C2. ARM Cortex Arch.

Embedded Essentials

Standup

- One thing from last course

- Brief status with setup environment
- GITHUB REPO
- Clarify openocd, gnu arm toolchain, stmlink usb drv,eclipse

- Feature courses schedule

- Q&A from last course

Course agenda

C2. ARM architecture

C1. Intro to Embedded

Ama Dablam - Himalaya mountain peak

Today’s agenda

1. Finish the toolchain install

1. Flash the simple program

1. Use gdb to debug

1. Important documents used in the course

1. The ARM and ARM Cortex M architectures

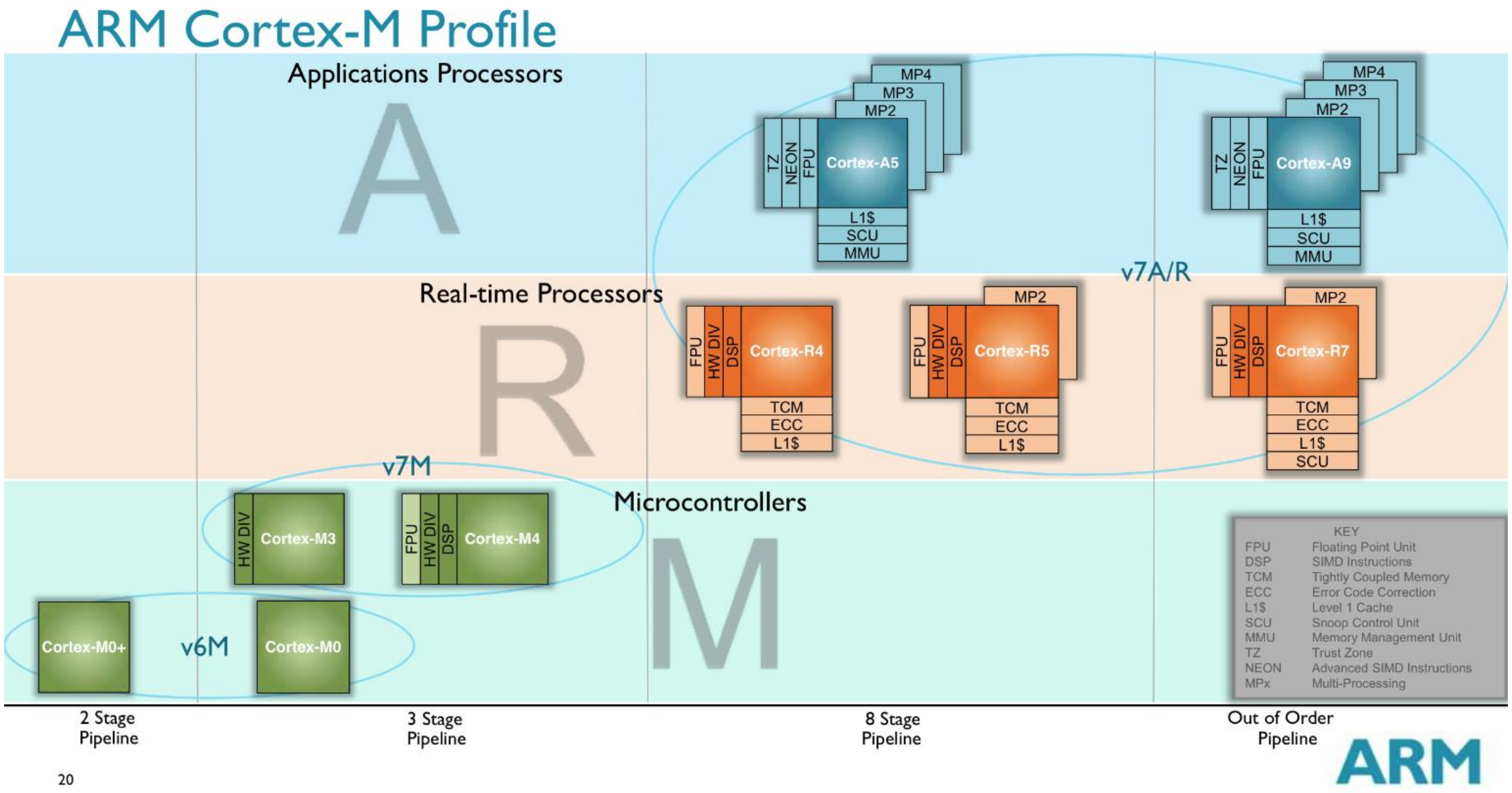
Toolchain installation

- Install toolchain
- Run the first example
- Debug the first example

Course documentation

- The **DATASHEET** keyword - for searching data related to a specific part :)
- Datasheet of STM32F051
- STM32F0 reference manual
- ARM Cortex M0 reference manual

