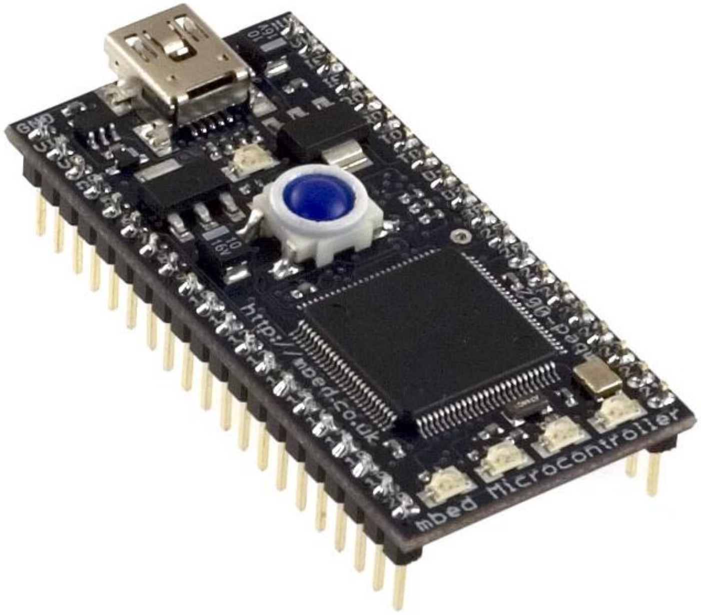


mbed
Hello World

Agenda

- Introduction to ARM

- 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: Rapid prototyping, Build a datalogger

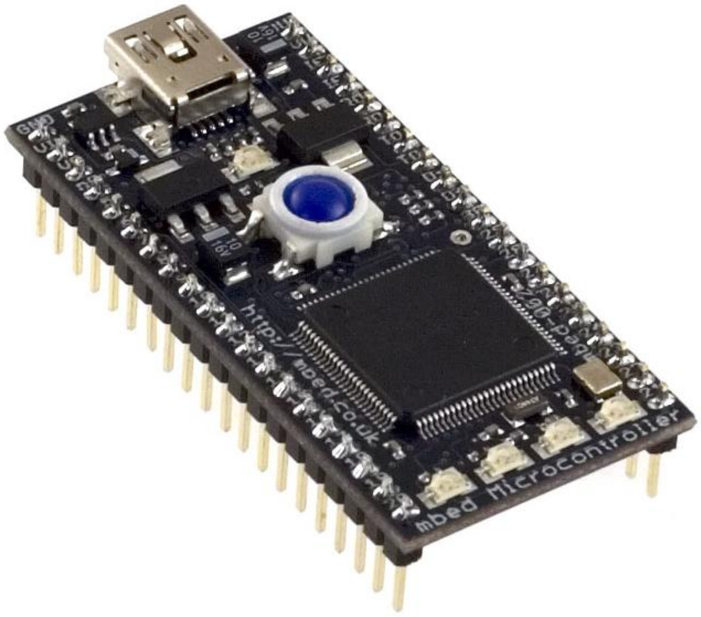
What does ARM do?

- ARM designs technology that lies at the heart of advanced digital products

ARM Overview

- ARM is the world's leading semiconductor IP company and The Architecture for the Digital World ®

- Over 15 billion ARM technology based chips shipped to date
- Unrivalled Partner ecosystem
  - Over 640 processor licenses sold to more than 200 companies
  - Millions of developers; billions of users
- ARM has the right technology – optimized for a mobilizing world
- We're customer-focused – listening harder and responding faster



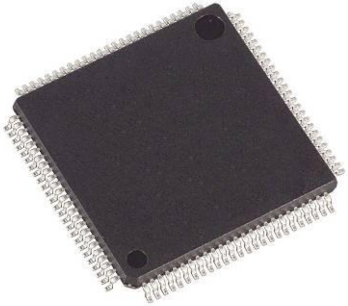
mbed
Hello World!

Introduction to mbed

What’s happening in Microcontrollers?

- Microcontrollers are getting cheap
  - 32-bit ARM Cortex-M3 Microcontrollers @ $1

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

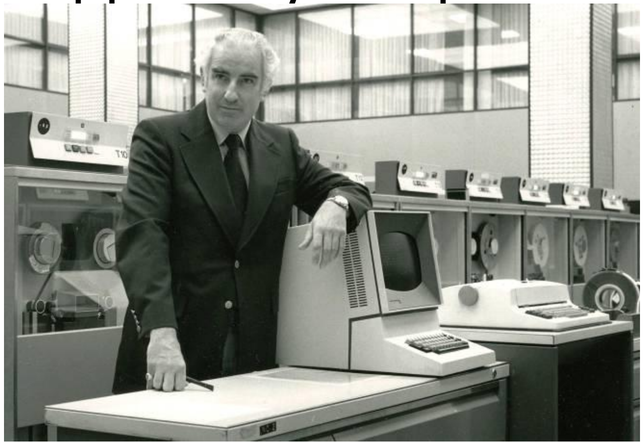


- Microcontrollers are getting interactive
  - Internet connectivity, new sensors and actuators

- 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

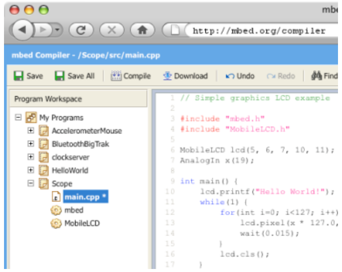
3D Printing

2D/3D Design

Web Frameworks

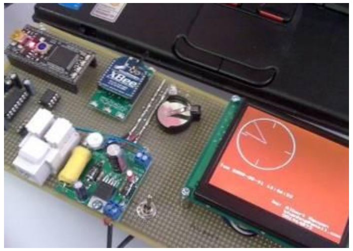
mbed.org - Rapid Prototyping for MCUs

- 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



```
// Simple graphics LCD example

#include "mbed.h"
#include "MobileLCD.h"

MobileLCD lcd(5, 6, 7, 10, 11);
AnalogIn x(19);

int main() {
    lcd.printf("Hello World!");
    while(1) {
        for(int i=0; i<127; i++)
            lcd.pixel(x * 127.0,
            wait(0.015);
        }
        lcd.cls();
    }
```

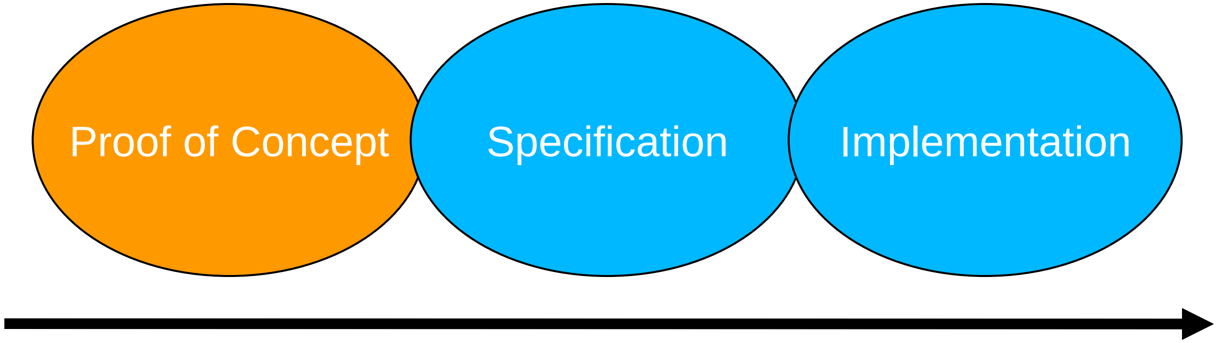
- Focused on rapid prototyping for a diverse audience
  - DIP form-factor, High-level APIs, Developer website
  - Technology and tradeoffs to enable fast experiments
  - Creates new applications for MCU technology



- 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

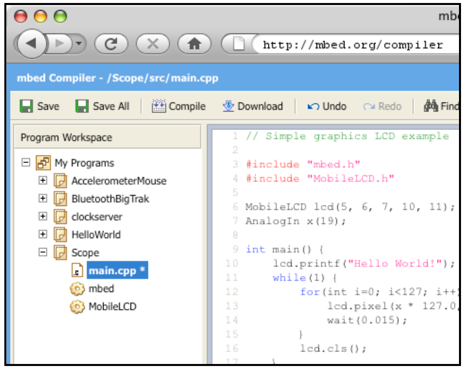
- Focus on tools supporting the earliest stage of design
  - Point of entry and Getting Started
  - Experimentation and Rapid Prototyping



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

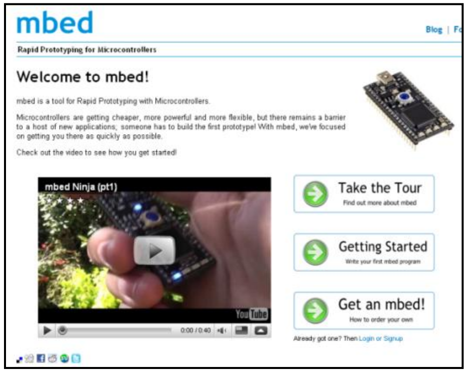
mbed Rapid Prototyping Platform

- Complete Hardware, Software and Web 2.0 Solution



```
// Simple graphics LCD example

#include "mbed.h"
#include "MobileLCD.h"

MobileLCD lcd(5, 6, 7, 10, 11);
AnalogIn x(19);

int main() {
    lcd.printf("Hello World!");
    while(1) {
        for(int i=0; i<127; i++
            lcd.pixel(x * 127.0
            wait(0.015);
        }
        lcd.cls();
```

Dedicated Developer Website



Welcome to mbed!

