Introduction to Embedded Systems
What is an Embedded System?

Hardware+Software

Performs specific tasks

Sensors

Actuators

Peripherals_
Embedded Systems Typical Architecture

Hardware

SoC, Memories, Peripherals, Sensors, etc

Software

__deviceDrivers

__middleware

application_
Hardware

Microprocessor vs Microcontroller

Peripherals: Timers, Interrupts, Communication buses, ADC, I/O

Actuators: Buttons, Touchscreens, etc

Sensors: Temperature, Light, Humidity, etc.
Software - __deviceDrivers

Very low level

Used to be written in assembly

Interfaces with registers and configurations
Example - Assembly (ARM) vs C

Example:

```assembly
MOV R0,#0
MOV R1,#64
MOV R2,#2

.loop
SUB R1,R1,R2
ADD R0,R0,#1
CMP R2,R1
BHI loop ;if R1>R2
```

C Code:

```c
uint8_t result = 64/2;
```
Software - _middleware_

Uses __deviceDrivers functions to provide system abstraction

Bridge between __deviceDrivers and application._
Software - application

High level functions

Code to perform the required tasks_
Structure of a typical firmware code

```c
#include<nrf52840.h>  //Register definition and addresses of the chip
#include<board.h>     //Hardware pins definition
#include<_middleware.h> //e.g. spi, i2c, timer and other peripherals
int main(void)
{
    for(;;)
    {
        //Here goes your printf(“Hello World”);
    }
}
```
Demo_