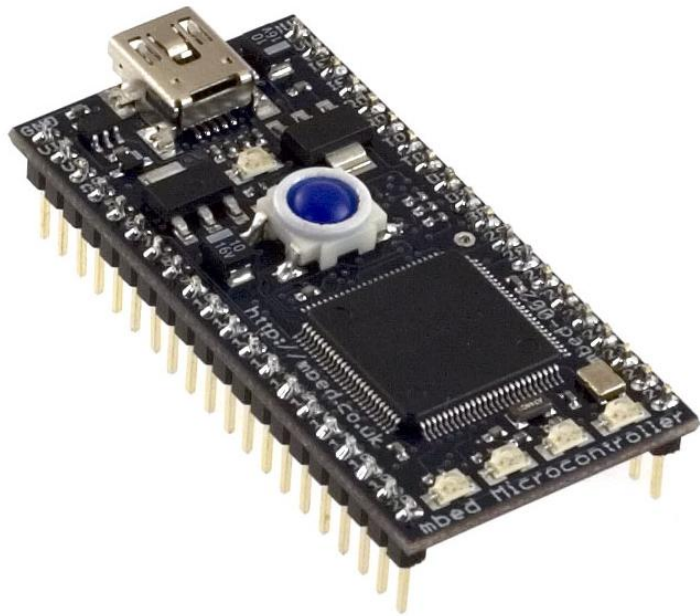


mbed + Sprint Hello World!

Agenda

- mbed
 - 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: SMS Alerts
 - Lab 6 : Mobile data with websockets

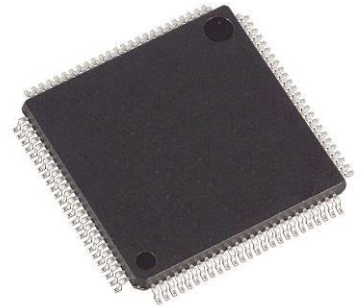


mbed + Sprint Hello World!

Introduction to mbed

What's happening in Microcontrollers?

- Microcontrollers are getting cheap
 - 32-bit ARM Cortex-M3 Microcontrollers @ \$1
 - 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 1980 computers were used and applied by computer scientists
- now they are:
 - Applied across all industries
 - Widely used in the home
 - Used by almost anybody
- Currently microcontroller technology is mainly applied by the embedded professional
- 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



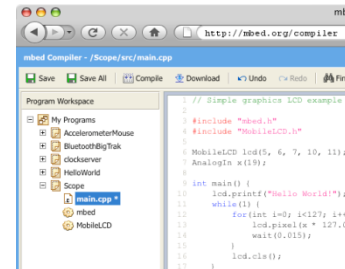
2D/3D Design



Web Frameworks

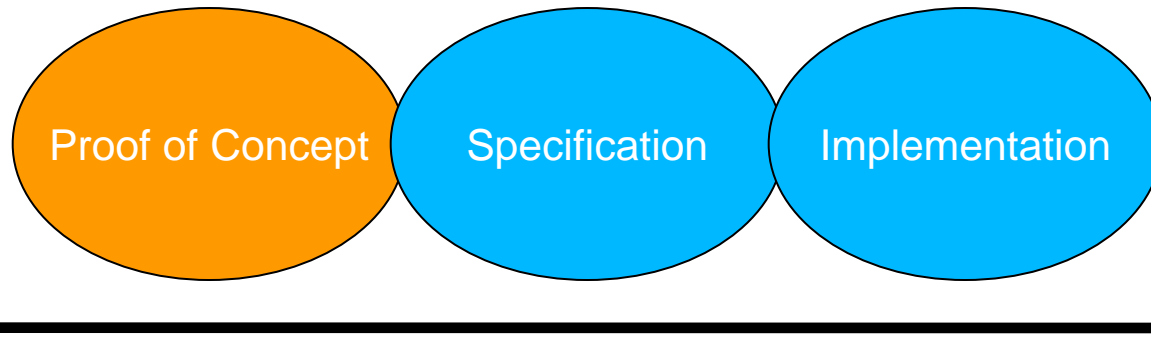
mbed.org - Rapid Prototyping for MCUs

- Fastest way to get started with ARM microcontrollers
 - Plug ‘n’ Play Hardware, Online Compiler
 - Get setup and run “Hello World!” in 60 seconds
 - 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
 - Creates new applications for MCU technology
- 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

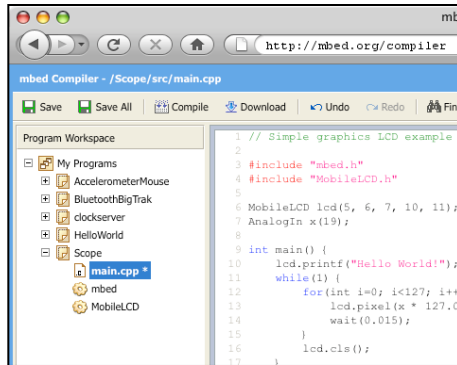
- Focus on tools supporting the earliest stage of design
 - Point of entry and Getting Started
 - Experimentation and Rapid Prototyping



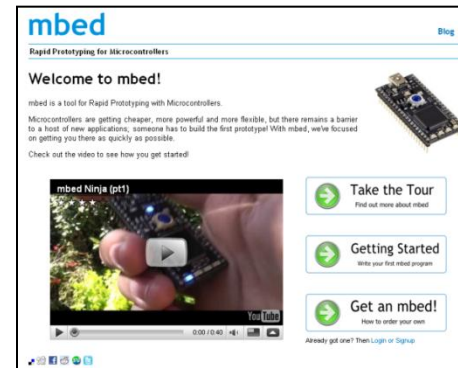
- Apply technology and trade-offs that support this goal
- What mbed is not trying to do:
 - Replace existing professional tools
 - Replace development or evaluation boards

mbed Rapid Prototyping Platform

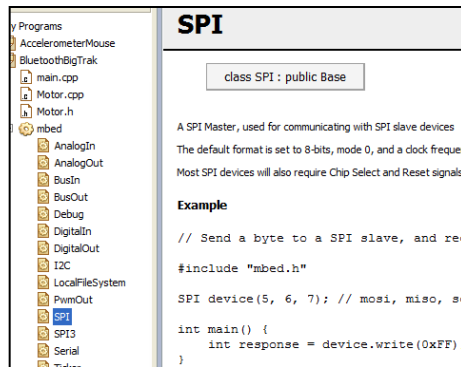
- Complete Hardware, Software and Web 2.0 Solution



