



电子科技大学
格拉斯哥学院
Glasgow College, UESTC

UESTC1008: Microelectronic Systems

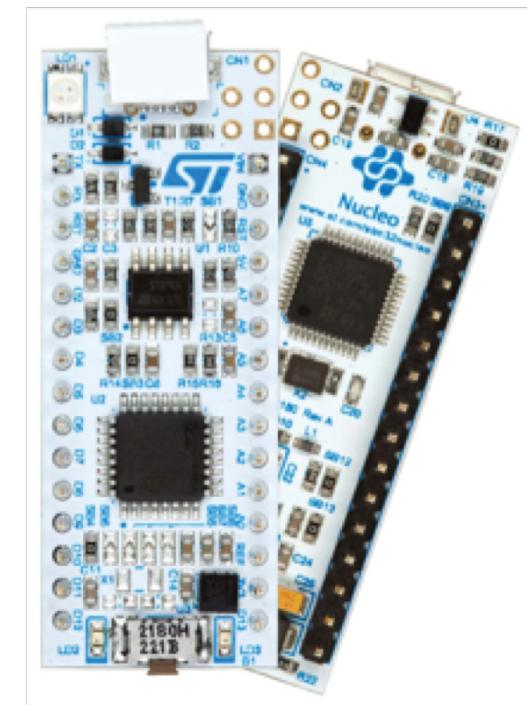
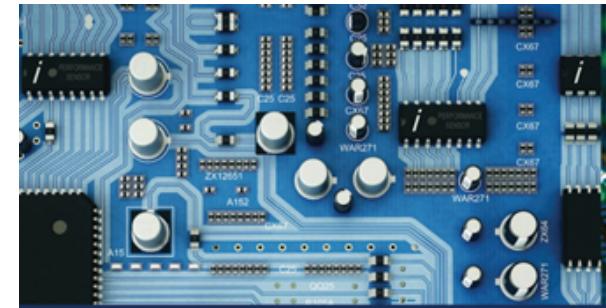
Academic year 2019/2020 – Semester 2 – Presentation 2

Qammer H. Abbasi, Lei Zhang, Sajjad Hussain, Muhammad Ali Imran and Guodong Zhao
{qammer.abbasi,Lei.zhang,sajjad.Hussain,Muhammad.Imran,guodong.zhao}@glasgow.ac.uk

“A good student never steals or cheats”

Agenda

- Review of previous lecture
 - Mbed details
 - Mbed programming
 - Summary



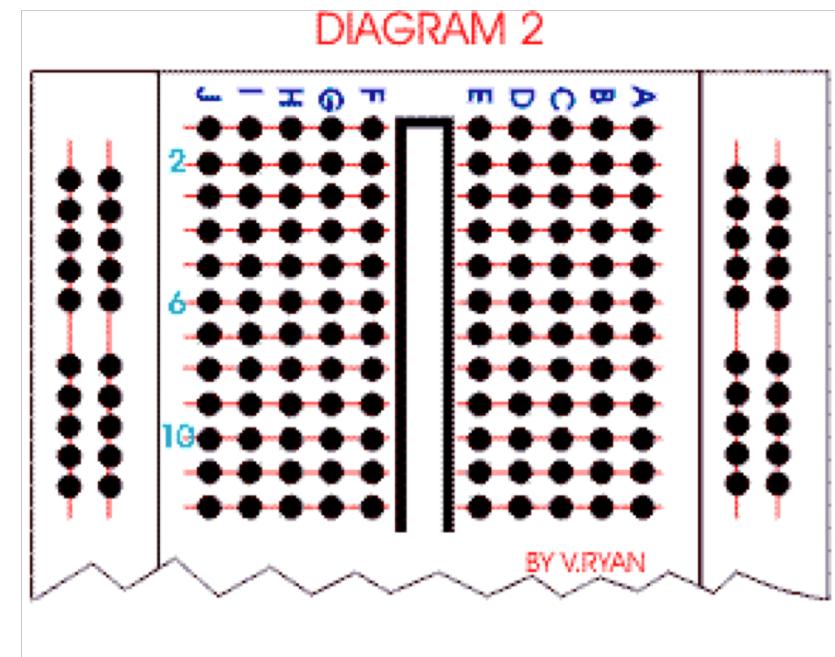
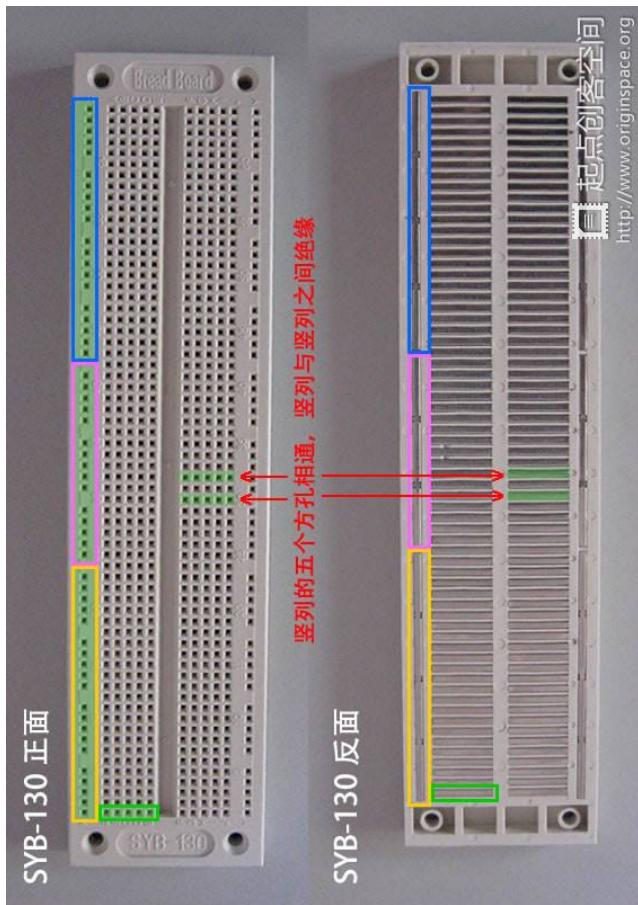
Toolkit and mbed box

- The set of experiments is based around the mbed microcontroller development board
- Each student will be given a toolkit and an mbed, which you will use widely throughout your years of study at Glasgow College, UESTC
- You need to keep your mbed for several courses so treat it well and protect it
- You will also be given a small breadboard to enable you to easily connect components to your beloved mbed

