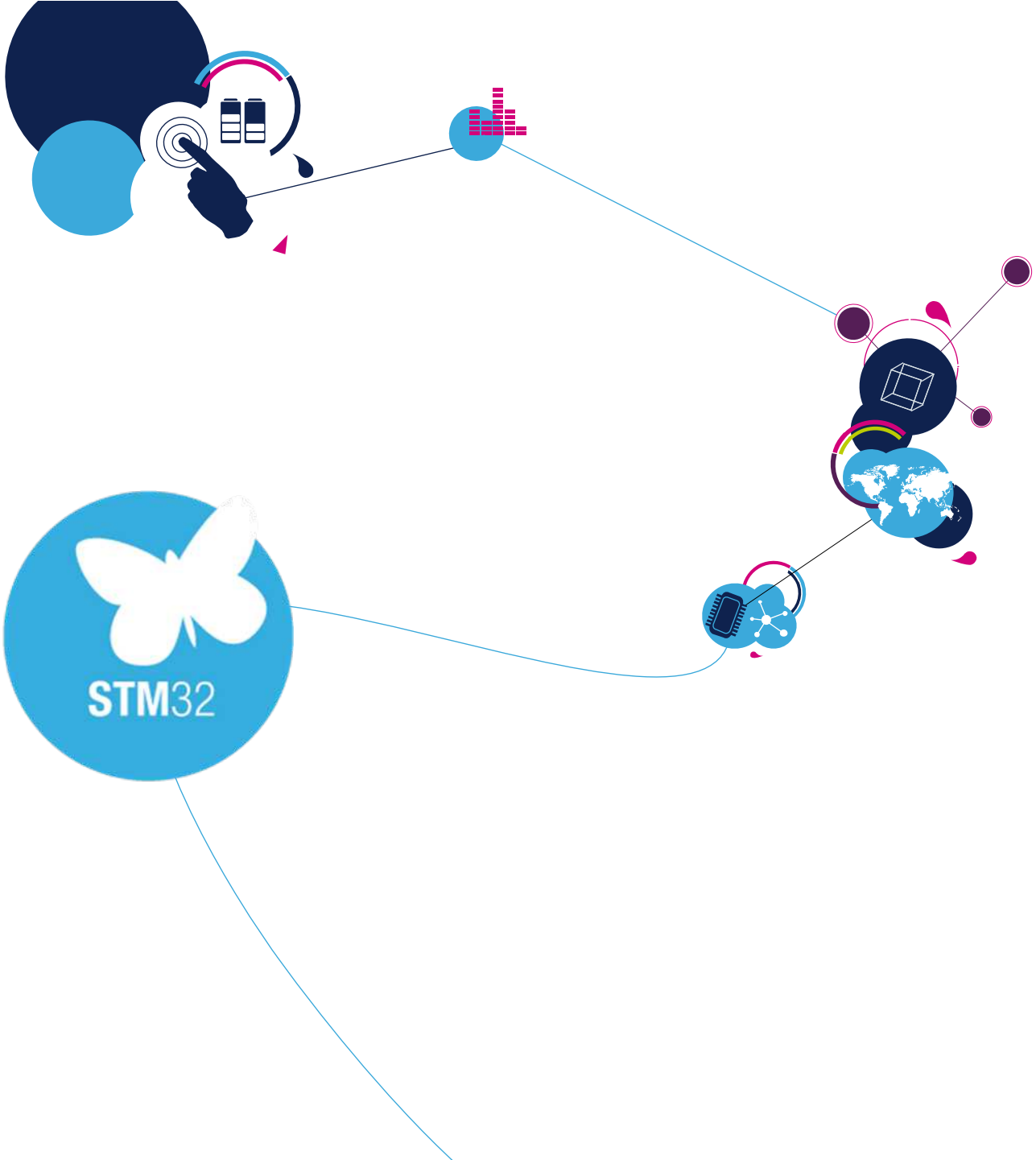
STM32 & IoT

M2PGI PM2M 2020



life.augmented

STM32 IoT 2020

Agenda

- Open development environment
- Board presentation
- Hands on Arduino:
  - Thermal Sensor
  - MEMs Sensor
  - Proximity Sensor
  - NFC
  - Bluetooth Low Energy with Blue NRG Application

life.augmented

STM32 Open Development Environment

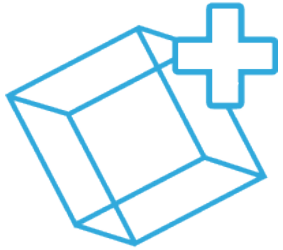
L’accélérateur de développement de produits IoT

life.augmented

STM32 Open Development Environment

Internet of Things

Smart Driving



Smart Things

Smart Home & City

Smart Industry

Smart Driving

ST, un contributeur clé de l'IoT

ST well positioned for this movement

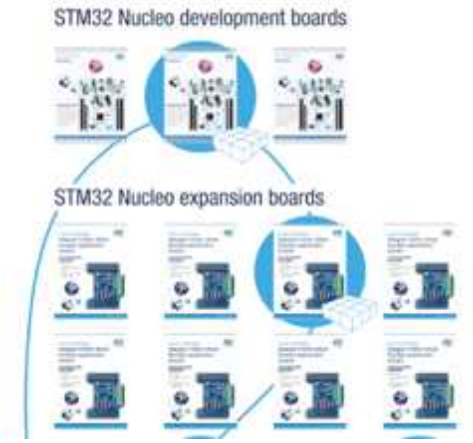
STM32 Open Development Environment

The right building blocks for IoT devices	Lower barriers for developers getting started	Lower barriers from prototyping to first product	Lower barriers to connect devices to the Cloud	Enable product & service commercialization
Building Blocks	**Stackable Boards & modular SW**	**Application specific SW**	**Integration of Cloud Provider SDKs**	**Partner Program**

Building Blocks

- Processing
- Security
- Sensing & Actuating
- Connectivity
- Motor Control
- Power & Energy Management
- Conditioning & Protection

Stackable Boards & modular SW



Application specific SW

- Audio Algorithms
- Sensor Fusion

Pre-integrated Software for vertical Applications

- Smart Home
- Building automation
- Wearable

Integration of Cloud Provider SDKs

- Microsoft Azure Certified
- IBM Watson IoT
- amazon web services Partner Network TECHNOLOGY PARTNER

