mbed
Hello World

Agenda

- Introduction to mbed
- Lab 1: mbed registration and Hello World demo
- Lab 2: Other IO
- Lab 3: Interfacing with sensors
- Lab 4: Output devices, a TextLCD
- Lab 5: Rapid prototyping, Build a datalogger
- Lab 6: Rapid Prototyping: Offline debug with CMSIS-DAP

mbed
Hello World!

Introduction to mbed

What's happening in Microcontrollers?

- Microcontrollers are getting cheap
  - 32-bit ARM Cortex-M0+ Microcontrollers < $0.50

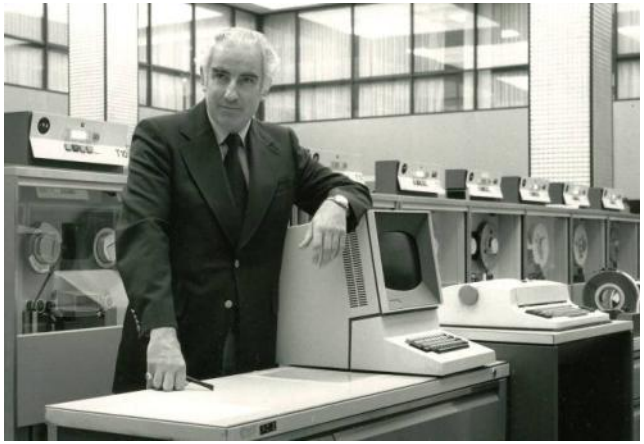
- Microcontrollers are getting powerful
  - Lots of processing, memory, I/O in one package

- Microcontrollers are getting interactive
  - Internet connectivity, new sensors and actuators

- Creates new opportunities for microcontrollers

Opportunities for Microcontrollers

- Before the 1980's computers were used and applied by computer scientists



- Now democratised for everyone to use
  - Office, home, entertainment, leisure
- Microcontroller technology is currently applied by professional embedded developers
- Microcontrollers interact with "the real world"
  - Sensors, actuators and communication, define their application
  - Their potential is greater than the home computer

Barriers for Microcontrollers

- What prevents microcontrollers from being designed in?
- Conceptually simple things can be hard to prototype
  - I want to send an SMS when my cat comes through the cat flap
- Repetition of choices to make:
  - Microcontroller, Tool chain
  - Dev board, Sensors
  - It's not difficult, but can be tedious and time consuming
- Overhead for starting a new project
  - Fine for a long complex projects
  - A deterrent for quick experiments and tests

Rapid Prototyping

- Rapid Prototyping helps industries create new products
  - Control, communication and interaction increasingly define products
  - Development cycles for microelectronics have not kept pace

3D Moulding

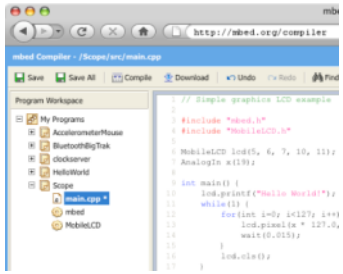
3D Printing

2D/3D Design

Web Frameworks

mbed.org - Rapid Prototyping for MCUs

- Fastest way to get started with ARM microcontrollers
  - Plug ‘n’ Play Hardware, Online Compiler
  - Get setup and run “Hello World!” very quickly
  - Removes entry barriers to MCU technology

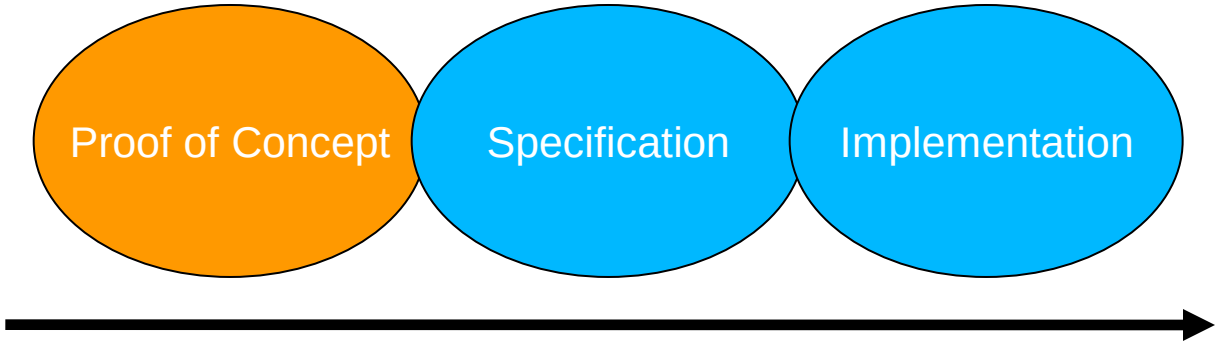


- Focused on rapid prototyping for a diverse audience
  - DIP form-factor, High-level APIs, Developer website
  - Technology and tradeoffs to enable fast experiments
  - Platform approach for developer ecosystem

- Launched at ESC Boston with live demo
  - Internet-enabled “Twittering Billy” read out tweets
  - An embedded internet device, prototyped in ½ day
  - Over ¼ million video views in first week!

mbed Approach

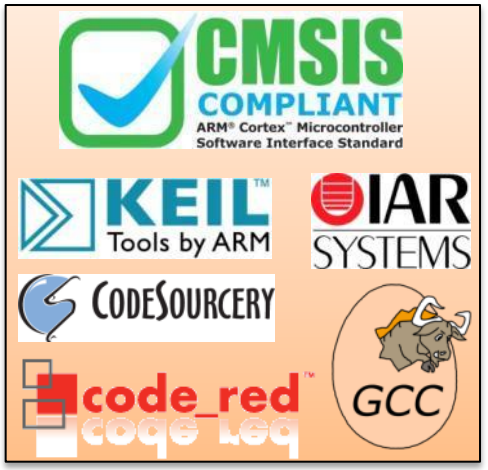
- Focus on tools supporting the earliest stage of design
  - Point of entry and Getting Started
  - Experimentation and Rapid Prototyping
  - Apply technology and trade-offs that support this goal



