ARM Microcontroller Course

May 6, 2015

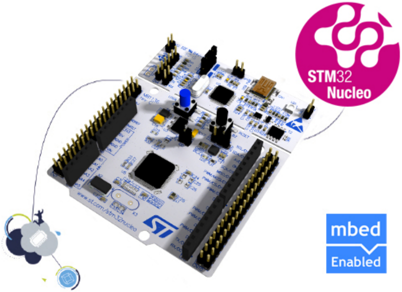
Table of Contents

The Course

- 4 Evenings

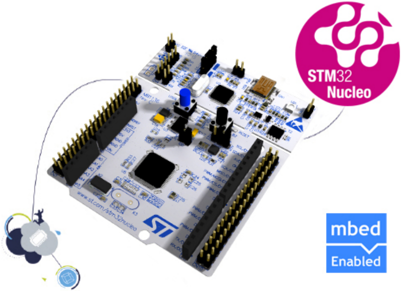
The Course

- 4 Evenings
- Nucleo-F411RE board



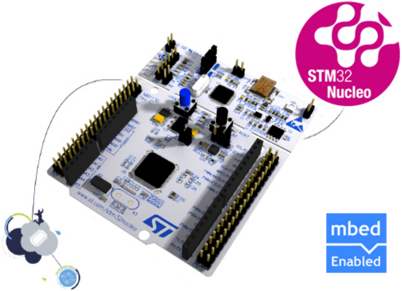
The Course

- 4 Evenings
- Nucleo-F411RE board
- Programming in C



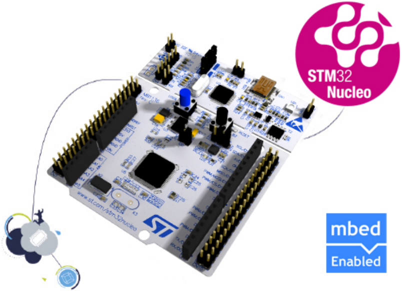
The Course

- 4 Evenings
- Nucleo-F411RE board
- Programming in C
- Manual and Datasheet



The Course

- 4 Evenings
- Nucleo-F411RE board
- Programming in C
- Manual and Datasheet
- Build a Function Generator



What is a Microcontroller?

- Processor

What is a Microcontroller?

- Processor
- Memory

What is a Microcontroller?

- Processor
- Memory
  - Program Memory

What is a Microcontroller?

- Processor
- Memory
  - Program Memory
  - RAM

What is a Microcontroller?

- Processor
- Memory
  - Program Memory
  - RAM
- Peripherals

What is a Microcontroller?

- Processor
- Memory
  - Program Memory
  - RAM
- Peripherals
  - Clock Generator

What is a Microcontroller?

- Processor
- Memory
  - Program Memory
  - RAM
- Peripherals
  - Clock Generator
  - Digital General Purpose I/O

What is a Microcontroller?

- Processor
- Memory
  - Program Memory
  - RAM
- Peripherals
  - Clock Generator
  - Digital General Purpose I/O
  - Analog I/O (eg. ADC, Comparator)

What is a Microcontroller?

- Processor
- Memory
  - Program Memory
  - RAM
- Peripherals
  - Clock Generator
  - Digital General Purpose I/O
  - Analog I/O (eg. ADC, Comparator)
  - Timers

What is a Microcontroller?

- Processor
- Memory
  - Program Memory
  - RAM
- Peripherals
  - Clock Generator
  - Digital General Purpose I/O
  - Analog I/O (eg. ADC, Comparator)
  - Timers
  - Hardware Serial Communication (eg. UART, SPI, I^{2}C)

Table of Contents

1 Introduction

2 C

- Data types
- Operators
- Events

3 Microcontroller

Data types

- Integer types (`uint8_t`,`uint16_t`,`int32_t`,..)

Data types

- Integer types (`uint8_t`,`uint16_t`,`int32_t`,..)
- Float types (`float`,`double`,..)

