Introduction to mbed OS

ARM

Mihail Stoyanov

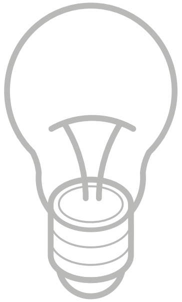
Partner Enablement Team Lead,ARM mbed

ARM mbed Connect / China
December / 05 / 2016

mbed OS Introduction

OS platforms for IoT and embedded

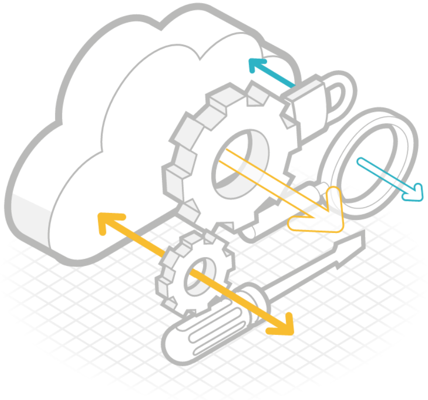
- IoT is a disruptive jump in complexity for embedded software
  - Requirements for Connectivity → Security → Management
- Adoption of OS platforms is necessary for industry success
  - The cost and risk of projects building from scratch is too high
  - A platform OS lowers the barrier to entry for new entrants



- IoT products require device management services
  - Will become essential components within a platform OS

Platform OS requirements

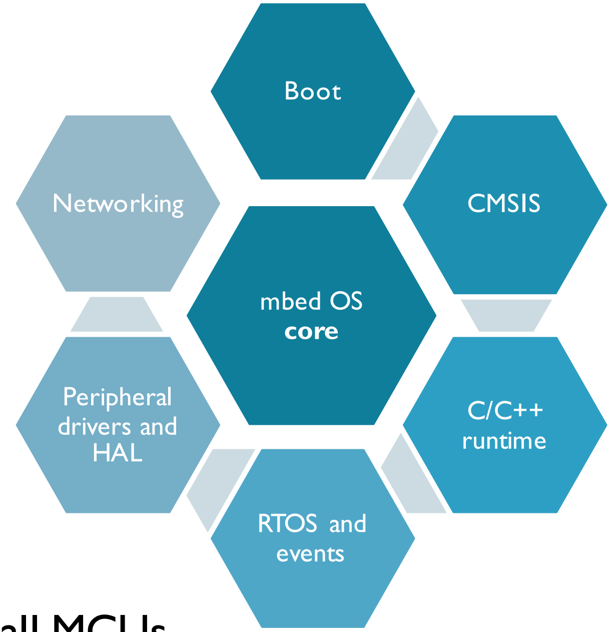
- Accelerate the development of IoT devices
  - Pre-integrate all the necessary connectivity and software components
  - Provide across many hardware solutions
- Accelerate the deployment of IoT devices
  - Solve the device management problems
- Utilize ecosystem scale
  - In collaboration to provide maximum gearing and pace



