



ISIE4 – LOO C++

Introduction à MBED

Rappels

- C++ et embarqué
 - Réservé à des cibles pouvant le supporter
 - ARM Cortex A (SBC type rPi, BeagleBone...)
 - ARM Cortex M/R (MCU type STM32, LPC1768...)
- Sur SBC
 - Développement pour l'OS embarqué
 - Similaire, dans la forme, à du dev pour MCU (PC)
- Sur MCU
 - Développement Bare Metal ou avec micro noyau (RTOS en général)
 - Compilateur croisé spécifique ou GCC

Notre cas d'utilisation

■ Côté Hard :

- Cartes de développement équipées de MCU de la famille des Cortex M
 - LPC1768 ou Nucleo-L073RZ ou Nucleo-L476RG
 - Ou autre cible compatible MBED

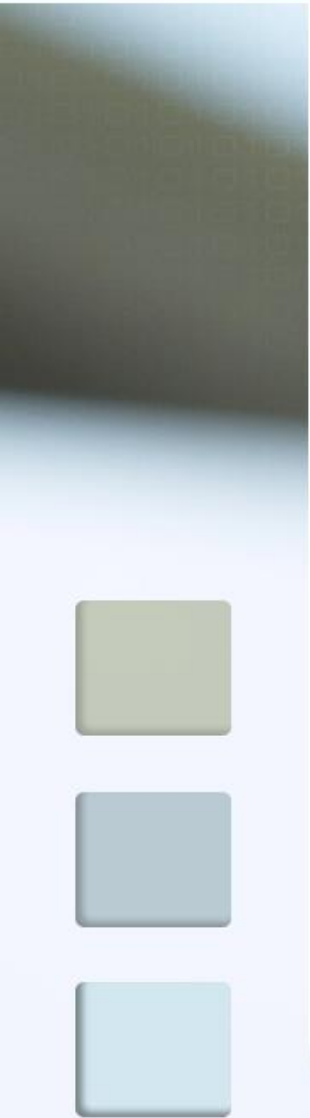
■ Côté Soft

- Développement avec MBED-OS
 - RTOS

Ce que l'on va évoquer

- La plateforme MBED
 - Qu'est-ce que c'est ?
 - A qui/quoi ça peut servir ?
- Deux cibles potentiellement utilisables
 - MBED-LPC1768
 - Nucleo-L073RZ
- Développer avec la plateforme MBED
- MBED OS
 - Points d'entrée principaux

La plateforme MBED



Introduction

- Ensemble d'outils
 - Soft + Hard
- Plusieurs situations envisageables
 - Prototypage rapide
 - Mise en production
- Souvent comparé à Arduino/Genuino
 - Rien à voir en réalité...

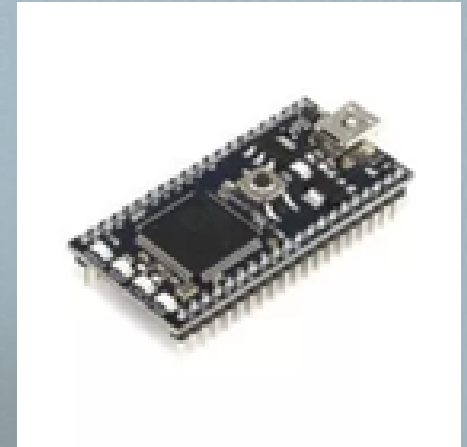
Prototypage rapide

- Carte embarquant le MCU et des ressources
 - Ecran LCD, interfaces COM...
- Système très autonome
- Exemples
 - BlueBoard-LPC11U24 (NGX/NXP)
 - u-blox C027 (ublox/NXP)
 - Cartes Nucleo (ST / ST)
 - FRDM-KL05Z (Freescale / Freescale)
 - Cartes « Gecko » (Silicon Labs / Silicon Labs)
 - RedBearLab nRF51822 (RedBearLab / Nordic SC)



Mise en production

- Module minimaliste, format C.I.
 - Installé sur une carte dédiée
- Exemples
 - mbed LPC1768 (NXP / NXP)
 - DipCortex Mx (SolderSplashLabs / NXP)
 - EA LPC11U35 QuickStart Board (Embedded Artists / NXP)
 - Modules Nucleo-32 (ST/ST)



Plateforme MBED

- Module MBED = 3 éléments
 - Le processeur – Fourni par le « Silicon Partner » (un fondeur : NXP,ST...)
 - La carte (équipée du MCU) – développée par le « Platform Partner » (uBlox, embedded artists...)
 - Certains fondeurs développent aussi des cartes, NXP par exemple
 - La compatibilité avec les outils MBED
 - Fournis par ARM
 - MBED SDK / MBED OS et MBED HDK