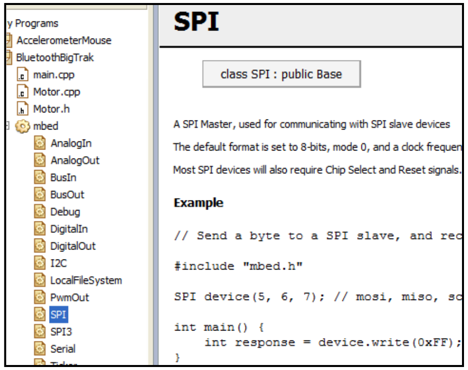
mbed is a tool for Rapid Prototyping with Microcontrollers.

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!

Already got one? Then Login or Signup

Lightweight Online Compiler



SPI

class SPI : public Base

A SPI Master, used for communicating with SPI slave devices

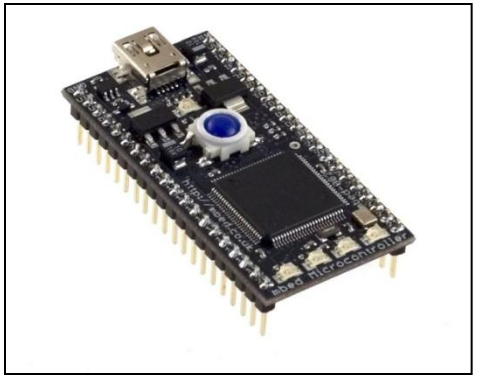
The default format is set to 8-bits, mode 0, and a clock frequen

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);
}
```

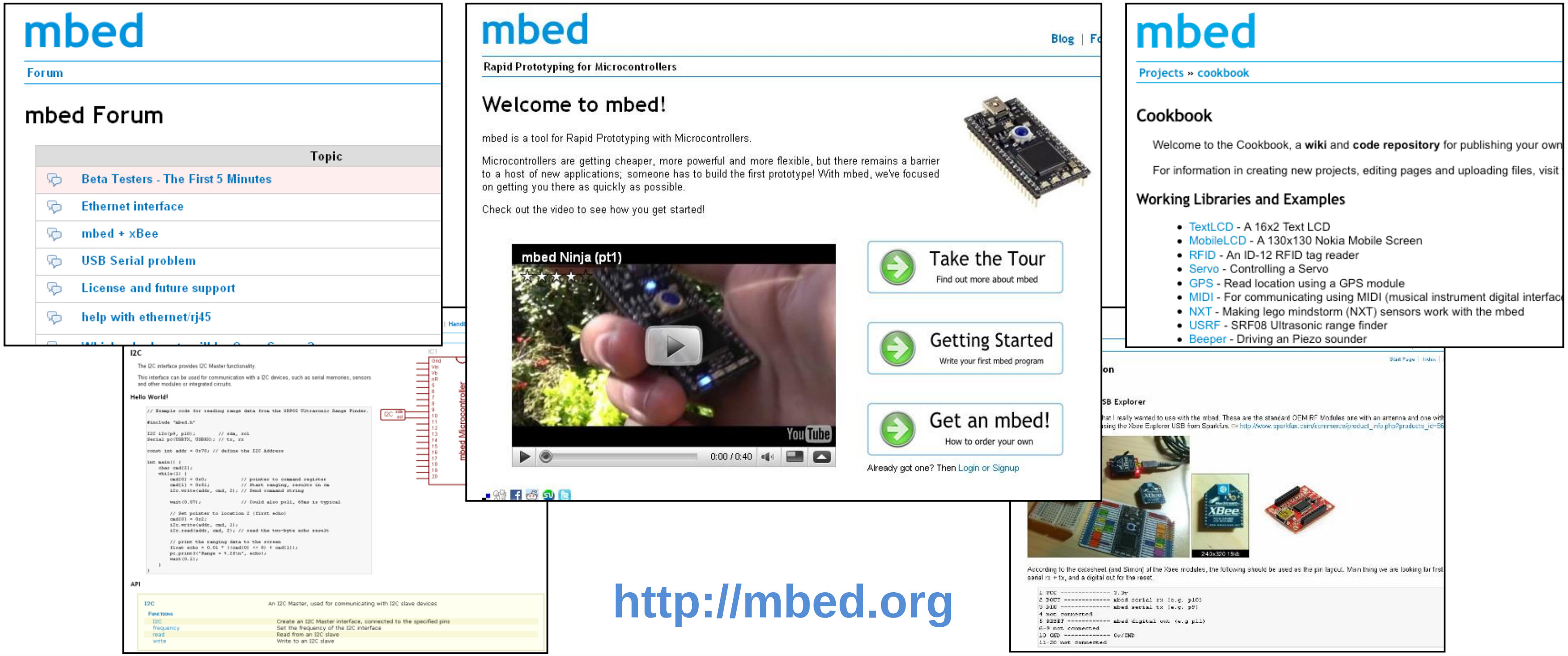
High-level Peripheral APIs



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 | Fo

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 interfac
- NXT - Making lego mindstorm (NXT) sensors work with the mbed
- USRF - SRF08 Ultrasonic range finder
- Beeper - Driving an Piezo sounder

I2C

The I2C interface provides I2C Master functionality.

This interface can be used for communication with a I2C devices, such as serial memories, sensors and other modules or integrated circuits.

Hello World!

```
// Example code for reading range data from the SRF08 Ultrasonic Range Finder.

#include "mbed.h"

I2C i2c(p9, p10);        // sda, scl
Serial pc(USBTX, USBRX); // tx, rx

const int addr = 0x70; // define the I2C Address

int main() {
    char cmd[2];
    while(1) {
        cmd[0] = 0x0;            // pointer to command register
        cmd[1] = 0x51;           // Start ranging, results in cm
        i2c.write(addr, cmd, 2); // Send command string

        wait(0.07);              // Could also poll, 65ms is typical

        // Set pointer to location 2 (first echo)
        cmd[0] = 0x2;
        i2c.write(addr, cmd, 1);
        i2c.read(addr, cmd, 2); // read the two-byte echo result

        // print the ranging data to the screen
        float echo = 0.01 * ((cmd[0] << 8) + cmd[1]);
        pc.printf("Range = %.2f\n", echo);
        wait(0.1);
    }
}
```

API

I2C	An I2C Master, used for communicating with I2C slave devices
Functions	
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

Start Page | Index

SB Explorer

hat I really wanted to use with the mbed. These are the standard OEM RF Modules one with an antenna and one with sing the Xbee Explorer USB from Sparkfun. http://www.sparkfun.com/commerce/product_info.php?products_id=86

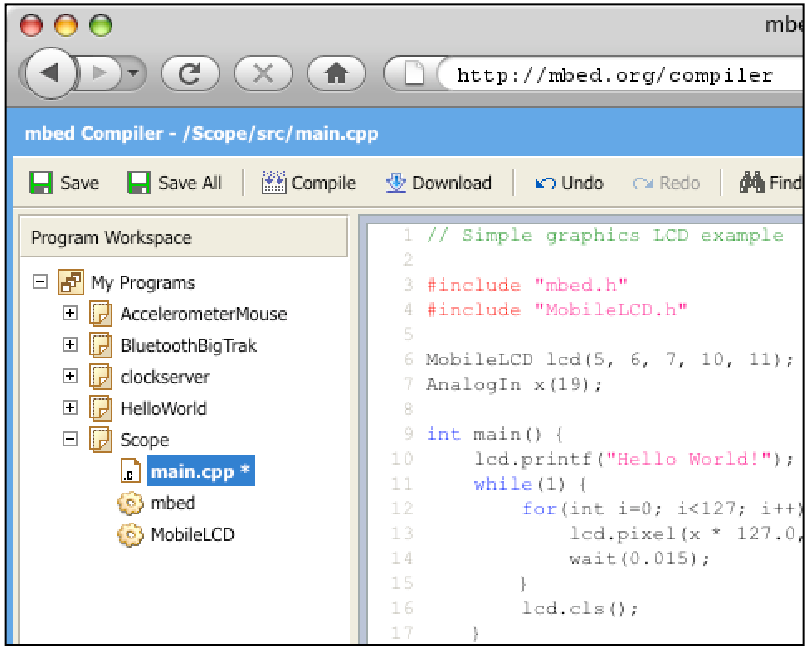
According to the datasheet (and Simon) of the Xbee modules, the following should be used as the pin layout. Main thing we are looking for first serial rx + tx, and a digital out for the reset.

