (Nested Vector Interrupt Controller)



From www.arm.com

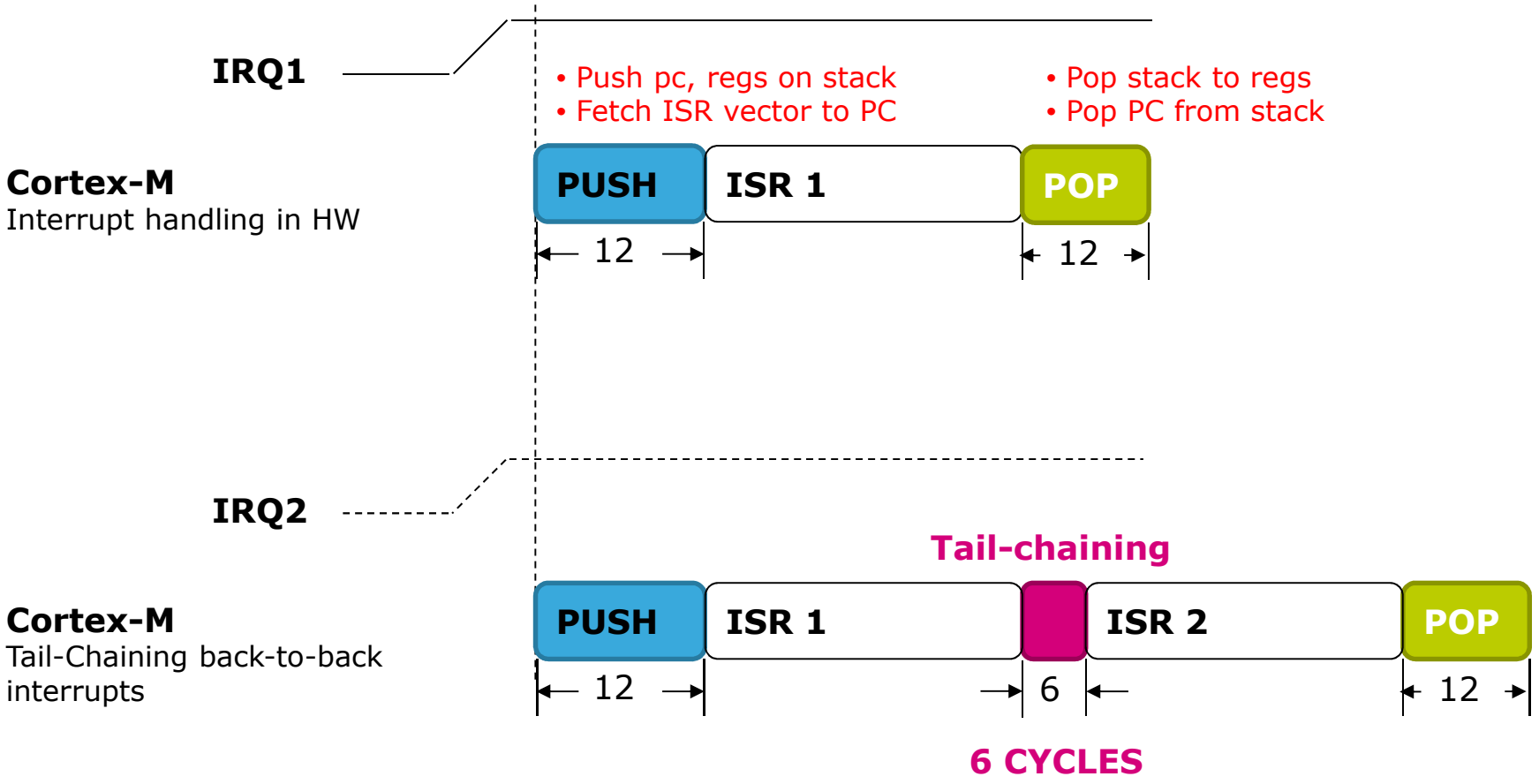
Cortex-M0+ Exception Types

No.	Exception Type	Priority	Type of Priority	Descriptions
1	Reset	-3 (Highest)	fixed	Reset
2	NMI	-2	fixed	Non-Maskable Interrupt
3	Hard Fault	-1	fixed	Default fault if other hander not implemented
4	MemManage Fault	0	settable	MPU violation or access to illegal locations
5	Bus Fault	1	settable	Fault if AHB interface receives error
6	Usage Fault	2	settable	Exceptions due to program errors
7-10	Reserved	N.A.	N.A.	
11	SVCall	3	settable	System Service call
12	Debug Monitor	4	settable	Break points, watch points, external debug
13	Reserved	N.A.	N.A.	
14	PendSV	5	settable	Pendable request for System Device
15	SYSTICK	6	settable	System Tick Timer
16	Interrupt #0	7	settable	External Interrupt #0
.....	.....	.....	settable	.....
47	Interrupt#31	38	settable	External Interrupt #31

The NVIC supports up to 32 dynamically re-prioritizable interrupts each with 4 levels of priority.

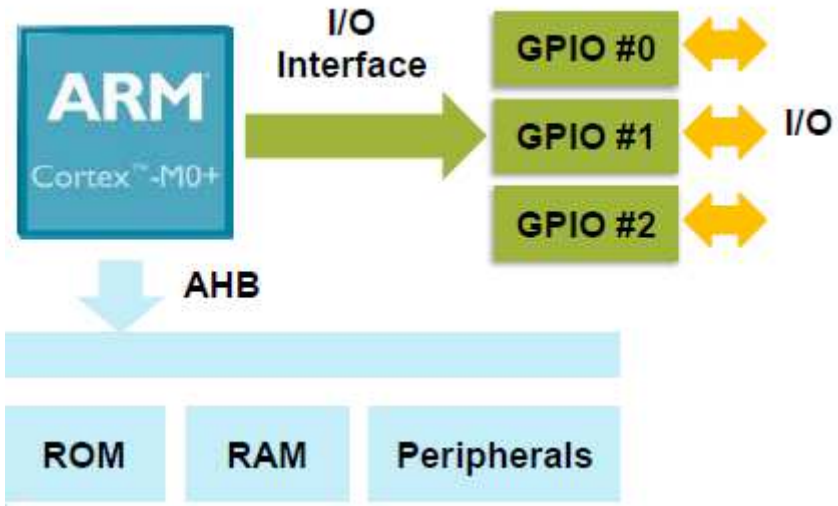
Cortex-M NVIC (Nested Vector Interrupt Controller)

- Interrupts are Fast AND Deterministic



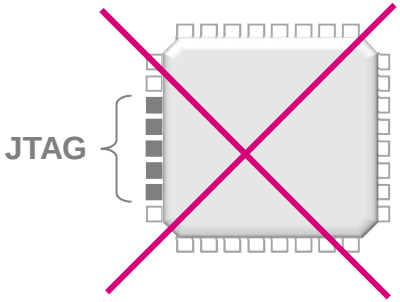
Cortex-M0+ AHB and I/O Interface

- AHB – 32-bit System bus
  - Program fetches
  - Data accesses
  - Peripheral accesses
  - Pipelined operations
  - Uses two cycles
    - (Address and Data phases)
- I/O Interface – 32-bit generic bus
  - Single cycle access to I/O peripherals
  - Ideal for I/O intensive applications
    - Faster operations
    - Better energy efficiency
  - Optional

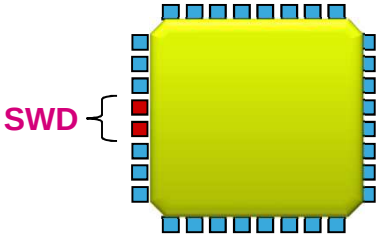


Debug Capabilities

- Full JTAG and also Serial Wire Debug interface



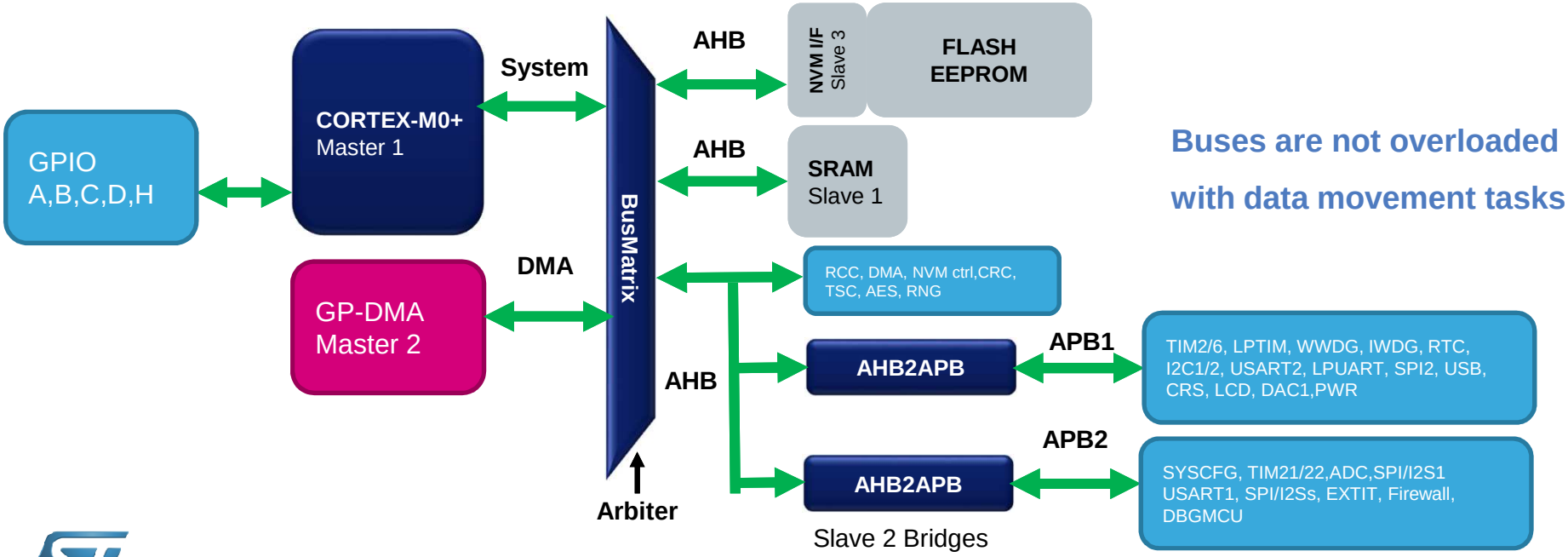
Supported in CM3/CM4



- Breakpoint and Watchpoint units
  - 4 hardware breakpoints (besides BKPT instruction)
  - 2 hardware watchpoints

STM32L0 System Architecture

- Multiply possibilities of bus accesses to SRAM, NVM, Peripherals, DMA
  - BusMatrix added to allow parallel access
- Efficient DMA and Rapid data flow
  - Direct path to SRAM through arbiter, guarantees alternating access
  - Von Neumann + BusMatrix allows Flash execution in parallel with DMA transfer
- Increase Peripherals Speed for better performance
  - Advanced Peripheral bus (APB) architecture up to 32MHz
  - Allows to optimize use of peripherals (16MBits/s SPI, 4Mbps USART, 32MHz GP-Timer, 16MHz toggling I/Os)

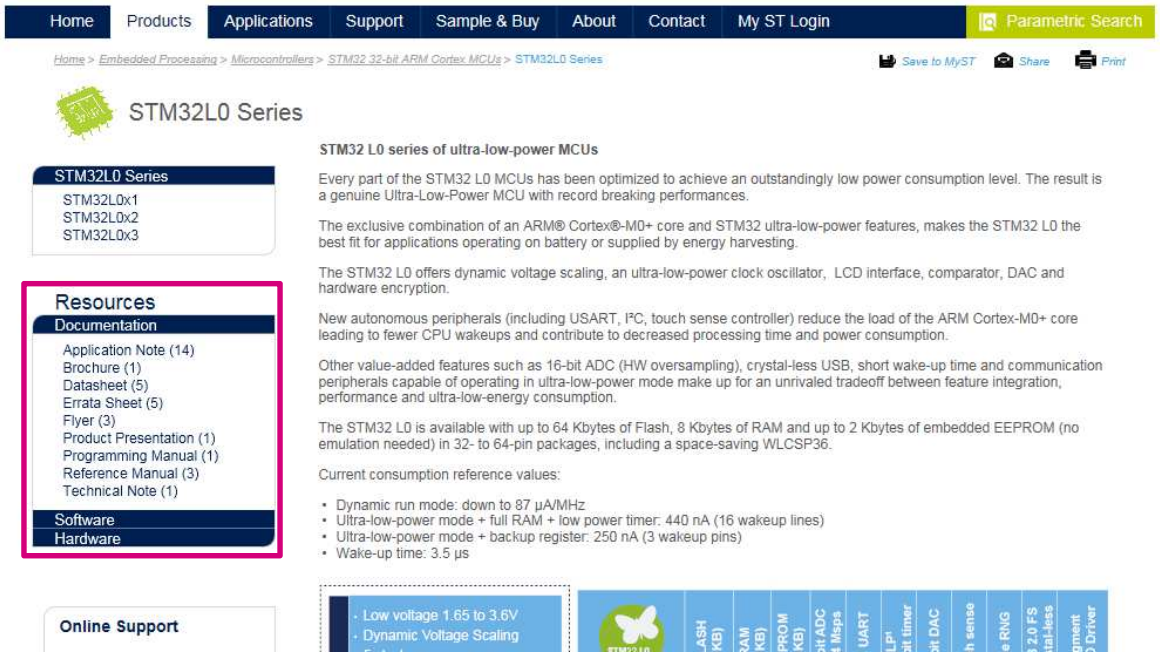


STM32L07x block diagram

- Key features
  - ARM Cortex-M0+ at 32MHz
    - Single-cycle I/O access
    - Single-cycle multiplier (MUL)
    - 0.95 DMIPS/MHz
  - 1.71V to 3.6V, 32MHz full functional
  - Digital down to 1.65V
  - -40°C to +125°C temperature range
  - ADC with build-in HW oversampling
    - Down to 1.65V
  - Flash + Ram code sector lock
  - USB 2,0 FS certified
    - Build-in 48MHz oscillator
    - Battery Charger Detection
    - Link Power Management
  - Independent clock domain
    - I2C, USART/UART
    - USB
  - **5x** timers
    - 1x 16-bit (4ch)
    - 3x 16-bit(2ch)
    - 1x 16-bit LP[1] available in stop

STM32L0 Documents

- Related documents downloadable from www.st.com/stm32l0
- STM32L0 Reference Manual
  - This reference manual targets application developers. It provides complete information on how to use the STM32L0xx microcontroller memory and peripherals.
  - Includes Peripheral Functional Description and Register Description.
- STM32L0 Series Cortex®-M0+ programming manual
  - Provides a full description of the STM32L0 Cortex-M0+ processor programming model, instruction set and core peripherals.
- Datasheet
  - Device overview
  - Functional overview
  - Pin descriptions
  - Memory mapping
  - Electrical characteristics