mbed OS core

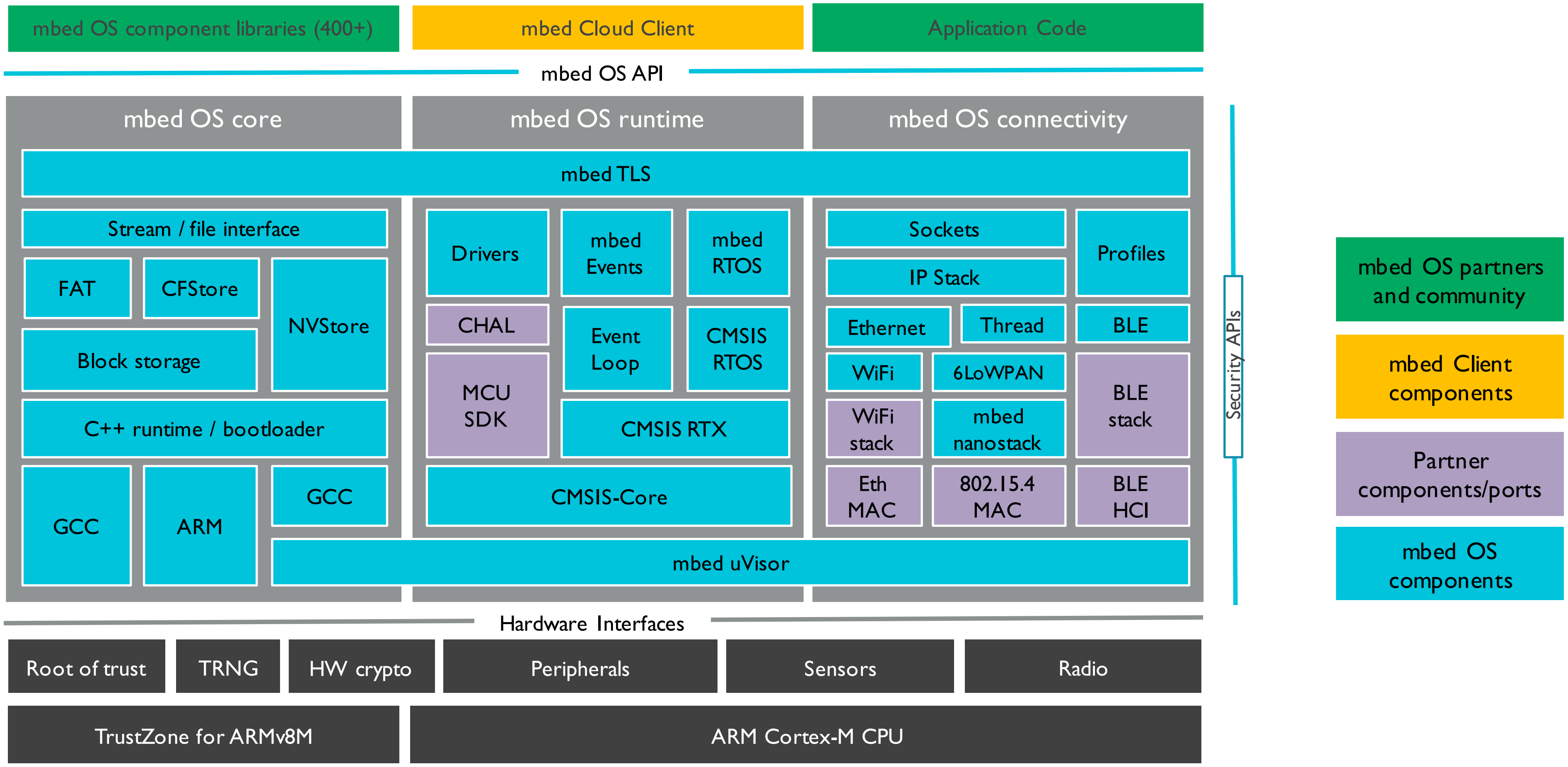
mbed OS core

- Enables application and component libraries to work unchanged across MCUs
  - Provides portability for developers
- Consistent boot and C/C++ runtime across MCUs
  - Including support across different toolchains, std library integrations
- RTOS kernel
  - Built on the established, widely used, open source CMSIS-RTOS RTX
  - Very small kernel optimised for constrained memory devices
- Peripheral driver APIs
  - Common Driver APIs for all common peripherals, supported across all MCUs



ARM

mbed OS stack



mbed OS component libraries (400+)

mbed Cloud Client

Application Code

mbed OS API

mbed OS core

mbed OS runtime

mbed OS connectivity

mbed TLS

Stream / file interface

FAT

CFStore

NVStore

Block storage

C++ runtime / bootloader

GCC

ARM

GCC

Drivers

mbed Events

mbed RTOS

CHAL

Event Loop

CMSIS RTOS

MCU SDK

CMSIS RTX

CMSIS-Core

Sockets

Profiles

IP Stack

Ethernet

Thread

BLE

WiFi

6LoWPAN

WiFi stack

mbed nanostack

BLE stack

Eth MAC

802.15.4 MAC

BLE HCI

mbed uVisor

Security APIs

Hardware Interfaces

Root of trust

TRNG

HW crypto

Peripherals

Sensors

Radio

TrustZone for ARMv8M

ARM Cortex-M CPU

🟩 mbed OS partners and community

🟨 mbed Client components

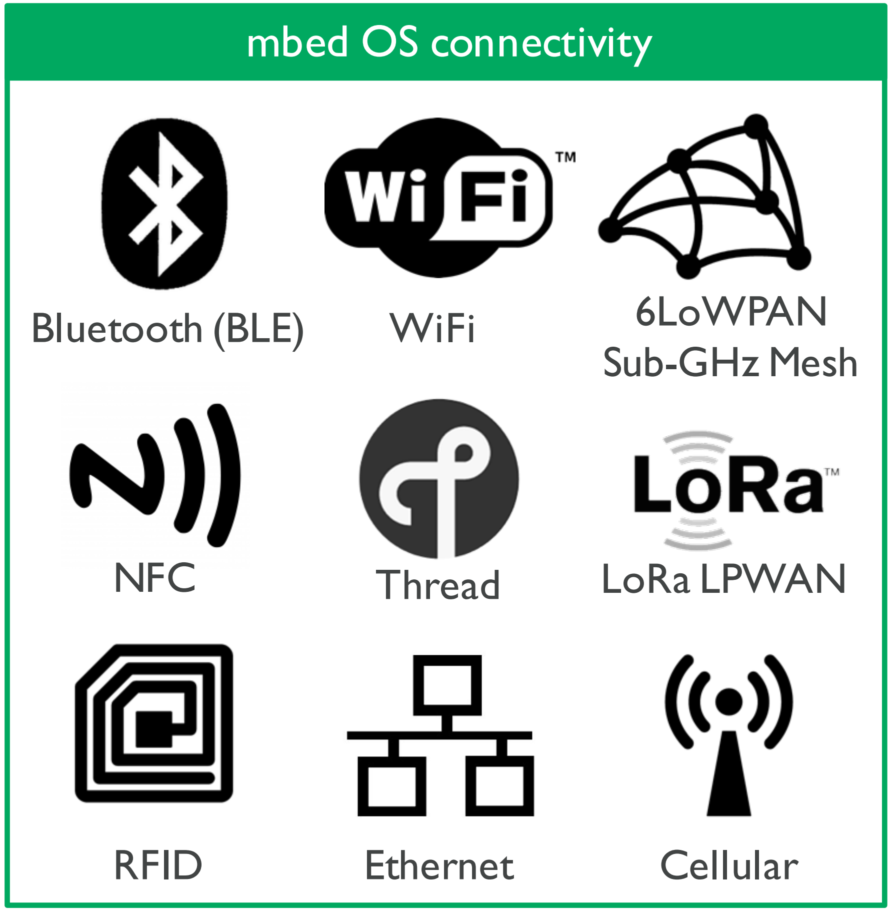
🟪 Partner components/ports

🟦 mbed OS components

mbed OS connectivity

mbed OS connectivity

- Broad IoT connectivity
  - Unified support of customer product portfolio
- mbed partner and community contributions
  - Development hardware
  - Software libraries
  - Tutorials and examples
  - Commercial products and support
- Future standards:
  - Next-gen Industrial 802.15.4 Mesh
  - NB-IoT



mbed OS connectivity

Bluetooth (BLE)

WiFi

6LoWPAN Sub-GHz Mesh

NFC

Thread

LoRa LPWAN

RFID

Ethernet

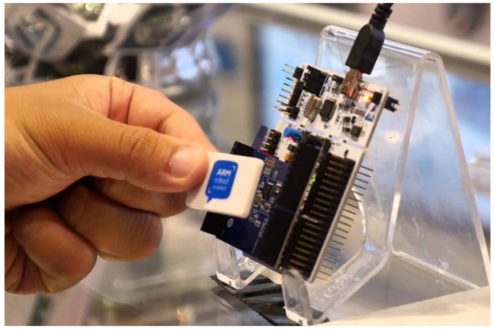
Cellular

Bluetooth Low Energy

- mbed OS BLE APIs already widely and successfully used
  - Portable across different vendors
  - Used in e.g. Google Eddystone beacons