```
1 VCC --------------- 3.3v
2 DOUT -------------- mbed serial rx (e.g. p10)
3 DIN --------------- mbed serial tx (e.g. p9)
4 not connected
5 RESET ------------- mbed digital out (e.g p11)
6-9 not connected
10 GND -------------- 0v/GND
11-20 not connected
```

http://mbed.org

mbed Compiler

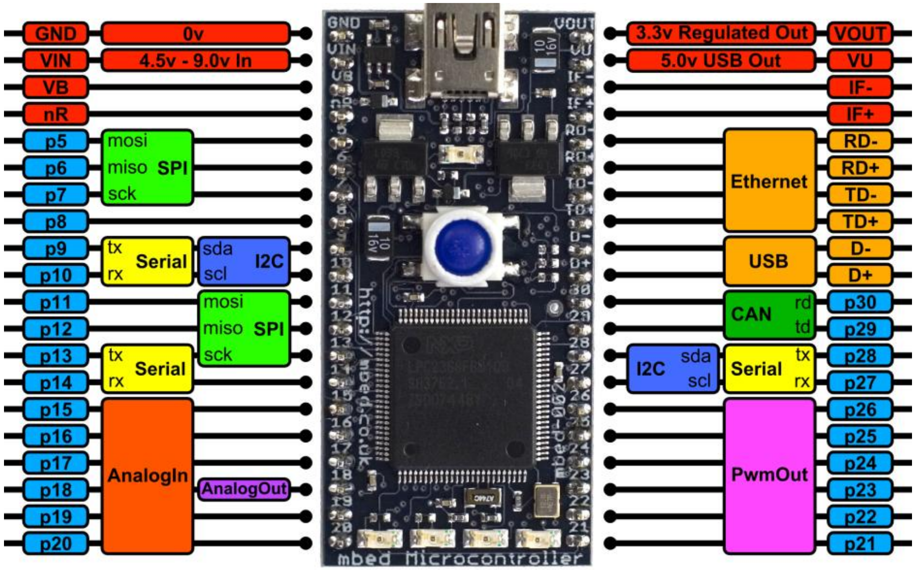
- Lightweight Online Compiler
  - Web 2.0 browser-based IDE with personal workspace “in the cloud”
  - Nothing to install or configure, login from anywhere
  - Industry leading RVCT 4.1 back end. It is a real tool!

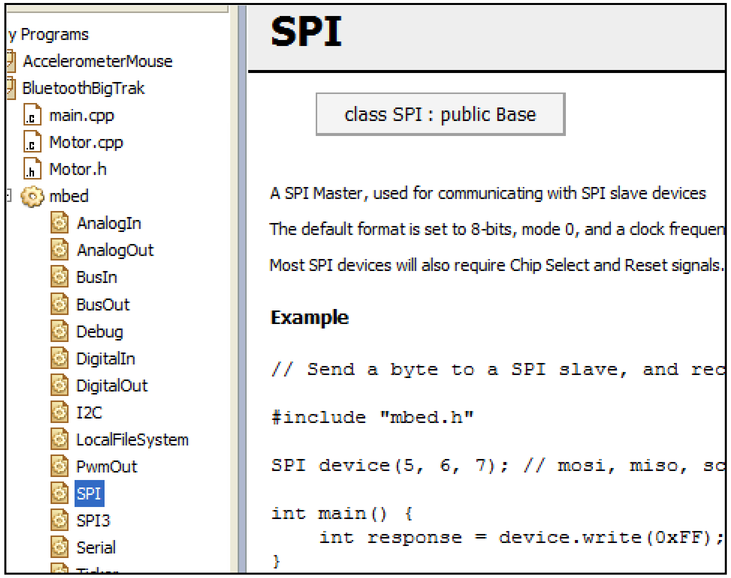


```
// Simple graphics LCD example

#include "mbed.h"
#include "MobileLCD.h"

MobileLCD lcd(5, 6, 7, 10, 11);
AnalogIn x(19);

int main() {
    lcd.printf("Hello World!");
    while(1) {
        for(int i=0; i<127; i++)
            lcd.pixel(x * 127.0,
            wait(0.015);
        }
        lcd.cls();
    }
```

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





SPI

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

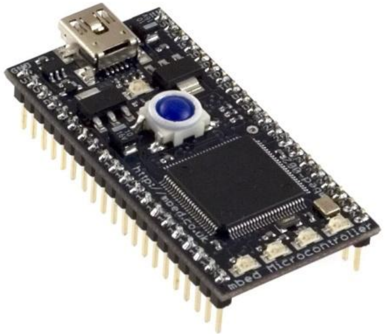
Most SPI devices will also require Chip Select and Reset signals.

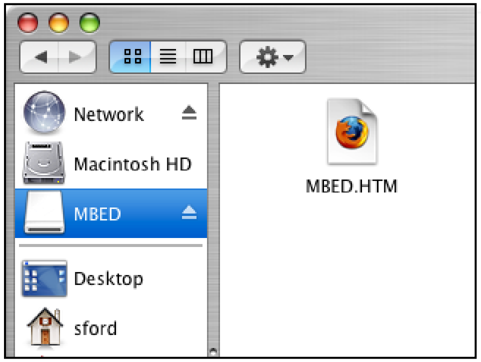
Example

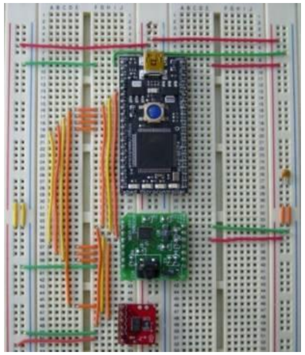
```
// Send a byte to a SPI slave, and rec

#include "mbed.h"

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

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

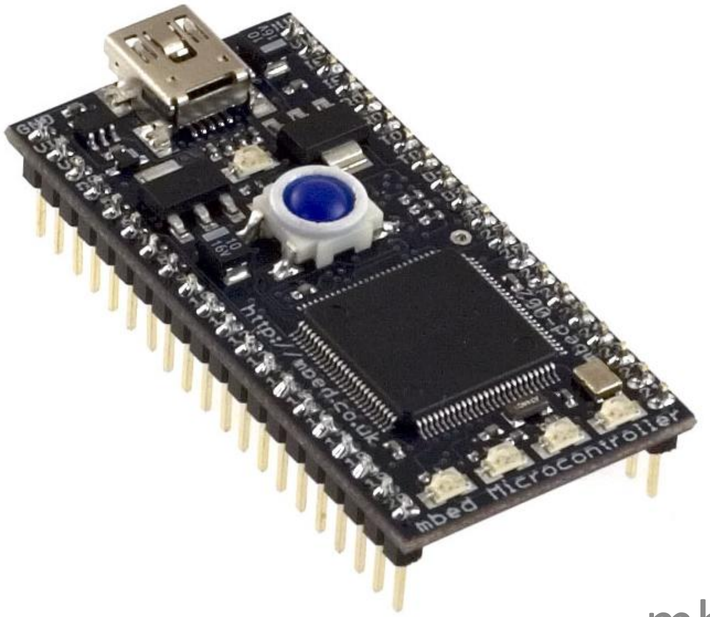
mbed Microcontroller

- 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









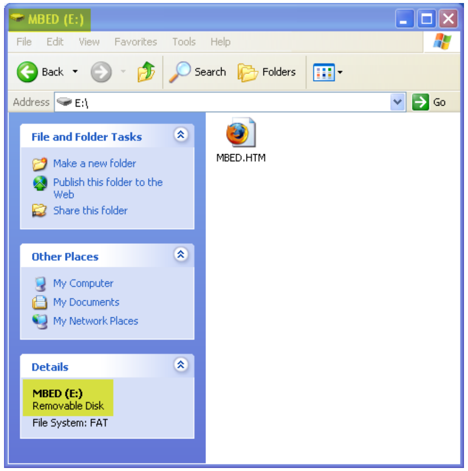
mbed
Hello World

Lab 1

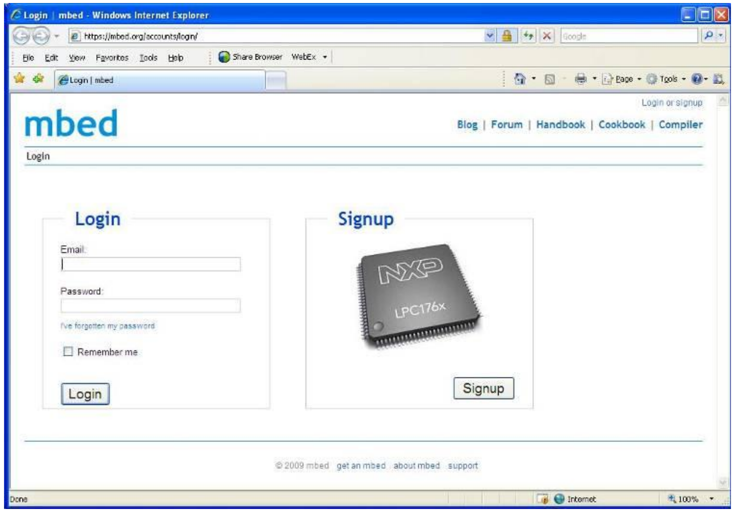
mbed registration and hello world!