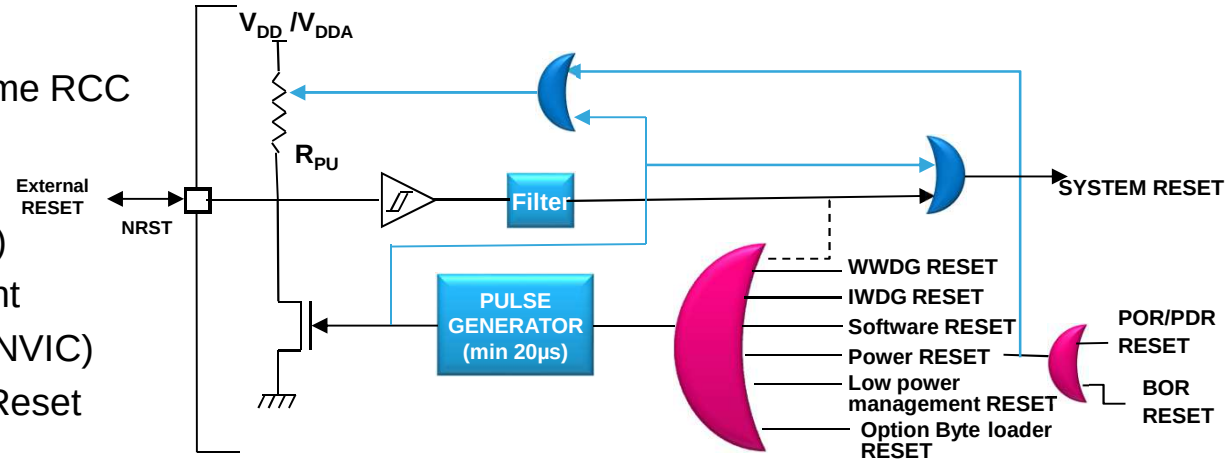


life.augmented

Reset and Clock Control

RESET Sources

- System RESET
  - Resets all registers except some RCC registers and RTC
  - Sources
    - NRST pin (External Reset)
    - WWDG/IWDG end of count
    - A software reset (through NVIC)
    - Low power management Reset
    - Option byte loader Reset
    - When exiting STANDBY mode
- Power RESET
  - Resets all registers
  - Sources
    - Power On/Power down Reset (POR/PDR)
    - BOR
- **Enhancement** (versus L1)
  - Pull-up is disabled when pin forced low
  - Internal reset is not delayed by reset pad



- RTC domain RESET
  - Resets all RTC domain: RTC registers + RTC Backup Registers + RCC CSR register
  - Sources
    - Setting RTCRST bit in RCC CSR register
    - POWER Reset

*Tamper detection can reset RTC backup register only

On Chip Oscillators: overview

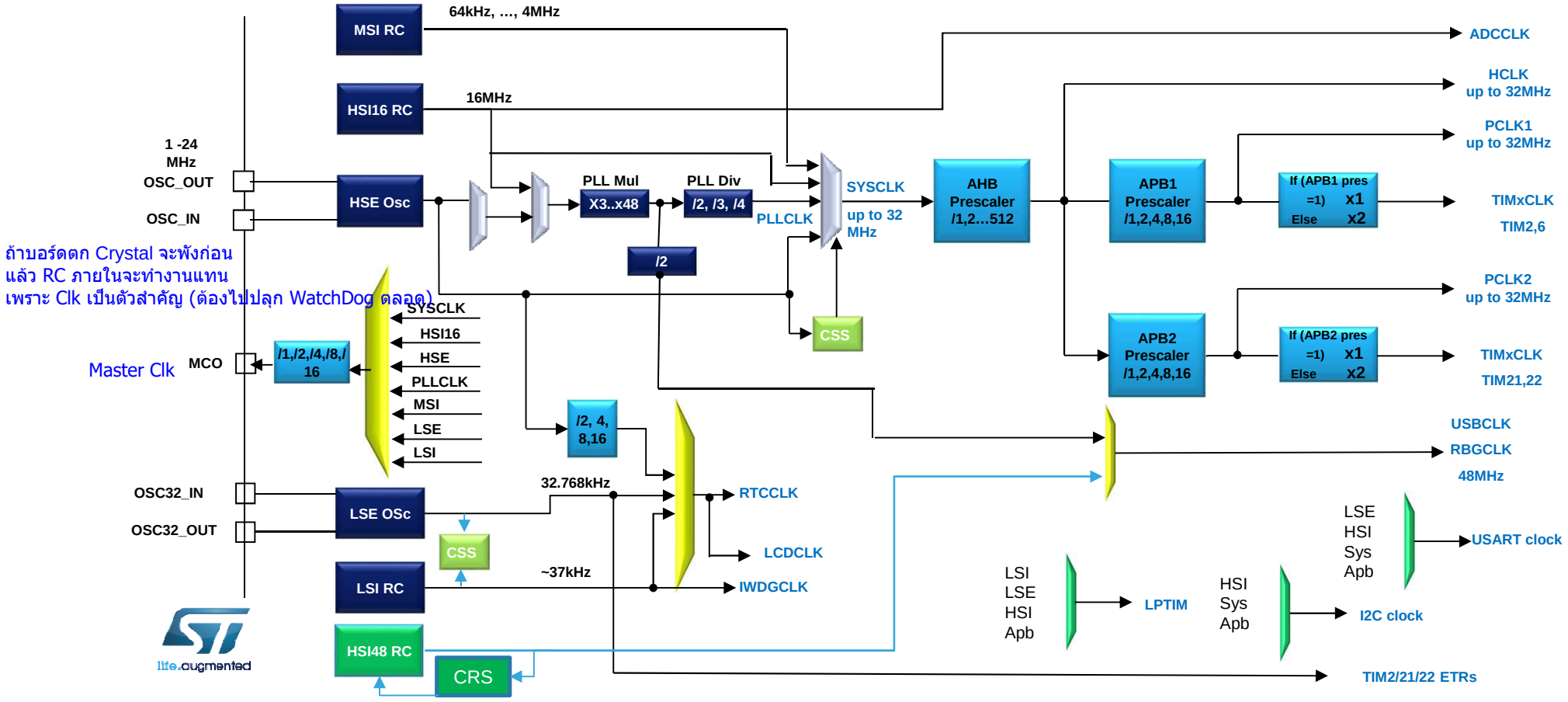
- Multiple clock sources for full flexibility in RUN/Low Power modes
  - HSE (High Speed External oscillator):
    - 1MHz to 24MHz oscillator or up to 32MHz external clock
  - HSI16 (High Speed Internal RC):
    - 16MHz
  - HSI48 (High Speed Internal RC):
    - 48MHz (for USB and RNG)
  - MSI(Multispeed internal RC oscillator): 7 possible frequencies:
    - 65 kHz, 131 kHz, 262 kHz, 524 kHz, 1.05 MHz, 2.1 MHz, 4.2 MHz
  - LSI (Low Speed Internal RC):
    - 37kHz Low consumption Low precision
  - LSE (Low Speed External oscillator):
    - 32.768kHz oscillator or external clock (1 kHz to 1MHz)

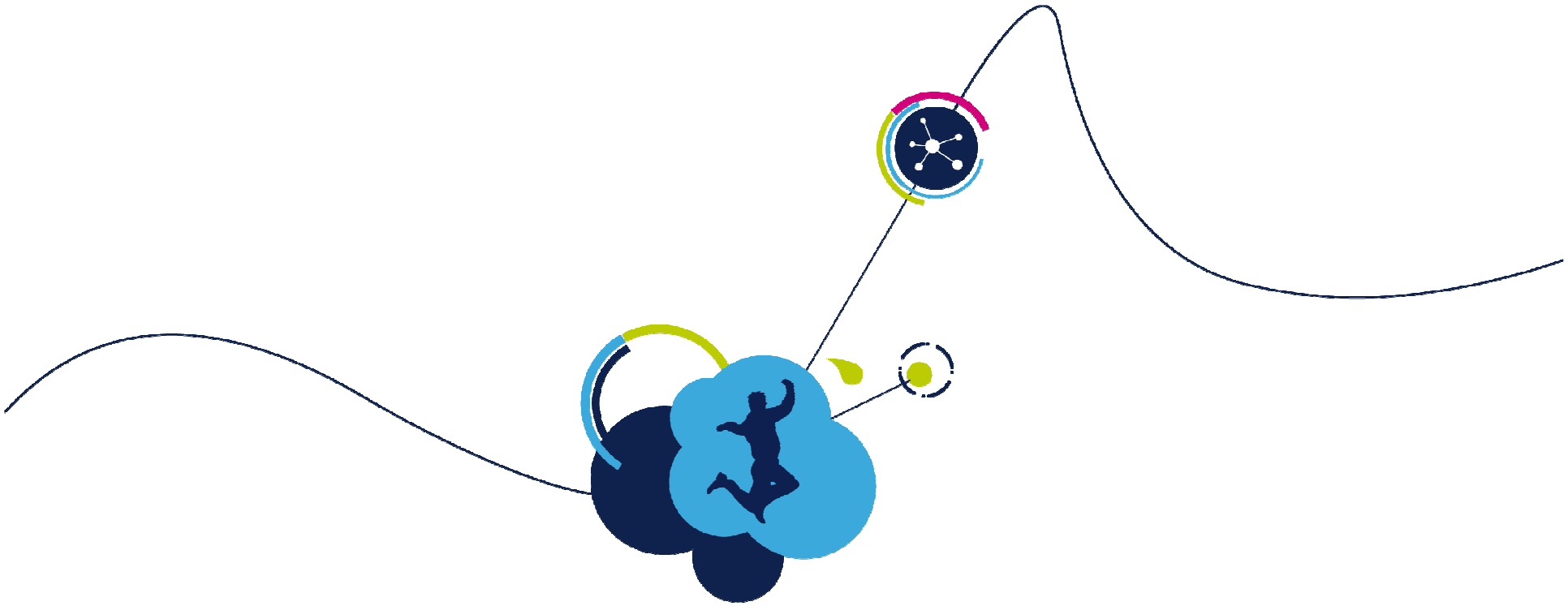
On Chip Oscillators: usage

- HSE, HSI16, MSI and PLL can feed system clock
  - MSI is the backup clock in case of HSE failure
- HSI16 and MSI are factory trimmed and user trimmable
- HSI48 can be trimmed through USB (SOF) or LSE
- MSI, HSI or HSI/4 feeds System clock after exit from STOP
- LSI feeds IWDG and optionally the RTC for Auto Wake-Up (AWU) from STOP/STANDBY mode
- The clock used for baud rate/communication frequency for USART and I2C is independent from APB clock
- Clock-independent system clock sources for TIM2/TIM21/TIM22: LSE clock is internally redirected to the 3 timers' ETR inputs.
- It is possible to indirectly measure the frequency of all on-board clock source generators (LSI/LSE/MSI/HSE) by means of the TIM21 channel 1 input capture.

Clock Scheme

- **System Clock (SYSCLK) sources**
  - MSI (default @ 2.097MHz)
  - HSI16
  - HSE
  - PLL
- **Configurable dividers provides AHB, APB1/2 and TIM clocks**
- **RTC Clock (RTCCLK) and LCD Clock (LCDCLK) sources**
  - LSE
  - LSI
  - HSE clock divided by 2, 4, 8 or 16
- **USB/SDIO Clock (USBCLK) provided from HSI48 or the internal PLL (PLLVCO/2)**
- **Clock-out capability on the MCO pin (PA08/PA09).**
- **Clock Security System (CSS) to backup clock in case of HSE clock failure (MSI feeds the system clock)**
  - Enabled by SW w/ interrupt capability linked to Cortex NMI





General-purpose I/Os (GPIO)

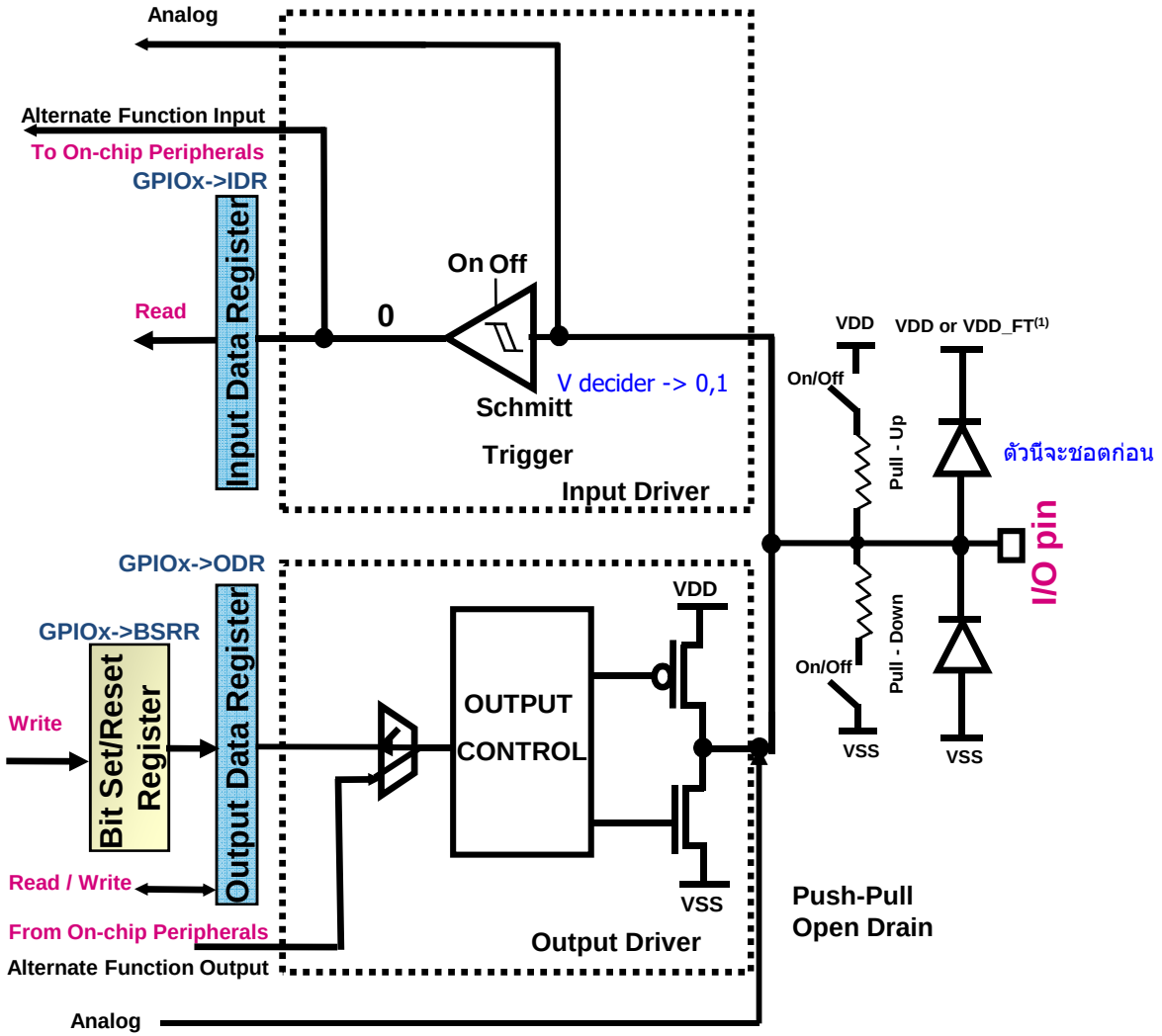
life.augmented

GPIO features

- Up to 51 multifunction bi-directional I/O ports available: 83% IO ratio
- Most standard I/Os are 5V tolerant* ปกติไฟควรเป็น 3.6 V แต่เอาชัวๆ 3.3 V ไปเลย
- All Standard I/Os are shared in 6 ports (GPIOA..GPIOF)
- Atomic Bit Set and Bit Reset using BSRR register
- GPIO connected to AHB bus: max toggling frequency = $f_{AHB}/2$ = 16 MHz
- Configurable Output Speed up to 40 MHz
- Ultralow leakage per I/O: 50 nA (maximum @ max temperature)
- Up to 51 GPIOs can be set-up as external interrupt (up to 16 lines at time) able to wake-up the MCU from low power STOP mode.
- Three I/Os (PA0, PC13 and PE6) can be used as Wake-Up sources from STANDBY mode, and as Tamper Pin (PA0, PC13 and PE6) to reset back-up registers
- One I/O (PC13) can be set-up TimeStamp, RTC Alarm Output, RTC Wakeup Output or RTC Clock output