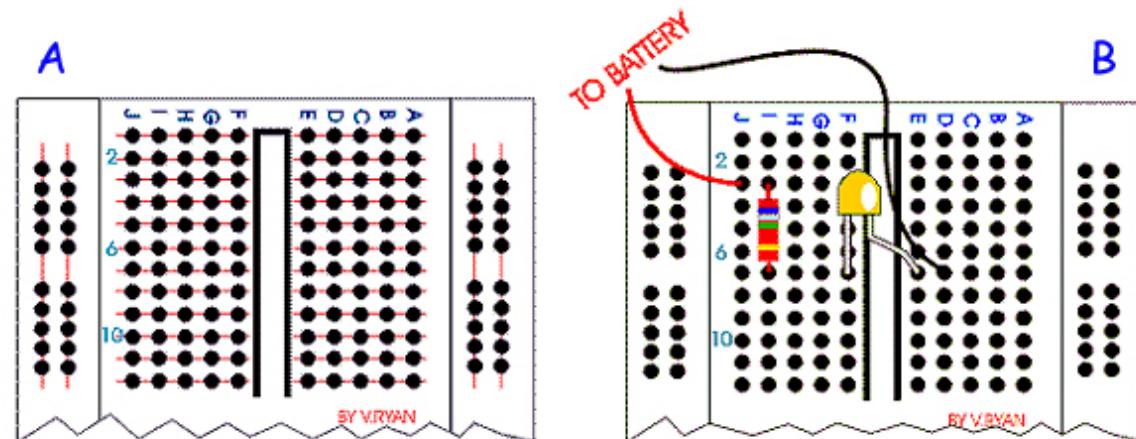
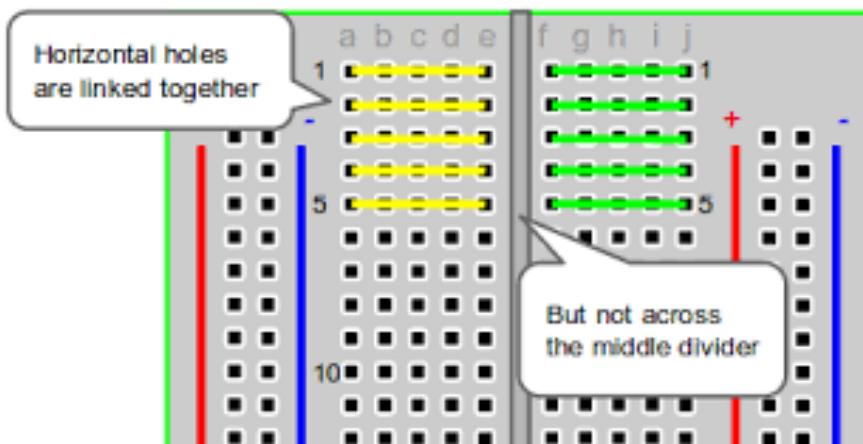
Toolkit