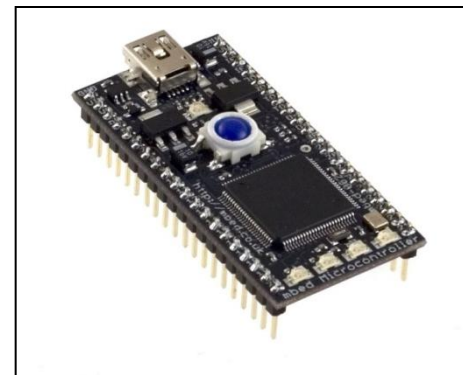
Dedicated Developer Website



Lightweight Online Compiler



High-level Peripheral APIs



Prototyping Form-Factor

mbed Website

- Dedicated Developer Web Platform
 - Custom Web 2.0 tools and environment focused on developers
 - Simple route to get started, comprehensive resources and support

mbed
Forum

mbed Forum

	Topic
👤	Beta Testers - The First 5 Minutes
👤	Ethernet interface
👤	mbed + xBee
👤	USB Serial problem
👤	License and future support
👤	help with ethernet/rj45

mbed Blog | For...

Rapid Prototyping for Microcontrollers

Welcome to mbed!

mbed is a tool for Rapid Prototyping with Microcontrollers.

Microcontrollers are getting cheaper, more powerful and more flexible, but there remains a barrier to a host of new applications; someone has to build the first prototype! With mbed, we've focused on getting you there as quickly as possible.

Check out the video to see how you get started!

Take the Tour
Find out more about mbed

Getting Started
Write your first mbed program

Get an mbed!
How to order your own

Already got one? Then [Login](#) or [Signup](#)

mbed

Projects » [cookbook](#)

Cookbook

Welcome to the Cookbook, a **wiki** and **code repository** for publishing your own...

For information in creating new projects, editing pages and uploading files, visit...

Working Libraries and Examples

- [TextLCD](#) - A 16x2 Text LCD
- [MobileLCD](#) - A 130x130 Nokia Mobile Screen
- [RFID](#) - An ID-12 RFID tag reader
- [Servo](#) - Controlling a Servo
- [GPS](#) - Read location using a GPS module
- [MIDI](#) - For communicating using MIDI (musical instrument digital interface)
- [NXT](#) - Making lego mindstorm (NXT) sensors work with the mbed
- [USRF](#) - SRF05 Ultrasonic range finder
- [Beeper](#) - Driving an Piezo sounder

```
// Example code for sending range data from the SRF05 Ultrasonic Range Finder
#include "mbed.h"
int length;
Serial port(Serial, 9600); // tx, rx
int pin_for_sonar = 18; // Define the I2C Address

int main() {
    char mbed();
    while(1) {
        // Send data to serial monitor
        length = 0;
        // Start ranging process on the
        SRF05.range(pin_for_sonar, 1); // Send command string
        while(!pin); // Wait until pin is triggered

        // See pointer to instruction 2 (start mbed)
        SRF05.range();
        SRF05.range(pin_for_sonar, 1);
        SRF05.range(pin_for_sonar, 1); // read the hex-type who range
        while(!pin); // Wait until pin is triggered

        // print the ranging data to the screen
        Serial.printf("SRF05: %d\n", length);
        printf("Range = %d", length);
        while(1);
    }
}
```

API

DC	An I2C Master, used for communicating with I2C slave devices
PinName	
I2C	Create an I2C Master interface, connected to the specified pins
Frequency	Set the Frequency of the I2C interface
read	Read from an I2C slave
write	Write to an I2C slave

<http://mbed.org>

XBee Explorer

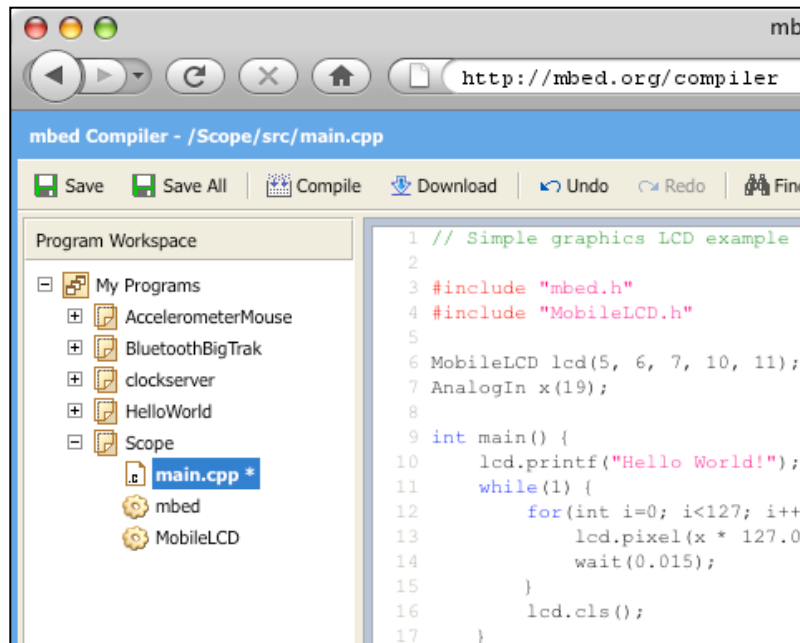
For really wanted to use with the mbed. These are the standard ODMR modules with an antenna and one with the XBee Explorer USB from Sparkfun. http://www.sparkfun.com/commerce/product_info.php?products_id=56

According to the datasheet (and Sitemap) of the XBee modules, the following should be used as the pin layout. Main thing one is looking for is that serial rx - tx, and a digital gnd for the rx.