- BLE has huge potential beyond its current application areas
  - Trusted robust radio, low cost chips, built into every mobile and laptop
  - Roadmap to longer range and IP

- Currently expanding support for BLE across more platforms
  - Already supporting Cordio Radio IP, Nordic SoC, Maxim, ST

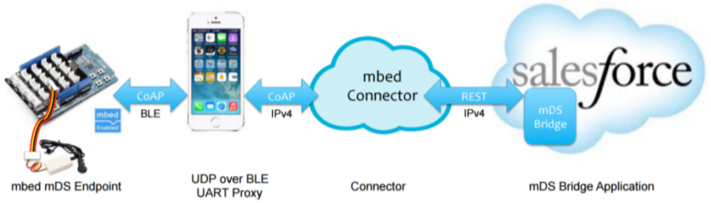


NORDIC SEMICONDUCTOR
Smarter Things

Bluetooth SMART

life.augmented

ARM CORDIO
Radio core IP



CoAP BLE

CoAP IPv4

mbed Connector

REST IPv4

salesforce

mDS Bridge

mbed mDS Endpoint

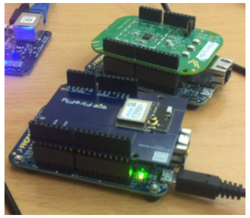
UDP over BLE UART Proxy

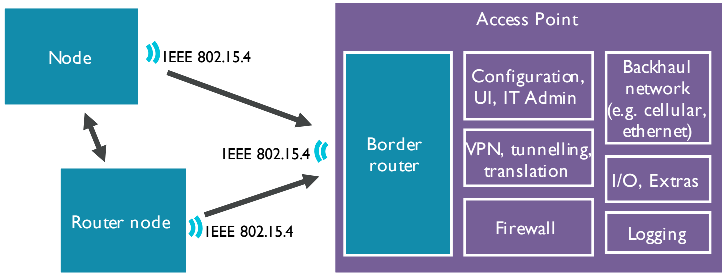
Connector

mDS Bridge Application

mbed OS 6LoWPAN 802.15.4 and Thread

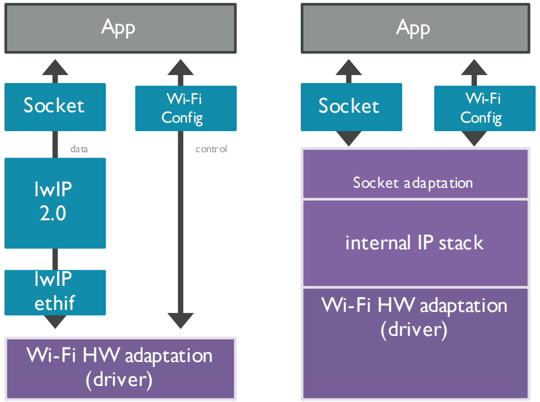
- Continue to be a leader in Thread implementation and standardisation
  - Also supporting generic 6LoWPAN 2.4GHz and Sub-GHz
- mbed OS implements MAC abstractions for simple transceiver support
  - Enables any 802.15.4 SoC or transceiver access to Thread/6LoWPAN
  - Multiple transceivers now working, can be used with any suitable mbed Enabled MCU
- Border router and access point references available
- Focus on commercial, industrial and smart city

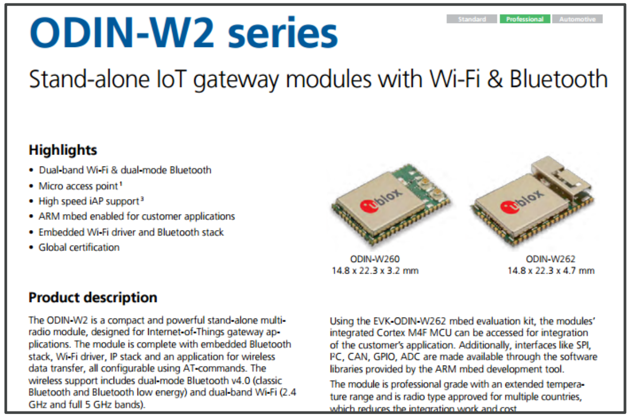




Enabling single module WiFi solutions

- mbed OS unlocks WiFi modules as a platform
  - Application and OS runs on the integrated wireless modules
  - Enables BOM reduction and reduced design complexity
- Supports integration of different WiFi architectures
  - WiFi MAC (mbed OS IP stack)
  - WiFi Network Processor (internal IP stack)
- u-blox ODIN-W2 module support in mbed OS 5.2
  - Look out for other partners landing their support




Standard
Professional
Automotive

ODIN-W2 series

Stand-alone IoT gateway modules with Wi-Fi & Bluetooth

Highlights

- Dual-band Wi-Fi & dual-mode Bluetooth
- Micro access point[1]
- High speed iAP support[3]
- ARM mbed enabled for customer applications
- Embedded Wi-Fi driver and Bluetooth stack
- Global certification

ODIN-W260
14.8 x 22.3 x 3.2 mm

ODIN-W262
14.8 x 22.3 x 4.7 mm

Product description

The ODIN-W2 is a compact and powerful stand-alone multi-radio module, designed for Internet-of-Things gateway applications. The module is complete with embedded Bluetooth stack, Wi-Fi driver, IP stack and an application for wireless data transfer, all configurable using AT-commands. The wireless support includes dual-mode Bluetooth v4.0 (classic Bluetooth and Bluetooth low energy) and dual-band Wi-Fi (2.4 GHz and full 5 GHz bands).