Breadboard



Breadboard

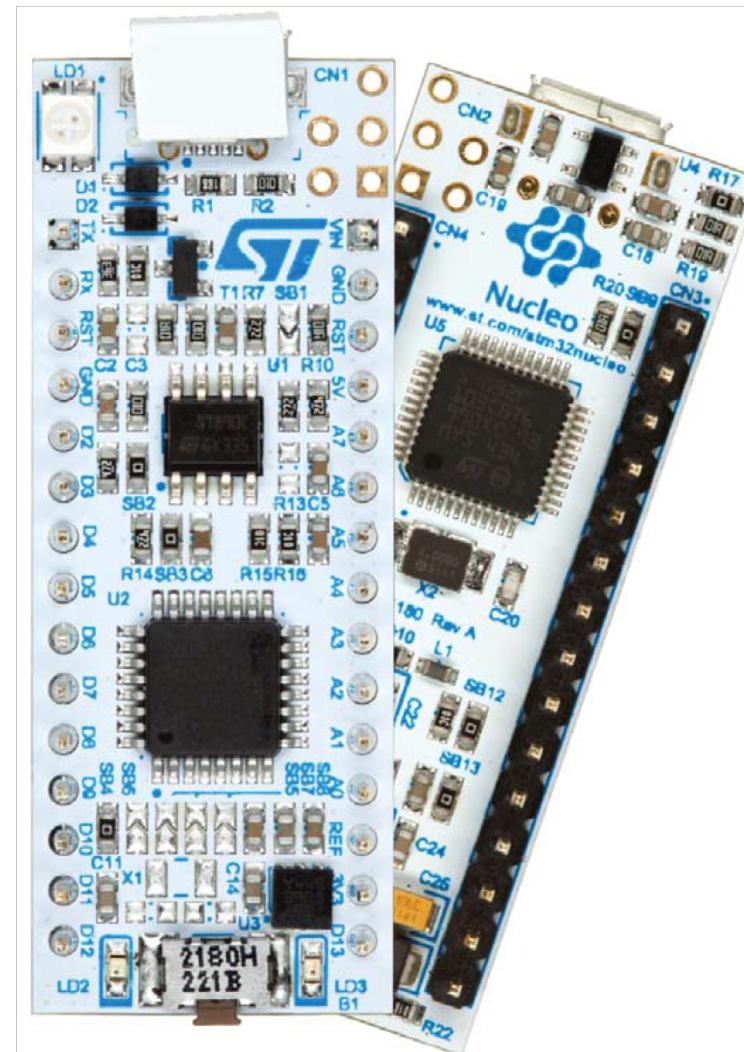


mbed Box

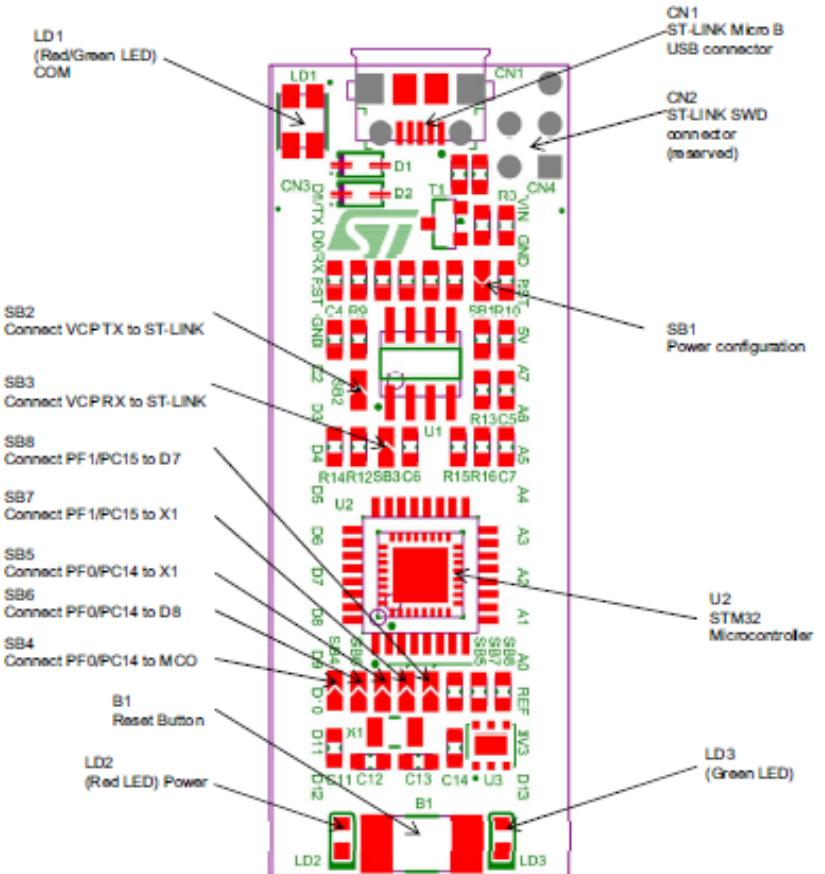
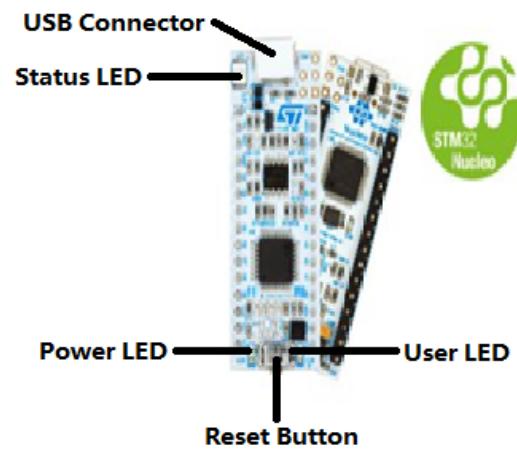
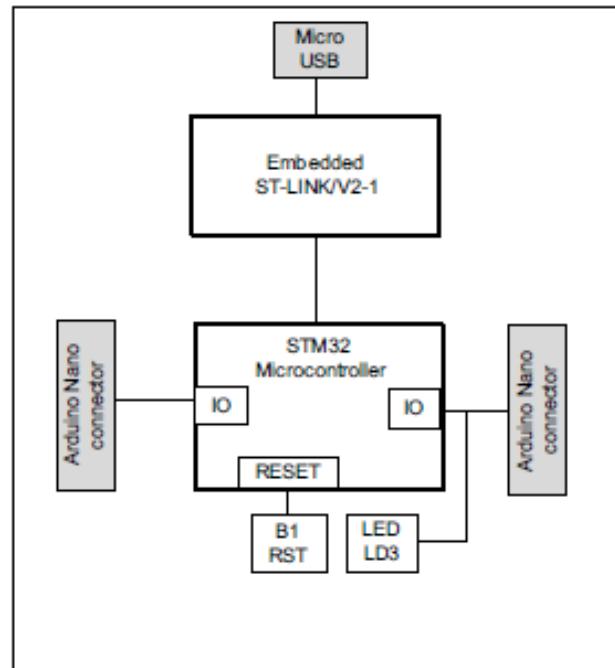
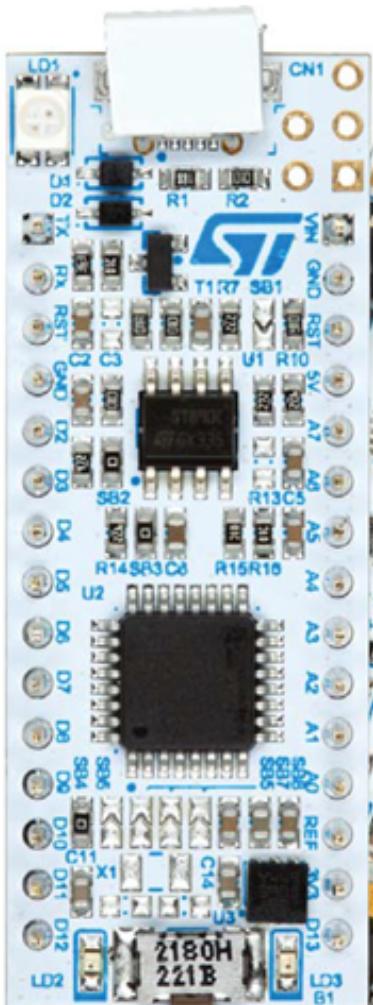


mbed STM32 Nucleo-32 board

- The development board is based on the Nucleo L432KC microcontroller.
- 32-bit ARM Cortex-M4 core running at 80 MHz.
- ARM processors are used in many contemporary products including iphones.



