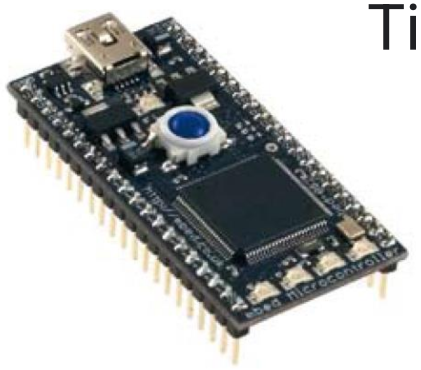
Introducing the Online (MOOC) Courses

Embedded Systems Essentials with Arm:

1. Getting Started

2. Get Practical with Hardware

Tim Wilmshurst. Course co-author



What is a MOOC?

Massive Open Online Courses (MOOCs) are free online courses available for anyone to enrol. MOOCs provide an affordable and flexible way to learn new skills, advance your career and deliver quality educational experiences at scale.

Millions of people around the world use MOOCs to learn for
…. career development, changing careers, college preparation,
….lifelong learning, corporate eLearning ….

https://www.mooc.org/

Embedded Systems Essentials with Arm: Getting Started

https://www.edx.org/course/embedded-systems-essentials-with-arm-getting-started

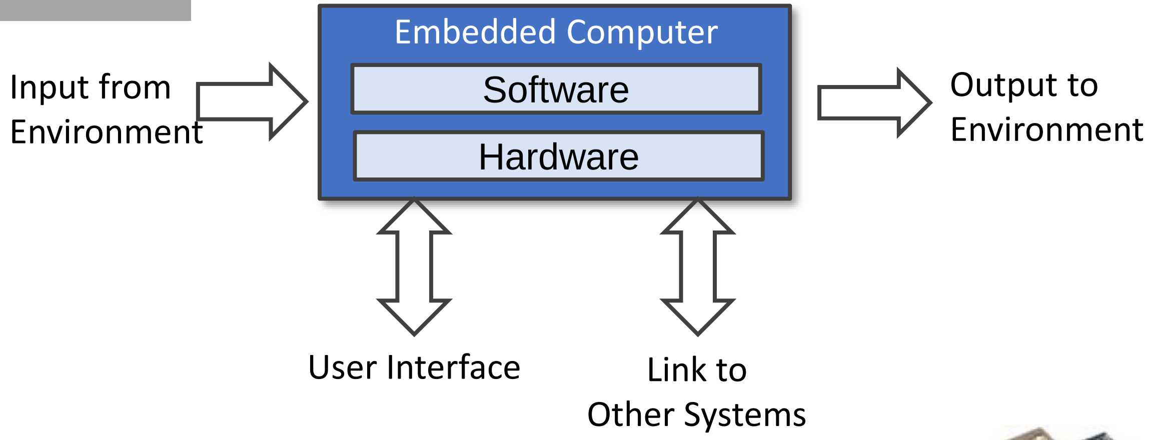
Course Content

1. Introduction to Embedded Systems
2. Introduction to the Mbed Ecosystem
3. Digital Input and Output
4. Interrupts
5. Analog Input and Output
6. Timers and Pulse-Width Modulation

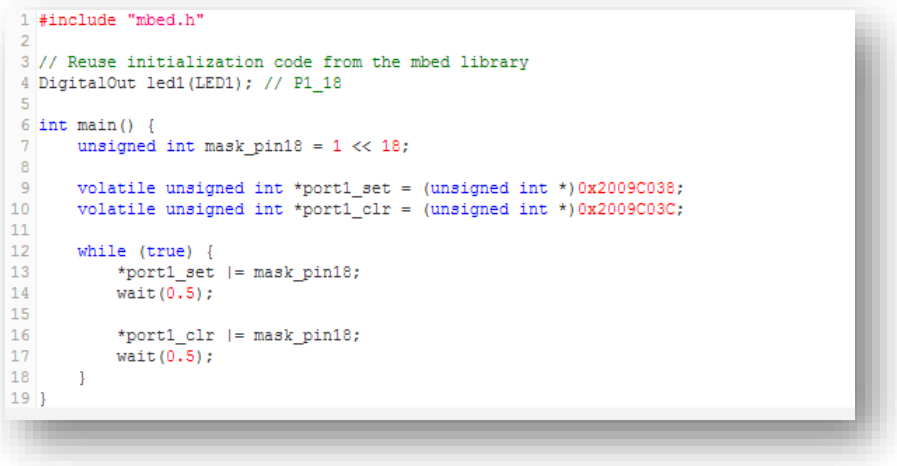
Course Content

1. Introduction to Embedded Systems
2. Introduction to the Mbed Ecosystem
3. Digital Input and Output
4. Interrupts
5. Analog Input and Output
6. Timers and Pulse-Width Modulation