GPIO Configuration Modes

MODER(i) [1:0]	OTYPER(i) [1:0]	PUPDR(i) [1:0]	I/O configuration
01	0	0 0	Output Push Pull
		0 1	Output Push Pull with Pull-up
		1 0	Output Push Pull with Pull-down
	1	0 0	Output Open Drain
		0 1	Output Open Drain with Pull-up
		1 0	Output Open Drain with Pull-down
10	0	0 0	Alternate Function Push Pull
		0 1	Alternate Function PP Pull-up
		1 0	Alternate Function PP Pull-down
	1	0 0	Alternate Function Open Drain
		0 1	Alternate Function OD Pull-up
		1 0	Alternate Function OD Pull-down
10	x	0 0	Input floating
		0 1	Input with Pull-up
		1 0	Input with Pull-down
11	x	x	Analog mode

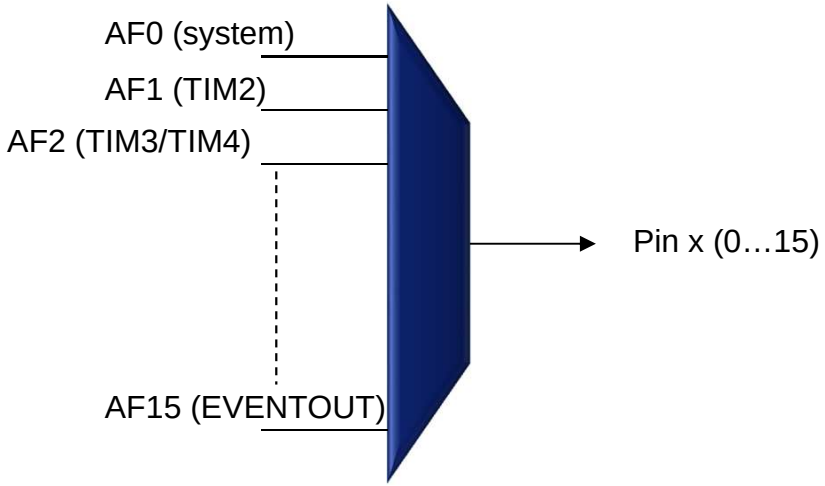


* In output mode, the I/O speed is configurable through OSPEEDR register: 400kHz, 2MHz, 10MHz or 40 MHz

(1) VDD_FT is a potential specific to five-volt tolerant I/Os and different from VDD.

Alternate Functions features

- Most of the I/O pins are shared with Alternate Functions pins (like USARTx_Tx, TIMx_CH2, I2Cx_SCL, SPIx_MISO, USBDM, EVENTOUT…)
- Some Alternate function can be remapped in different pins allowing optimization of the pin out
- Alternate functions can be connected to onboard peripherals through a multiplexer that allows only one peripheral's alternate function to be connected to an I/O pin at a time.

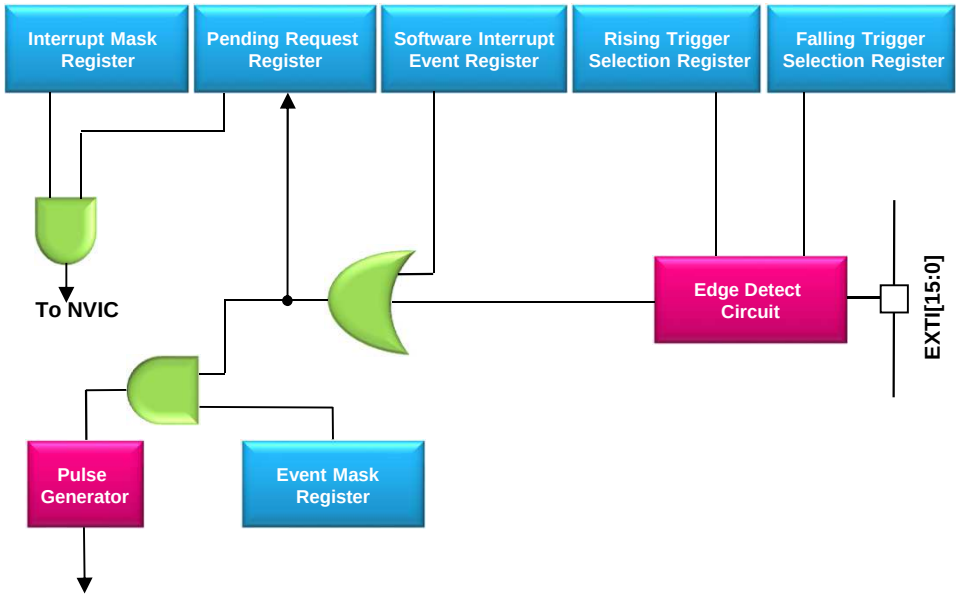


In this way, there can be no conflict between peripherals sharing the same I/O pin.

Refer to **the "Alternate function input/output"** table in the STM32L0xx datasheet for peripherals' alternate function I/O pins mapping

EXTENDED INTERRUPT/EVENT CONTROLLER (EXTI) Features

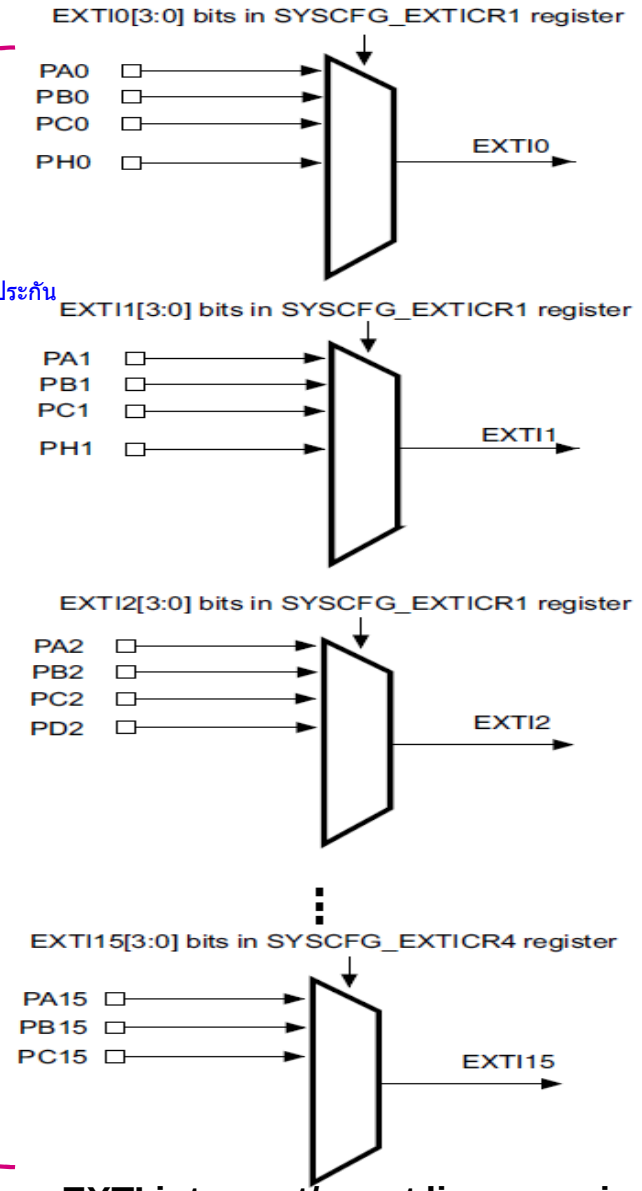
- Manages the external and **internal asynchronous** events/interrupts and generates the event request to the CPU/Interrupt Controller and a wake-up request to the Power Manager
- Some communication peripherals (UART, I2C, comparators) are able to generate events when the system is in run/sleep mode and also when the **system is in stop mode** allowing to wake up the system from stop mode.
- These peripherals are able to generate both a synchronized (to the system APB clock) and an asynchronous version of the event.
- All others features are same as STM32L1x series



Extended interrupt/event GPIO mapping

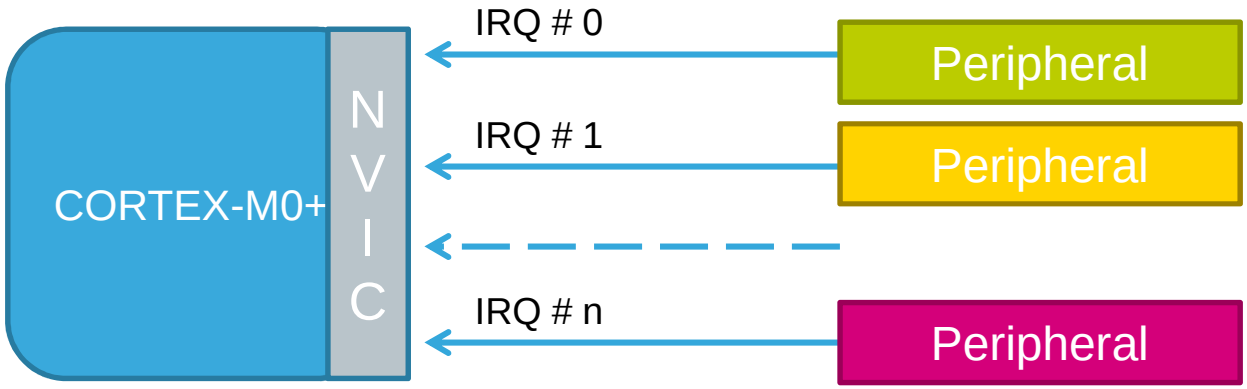
- Up to 28 Interrupts / Events requests :
  - Up to 55 GPIOs can be used as EXTI line(0..15)
  - EXTI line 16 is connected to the PVD output
  - EXTI line 17 is connected to the RTC Alarm event
  - EXTI line 18 is reserved (internally held low)
  - EXTI line 19 is connected to RTC tamper and Timestamps
  - EXTI line 20 is reserved (internally held low)
  - EXTI line 21 is connected to Comparator 1 output
  - EXTI line 22 is connected to Comparator 2 output
  - EXTI line 23 is connected to I2C1 wakeup
  - EXTI line 24 is reserved (internally held low)
  - EXTI line 25 is connected to USART1 wakeup
  - EXTI line 26 is reserved (internally held low)
  - EXTI line 27 is connected to CEC wakeup.
- The GPIOs are connected to 16 configurable interrupt/event lines as shown.

ตัวเช็คประกัน



EXTI interrupt/event line mapping

Cortex-M0+ Interrupt processing



- Set Interrupt Priority
  - 4 levels of priority
    - Level 0 (highest) – Level 3 (lowest)
- Enable Interrupt Line at the NVIC side.
- When an interrupt occurs, the NVIC will check the priority, if highest, then the core will fetch and execute the service routine.
- If interrupts have the same software priority, the NVIC will resolve the conflict through their hardware priority as defined by their position in the Interrupt Vector Table. Refer to NVIC section in the STM32L0 Reference Manual.

- Enable Interrupt request generation at the Peripheral side.

  Example:
  - GPIO EXTI (Rising\falling edge detect)
  - USART RxNE (Receive not empty)
  - USART TxE (Transmit Empty)

STM32LoRa discovery kit
Overview

life.augmented

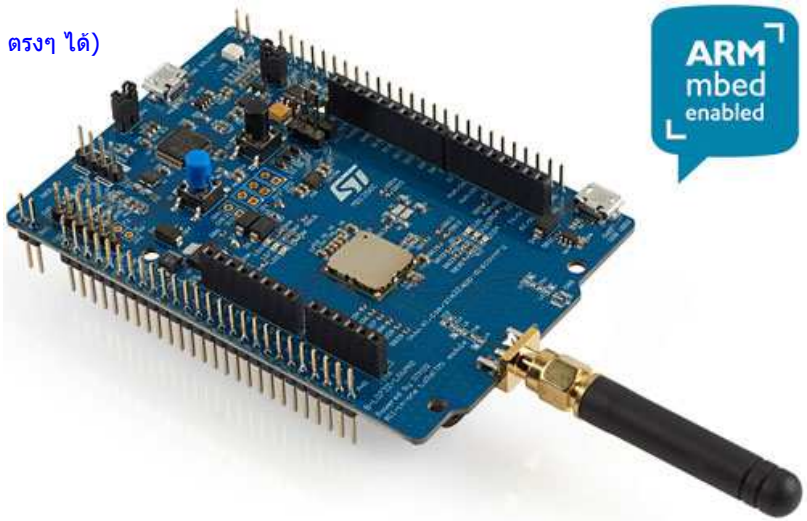
B-L072Z-LRWAN1 Overview

This Discovery kit features an all-in-one open module *CMWX1ZZABZ-091* (by Murata). The module is powered by an STM32L072CZ and an SX1276 transceiver. The transceiver features the LoRa®long-range modem, providing ultra-long-range spread spectrum communication and high interference immunity, minimizing current consumption. Since CMWX1ZZABZ-091 is an open module, user has access to all STM32L072 peripherals such as ADC, 16-bit timer, LP-UART, I2C, SPI and USB 2.0 FS (supporting BCD and LPM).

- **Key Features**

open module (แก้โค้ดใน module ตรงๆ ได้)

  - CMWX1ZZABZ-091 LoRa® module (Murata)
  - SMA and U.FL RF interface connectors
  - Including 50 Ohm SMA RF antenna
  - On-board ST-LINK/V2-1 supporting USB re-enumeration capability
  - USB ST-LINK functions:
  - Board power supply:
  - Through USB bus or external VIN /3.3 V supply voltage or batteries
  - 3xAAA-type-battery holder for standalone operation
  - 4 general-purpose LEDs
  - 2 push-buttons (user and reset)
  - Arduino™ Uno V3 connectors
  - ARM® mbed™ (see http://mbed.org)

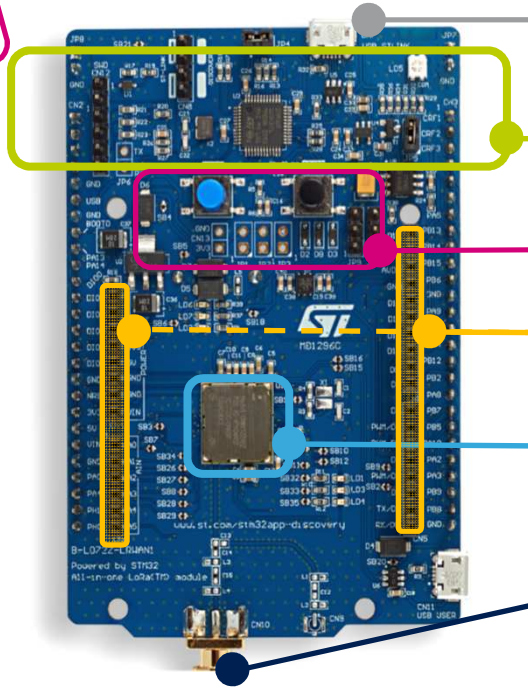


life.augmented

New Hardware tool

B-L072Z-LRWAN1: STM32 and LoRa® Discovery kit

Available now!