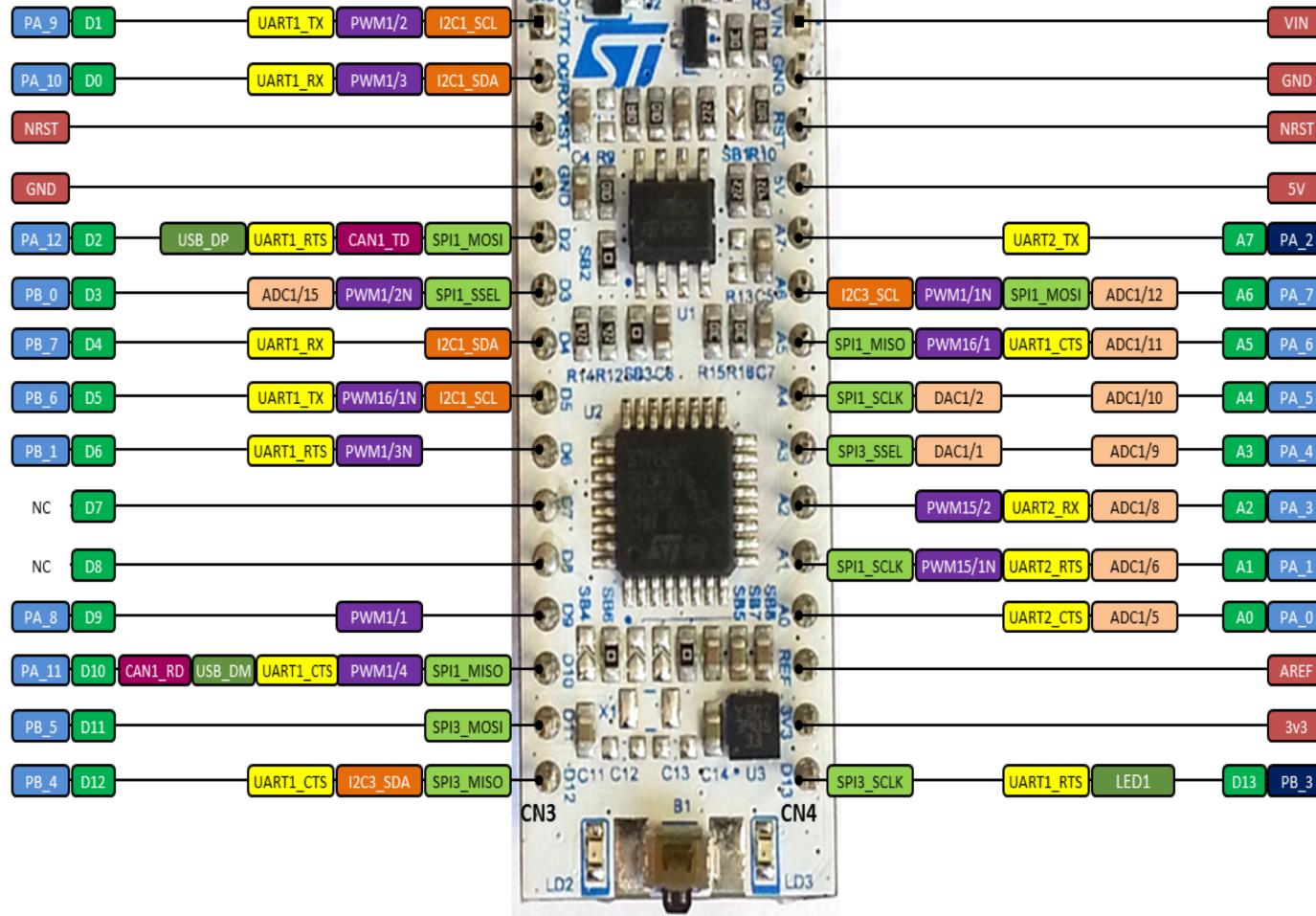
Top-view of the mbed



Pinout Diagram of mbed



life.augmented
NUCLEO-L432KC



Labels usable in code

PX_Y MCU pin without conflict

PX_Y MCU pin connected to other components

See [PeripheralPins.c](#) (link below) for more information

XXX Arduino connector names (A0, D1, ...)

XXX LEDs and Buttons (LED_1, USER_BUTTON, ...)

Labels not usable in code (for information only)

XXX Serial pins (USART/UART)

XXX SPI pins

XXX I2C pins

XXX PWMOut pins (TIMER n/c[N])

n = Timer number c = Channel

N = Inverted channel

XXX AnalogIn (ADC) and AnalogOut pins (DAC)

XXX CAN pins

XXX Power and control pins (3V3, GND, RESET, ...)

Benefits

- Get started right away, with nothing to install
- Get working fast, using high-level APIs
- Explore, test, and demonstrate ideas more effectively
- Write clean, compact code that's easy to modify
- Log in from anywhere, on Windows, Mac or Linux

Features

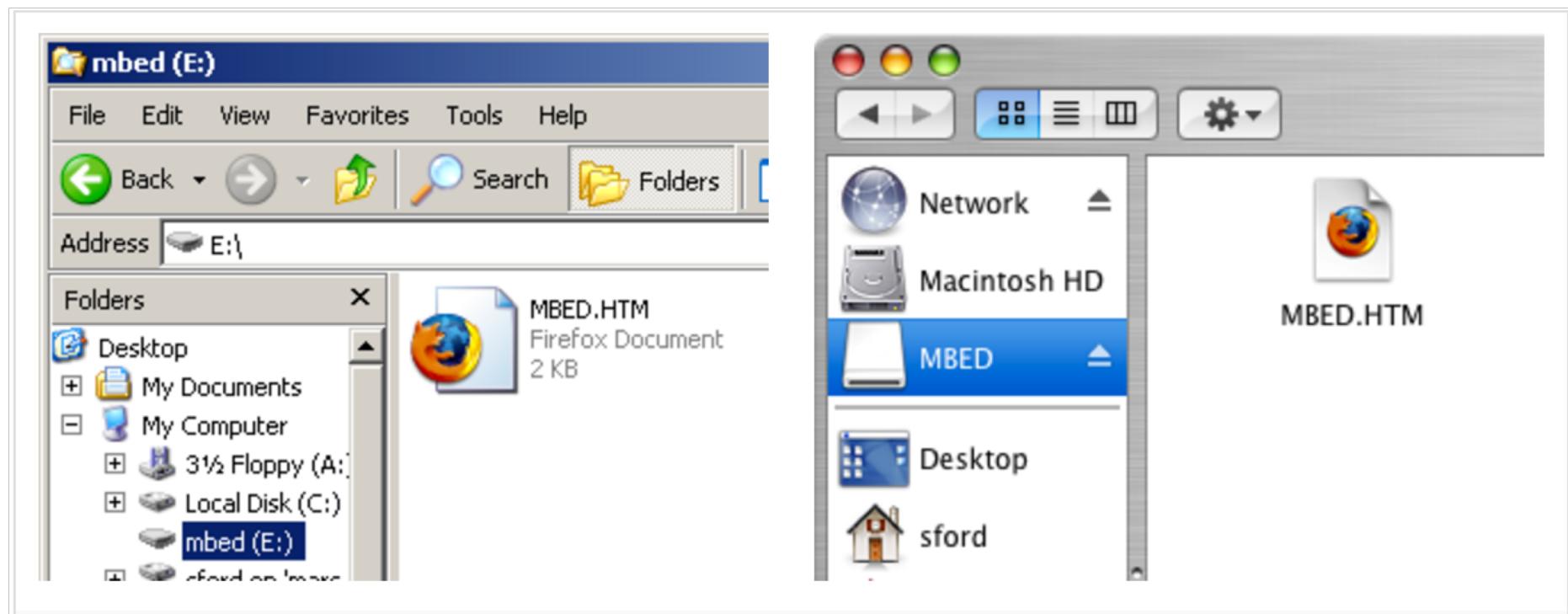
- STM32 microcontrollers in 32-pin packages
- Three LEDs: USB communication (LD1), power LED (LD2) and user LED (LD3)
- Reset push-button
- Board expansion connector:
 - Arduino™ Nano
- Flexible power-supply options: ST-LINK USB V_{BUS} or external sources
- On-board ST-LINK/V2-1 debugger/programmer with USB re-enumeration capability: mass storage, Virtual COM port and debug port
- Support of a wide choice of Integrated Development Environments (IDEs) including IAR™, Keil®, GCC-based IDEs, Arm® Mbed™
- Arm® Mbed Enabled™ compliant (only for some Nucleo part numbers)

Hassle-free startup

- Getting started is as simple as using a USB Flash drive
- Simply connect the mbed stm32l432kc board to a Windows, Mac or Linux computer and it will appear as a USB drive
- Follow the link on the board to connect to the mbed website, where you can sign up and begin designing
- There are no drivers to install or setup programs to run
- It's so easy, in fact, that you can have a "Hello World!" program running in as little as five minutes.