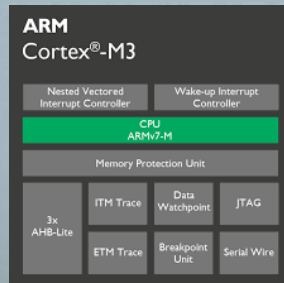
Les processeurs

- Processeurs ARM
 - IP Cores
- Facile et peu coûteux pour les fondeurs de « fabriquer » un MCU à base de cœur ARM
- Les modules MBED sont eux-mêmes
 - Peu coûteux
 - Puissants
 - Adaptables à énormément de situations



mbed LPC1768

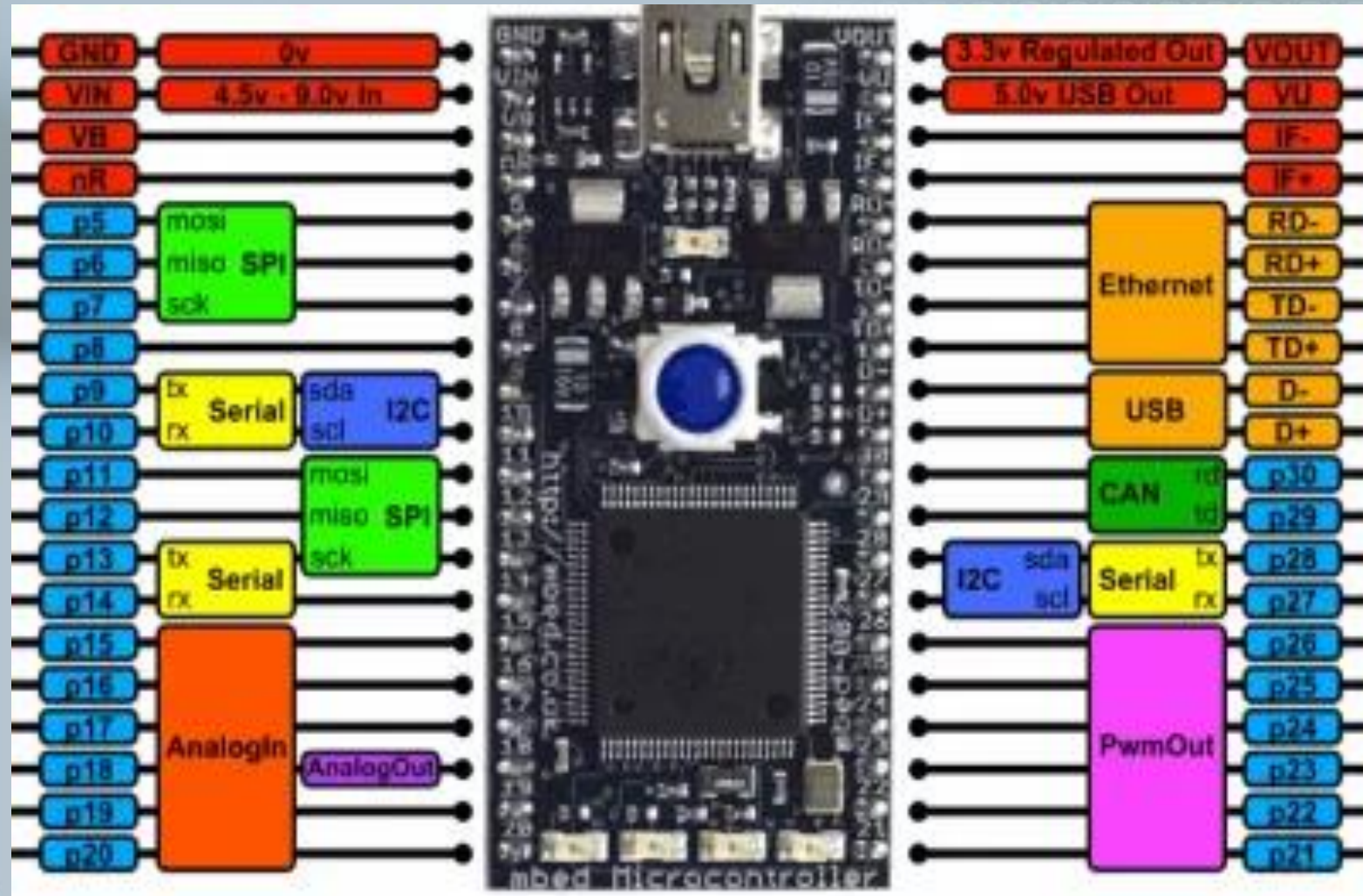
- Module utilisé en TP C++
- Totalelement développé par ARM
- Distribué autour de 50€
- MCU : NXP LPC1768
 - Bâti autour d'un ARM CORTEX-M3
 - Cadencé en interne à 96MHz
- Mémoire embarquée
 - 32ko de mémoire donnée (RAM)
 - 512ko de mémoire flash
 - Programme et data non volatile



Ressources LPC1768

- 26 GPIO max (multiplexage)
- 6 entrées analogiques (CAN 12 bits)
- 1 sortie analogique (CNA 10 bits)
- 6 sorties PWM
- 4 interfaces série asynchrones (UART)
- 1 Interface Ethernet
- 1 interface USB (Host / Device)
- 2 interfaces I2C
- 2 interfaces SPI
- 2 interfaces bus CAN

Le module LPC1768



Autres ressources / Infos

- Liste précédente = ressources directement exploitables via le SDK MBED
- Autres ressources disponibles
- Plus difficiles d'accès
 - Timers
 - DMA
 - RTC
 - Watchdog
 - ...
- Alimentation module : 4,5V à 14V
 - ***Le MCU lui-même est alimenté en 3,3V***
 - Attention /!\