Course Content



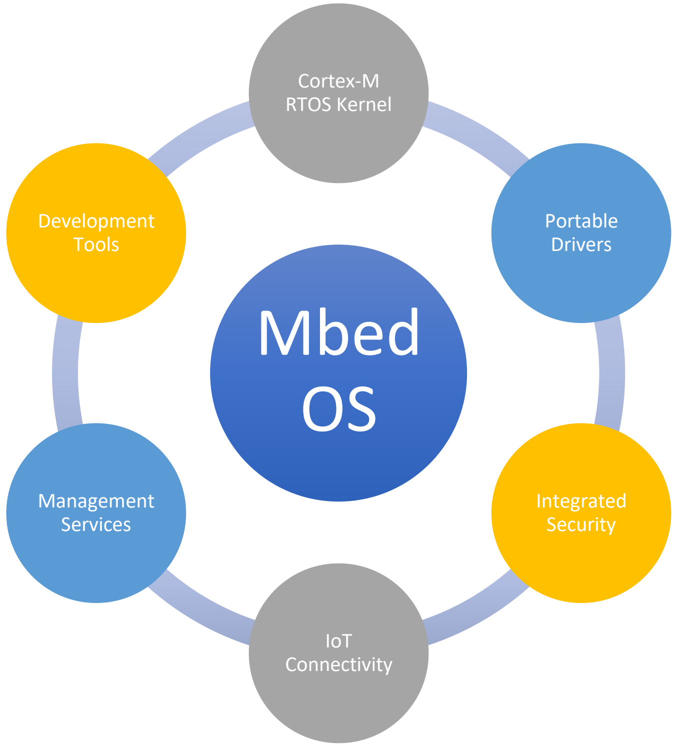
1. Introduction to Embedded Systems
2. Introduction to the Mbed Ecosystem
3. Digital Input and Output
4. Interrupts
5. Analog Input and Output
6. Timers and Pulse-Width Modulation



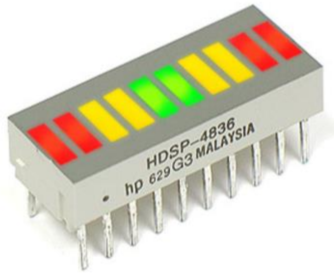
```
#include "mbed.h"

// Reuse initialization code from the mbed library
DigitalOut led1(LED1); // P1_18

int main() {
    unsigned int mask_pin18 = 1 << 18;

    volatile unsigned int *port1_set = (unsigned int *)0x2009C038;
    volatile unsigned int *port1_clr = (unsigned int *)0x2009C03C;

    while (true) {
        *port1_set |= mask_pin18;
        wait(0.5);

        *port1_clr |= mask_pin18;
        wait(0.5);
    }
}
```

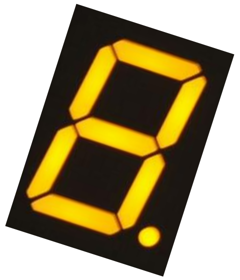
Course Content

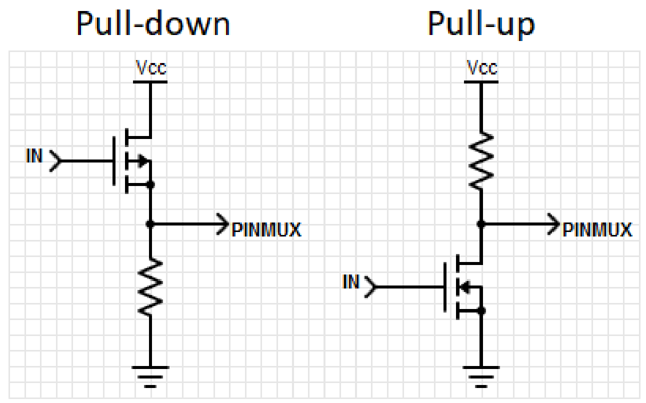
1. Introduction to Embedded Systems
2. Introduction to the Mbed Ecosystem
3. Digital Input and Output
4. Interrupts
5. Analog Input and Output
6. Timers and Pulse-Width Modulation