- Provide a platform that can built upon
  - HDK is a “recipe” enabling mbed features in 3rd party designs
  - SDK enables reuse and portability in the developer ecosystem

mbed SDK

- **C/C++ SDK for ARM Microcontrollers**
  - High-level APIs and standard environment
  - Low level control as needed
  - Portable across different ARM silicon vendor MCUs
- **Built on industry standard technology**
  - ANSI/ISO C/C++
  - CMSIS Compliant
  - Compatible with all major professional MCU tools
- **Open Source**
  - Released under permissive Apache 2.0 license
  - Suitable for commercial and non-commercial use
  - Managed, maintained and tested by ARM

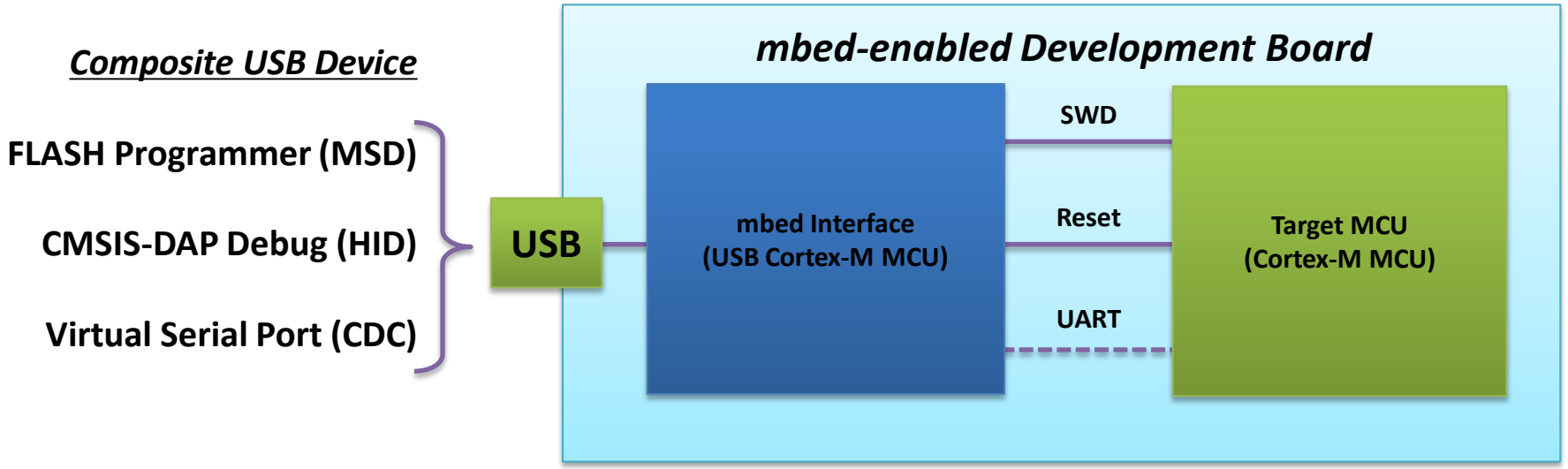


mbed HDK

- **HDK for ARM Microcontroller Boards**
  - MCU sub-system and debug interface architecture
  - Includes on-board USB interface with 3 endpoints:
    - Drag-and-drop FLASH programmer, Virtual Serial Port, Debug
  - Enables standard connection on low-cost boards, starter kits, modules
- **Built on industry standard technology**
  - USB Device
  - CMSIS-DAP Debug Interface Protocol
  - Compatible with Windows, Mac, Linux
- **Free for commercial and non-commercial use**
  - Portable across different ARM silicon vendor MCUs
  - Developed, tested and maintained by ARM

mbed HDK On-board Interface

- **On-board USB interface** for low-cost development boards
  - Supports **driverless MSD Programming and CMSIS-DAP Debug**
  - Interface implemented as firmware on selected Cortex-M MCUs



Enables simple USB drag-n-drop reprogramming of demo code through to full debug connection to ARM toolchains

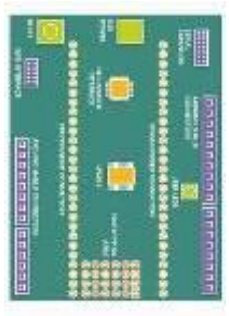
mbed-enabled Hardware

- **Expanding range of off-the-shelf mbed-enabled hardware**
  - ARM Cortex-M0, M0+, M3 all represented; M4 TBA
  - DIP prototyping modules, Arduino eval board form-factors
  - Available through worldwide distribution

$49

$45

$13



$20

- **Design and production**
  - All boards implement the mbed HDK
  - Hardware is designed and/or made by ARM or 3rd parties
  - mbed HDK enables anyone to build alternate board designs

Current message: mbed.org/explore

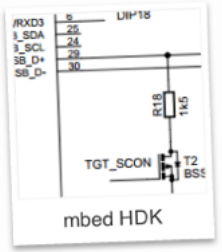
Software Development Kit (SDK)

The mbed Software Development Kit (SDK) is an open source C/C++ microcontroller software platform relied upon by tens of thousands of developers to build projects fast. We've worried about creating and testing startup code, C runtime, libraries and peripheral APIs, so you can worry about coding the smarts of your next product.

The SDK is licensed under the permissive Apache 2.0 licence, so you can use it in both commercial and personal projects with confidence.

The mbed SDK has been designed to provide enough hardware abstraction to be intuitive and concise, yet powerful enough to build complex projects. It is built on the low-level ARM CMSIS APIs, allowing you to code down to the metal if needed. In addition to RTOS, USB and Networking libraries, a cookbook of hundreds of reusable peripheral and module libraries have been built on top of the SDK by the mbed Developer Community.

mbed SDK | mbed RTOS



Hardware Development Kit (HDK)

The mbed Hardware Development Kit (HDK) provides full microcontroller sub-system design files and firmware for building development boards and custom products that benefit from the native support of the mbed SDK and free mbed Online Compiler and mbed Developer Platform.

The HDK specifies all support components and circuits including the mbed Onboard Interface design that provides simple USB drag-n-drop programming and CMSIS-DAP debug interface for the target microcontroller.

Development boards that are already based on the HDK are the quickest way to get started with the mbed platform. We manufacture official mbed Microcontroller modules that are specifically optimised for flexible rapid prototyping, and are available from distributors worldwide. Our partners are now also creating mbed-enabled hardware such as ultra low-cost ARM evaluation boards in the popular Arduino form-factor.

mbed HDK | mbed Microcontrollers

Free Online Development Tools

The mbed Compiler is a powerful online IDE that is free for use with hardware implementing the mbed HDK, and tightly integrated with the mbed SDK and Developer Website. Under the hood, it relies on the industry standard ARM professional C/C++ compiler, pre-configured and tested to generate fast, efficient code without fuss.