Registration

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



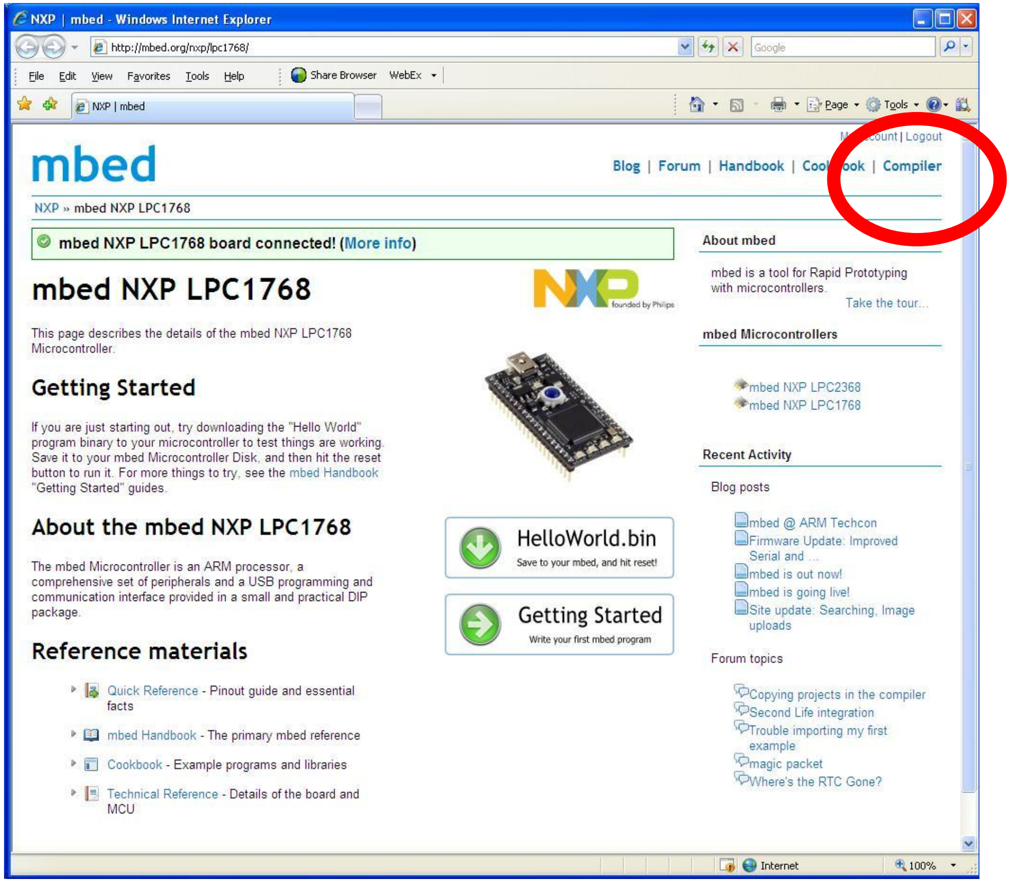
- Log in or sign up for a new account
- The mbed microcontroller contains your license to the compiler


Login | mbed - Windows Internet Explorer
https://mbed.org/accounts/login/
File Edit View Favorites Tools Help
Share Browser WebEx
Login | mbed
Page Tools
Login or signup
mbed
Blog | Forum | Handbook | Cookbook | Compiler
Login
Login
Email:
Password:
I've forgotten my password
☐ Remember me
Login
Signup
NXP
LPC176x
Signup
© 2009 mbed get an mbed about mbed support
Done
Internet
100%

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


NXP | mbed - Windows Internet Explorer
http://mbed.org/nxp/lpc1768/
Google
File Edit View Favorites Tools Help
Share Browser WebEx
NXP | mbed
Page
Tools
M…count | Logout
mbed
Blog | Forum | Handbook | Cook…ook | Compiler
NXP » mbed NXP LPC1768
founded by Philips
Internet
100%

mbed NXP LPC1768 board connected! (More info)

mbed NXP LPC1768

This page describes the details of the mbed NXP LPC1768 Microcontroller.

Getting Started

If you are just starting out, try downloading the "Hello World" program binary to your microcontroller to test things are working. Save it to your mbed Microcontroller Disk, and then hit the reset button to run it. For more things to try, see the mbed Handbook "Getting Started" guides.

About the mbed NXP LPC1768

The mbed Microcontroller is an ARM processor, a comprehensive set of peripherals and a USB programming and communication interface provided in a small and practical DIP package.

Reference materials

- Quick Reference - Pinout guide and essential facts
- mbed Handbook - The primary mbed reference
- Cookbook - Example programs and libraries
- Technical Reference - Details of the board and MCU

HelloWorld.bin
Save to your mbed, and hit reset!

Getting Started
Write your first mbed program

About mbed

mbed is a tool for Rapid Prototyping with microcontrollers.
Take the tour...

mbed Microcontrollers

- mbed NXP LPC2368
- mbed NXP LPC1768

Recent Activity

Blog posts

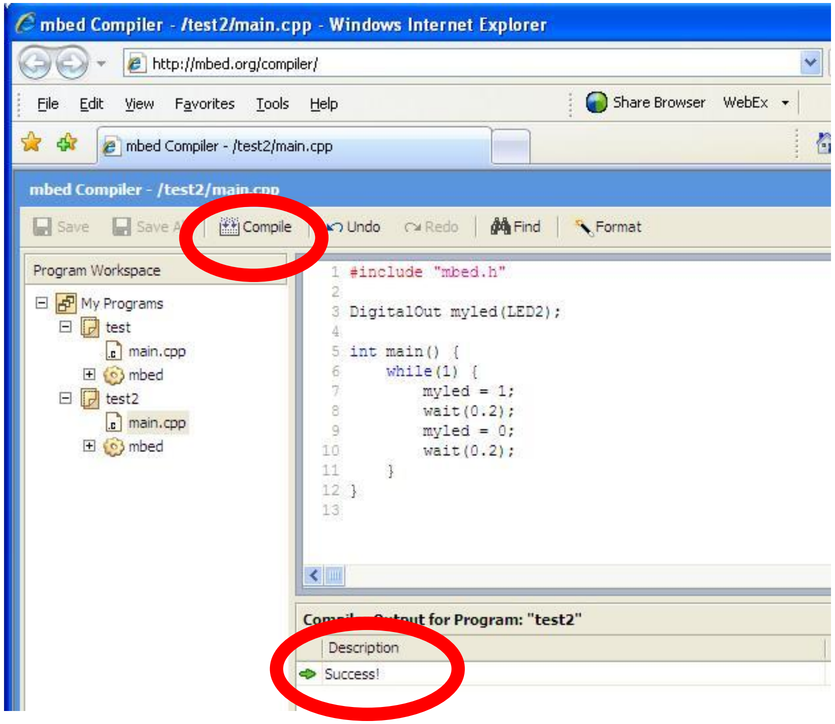
- mbed @ ARM Techcon
- Firmware Update: Improved Serial and ...
- mbed is out now!
- mbed is going live!
- Site update: Searching, Image uploads

Forum topics

- Copying projects in the compiler
- Second Life integration
- Trouble importing my first example
- magic packet
- Where's the RTC Gone?