Connect your mbed to a PC

- Use the USB lead to connect your mbed to a PC
- The status light will come on, indicating it has power
- After a few seconds of activity, the PC will recognise the mbed Microcontroller as a standard USB drive



mbed Account

- Go to the new USB Drive, and click **MBED.HTM** to open it in a web browser
- If you do not have an mbed account, choose "Signup", and create your mbed Account
- Once you have created an account, log in with your username and password
- This will give you access to the website, tools, libraries and documentation

mbed Account

The screenshot shows the ARMmbed website's login and signup interface. At the top, there are navigation links for Hardware, Documentation, Code, Questions, Forum, Log In/Signup, and Compiler. A search bar is also present. Below the header, a large teal bar contains the text "Login or Signup". To the left is a "Login" form with fields for Username and Password, and links for "I've forgotten my username" and "I've forgotten my password". There is also a "Remember me" checkbox and a "Login" button. To the right is a "Signup" form featuring a large blue "mbed" logo.

The screenshot shows the ARMmbed website's signup process. The header and navigation bar are identical to the previous screenshot. A teal bar at the top says "Create a developer.mbed.org account". Below it, a message reads "Let's get started! Just one question first, though." It asks if the user has ever signed up on developer.mbed.org before, with "Yes, I have created an account before" and "No, I haven't created an account before" buttons. The main form is titled "Signup" and includes fields for "Enter your email address", "Choose a username", "Choose a password", and "Confirm your password". To the right, a "Summary" section provides information about account creation, stating: "You are about to... Create an mbed user account. An account will be set up for you, giving you access to the mbed website and resources."

mbed Account

Complete the form below to sign up to mbed!

Signup

Enter your email address:

I already have an account!

Choose a username:

Choose a password:

Confirm your password:

First name:

Last name:

Country:

 Select a country...

I agree to ARM's [Terms and Conditions of Use](#). (required)

ARM will process your information in accordance with the Account Registration

Summary

You are about to...

 [Create an mbed user account](#)

An account will be set up for you, giving you access to the mbed website and resources.

Confirm your password:

First name:

Last name:

Country:

 Select a country...

I agree to ARM's [Terms and Conditions of Use](#). (required)

ARM will process your information in accordance with the Account Registration section of our [Privacy Policy](#).

By ticking this box you indicate your consent to receiving marketing communications from ARM in accordance with our Privacy Policy. Please visit our [Subscription Center](#) to manage your marketing preferences or unsubscribe from further communications.

[Signup](#)

mbed – Getting Started

https://os.mbed.com/platforms/ST-Nucleo-L432KC/

arm
MBED Overview ▾ Hardware ▾ Docs ▾ Code ▾ Support ▾ Built with Mbed ▾

Boards » NUCLEO-L432KC

NUCLEO-L432KC

Affordable and flexible platform to ease prototyping using a STM32L432KCU6 microcontroller.



Overview

The STM32 Nucleo board provides an affordable and flexible way for users to try out new ideas and build prototypes with any STM32 microcontroller line, choosing from the various combinations of performance, power consumption and features.

The Arduino™ connectivity support and ST Morpho headers make it easy to expand the functionality of the STM32 Nucleo open development platform with a wide choice of specialized shields.