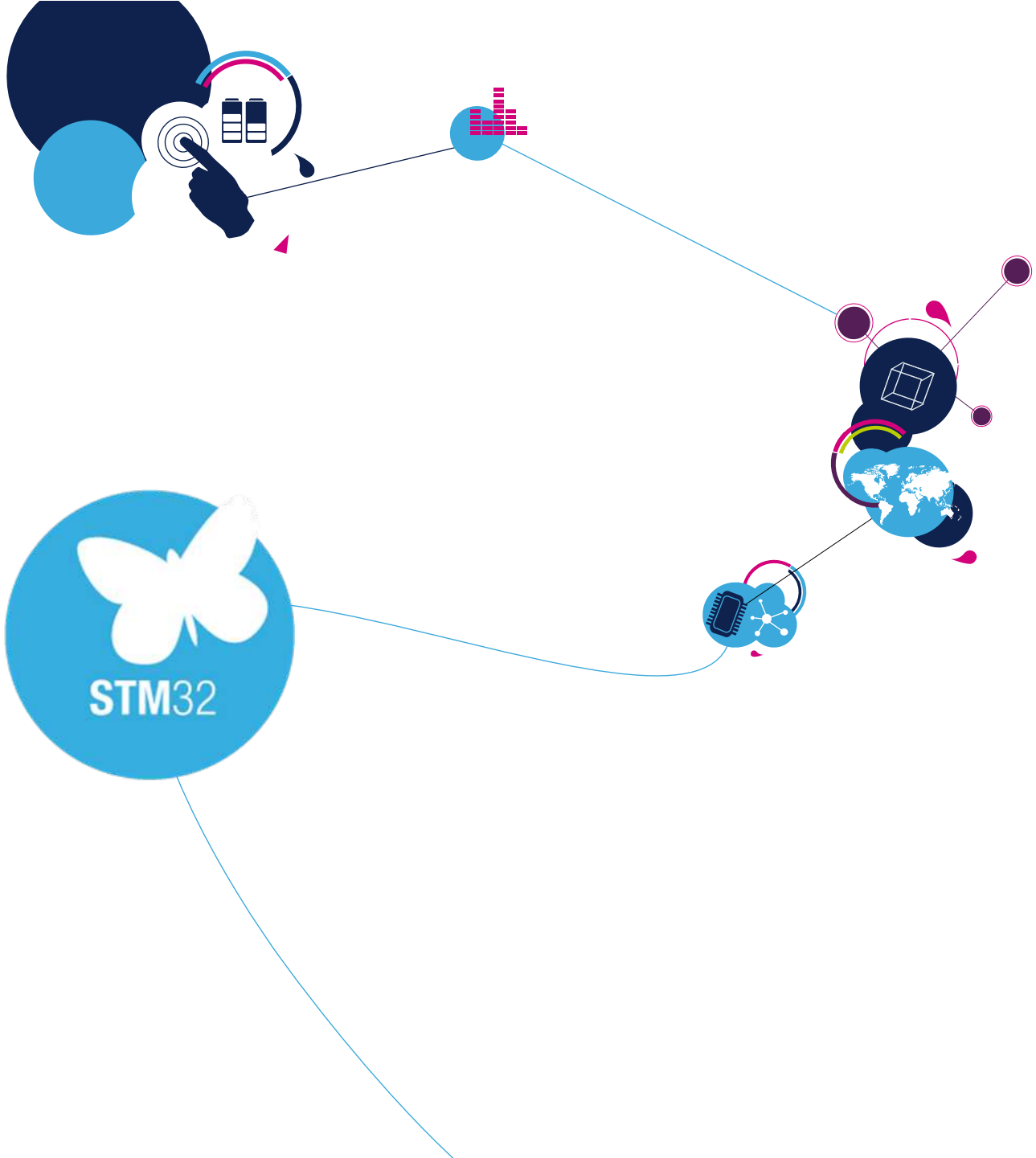
Ready to use Smartphone Apps

Partner Program

life.augmented

Partner Program

Board ecosystem



life.augmented

ST Solutions for the IoT

INTEGRATION

Common SW Platform

3 Cloud Provider SDKs supported, enabling sensor-to-cloud

131 SW packages from drivers to full application examples and Mobile Applications

STM32 Cube

STM32 Open Development Environment

Wearable | IoT Smart Things | Smart Home | Building automation

Modular Hardware

STM32

27 STM32 Nucleo development boards
Covering the broad portfolio of STM32 MCU families

36 STM32 Nucleo expansion boards (X-NUCLEO)
Offering peripheral functions

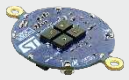
STM32 Nucleo

Form factor boards

ST & 3rd party form-factor boards


Cloud JAM


SensiBLE

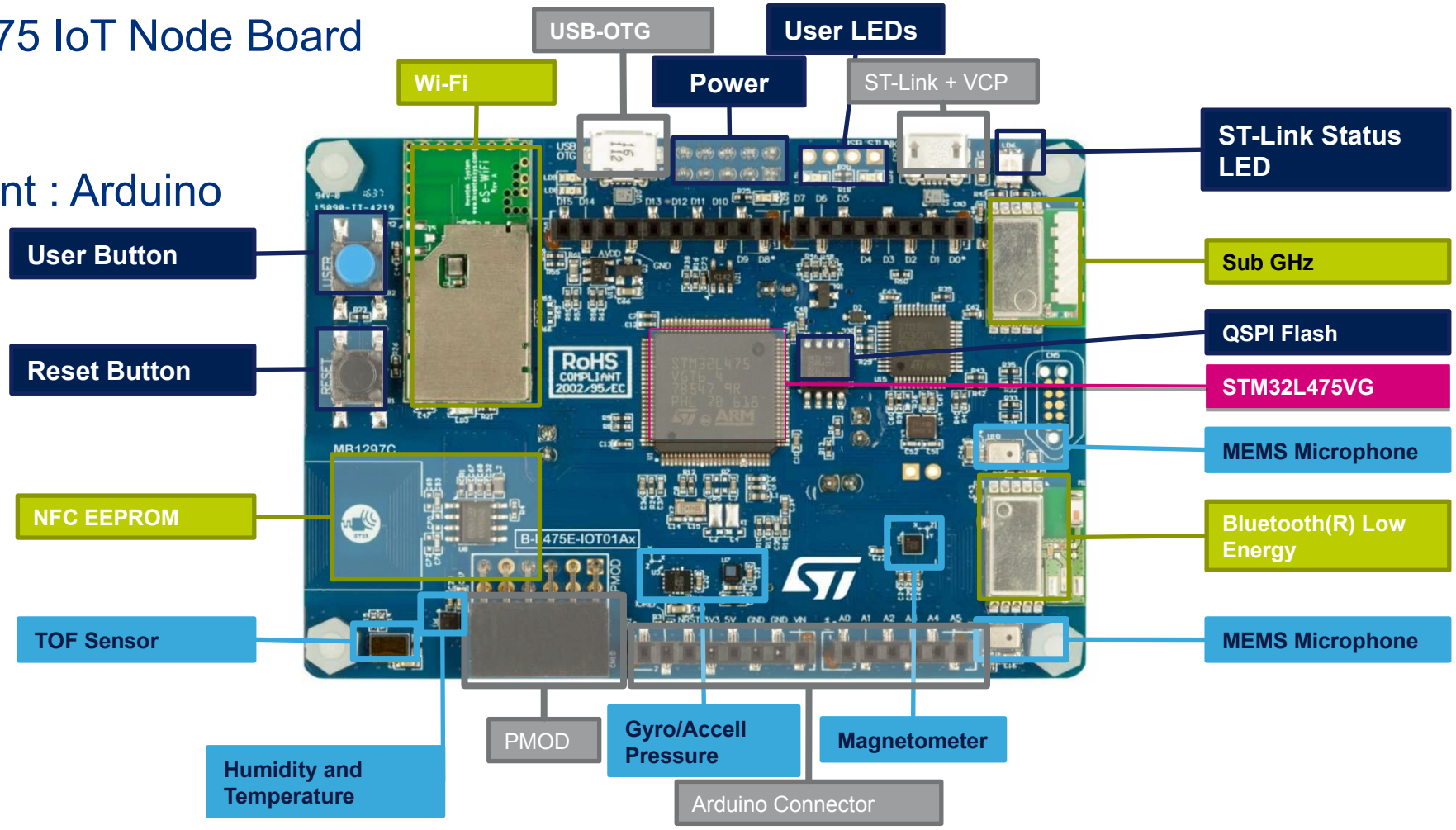

Bluecoin


IoT Discovery

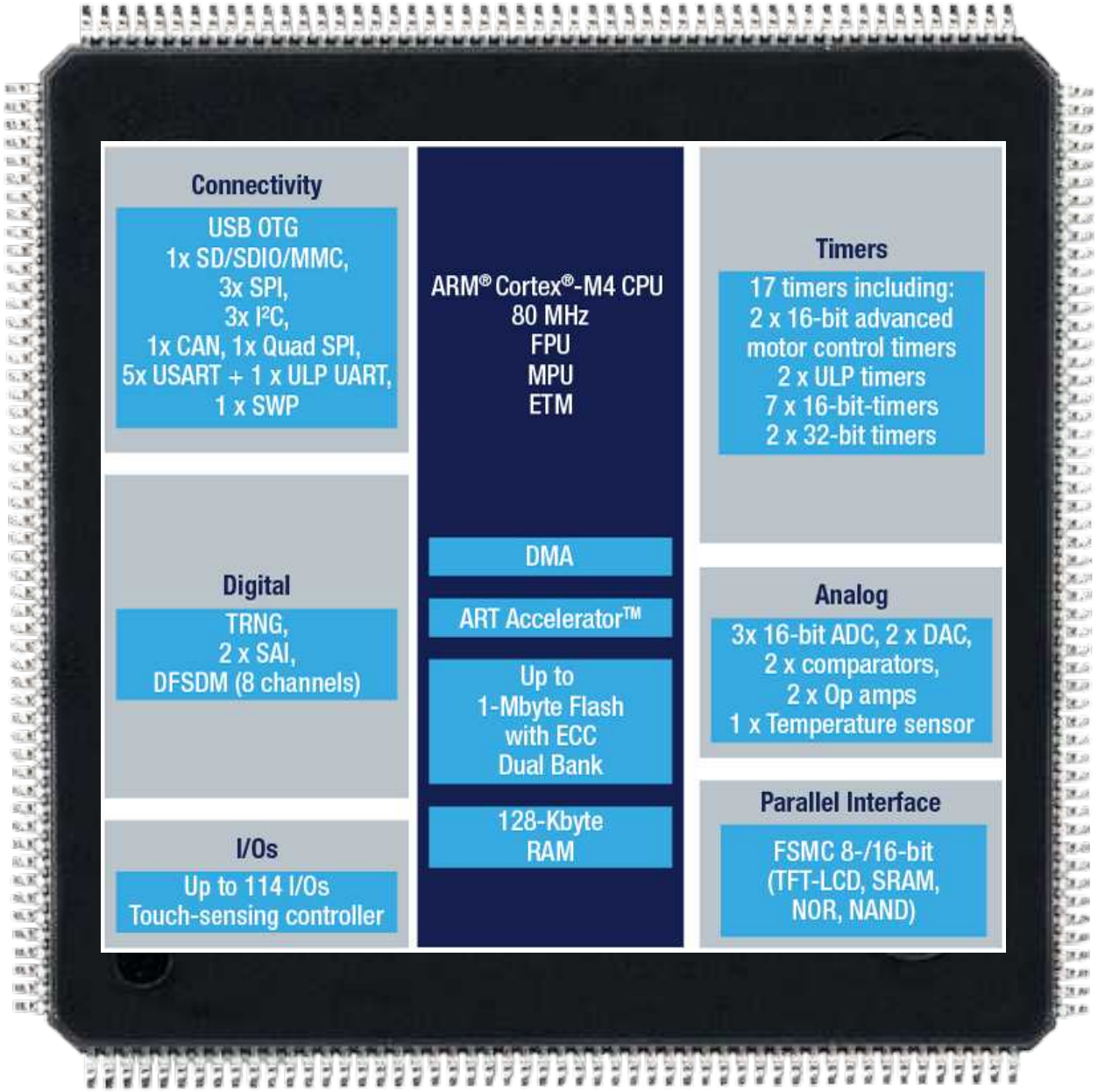

Sensor Tile

Discovery L475

- Board presentation
  - Board : L475 IoT Node Board