Login anywhere to get instant access to your development environment, on Windows, Mac, Linux. You can even work from tablets!

Whilst the mbed Compiler provides you your own private workspace, it is also fully integrated with the mbed.org Developer Website so you can easily import libraries and examples. If you choose to, publishing your own code and collaborating with other mbed users is just a few clicks too. The mbed Compiler also supports full export to different toolchains, incase your project demands it as you go to production.

Worldwide Developer Community

Using mbed means a huge shared context with other developers, and that means when you have a question, there is less pre-amble, less explanation and less time reproducing issues, and more time getting answers. We're proud that this has helped us grow an active and friendly community of skilled developers that are collectively helping get prototypes made even faster.

But where it really gets interesting is with code. Our developers are sharing thousands of open source repositories and building an extensive cookbook of recipes that you can reuse to build your products.

We've also made contributing back is easy; you can publish a library to mbed.org with a few clicks in the IDE, and let others build on your hard work. In fact, this is how some of our users end up collaborating on hard problems, and even getting contract work.

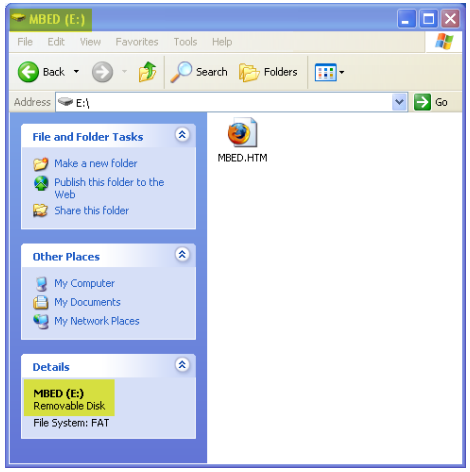
mbed
Hello World

Lab 1

mbed registration and hello world!

Registration

- For these lab sessions, we are using :
  - mbed LPC1768
  - mbed Application board

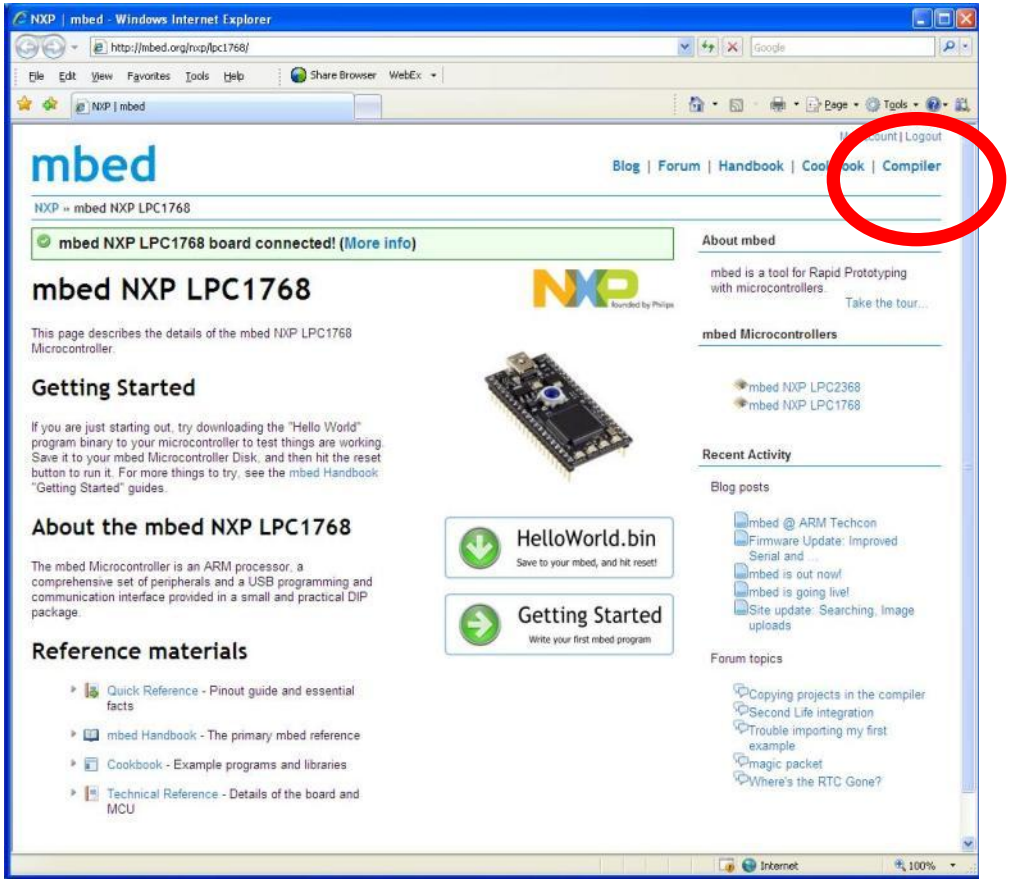


- mbed microcontrollers enumerate as a Mass Storage Device (USB disk)
- Double-click the mbed.htm file on the mbed USB disk