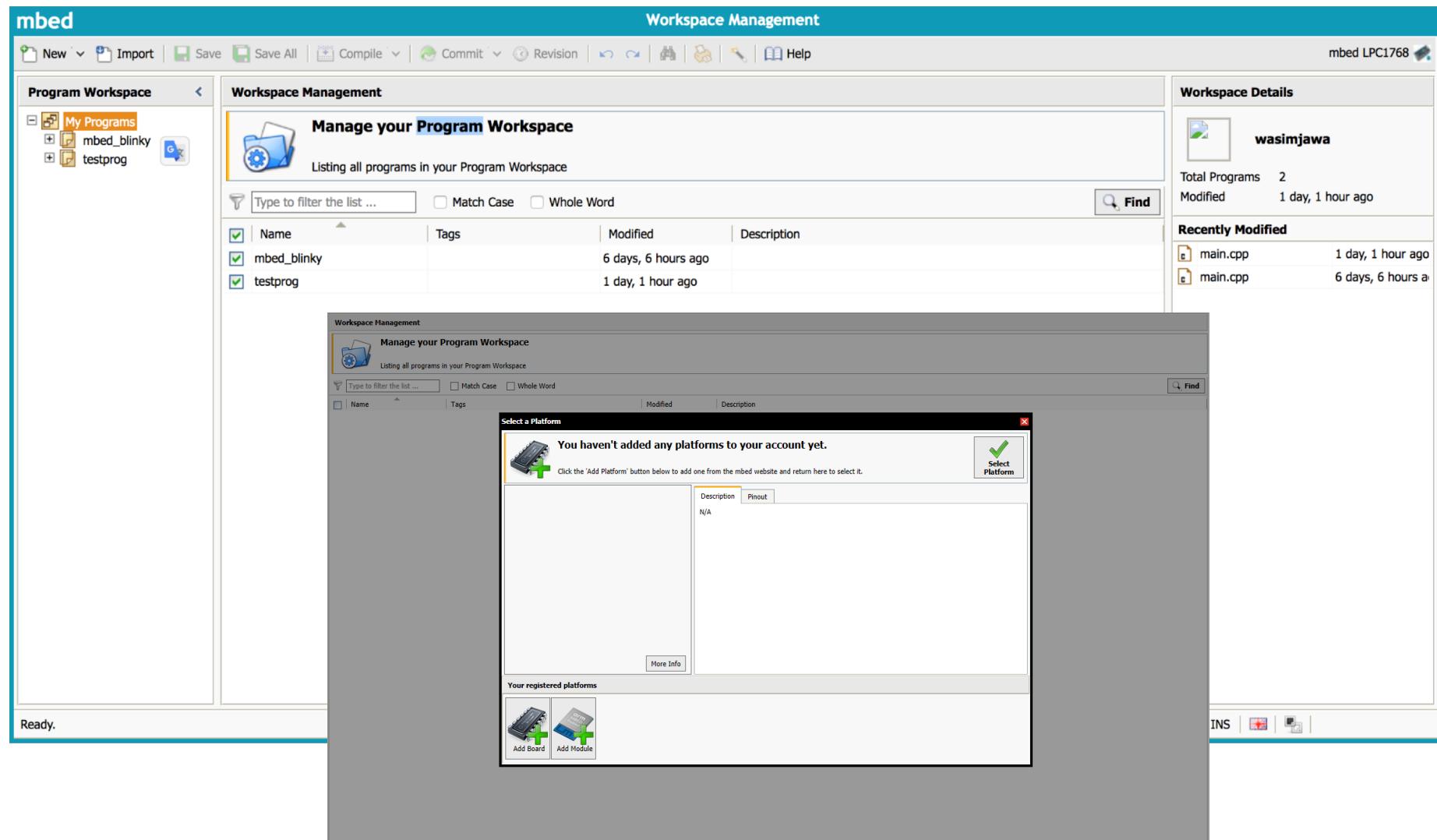
Table of Contents

1. Overview
2. Microcontroller features
3. Nucleo features
4. Board pinout
5. Supported Shields
6. Technical references
7. Known limitations
8. Tips and Tricks

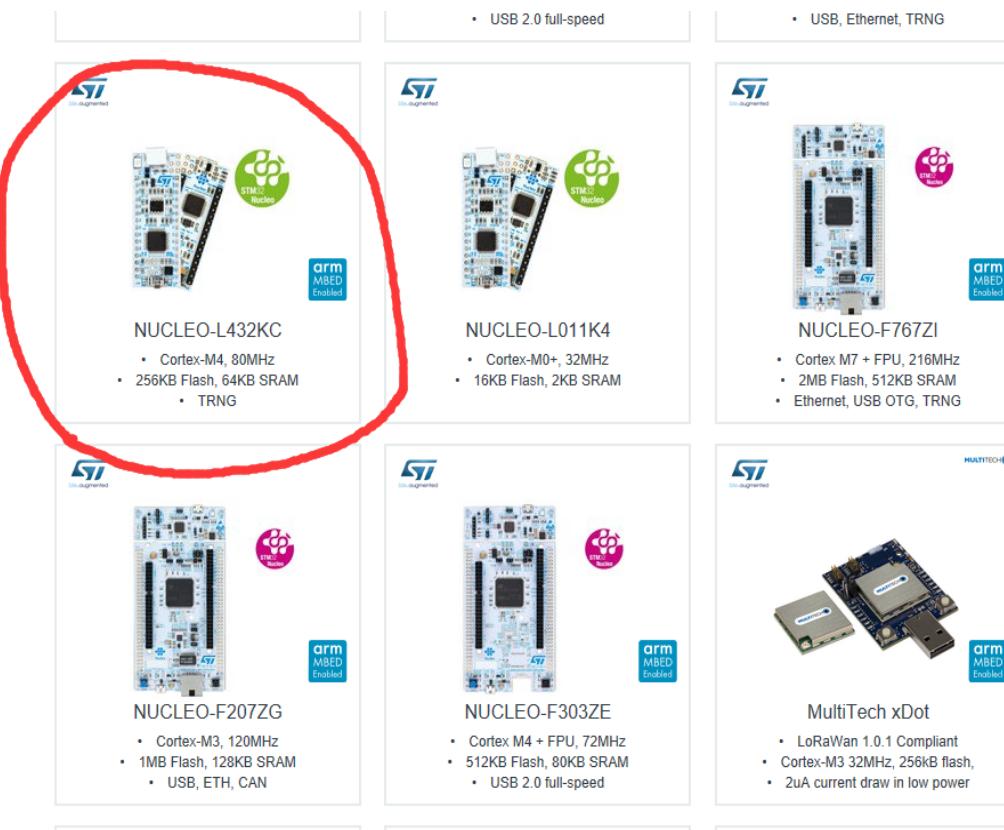
mbed Online Compiler

- The mbed Compiler lets you write programs in C++ and then compile and download them to run on the mbed Nucleo L432KC microcontroller.
- There's no need to run an install or setup program, since the compiler runs online
- Supported browsers include Internet Explorer, Firefox, Safari, or Chrome running on a Windows, Mac, or Linux PC
- You can log in from anywhere and simply pick up where you left off. And, since you're working with a web-based tool, you can be confident that it's already configured and will stay up-to-date

mbed Online Compiler



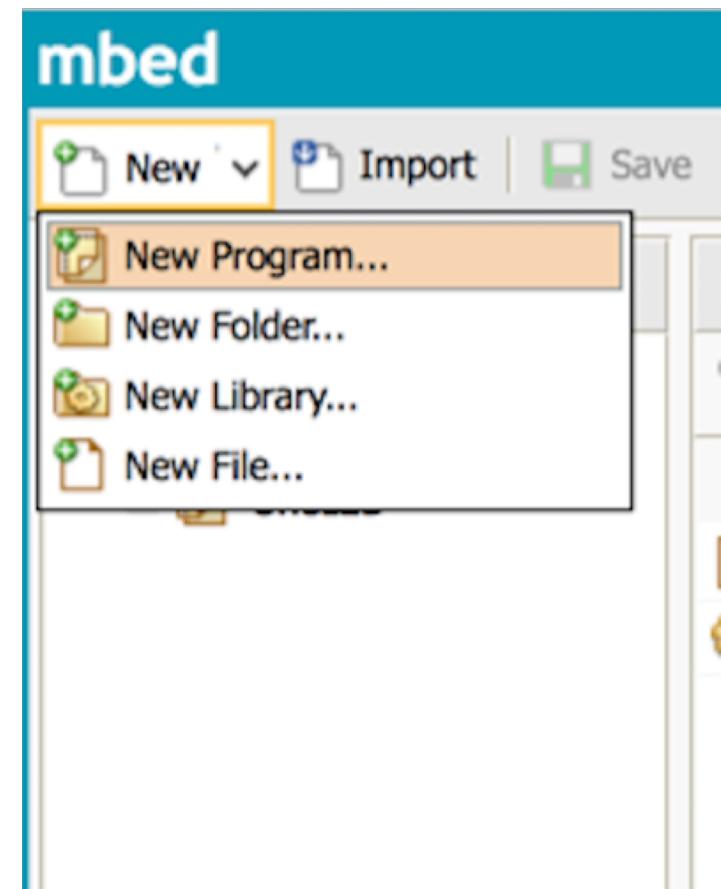
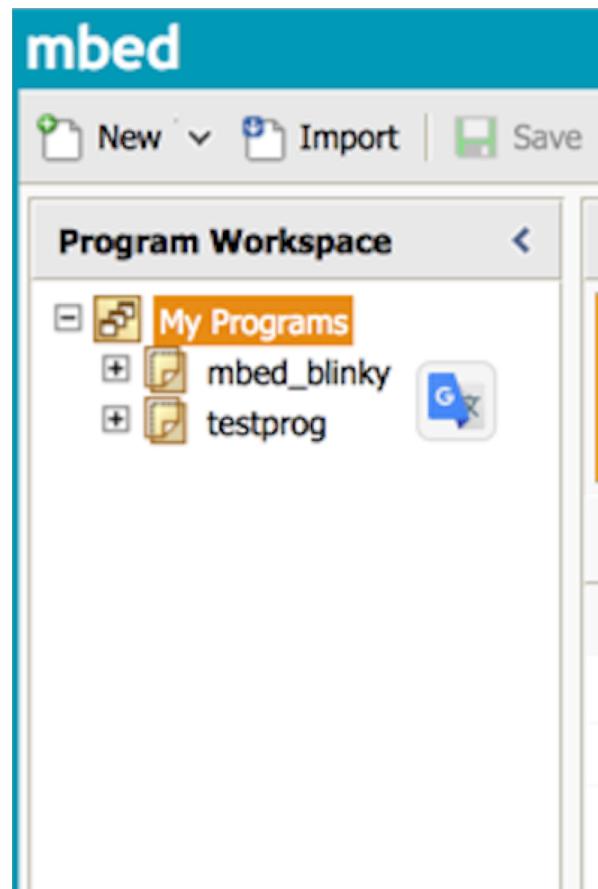
NUCLEO-L432KC as the platform



