



Hands-On Workshop: **ARM mbed™**

FTF-DES-F1302

Sam Grove - ARM

Michael Norman – Freescale

J U N . 2 0 1 5

ARM®mbed™



External Use

Freescale, the Freescale logo, AltVec, C-5, CodeTEST, CodeWarrior, ColdFire, ColdFire+, C-Ware, the Energy Efficient Solutions logo, Kinetis, mobileGT, PEG, PowerQUICC, Processor Expert, QorIQ, Qorivva, SafeAssure, the SafeAssure logo, StarCore, Symphony and VortiQa are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. Alrfast, BeeKit, BeeStack, CoreNet, Flexis, Layerscape, MagnIV, MXC, Platform in a Package, QorIQ Qonverge, QUICC Engine, Ready Play, SMARTMOS, Tower, TurboLink, UMEMS, Vybrid and Xtrinsic are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © 2014 Freescale Semiconductor, Inc.



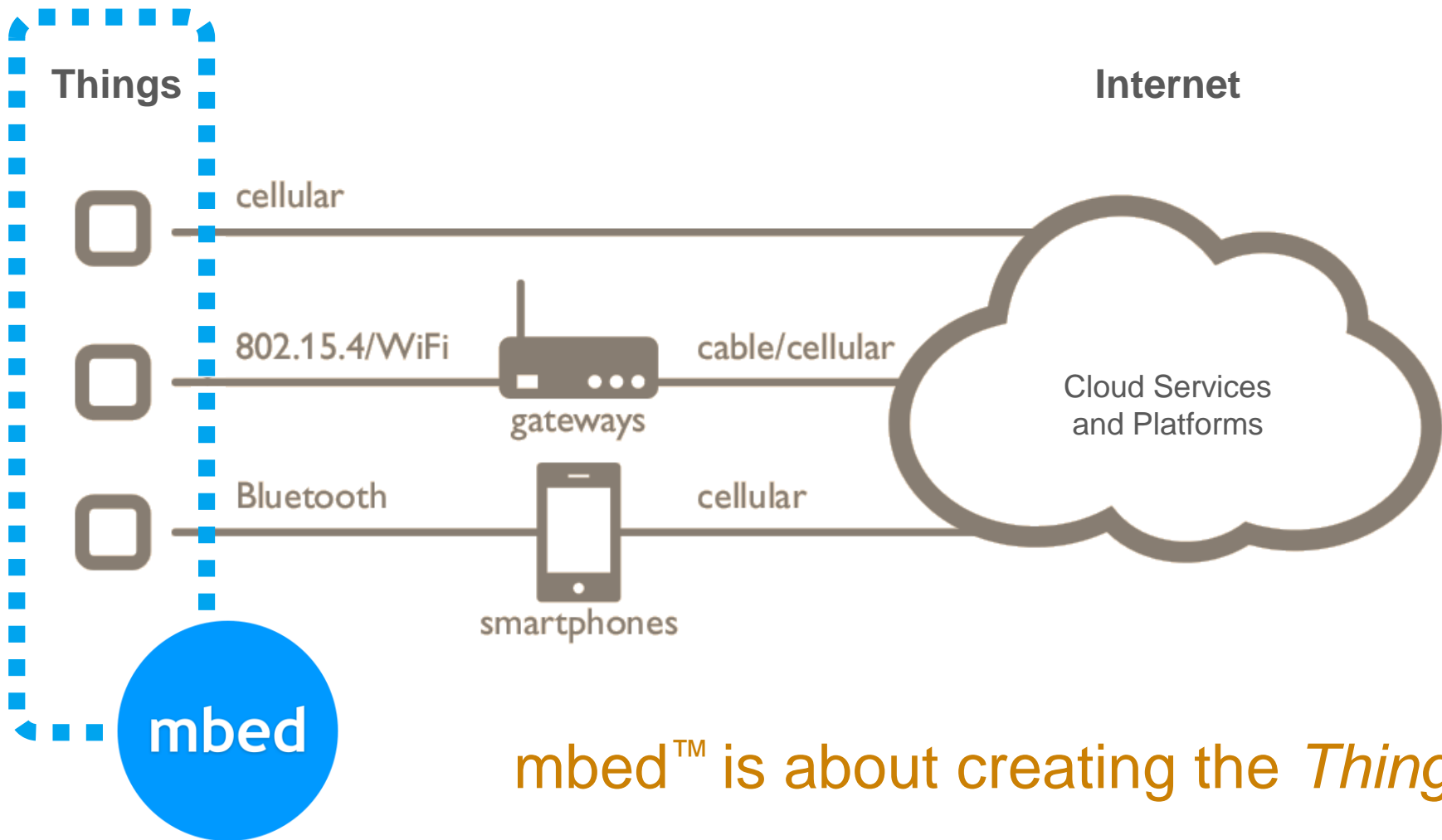
Agenda

- What is mbed
- mbed Hardware
- mbed Software
- mbed Tools
- mbed Support and Community
- Hands-On Workshop – FRDM-K64F