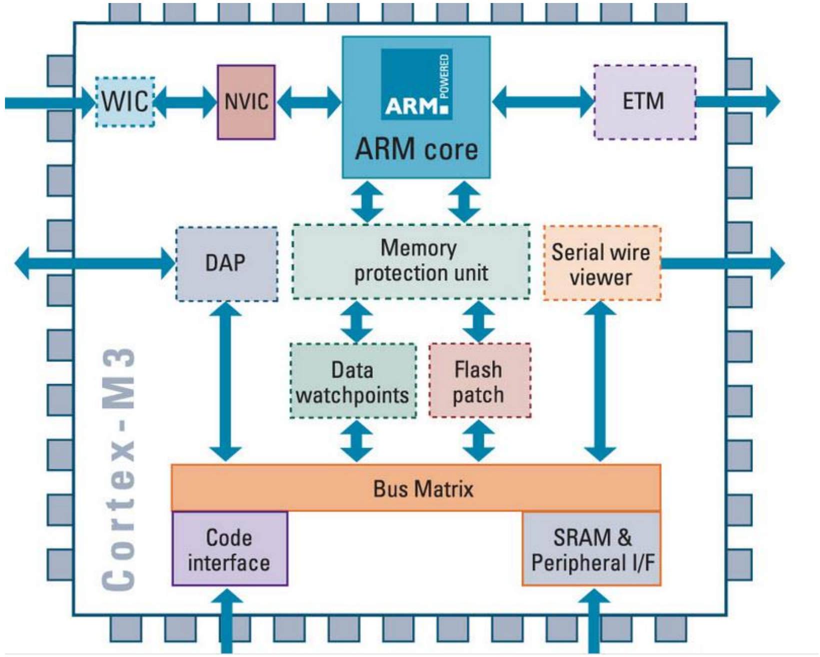
ARM architecture - general

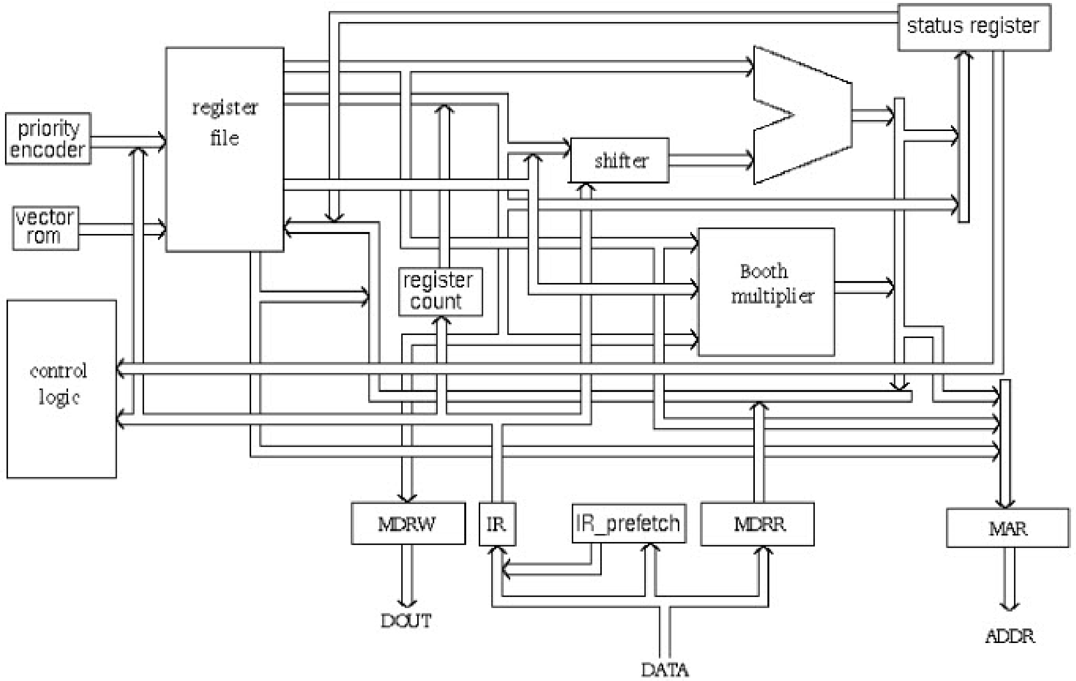


KEY

FPU	Floating Point Unit
DSP	SIMD Instructions
TCM	Tightly Coupled Memory
ECC	Error Code Correction
L1$	Level 1 Cache
SCU	Snoop Control Unit
MMU	Memory Management Unit
TZ	Trust Zone
NEON	Advanced SIMD Instructions
MPx	Multi-Processing