  - Environment : Arduino



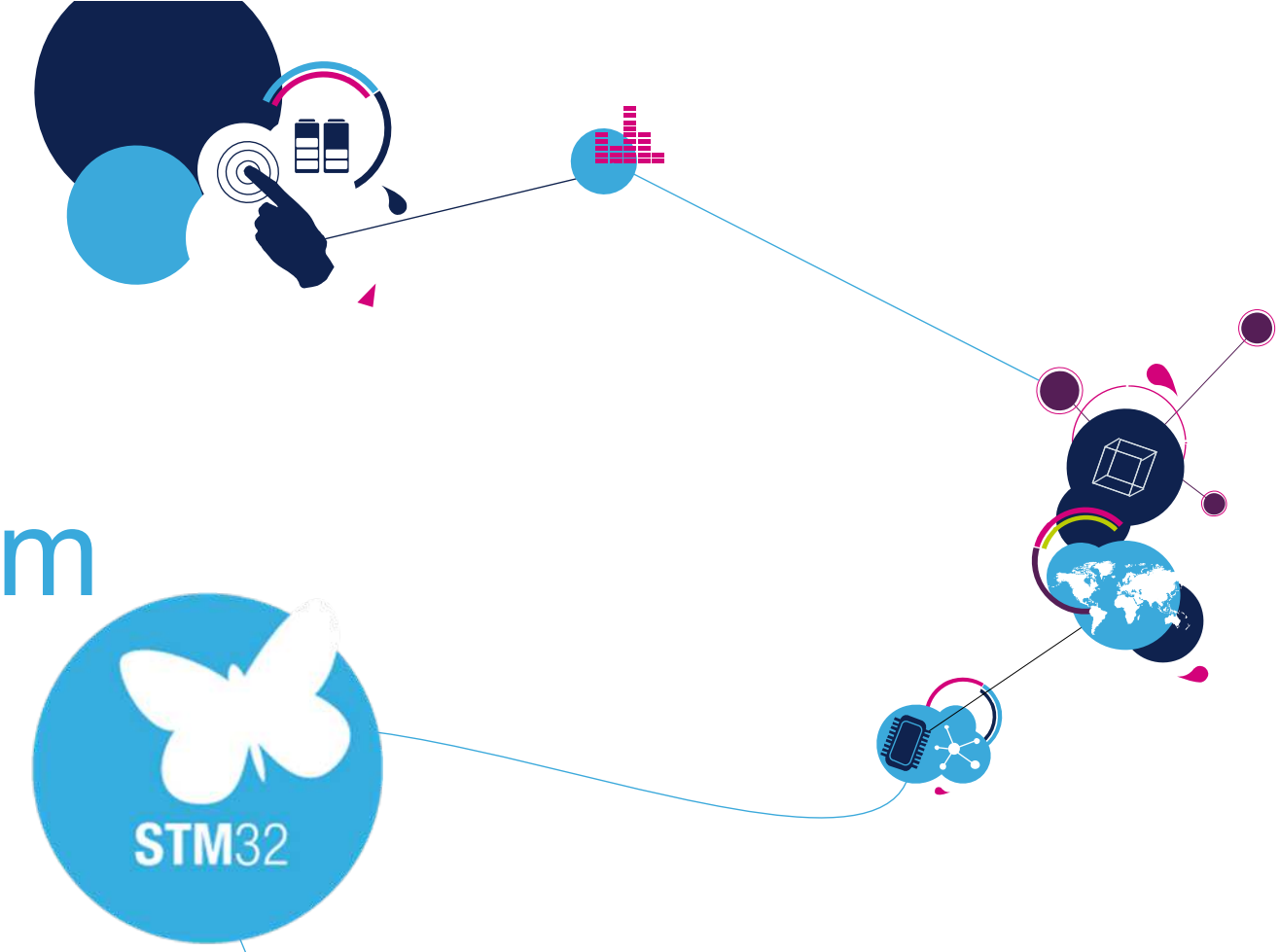
STM32L475



- STM32L4x devices are the ultra-low-power microcontrollers
  - 100 µA/MHz run mode
  - 1.1 µA Stop 2 mode,
  - 1.4 µA with RTC
  - 120 nA Standby mode
  - 30 nA Shutdown mode

life.augmented

Software ecosytem



life.augmented

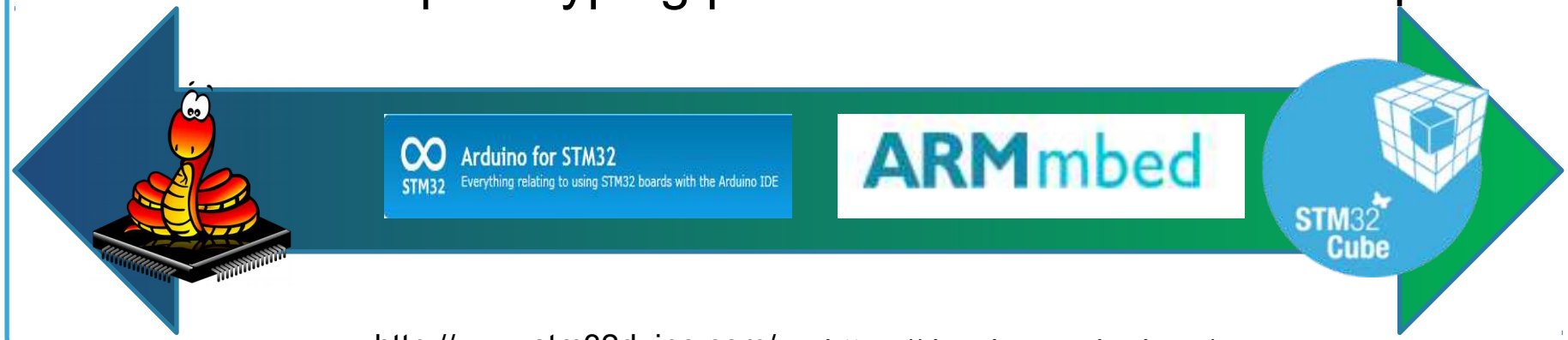
STM32 ODE, mbed and Arduino

same HW boards leveraging 3 ecosystems

SW & programming tools

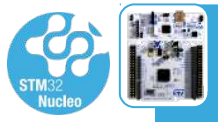
Ease of use / prototyping path

Industrialization path



http://www.stm32duino.com/ https://developer.mbed.org/ www.st.com/stm32cube

HW



STM32 Nucleo development boards

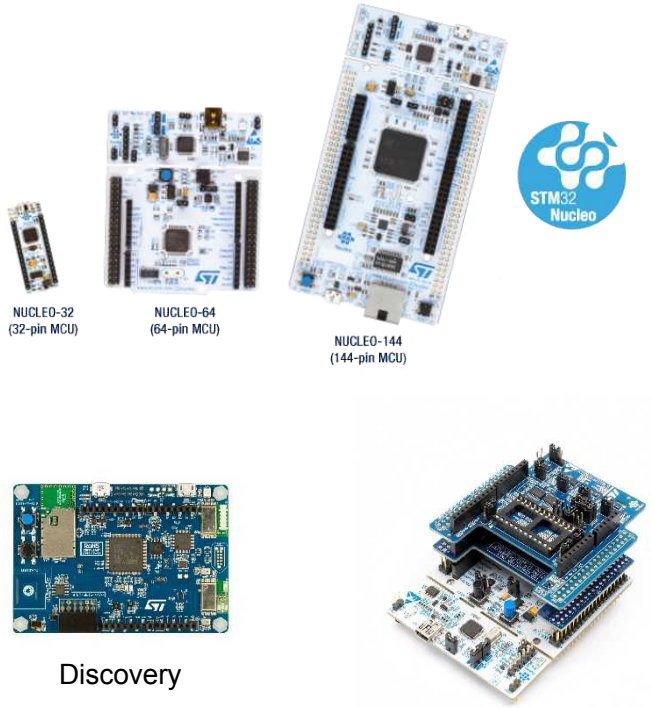
STM32 Nucleo expansion boards

Compatible integrated boards can be supported additionally (STEVAL, Discovery, 3rd party)

life.augmented

Arduino VS STM32

- Most of STM32 boards are compliant with Arduino Eco-system.
- Many libraries/examples are available.