What is mbed



mbed™ is about creating the *Things*



Assembler

1990s



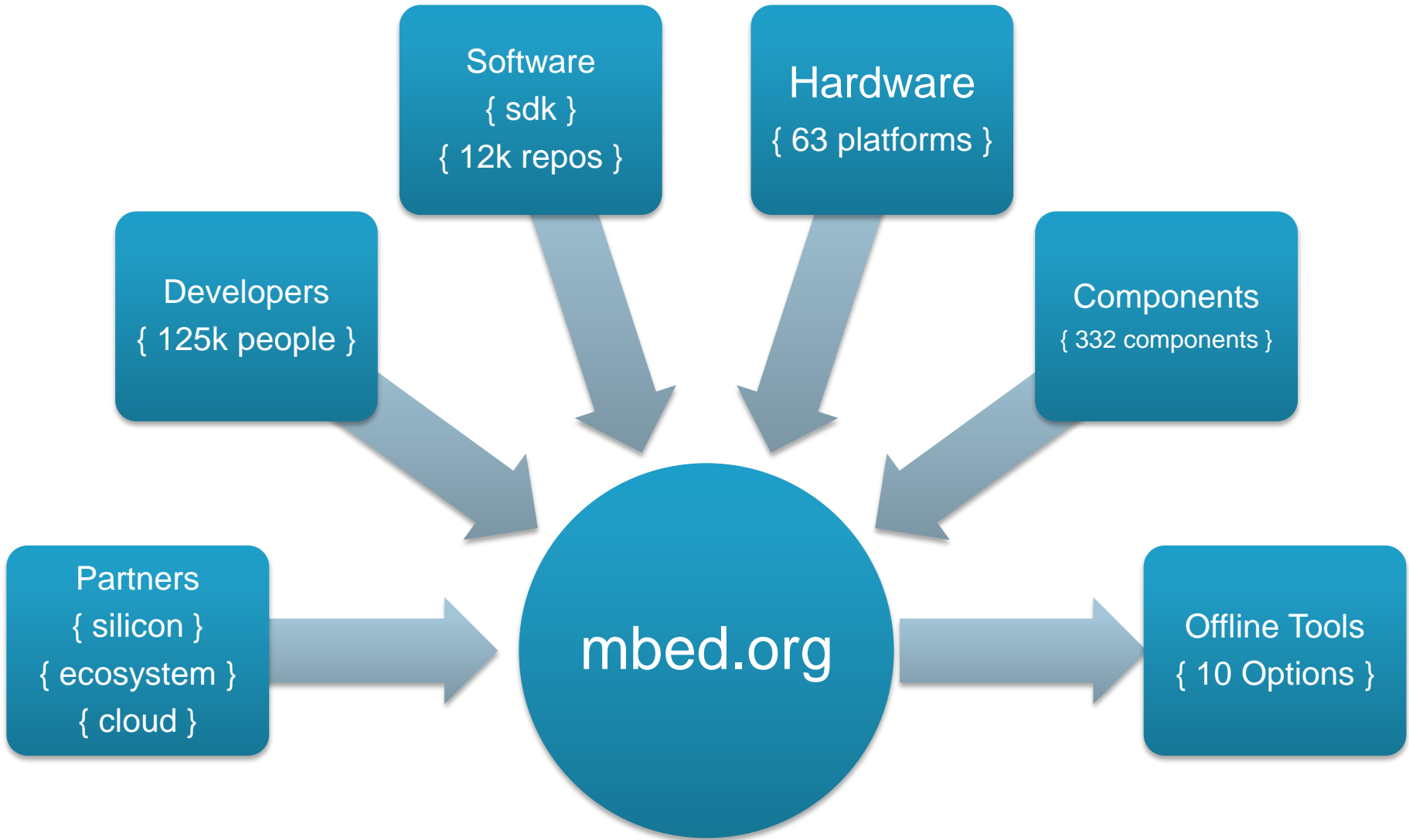
C

2000s



Platform

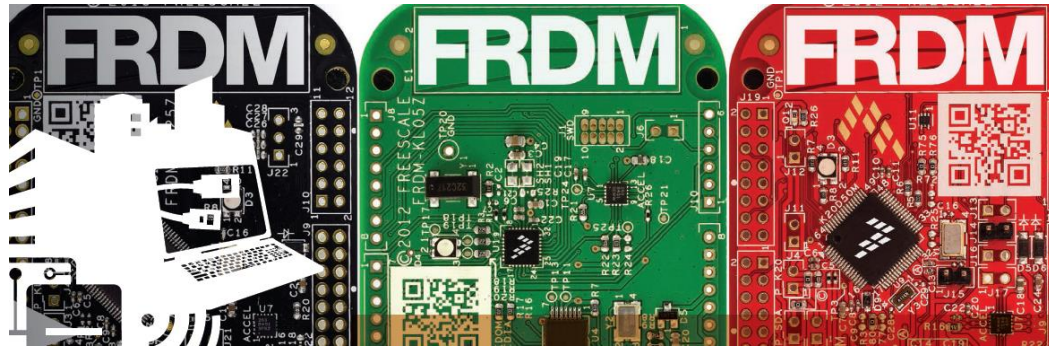
2010s








mbed Hardware




Freescale Freedom Boards



Kinetis K Series

- FRDM-K20D50M for Kinetis K20 MCUs 
- FRDM-K22F for Kinetis K22 MCUs 
- FRDM-K64F for Kinetis K64, K63, and K24 MCUs 

Kinetis L Series

- FRDM-KL02Z for Kinetis KL02 MCUs
- FRDM-KL03Z for Kinetis KL03 MCUs
- FRDM-KL05Z for Kinetis KL05Z and KL04Z MCUs 
- FRDM-KL25Z for Kinetis KL1x and KL2x MCUs 
- FRDM-KL26Z for Kinetis KL16 and KL26 MCUs
- FRDM-KL46Z for Kinetis KL3x and KL4x MCUs 

Platforms



FRDM-KL25Z

- Cortex-M0+
- 128KB Flash, 16KB RAM
- USB OTG



FRDM-KL46Z

- Cortex-M0+, 48MHz
- 256KB Flash, 32KB RAM
- USB OTG



FRDM-K64F

- Cortex-M4, 120MHz
- 1MB Flash, 256KB RAM
- Ethernet, SD Filesystem



Ethernet IoT Starter Kit

- Freescale K64F Processor
- mbed application shield
- IBM IoT Client pre-loaded



FRDM-KL05Z

- Cortex-M0+, 48MHz
- 32KB Flash, 4KB RAM



FRDM-K20D50M

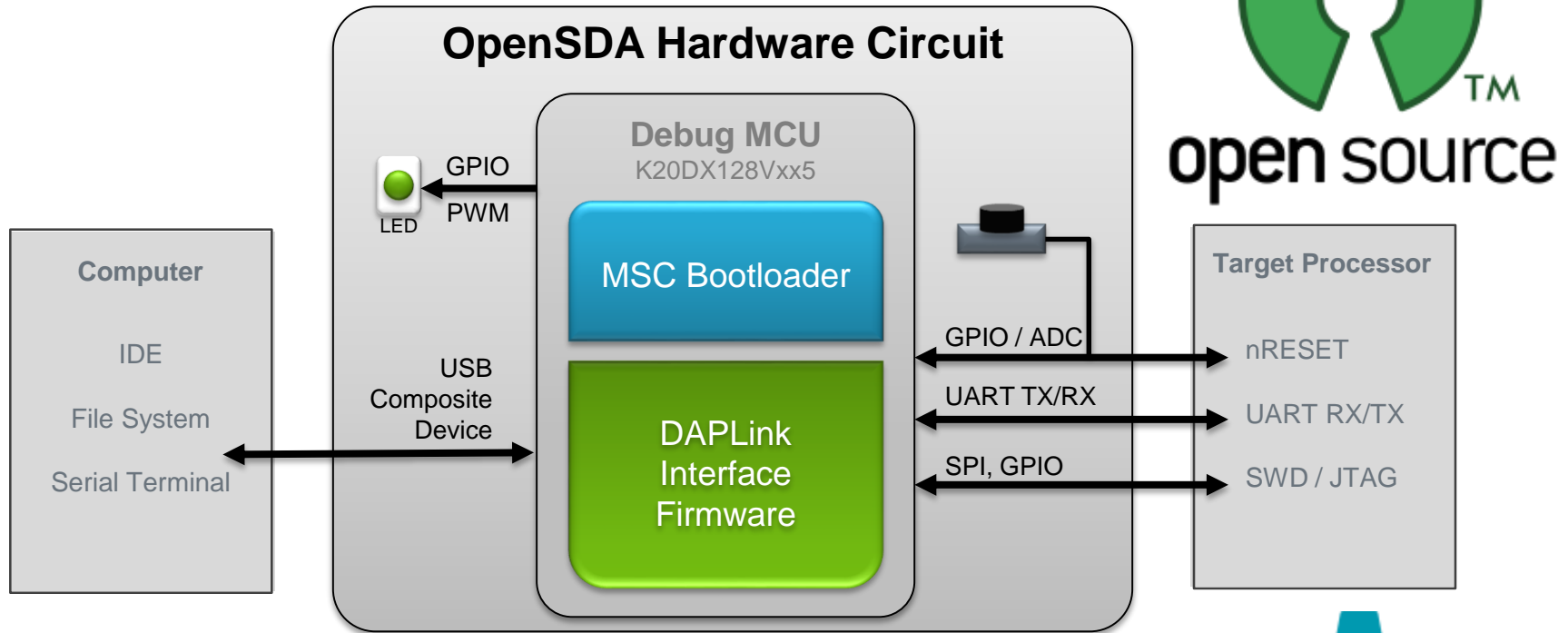
- Cortex-M4, 48MHz
- 128KB Flash, 16KB RAM, 32KI