Flexible board power supply :
through USB or external source

Integrated ST-Link/V2-1:
mass storage device flash programming

2 push buttons, 2 color Leds,
Jumper settings

Arduino™ extension connectors :
easy access to add-ons

Murata module

SMA Antenna connector

life.augmented

Arduino Header

Labels usable in code

PX_Y — MCU pin without conflict

PX_Y — MCU pin connected to other components
See **PeripheralPins.c** (link below) for more information

XXX — Arduino connector names (A0, D1, ...)

XXX — LEDs and Buttons (LED_1, USER_BUTTON, ...)

Labels not usable in code (for information only)

XXX — Serial pins (USART/UART)

XXX — SPI pins

XXX — I2C pins

XXX — PWMOut pins (TIMER n/c[N])
n = Timer number c = Channel
N = Inverted channel

XXX — AnalogIn (ADC) and AnalogOut pins (DAC)

XXX — CAN pins

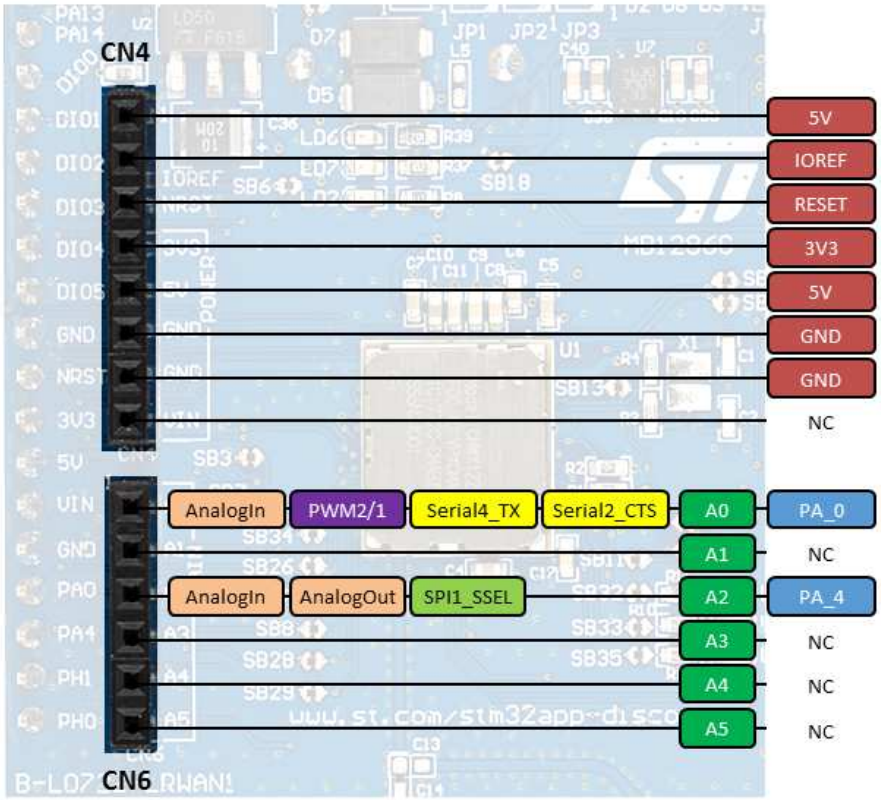
XXX — Power and control pins (3V3, GND, RESET, ...)

Arduino Header

life.augmented

DISCO-L072CZ-LRWAN1

ARDUINO HEADER
(top left side)

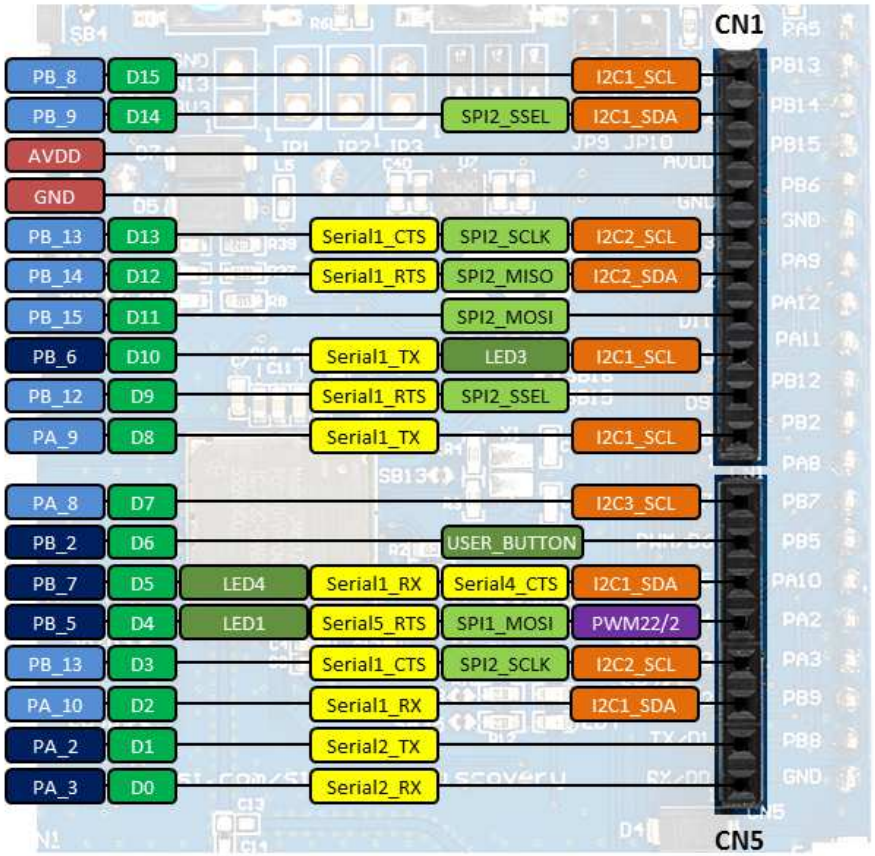


Arduino Header

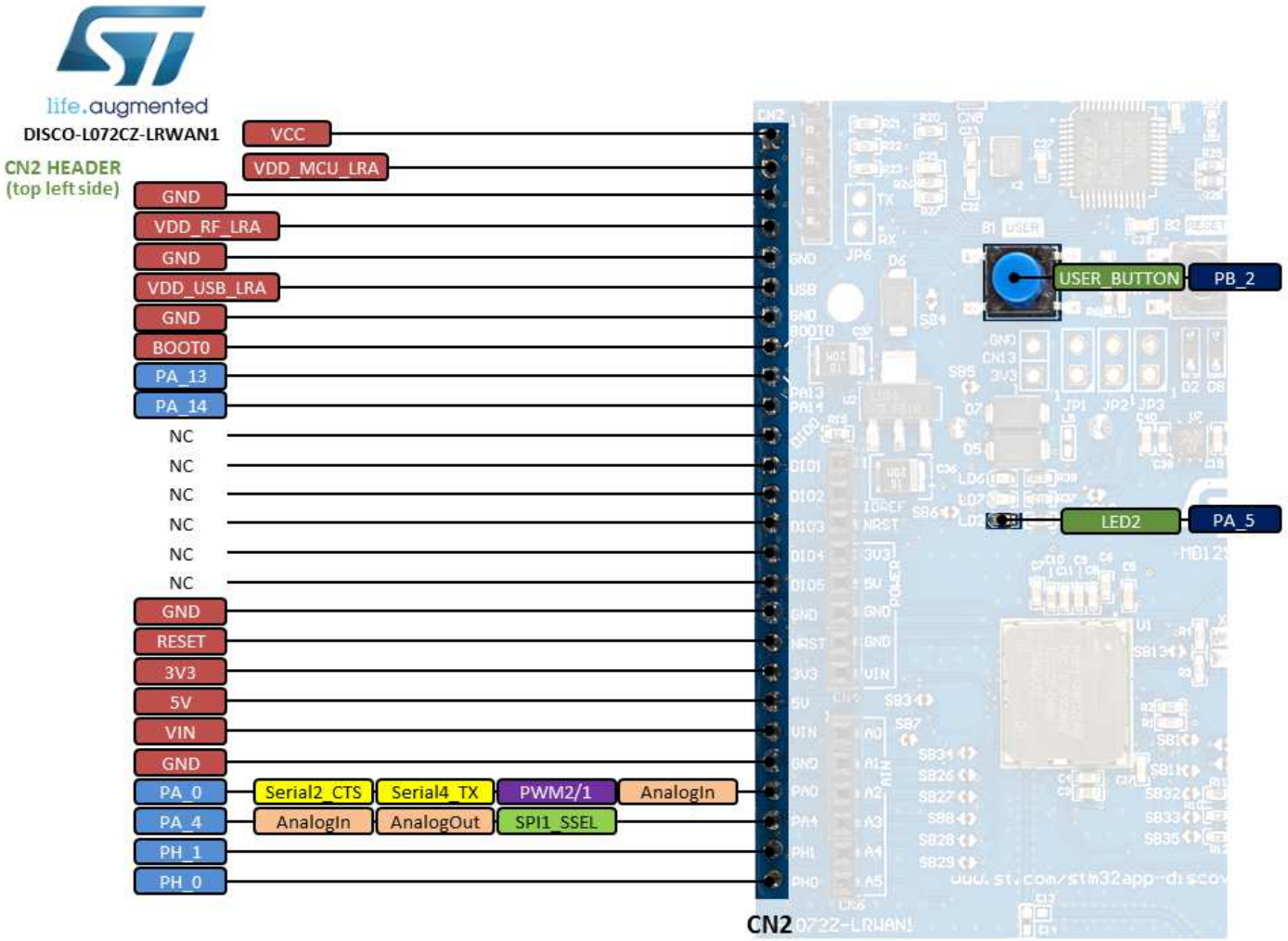
life.augmented

DISCO-L072CZ-LRWAN1

ARDUINO HEADER
(top right side)



Arduino Header

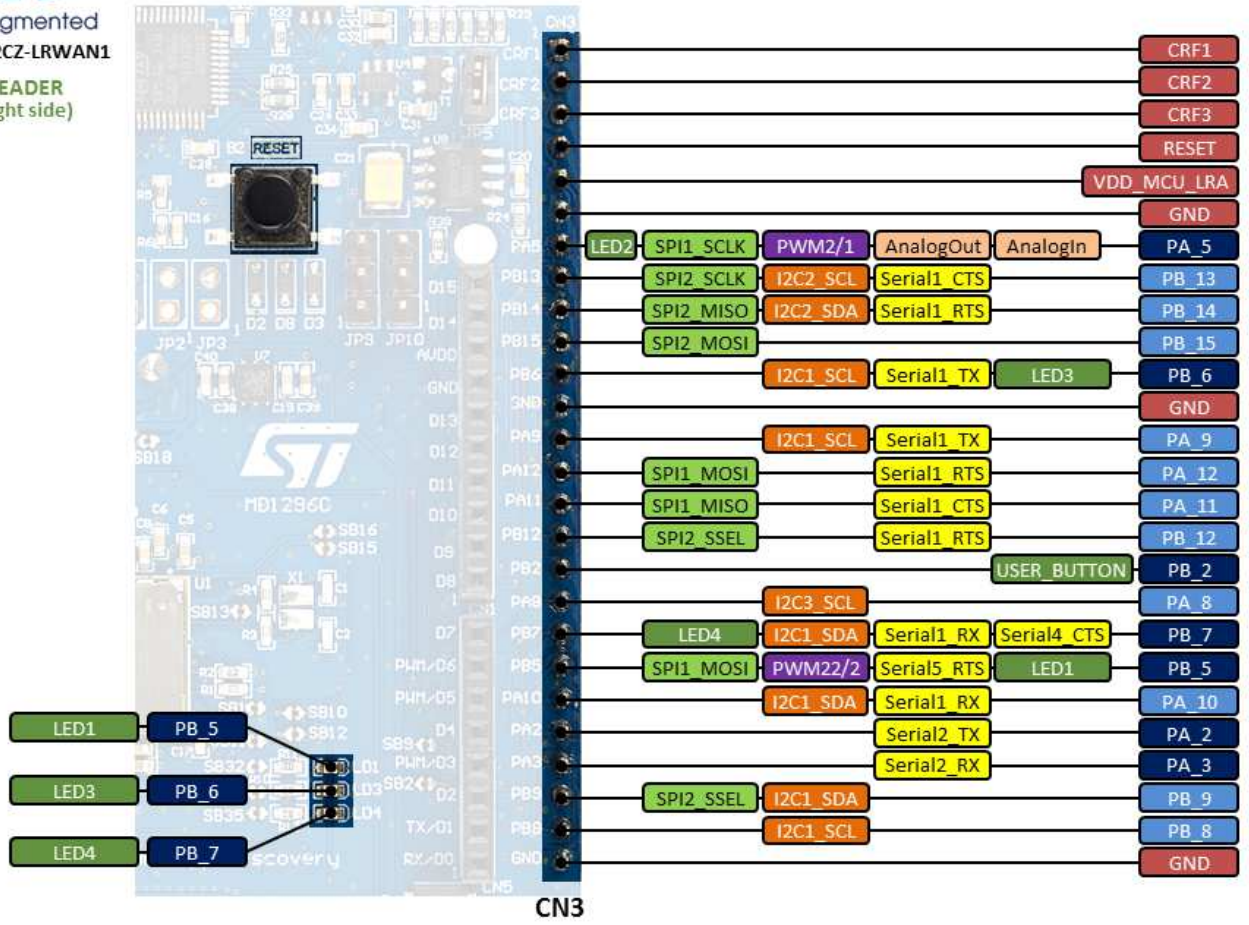


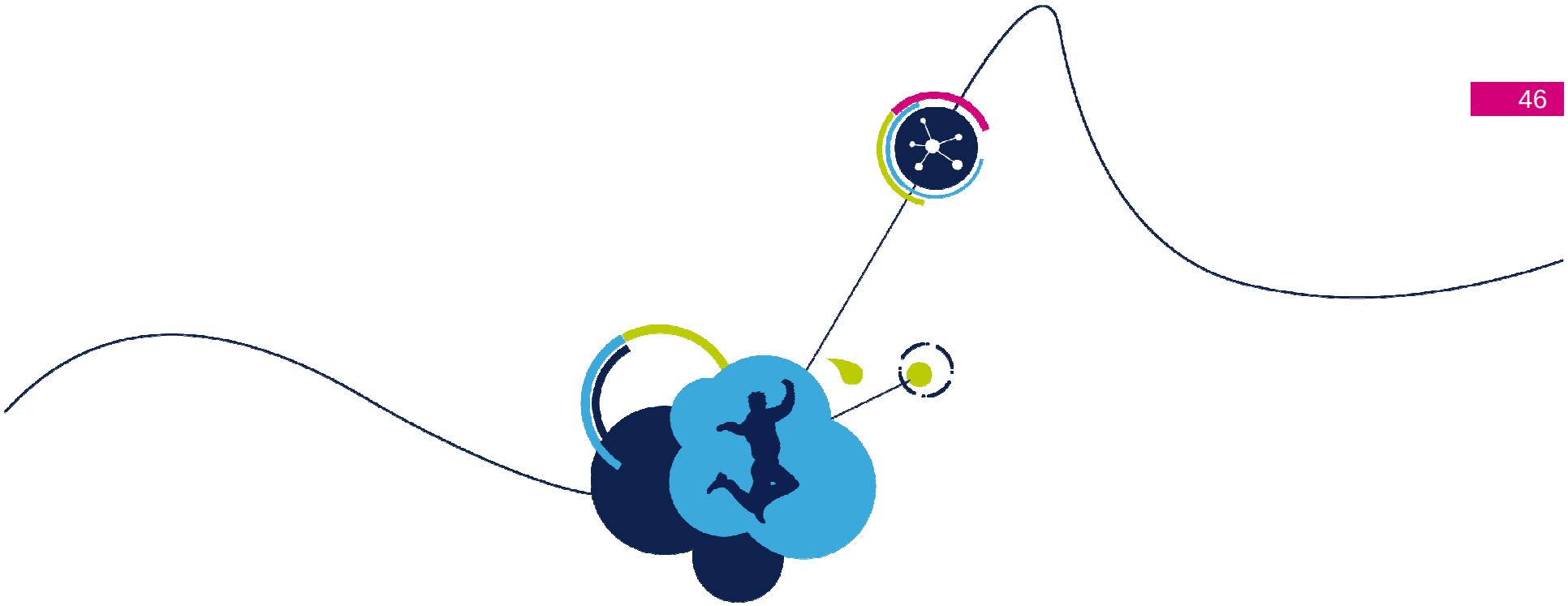
Arduino Header

life.augmented

DISCO-L072CZ-LRWAN1

CN3 HEADER
(top right side)