Arduino ecosystem

- C like(C++) language
- Computer embedded IDE
- Beginner
- Most of STM32 board supported
- Arduino shield supported
- Arduino download: https://www.Arduino.cc

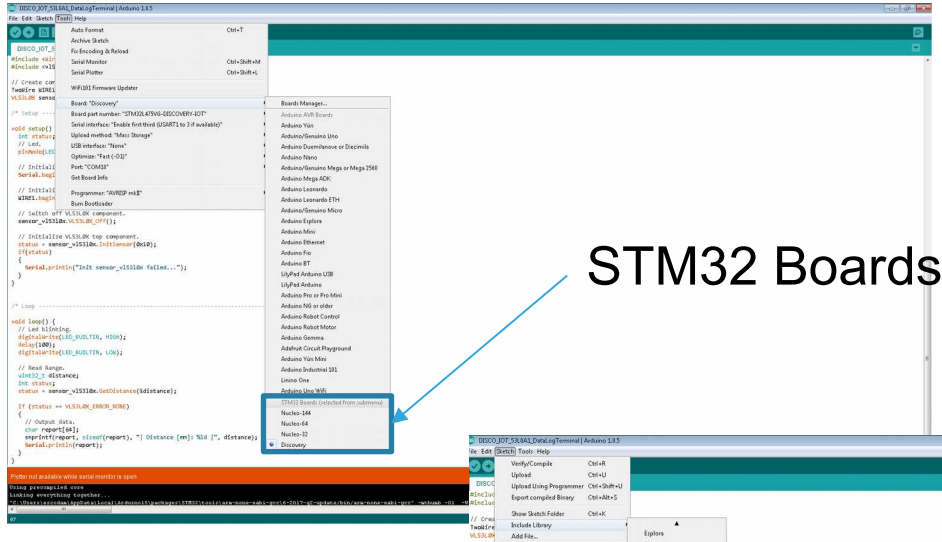
STM32 ODE on stm32duino

- Nucleo and Discovery boards supported
  - NUCLEO-L476RG, -F401RE, -F411RE, -L053R8, -L152RE, -F030R8, -F091RC, -F103RB, -F207ZG, -F303RE, -F429ZI, -L432KC
  - STM32VLDISCOVERY, STM32F407G-DISC1, STM32F746G-DISCOVERY, B-L072Z-LRWAN1 and **B-L475E-IOT01A**
- X-Nucleo boards supported
  - X-NUCLEO-IKS01A1 (+ LSM6DS3), X-NUCLEO-IKS01A2
  - X-NUCLEO-NFC01A1, X-NUCLEO-NFC04A1
  - X-NUCLEO-6180XA1, X-NUCLEO-53L0A1
  - X-NUCLEO-IDB05A1 (only Device Mode)
  - X-NUCLEO-LED61A1
  - X-NUCLEO-IKA01A1
  - X-NUCLEO-IHM02A1

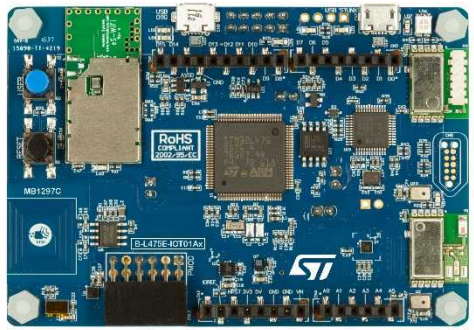
Ajouter des cartes STM32 Arduino dans le « Board Manager » :
https://raw.githubusercontent.com/stm32duino/BoardManagerFiles/master/STM32/package_stm_index.json

Where you can find software, help & libraries
https://www.Arduino.cc
https://www.stm32duino.com/
https://github.com/stm32duino/Arduino_Core_STM32