Course Content

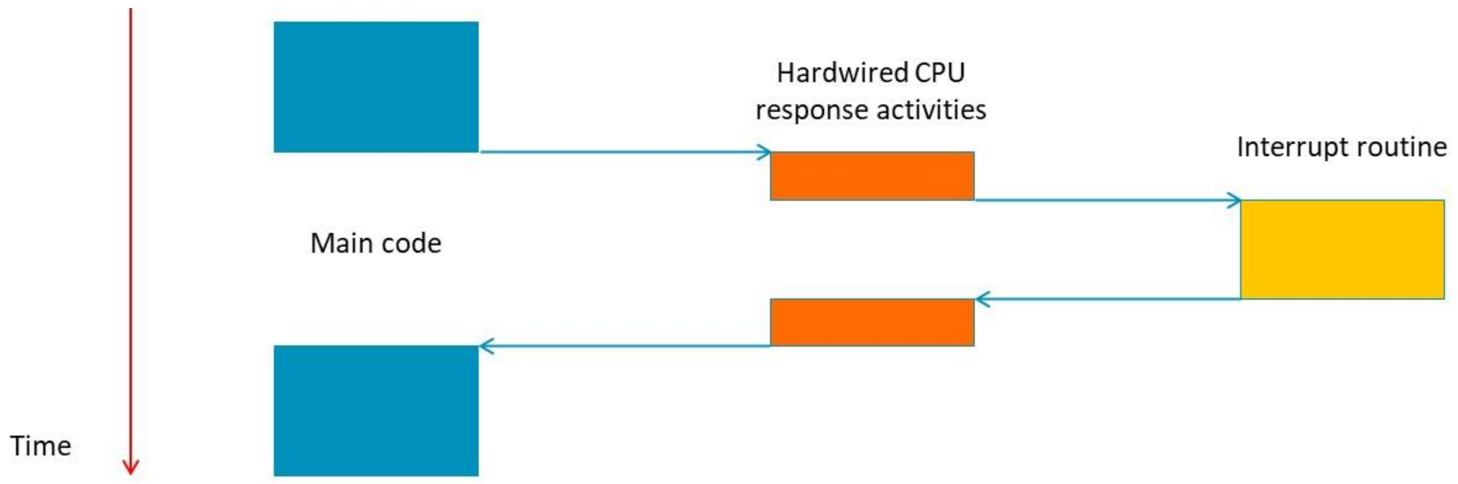






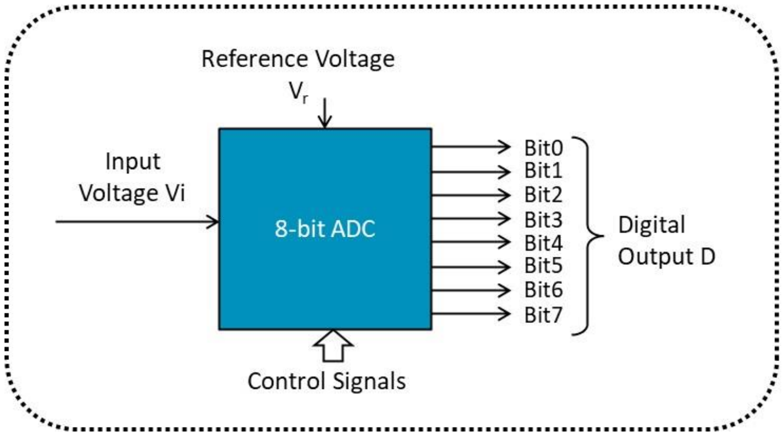
Course Content

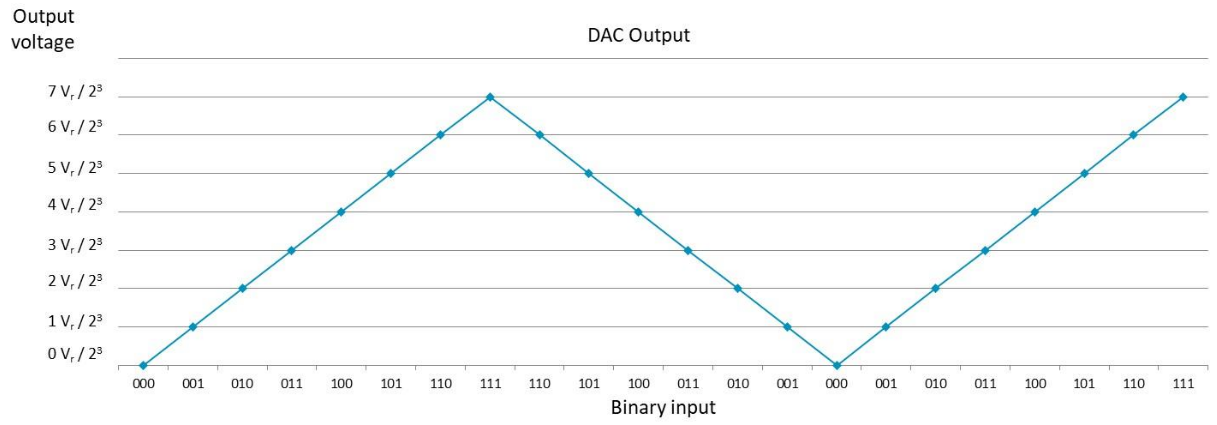
1. Introduction to Embedded Systems
2. Introduction to the Mbed Ecosystem
3. Digital Input and Output
4. Interrupts
5. Analog Input and Output
6. Timers and Pulse-Width Modulation



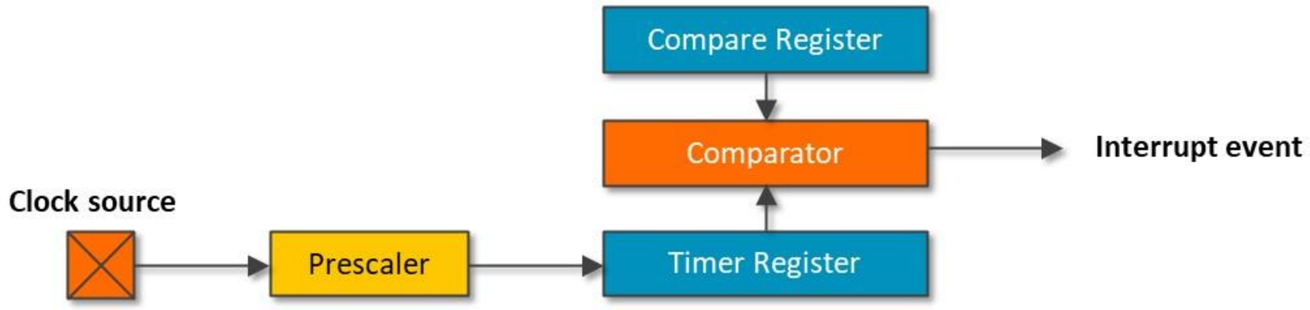
Course Content

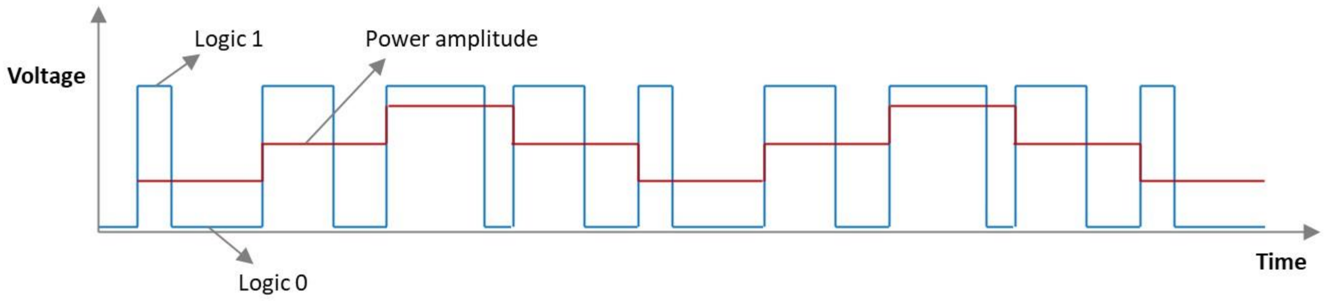
1. Introduction to Embedded Systems
2. Introduction to the Mbed Ecosystem
3. Digital Input and Output
4. Interrupts
5. Analog Input and Output
6. Timers and Pulse-Width Modulation