LoRa Device Firmware Library Overview

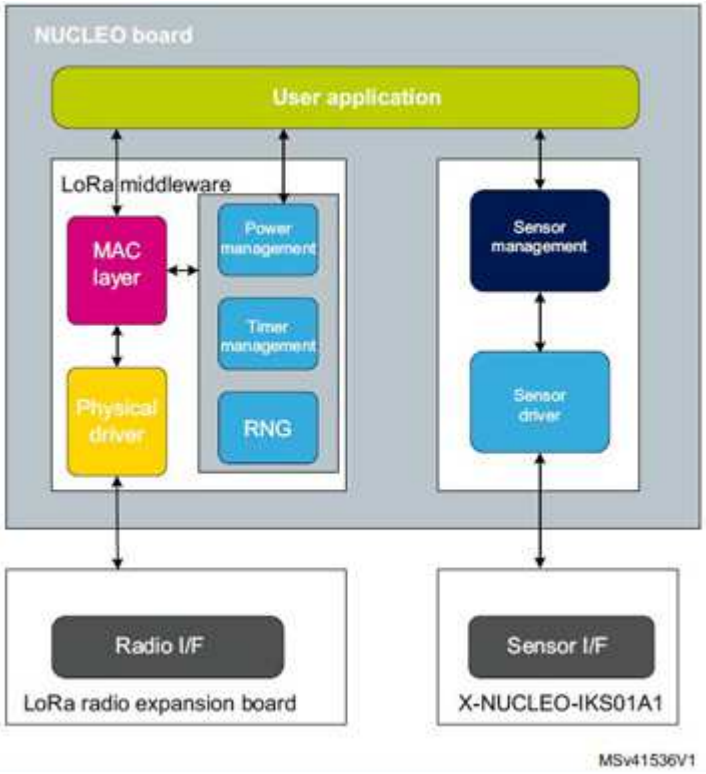
life.augmented

I-CUBE-LRWAN: LoRaWAN software expansion for STM32Cube

The I-CUBE-LRWAN software expansion package consists of a set of libraries and application examples for STM32L0, STM32L1 and STM32L4 devices acting as end devices.

Downloadable from www.st.com/i-cube-lrwan

- **Key Features**
  - Compliant with the LoRa™ Alliance specification protocol named LoRaWAN™ version V1.0.2
  - Bidirectional end-devices with class A and class C protocol support
  - Class A certified (V1.0) and class C functional
  - EU 868 MHz ISM band ETSI (European telecommunications standards institute) compliant
  - EU 433 MHz ISM band ETSI compliant
  - US 915 MHz ISM band FCC (federal communications commission) compliant
  - End-device activation either via OTAA (over-the-air activation) or via ABP (activation-by-personalization)
  - Adaptive data rate support
  - LoRaWAN™ test application for certification tests included
  - Low-power optimized
  - Full STM32 portfolio compatible

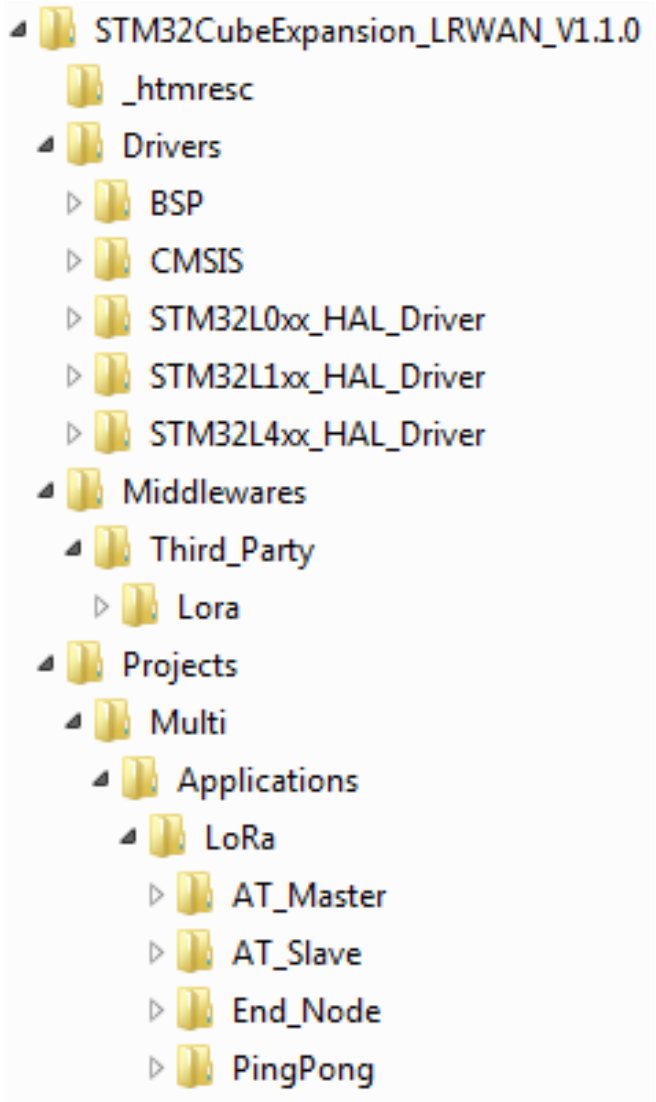


- Supported LoRa radio expansion boards by Semtech
  - SX1276MB1MAS, SX1276MB1LAS and SX1272MB2DAS
- Supported ST boards
  - NUCLEO-L053R8, NUCLEO-L152RE, NUCLEO-L476RG, and B-L072Z-LRWAN1.
  - X-NUCLEO-IKS01A1 (sensor expansion board)

life.augmented

I-CUBE-LRWAN firmware package

- LoRaWan 1.0.2 compliant with lots of new regions supported!. To choose your region, select one of the supported compile switch:
  - REGION_AS923
    - Brunei, Cambodia, Hong Kong, Indonesia, Japan, Laos, New Zealand, Singapore, Taiwan, Thailand, Vietnam
  - REGION_AU915
    - Australia
  - REGION_CN470\REGION_CN779
    - China
  - REGION_EU433\REGION_EU868
    - Europe
  - REGION_KR920
    - Korea
  - REGION_US915\REGION_US915_HYBRID
    - US
- Includes libraries for the MEMS and Enironmental Sensors expansion board (X-NUCLEO-IKS01A1)
- Application examples available for
  - NUCLEO-L053R8, NUCLEO-L152RE and NUCLEOL476RG using Semtech expansion boards
  - P-NUCLEO-LRWAN1, STM32 Nucleo pack for LoRa® technology
  - B-L072Z-LRWAN1, STM32 Discovery kit embedding the CMWX1ZZABZ-091 LoRa® module (Murata)
  - I-NUCLEO-LRWAN1, LoRa® expansion board for STM32 Nucleo, based on the WM-SG-SM-42 LPWAN module (USI®).



life.augmented

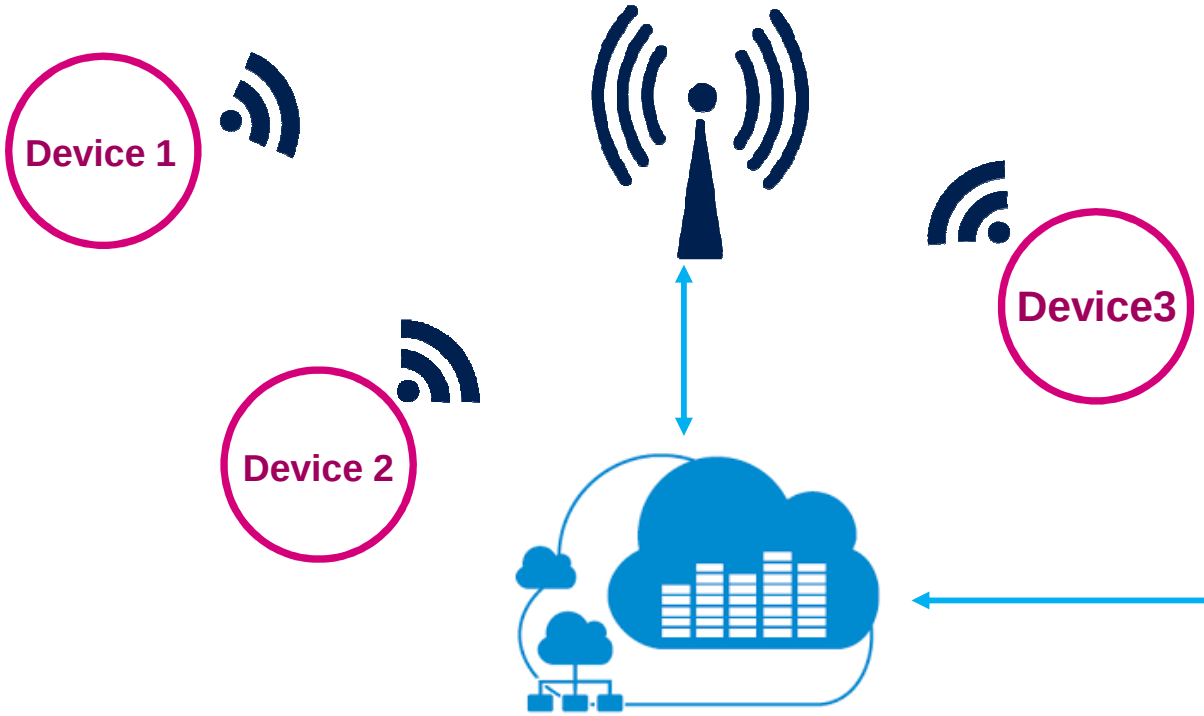
LoRaWAN™ Network Overview

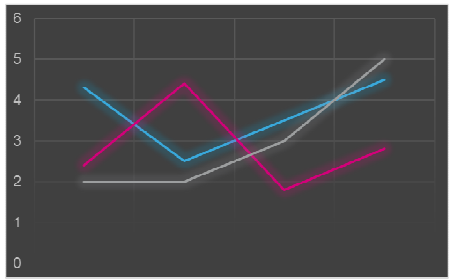
LoRaWAN™ Network

Get connected to the cloud !

Application Dashboard:

- Seamless provisioning
- Scheduling
- Rules Engine
- Visualization
- Threshold Alerts
- Remote Monitoring & Control
- Asset Tracking

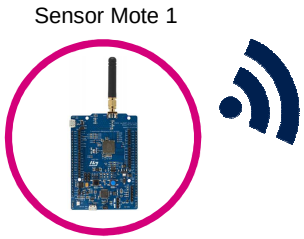


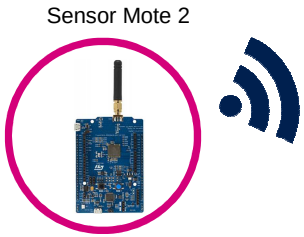


life.augmented

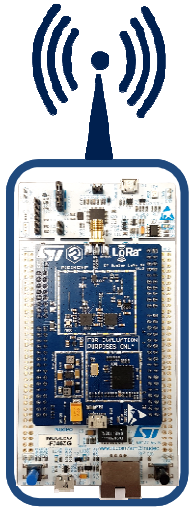
LoRaWAN™ Network Protocol

Network Topology Overview





Sub-GHz



3G / 4G / ***Ethernet*** (IP)

Ethernet (IP)

Devices

The aim is to sense, track, record, monitor any activity. They could be connected to several Gateway and many devices could be connected to the same Gateway. Devices are base on SX1276(100MHz – 1020MHz) or SX1272 (800MHz – 1020MHz) radio.

- **Each device has its own unique device address (DevAddr)**

Gateways

Is ‘’just’’ a bridge between the devices and the Network Server. The Gateway(Base station) are Telco properties (Verizon, Orange, Bouygue-T …), The Semtech radio chipset is the SX1301.

Network Servers (NS)

Will decode and authenticates data from the device. If data is addressed to the NS then it will be forwarded to the right Application Server. Other players on top of Telco may offer Network Server services.

- **Each N.S has its own and unique Network Session Key (NwkSKey)**

Application Servers (AS)

Runs the application by decrypting the packets coming fro the NS. The App could run on a cloud server, PC, Phone … . Then the data are sent back to the right device via the N.S and the Gateway. Many AS can coexist in the same Network.