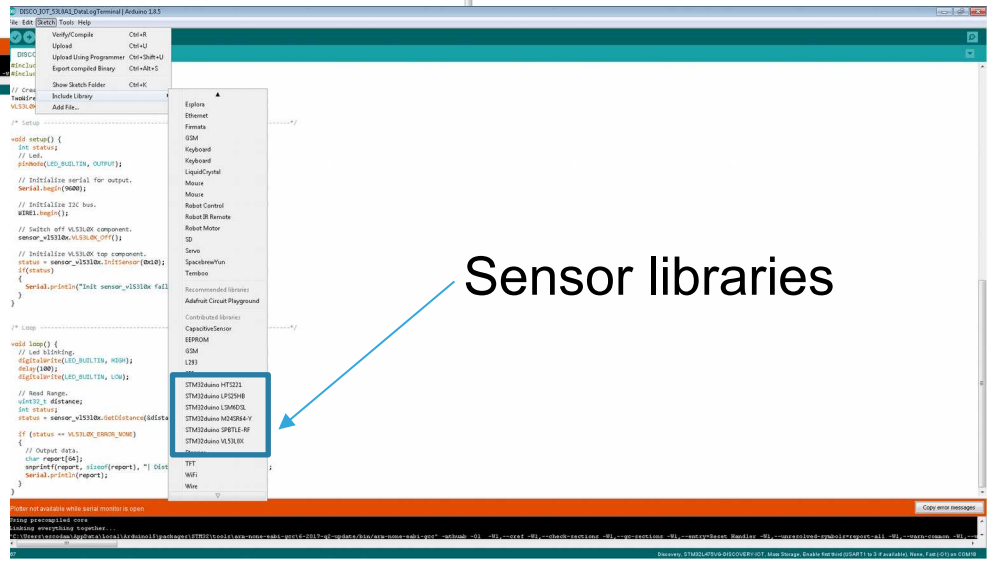
Arduino for STM32



STM32 Boards



B-L475E-IOT01AX board ready for IoT



Sensor libraries

STM32 DUINO

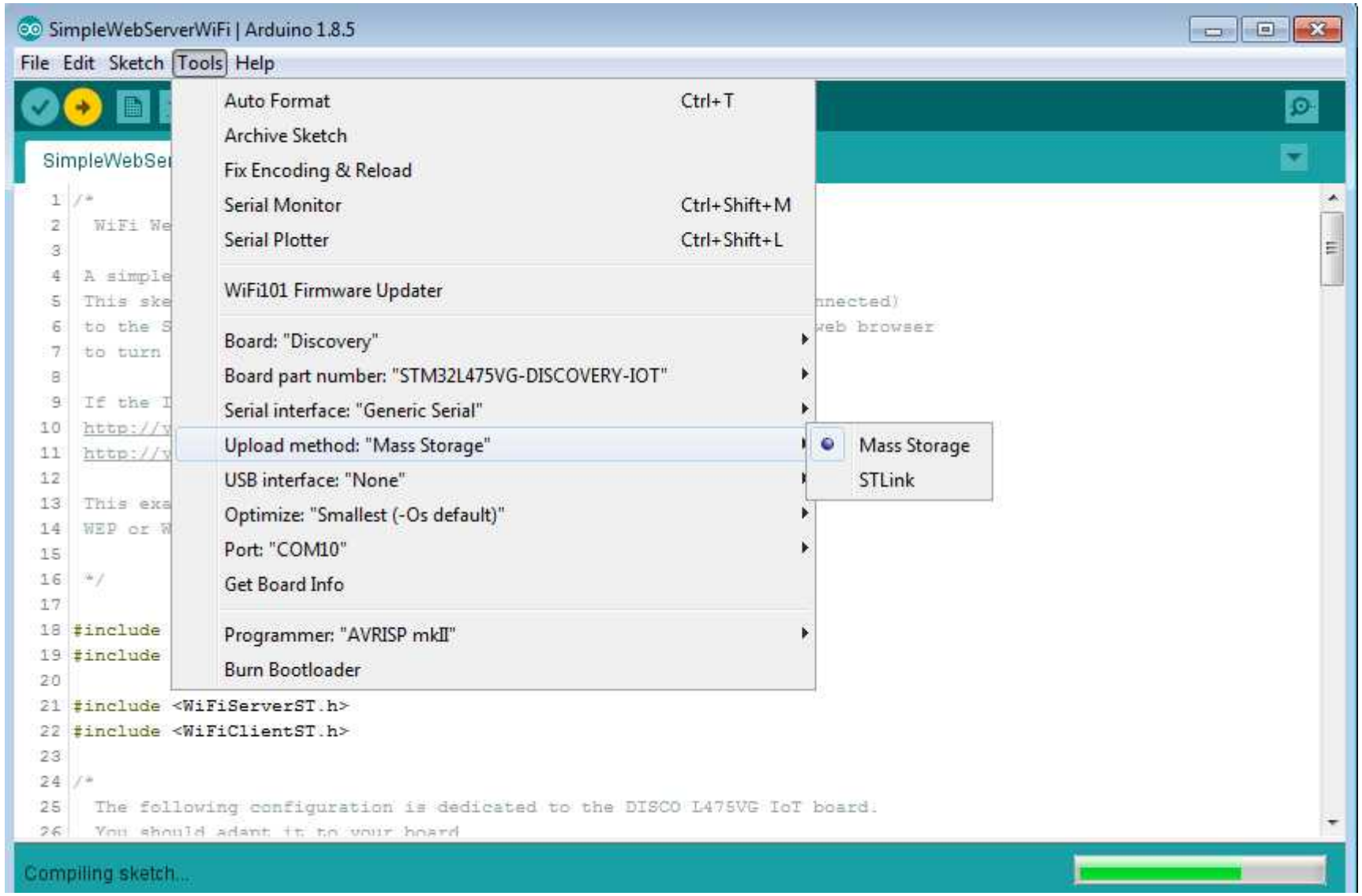
life.augmented

STM32

Arduino IDE

- Installation de la dernière version de l'IDE Arduino disponible ici : https://www.arduino.cc/en/main/software (Linux/Windows)
- Rajouter le support des cartes Arduino STM32
  - https://raw.githubusercontent.com/stm32duino/BoardManagerFiles/master/STM32/package_stm_index.json
- Selectionner la Nucleo L475IoT dans le « Board Manager »
- Selectionner l'interface UART dans Tools->Port
  - Sur Mac, on trouve /dev/tty.usbmodem-1511
  - Sur Windows, un nouveau COM port.. Par example, COM5
  - Sur Linux, une entrée du style /dev/ttyACM0
- Methode de programmation (STLINK ou Mass storage)

Arduino IDE presentation



life.augmented

Arduino libraries

- Dans Sketch -> Include libraries -> Manage Libraries
- Rechercher avec le terme “STM32duino” et ajouter les librairies suivantes:
    - HTS221 (Humidity/Temperature)
    - ISM43362-M3G-L44 (Wifi)
    - LIS3MDL (3D magnetometre)
    - LPS22HB (Barometer)
    - LSM6DSL (Accelometer/Gyroscope)
    - M24SR64-Y (NFC/RFID)
    - SPBTLE-RF (BlueNRG)
    - VL53L0X (Time-Of-Flight)

Checkpoint #1
Libraries installation

Applications

- BlueNRG :

  App allows you to access all the sensor data directly from your mobile device via the Bluetooth® Low Energy protocol

- NFC Tools :

  App allows you to read and write NFC tags

Hands On Thermal sensor

HTS221 (Humidity/Temperature)

life.augmented

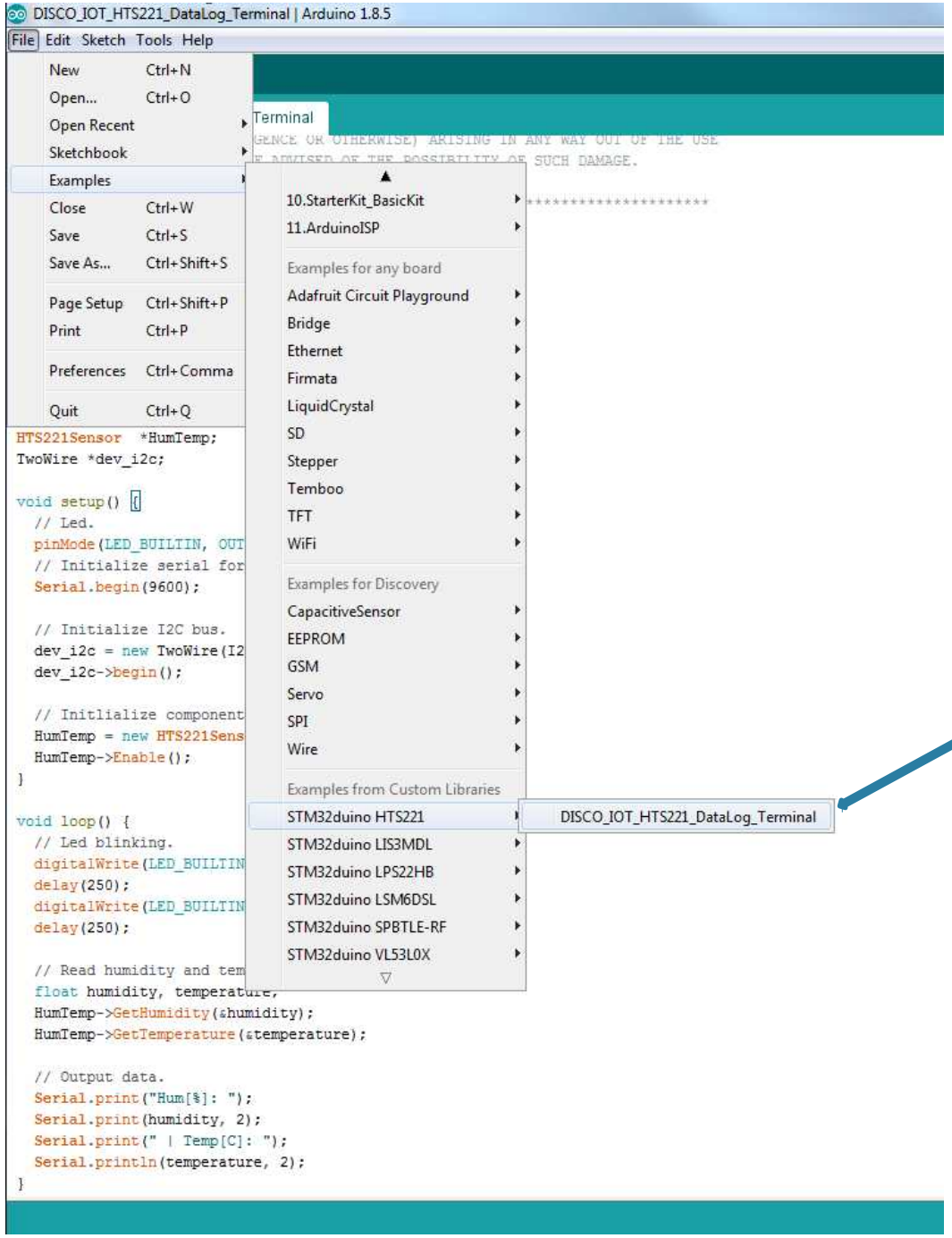
Thermal Sensor

- Objectif:
  - Lire les valeurs d'humidité et de température du HTS221 se trouvant sur la carte Discovery B-L475E-IOT01A IoT node
  - Utiliser l'UART pour récupérer ces valeurs et les afficher sur le terminal

- Outils:
  - ARDUINO IDE et serial monitor
  - Librairie « STM32duino_HTS221 »
  - Exemple " DISCO_IOT_HTS221_DataLog_Terminal "

Thermal Sensor

- Télécharger la librairie « STM32duino_HTS221 »
- Ouvrir l'exemple « DISCO_IOT_HTS221_DataLog_Terminal »



life.augmented

Thermal Sensor

DISCO_IOT_HTS221_DataLog_Terminal | Arduino 1.8.5

File Edit Sketch Tools Help

DISCO_IOT_HTS221_DataLog_Terminal

```
 * 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 <HTS221Sensor.h>

#define I2C2_SCL    PB10
#define I2C2_SDA    PB11

// Components.
HTS221Sensor  *HumTemp;
TwoWire *dev_i2c;

void setup() {
  // Led.
  pinMode(LED_BUILTIN, OUTPUT);
  // Initialize serial for output.
  Serial.begin(9600);

  // Initialize I2C bus.
  dev_i2c = new TwoWire(I2C2_SDA, I2C2_SCL);
  dev_i2c->begin();

  // Initlialize components.
  HumTemp = new HTS221Sensor (dev_i2c);
  HumTemp->Enable();
}

void loop() {
  // Led blinking.
  digitalWrite(LED_BUILTIN, HIGH);
  delay(250);
  digitalWrite(LED_BUILTIN, LOW);
  delay(250);

  // Read humidity and temperature.
  float humidity, temperature;
  HumTemp->GetHumidity(&humidity);
  HumTemp->GetTemperature(&temperature);

  // Output data.
  Serial.print("Hum[%]: ");
  Serial.print(humidity, 2);
  Serial.print(" | Temp[C]: ");
  Serial.println(temperature, 2);
}
```