- Log in or sign up for a new account

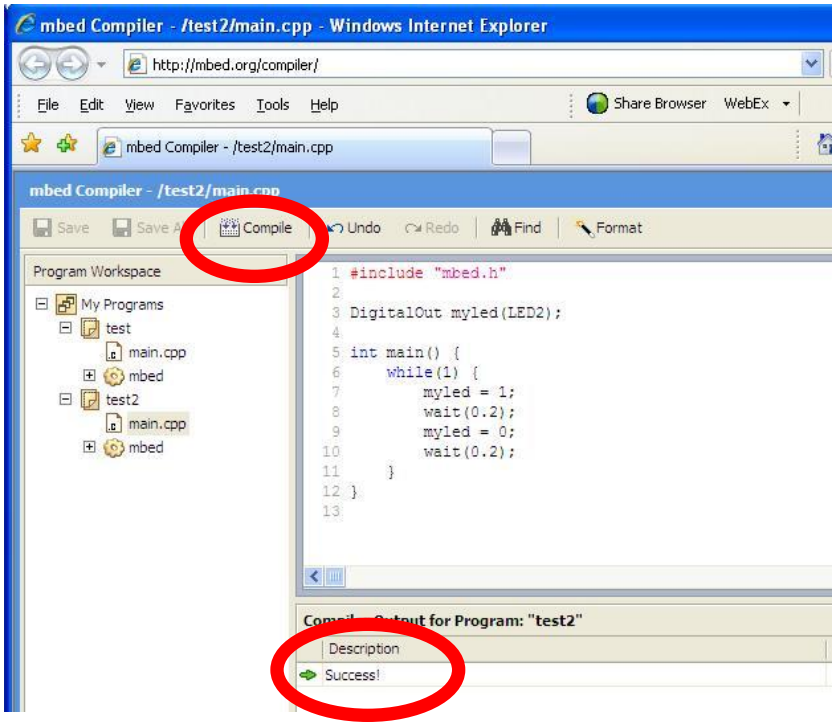
Getting Started

- Useful resources linked from the first page, including very clear links to “Hello World” and the Getting Started guide
- Compiler linked from front page



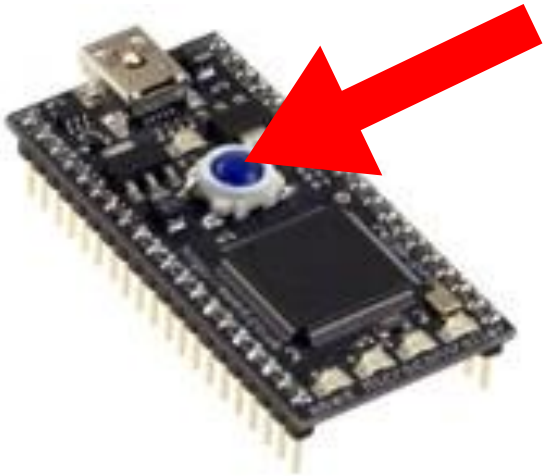
Getting Started

- Create or open a project in the Program Workspace
- Develop code in the text editor
- Save and compile
- Compiler outputs
  - Errors and warnings
  - -or-
  - A downloadable binary
- Save to the USB flash disk



Getting Started

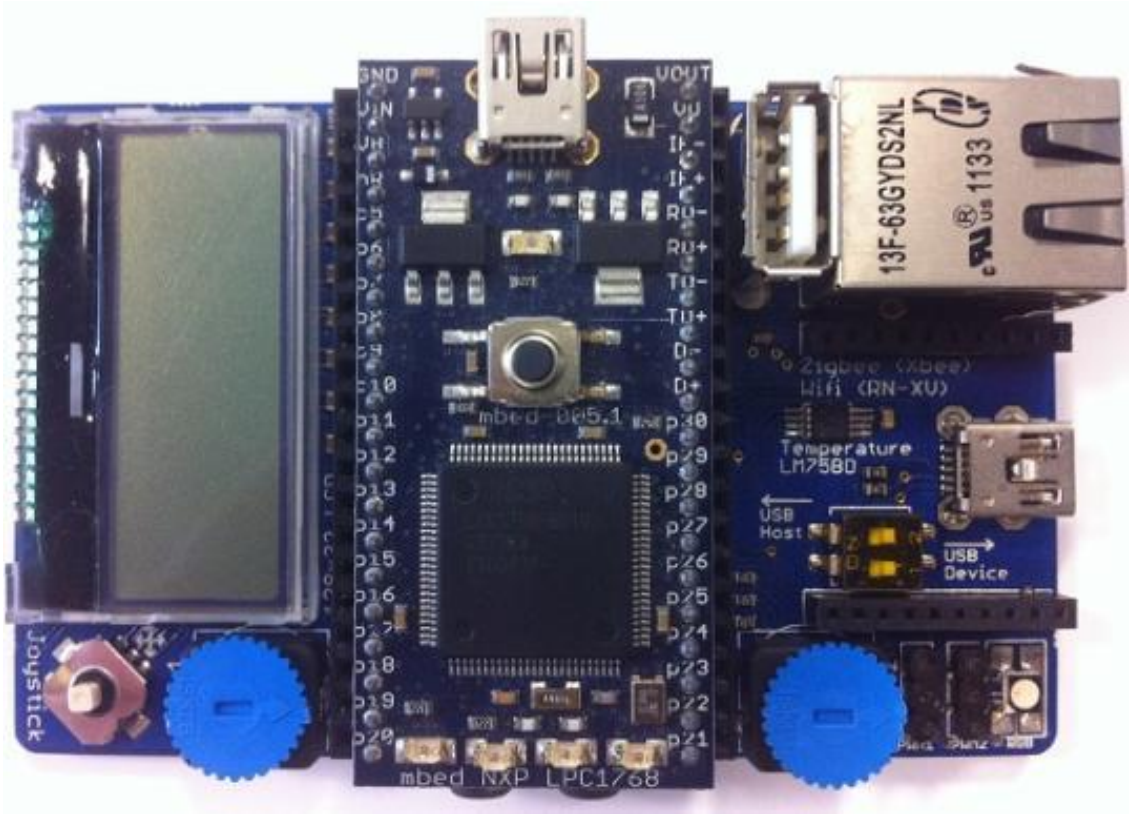
- Once the file has saved to the flash disk, it needs to be programmed into the microcontroller
- Press the button on the mbed module
- Your code will start running!



mbed
Hello World

Lab 2
Rapid Prototyping:
Other IO

mbed Application board



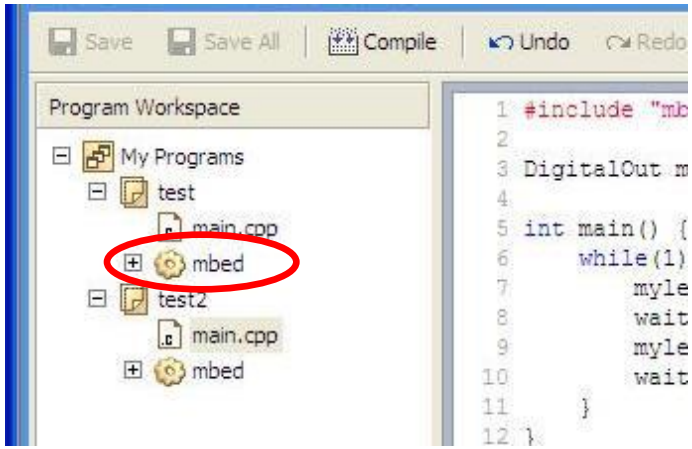
http://mbed.org/cookbook/mbed-application-board

DigitalOut and Analog Input

- In the hello world session, we simply compiled the default program – blinky, but we didnt take too much notice of the code
- It was simple, it set up a digital output (DigitalOut) called “myled” and run a loop forever turning it on and off.
- Lets see if we can begin to influence this.