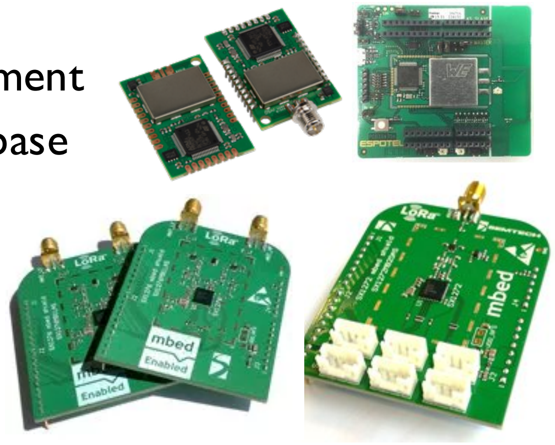
Using the EVK-ODIN-W262 mbed evaluation kit, the modules' integrated Cortex M4F MCU can be accessed for integration of the customer's application. Additionally, interfaces like SPI, I²C, CAN, GPIO, ADC are made available through the software libraries provided by the ARM mbed development tool.

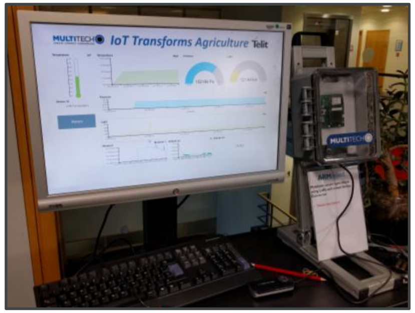
The module is professional grade with an extended temperature range and is radio type approved for multiple countries, which reduces the integration work and cost

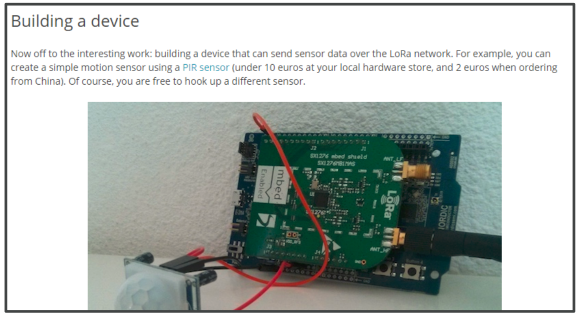
Native support for LoRaWAN

- mbed OS is already used for LoRa development
  - We invested early in LoRa, and are planning to increase investment
  - 7 LoRa hardware devices already in platform/component database
  - mbed LoRa examples imported 1000's times

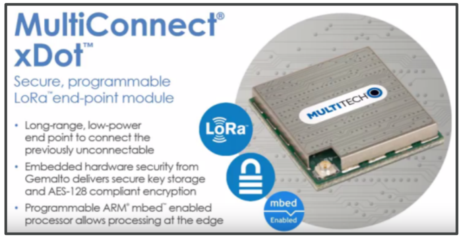
- Extending to provide integrated LoRaWAN support






Building a device

Now off to the interesting work: building a device that can send sensor data over the LoRa network. For example, you can create a simple motion sensor using a PIR sensor (under 10 euros at your local hardware store, and 2 euros when ordering from China). Of course, you are free to hook up a different sensor.

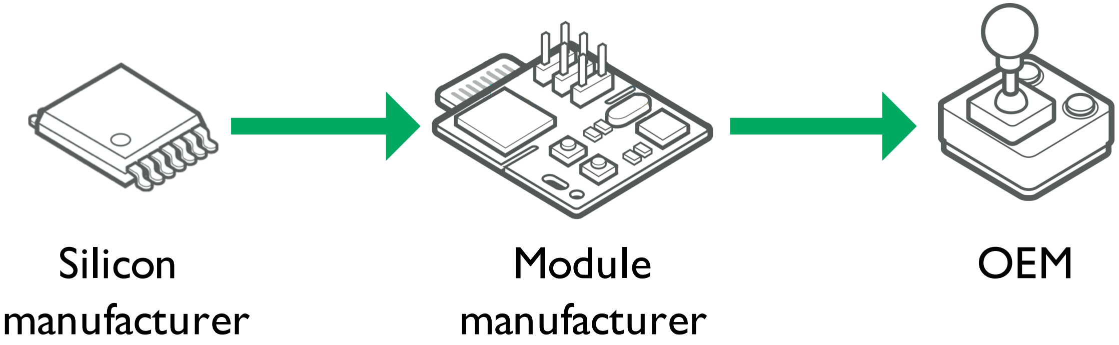

MultiConnect® xDot™

Secure, programmable LoRa™ end-point module

- Long-range, low-power end point to connect the previously unconnectable
- Embedded hardware security from Gemalto delivers secure key storage and AES-128 compliant encryption
- Programmable ARM® mbed™ enabled processor allows processing at the edge

mbed OS modules

- Smart Modules will become a key building block for IoT products
  - Wraps up complexity to simplify product design – radio, application, services
  - Common design-in for IoT applications with initial runs of 10-100k units



- Partnering with manufacturers on mbed Enabled wireless modules
  - WiFi, BLE, 802.15.4, LoRa

mbed OS security

mbed OS security

- Covers three main types of threat

- Security of system, including ability to provision, manage and update devices (e.g. security fix)

- Security of communications between device and cloud services

- Security and integrity of device itself from untrusted or malicious code

mbed OS security

mbed client
Lifecycle Security

mbed TLS
Communication Security

mbed uVisor
Device Security

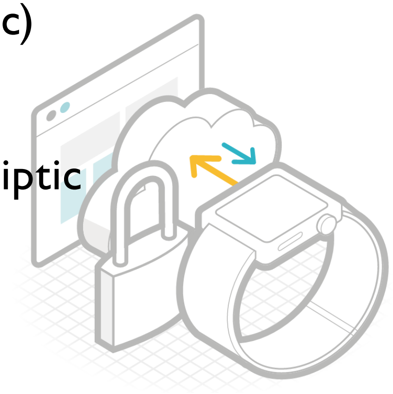
mbed TLS

- mbed TLS enables mbed OS cryptographic and SSL/TLS capabilities

Features:

- SSL / TLS (SSL 3.0 up to TLS 1.2)
- Extensive TLS Extension support
- X.509 certificate handling
- Abstraction layers
- DTLS