- **Each A.S has its own and unique Application Session Key (AppSKey)**

life.augmented

LoRaWAN™ – Joining Process

2 activation methods exist to join an existing network

1. OTTA: Over the Air Activation = Over the air handshaking

- End-device send a Join Request (J.S) message to A.S including DevEUI, AppEUI, AppKey
  - DevEUI (Global unique end-device identifier) will allow the network to identify the device
  - AppEUI to identify which A.S the end-device is talking to
  - AppKey to identify which application on the A.S the end-device is referring to
- End device receives the Join Accept from the A.S and proceed with
  - Join Accept (authenticates and decrypt the Join Accept)
  - Extract and store DevAddr (the 32-bit device address)
  - Derives the 2 security: Network session Key (NwkSKey) and Application Session Key (AppSKey)

2. ABP: Activation By Personalization (no Over the air handshaking)

- The Device ready out-of-the-box to talk on the network
- The DevAddr, NwkSKey and AppSKey are programed at factory level (no Over the air handshaking)

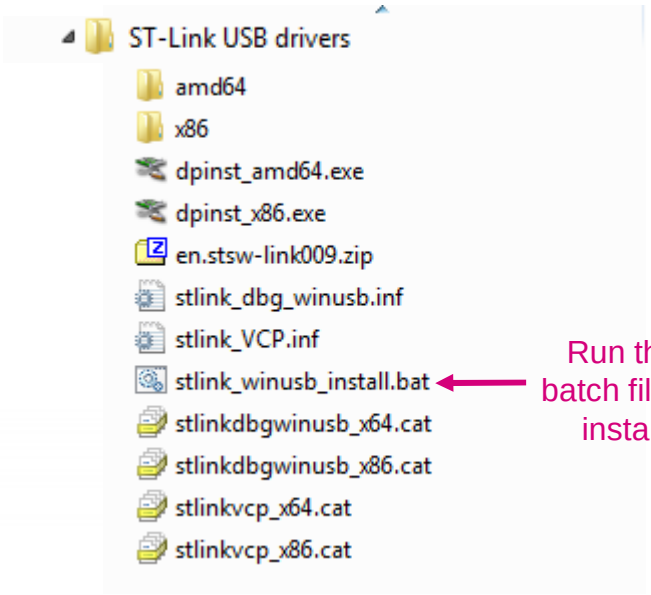
LoRa Sensor Device Setup and Reconfiguration

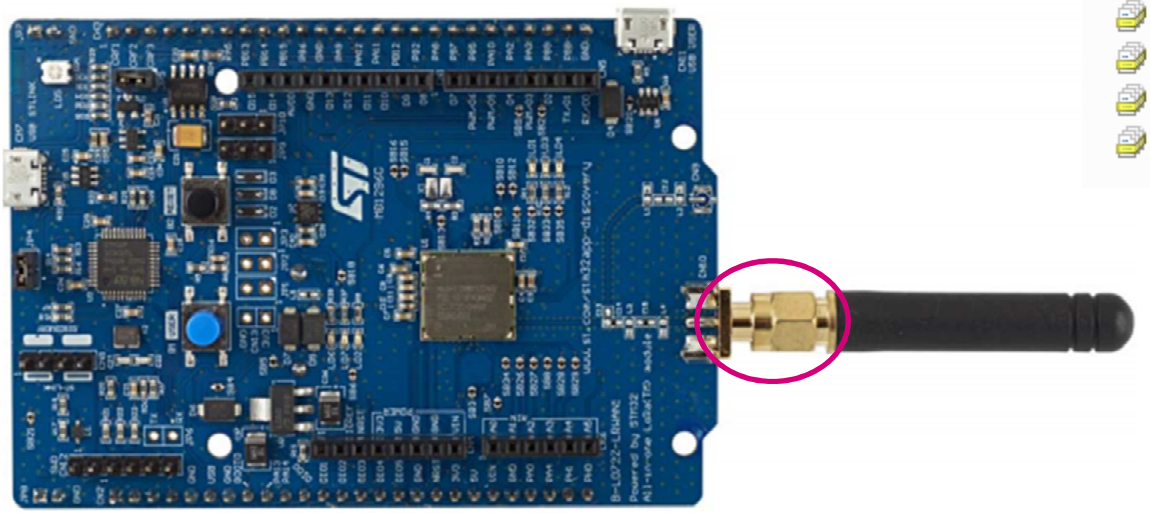
life.augmented

Sensor device power up (1/3)

LoRa sensor device setup

1. Make sure that the *USB drivers are installed*. (drivers downloadable from www.st.com/stlinkv2 or get from files shared to you)
2. On the B-L072Z-LRWAN1, connect the antenna to the SMA connector.

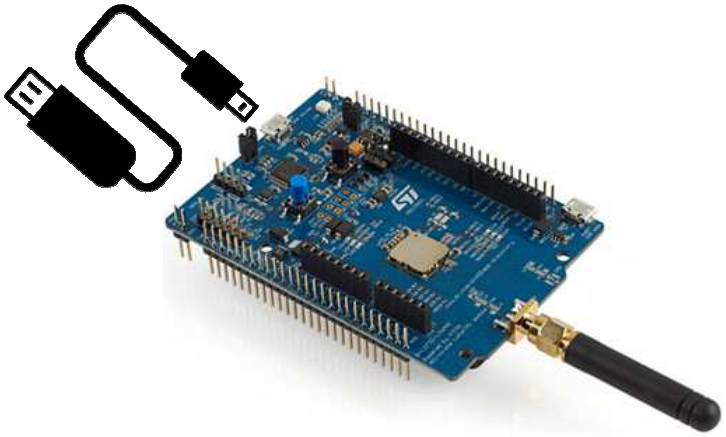


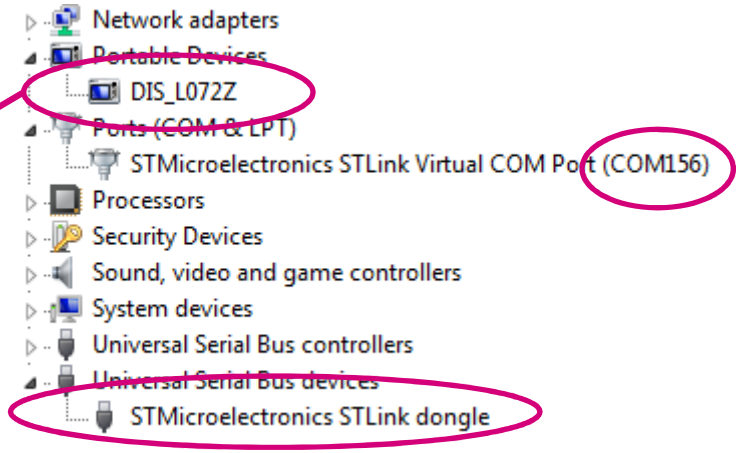


Sensor device power up (2/3)

3. Connect the Nucleo board to a PC with a USB cable type A to mini-B through USB connector CN1 to the power the board. Then green LED LD3 (PWR) and LD1 (COM) light up.
4. Allow the PC to enumerate and install the USB drivers. *Take note of the virtual COM port number assigned to the board*.

Note: The Nucleo board is also enumerated as an mbed removable storage device.

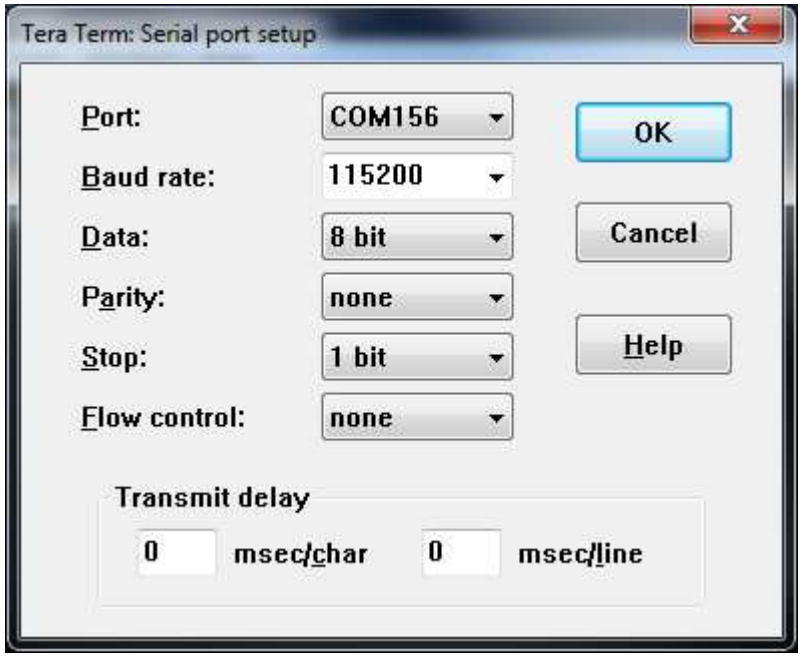


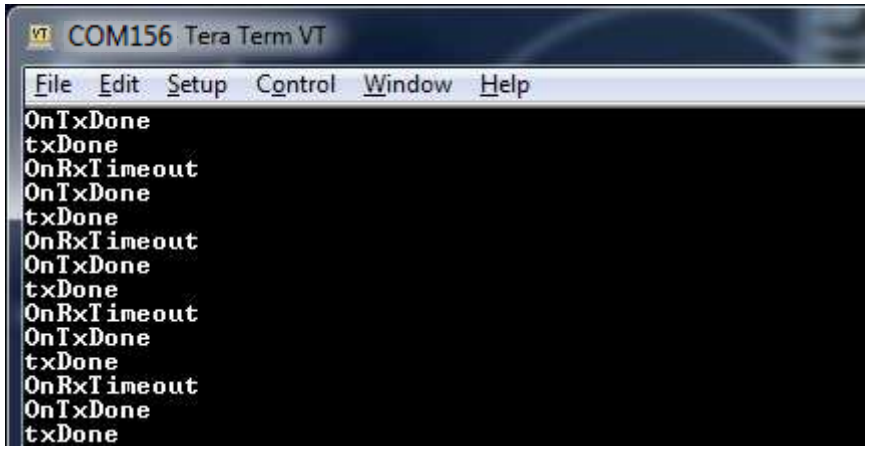


life.augmented

Sensor device power up (3/3)

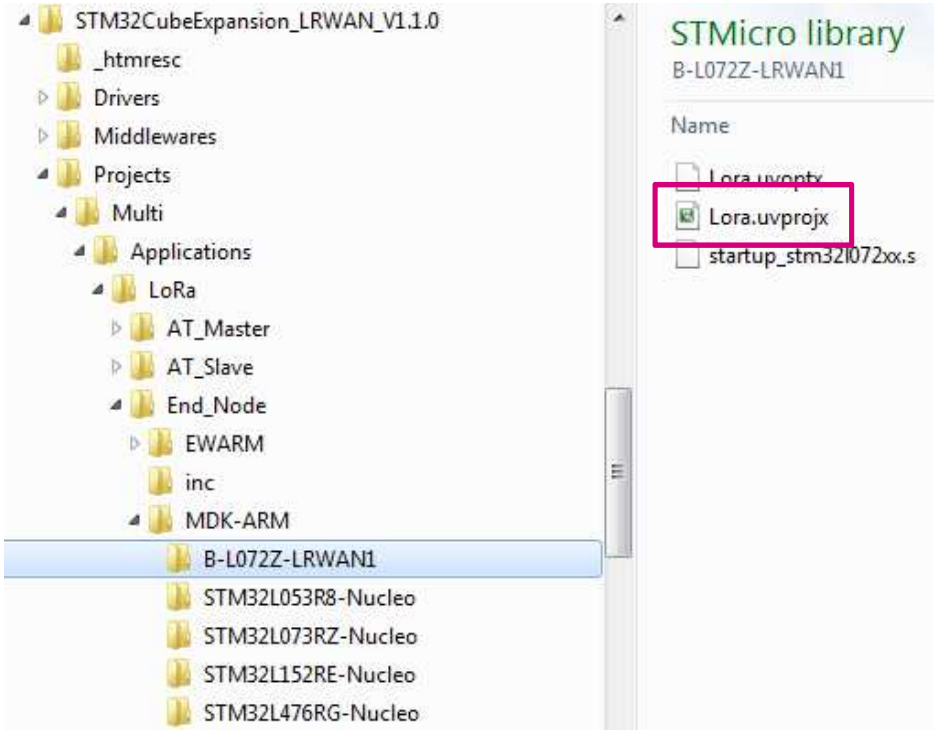
5. Open a terminal emulation software (e.g. Tera Term) and configure it with the following settings:
   - Port: (from step 4)
   - Baud rate: 115200
   - Data: 8 bit
   - Parity: none
   - Stop bit: 1 bit
6. Out of the box default demo:
   - Ping Pong demo





End Node Sample Project (1/4)

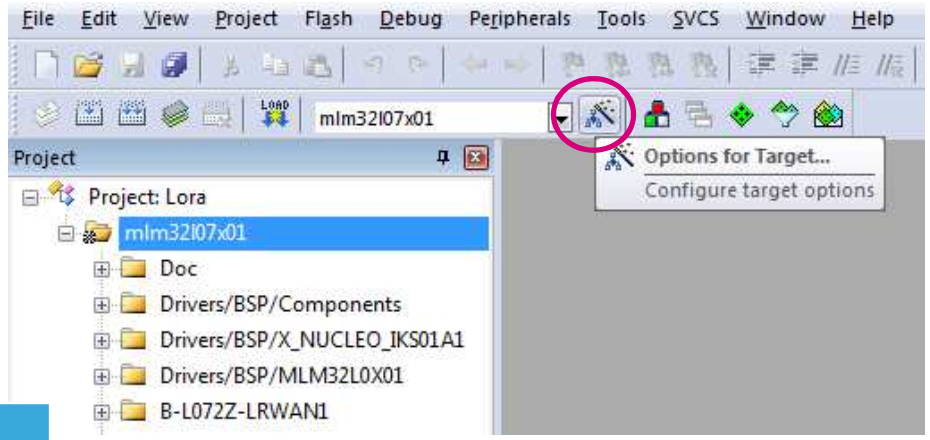
1. Download the firmware package from www.st.com/i-cube-lrwan and extract the files.
2. Go to \STM32CubeExpansion_LRWAN_V1.1.0\Projects\Multi\Applications\LoRa\End_Node
3. Open the Keil project in …\MDK-ARM\B-L072Z-LRWAN1 folder.



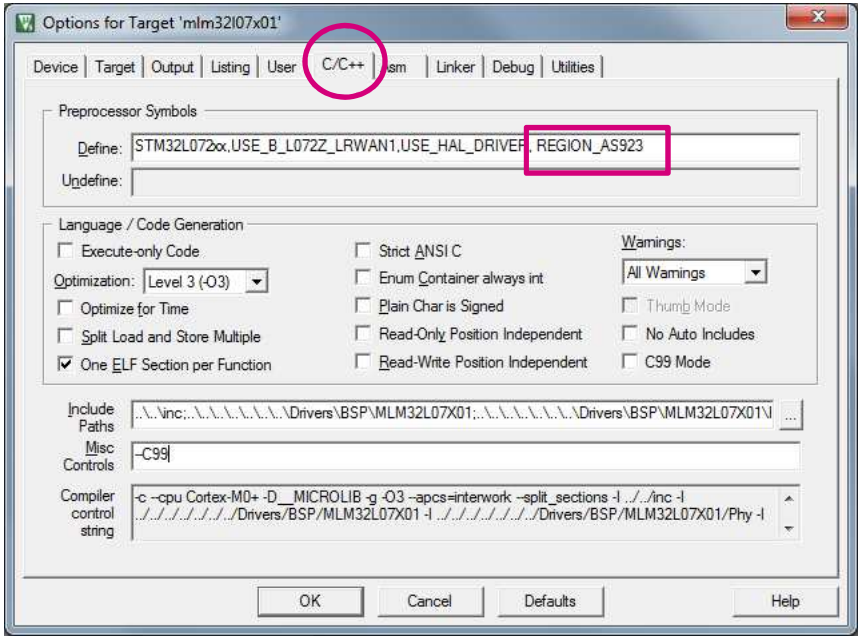
End Node Sample Project (2/4)

4. Got to Project options.
5. Go to the preprocessor settings and set to the desired frequency band. Use only one setting.

   For this workshop, use the define for REGION_AS923.



Define	Region
REGION_EU433 REGION_EU868	Europe
REGION_US915 REGION_US915_HYBRID	US
REGION_KR920	Korea
REGION_AU915	Australia
REGION_CN470 REGION_CN779	China
REGION_AS923	Brunei, Cambodia, Hong Kong, Indonesia, Japan, Laos, New Zealand, Singapore, Taiwan, Thailand, Vietnam



life.augmented

End Node Sample Project (3/4)

6. Edit the ***comissioning.h*** file in End_Node/inc to set the end device comissioning parameters.
   - For this workshop, set the following parameters.