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



mbed Compiler - /test2/main.cpp - Windows Internet Explorer
http://mbed.org/compiler/
File
Edit
View
Favorites
Tools
Help
Share Browser
WebEx
mbed Compiler - /test2/main.cpp
mbed Compiler - /test2/main.cpp
Save
Save A
Compile
Undo
Redo
Find
Format
Program Workspace
My Programs
test
main.cpp
mbed
test2
main.cpp
mbed

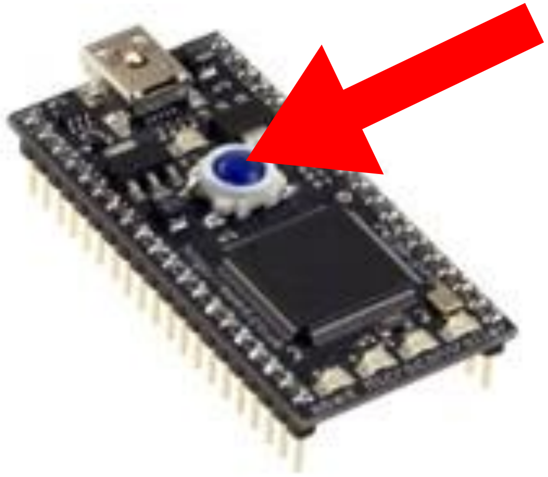
```
#include "mbed.h"

DigitalOut myled(LED2);

int main() {
    while(1) {
        myled = 1;
        wait(0.2);
        myled = 0;
        wait(0.2);
    }
}
```

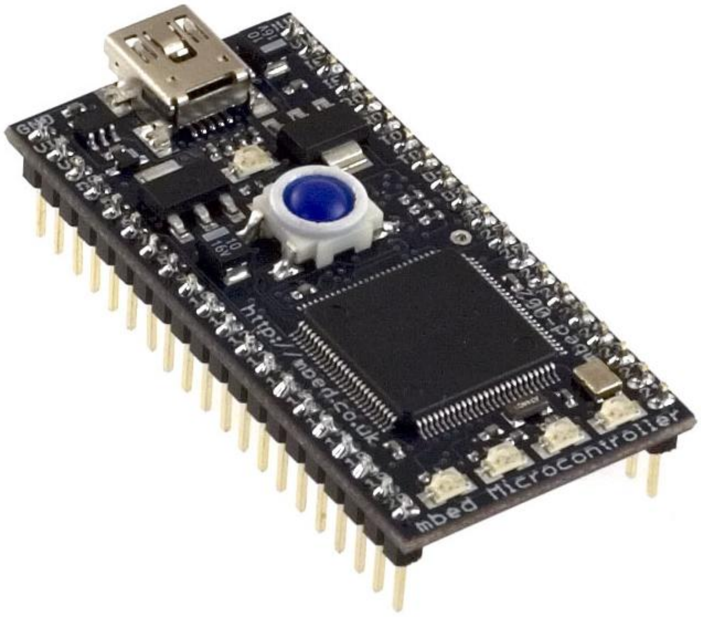
Compile Output for Program: "test2"

Description
Success!

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

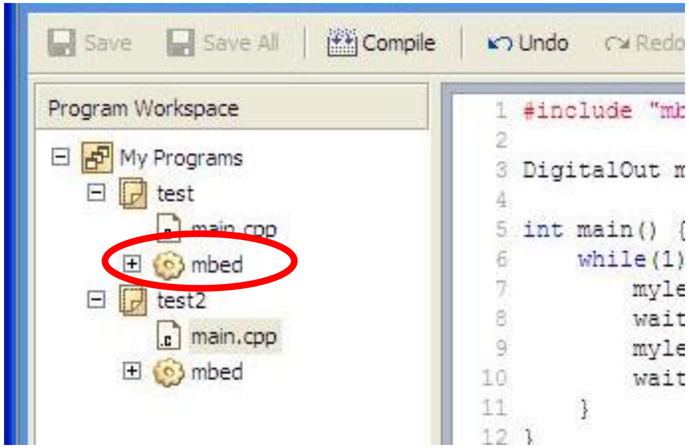
Rapid Prototyping: Other IO

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.



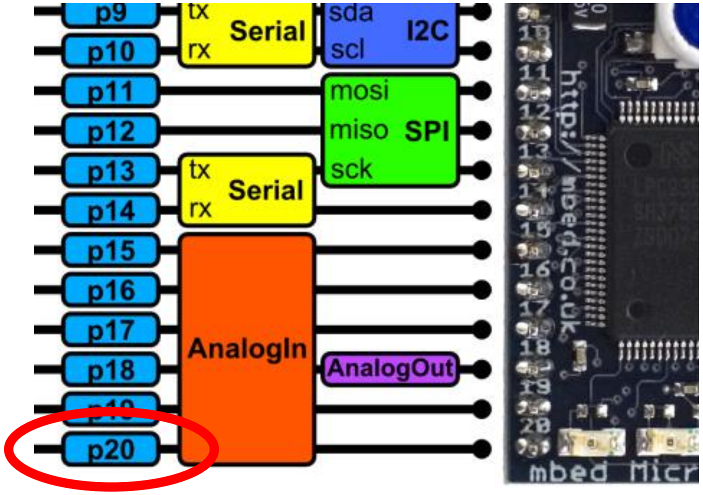
```
#include "mb

DigitalOut m

int main() {
    while(1)
        myle
        wait
        myle
        wait
    }
}
```

- 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)

- Wire your potentiometer between GND (0v) and Vout (3.3v), and connect the wiper (the middle pin) to pin “p20” – an AnalogIn

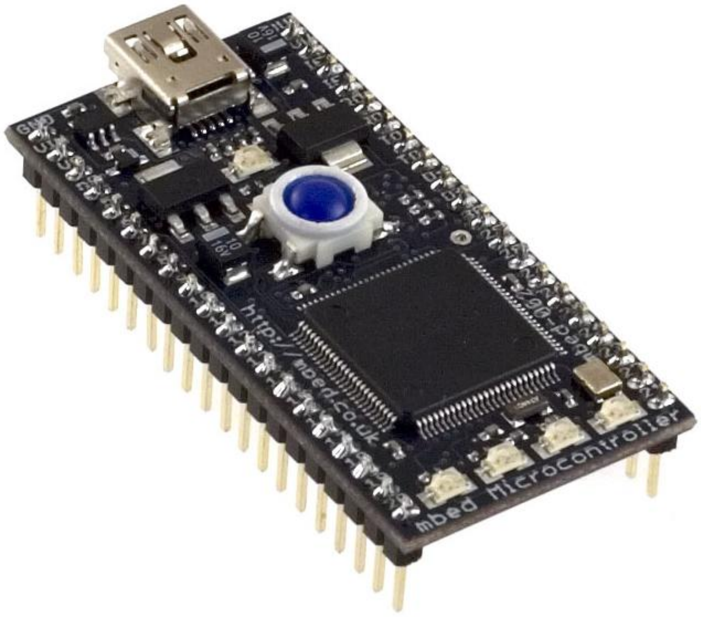


Challenge: DigitalOut and Analog Input

- Write a program to give the LED in the first blinky program a delay of 1-5 seconds.

```
#include “mbed.h”

DigitalOut myled(LED1);
AnalogIn pot(p20);


int main () {
  while(1) {
    myled = !myled;                  // toggle
    wait (1.0 + (4.0 * pot.read())); // 1.0s - 5.0s
  }
}
```

- 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 TMP102 which has a digital interface using the I2C bus.

Connecting up the sensor

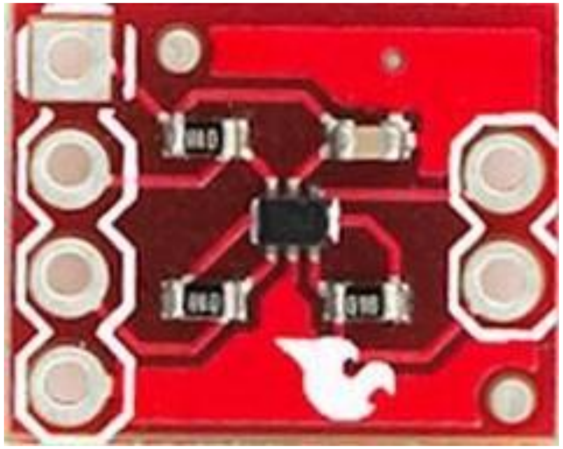
- The TMP102 has just four pins, Vcc, Gnd for the power, and SCL, SDA for the I2C interface.

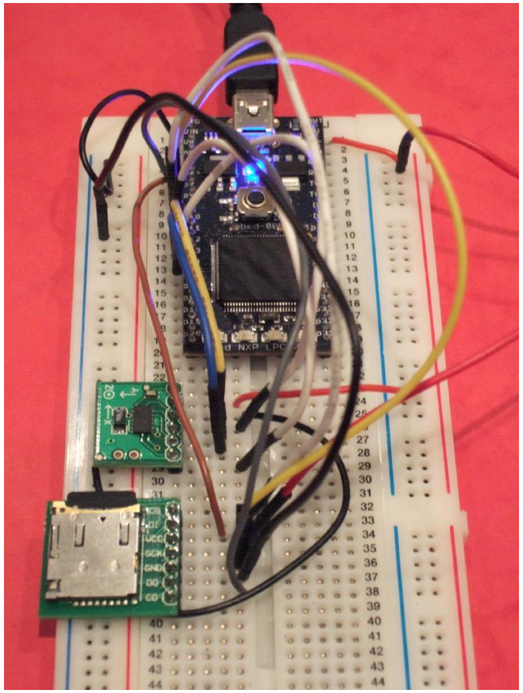
Vcc (Vout)

SDA (p9)

SCL (p10)