Librairie des fonctions du HTS221

Définition des broches du bus I2C

Instantiation des classes HTS221 et I2C

Début de la fonction d'initialisation

Initialisation du port en sortie pour la LED

Initialisation de l'UART a 9600 BAUD

Initialisation du bus I2C

Initialisation du HTS221

Fin de la fonction d'initialisation

Début de la fonction de la boucle principale

Clignotement de la LED

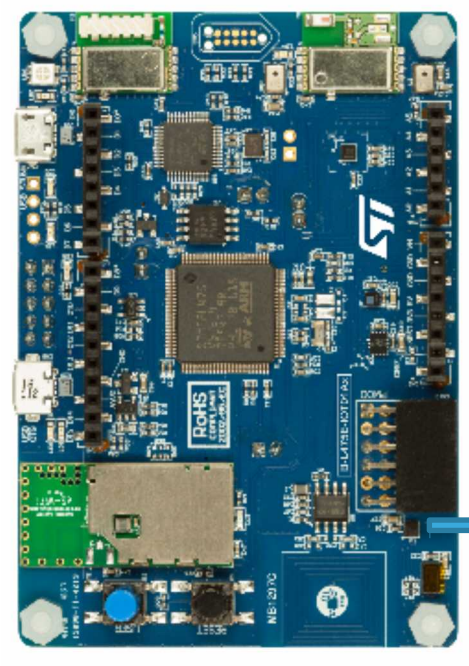
Lecture de l'humidité et de la temperature provenant du HTS221

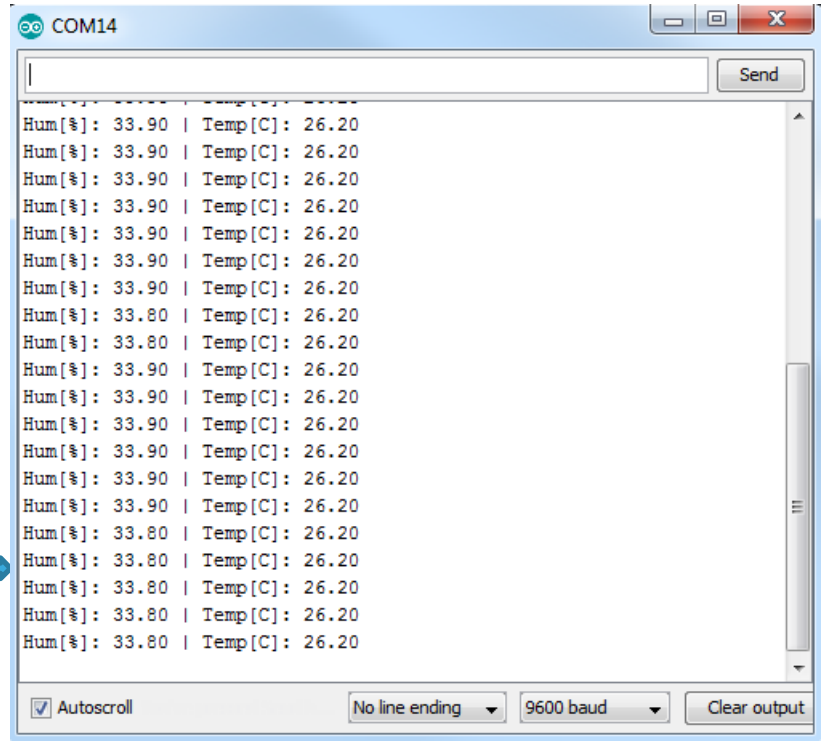
Ecriture de l'humidité et de la temperature sur le terminal via l'UART