- Symmetric encryption (AES, etc)
- Hash algorithms (SHA-2, etc)
- Public Key algorithms (RSA, Elliptic Curves, etc)
- Random number generation (CTR_DRBG, etc)



mbed OS tools

mbed OS tools

- Free core tools provide build, debug, test and collaboration workflows
- Third party partner industry tools support

mbed OS DVCS support

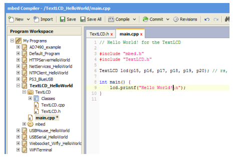
git mercurial

GitHub Atlassian Bitbucket

GitLab mbed.org/code

mbed OS core tools

mbed CLI
Command Line Interface


mbed Compiler - /TextLCD_HelloWorld/main.cpp

```
// Hello World! for the TextLCD

#include "mbed.h"
#include "TextLCD.h"

TextLCD lcd(p15, p16, p17, p18, p19, p20); // rs,

int main() {
    lcd.printf("Hello World!\n");
}
```

mbed Compiler
Free Online IDE

mbed Greentea
Porting Testsuite and CI

mbed pyOCD
CMSIS-DAP Debug Library

mbed DAPLink
CMSIS-DAP Debug Firmware

mbed OS IDEs and toolchains

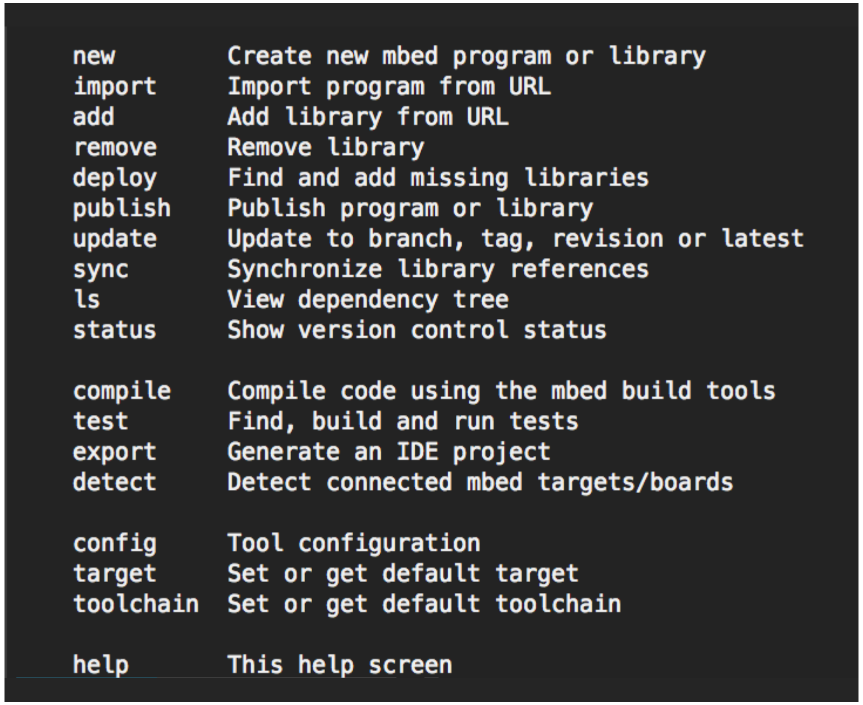
ARMKEIL
Microcontroller tools
ARM Compiler 5

IAR SYSTEMS

GCC

mbed CLI

- Single tool for fetching, configuring, building and updating all source code that an mbed OS program depends on
  - Offline development - simple to install for Windows, Mac OS X, Linux
  - Backwards compatible with existing mbed OS 2 programs and libraries
- Supports mixing multiple version control backends: mercurial (hg) and git
- Supports building for multiple toolchains: ARMCC, GCC, IAR and exporting to IDEs

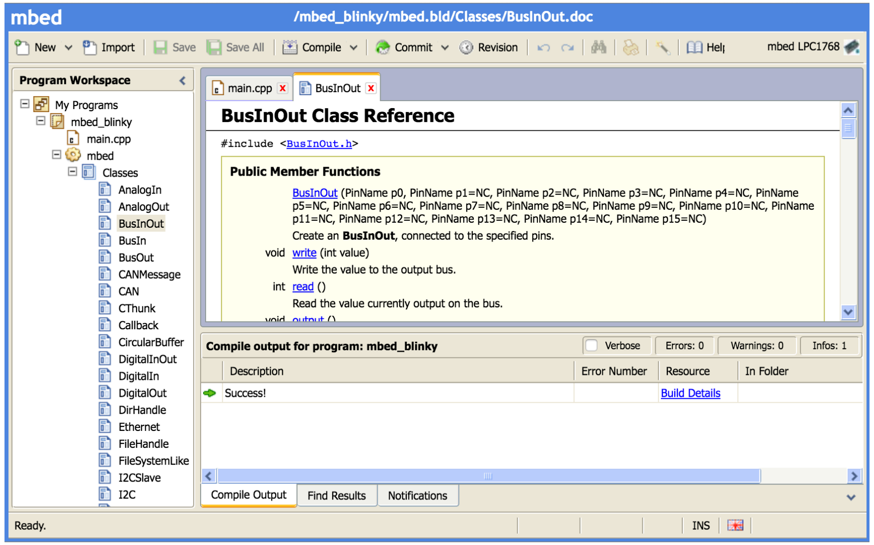


```
new        Create new mbed program or library
import     Import program from URL
add        Add library from URL
remove     Remove library
deploy     Find and add missing libraries
publish    Publish program or library
update     Update to branch, tag, revision or latest
sync       Synchronize library references
ls         View dependency tree
status     Show version control status

compile    Compile code using the mbed build tools
test       Find, build and run tests
export     Generate an IDE project
detect     Detect connected mbed targets/boards

config     Tool configuration
target     Set or get default target
toolchain  Set or get default toolchain

help       This help screen
```

mbed Online Compiler and third party IDEs

- mbed Online Compiler
  - Instant access to working build environment
  - Supports compilation, version control, publishing projects, library documentation
  - Compilation using ARM Compiler 5
  - Free to use

- Third party IDEs
  - Develop and debug with your favourite IDE
  - Export from mbed CLI or website to 3rd party IDEs
  - Keil uVision, IAR Workbench, GCC/Makefile, etc