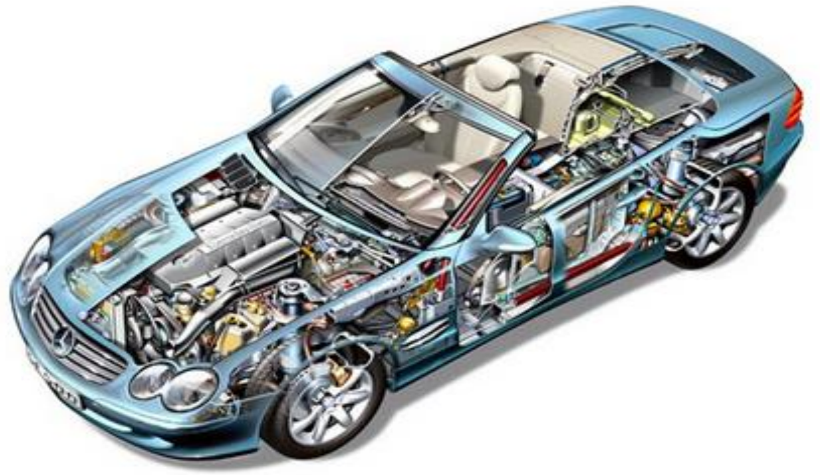
Course Content





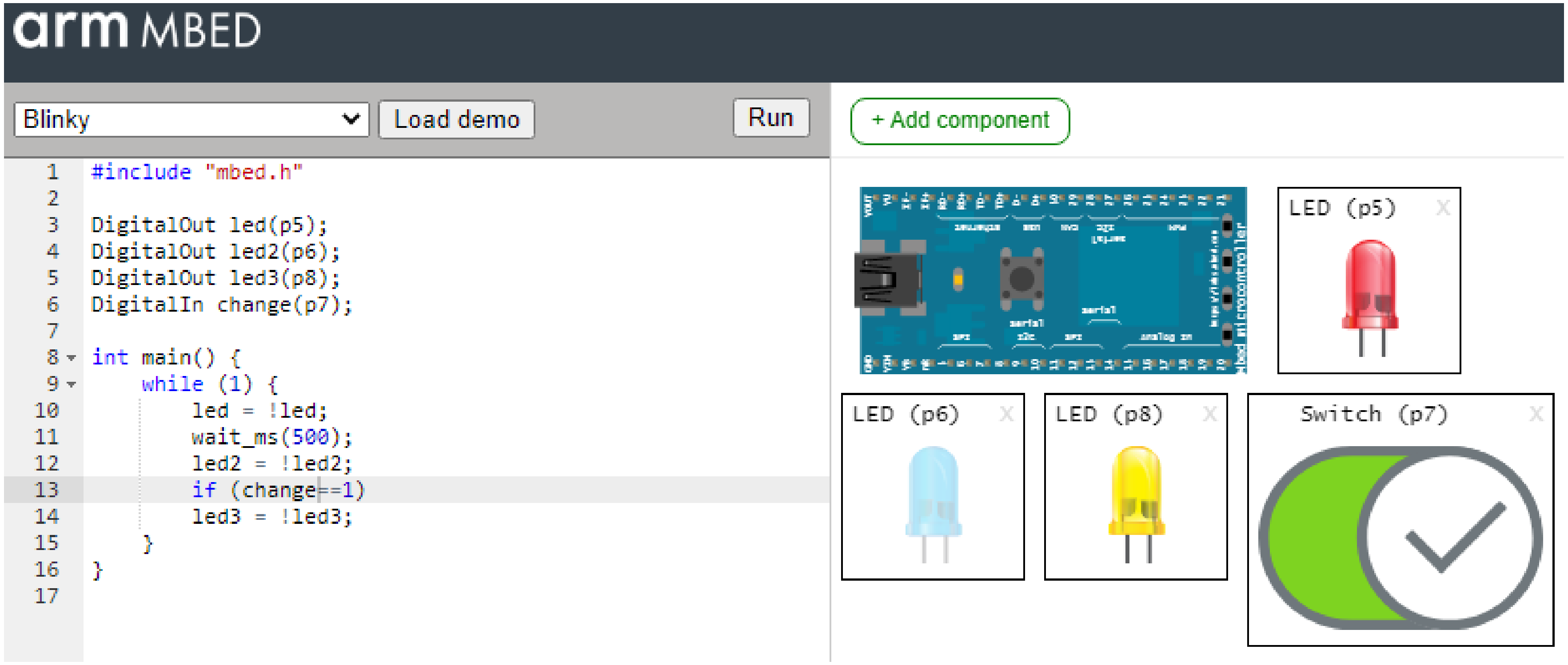
Course Delivery

- Opening “thinking point” which sets the scene;
- Video clips introduce important concepts;
- Quiz questions test your own knowledge and progress;
- Code and circuit examples;
- Practical work with the Mbed simulator;
- End-of-module assessment questions.