What IO is there?

- Take another look at your compiler window. In your default project there the mbed library with a “+” box. Try expanding this, and exploring the libraries.
- Note that these are libraries that relate to the microcontroller on chip hardware.

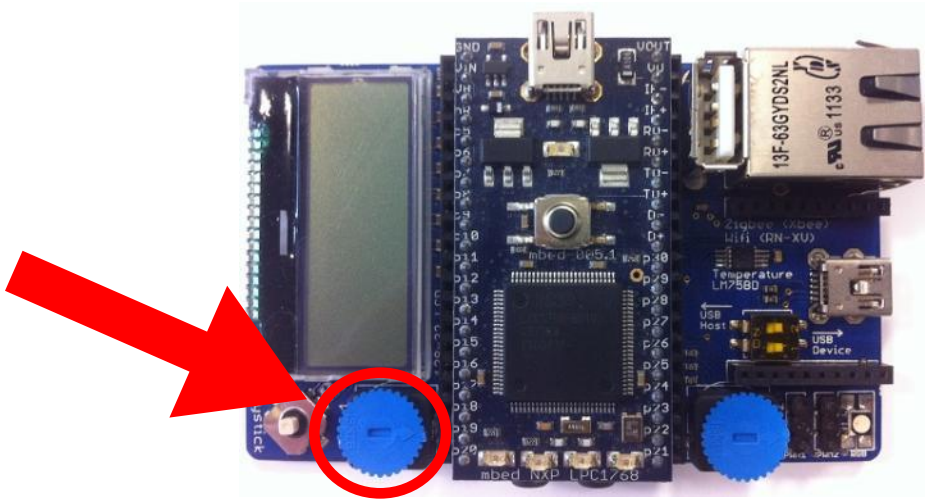


- We’ll be using the AnalogIn object, so take time to have a look at it’s API

DigitalOut and Analog Input

- The AnalogIn object returns a normalised float between 0.0 (0.0v) and 1.0 (3.3v)

- A potentiometer “pot1” is connected to pin 19 of the mbed NXP LPC1768



Challenge: DigitalOut and Analog Input

- Write a program to give the LED in the first blinky program a delay of 0.1 - 1.0 seconds.

main.cpp x

```
1  #include "mbed.h"
2
3  DigitalOut myled(LED1);
4  AnalogIn pot1(p19);
5
6  int main ()
7  {
8      while(1) {
9          myled = !myled;                    // toggle
10         wait (0.1 + (0.9 * pot1.read())); // 0.1s - 0.9s
11     }
12 }
13
```

- Write a program that turns LED1 on at 0.66v, LED2 on at 1.32v, LED3 on at 1.98v and LED4 at 2.64v

mbed
Hello World

Lab 3
Rapid Prototyping:
Interfacing a sensor

Example : Interfacing with sensors

- A good deal of microcontroller applications require some form of sensors to detect events or conditions in the immediate environment.
- This experiment show how to implement a simple temperature sensor.
- The sensor in question is the LM75B which has a digital interface using the I2C bus.

Connecting up the sensor

- The LM75B has just four pins, Vcc, Gnd for the power, and SCL, SDA for the I2C interface.

- As before, mbed keeps I2C simple, and online code reuse helps out!
  - http://mbed.org/handbook/I2C
  - http://mbed.org/cookbook/LM75B-Temperature-Sensor

Challenge : Interfacing with sensors

- Using the mbed application board cookbook page as a resource, write a program that turns LED1 on at 26^C, LED2 at 27^C, LED3 and 28^C and LED4 at 29^C.

- Note : To use “printf” in the hello world example, you will need to install the mbed Serial port driver, and a terminal application.

  – See : http://mbed.org/handbook/SerialPC for details to set this up

mbed
Hello World

Lab 4
Rapid Prototyping:
Output device, LCD

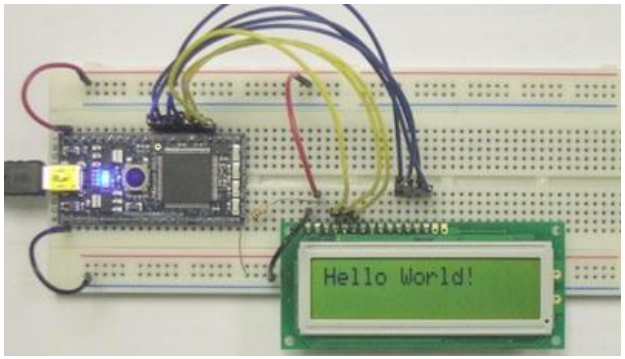
Example : Output device, LCD

- It is not uncommon for devices that are embedded to have some for of user interface, or display output.

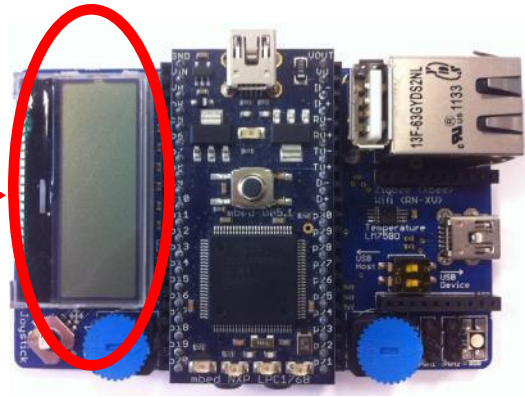
- This example shows how a LCD can be connected to mbed and be driven simply form software.

Connecting up the LCD

- Text LCD modules have almost standardised, although they still have their quirks.
  - Six wires and a resistor for contrast
  - As before, mbed keeps it simple
  - Standard C/C++ interface via printf



- The mbed application board has a graphics LCD
  - SPI interface with reset and chip select
  - Same API for character printing
  - Application code is the same!
  - http://mbed.org/cookbook/mbed-application-board



Challenge: Digital Thermometer

- Using the cookbook TextLCD page and the temperature sensor page, make a thermometer that displays the current temperature.