1	GND 3.3V
2	NEXT analog digital rx (e.g. pin0)
3	TX analog digital tx (e.g. pin1)
4	Not connected analog digital rx (e.g. pin2)
5	SRF05 analog digital rx (e.g. pin3)
6	Not connected analog digital rx (e.g. pin4)
7	GND GND
8	DC unconnected

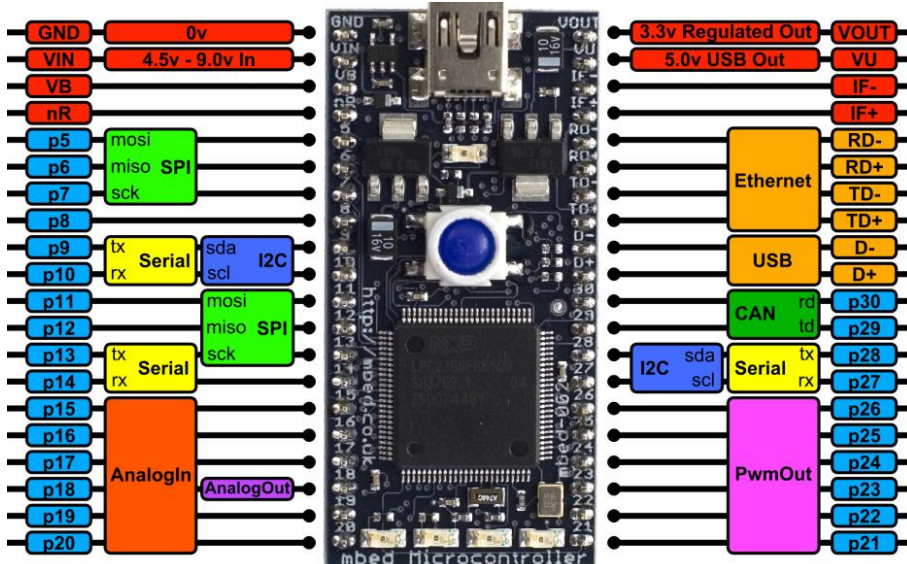
mbed Compiler

- Lightweight Online Compiler
 - Web 2.0 browser-based IDE with personal workspace “in the cloud”
 - Nothing to install or configure, login from anywhere
 - Industry leading RVCT 4.1 back end. It is a real tool!



mbed Library

- High-level Peripheral APIs
 - Trading memory and CPU performance for ease of use
 - Abstract software interfaces for controlling microcontroller hardware
 - Intuitive peripheral access, encapsulation of implementation details
 - Treat hardware and software the same



The screenshot shows an IDE with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with files like main.cpp, Motor.cpp, Motor.h, and a folder named mbed. The code editor shows the following code:

```
class SPI : public Base

A SPI Master, used for communicating with SPI slave devices

The default format is set to 8-bits, mode 0, and a clock frequen

Most SPI devices will also require Chip Select and Reset signals.

Example

// Send a byte to a SPI slave, and rec

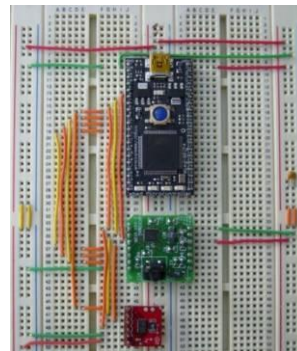
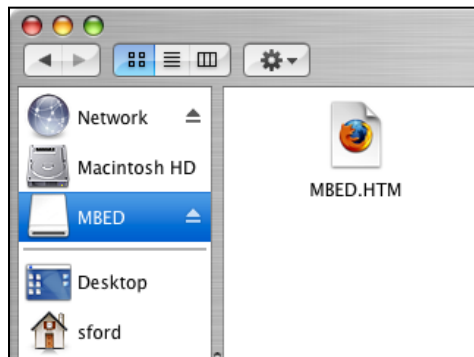
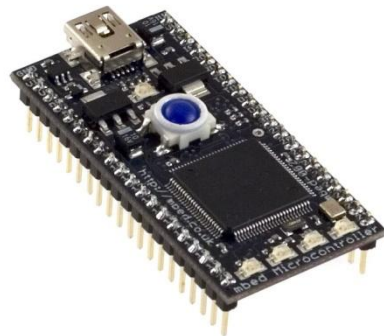
#include "mbed.h"

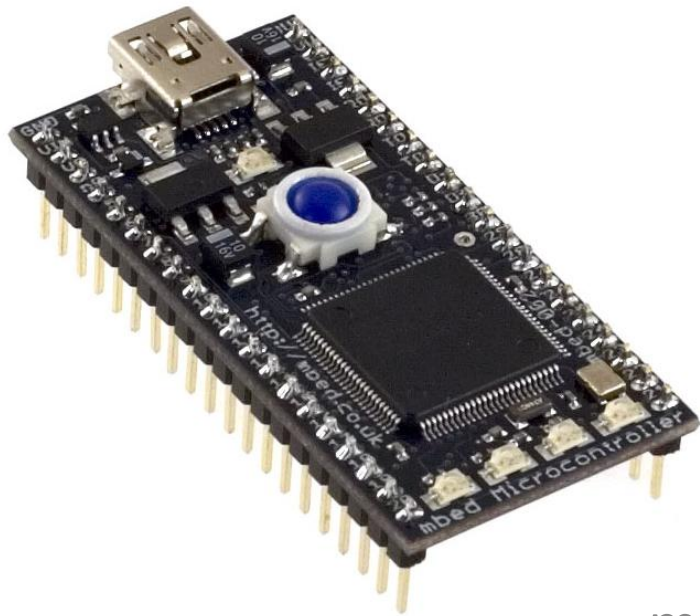
SPI device(5, 6, 7); // mosi, miso, sc

int main() {
    int response = device.write(0xFF);
}
```

mbed Microcontroller

- Cortex-M3 MCU in a Prototyping Form-Factor
 - 0.1” pitch DIP with “USB Disk” interface and support components
 - Nothing to install or configure, practical for breadboard and PCBs





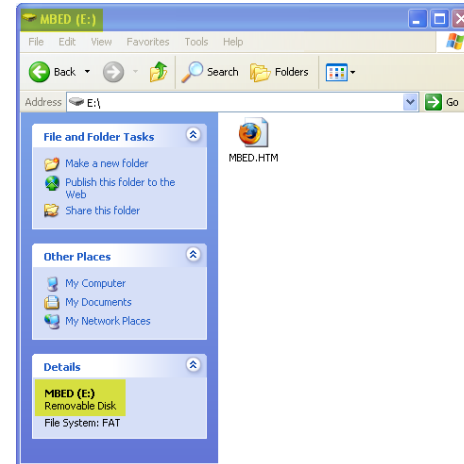
mbed + Sprint Hello World

Lab 1

mbed registration and hello world!