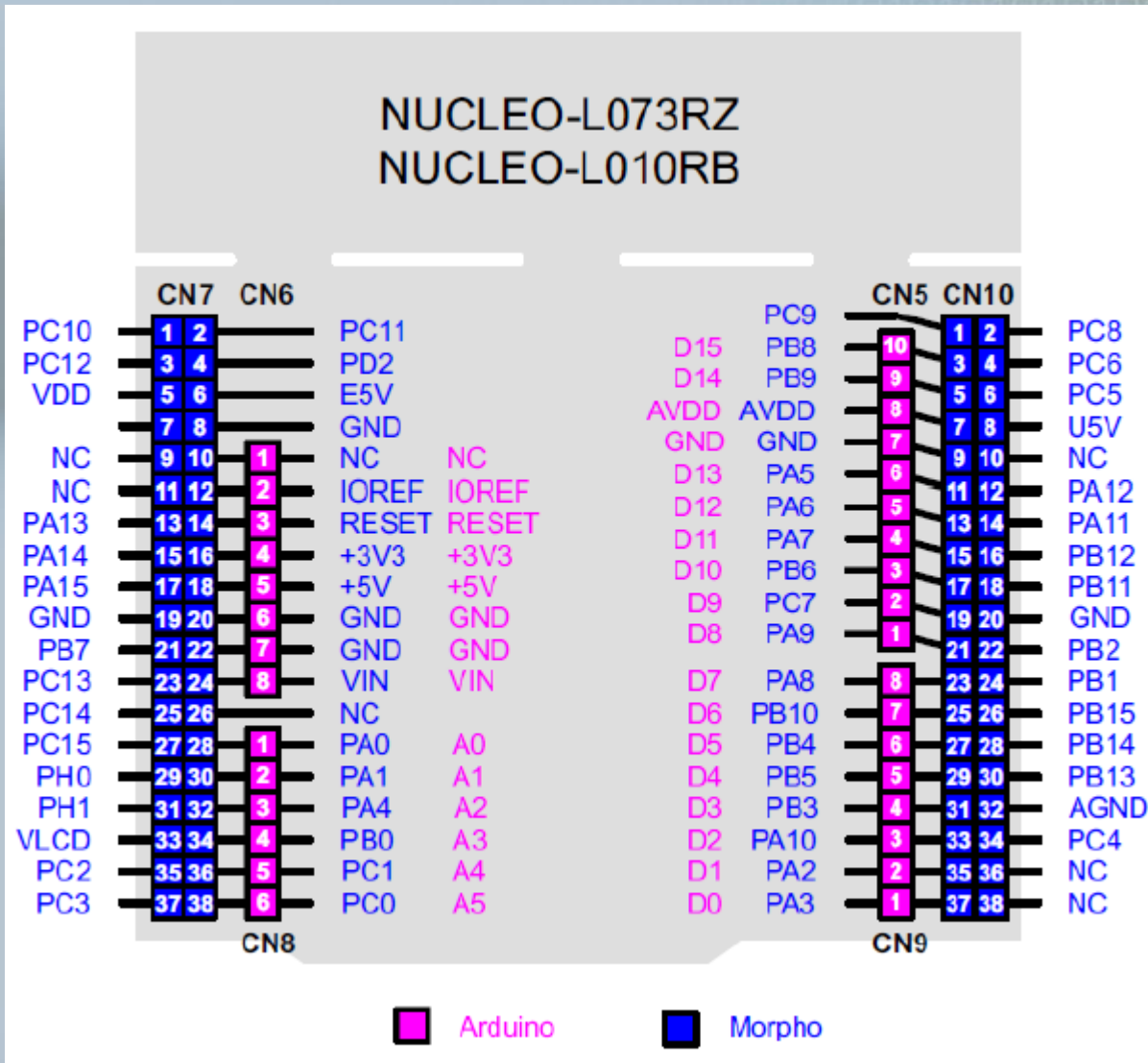
Carte Nucleo-L073RZ

- Module pouvant être utilisé pour la partie « projet » LOO C++
- Totalelement développé par ST
- Disponible pour environ 15€
- MCU : STM32L073RZT6
 - Bâti autour d'un ARM CORTEX-M0+
- Mémoire embarquée
 - 20ko de mémoire donnée (RAM)
 - 192ko de mémoire flash
 - Programme et data non volatile