- If you have time, you could also add Min/Max to the display too

mbed
Hello World

Lab 5
Rapid Prototyping:
Data Logging

Example : Data Logging

- Applications often include data logging capabilities, and access to the data often involves bespoke software and interface cables.

- This example shows how standard methods and interfaces can be used to display, save and retrieve data from an application

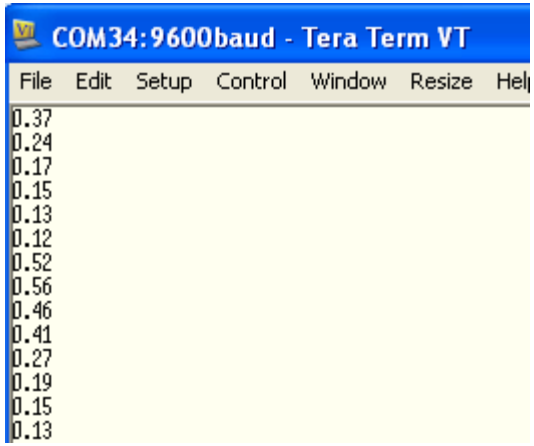
- For the purposes of the experiment, we will be displaying and logging the value read from the potentiometer.

Example : See the data

- The USB connection to mbed can also be used to provide a serial port
- Windows requires a driver, linux and Mac “just work”
- http://mbed.org/handbook/SerialPC
- Standard C functions, printf and scanf
- This example displays 100 samples to a terminal application
- http://mbed.org/users/chris/code/app-board-USB-printf/

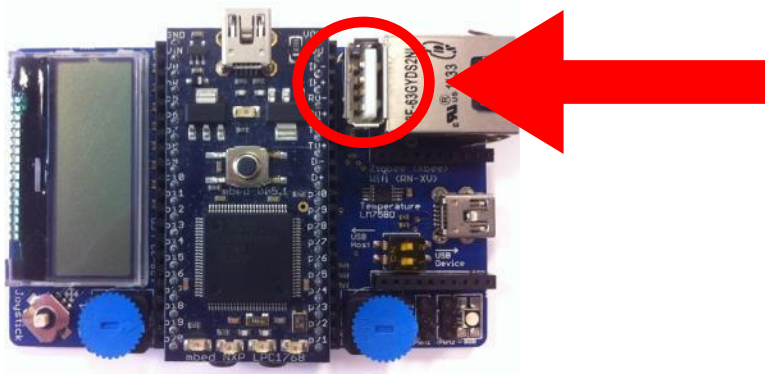
main.cpp x

```
1  #include "mbed.h"
2
3  AnalogIn pot1(p19);
4  DigitalOut myled(LED1);
5
6  int main()
7  {
8
9      for (int i=0; i < 100 ; i++) {
10         printf("%.2f\n",pot1.read());
11         wait(0.05);
12         myled = !myled;
13     }
14     myled = 1;
15
16 }
17
```



Storing lots of data

- Perhaps a final system might want to store lots of data
  - USB sticks ideal, ubiquitous and recognisable by everyone
  - Minimal hardware : USB socket, 4 wires (Gnd, +5v, D+, D-)
  - The mbed application board already provides this



Storing lots of data

- mbed keeps it simple :
  - Using the MSCFileSystem library
  - Make and object called “fs”

main.cpp x

```
1  #include "mbed.h"
2  #include "MSCFileSystem.h"
3
4  AnalogIn pot1(p19);
5  DigitalOut myled(LED1);
6  MSCFileSystem fs("fs");
7
8  int main()
9  {
10
11     FILE *fp = fopen("/fs/data.csv","w");
12
13     for (int i=0; i < 100 ; i++) {
14         fprintf(fp,"%.2f\n",pot1.read());
15         wait(0.05);
16         myled = !myled;
17     }
18
19     fclose(fp);
20     myled = 1;
```

  - http://mbed.org/users/chris/code/app-board-MSCFileSystem/

Challenge: Data Logging

- Use all you have learnt to build a digital thermometer that also data logs to a USB flash disk.

- Use a .csv file so that the file can be opened in Microsoft Excel, and a graph drawn.

mbed
Hello World

Lab 6
Rapid Prototyping:
Offline debug with CMSIS-DAP

Debugging with CMSIS-DAP

- The online tools are designed for rapid prototyping
- Sometimes product development requires full debug
  - Source code breakpoints
  - Watch points

- mbed enabled microcontrollers support CMSIS-DAP
  - Access to the DAP is provided over a driverless USB HID connection
  - All leading tools support this new “debug probe” interface
  - You can even write your own debugger (e.g. Python) to drive the DAP

- The mbed online tools enable you to export your project to an offline tool (MDK, IAR, Crossworks, Code Red, etc)

Installing Keil MDK

- This lab assumes Keil MDK (currently Windows only)
- To download and evaluation copy of MDK, visit

https://www.keil.com/demo/eval/arm.htm

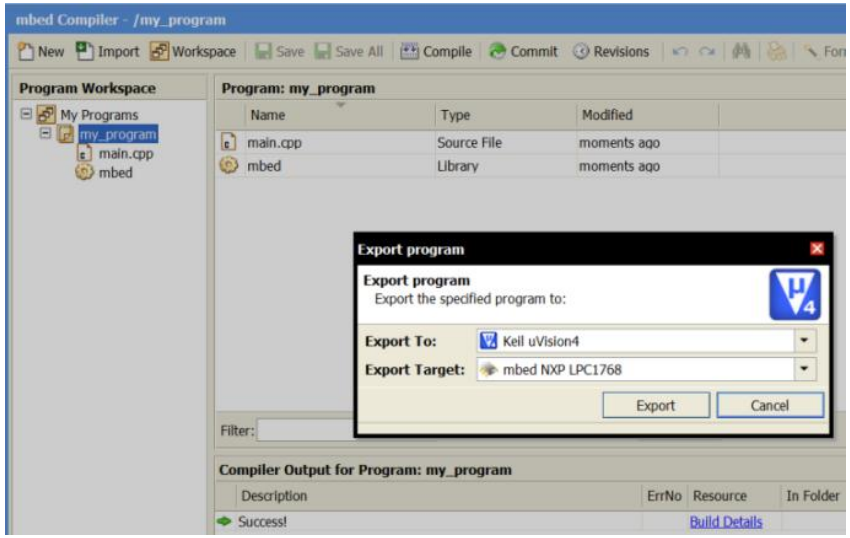
- Follow the install instructions

Upgrading your mbed NXP LPC1768

- By default, the mbed NXP LPC1768 is not CMSIS-DAP enabled
- Visit http://mbed.org/handbook/Firmware-LPC1768-LPC11U24
- Save the upgrade file (version 141212 or later) to your mbed
- Power cycle your mbed, and you now have CMSIS-DAP support