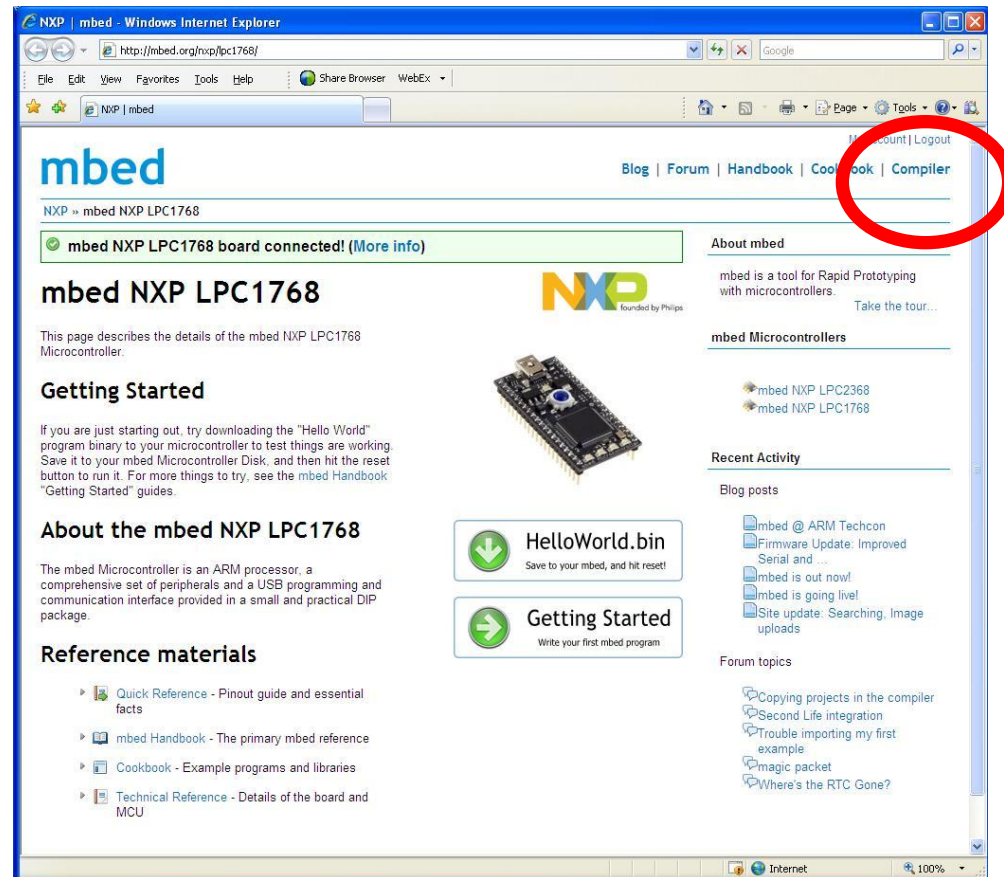
Registration

- mbed microcontroller enumerates 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
- The mbed microcontroller contains your license to the compiler