Ressources Nucleo-L073RZ

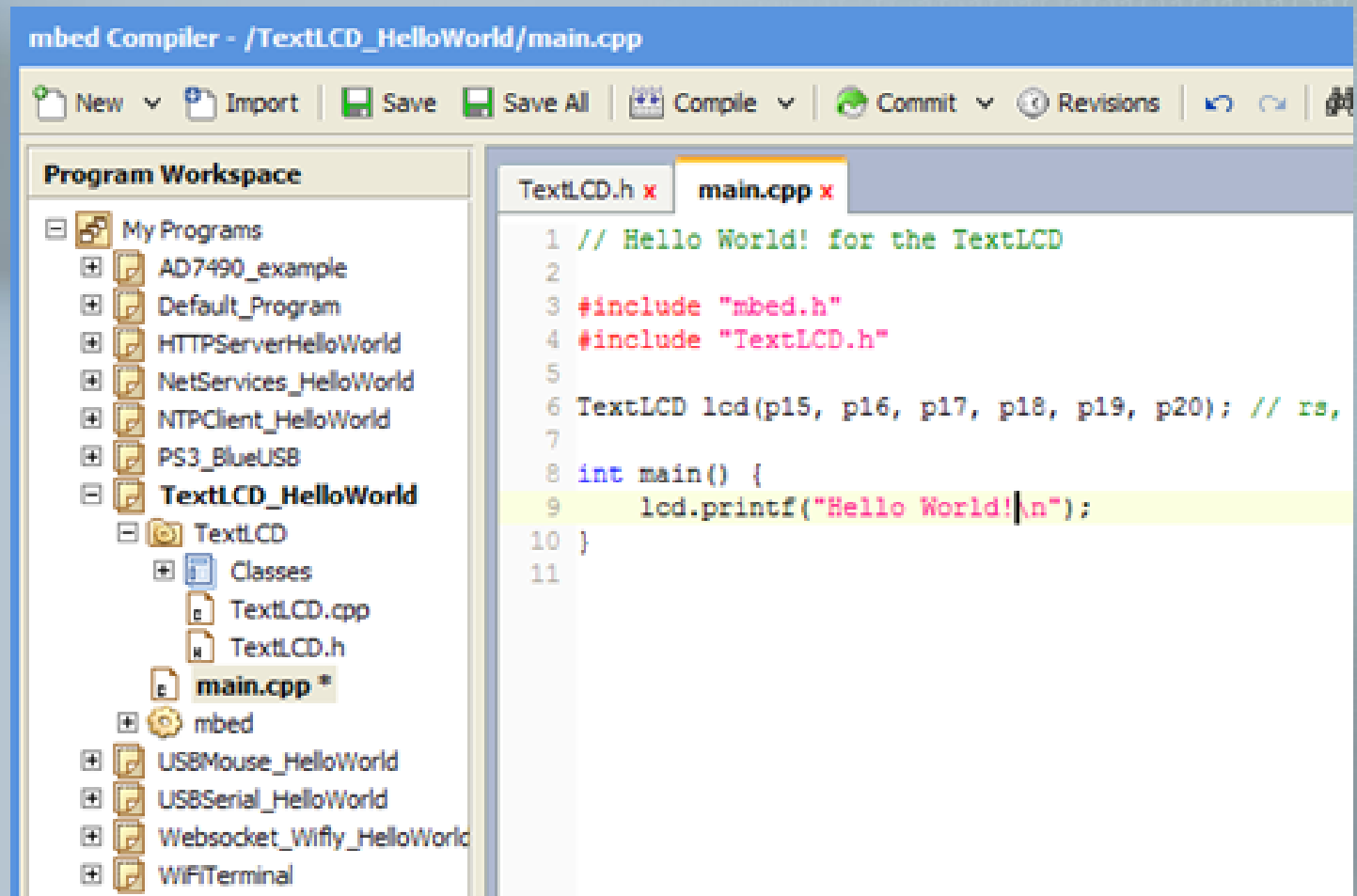
- 51 GPIO max (multiplexage)
- 16 entrées analogiques 12 bits
- 2 sorties analogiques 12 bits
- 2xSPI + 3xI2C + 4xUART
- Interface USB
- 2 Comparateurs analogiques
- Interface LCD
- Interface pour capteurs capacitifs

Nucleo-L073RZ



Développement sur module MBED


■ Via un IDE en ligne



Nombreuses bibliothèques

- Import « direct » dans le projet via l'EDI en ligne

Import Wizard

 **Import a library from mbed.org**
Select library from the list. You can also drag&drop them in your workspace.
[Click here](#) to import from URL.

Programs Libraries Bookmarked Upload

GPS Search

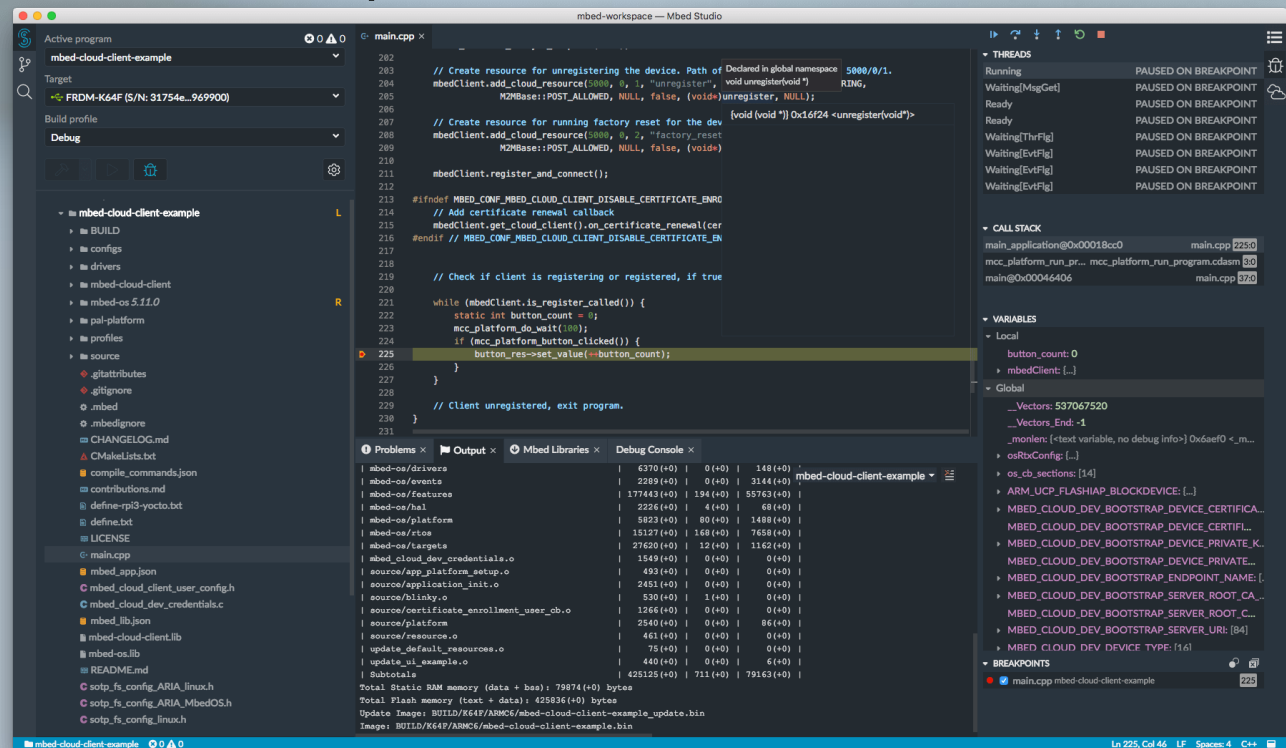
Listing published libraries on mbed.org matching "GPS"