Exporting your mbed project

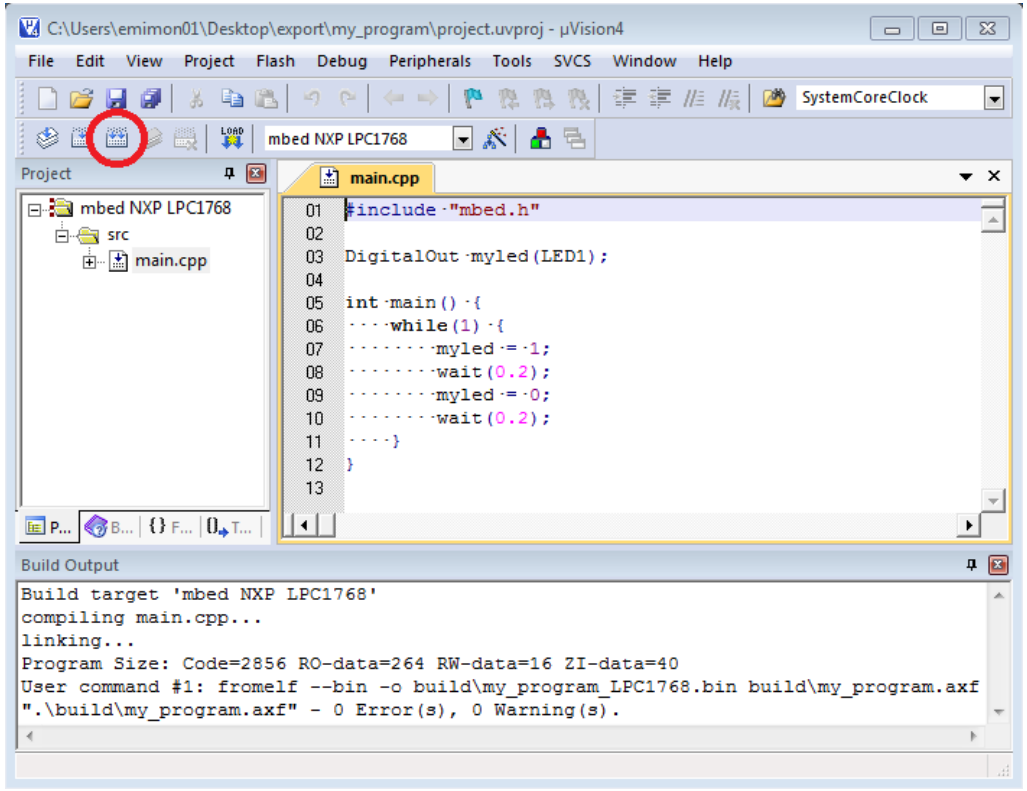
- For the purpose of this lab, we are using Keil MDK
- In the online compiler, right click your program, select “export”
- Unzip the downloaded project and click the .uvproj file to launch Keil MDK



- For more information see
  - http://mbed.org/handbook/CMSIS-DAP-MDK

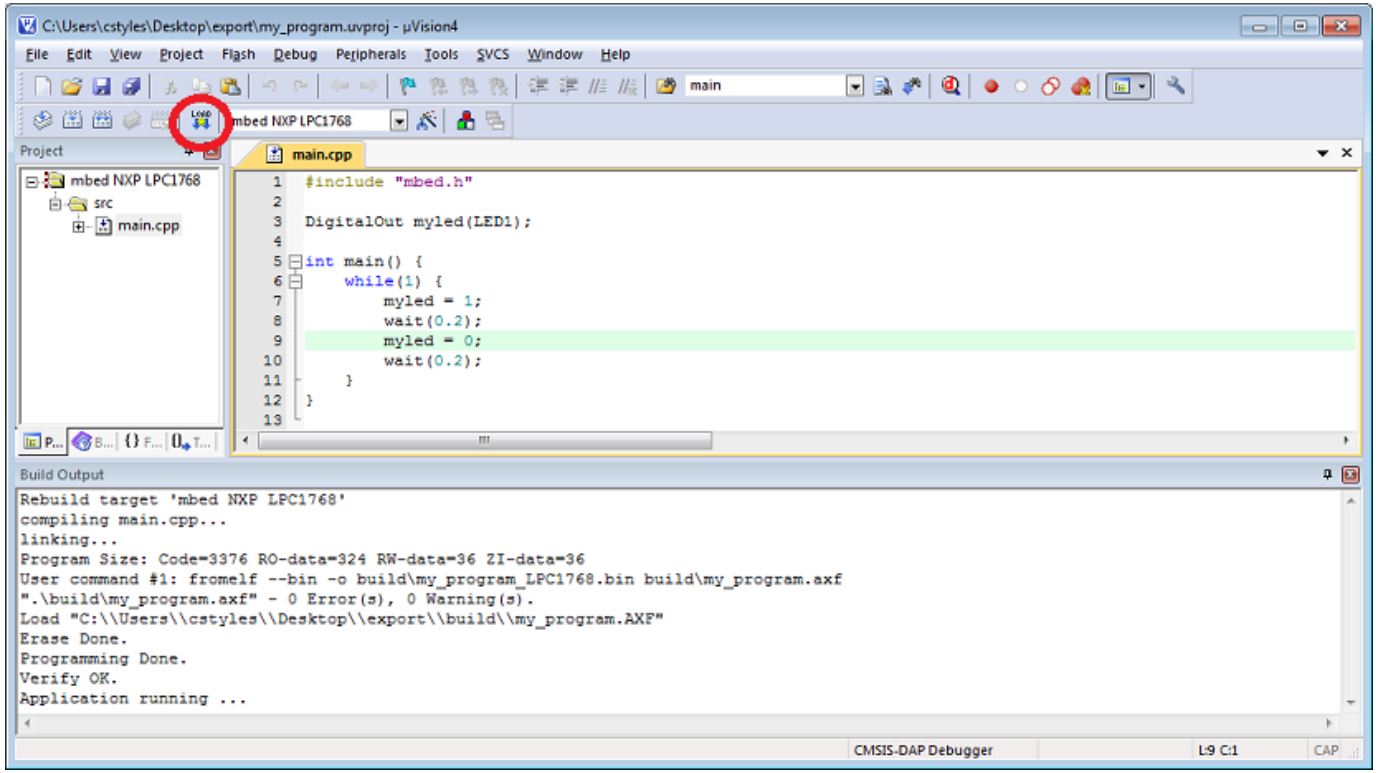
Compile your program

- Click the compile button in the MDK IDE to build your project
- This example shows the default “blinky” program