The mbed Simulator

- Write simple programs, then immediately compile them and see them run;
- Add simple components, like LEDs (light emitting diodes) and switches;
- Build your confidence in C/C++ programming;
- Some limitations, and not the real thing, but a useful stepping stone!



```
#include "mbed.h"

DigitalOut led(p5);
DigitalOut led2(p6);
DigitalOut led3(p8);
DigitalIn change(p7);

int main() {
    while (1) {
        led = !led;
        wait_ms(500);
        led2 = !led2;
        if (change==1)
        led3 = !led3;
    }
}
```

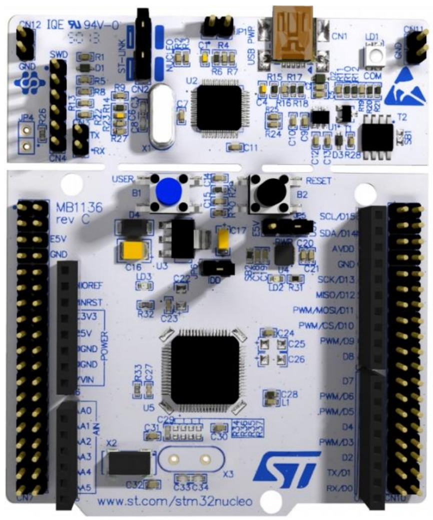
Conclusion

At the end of the ***Getting Started*** course you will have learned the basics of embedded systems. This will:

- Enhance your introductory or intermediate studies, if you’re at university or college;
- Give you entry level expertise, for a trainee job in embedded systems or Internet of Things;
- Give a foundation for further study, for example on the “Get Practical with Hardware” course.

Embedded Systems Essentials with Arm: Get Practical with Hardware

https://www.edx.org/course/embedded-systems-essentials-with-arm-get-practical-with-hardware



Course Content

Module 1: Introducing Serial Communication

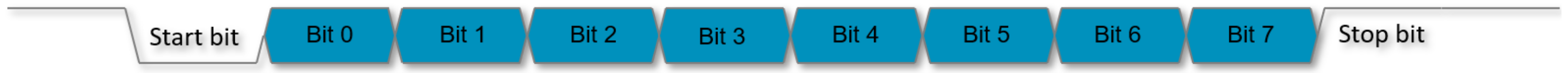
Module 2: Further Serial Communication

Module 3: Introducing Real-Time Operating Systems

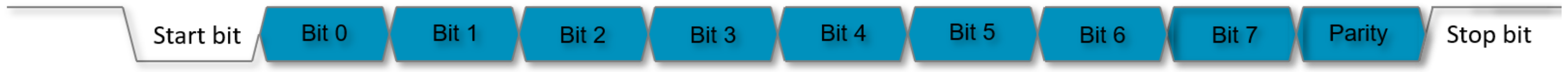
Module 4: Further Real-Time Operating Systems

Module 1: Introducing Serial Communication

- What is serial data communication?
- Serial Peripheral Interface (SPI);
- Asynchronous serial, with the Universal Asynchronous Receiver/Transmitter (UART);
- Making serial connections, including to intelligent instrumentation;
- Limitations of these simple protocols;
- A lab, including a real physical build applying these features.