Fin de la fonction de la boucle principale

Thermal Sensor

- Ouvrir “serial monitor” pour espionner la lecture/ecriture

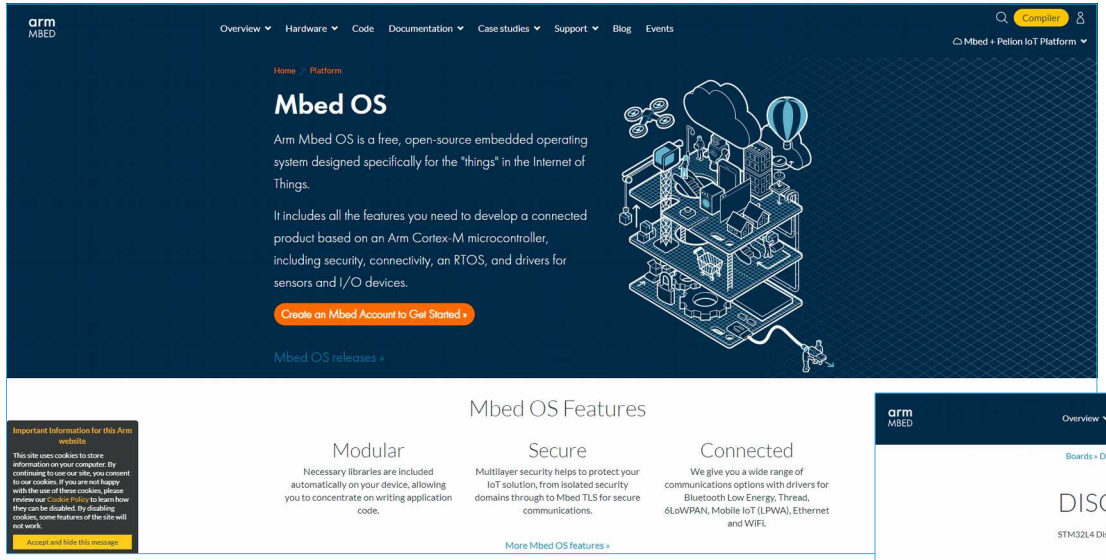


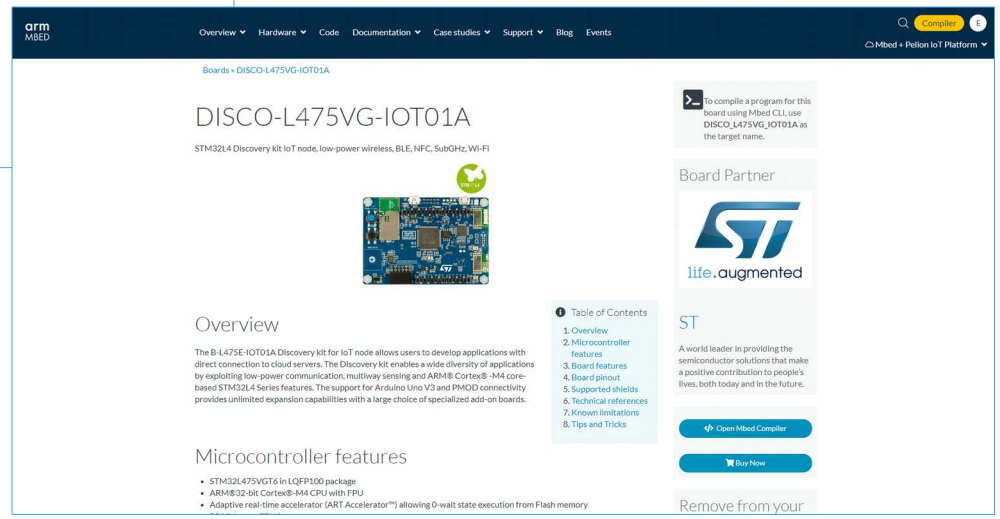


Checkpoint # 2
Thermal Sensor data

life.augmented

MBED STM32





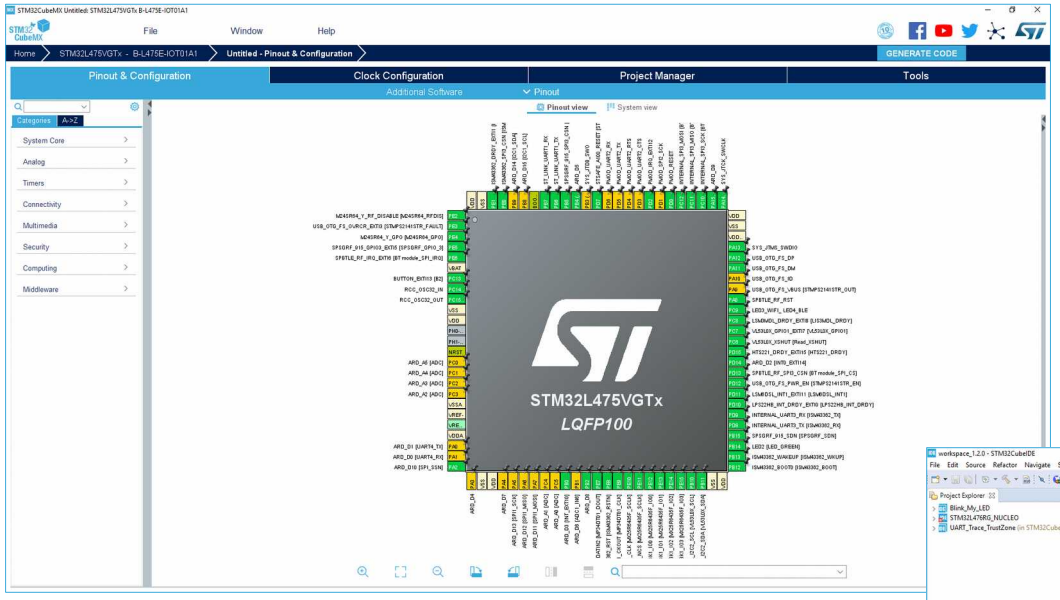
life.augmented

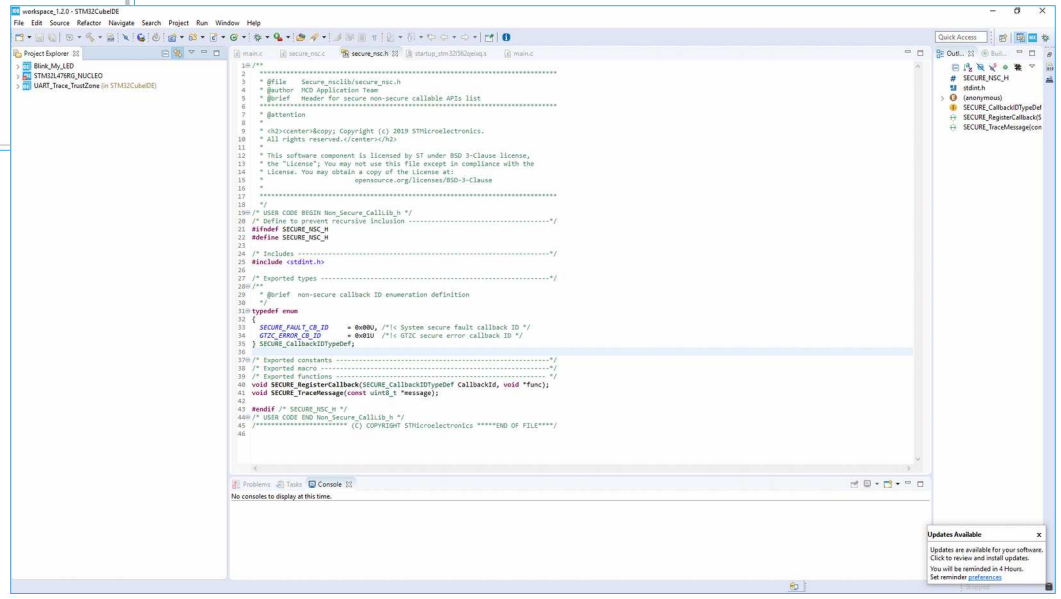
STM32

Mbed ecosystem

- C language
- Online IDE
- Online compiler
- Team/collab programmation
- Beginner/Pro ecosystem
- Most of STM32 board/products supported
- Most of Arduino shield supported
- Mbed OS: https://www.mbed.com/en/platform/mbed-os/

STM32Cube





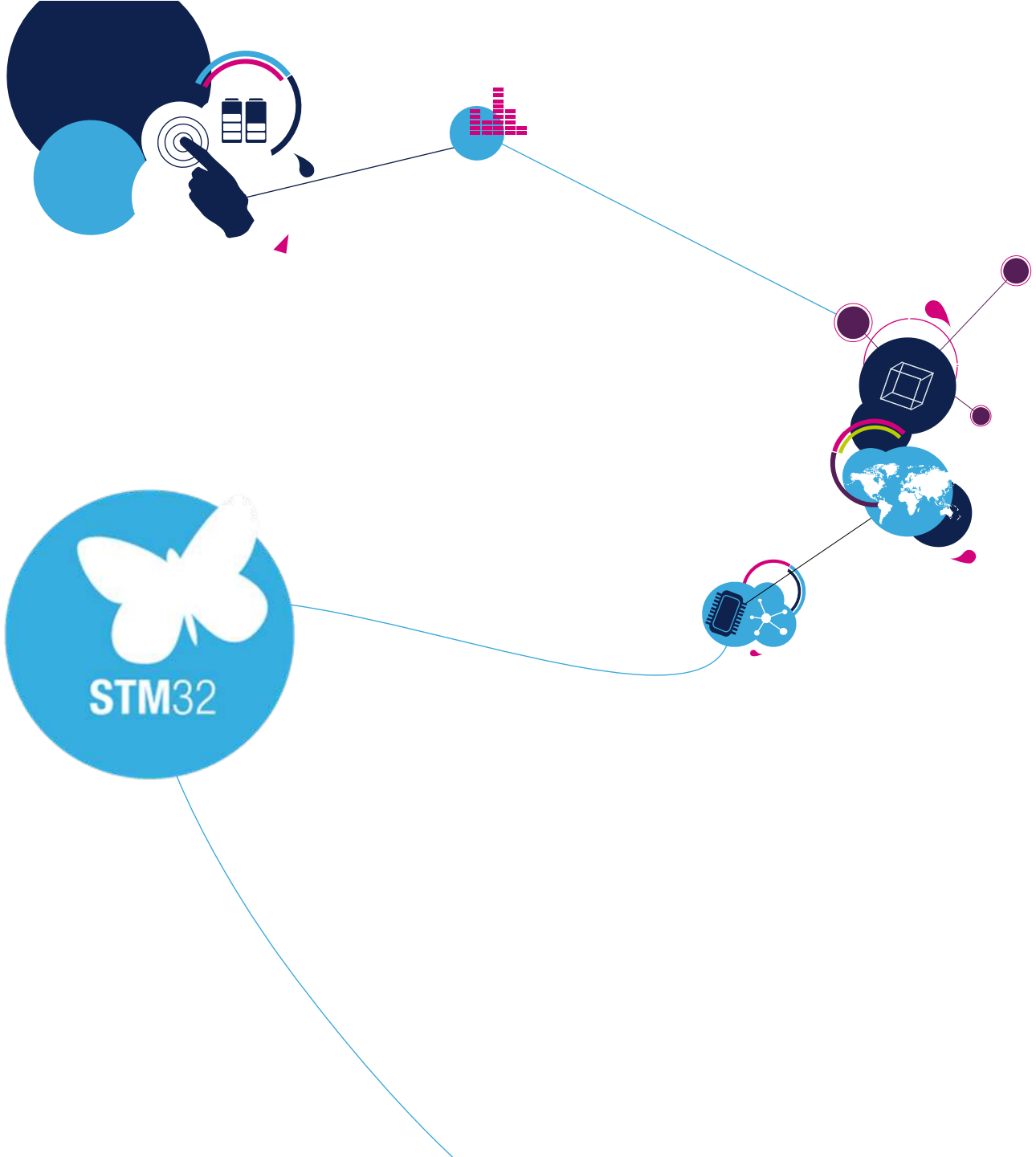
life.augmented

STM32

STM32Cube ecosystem

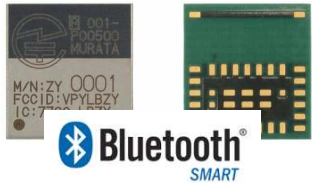
- C language
- Computer embedded IDE
- Graphic tool
- IDE supported IAR/KEIL/STM32CubeIDE(eclipse)
- Pro ecosystem
- All of STM32 board/products supported
- Most of STM32 shield supported
- HAL libraries support by STMicroelectronics
- Cloud integration
- STM32cube: https://www.st.com/content/st_com/en/stm32cube-ecosystem.html

RF vs STM32…



life.augmented

Technologie de communication



LAN

Short Range
Communicating Devices

Bluetooth 4.0 · WiFi

35% SOM

- ✓ **Well established standards**
- ✓ **Good for:**
  - Mobile devices
  - In-home
  - Short range
- ❑ **Not good:**
  - Battery life
  - Long range

LPWAN

Long Range w/ Battery
Internet of Objects

SIGFOX — One network A billion dreams · LoRa™

55% SOM

- ✓ **Emerging PHY solutions**
- ✓ **Good for:**
  - Long range
  - Long battery
  - Low cost
  - Positioning
- ❑ **Not good:**
  - High data-rate