Name	Tags	Author	Imports	Modified	Description
☆ GPS	EM406 GPS library Serial	Simon Ford	1161	08 Jun 2010	Library for the EM-406 GPS module
★ MODGPS	GPS	Andy Kirkham	1019	21 Apr 2011	Allows for a GPS module to be connected to a serial port and exposes an easy to use API to get the GPS data. New feature, added Mbed/LPC17xx RTC
☆ C027_Support	C027 u-blox	Team u-blox	678	25 Feb 2016	support library for C027 helper functions for Buffer Pipes, Buffered Serial Port (rtos capable) and GPS parsing. It includes modem APIs for USSD, SMS
☆ TinyGPS	GPS parse	Michael Shimniok	270	28 Apr 2011	Port of Arduino Tiny GPS library to mbed. Added extra methods to flag receipt/parsing of particular sentences.
☆ Mbed_Adafruit-GPS-Lib	Adafruit GPS library Ultimate	Myron Lee	256	22 Mar 2014	Adafruit Ultimate GPS Arduino library adapted for mBed use. Original found at https://github.com/adafruit/Adafruit-GPS-Library .
☆ SerialGPS	GPS Serial	Shinichiro Nakam	253	12 Oct 2010	Serial GPS .
☆ GPS		Tyler Weaver	191	23 Nov 2012	nmea gps library
☆ GPS	Adafruit GPS NEMA	Sam Clarke	186	19 Jan 2014	A basic library for the Adafruit Ultimate GPS module. (MTK3339) http://www.adafruit.com/products/746
☆ GPS		Sukkin Pang	143	27 Jul 2012	GPS for mbed
☆ GPS_CanSat		YST 2011	138	01 Jul 2012	Serial RX interrupt driving GPS (NMEA) library
☆ GPS		Keerthana Santhi	115	04 Feb 2011	
☆ GP2xx		Christopher Hasle	98	25 Nov 2010	library to return a range for sharp IR rangefinders. values are taken from the datasheet. note this is not accurate, but is good enough for seeing if thin

- Communauté très active

Nouveauté (2019)

- MBED Studio
 - Desktop IDE pour MBED
 - Windows / Mac



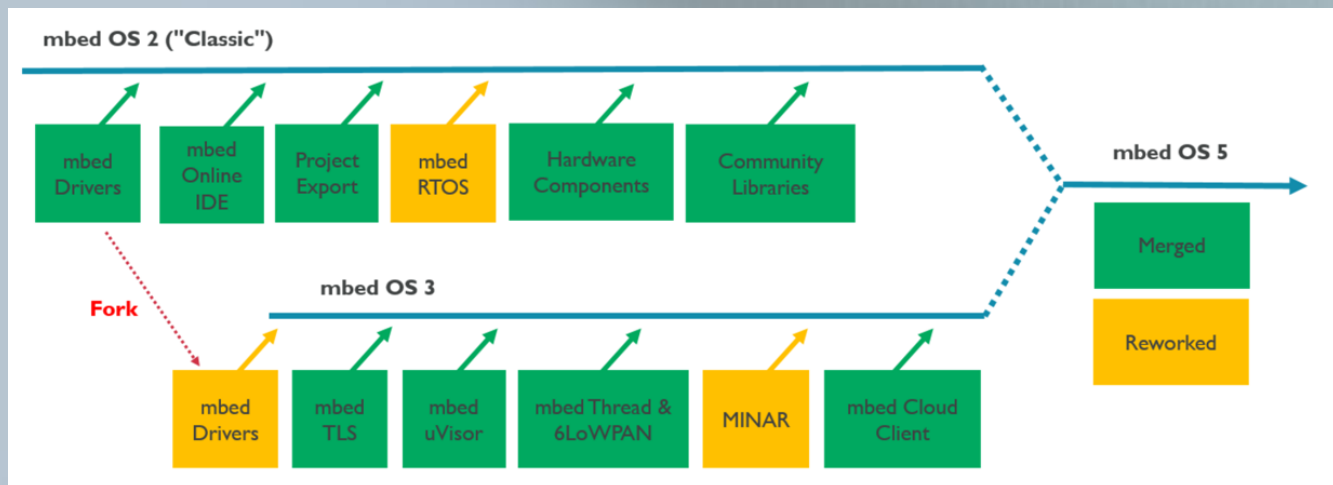
MBED-OS

- FKA Bibliothèque MBED
- A intégrer systématiquement pour chaque projet
 - Notion de micro kernel
- Contient l'ensemble des outils logiciels MBED
 - Le système (RTOS)
 - Les classes de gestion des périphériques
 - ...
- Ça c'était avant...