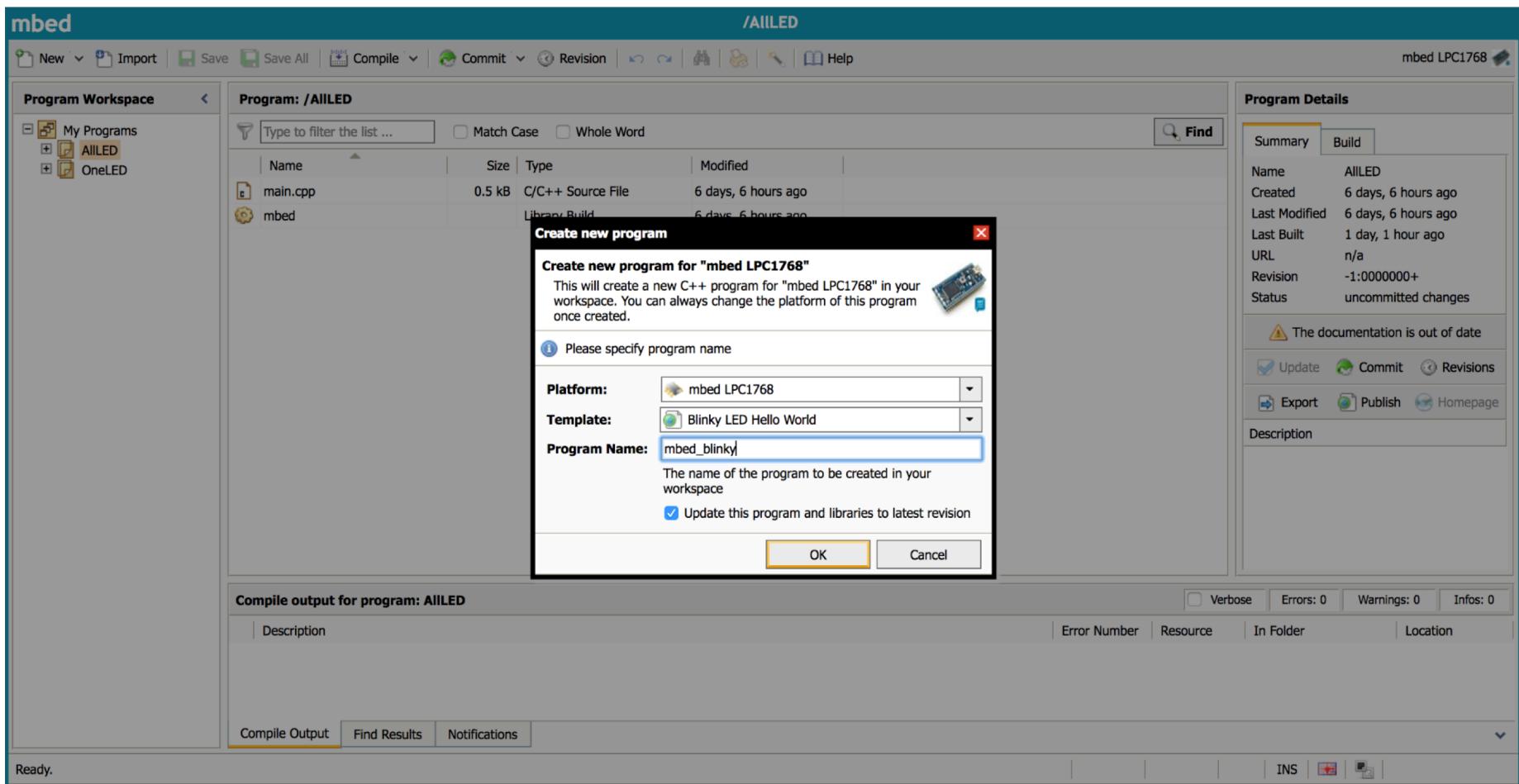
Create a New Program

- Click **New** to create a new program, and enter a filename
- Your newly created program will appear in the Program Workspace on the left of the screen
- If you expand the program by clicking on the +, you will see something called **main.cpp** - double click on this
- It will open your program in the compiler environment
- As you can see, any new program already has some lines of code

Create a New Program



Platform and Program Name



First Program - mbed_blinky

The screenshot shows the mbed IDE interface. The title bar reads "mbed" and the current file is "/mbed_blinky/main.cpp". The menu bar includes New, Import, Save, Save All, Compile, Commit, Revision, and Help. The toolbar has icons for New, Import, Save, Save All, Compile, Commit, Revision, Find, Replace, and Help.