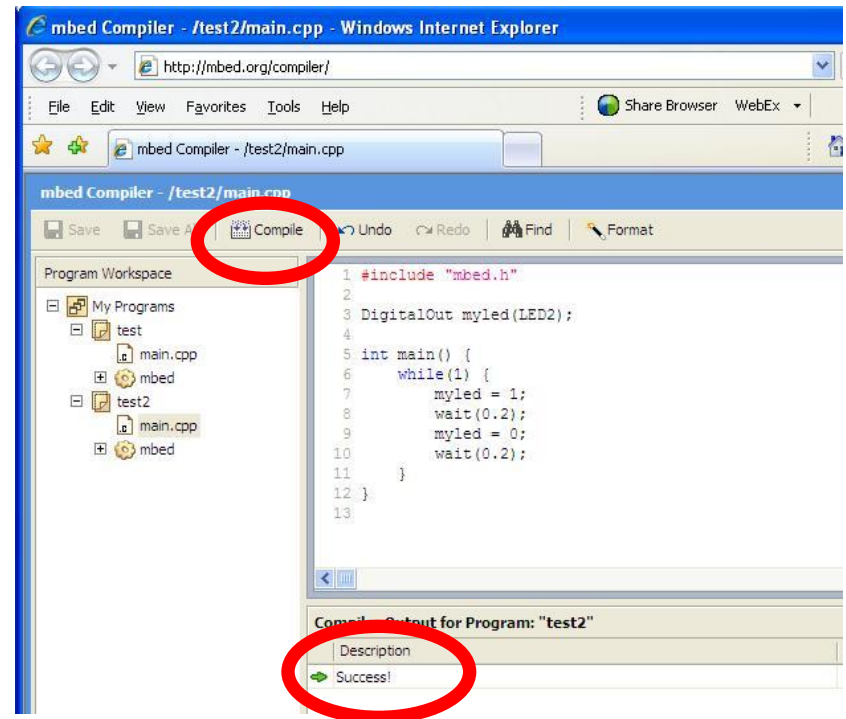
Getting Started

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



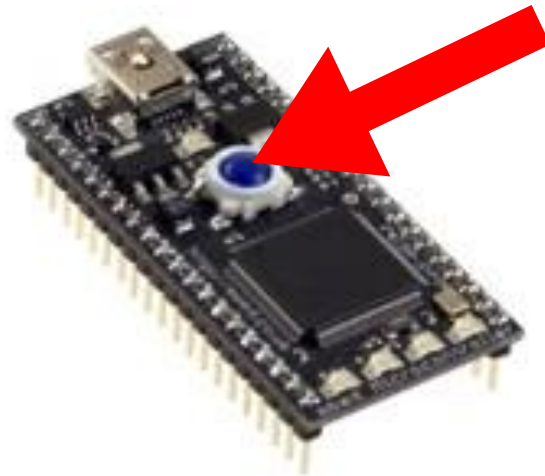
Getting Started

- 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

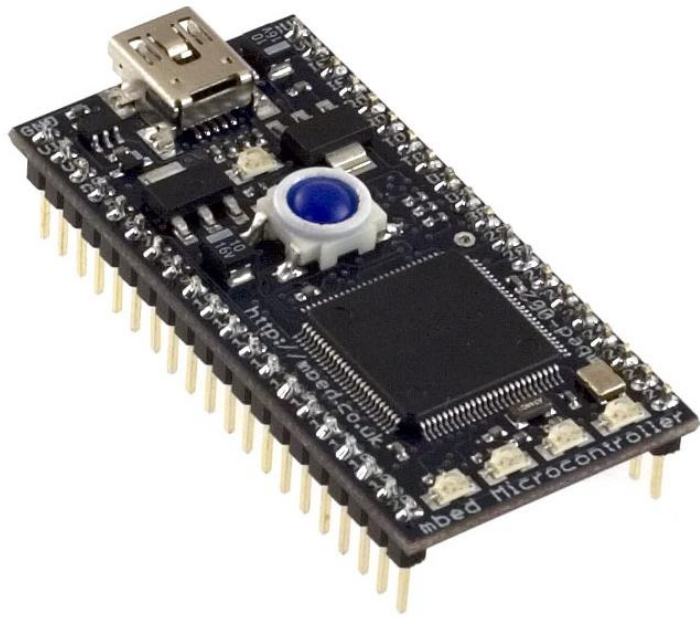
- Once the file has saved to the flash disk, it needs to be programmed into the microcontroller
- Press the button on the mbed module
- Your code will start running!



Over to you...