FRDM-K22F

- Cortex-M4, 120MHz
- 512KB Flash, 128KB RAM
- Crystal-less USB OTG

OpenSDA & DAPLink Interface Firmware

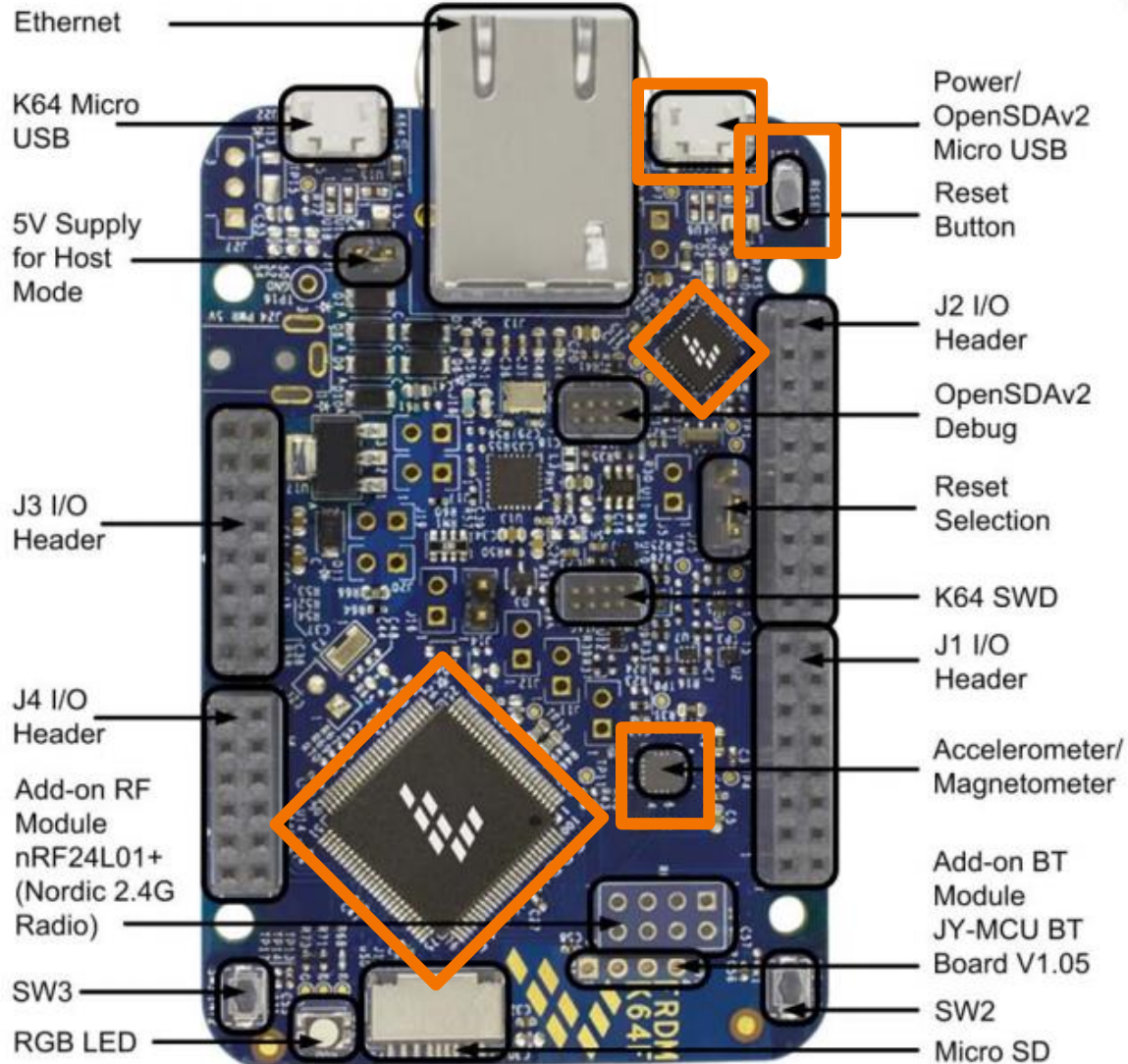


- **DAPLink Interface Firmware includes:**
 - USB HID CMSIS-DAP Run-control debug interface
 - USB MSC disk for drag 'n' drop flash programming
 - USB CDC serial interface between the host and target

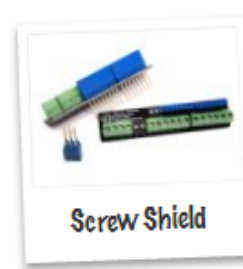
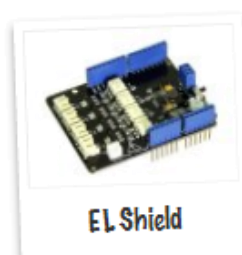


FRDM-K64F Overview

- **Quick, simple development experience with rich features**
 - Easy access to MCU I/O
 - 3-axis **accelerometer**/3-axis **magnetometer**
 - RGB LED
 - Add-on **Bluetooth** Module
 - Built-in Ethernet/Add-on **Wireless** Module
 - Micro SD
- **Arduino** shield compatible
- Flash programming functionality enabled by OpenSDA debug interface
- \$29.00