     Use Activation by Personalization (ABP)

     ```
     #define OVER_THE_AIR_ACTIVATION                    0
     ```

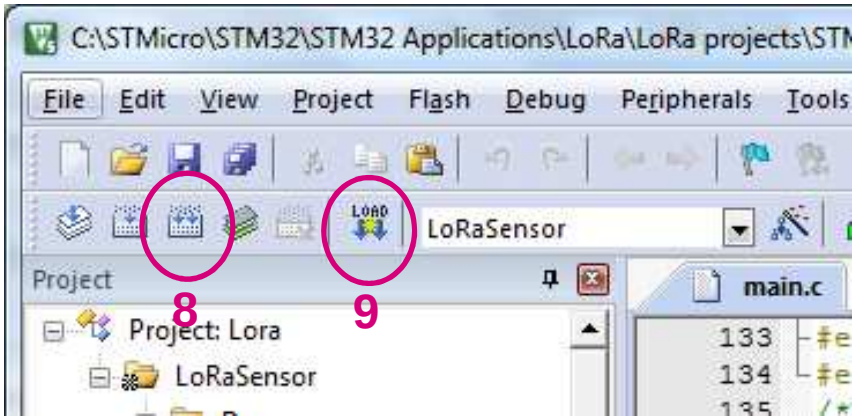
     Set the NWKSKEY and APPSKEY (***Default setting. No changes needed***)

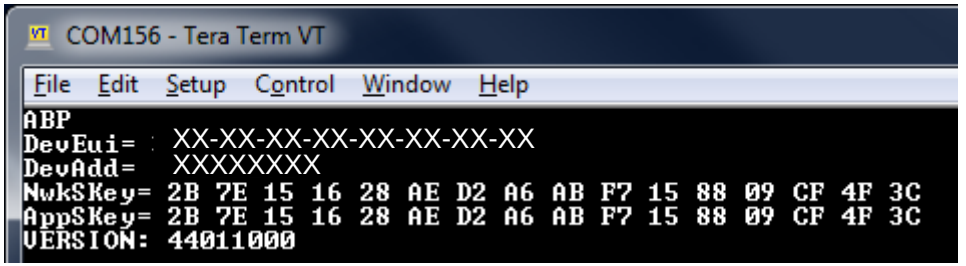
     ```
     #define LORAWAN_NWKSKEY   { 0x2B, 0x7E, 0x15, 0x16, 0x28, 0xAE, 0xD2,
                                 0xA6, 0xAB, 0xF7, 0x15, 0x88, 0x09, 0xCF,
                                 0x4F, 0x3C }
     #define LORAWAN_APPSKEY   { 0x2B, 0x7E, 0x15, 0x16, 0x28, 0xAE, 0xD2,
                                 0xA6, 0xAB, 0xF7, 0x15, 0x88, 0x09, 0xCF,
                                 0x4F, 0x3C }
     ```

7. Edit the hw_conf.h file in End_Node/inc to set additional options
   - Debug switches – to show debug printf messages
   - Low power mode – enable/disable low power mode
   - Sensor enable switch – enable it if X-NUCLEO-IKS01A1 (sensor expansion board) is attached

End Node Sample Project (4/4)

8. Rebuild all files and load your image into target memory
9. “Load” the firmware to the device.
5. Press the reset button B2 (black button) to view the device activation parameters which are shown only once after reset: ***DevEUI, DevAdd, NwkSKey, AppSKey***. These parameters will be used to enroll the device to the application server. See Network Setup.





Debug mode

- Debug, it can be disable and enable by comment/uncomment the define function in file “**hw_conf.h**”.

- Enable :

```
127 /* --------Preprocessor compile swicth------------ */
128 /* debug swicthes in debug.h */
129 #define DEBUG
130 //#define TRACE
```

- Disable:

```
127 /* --------Preprocessor compile swicth------------ */
128 /* debug swicthes in debug.h */
129 //#define DEBUG
130 //#define TRACE
```

- If the application need to lowest power consumption, we recommended to disable debugging mode.

Low Power mode

- Low power mode, the default source code is enable this feature in order to enter to low power mode when MCU finish send the data. But when MCU enter to low power mode the MCU will not debug and monitor the register, variable and memory.

- Enable :

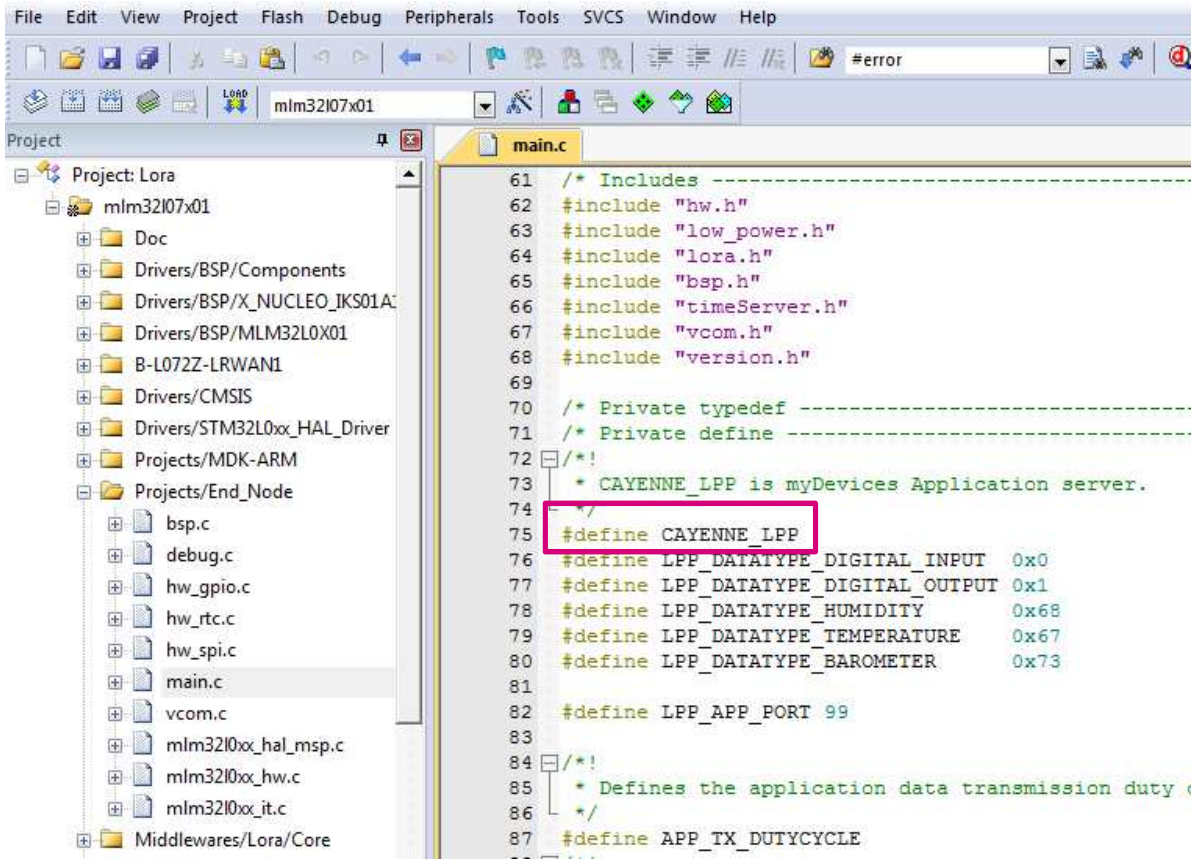
```
132 /* uncomment below line to never enter lowpower modes in main.c*/
133 //#define LOW_POWER_DISABLE
```

- Disable:

```
132 /* uncomment below line to never enter lowpower modes in main.c*/
133 #define LOW_POWER_DISABLE
```

Cayenne LLP payload format

- By default, the Cayenne LLP payload format is enabled. This is to be compatible with the myDevices Cayenne Dashboard application which will be used later for the visualization of the data. (https://mydevices.com/)
- Note that for US915\US915 HYBRID, the payload is limited to 11 bytes. You may need to remove some of the Cayenne LLP payload data to comply.

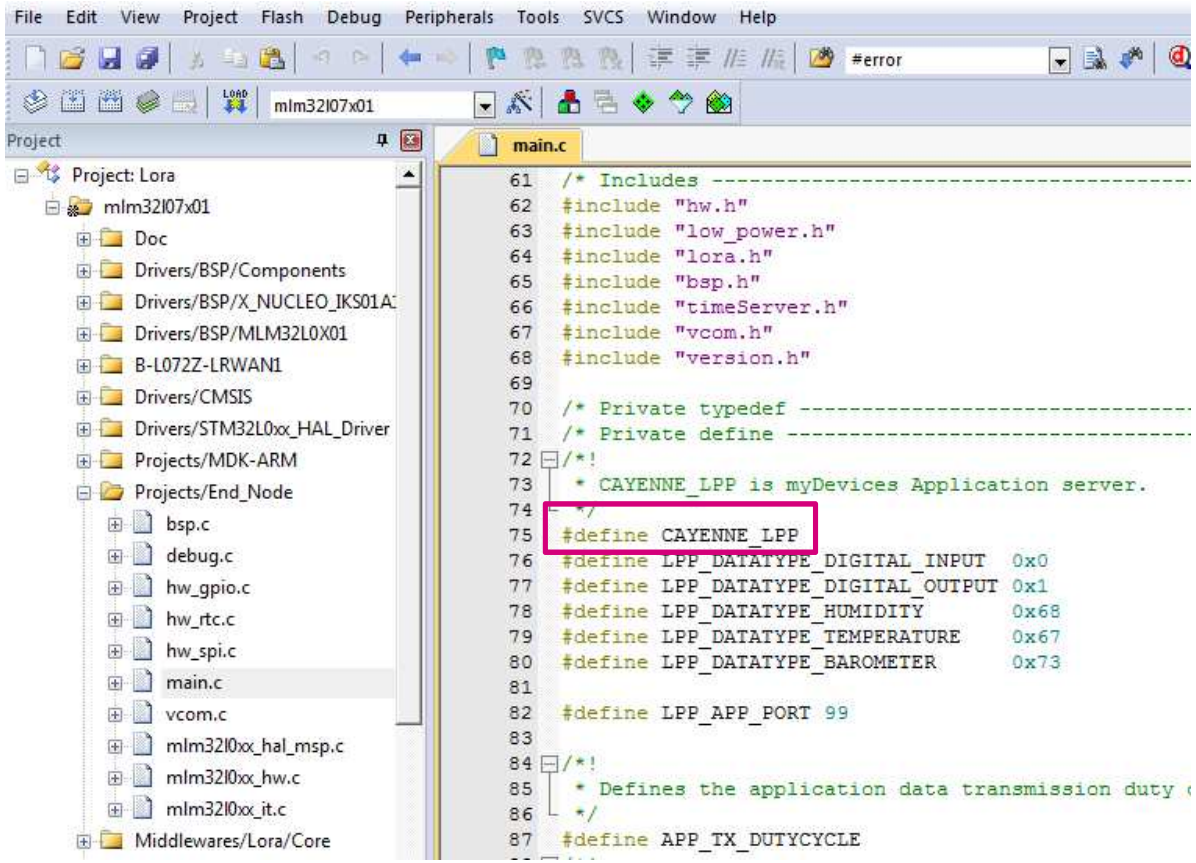


life.augmented

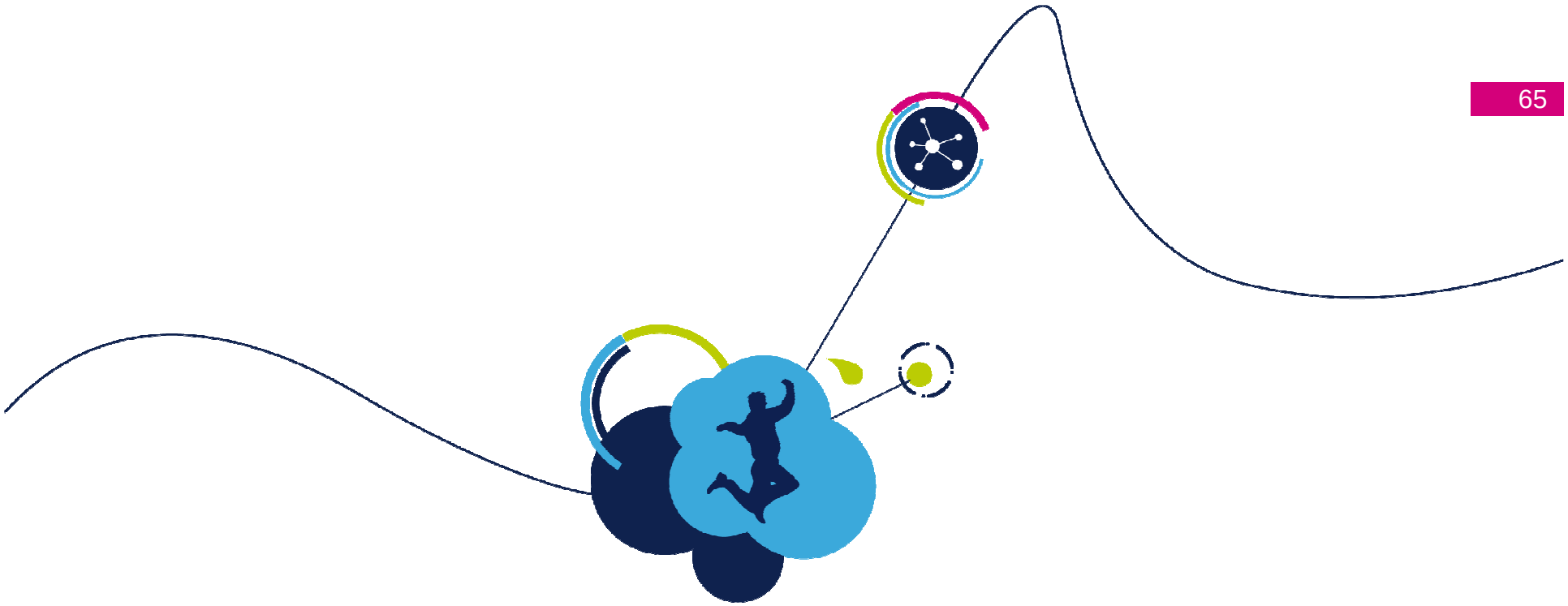
Follow the instructions in myDevices Cayenne

- MyDevices Cayenne and Loriot setup
  - https://mydevices.com/cayenne/docs/#lora-loriot-network