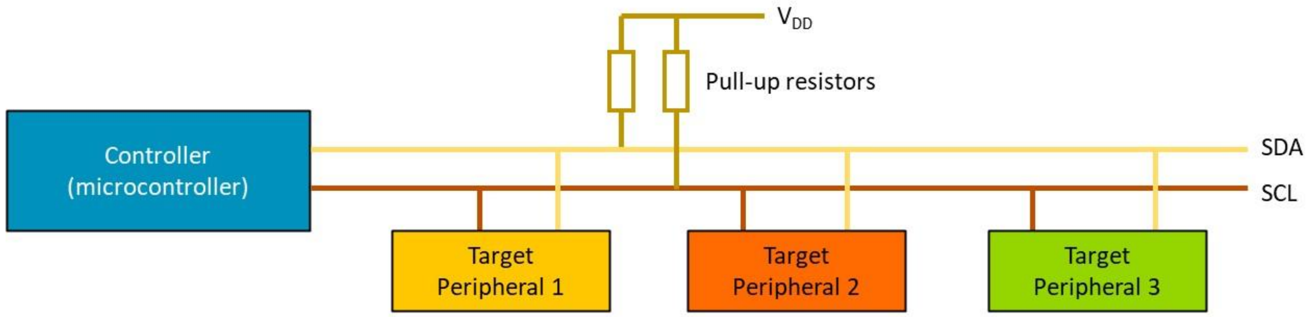
Transfer one byte without parity bit

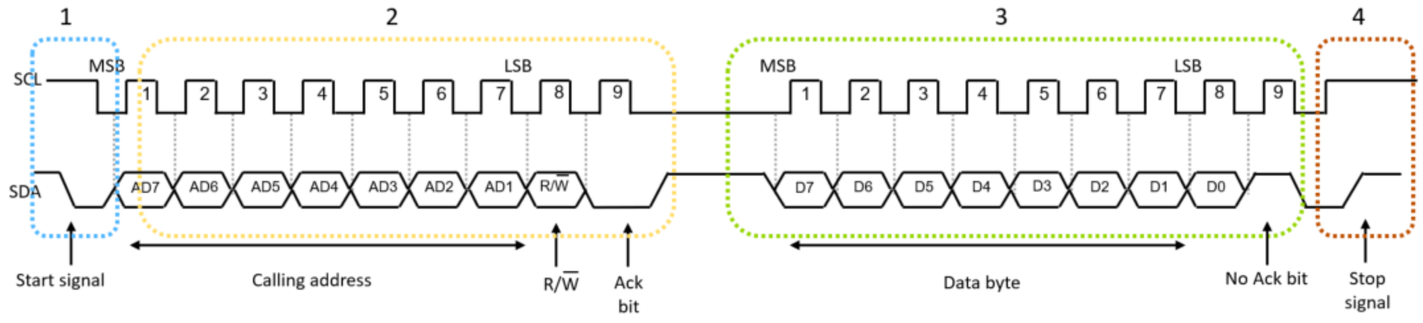


Transfer one byte with parity bit

Module 2: Further Serial Communication

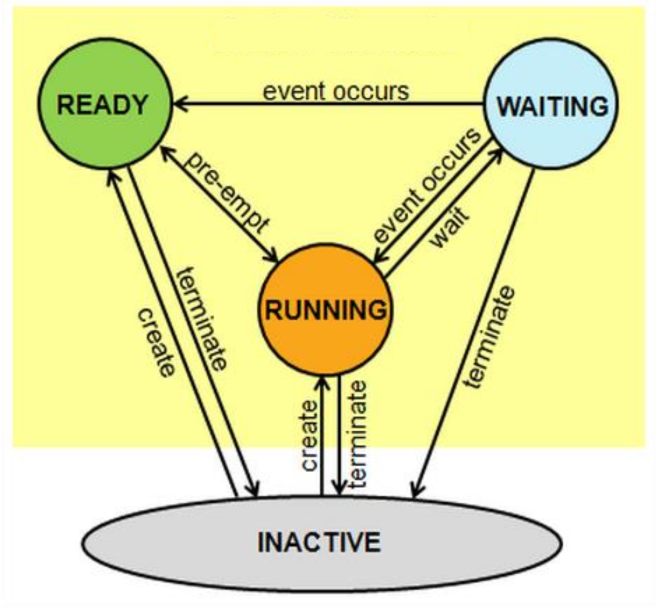
- Understand the need for more advanced serial protocols;
- The Inter Integrated Circuit (I^2C) serial protocol;
- The Universal Serial Bus;
- Making informed design choices with serial protocols;
- A lab, including a real physical build applying these features.

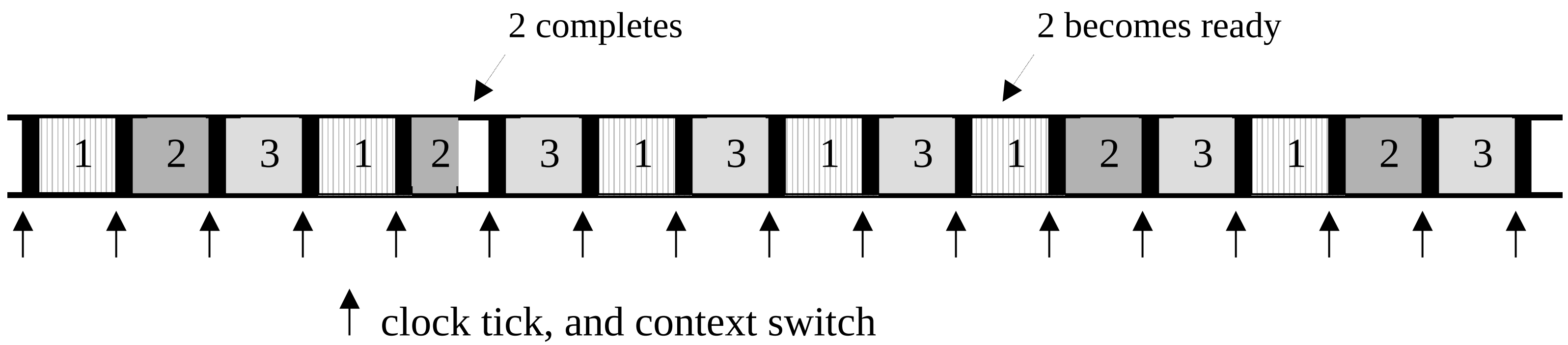




Module 3: Introducing Real-Time Operating Systems

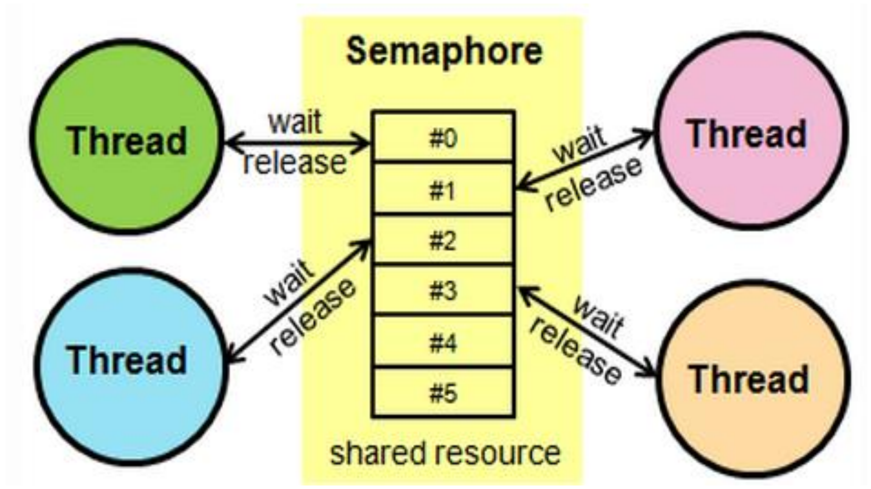
- What is *Real Time?*
- Operating Systems, what they are and why we need them;
- Thinking in terms of tasks, threads and time, in program design;
- Basic scheduling strategies;
- Introducing the Mbed RTOS;
- A lab, including a real physical build applying these features.