Arduino Shields

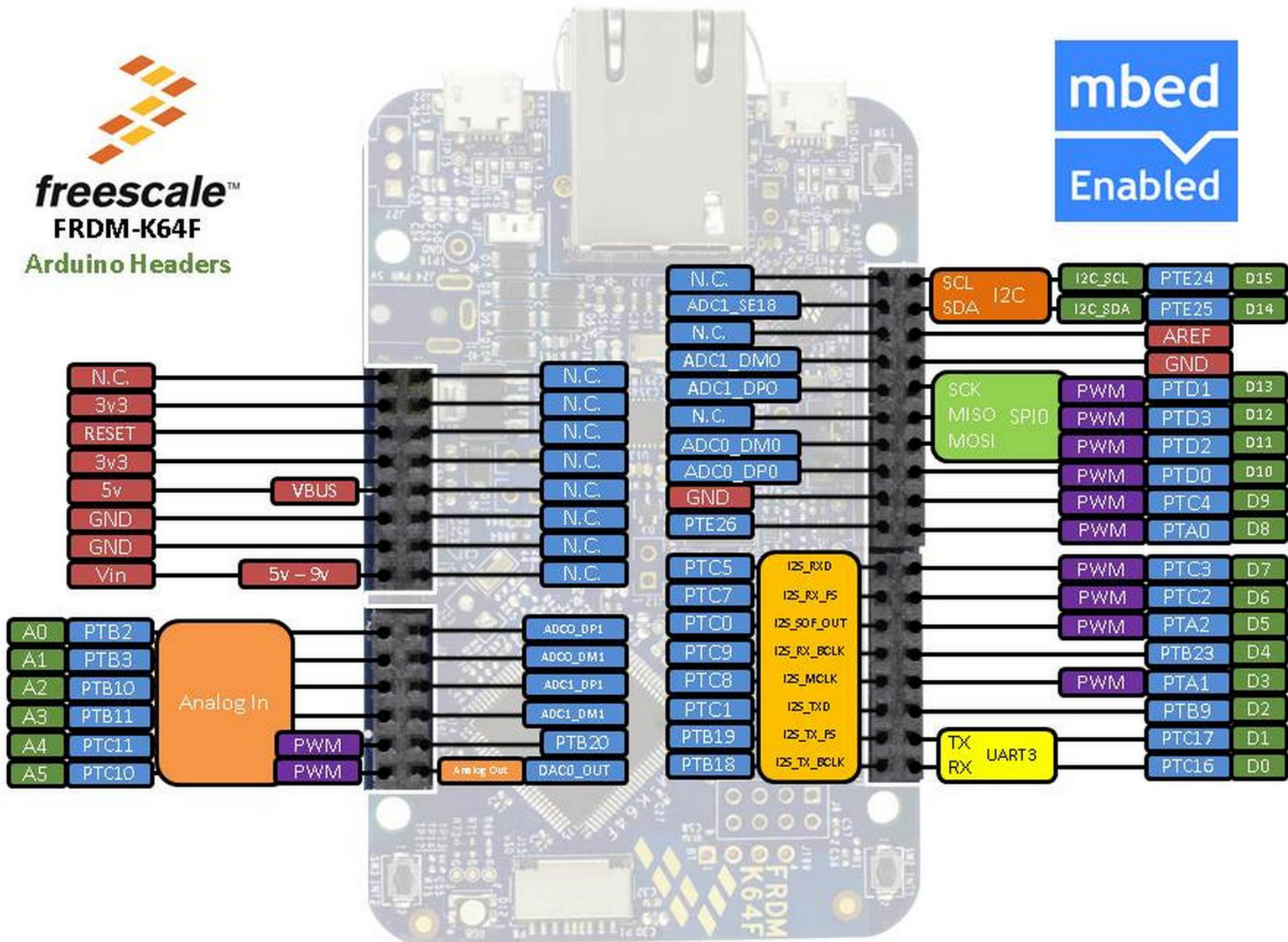




freescale™

FRDM-K64F

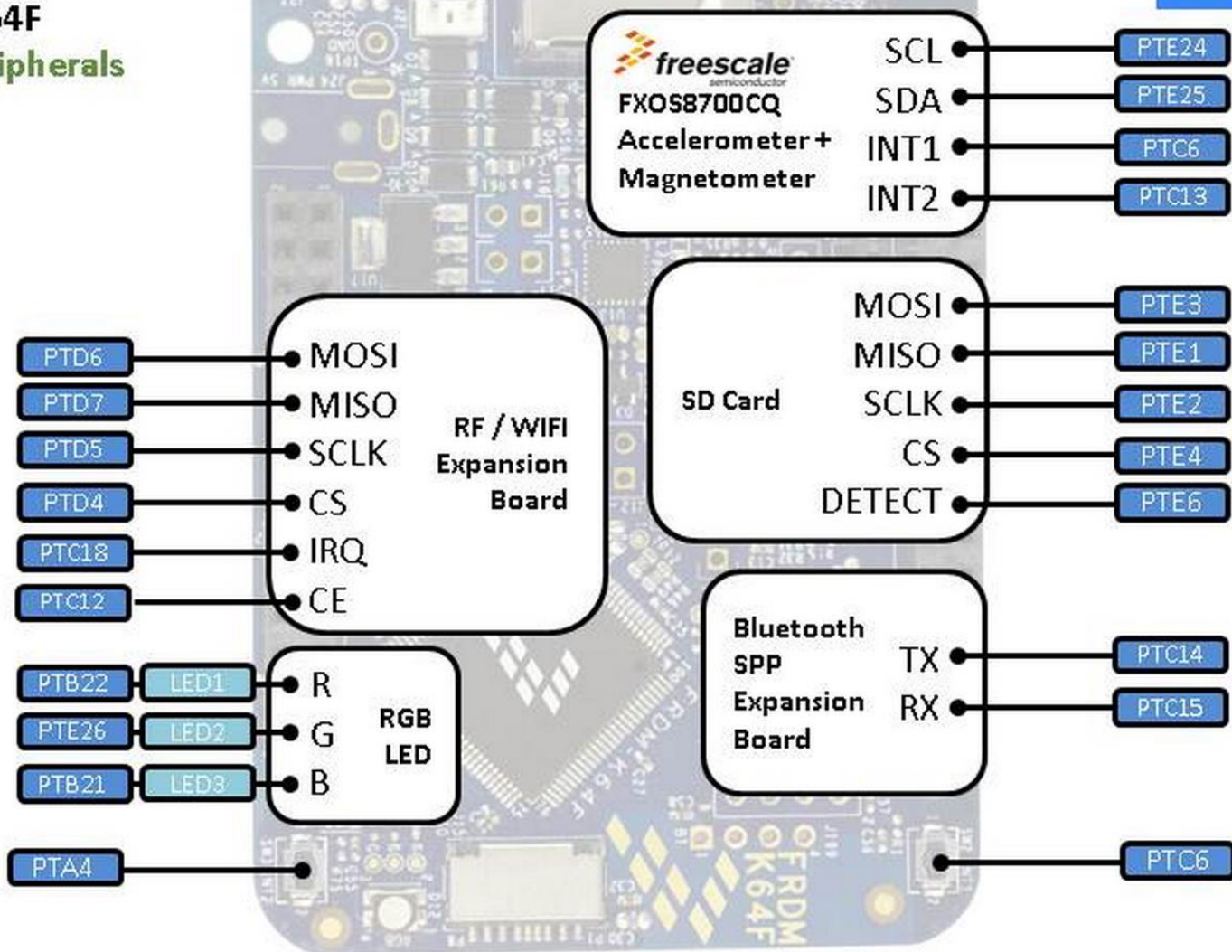
Arduino Headers





freescale™
FRDM-K64F

Additional Peripherals

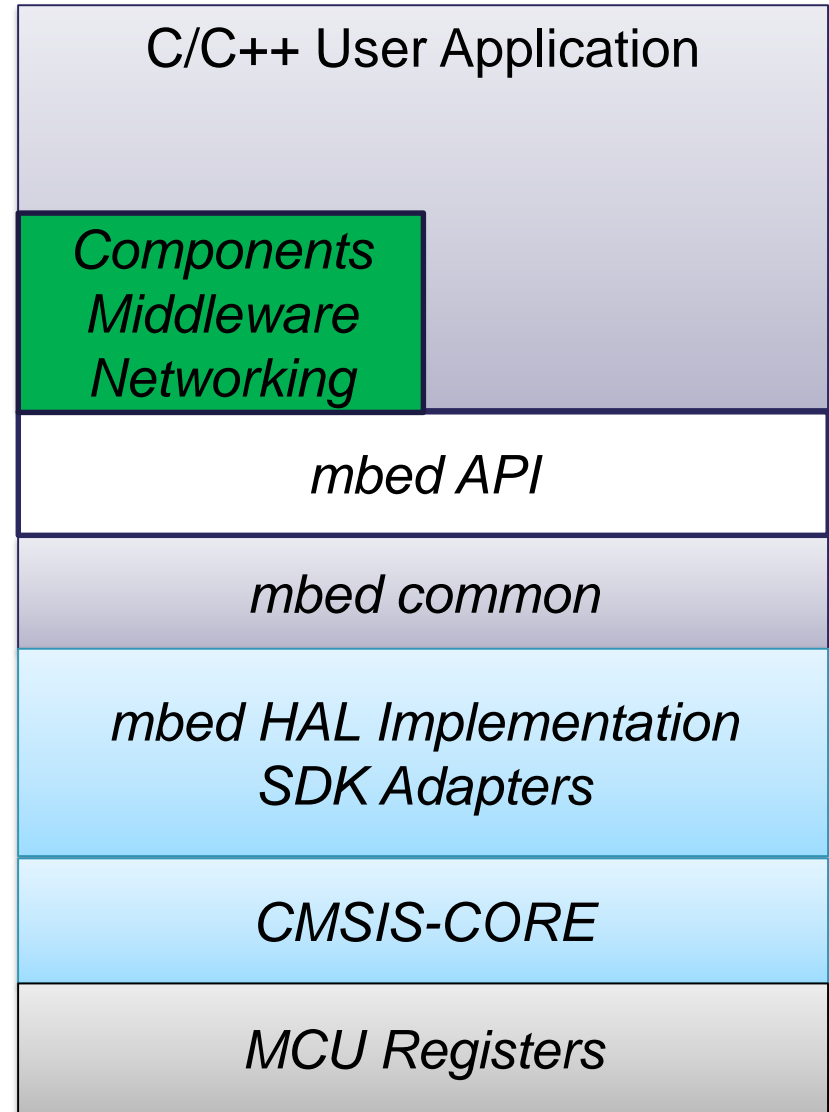




mbed Software

mbed SDK Software Stack

- Networking and USB stacks
- CMSIS-RTOS implementation
- Easy-to-use C++ APIs
- stdlib setup, board support, systems configuration
- Hardware Abstraction Layer (HAL) for MCU peripherals
- CMSIS-CORE: hardware register access and Cortex-M startup code



mbed Program Example

```
1 #include "mbed.h"
2
3 DigitalOut myled(LED1);
4
5 int main() {
6     while(1) {
7         myled = 1;
8         wait(0.2);
9         myled = 0;
10        wait(0.2);
11    }
12 }
```

- Hello World example
 - All startup code is taken care of by the mbed SDK – user code starts at main()
 - Definition of a digital output variable, myled
 - Use of overloaded “=” operator
 - Alter digital output (and LED) by simply assigning a value to the variable.

- [mbed Memory Model](#) - The memory model used by the mbed Library

Analog I/O

- [AnalogIn](#) - Read the voltage applied to an analog input pin
- [AnalogOut](#) - Set the voltage of an analog output pin

Digital I/O

- [DigitalIn](#) - Configure and control a digital input pin.
- [DigitalOut](#) - Configure and control a digital output pin.
- [DigitalInOut](#) - Bi-directional digital pins

- [BusIn](#) - Flexible way to read multiple DigitalIn pins as one value
- [BusOut](#) - Flexible way to write multiple DigitalOut pins as one value
- [BusInOut](#) - Flexible way to read/write multiple DigitalInOut pins as one value

- [PortIn](#) - Fast way to read multiple DigitalIn pins as one value
- [PortOut](#) - Fast way to write multiple DigitalOut pins as one value
- [PortInOut](#) - Fast way to read/write multiple DigitalInOut pins as one value

- [PwmOut](#) - Pulse-width modulated output

- [InterruptIn](#) - Trigger an event when a digital input pin changes.

Timers

- [Timer](#) - Create, start, stop and read a timer
- [Timeout](#) - Call a function after a specified delay
- [Ticker](#) - Repeatedly call a function

- [wait](#) - Wait for a specified time
- [time](#) - Get and set the realtime clock

Digital Interfaces

- [Serial](#) - Serial/UART bus

- [SPI](#) - SPI bus master
- [SPISlave](#) - SPI bus slave

- [I2C](#) - I²C bus master
- [I2CSlave](#) - I²C bus slave

- [CAN](#) - Controller-area network bus

Real-time Operating System

- [mbed RTOS](#)

File System

- [LocalFileSystem](#) - Using the mbed disk as storage from within a program
- [SDFFileSystem](#) - Using the mbed disk as storage from within a program

USB

- [USBDevice](#) - Using mbed as a USB Device
 - [USBMouse](#) - Emulate a USB Mouse with absolute or relative positioning
 - [USBKeyboard](#) - Emulate a USB Keyboard, sending normal and media control keys
 - [USBMouseKeyboard](#) - Emulate a USB Keyboard and a USB mouse with absolute c
 - [USBHID](#) - Communicate over a raw USBHID interface, great for driverless commur
 - [USBMIDI](#) - Send and recieve MIDI messages to control and be controlled by PC m
 - [USBSerial](#) - Create a virtual serial port over the USB port. Great to easily communi
 - [USBAudio](#) - Create a USBAudio device able to receive audio stream from a compu
 - [USBMSD](#) - Generic class which implements the Mass Storage Device protocol in o

- [USBHost](#) - Using mbed to act as USBHost
 - [USBHostMouse](#) - Receive events from a USB mouse
 - [USBHostKeyboard](#) - Read keycode-modifier from a USB keyboard
 - [USBHostMSD](#) - Read-write a USB flash disk
 - [USBHostSerial](#) - Communicate with a virtual serial port
 - [USBHostHub](#) - You can plug several USB devices to an mbed using a USB hub

Digital Inputs and Outputs

mbed - DigitalInOut Class Reference

Public Member Functions

DigitalInOut (PinName pin)
Create a **DigitalInOut** connected to the specified pin.