...et maintenant



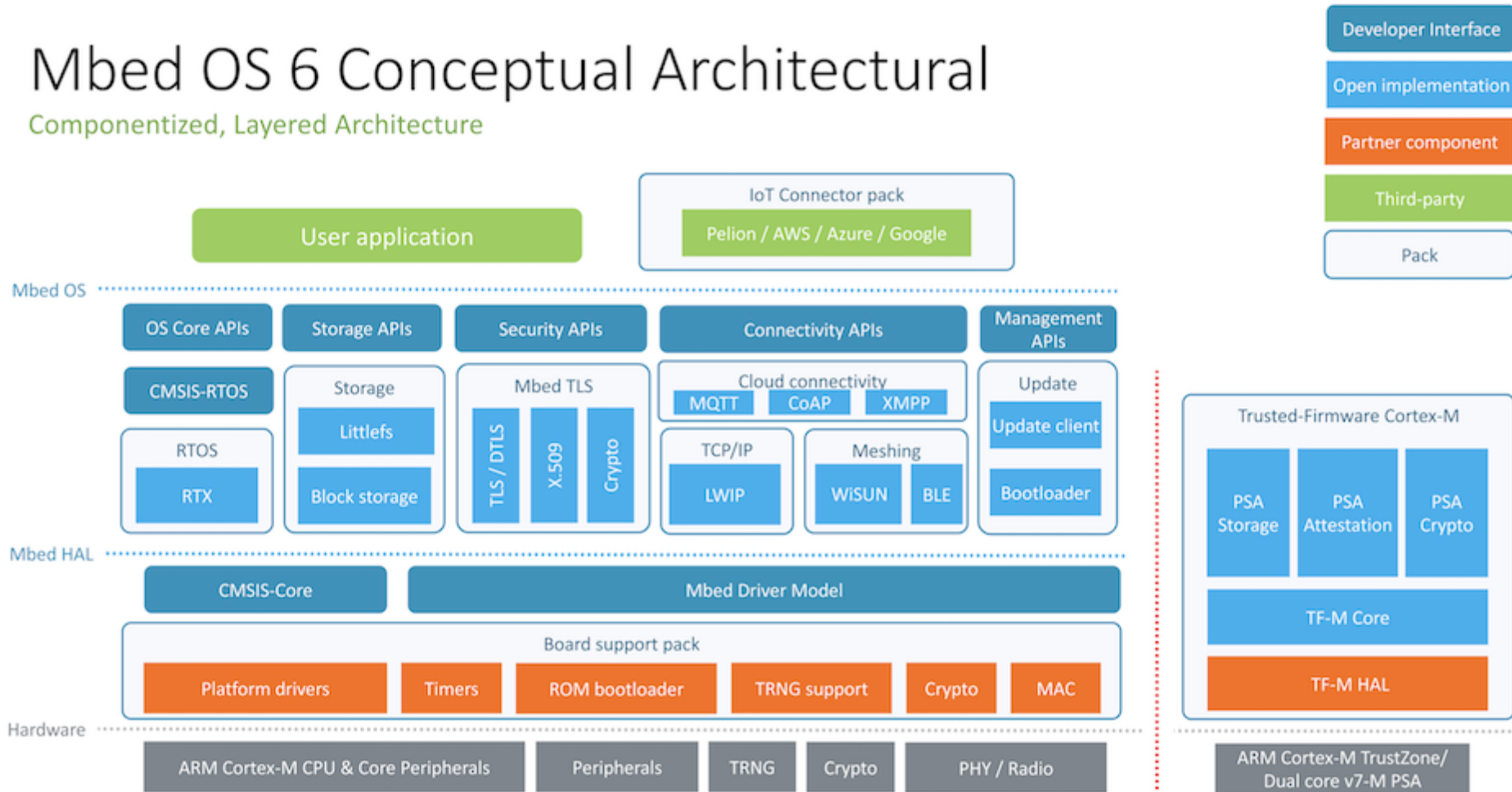
- ARM MBED OS (version 6)
 - Nouvelle version de la bibliothèque MBED
 - Refonte anciennes classes / fonctions
 - Evolution vers un RTOS
 - Support de plus en plus complet pour les protocoles de communication