<http://mbed.org/cookbook/SprintUSBModem>

This presentation and the rest of the notes are
published as a PDF



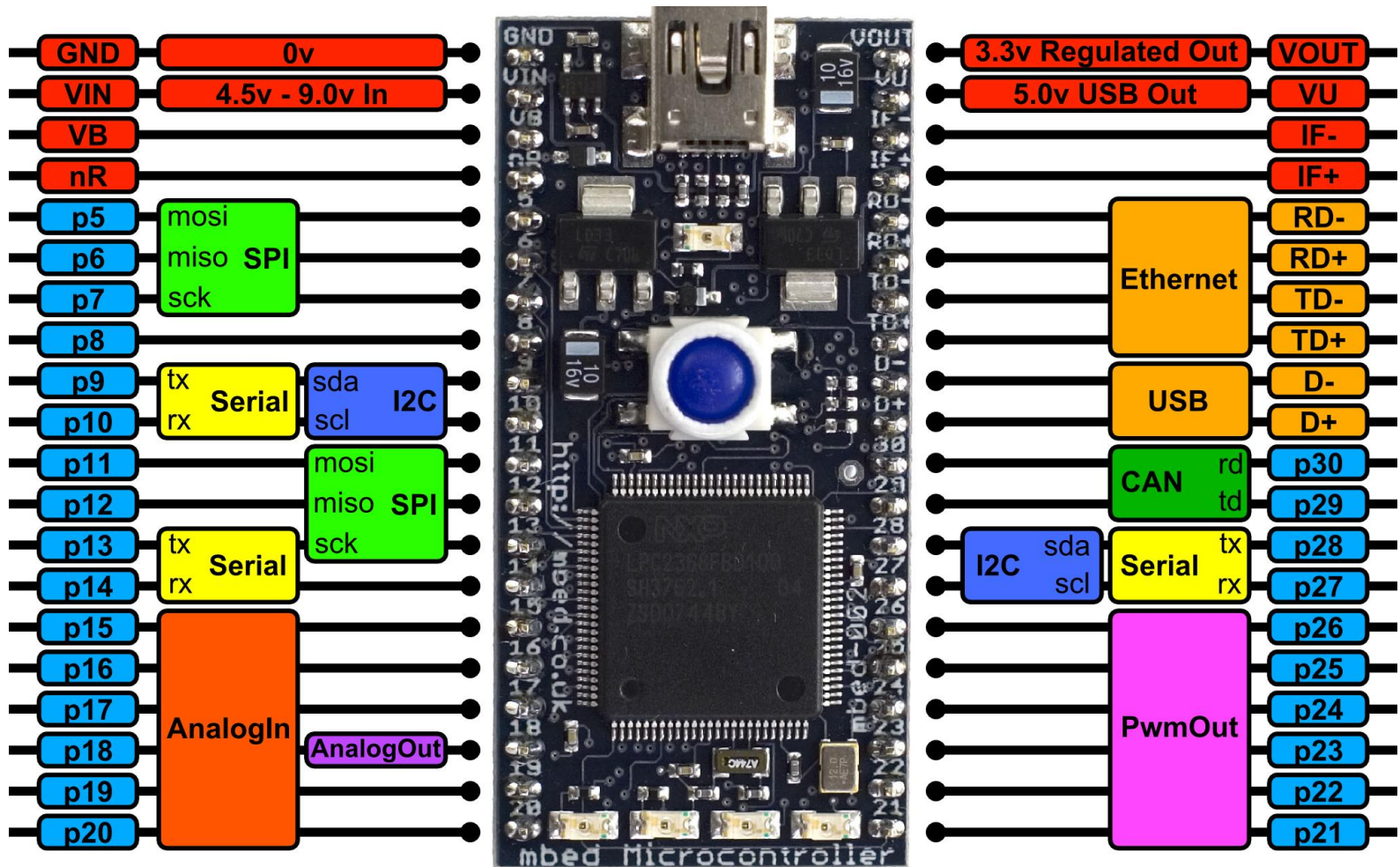
mbed + Sprint Hello World

Lab 2
Rapid Prototyping: Other IO

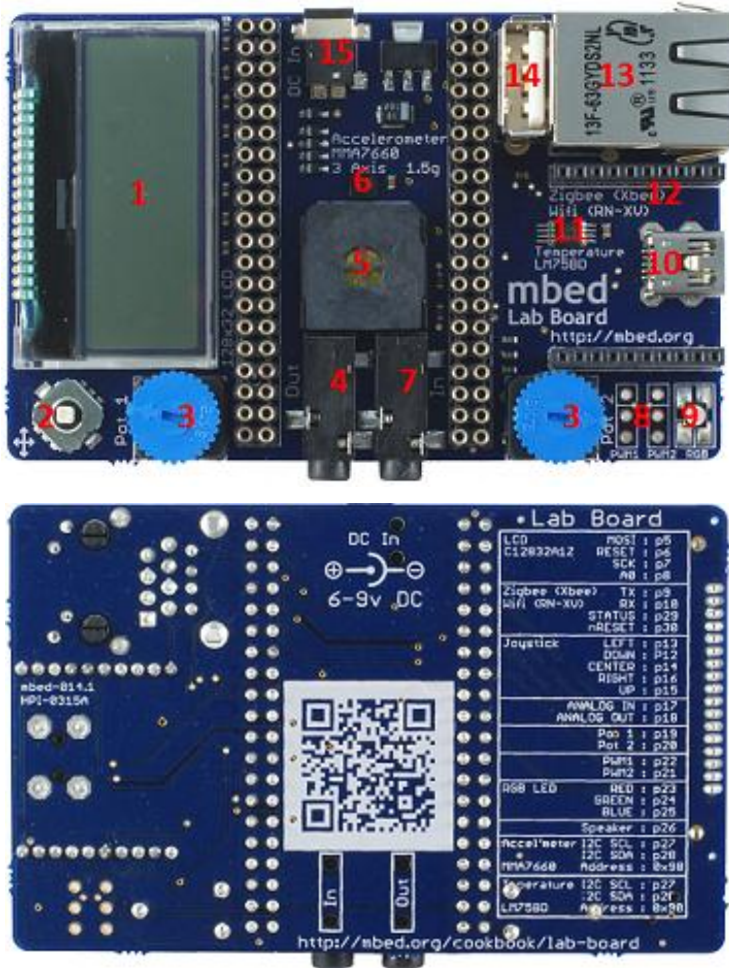
DigitalOut and Analog Input

- 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 using and Analog input

What IO is there?



mbed application board

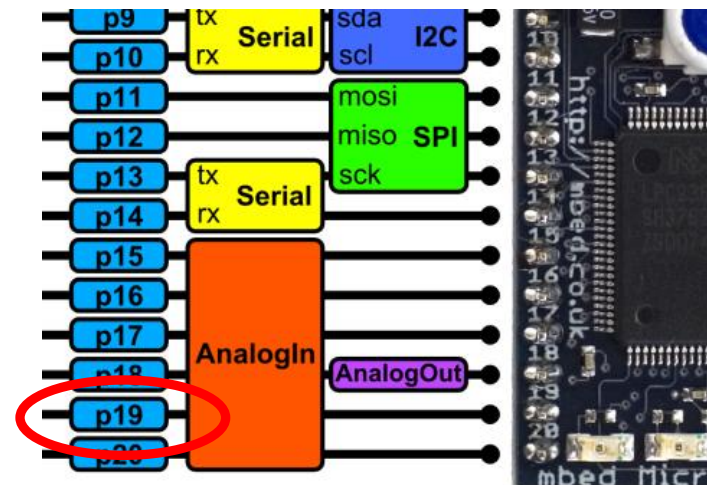
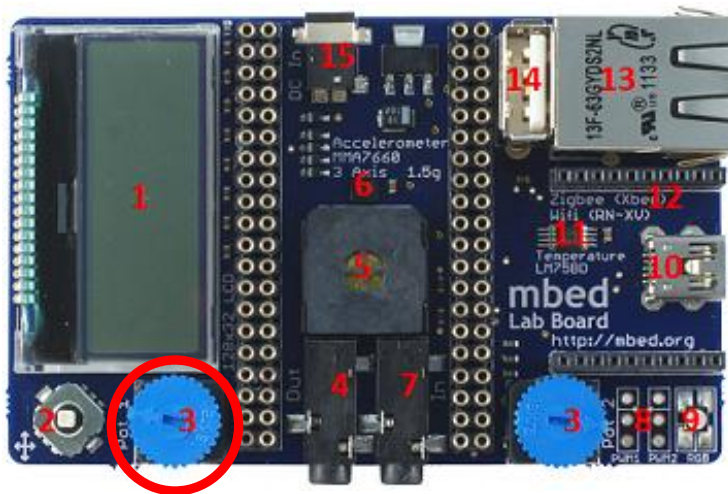


1. 128x32 Graphics LCD
2. 5 way joystick
3. 2 x Potentiometers
4. 3.5mm Audio jack (Analog Out)
5. Speaker, PWM Connected
6. 3 Axis +/- 1.5g Accelerometer
7. 3.5mm Audio jack (Analog In)
8. 2 x Servo motor headers
9. RGB LED, PWM connected
10. USB-B Connector
11. Temperature sensor
12. Socket for Xbee (Zigbee) or RN-XV (Wifi)
13. USB-A Connector
14. RJ45 Ethernet connector
15. 1.3mm DC Jack input

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

DigitalOut and Analog Input

- The AnalogIn object returns a normalised float between 0.0 (0.0v) and 1.0 (3.3v)
- Pot1 is wired between GND (0v) and Vout (3.3v), and is connected to pin “p19” – an AnalogIn

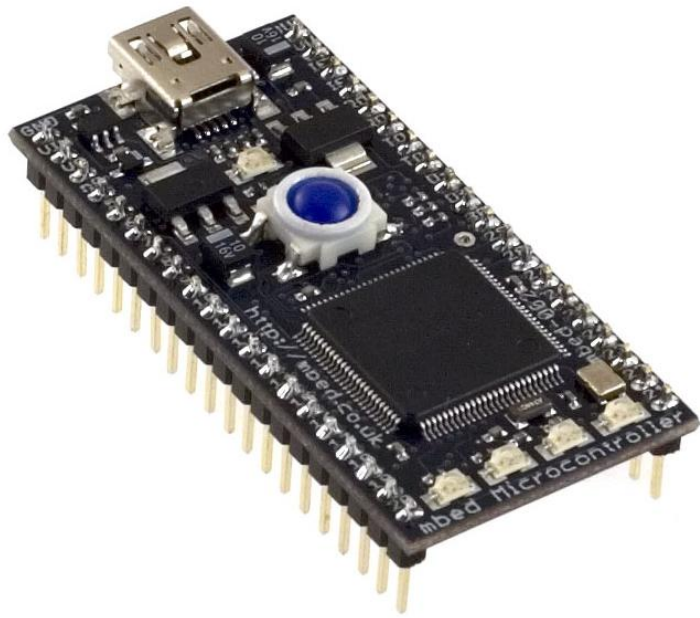


Challenge: DigitalOut and Analog Input

- Write a program to give the LED in the first blinky program a delay of 0.1-1.1 seconds.

```
main.cpp x
1 #include "mbed.h"
2
3 DigitalOut myled(LED1);
4 AnalogIn pot(p19);
5
6 int main() {
7     while(1) {
8         myled = !myled;
9         wait(0.1 + pot.read());
10    }
11 }
12
```

- 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
- Hint : Look at BusOut in the mbed Handbook!



mbed + Sprint Hello World

Lab 3
Rapid Prototyping: Interfacing a sensor

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.