DigitalInOut (PinName pin, PinDirection direction, PinMode mode,
Create a **DigitalInOut** connected to the specified pin.

void **write** (int value)
Set the output, specified as 0 or 1 (int)

int **read** ()
Return the output setting, represented as 0 or 1 (int)

void **output** ()
Set as an output.

void **input** ()
Set as an input.

void **mode** (PinMode pull)
Set the input pin mode.

int **is_connected** ()
Return the output setting, represented as 0 or 1 (int)

DigitalInOut & **operator=** (int value)
A shorthand for **write()**

operator int ()
A shorthand for **read()**

Boolean logic - NOT, AND, OR, XOR

```
1 #include "mbed.h"
2
3 DigitalIn a(p5);
4 DigitalIn b(p6);
5 DigitalOut z_not(LED1);
6 DigitalOut z_and(LED2);
7 DigitalOut z_or(LED3);
8 DigitalOut z_xor(LED4);
9
10 int main() {
11     while(1) {
12         z_not = !a;
13         z_and = a && b;
14         z_or = a || b;
15         z_xor = a ^ b;
16     }
17 }
```

Ticker

mbed - Ticker Class Reference

Public Member Functions

void **attach** (void(*fptr)(void), float t)
Attach a function to be called by the **Ticker** , specifying the interval.

template<typename T >
void **attach** (T *tptr, void(T::*mptr)(void), float t)
Attach a member function to be called by the **Ticker** , specifying the interval.

void **attach_us** (void(*fptr)(void), timestamp_t t)
Attach a function to be called by the **Ticker** , specifying the interval in microseconds.

template<typename T >
void **attach_us** (T *tptr, void(T::*mptr)(void), timestamp_t t)
Attach a member function to be called by the **Ticker** , specifying the interval in microseconds.

void **detach** ()
Detach the function.

Static Public Member Functions

static void **irq** (uint32_t id)
The handler registered with the underlying timer interrupt.









































Protected Attributes

timestamp_t **_delay**
Time delay (in microseconds) for re-setting the multi-shot callback.

FunctionPointer **_function**
Callback.

```
1 #include "mbed.h"
2
3 // A class for flip()-ing a DigitalOut
4 class Flipper {
5 public:
6     Flipper(PinName pin) : _pin(pin) {
7         _pin = 0;
8     }
9     void flip() {
10         _pin = !_pin;
11     }
12 private:
13     DigitalOut _pin;
14 };
15
16 DigitalOut led1(LED1);
17 Flipper f(LED2);
18 Ticker t;
19
20 int main() {
21     t.attach(&f, &Flipper::flip, 2.0); //
22
23     // spin in a main loop. flipper will
24     while(1) {
25         led1 = !led1;
26         wait(0.2);
27     }
28 }
```

http://developer.mbed.org/ **components**

| Component | Short Description | Notes | Image | Component | Short Description | Notes | Image | Component | Short Description | Notes | Image |
|---|-------------------|-------|---|---|-------------------|-------|---|------------------------------------|-------------------|-------|---|
| 1.3" SHARP Memory LCD | ✓ | ✗ |  | DisplayModule 2.2" TFT with 8-bit interface | ✓ | ✓ |  | Grove - Air Quality Sensor | ✓ | ✓ |  |
| 128x32 LCD | ✓ | ✓ |  | DisplayModule 2.4" Touch TFT with 8-bit Interface | ✓ | ✓ |  | Grove - Alcohol Sensor | ✓ | ✓ |  |
| 2.2 QVGA display with SD card socket | ✓ | ✓ |  | DisplayModule 2.8" Touch TFT with 8-bit Interface | ✓ | ✓ |  | Grove - Barometer Sensor | ✓ | ✓ |  |
| 2.7 inch E-paper display | ✓ | ✗ |  | DisplayModule 2.8" Touch TFT with SPI and 4MB Flash | ✓ | ✓ |  | Grove - Button | ✓ | ✓ |  |
| 24LCxx Serial EEPROM library | ✓ | ✓ |  | DisplayModule 2.8" Touch TFT with SPI and 4MB Flash | ✓ | ✓ |  | Grove - Buzzer | ✓ | ✓ |  |
| 25LCxxx SPI | ✓ | ✓ |  | DisplayModule 3.5" Touch TFT with SPI and 4MB Flash | ✓ | ✓ |  | Grove - Collision Sensor | ✓ | ✓ |  |
| 4D SGC TFT Screen | ✓ | ✓ |  | DMU02 Dynamics Measurement Unit | ✓ | ✓ |  | Grove - Colour Sensor | ✓ | ✗ |  |
| 4D Systems 128 by 128 Smart Color LCD uLCD-144-G2 | ✓ | ✓ |  | DORJI Data Radio Modem (433Mhz) | ✓ | ✓ |  | Grove - Digital Light Sensor | ✓ | ✓ |  |
| ACM1602NI-FLW-FBW-M01 | ✓ | ✓ |  | DS1302 Timekeeping Chip | ✓ | ✓ |  | Grove - Ear-clip Heart Rate Sensor | ✓ | ✓ |  |
| AD8556 | ✓ | ✓ |  | DS1307 RTC | ✓ | ✓ |  | Grove - Electricity Sensor | ✓ | ✓ |  |
| Adafruit / SSD1306 OLED 128x32 or 128x64 | ✓ | ✗ |  | DS1721 | ✓ | ✗ |  | Grove - HCHO Sensor | ✓ | ✓ |  |
| Adafruit NeoPixels (WS2812) | ✓ | ✓ |  | DS1820 | ✓ | ✓ |  | Grove - I2C Touch Sensor | ✓ | ✓ |  |
| Adafruit Ultimate GPS Breakout v3 | ✓ | ✓ |  | EA DOGS102-6 Graphic LCD | ✓ | ✓ |  | Grove - Moisture Sensor | ✓ | ✓ |  |
| | | | | | | | | Grove - PIR Motion Sensor | ✓ | ✓ |  |

Component Database

The screenshot shows the mbed Component Database website. The navigation bar includes links for Platforms, Components (highlighted), Handbook, Cookbook, Code, Questions, and Forum. On the right, there are links for Dashboard and Compiler, and a user profile for sam_grove with a Logout button. A search bar is highlighted with a red box, containing the text "Search components on mbed.org..." and a "Go" button. Below the search bar, a sidebar on the left is also highlighted with a red box, listing various component categories with their respective counts: Actuators (6), Communication (24), Display (39), Expansion boards (16), Internet of Things (7), Online Services (2), and Robotics (7). The main content area is titled "Components" and features a grid of ten component categories, each with an image and a label: Actuators, Communication, Display, Expansion boards, Internet of Things, Online Services, Robotics, Sensors, Storage, and Other. A blue button labeled "Add a component" is located in the top right corner of the main content area.

Components

Actuators (6)
Motor (3)
Servomotor (3)
Solenoid (0)

Communication (24)
Bluetooth (2)
CAN (1)
Cellular (4)
Ethernet (3)
Infrared (1)
NFC (1)
RFID (1)
Wifi (4)

Display (39)
LCD (16)
LED Controller (11)
Touchscreen (5)

Expansion boards (16)

Internet of Things (7)

Online Services (2)

Robotics (7)

Components

The Component Database hosts reusable libraries for different hardware, middleware and IoT services that you can use with ARM Microcontrollers. These components can be used as building blocks for quickly developing prototypes and products.

Components and the associated libraries, examples and documentation are created and added to the database by mbed developers, component manufacturers and service providers. The goal is to create a canonical database of rock-solid code and resources for every useful component that can be used with ARM microcontrollers.