BusInOut Class Reference

```
#include <BusInOut.h>
```

Public Member Functions

BusInOut (PinName p0, PinName p1=NC, PinName p2=NC, PinName p3=NC, PinName p4=NC, PinName p5=NC, PinName p6=NC, PinName p7=NC, PinName p8=NC, PinName p9=NC, PinName p10=NC, PinName p11=NC, PinName p12=NC, PinName p13=NC, PinName p14=NC, PinName p15=NC)

Create an **BusInOut**, connected to the specified pins.

void write (int value)

Write the value to the output bus.

int read ()

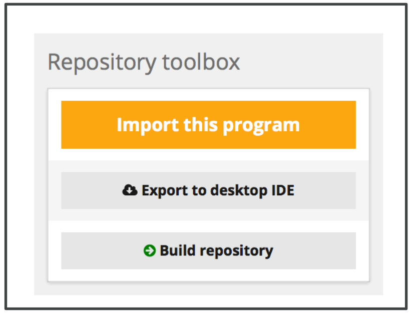
Read the value currently output on the bus.

void output ()

Compile output for program: mbed_blinky

Description	Error Number	Resource	In Folder
Success!		Build Details	

```
> mbed export -i uvision -m K64F
```



ARMKEIL
Microcontroller tools

ARM Compiler 5

IAR SYSTEMS

mbed OS ecosytem

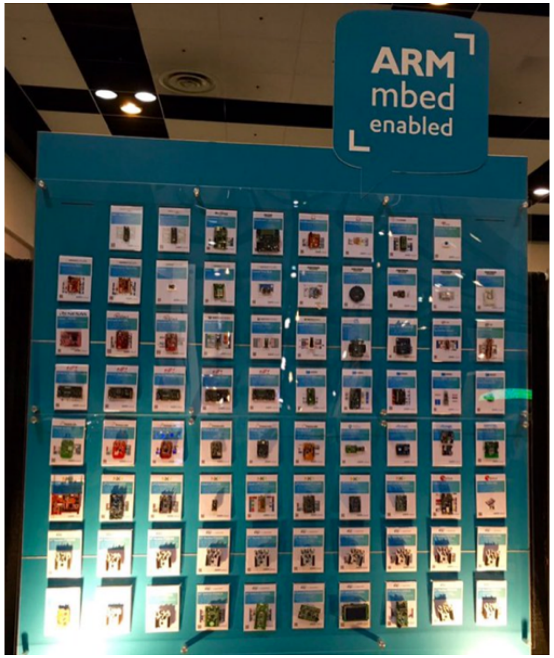
mbed OS partner ecosystem

- mbed OS runs on platforms from a wide range of MCU partners
  - Cortex-M0 to Cortex-M7
  - Simple MCUs to full-featured IoT SoCs including radios
- Incremental effort for each partner, network effects for all

mbed OS Platforms

mbed Enabled

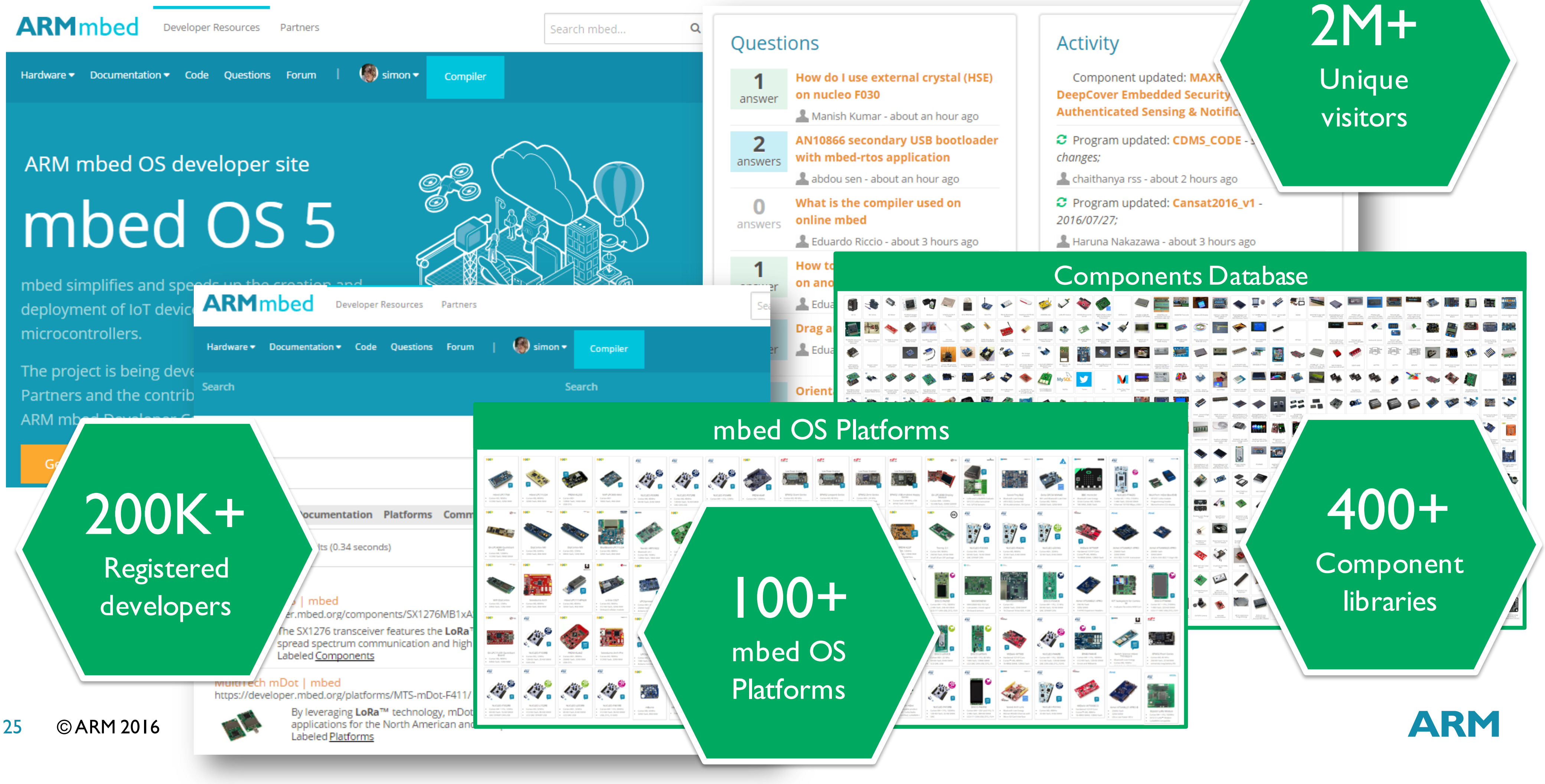
- mbed developers are looking for trusted building blocks they know work with mbed OS
- mbed Enabled offers partners a unique opportunity to demonstrate IoT compatibility, gaining customer mindshare and reach new customers