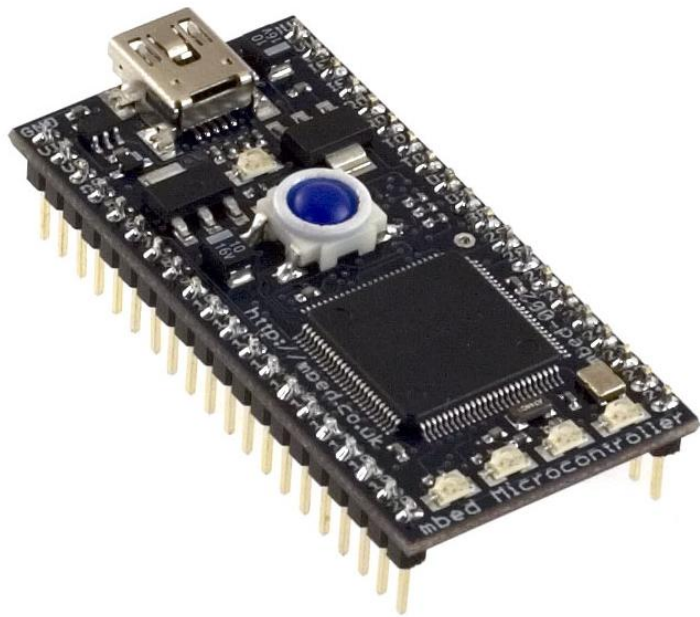
Conencting a Sensor

- mbed keeps I2C simple, and a library and example exists
 - I2C handbook page - <http://mbed.org/handbook/I2C>
 - Cookbook - <http://mbed.org/cookbook/LM75B-Temperature-Sensor>
 - Hello World - <http://mbed.org/users/chris/code/app-board-LM75B/>

```
main.cpp x
1 #include "mbed.h"
2 #include "LM75B.h"
3
4 LM75B temp(p28,p27,0x90);
5
6 int main() {
7     while(1) {
8         printf("Temperature : %f\n",temp.read());
9         wait(5);
10    }
11 }
```

Challenge : Interfacing with sensors

- Write a program that turns LED1 on at 26°C, LED2 at 27°C, LED3 and 28°C and LED4 at 29°C.
- As an extended challenge, add Min/Max recordings to the program



mbed + Sprint Hello World

Lab 4

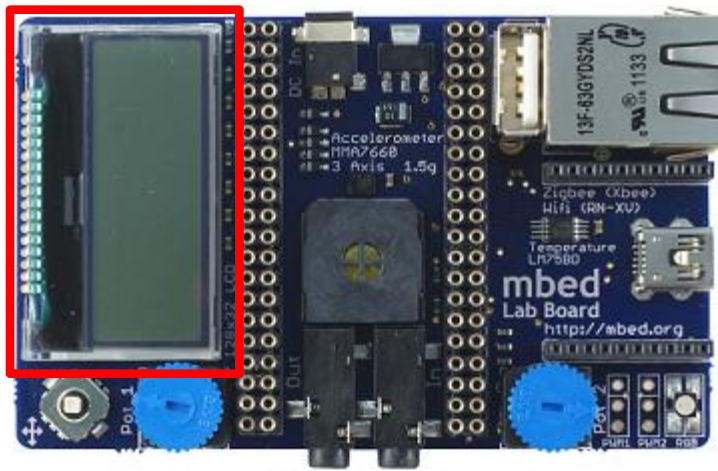
Rapid Prototyping: Output device, Text LCD

Output device, LCD

- It is not uncommon for devices that are embedded to have some form of user interface, or display output.
- This example shows an LCD connected to mbed and be driven simply from software.

Connecting up the TextLCD

- The LCD module has an SPI interface and a few digital outputs for reset, chips select and so on

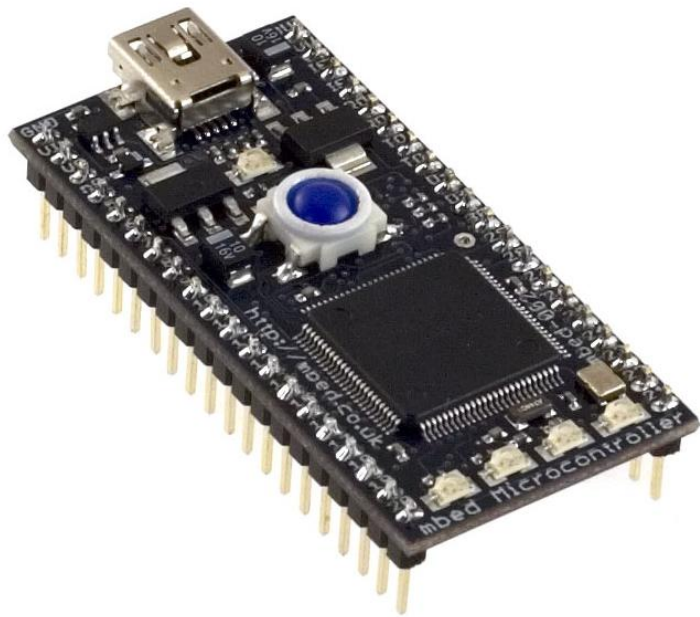


```
main.cpp x
1 #include "mbed.h"
2 #include "C12832_lcd.h"
3
4 C12832_LCD lcd;
5
6 int main()
7 {
8     lcd.cls();
9     lcd.locate(0,3);
10    lcd.printf("mbed application board!");
11 }
12
```

- mbed keeps it simple
 - Standard C/C++ interface via printf
 - Hello World : <http://mbed.org/users/chris/code/app-board-LCD>

Challenge: Digital Thermometer

- Make a digital thermometer that displays the current temperature.
- If you have time, you could also add Min/Max to the display too
- <http://mbed.org/users/chris/code/app-board-LM75B/>



mbed + Sprint Hello World

Lab 5
Rapid Prototyping: SMS alerts

SMS Alerts

- Some applications need to include an element of communication
- Taking the previous example of registering Min/Max temperatures, it might be desirable to communicate if limits are exceeded
- This example shows how the SprintUSBModem can be used to achieve mobile communication,