Actuators

Communication

Display

Expansion boards

Internet of Things

Online Services

Robotics

Sensors

Storage

Other

Add a component

Components are portable across all platforms and tools

Component Entry

Components » Sensors » FXOS8700Q Accelerometer / Magnetometer

FXOS8700Q Accelerometer / Magnetometer

This is a 6 axis combination Accelerometer / Magnetometer

Hello World

Hello_FXOS8700Q

Example program for FXOS8700Q sensor

Last commit 4 days ago by [Freescale](#)

Import program

Library

FXOS8700Q

Driver library for the Freescale FXOS8700Q sensor

Last commit 3 days ago by [Freescale](#)

Import library

Delete Edit this component



Follow this component

Follow

Tested platforms



freescaler
K64F
FRDM-



freescaler
IoT Starter Kit
Ethernet

Example program to evaluate the component

Directly import into Your current program



Public Member Functions

FXOS8700Q

| | |
|------------------|--|
| virtual void | enable (void) const =0 Enable the sensor for operation. |
| virtual void | disable (void) const =0 disable the sensors operation |
| virtual uint32_t | sampleRate (uint32_t frequency) Set the sensor sample rate. |
| virtual uint32_t | dataReady (void) const =0 Tells of new data is ready. |
| virtual int16_t | getX (int16_t &x) const =0 Get the x data in counts. |
| virtual int16_t | getY (int16_t &y) const =0 Get the y data in counts. |
| virtual int16_t | getZ (int16_t &z) const =0 Get the z data in counts. |
| virtual float | getX (float &x) const =0 Get the x data in units. |
| virtual float | getY (float &y) const =0 Get the y data in units. |
| virtual float | getZ (float &z) const =0 Get the z data in units. |
| virtual void | getAxis (motion_data_counts_t &xyz) const =0 Get the x,y,z data in counts. |
| virtual void | getAxis (motion_data_units_t &xyz) const =0 Get the x,y,z data in units. |

```
#include "mbed.h"
#include "FXOS8700Q.h"
I2C i2c(PTE25, PTE24);
FXOS8700QAccelerometer acc(i2c, FXOS8700CQ_SLAVE_ADDR1);
FXOS8700QMagnetometer mag(i2c, FXOS8700CQ_SLAVE_ADDR1);
int main(void)
{
    motion_data_units_t acc_data, mag_data;
    motion_data_counts_t acc_raw, mag_raw;
    float faX, faY, faZ, fmX, fmY, fmZ, tmp_float;
    int16_t raX, raY, raZ, rmX, rmY, rmZ, tmp_int;
    acc.enable();
    mag.enable();
    while (true) {
        // counts based results
        acc.getAxis(acc_raw);
        mag.getAxis(mag_raw);
        acc.getX(raX);
        acc.getY(raY);
        acc.getZ(raZ);
        mag.getX(rmX);
        mag.getY(rmY);
        mag.getZ(rmZ);
        // unit based results
        acc.getAxis(acc_data);
        mag.getAxis(mag_data);
        acc.getX(faX);
        acc.getY(faY);
        acc.getZ(faZ);
        mag.getX(fmX);
        mag.getY(fmY);
        mag.getZ(fmZ);
        wait(0.1f);
    }
}
```



Table of Contents

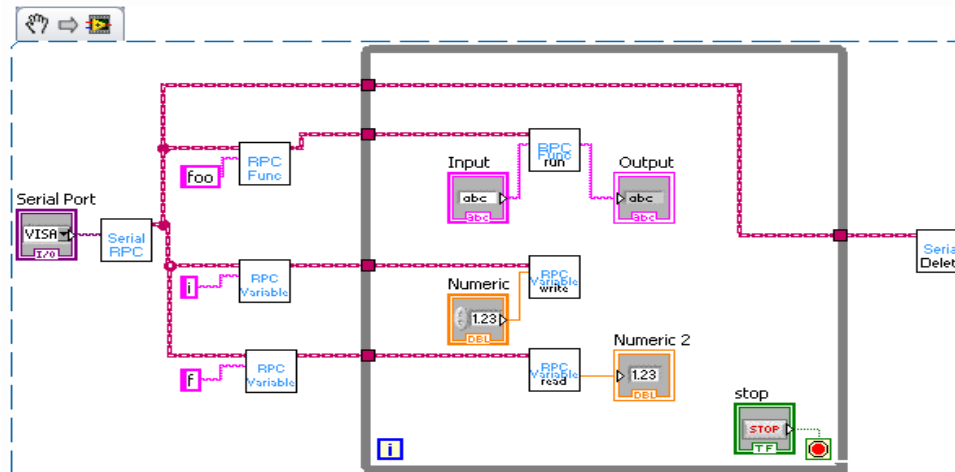
1. Introduction and Help
2. Notebooks
3. Components and Libraries
4. TCP/IP Networking
5. USB
6. LCDs and Displays
7. Audio
8. Wireless
9. Motors and Actuators
10. Sensors
11. Cameras
12. Accelerometer
13. Inclinometers
14. Compass
15. NFC/RFID
16. Barcode
17. Temperature
18. Clocks and Oscillators
19. External ADC/DAC
20. Interfaces and Drivers
21. Storage, Smart Cards
22. Magnetic, Proximity Card Readers
23. Digital Signal Processing

Interfacing with other languages

- a Forth implementation [MbedForth](#) with ready to flash .bin
- [Interfacing Using RPC](#) (Remote Procedure Call)
- [Interfacing with Matlab](#)
- [Interfacing with Python](#)
- [Interfacing with LabVIEW](#)
- [Interfacing with Java](#)
- [Interfacing with JavaScript](#)
- [Assembly Language](#) - Using Assembly Language and how to debug using the free Keil Tools ARM emulator
- [Python-on-a-chip](#)
- [Interfacing with .NET](#)

Using RPC with Custom Code

The [RPC-Interface-Library](#) provides a mechanism for quickly adding RPC functionality to your own code. This LabVIEW library includes support for the [RPCFunction](#) and [RPCVariable](#) Objects. You can tie an [RPCFunction](#) or [RPCVariable](#) object to the corresponding object on mbed. You can then run the function or read and write to the variables which are attached.

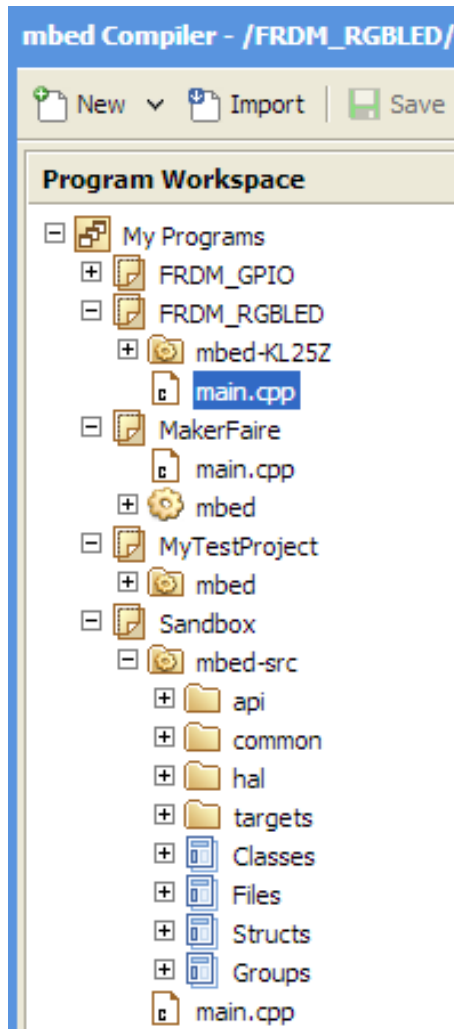


The [RPCVariable](#) read and write blocks have both numeric and string inputs and outputs. You should wire to the appropriate one according to the type of variable it corresponds with on mbed. The motor control IMU display demo at the bottom of the page show examples of these VIs in use.



mbed Tools

mbed Online IDE



- Cloud-based ARM professional C/C++ compiler
 - Private account based access
 - Pre-configured compiler that “just works”
 - Informative compile-time messages with links to error and error message wiki
 - Provides build information including RAM and flash usage in a graphical display
 - Generates binary: save to mbed enabled hardware
- Browser-based editor and project manager
 - Syntax highlighting
 - Keyboard shortcuts
 - Integrated project manager
 - Tabbed file organizer
 - Code auto-formatter
- Easily imports mbed software libraries and examples
- Runs on Windows, Mac iOS, Android, Linux