The left sidebar is titled "Program Workspace" and lists "My Programs": AILED, mbed_blinky (selected), mbed, OneLED. Inside mbed_blinky, there are sub-folders: main.cpp (selected), mbed, Classes, Structs, Groups, and OneLED. The main editor window displays the code for main.cpp:

```
#include "mbed.h"
DigitalOut myled(LED1);
int main() {
    while(1) {
        myled = 1;
        wait(0.2);
        myled = 0;
        wait(0.2);
    }
}
```

The bottom pane is titled "Compile output for program: mbed_blinky" and contains tabs for "Description", "Compile Output" (selected), "Find Results", and "Notifications". The status bar at the bottom says "Ready."

Download program to mbed

- To get the program onto the mbed, you first have to compile it, by clicking the "**Compile**" button in the toolbar
- This will create the binary code, which will ultimately be downloaded to the mbed
- If the compiling goes well, you will get a "**Success!**" message in the compiler output, and a popup will prompt you to download the compiled **.bin** file to the mbed
- Whilst the program is downloading, the status LED on the mbed will flash
- Once this has stopped, press the **Reset Button** on the mbed to start your program running
- You should see **LED1** flashing on and off every 0.2 seconds

mbed_blinky Program

<code>#include "mbed.h"</code>	This is a header that needs to be at the start of each mbed.
<code>DigitalOut myled(LED1);</code>	<p>Essentially, what you are doing here is using the DigitalOut component to set LED1 as a digital output, and to give that "port" a name, myled. You can do something similar for each of pins 5-30, using the format: DigitalOut variablename (pinnumber)</p> <p>For example DigitalOut greenled (p5) would set pin 5 as a digital output, with the name greenled.</p>
<code>int main() {</code>	The action of any C++ program is contained within its main() function. The function definition, what goes on inside the function, is contained within the curly brackets, starting immediately after main(), and continuing until the last closing curly bracket.
<code>while(1) {</code>	Many embedded systems programs contain endless infinite loop, a section of the program that just goes on repeating forever. In other branches of programming, this is bad practice, but for embedded systems, this is quite standard. The endless loop is created using the while keyword; this controls the code within the curly brackets, which follow. Normally, while is used to set up a loop, which repeats if a certain condition is satisfied, but if we write while(1), this will make the loop repeat endlessly.

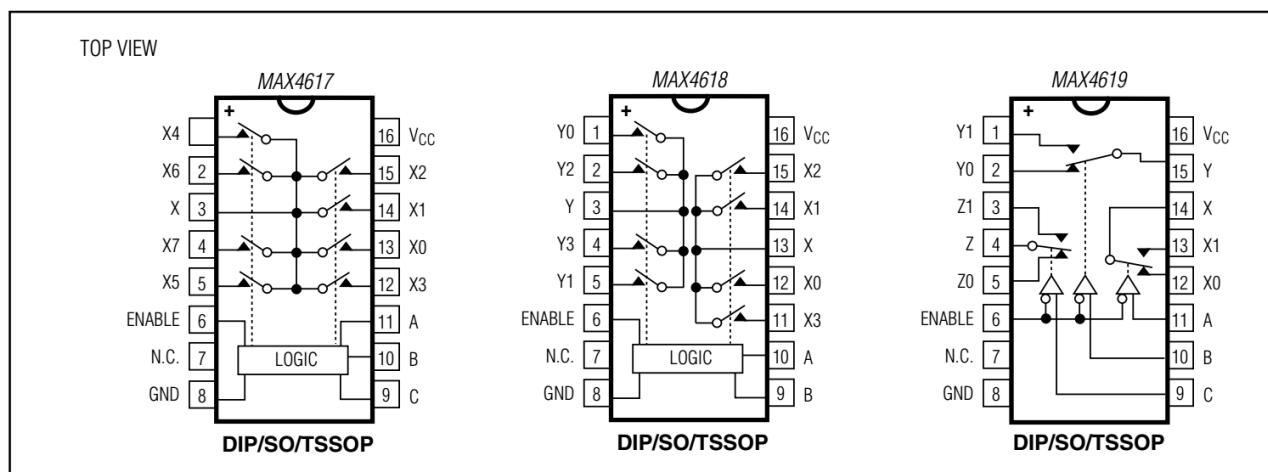
mbed_blinky Program

	means that the variable myled is set to the value 1, no matter what its previous value was.
myled=1;	In C++, this is different to "equals", which is represented by "=". "=" has the meaning "is set to". This sets a logic 1, or voltage of ~3.3V on the pin to which LED1 is connected. This will cause the LED to light up.
wait(0.2);	function is from the mbed library - you can find out more about it on the mbed website. The 0.2 parameter is in seconds, and defines the delay length caused by this function. So, the LED will be lit for 0.2s
myled=0;	means that the variable myled is set to the value 0, whatever its previous value was. This sets a logic 0, or voltage of 0 V on the pin to which LED1 is connected. This will cause the LED to switch off. Note that this will not always turn an LED off; it depends on how the LED is wired in the circuit.
wait(0.2);	the LED will be switched off for 0.2 s

MAX4617/MAX4618/ MAX4619

The MAX4617/MAX4618/MAX4619 are high-speed, low voltage, CMOS analog ICs configured as an 8-channel multiplexer (MAX4617), two 4-channel multiplexers (MAX4618), and three single-pole/double-throw (SPDT) switches (MAX4619)

Pin Configurations/Functional Diagrams



MAX4617/MAX4618/MAX4619

High-Speed, Low-Voltage, CMOS Analog Multiplexers/Switches

Table 1. Truth Table/Switch Programming

ENABLE INPUT	SELECT INPUTS			ON SWITCHES		
	C*	B	A	MAX4617	MAX4618	MAX4619
H	X	X	X	All switches open	All switches open	All switches open
L	L	L	L	X-X0	X-X0, Y-Y0	X-X0, Y-Y0, Z-Z0
L	L	L	H	X-X1	X-X1, Y-Y1	X-X1, Y-Y0, Z-Z0
L	L	H	L	X-X2	X-X2, Y-Y2	X-X0, Y-Y1, Z-Z0
L	L	H	H	X-X3	X-X3, Y-Y3	X-X1, Y-Y1, Z-Z0
L	H	L	L	X-X4	X-X0, Y-Y0	X-X0, Y-Y0, Z-Z1
L	H	L	H	X-X5	X-X1, Y-Y1	X-X1, Y-Y0, Z-Z1
L	H	H	L	X-X6	X-X2, Y-Y2	X-X0, Y-Y1, Z-Z1
L	H	H	H	X-X7	X-X3, Y-Y3	X-X1, Y-Y1, Z-Z1

X = Don't care

*C not present on MAX4618.

Note: Input and output pins are identical and interchangeable. Either may be considered an input or output; signals pass equally well in either direction.

Summary

- Mbed
- Mbed online compiler
- What will we study in next lecture