Using mbed libraries

The mbed library provides the C/C++ software platform and libraries to build your applications. Note: Not all platforms have the resource required to implement these APIs or have them implemented.

* [mbed Memory Model](https://os.mbed.com/handbook/Memory-Model) - The memory model used by the mbed Library
* [RTOS Memory Model](https://os.mbed.com/handbook/RTOS-Memory-Model) - The memory model used when the mbed-rtos is included in a program

Analog I/O

* [AnalogIn](https://os.mbed.com/handbook/AnalogIn) - Read the voltage applied to an analog input pin
* [AnalogOut](https://os.mbed.com/handbook/AnalogOut) - Set the voltage of an analog output pin

Digital I/O

* [DigitalIn](https://os.mbed.com/handbook/DigitalIn) - Configure and control a digital input pin.
* [DigitalOut](https://os.mbed.com/handbook/DigitalOut) - Configure and control a digital output pin.
* [DigitalInOut](https://os.mbed.com/handbook/DigitalInOut) - Bi-directional digital pins
* [BusIn](https://os.mbed.com/handbook/BusIn) - Flexible way to read multiple DigitalIn pins as one value
* [BusOut](https://os.mbed.com/handbook/BusOut) - Flexible way to write multiple DigitalOut pins as one value
* [BusInOut](https://os.mbed.com/handbook/BusInOut) - Flexible way to read/write multiple DigitalInOut pins as one value
* [PortIn](https://os.mbed.com/handbook/PortIn) - Fast way to read multiple DigitalIn pins as one value
* [PortOut](https://os.mbed.com/handbook/PortOut) - Fast way to write multiple DigitalOut pins as one value
* [PortInOut](https://os.mbed.com/handbook/PortInOut) - Fast way to read/write multiple DigitalInOut pins as one value
* [PwmOut](https://os.mbed.com/handbook/PwmOut) - Pulse-width modulated output
* [InterruptIn](https://os.mbed.com/handbook/InterruptIn) - Trigger an event when a digital input pin changes.

Timers

* [Timer](https://os.mbed.com/handbook/Timer) - Create, start, stop and read a timer
* [Timeout](https://os.mbed.com/handbook/Timeout) - Call a function after a specified delay
* [Ticker](https://os.mbed.com/handbook/Ticker) - Repeatedly call a function
* [wait](https://os.mbed.com/handbook/Wait) - Wait for a specified time
* [time](https://os.mbed.com/handbook/Time) - Get and set the realtime clock

Digital Interfaces

* [Serial](https://os.mbed.com/handbook/Serial) - Serial/UART bus
* [SPI](https://os.mbed.com/handbook/SPI) - SPI bus master
* [SPISlave](https://os.mbed.com/users/mbed_official/code/mbed/docs/tip/classmbed_1_1SPISlave.html) - SPI bus slave
* [I2C](https://os.mbed.com/handbook/I2C) - I²C bus master
* [I2CSlave](https://os.mbed.com/users/mbed_official/code/mbed/docs/tip/classmbed_1_1I2CSlave.html) - I²C bus slave
* [CAN](https://os.mbed.com/handbook/CAN) - Controller-area network bus

Real-time Operating System

* [mbed RTOS](https://os.mbed.com/handbook/RTOS)

File System

* [LocalFileSystem](https://os.mbed.com/handbook/LocalFileSystem) - Using the mbed disk as storage from within a program
* [SDFileSystem](https://os.mbed.com/handbook/SDFileSystem) - Using the mbed disk as storage from within a program

USB

* [USBDevice](https://os.mbed.com/handbook/USBDevice) - Using mbed as a USB Device
  + [USBMouse](https://os.mbed.com/handbook/USBMouse) - Emulate a USB Mouse with absolute or relative positioning
  + [USBKeyboard](https://os.mbed.com/handbook/USBKeyboard) - Emulate a USB Keyboard, sending normal and media control keys
  + [USBMouseKeyboard](https://os.mbed.com/handbook/USBMouseKeyboard) - Emulate a USB Keyboard and a USB mouse with absolute or relative positioning
  + [USBHID](https://os.mbed.com/handbook/USBHID) - Communicate over a raw USBHID interface, great for driverless communication with a custom PC program
  + [USBMIDI](https://os.mbed.com/handbook/USBMIDI) - Send and receive MIDI messages to control and be controlled by PC music sequencers etc
  + [USBSerial](https://os.mbed.com/handbook/USBSerial) - Create a virtual serial port over the USB port. Great to easily communicate with a computer.
  + [USBAudio](https://os.mbed.com/handbook/USBAudio) - Create a USBAudio device able to receive audio stream from a computer over USB.
  + [USBMSD](https://os.mbed.com/handbook/USBMSD) - Generic class which implements the Mass Storage Device protocol in order to access all kinds of block storage chips
* [USBHost](https://os.mbed.com/handbook/USBHost) - Using mbed to act as USBHost
  + [USBHostMouse](https://os.mbed.com/handbook/USBHostMouse) - Receive events from a USB mouse
  + [USBHostKeyboard](https://os.mbed.com/handbook/USBHostKeyboard) - Read key code modifier from a USB keyboard
  + [USBHostMSD](https://os.mbed.com/handbook/USBHostMSD) - Read-write a USB flash disk
  + [USBHostSerial](https://os.mbed.com/handbook/USBHostSerial) - Communicate with a virtual serial port
  + [USBHostHub](https://os.mbed.com/handbook/USBHostHub) - You can plug several USB devices to an mbed using a USB hub

Networking

* [Ethernet](https://os.mbed.com/handbook/Ethernet) - Ethernet network
  + [Ethernet Interface](https://os.mbed.com/handbook/Ethernet-Interface)
  + [TCP/UDP Socket API](https://os.mbed.com/handbook/Socket)
  + [TCP/IP Protocols and APIs](https://os.mbed.com/handbook/TCP-IP-protocols-and-APIs)