Online IDE and Compiler

Platform Selection

The screenshot shows the mbed IDE interface. The browser address bar displays `https://developer.mbed.org/compiler/#nav:/test_uart_echo/mbed.bld;`. The interface includes a menu bar with options like New, Import, Save, Save All, Compile, Commit, Revisions, and Help. A toolbar contains icons for file operations and version control. The main workspace is divided into three panels: a left sidebar for the Program Workspace, a central file browser, and a right sidebar for Library Build Details. The file browser shows a table of files and folders. The Library Build Details panel provides information about the selected library, including its name, URL, and version. The bottom status bar shows the current state of the IDE and the number of errors and warnings.

| Name | Size | Type | Modified |
|---------|------|-----------------------|-------------|
| Classes | | Classes Documentation | 28 Oct 2014 |
| Files | | Files Documentation | 28 Oct 2014 |
| Groups | | Grouped Documentati | 28 Oct 2014 |
| Structs | | Structs Documentation | 28 Oct 2014 |

| Name | URL | Last Updated | Revision | Status |
|------|---------------------------------|--------------|-----------------|-------------------------|
| mbed | Open build page | moments ago | 90:cb3d968589d8 | newer version available |

■ Programs Workspace

■ Integrated Version Control

■ Program/Library Details



Revision Control and Collaboration

- Built-in revision system
 - Mercurial DVCS
 - Collaboration workflow
- Local version control
 - **Commit** a version of your project, and view the revision history
 - **View** changes a version made, and **compare** changes between versions
 - **Switch** and **revert** to a different version
 - **Branch** and **merge** versions
- Collaborate with others
 - **Pull** - Copy changes from remote repository to a local repository in your workspace.
 - **Push** - Copy changes from local repository to a remote one.
 - **Fork** - Create remote repository on mbed.org from imported local repository
 - **Publish** - Publish changes to existing repository (push) or create new one (fork)
 - **Update** - Pull from a remote repository and switch your local repository to the latest revision.

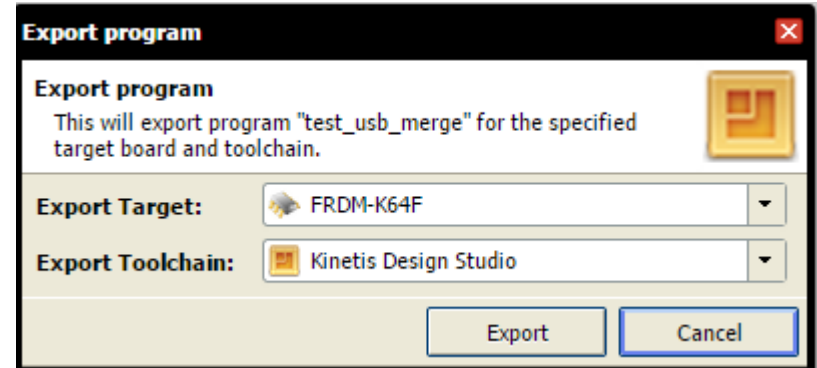
The screenshot shows the 'Revision History' window for the library 'cc3000_hostdriver_mbedsocket'. It includes a toolbar with actions like Commit, Discard, Changes, Compare, Switch, Revert, and Merge. Below the toolbar is a table of revisions with columns for Graph, Revisor, When, Who, and Comment. The current revision is 44, committed 2 weeks, 2 days ago by Kojto, with the comment 'default tip all _DEBUG_ are set to 0, to'. The table lists revisions 28 through 44, showing a linear history of updates.

| Graph | Revisor | When | Who | Comment | |
|-------|----------------------------------|------|---------------------|------------------|---|
| | <input checked="" type="radio"/> | 44 | 2 weeks, 2 days ago | Kojto | default tip all _DEBUG_ are set to 0, to |
| | <input type="radio"/> | 43 | 2 weeks, 2 days ago | SolderSplashLabs | Disabled debugging |
| | <input type="radio"/> | 42 | 2 weeks, 2 days ago | SolderSplashLabs | Added David's IRQ checking before re-enabli |
| | <input type="radio"/> | 41 | 2 weeks, 2 days ago | SolderSplashLabs | Removed a debug timer, added debug messa |
| | <input type="radio"/> | 40 | 2 weeks, 2 days ago | SolderSplashLabs | Merged with Martin's |
| | <input type="radio"/> | 39 | 2 weeks, 2 days ago | SolderSplashLabs | Added mbed ethernet library compatability, g |
| | <input type="radio"/> | 38 | 3 weeks, 3 days ago | SolderSplashLabs | Enabling debugging |
| | <input type="radio"/> | 37 | 2 weeks, 3 days ago | Kojto | GCC conflict with FD_SET |
| | <input type="radio"/> | 36 | 2 weeks, 3 days ago | Kojto | smart config start - erased delete all profiles |
| | <input type="radio"/> | 35 | 3 weeks, 3 days ago | Kojto | disable debug messages by default |
| | <input type="radio"/> | 34 | 3 weeks, 3 days ago | Kojto | doxygen comments added by Frank V., htonl |
| | <input type="radio"/> | 33 | 3 weeks, 3 days ago | Kojto | Removed code (set buffers) - non-sense, do: |
| | <input type="radio"/> | 32 | 3 weeks, 3 days ago | Kojto | TCP/UDP client/server old implementation ren |
| | <input type="radio"/> | 31 | 3 weeks, 3 days ago | Kojto | Doxygen comments above each class |
| | <input type="radio"/> | 30 | 3 weeks, 3 days ago | Kojto | TCP/UDP client/server removal (my own vers |
| | <input type="radio"/> | 29 | 3 weeks, 3 days ago | Kojto | TCP accept - big endian s_addr, doxygen cor |
| | <input type="radio"/> | 28 | 3 weeks, 4 days ago | Kojto | Commented select on UDP write (send), retu |



Exporting to Offline Tools

- The mbed Compiler can export to professional toolchains:
 - Kinetis Design Studio
 - Keil uVision
 - ARM DS-5
 - GCC
 - IAR Embedded Workbench
 - Coocox
 - Emblocks
 - Zip with repositories
- Why export?
 - Run-control debug
 - Optimizations
 - Automated testing





mbed Support

mbed Community

Vendors



Developers



Cloud services



Manufacturers



Components

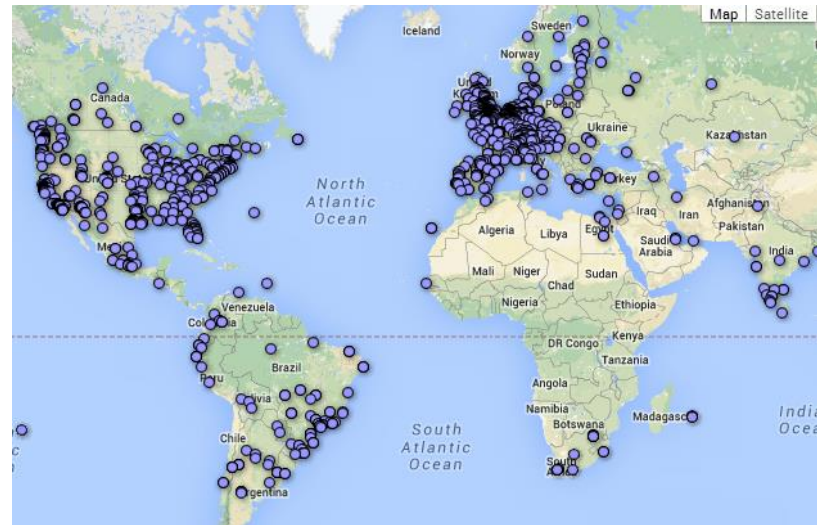


Operators



mbed Community Support

- Worldwide community of developers
- Shared context of
 - common tools
 - common SDK
 - common hardware
 - common experiences



- Issues can be reproduced and resolved efficiently



Hands-On

Hands-On Agenda

- Lab 1 – Hello World
 - Input / Output and serial module
 - Challenge – Change LED state based on button state
 - Challenge – Read serial characters and change RGB LED state
- Lab 2 – Interrupts and Timers
 - Interrupts and timers
 - Challenge – Drive RGB LED while sleeping between state change
 - Challenge – Change RGB LED state but sleep between state change
- Lab 3 – Using Sensors
 - I2C accelerometer / magnetometer
 - Challenge – Control an LED in a meaningful way based on the sensor readings
 - Challenge – Add sensor handling using the RTOS