Architecture MBED OS6

Mbed OS 6 Conceptual Architectural

Componentized, Layered Architecture



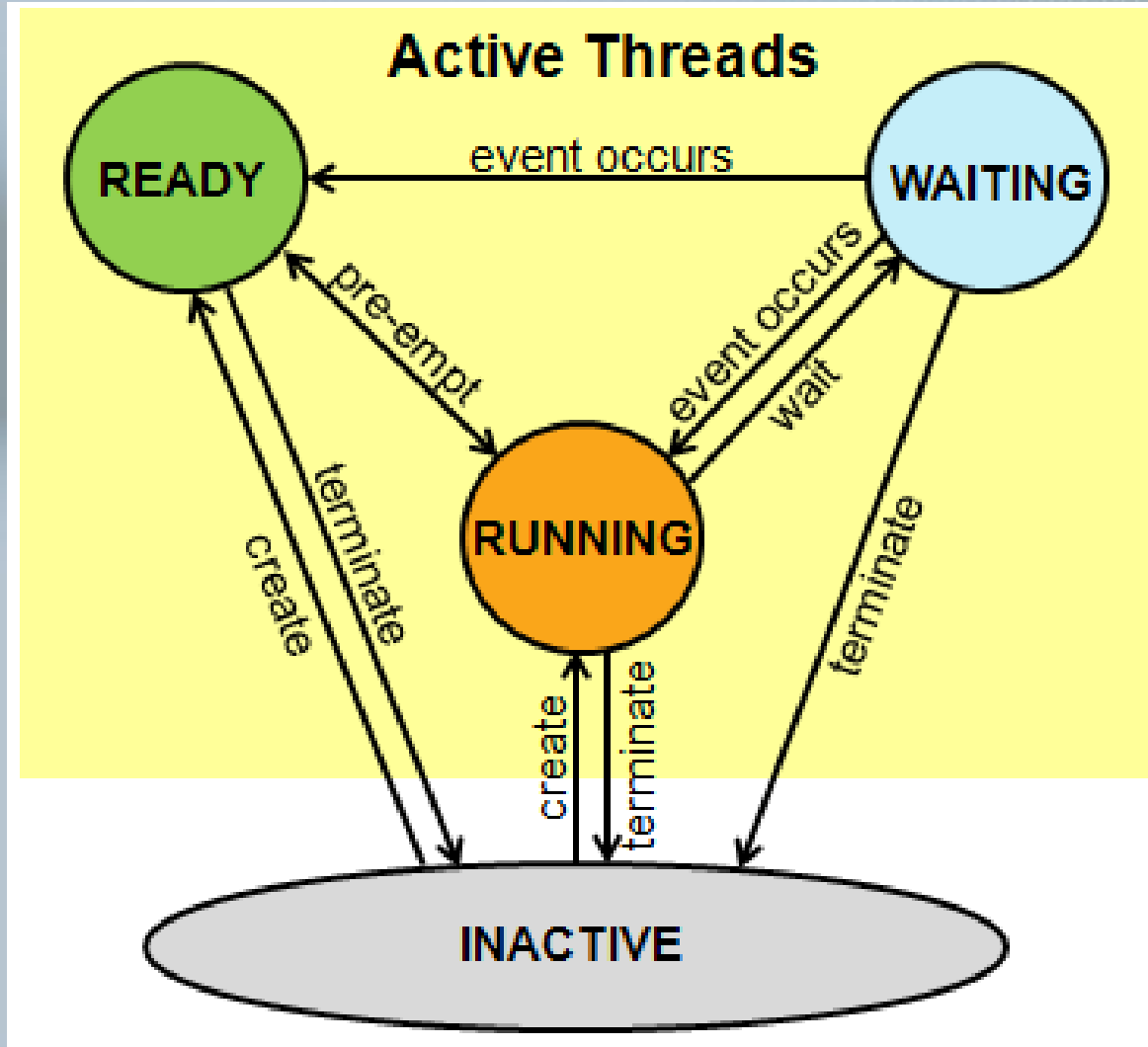
La dernière nouveauté...

- Le projet MBED-OS est abandonné par ARM
- Fin des activités juillet 2026
- Ne pas démarrer de nouveau projet avec MBED-OS !
- Surveiller l'activité du fork MBED-CE
 - Community Edition
 - Si ça marche, pourquoi pas...

Le minimum pour débuter

- Etude préalable de l'API MBED OS6
 - Gestion des tâches
 - Scheduling, RT...
 - Gestion des E/S simples
 - E/S TOR ou analogiques
 - PWW
 - Interruptions
 - Interfaces de communication de base
 - Asynchrone (RS232)
 - I²C, SPI
 - CAN

Gestion des tâches

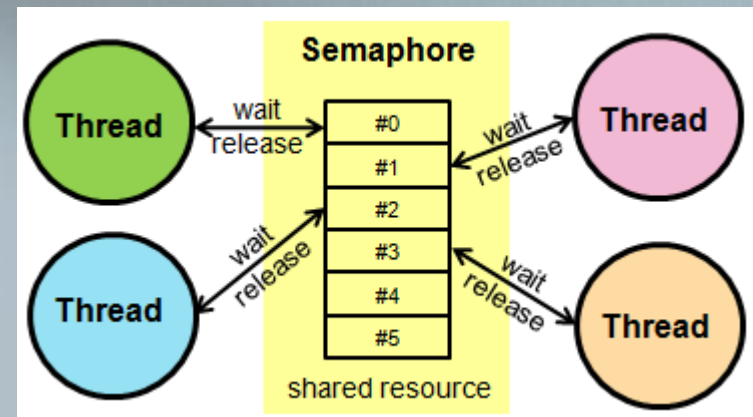
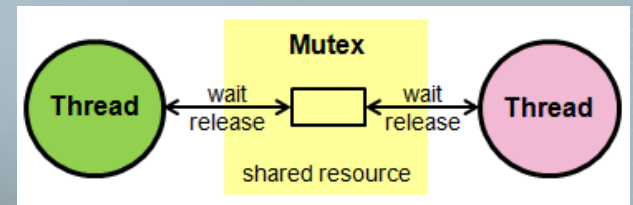