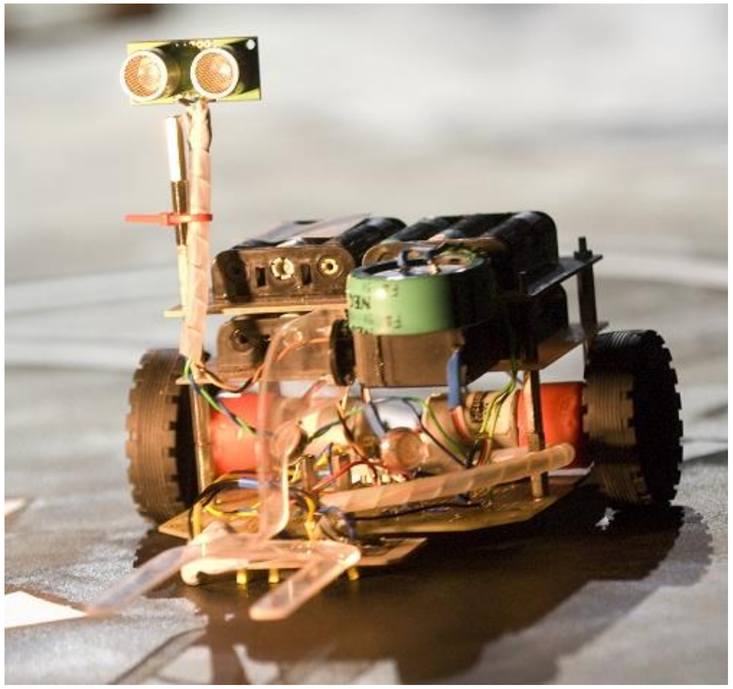




Module 4: Further Real-Time Operating Systems

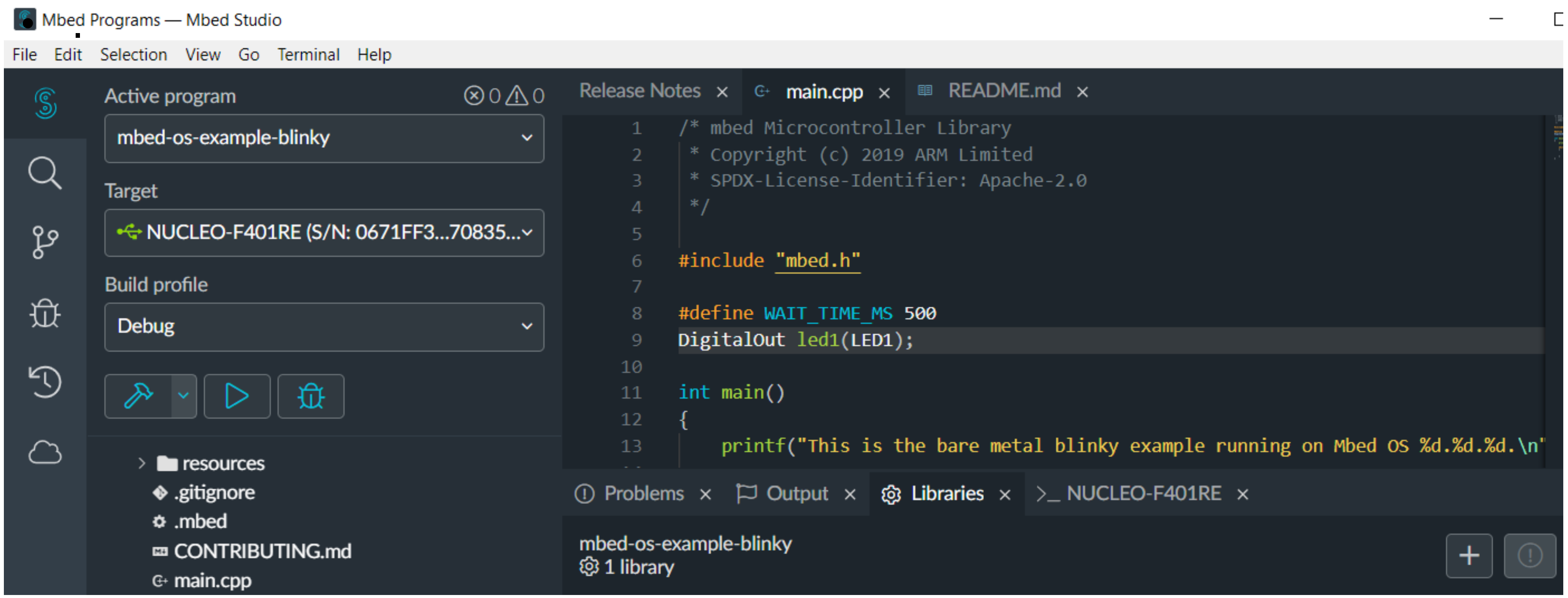
- Further RTOS features, including semaphore and mutex;
- Using interrupts with an RTOS;
- Example evaluation of a RTOS application;
- A final mini-project, drawing together all the skills and knowledge developed in this course.





Practical Work 1

- Comprehensive written guidance given, with circuit diagrams, build images, videos and troubleshooting tips;
- Work with free Mbed Studio IDE (Integrated Development Environment);
- Construct working circuits based on low-cost Nucleo F401 development board;
- Develop your embedded C/C++ programming skills, following clear examples.



Active program: mbed-os-example-blinky

Target: NUCLEO-F401RE (S/N: 0671FF3...70835...