GND

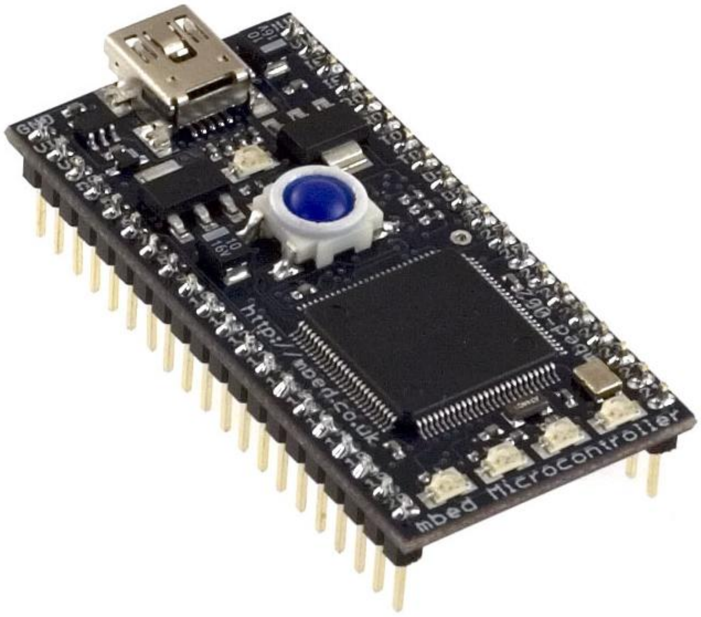




- As before, mbed keeps I2C simple
  - http://mbed.org/handbook/I2C
  - http://mbed.org/cookbook/TMP102-Temperature-Sensor

Challenge : Interfacing with sensors

- Using the Cookbook 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.



mbed
Hello World

Lab 4

Rapid Prototyping: Output device, Text LCD

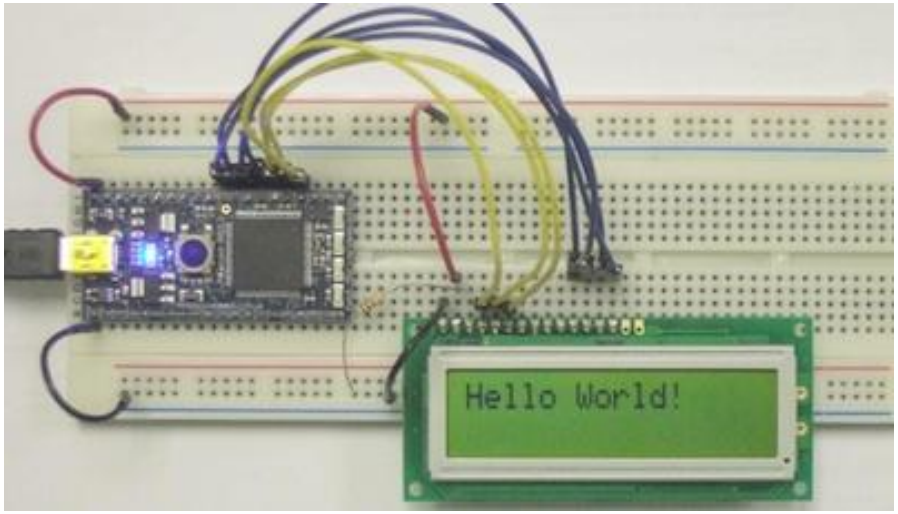
Example : Output device, Text LCD

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

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

Connecting up the TextLCD

- Text LCD modules have almost standardised, although they still have their quirks.

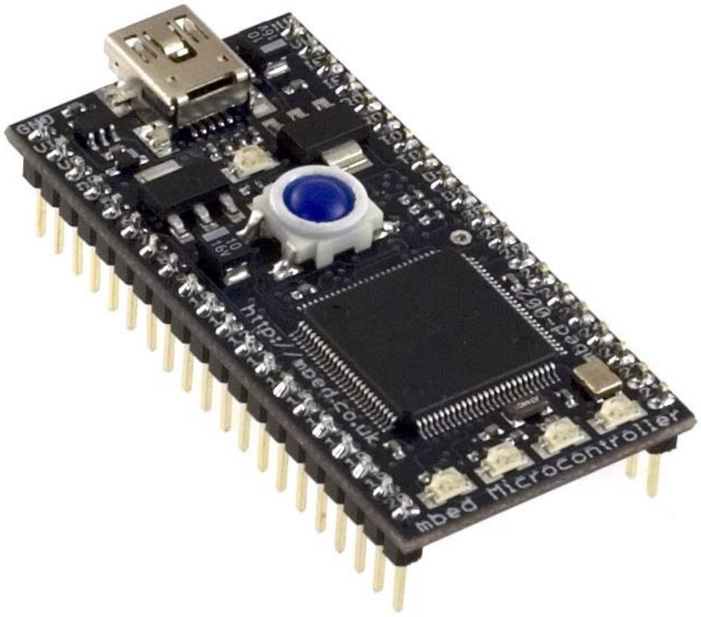


- Six wires and a resistor for contrast
- As before, mbed keeps it simple
  - Standard C/C++ interface via printf
  - http://mbed.org/cookbook/Text-LCD

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

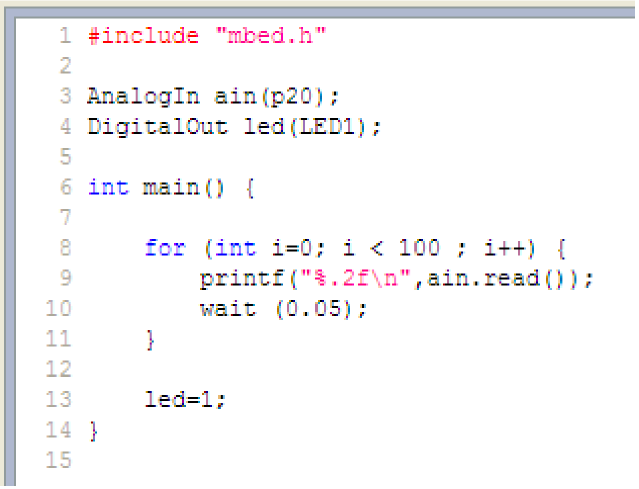
- Applications often include data logging capabilities, and access to the data often involves bespoke software and interface cables.

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

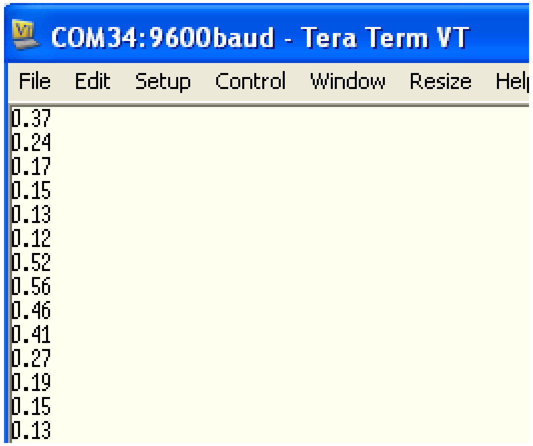
- For the purposes of the experiment, we will be displaying and logging noise from an unconnected ADC. Touching the pin will influence the noise, it is a demonstration, imagine it is real data!

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



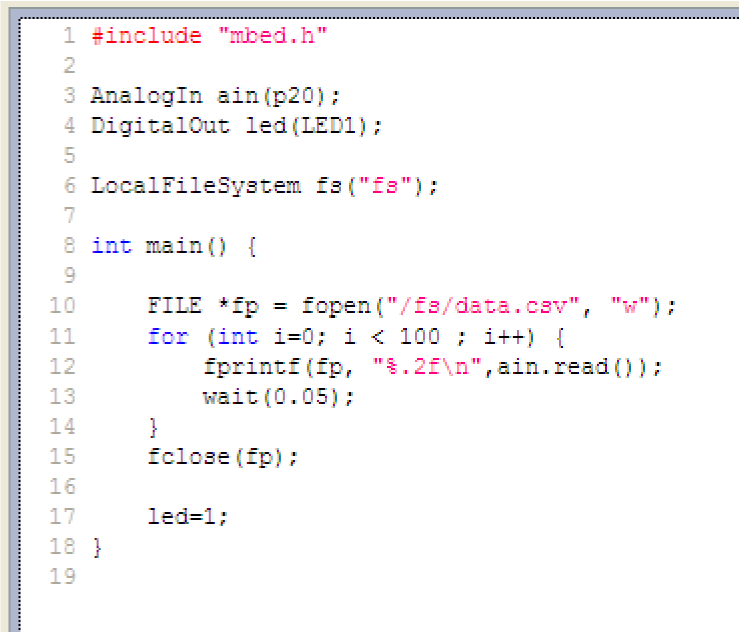
```
#include "mbed.h"

AnalogIn ain(p20);
DigitalOut led(LED1);

int main() {

    for (int i=0; i < 100 ; i++) {
        printf("%.2f\n",ain.read());
        wait (0.05);
    }

    led=1;
}

```



```
0.37
0.24
0.17
0.15
0.13
0.12
0.52
0.56
0.46
0.41
0.27
0.19
0.15
0.13
```