ARM mbed enabled

mbed Enabled is a mark of confidence for developers

mbed OS developer community



2M+ Unique visitors

200K+ Registered developers

100+ mbed OS Platforms

400+ Component libraries

Components Database

mbed OS Platforms

ARMmbed Developer Resources Partners Search mbed...

Hardware ▾ Documentation ▾ Code Questions Forum | simon ▾ Compiler

ARM mbed OS developer site

mbed OS 5

mbed simplifies and speeds up the creation and deployment of IoT devic microcontrollers.

The project is being deve Partners and the contrib ARM mbed Developer C

Questions

1 answer How do I use external crystal (HSE) on nucleo F030
Manish Kumar - about an hour ago

2 answers AN10866 secondary USB bootloader with mbed-rtos application
abdou sen - about an hour ago

0 answers What is the compiler used on online mbed
Eduardo Riccio - about 3 hours ago

1 answer How to on ano
Edua

Drag a
Edua

Orient

Activity

Component updated: MAXR DeepCover Embedded Security Authenticated Sensing & Notific

Program updated: CDMS_CODE - changes;
chaithanya rss - about 2 hours ago

Program updated: Cansat2016_v1 - 2016/07/27;
Haruna Nakazawa - about 3 hours ago

ARMmbed Developer Resources Partners

Hardware ▾ Documentation ▾ Code Questions Forum | simon ▾ Compiler

Search Search

ocumentation Platforms Comm

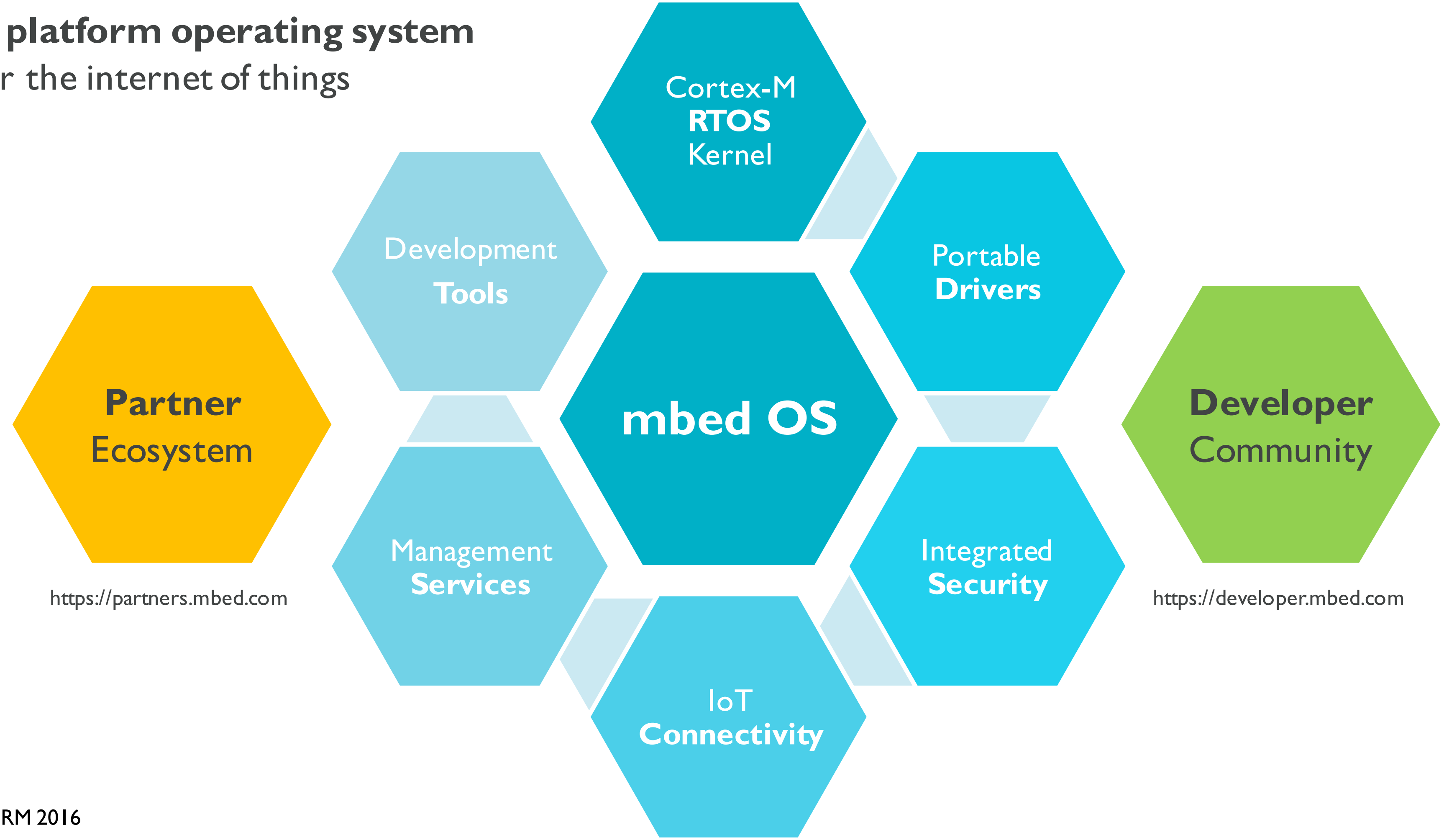
ts (0.34 seconds)

s | mbed
r.mbed.org/components/SX1276MB1xA
The SX1276 transceiver features the LoRa™ spread spectrum communication and high
Labeled Components