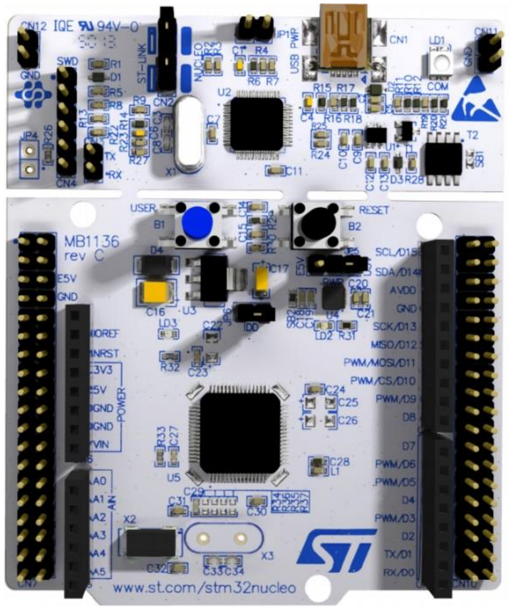
Build profile: Debug

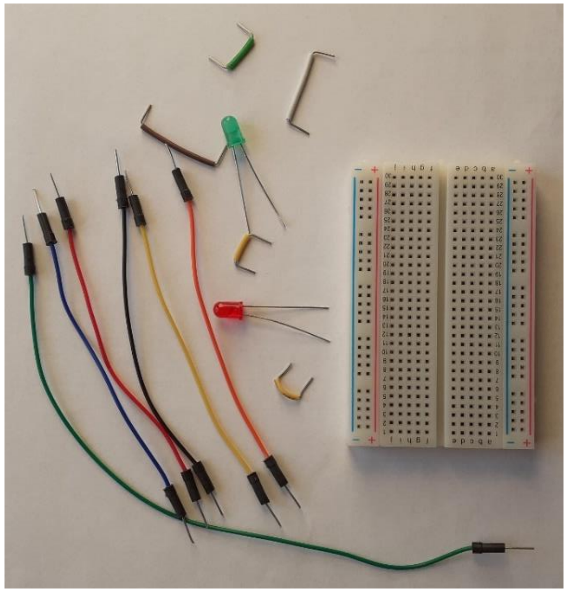
```
/* mbed Microcontroller Library
 * Copyright (c) 2019 ARM Limited
 * SPDX-License-Identifier: Apache-2.0
 */

#include "mbed.h"

#define WAIT_TIME_MS 500
DigitalOut led1(LED1);

int main()
{
    printf("This is the bare metal blinky example running on Mbed OS %d.%d.%d.\n"
```

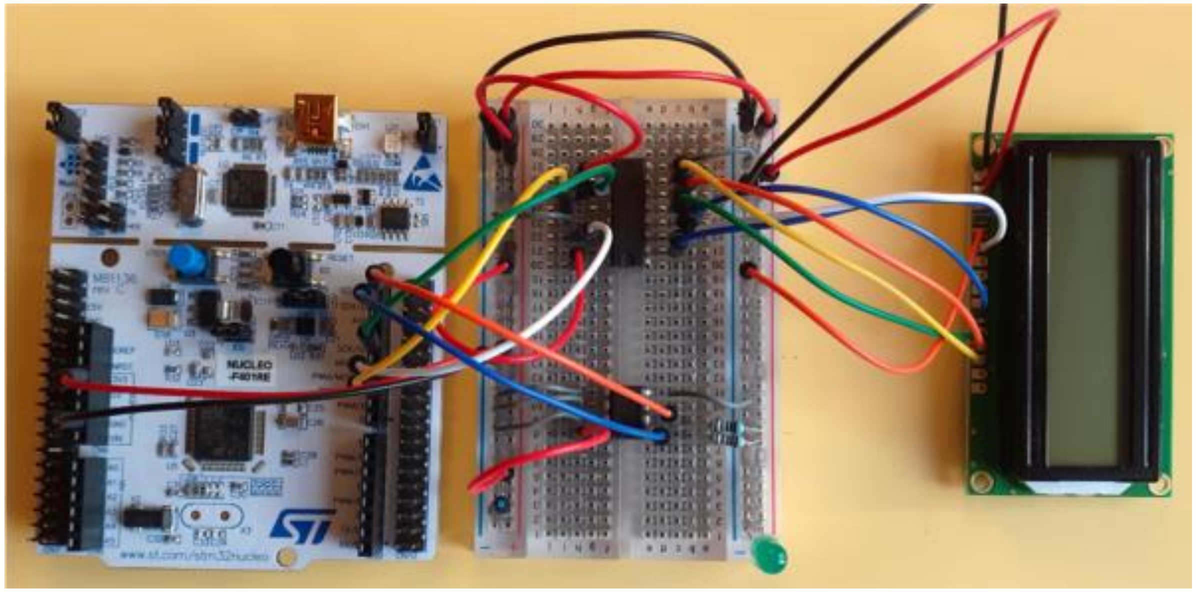
mbed-os-example-blinky
1 library

Practical Work 2

- Circuits are developed step by step using “breadboards” and plug-in components;
- Programs are downloaded to the Nucleo board;
- Following careful debugging, on both software and hardware, you have a working system up and running!
- Then change, adapt and develop it, applying your own creativity!







Conclusion

On successful completion of the ***Get Practical with Hardware*** course you will have developed some significant skills in embedded systems. This will:

- Enhance your intermediate or advanced studies, if at university or college;
- Equip you with skills that will be of significant interest to an employer in this field;
- Give a foundation for ongoing study, for example in the fields of Internet of Things, Robotics, or Machine Learning.