Hardware

- Simple hardware :
 - mbed NXP LPC1768 microcontroller
 - mbed application board
 - Sprint Mobile Broadband
 - DC adaptor



Example Program : SMS Send

- An example program of how to send SMS messages using the USB Modem can be found here

<http://mbed.org/users/donatien/code/SprintUSBModemSMSTest/>

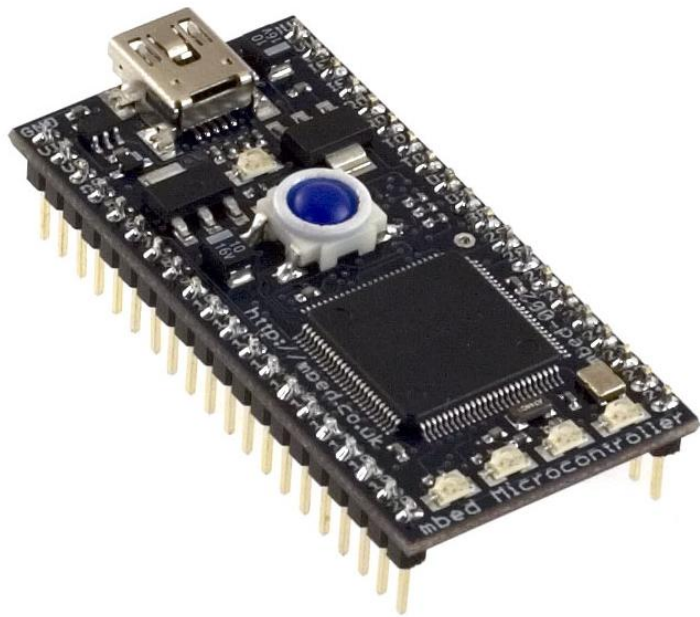
- Experiments :
 - Use the example program to send your self a message
 - Using “ssprintf” send yourself a over temperature alert message

Challenge: SMS Receive

- Once you have successfully sent SMS messages to your cell, try this :

Challenge

- Set the minimum and maximum alert points by SMS
 - Receive a reply message from your cell back to the mbed
 - Use “scanf” to retrieve the new minimum and maximum values for your thermometer
 - Trigger alert SMS messages when these new limits are exceeded
 - Include the temperature reading in the alert message



mbed + Sprint Hello World

Lab 6
Rapid Prototyping: Mobile data logging

Mobile data logging

- This example shows how the Sprint Mobile Broadband can be used to achieve remote data logging, where the data is sent live to online storage
- The driver is now providing a socket interface over which various protocol APIs and. For this example, we are using HTML5 web sockets
- Take 5 minutes to familiarise yourself with web sockets :
 - <http://mbed.org/cookbook/Websockets>
 - <http://mbed.org/cookbook/WebSocket-Mbed-Server>

WebSocket server channels

The mbed.org websocket uses “channels”, with “demo” as the default. To avoid conflict during a workshop, create your own channel by substituting “demo” with your own channel name

In mbed code :

```
WebSocket ws("ws://sockets.mbed.org:443/ws/demo/rw");
```

Becomes

```
WebSocket ws("ws://sockets.mbed.org:443/ws/<your_channel>/rw");
```

In browser URLs :

<http://sockets.mbed.org/demo/viewer>

becomes

http://sockets.mbed.org/<your_channel>/viewer

Hardware

- Simple hardware :
 - mbed NXP LPC1768 microcontroller
 - mbed application board
 - Consumer USB Modem
 - DC adaptor



HTML5 and Websockets



- New feature of HTML5 (RFC 6455) providing:
 - Full-duplex communication
 - Over a single TCP socket
 - Standard and secure connections (ws:// and wss://)
- Motivation:
 - Replace existing polling techniques (AJAX) used in modern websites
 - Provide a two-way communication without multiple HTTP connections
 - Enable new classes of application
- Other notable HTML5 features:
 - HTML5 Canvas Element – For dynamic, scriptable 2D rendering

Example Program : Web sockets

- An example program of how to send the current temperature as a web socket message to the mbed web socket server using the Sprint USB Modem can be found here

<http://mbed.org/users/chris/code/SprintUSBModemWebsocketTest-Temp/>

See the output here :

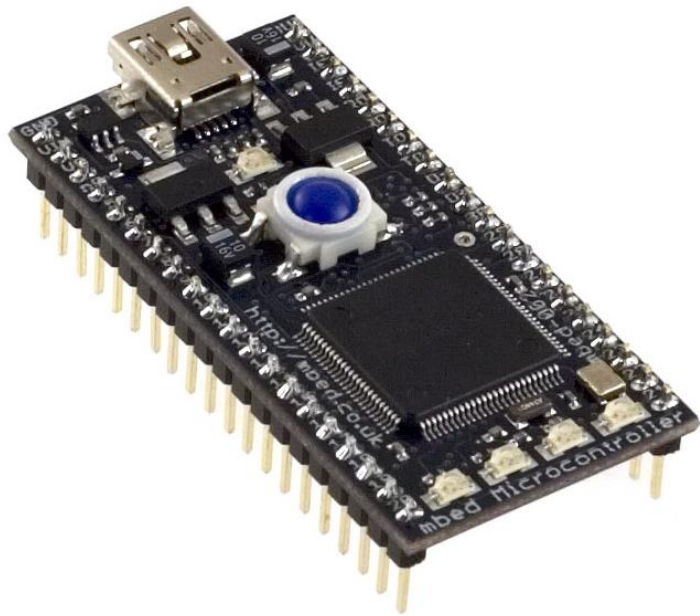
<http://sockets.mbed.org/demo/viewer>

Experiments :

- Trigger a web socket message using navigation switch

Challenge : Web sockets

- The previous example was write-only (“wo”) and sent data to the socket server
- Try making a connection that is read-only “ro”
 - To see how to receive web socket messages refer to :
<http://mbed.org/users/donatien/code/SprintUSBModemWebsocketTest/>
 - Display received messages on the LCD, for examples see :
<http://mbed.org/cookbook/mbed-application-board>
 - To send messages, use the “sender”
<http://sockets.mbed.org/demo/sender>
- If you get stuck... But try to do it yourself first !
<http://mbed.org/users/chris/code/SprintUSBModemWebsocketTest-LCD-RO/>



mbed + Sprint 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 it 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