MultiTech mDot | mbed
https://developer.mbed.org/platforms/MTS-mDot-F411/
By leveraging LoRa™ technology, mDot applications for the North American and
Labeled Platforms

mbed OS 5

A **platform operating system**
for the internet of things



https://partners.mbed.com

https://developer.mbed.com

Thank you

ARM

mbed OS APIs – Threads and Timing

- Task
  - Thread: defining, creating, and controlling thread functions in the system
  - Mutex: synchronize the execution of threads
  - Semaphore: manages thread access to a pool of shared resources of a certain type
  - Signals: wait for signals and be notified of events
  - Queue: queue pointers to data from producer threads to consumer threads
  - MemoryPool: define and manage fixed-size memory pools
  - Mail: a queue providing a memory pool for allocating messages
- Timing
  - Ticker: set up a recurring interrupt
  - TimeOut: set up an interrupt to call a function after a specified delay
  - Timer: create, start, stop and read a timer for measuring small times
  - wait: busy wait of various resolutions
  - sleep: put the MCU to sleep
  - time: date and time functions

```
#include "mbed.h"

DigitalOut led1(LED1);

void led2_thread() {
    while (1) {
        led2 = !led2;
        wait(0.5);
    }
}

int main() {
    Thread thread(led2_thread);
    while (1) {}
}
```

```
#include "mbed.h"

Ticker tick;
DigitalOut led1(LED1);

void flip() { led2 = !led2;}

int main() {
    tick.attach(&flip, 2.0);
    while(1) {}
}
```

https://docs.mbed.com/docs/mbed-os-api-reference

mbed OS APIs – Basic I/O

- Analog I/O
  - AnalogIn: read the voltage of an analog input pin.
  - AnalogOut: set the voltage of an an along output pin.
- Digital I/O
  - DigitalIn: read a single digital input pin.
  - DigitalOut: write to a single digital output pin.
  - DigitalInOut: read and write to a single bi-directional digital pin.
- Grouped control
  - BusIn: read multiple pins as a single value.
  - BusOut: write a single value to multiple pins.
  - BusInOut: read and write to multiple bi-directional pins.
  - PortIn: read multiple pins as a single value.
  - PortOut: write a single value to multiple pins.
  - PortInOut: read and write to multiple bi-directional pins.
- Interrupts
  - InterruptIn triggers an event when a digital input pin changes value.
- Pwm
  - PwmOut controls the frequency and mark to space ratio of a digital pulse wave (or train).

```
#include "mbed.h"

BusOut myleds(LED1, LED2, LED3, LED4);

int main() {
    while(1) {
        for(int i=0;i<16;i++) {
            myleds = i;
            wait(0.25);
        }
    }
}
```

```
#include "mbed.h"

InterruptIn button(p5);
DigitalOut led(LED1);

void flip() { led = !led; }

int main() {
    button.rise(&flip);
    while(1) {}
}
```

https://docs.mbed.com/docs/mbed-os-api-reference

mbed OS APIs – Serial Communication

- Digital Serial
  - Serial: generic asynchronous UART protocol
  - SPI, SPISlave: master and slave SPI interfaces
  - I2C, I2CSlave: master and slave inter-integrated circuit interfaces
  - CAN: controller area network
- USB device
  - USBSerial: create a virtual serial port over the USB port
  - USBMouse: emulate a mouse with absolute or relative positioning.
  - USBKeyboard: emulate a keyboard, sending normal and media controls
  - USBHID: communicate over a raw USBHID interface
  - USBMIDI: send and receive MIDI messages
  - USBAudio: receive audio stream from a computer over USB.
  - USBMSD: Generic class implementing the Mass Storage Device protocol
- USB host
  - USBHostMouse: receive events from a USB mouse.
  - USBHostKeyboard: read key code modifier from a USB keyboard.
  - USBHostMSD: read and write to a USB flash disk.
  - USBHostSerial: communicate with a virtual serial port.
  - USBHostHub: plug several USB devices to an mbed board using a USB hub.

```
#include "mbed.h"

SPI spi(p5, p6, p7);// mosi, miso, sclk
DigitalOut cs(p8);

int main() {
    cs = 1;
    spi.format(8,3);
    spi.frequency(1000000);
    cs = 0;
    int data = spi.write(0x8F);
    cs = 1;
}
```

```
#include "mbed.h"
#include "USBSerial.h"

USBSerial serial;

int main(void) {
    while(1) {
        serial.printf("Hello USB World!\n");
        wait(1);
    }
}
```

https://docs.mbed.com/docs/mbed-os-api-reference

mbed OS APIs – Network Communication

- Network Sockets
  - Socket: blocking and non-blocking sockets for IPv4 and IPv6 IP connections
  - TCPSocket: send a stream of data over TCP
  - UDPSocket: send packets of data over UDP
  - TCPServer: accept incoming TCP connections
- IP Transport Interfaces
  - EthernetInterface: On-chip and off-chip Ethernet MAC/PHYs
  - WiFiInterface: WiFi SoC MACs and network processor modules
  - MeshInterface: 802.15.4 radios supporting 6LoWPAN mesh
  - CellularInterface: cellular modem module interfaces
- Bluetooth LE
  - BLE: Base GATT/GAP interface for BLE SoCs and Chipsets
  - Battery, Device Firmware Update (DFU), Device Information, Health Thermometer, Heart Rate, UART, UriBeacon, iBeacon: supported reference services

```
#include "mbed.h"
#include "TCPSocket.h"
#include "WiFiInterface.h"

WiFiInterface wifi;
TCPSocket socket;

int main() {
    wifi.connect("wifissid", "password");
    const char *ip = wifi.get_ip_address();

    socket.open(&wifi);
    socket.connect("mbed.com", 80);

    socket.close();
    wifi.disconnect();
}
```

https://docs.mbed.com/docs/mbed-os-api-reference