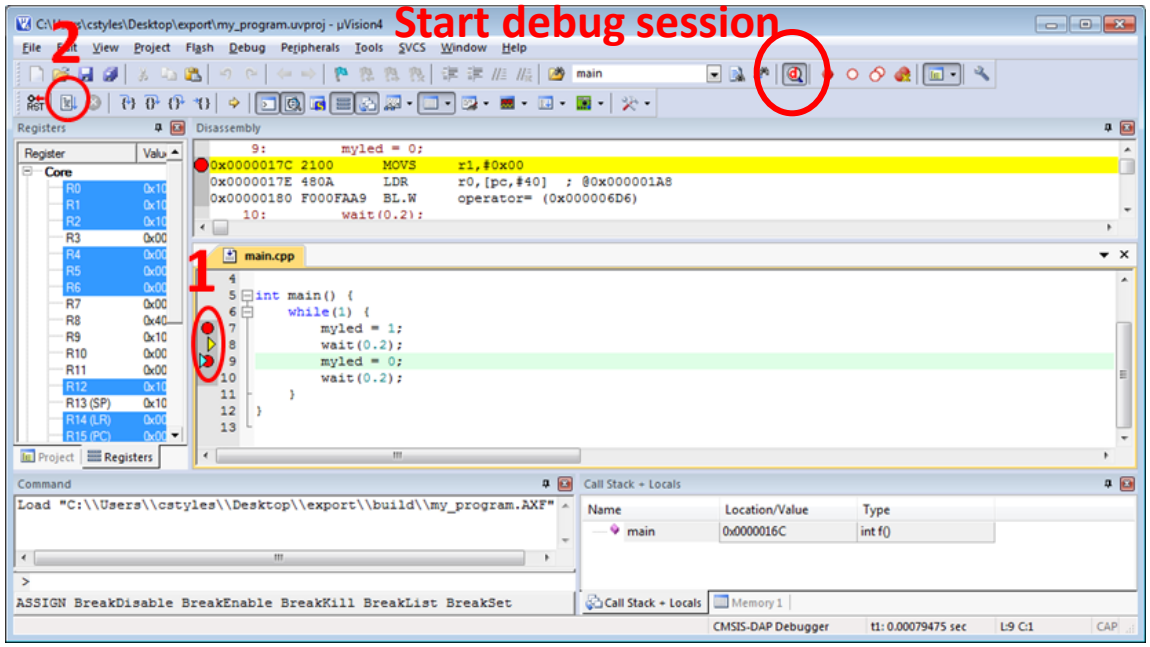
Download your program

- Instead of drag and dropping your program, the MDK IDE can load it to the LPC1768 over the CMSIS-DAP connection



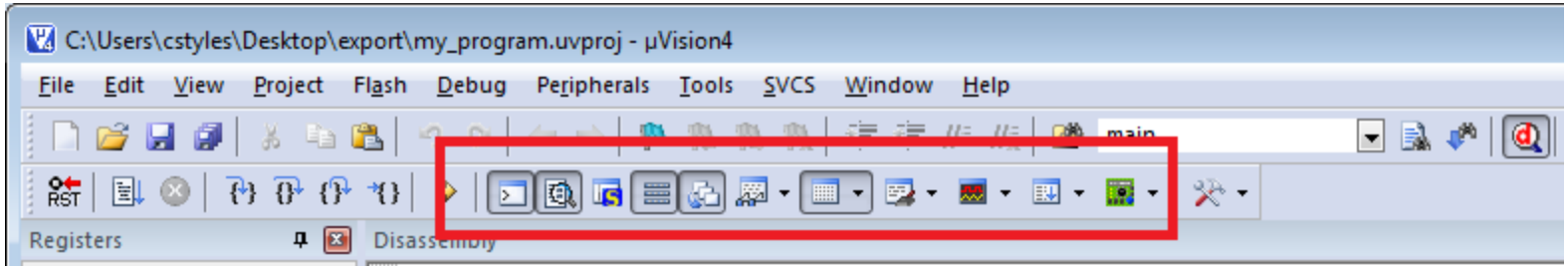
Debugging your program

- Start a debug session
- (1) Add break points by clicking on the line of code you wish to stop at. There are a maximum of 2 breakpoints
- (2) and use the "run" button to execute the program until the break point it hit.



Debug features

- Use the tool bar to add other debug feature to the IDE
  - Registers : The values in the CPU registers at the breakpoint
  - memory : View the current contents of RAM
  - call stack : The sequence of function call that lead to the breakpoint
  - Symbol : View (and modify) current variables



mbed
Hello World

Summary

Summary

- There is huge opportunity for microcontroller applications
  - A major barrier to adoption is simple experimentation
- mbed helps with getting started and rapid prototyping
  - Fast turnaround of experiments and prototyping new ideas
  - Try out new technology and new ideas
- Makes the technology very accessible
  - Demo showed a start to finish prototyping example
  - From getting a user started to enabling an application experiment
- Use at as a tool when you need to experiment!

Summary

- A solution focused on prototyping has a broad appeal
- Engineers new to embedded applications
  - Enables experimentation and testing product ideas for the first time
  - Create designs where electronics and MCUs are not the focus
- Experienced engineers
  - Provides a way to be more productive in the proof-of-concept stages
  - Introduce 32 bit microcontroller technology to existing designs
- Marketing, distributors and application engineers
  - Provides a consistent platform for demonstration, evaluation, support
  - Make promotion of MCUs more effective and efficient

Q&A

Slides available at …

http://mbed.org/cookbook/Workshop

Under “ARM University Program”