Data types

- Integer types (`uint8_t`,`uint16_t`,`int32_t`,..)
- Float types (`float`,`double`,..)
- Enumerated types (`enum`)

Data types

- Integer types (`uint8_t`,`uint16_t`,`int32_t`,..)
- Float types (`float`,`double`,..)
- Enumerated types (`enum`)
- Void (`void`)

Data types

- Integer types (uint8_t,uint16_t,int32_t,..)
- Float types (float,double,..)
- Enumerated types (enum)
- Void (void)
- Derived types (pointers, arrays, structs, unions, function types,..)

Operators

Arithmetic

$+$	Adds two operands
$-$	Subtracts second operand from first
$*$	Multiplies both operands
$/$	Divides numerator by de-numerator
$++$	Increases integer by 1
$--$	Decreases integer by 1

Operators

Logical

&& Logical AND. Returns True when both operands are non-zero

|| Logical OR. Returns True when any of the operands is non-zero

! Logical NOT. Reverses the logical state of the operand.

Operators

Bitwise

- & Bitwise AND. Copies bit when it exists in both operands.
- | Bitwise OR. Copies bit when it exists in either operand.
- ^ Bitwise XOR. Copies the bit if set in one operand, but not both.
- ~ Flips the bits.
- << Binary Left Shift. Left operands value is moved left by right number of bits.
- >> Binary Right Shift. Left operands value is moved right by right number of bits.

Operators

Example

```
uint8_t A = 0xEE; // equal to 0b11101110
uint8_t B = 5;    // equal to 0b00000101
uint8_t C;        // declare C
C = A + B;       // C = 0b11110011
C = A && B;      // C = True = 0b00000001
C = A << 2;      // C = 0b10111000
C = A & B;       // C = 0b00000100
```

Polling and Interrupts

Two approaches to checking a state

Polling

- Check a value
- If changed, perform some action

Polling and Interrupts

Two approaches to checking a state

Polling

- Check a value
- If changed, perform some action

Interrupt

- When a change of a value happens, go immediately to ISR
- Perform Interrupt Service Routine (ISR)
- Resume code

Table of Contents

1 Introduction

2 C

- Data types
- Operators
- Events

3 Microcontroller

Memory

- Register

Memory

- Register
- Memory Space

Memory

- Register
- Memory Space
- Memory Mapped Peripherals

Memory

- Register
- Memory Space
- Memory Mapped Peripherals

Register	Name	Description
0x0800 0000		Flash Memory Start Address
0x2000 0000		SRAM Start Address
0x4002 0400	GPIOB_MODER	GPIO Port B Mode register
0x4002 000C	GPIOA_PUPDR	GPIO Port A Pullup register
0x4001 300C	SPI1_DR	SPI Data register

Programming in Eclipse

The procedure:

1. Read the manual

Programming in Eclipse

The procedure:

1. Read the manual
2. Start a project in Eclipse

Programming in Eclipse

The procedure:

1. Read the manual
2. Start a project in Eclipse
3. Write your code

Programming in Eclipse

The procedure:

1. Read the manual
2. Start a project in Eclipse
3. Write your code
4. Compile the code

Programming in Eclipse

The procedure:

1. Read the manual
2. Start a project in Eclipse
3. Write your code
4. Compile the code
5. Flash to Nucleo board with STLink

Planning

Today:

- Read the manual[1]

[1] Yes I know, that was on the previous slide as well. Do it! :-)

Planning

Today:

- Read the manual
- Get used to Eclipse

Planning

Today:

- Read the manual
- Get used to Eclipse
- Turn a LED on/off

Planning

Today:

- Read the manual
- Get used to Eclipse
- Turn a LED on/off

Planning

Today:

- Read the manual
- Get used to Eclipse
- Turn a LED on/off

What's yet to come:

- Timers
- Analog Peripherals
- SPI
- Build a Function Generator using DDS

Material

You can find all material on
http://www.scintilla.utwente.nl/docs/cursus
Make sure you download:

- The Manual
- The Usermanual of the Nucleo-F411RE
- The Reference Manual of the STM32F411RE