Cellular

Long Range w/Power
Traditional M2M

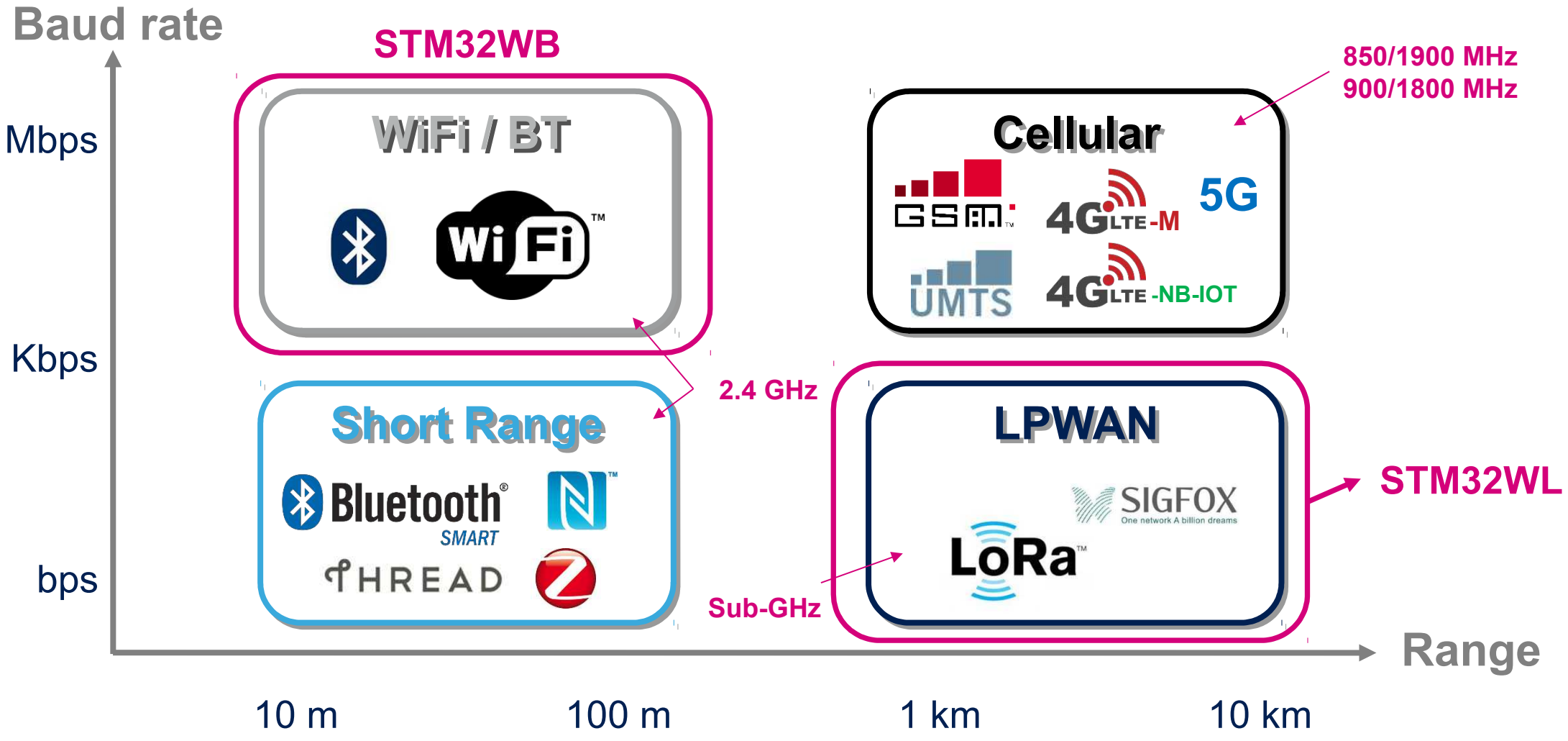
3G+ / H+ · GSM · 4G · 4G LTE -NB-IOT

10% SOM

- ✓ **Well established standards**
- ✓ **Good for:**
  - Long range
  - High data-rate
  - Coverage
- ❑ **Not good:**
  - Battery life
  - Cost

life.augmented

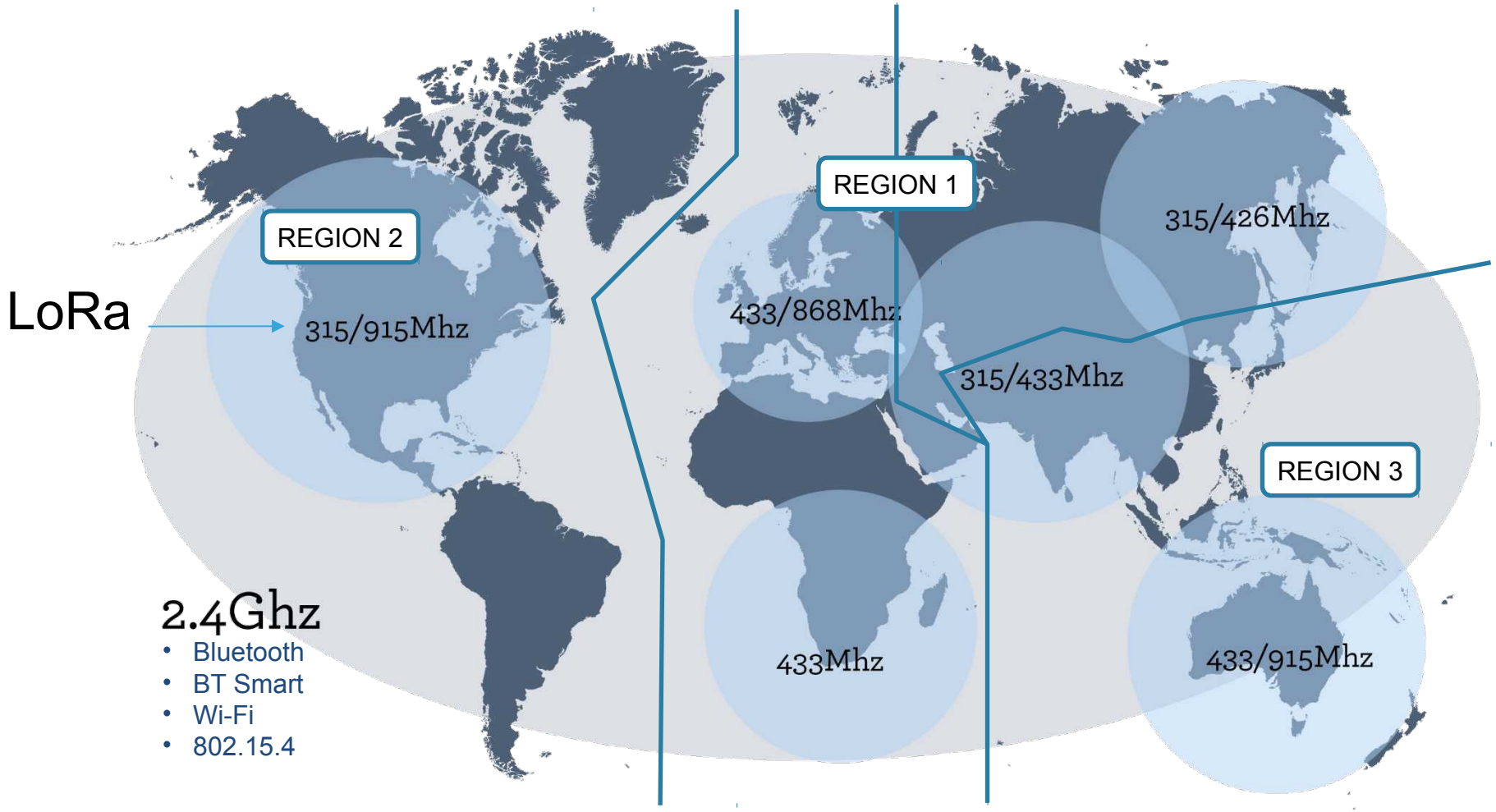
Communication Technologies - Overview



ST Confidential

World wide frequencies regulation

2.4GHz is the only ww standard

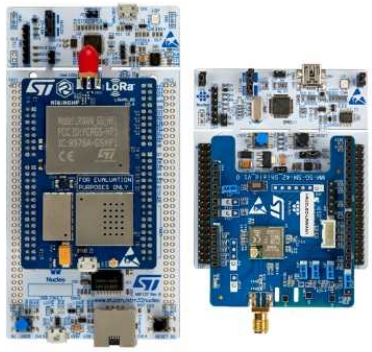


- GSM cell phone Frequencies are split in 2 regions only.
- **North America** and South west are based on **850/1900** MHz
- The rest of the **world** is using **900/1800MHz** frequencies range

life.augmented

STM32 Lora hardware

I-NUCLEO-SX1272D



P-NUCLEO-LRWAN2

I-NUCLEO-LRWAN1

Thank you

www.st.com/stm32