- Make sure that the Cayenne LLP payload format is enabled during the compilation of the End_Node code for the sensor device. (Enabled by default)



life.augmented

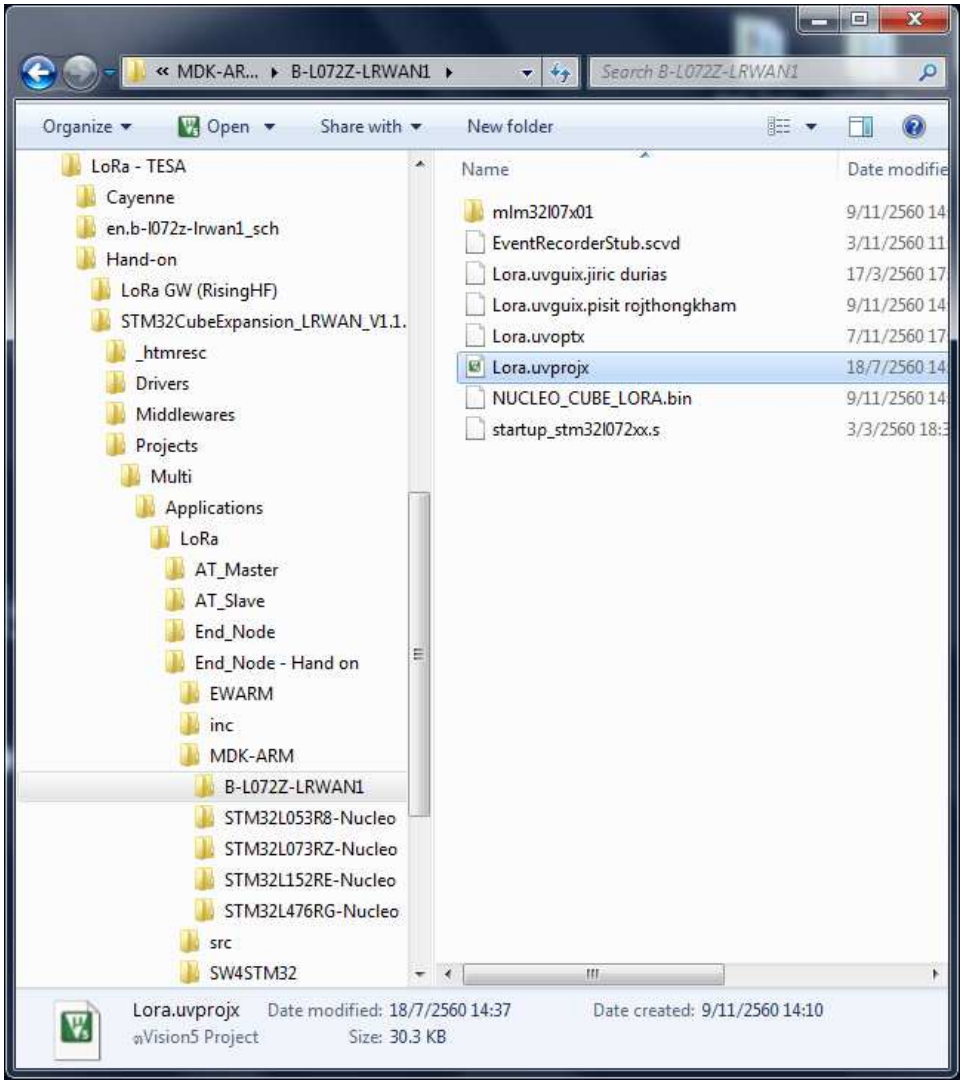


Hand-on GPIO

Hand on GPIO(1/5)

1. Go to \STM32CubeExpansion_LRWAN_V1.1.0\Projects\Multi\Applications\LoRa\End_Node – Hand on
2. Open the Keil project in …\MDK-ARM\B-L072Z-LRWAN1 folder.

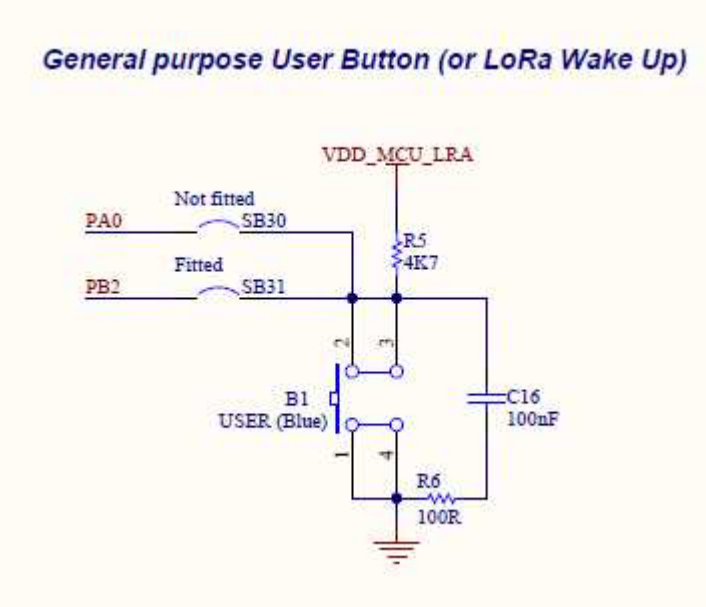
- The expectation of the LAB.
  - Count number of switch
  - Toggle LED when switch is pushed.

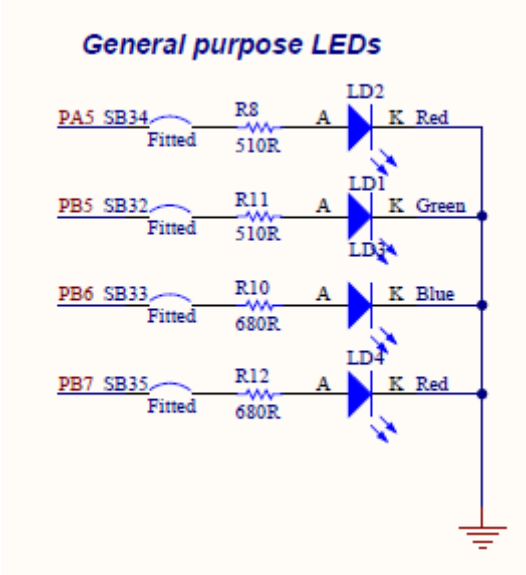


life.augmented

Hand on GPIO(2/5)

- Schematic overview (Input Button and Output LEDs)

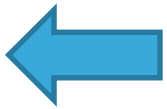




Hand on GPIO(3/5)

- 3. Initial input (User Button) by open file “mlm32l0xx_hw.c”
- 4. Add code in void HW_Init(void)

```
122  void HW_Init( void )
123  {
124    if( McuInitialized == false )
125    {
126  #if defined( USE_BOOTLOADER )
127      /* Set the Vector Table base location at 0x3000 */
128      NVIC_SetVectorTable( NVIC_VectTab_FLASH, 0x3000 );
129  #endif
130
131      HW_AdcInit( );
132
133      Radio.IoInit( );
134
135      HW_SPI_Init( );
136
137      HW_RTC_Init( );
138
139      vcom_Init( );
140
141      BSP_sensor_Init( );
142
143      BSP_LED_Init( LED1 );
144      BSP_LED_Init( LED2 );
145      BSP_LED_Init( LED3 );
146      BSP_LED_Init( LED4 );
147
148      BSP_PB_Init(BUTTON_KEY,BUTTON_MODE_GPIO);
149
150      McuInitialized = true;
151    }
152  }
```



life.augmented

Hand on GPIO(4/5)

- 5. Back to main.c and add function to read input switch
  - 5.1 add “**unsigned char COUNT_NUMBER_SW;**” to be global variable.

```
unsigned char COUNT_NUMBER_SW;
```

  - 5.2 add function below to read input switch in USER CODE BEGIN and END 2

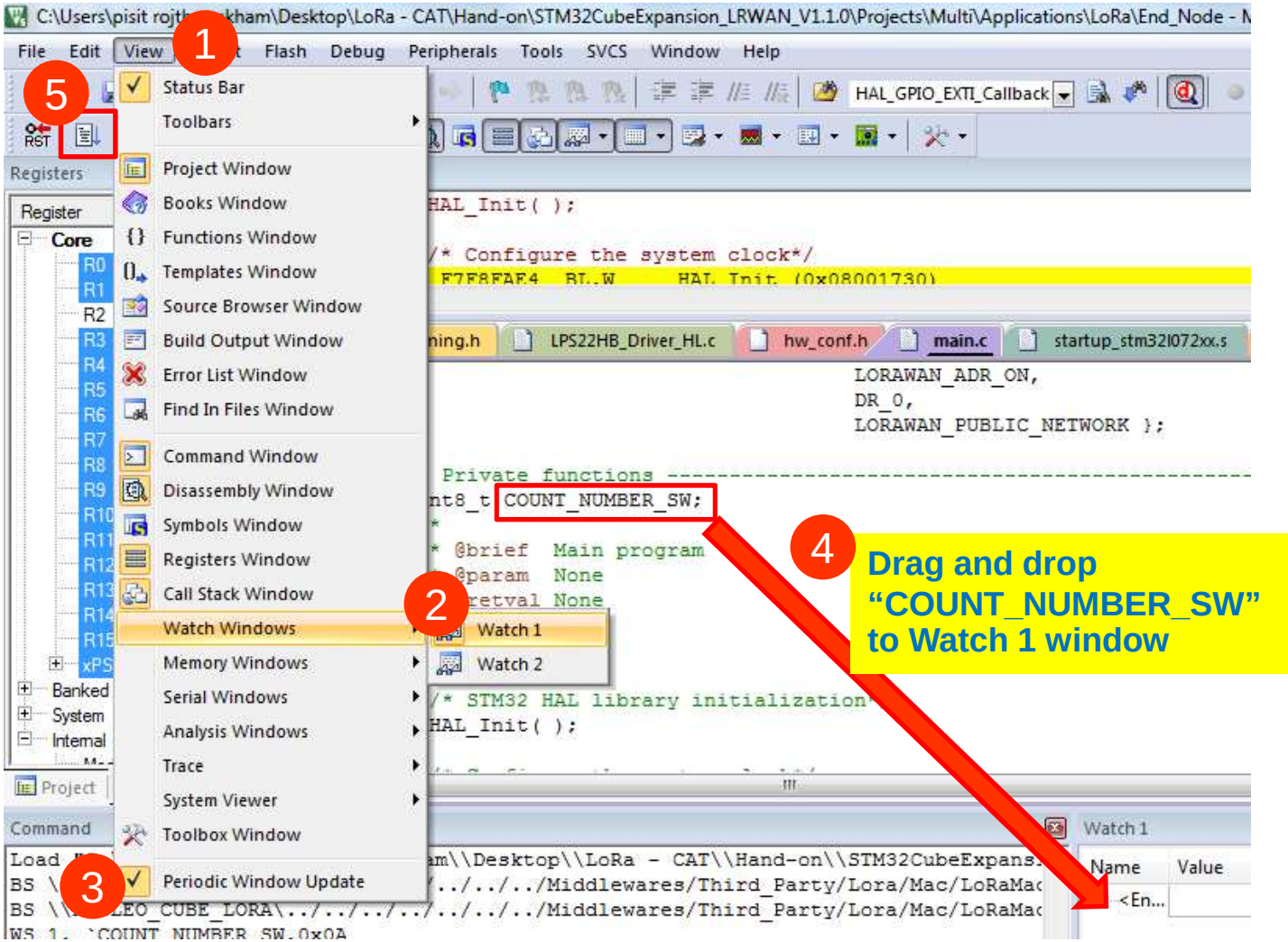
```
/* USER CODE BEGIN 2 */
  if(HAL_GPIO_ReadPin(????,????)==????){
    COUNT_NUMBER_SW++;
    HAL_Delay(250);
  }
/* USER CODE END 2 */
```

- 6.Replace all missing code “????" according to the comments
- 7. Rebuild all and debug code

Hand on GPIO(5/5)

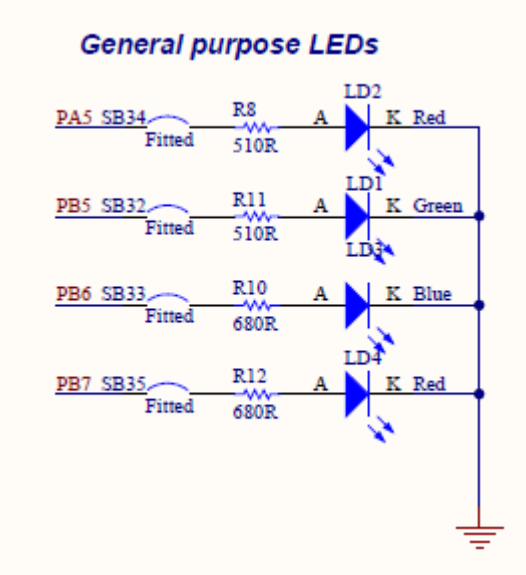
- 8. Add watching window, enable periodic update and run the code.

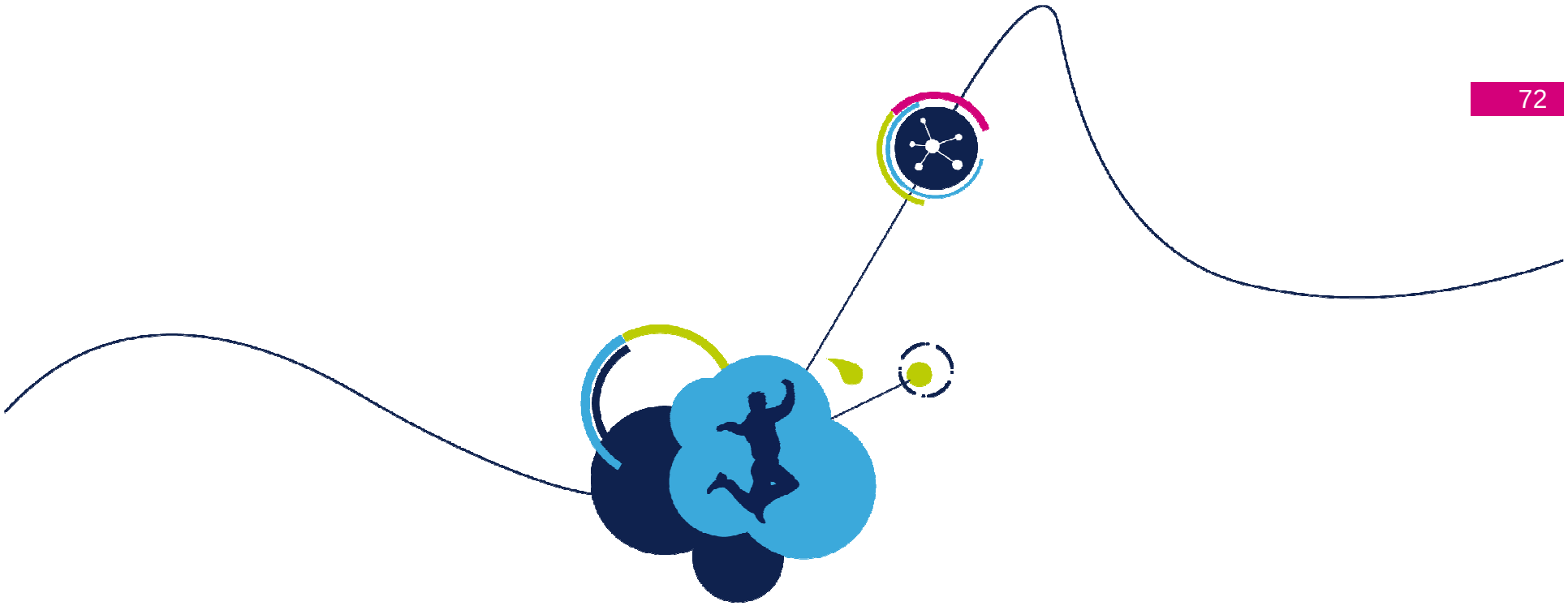
1. **Click View**
2. **Click Watch Windows -> Watch 1**
3. **Tick Periodic Window update**
4. **Drag and drop “COUNT_NUMBER_SW” to Watch 1 Window**
5. **Click RUN or F5 to run the code**
6. **Press the USER Button and see the COUNT_NUMBER_SW value**



Extra section

- Add Green LED to toggle when switch user button was pushed.





End of the hands-on