Au cœur : les threads

- Processus légers
 - Partagent un environnement d'exécution
- Classe Thread
 - La fonction main est elle-même un thread particulier
 - Créé et lancé automatiquement
 - Au démarrage de l'application
 - Possibilité de définir des priorités
 - Gestion temps réel

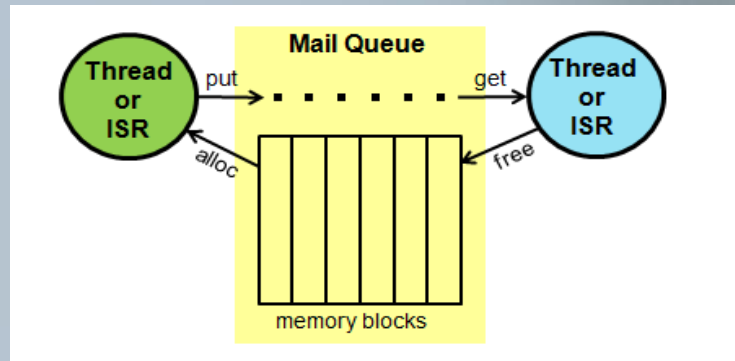
Partage des ressources

- L'accès à des ressources partagées est le premier problème à gérer pour un noyau multitâche
- Outils classiques
 - Le MUTEX
 - Le SEMAPHORE



Synchronisation et communication

- Second problème à résoudre
- Outils classiques
 - Le SIGNAL
 - Pour la synchronisation entre threads
 - EventFlags
 - Mail & Mail Queue
 - Pour la communication entre threads



Mais aussi...

- Condition Variable
 - Mise en sommeil de thread(s) dans l'attente d'un signal ou du changement d'état d'une variable
- Event handling
 - Event & EventQueue

Une application sans RTOS ?

- Toujours possible
 - Ne pas utiliser les objets spécifiques du RTOS
 - Ne pas inclure la bibliothèque RTOS
- Programmation MCU traditionnelle
 - Large utilisation des interruptions
 - Mais les classes mises à disposition facilitent leur utilisation
 - Exemple : classe ***Ticker***
 - Appel périodique d'une fonction via IRQ Timer

Mbed OS bare metal profile

- Bibliothèque « Hybride »
- Sans RTOS
- L'accès à la plupart des APIs reste possible
 - Y compris certaines classes « RTOS »
- Avantage principal
 - Compacité de l'exécutable généré
- Inconvénient principal
 - Pas RT

Le cas (épineux) du wait...

- Les fonctions « wait »
 - wait
 - wait_ms
 - wait_us
- Lancent une boucle de « NOP » durant un temps donné
 - Ne pas utiliser dans un cas réel
 - Programmer des Ticker
- La méthode statique ***ThisThread::sleep_for*** est quant à elle largement utilisable

Gestion des E/S

- E/S Numériques TOR
- Gestion individuelle
 - DigitalIn
 - DigitalOut
 - DigitalInOut
 - InterruptIn
- Gestion groupée (par bus)
 - BusIn, BusOut, BusInOut
 - PortIn, PortOut, PortInOut
- Les classes Bus sont moins rapides que les classes Port, mais plus flexibles

PWM et IRQ TOR

- Interruption sur événement TOR
 - Classe Interruptin
 - Attention : seules certaines broches du MCU sont utilisables avec ces IRQ
 - Méthodes principales
 - `enable_irq`, `disable_irq`
 - `rise`, `fall`
- Génération de signaux PWM
 - Classe `PwmOut`
 - Grandes possibilités de gestion PWM
 - Par rapport cyclique, par $t_{on}...$

E/S Analogiques

- Les modules MBED sont dotés (selon les références)
 - D'une ou plusieurs entrées analogiques
 - CAN
 - D'une ou plusieurs sorties analogiques
 - CNA
- Classes
 - AnalogIn
 - AnalogOut

Communication bas niveau

- Asynchrone (RS232)
 - BufferedSerial
 - UnbufferedSerial
- I2C
 - I2C
 - I2CSlave
- SPI
 - SPI
 - SPISlave
 - QuadSPI (QSPI)
- Réseau CAN
 - CAN



Communication haut niveau

- Des classes existent pour faciliter la mise en œuvre de connexions réseau à plusieurs niveaux
 - Niveau Socket
 - Niveau Liaison
 - ...
- Prise en charge de protocoles spécifiques (ou pas) à l'embarqué (IoT)
 - BLE, NFC, LoRaWAN...

Et côté USB ?

- Plusieurs drivers USB disponibles
 - Héritant généralement de USBDevice
- Gestion haut niveau de la plupart des devices USB « standard »
 - USBMouse / USBKeyboard / USBMouseKeyboard
 - USBSerial / USBCDC
 - USBHID
 - USBAudio
 - ...

Sans oublier

- Plusieurs autres API
 - Platform
 - Time
 - Power
 - Memory
 - Assert
 - ...
 - Data Storage
 - FATFileSystem
 - ...
 - ...

Les autres bibliothèques

- Les modules MBED forment un écosystème très actif
- Une multitude de classes existent pour une multitude d'applications, de composants, de protocoles
 - C'est une source de solutions importante
 - Même pour d'autres cibles que MBED
 - Portage de code MBED vers d'autres cibles par exemple