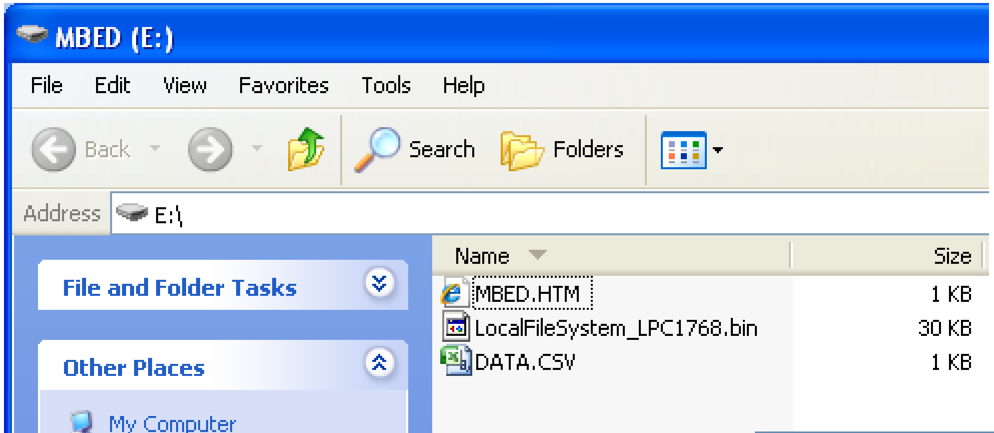
Example : Data Logging

- The mbed Flash disk is accessible from user code using the LocalFileSystem object
- Standard C file handling techniques apply
- fscanf for runtime configuration
- fprintf for data logging purposes
- This example logs 100 samples to a CSV file



```
#include "mbed.h"

AnalogIn ain(p20);
DigitalOut led(LED1);

LocalFileSystem fs("fs");

int main() {

    FILE *fp = fopen("/fs/data.csv", "w");
    for (int i=0; i < 100 ; i++) {
        fprintf(fp, "%.2f\n",ain.read());
        wait(0.05);
    }
    fclose(fp);

    led=1;
}
```

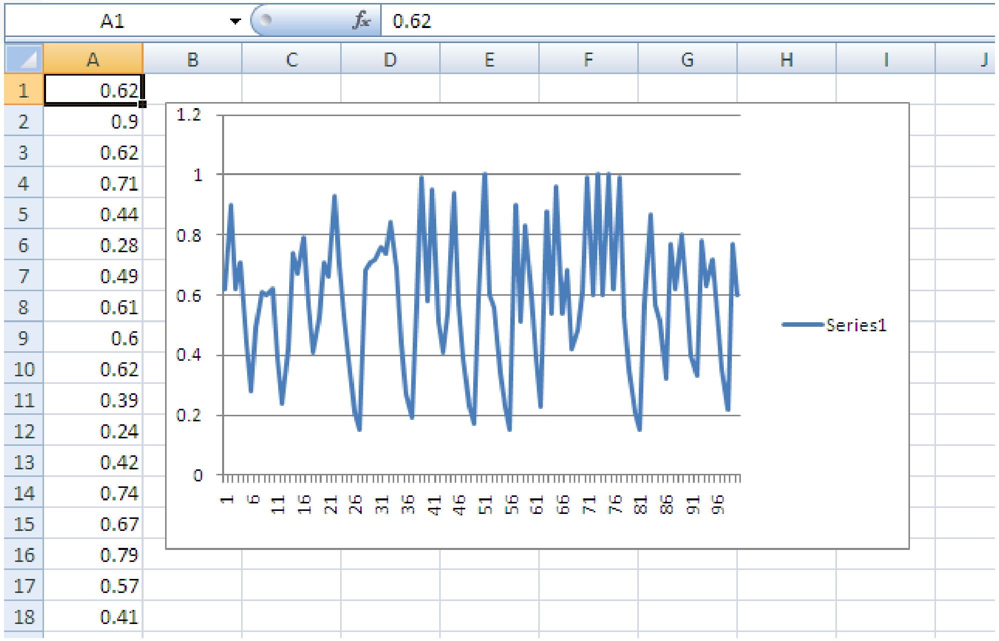
Data quickly visible to a PC



Name	Size
MBED.HTM	1 KB
LocalFileSystem_LPC1768.bin	30 KB
DATA.CSV	1 KB

While the program executes the flash drive disappears from the PC, and returns when the file is closed

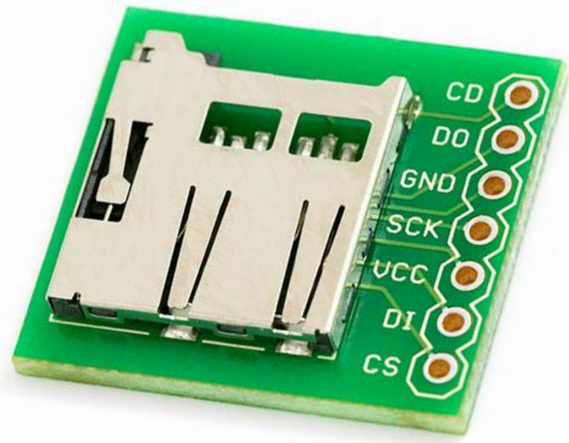
Logging to a CSV file means Excel can open the file and interpret, manipulate or plot the data.



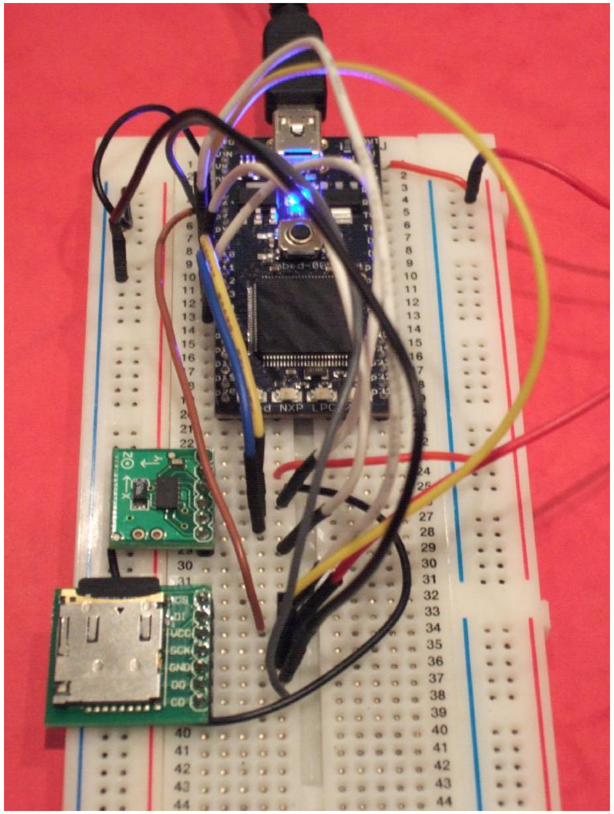
	A
1	0.62
2	0.9
3	0.62
4	0.71
5	0.44
6	0.28
7	0.49
8	0.61
9	0.6
10	0.62
11	0.39
12	0.24
13	0.42
14	0.74
15	0.67
16	0.79
17	0.57
18	0.41

Extend it to store lots of data

- Perhaps a final system might want to store lots of data
  - SD cards are ideal, ubiquitous and recognisable by everyone



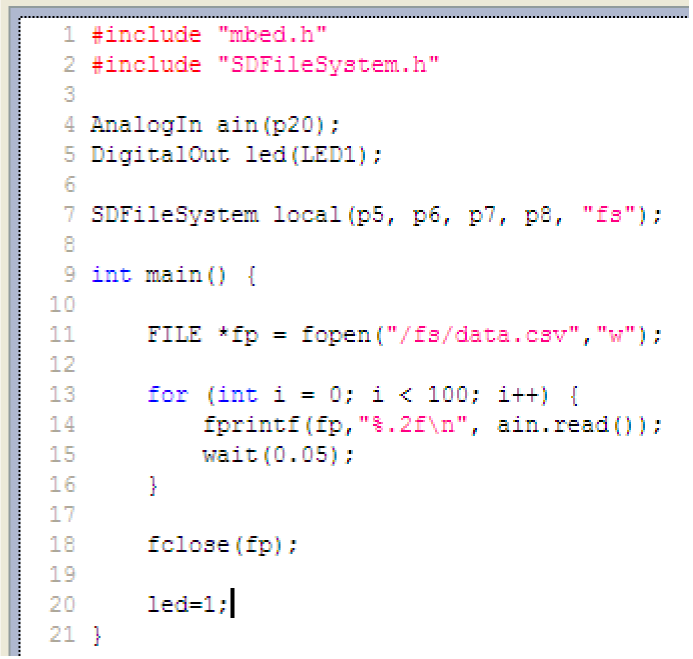
GND		
MISO	–	p6
SCL	-	p7
Vcc		
MOSI	–	p5
nCS	-	p8



- Hardware for an SD Card is minimal
  - SPI Port connection using simple breakout
- As before, mbed keeps it simple