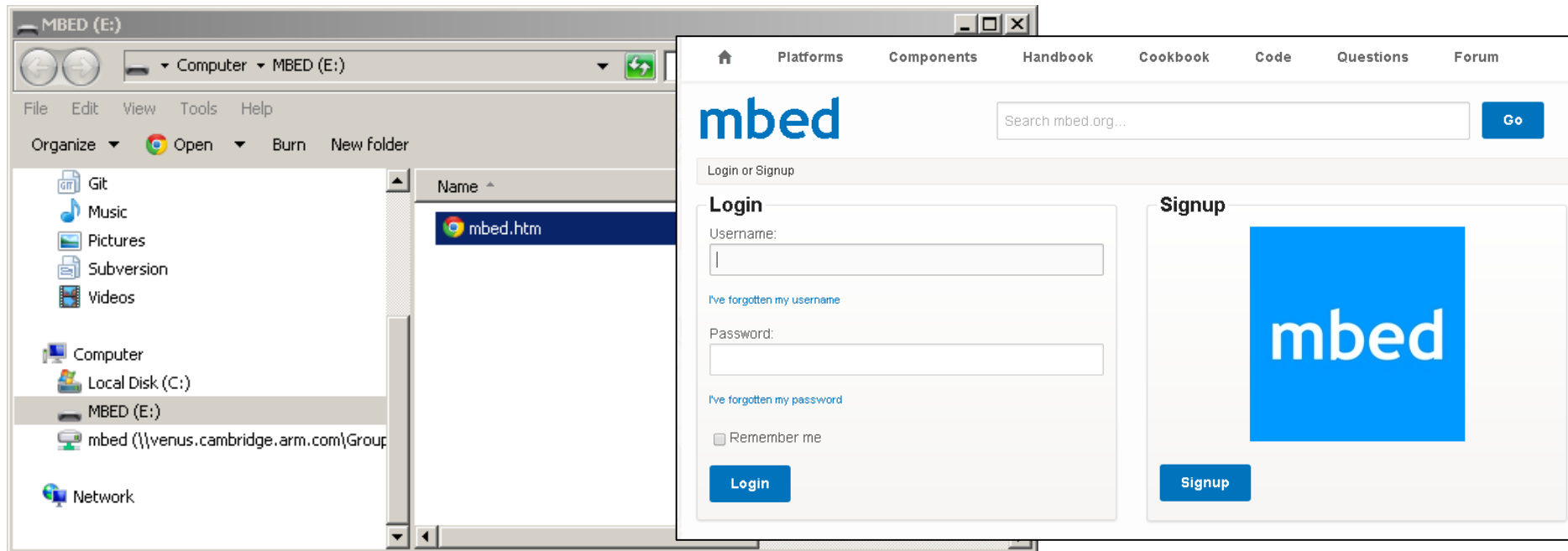
<http://mbed.org/ftf2015>



Creating an Account

■ Registration

1. Connect a mbed platform to a Windows / Mac / Linux computer
2. mbed platforms is identified as a mass storage device (USB disk)
3. Double-click the mbed.htm file on the mbed USB disk
4. Log in or sign up for a new account



Know your Hardware

- Connection diagram and example programs on the platform page

Overview

The Flagship FRDM-K64F has been designed by Freescale in collaboration with mbed for prototyping all sorts of devices, especially those requiring the size and price point offered by Cortex-M4®. It is packaged as a development board with connectors to break out to strip board and breadboard, and includes a built-in USB FLASH programmer.

Table of Contents

1. Overview
2. Features
3. Where to buy
4. Firmware
5. Getting Started with mbed
6. PC Configuration
7. Downloading A program
8. Hello World!
9. Where Next
10. Technical Reference
11. K64F Freedom Sensor Libraries and Examples

[Buy Now](#)

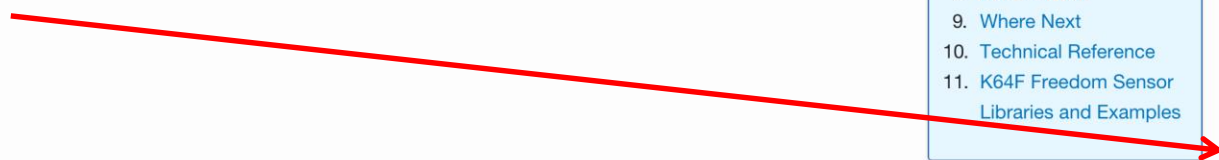
Remove from your mbed Compiler

You have this platform in your mbed Compiler, if you do not use it you can remove it.

[Remove](#)

[Following](#)

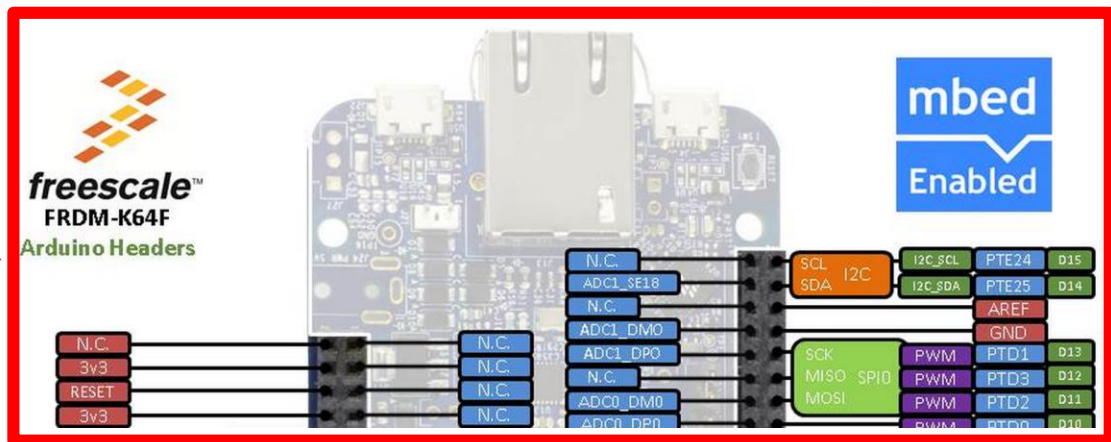
Example Program



Example programs

- frdm_gpio** (12005) - gpio example for the Freescale freedom platform
Last updated: 16 Jul 2014
- frdm_serial** (3009) - frdm serial example for the Freescale freedom platform
Last updated: 16 Jul 2014
- frdm_ticker** (791) - frdm ticker example for the Freescale freedom platform

Connection Diagram



About Programs

The example program for mbed pin-compatible platforms

 **Dependencies:**  mbed

[Home](#) [History](#) [Graph](#) [API Documentation](#) [Wiki](#) [Pull Requests](#) [Admin settings](#)

You can edit this area!

Download repository: [zip](#) [gz](#)

[Edit repository homepage](#)

Files at revision 6:e8cd76f38fa9

/ [default](#) [tip](#)

| Name | Size | Actions |
|--------------------------|------|--|
| ↑ [up] | | |
| main.cpp | 168 | Revisions Annotate |
| mbed.bld | 53 | Revisions Annotate |

[Ask a question](#)

[Start a discussion](#)

Discussion topics

Nothing here yet!

Questions

Nothing here yet!

Repository toolbox

[Import this program](#)

[Export to desktop IDE](#)

[Build repository](#)

[Send Pull Request from here](#)

[Make featured](#)

[Following](#)

[Embed url:](#)

<<program /teams/mbed/cod

[Clone repository to desktop:](#)

hg clone https://sam_grove@

Repository details

Type:  Program

Created: 11 Oct 2013


Imports:  99080

Forks:  24

Commits:  7

Dependents:  0

Dependencies:  1

Followers:  134



Compiling your Program

The screenshot displays the mbed IDE interface for the project `/mbed_blinky/main.cpp`. The **Compile** button in the top toolbar is highlighted with a red box. A **Save As** dialog box is open, showing the file name `mbed_blinky_LPC111424.bin` and the save type `*.bin`. The **Program Details** panel on the right shows memory usage for Flash (9.9 kB, 31%) and RAM (0.2 kB, 2%). The **Compile output** panel at the bottom shows a **Success!** message, and the **Build Details** link is highlighted with a red box.

```
1 #include "mbed.h"
2
3 DigitalOut myled(LED1);
4
5 int main() {
6     while(1) {
7         myled = 1;
8         wait(0.2);
9         myled = 0;
10        wait(0.2);
11    }
12 }
13
```

| | Flash | RAM |
|-------|--------------|-------------|
| Usage | 9.9 kB (31%) | 0.2 kB (2%) |

| Description | Error Number | Message | File Name | Location |
|-------------|--------------|---------|-----------|----------|
| Success! | | | | |





www.Freescale.com