Extend it to store lots of data

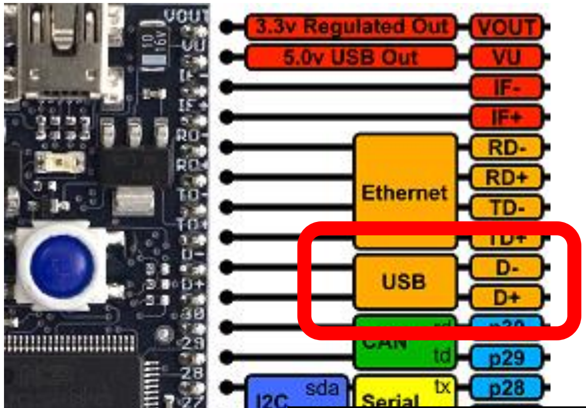
- Import the SDFileSystem Library into the project
- Include the SDFileSystem header
- Swap LocalFileSystem for SDFileSystem
- Everything else remains the same



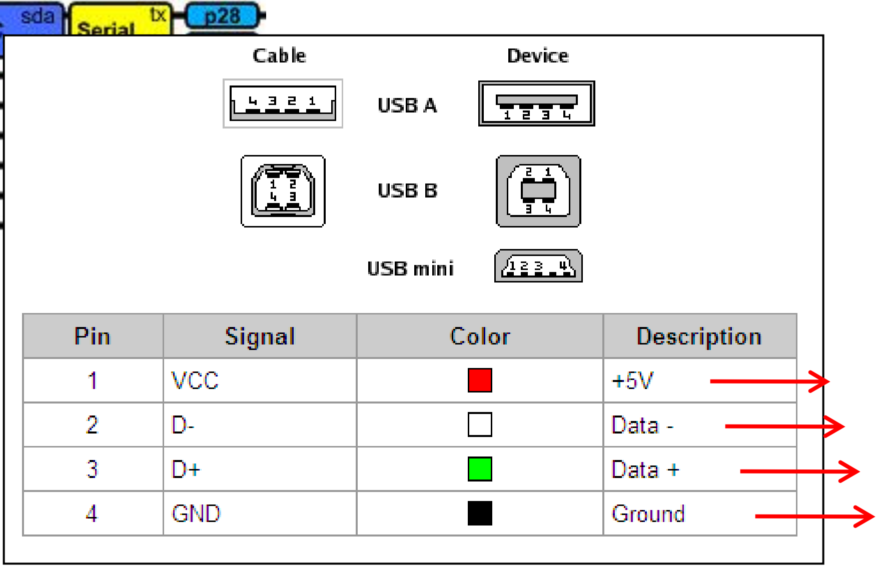
```
#include "mbed.h"
#include "SDFileSystem.h"

AnalogIn ain(p20);
DigitalOut led(LED1);

SDFileSystem local(p5, p6, p7, p8, "fs");

int main() {

    FILE *fp = fopen("/fs/data.csv","w");

    for (int i = 0; i < 100; i++) {
        fprintf(fp,"%.2f\n", ain.read());
        wait(0.05);
    }

    fclose(fp);

    led=1;
}
```

What about a USB drive?

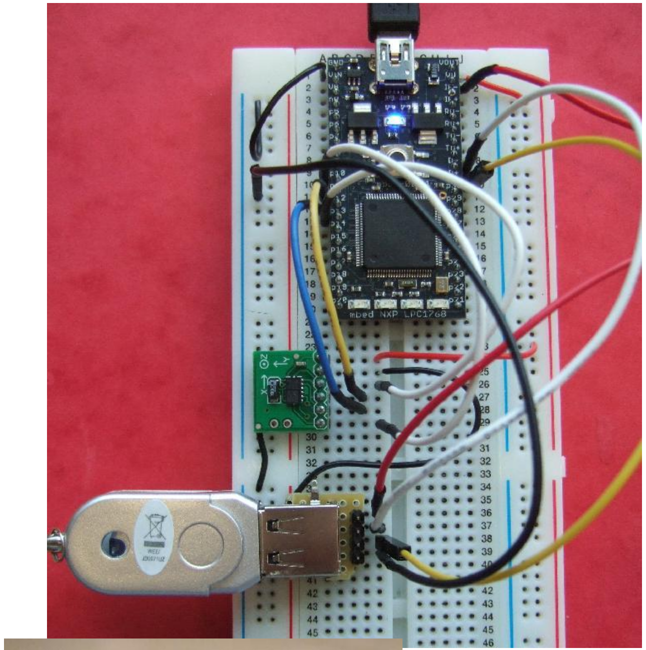
- USB Host hardware is minimal; a USB A connector

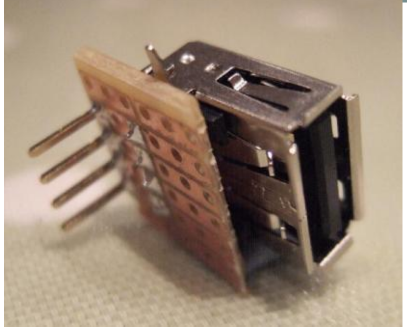


VCC	->	VU
D-	->	D-
D+	->	D+
GND	->	GND



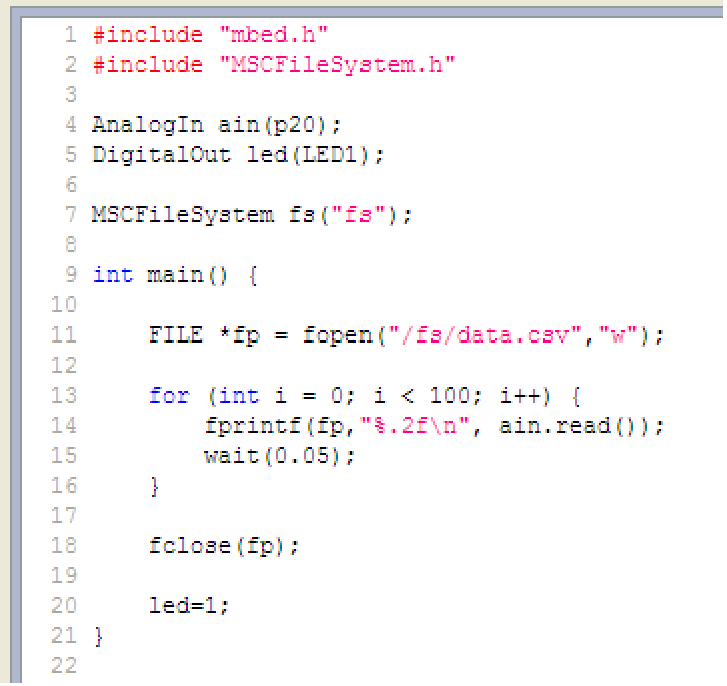
Pin	Signal	Color	Description
1	VCC	🟥	+5V
2	D-	□	Data -
3	D+	🟩	Data +
4	GND	■	Ground





What about a USB drive?

- On your project, right click -> Import Library -> MSCFileSystem
  - Add #include for MSCFileSystem
  - Call it “fs”, as before
  - The change in storage medium is transparent to the application

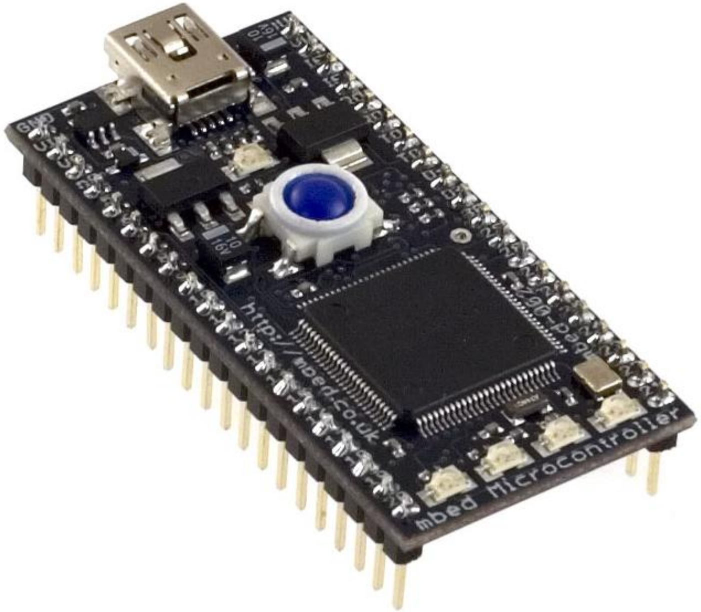


```
#include "mbed.h"
#include "MSCFileSystem.h"

AnalogIn ain(p20);
DigitalOut led(LED1);

MSCFileSystem fs("fs");

int main() {

    FILE *fp = fopen("/fs/data.csv","w");

    for (int i = 0; i < 100; i++) {
        fprintf(fp,"%.2f\n", ain.read());
        wait(0.05);
    }

    fclose(fp);

    led=1;
}

```

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

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