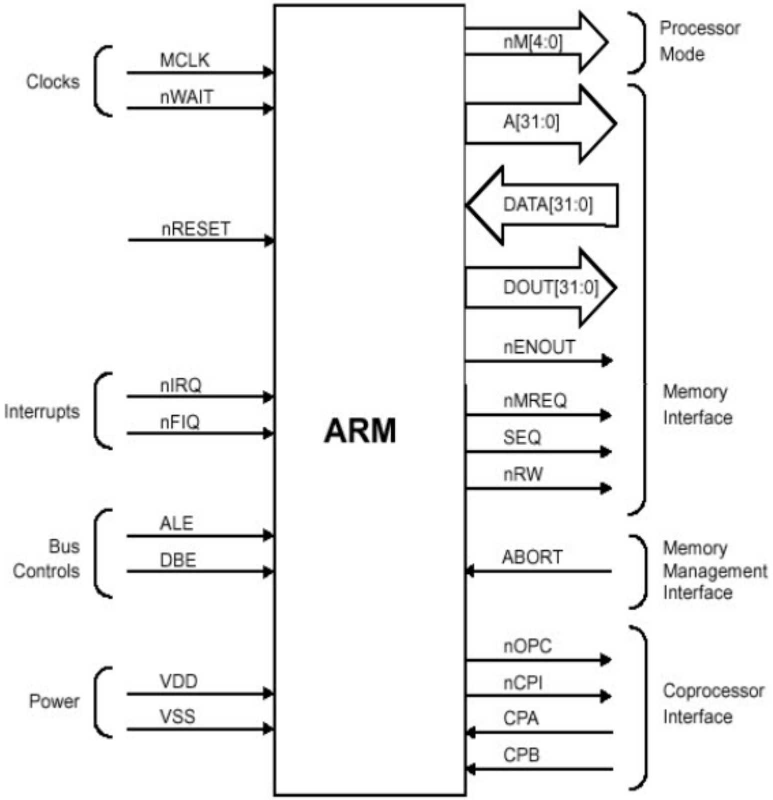
ARM architecture - general

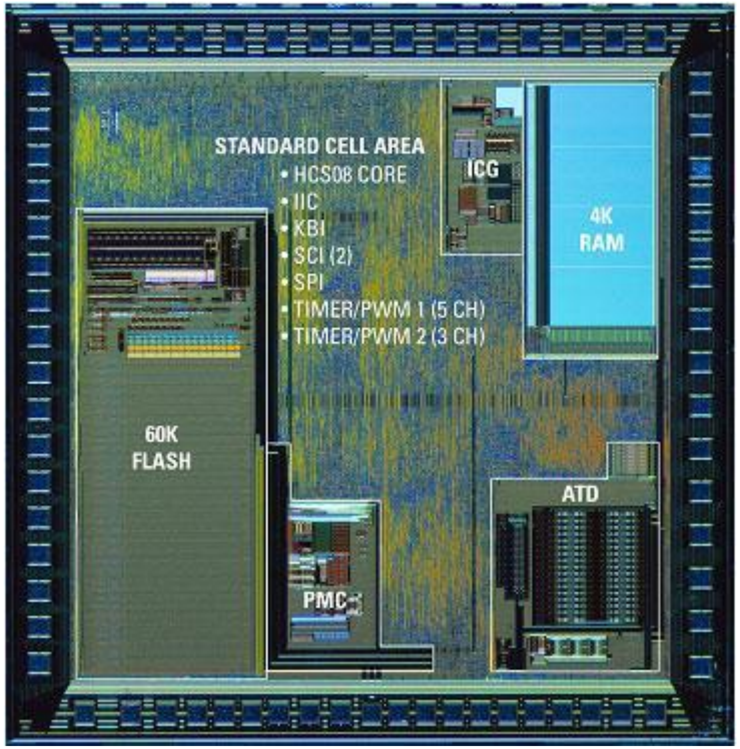




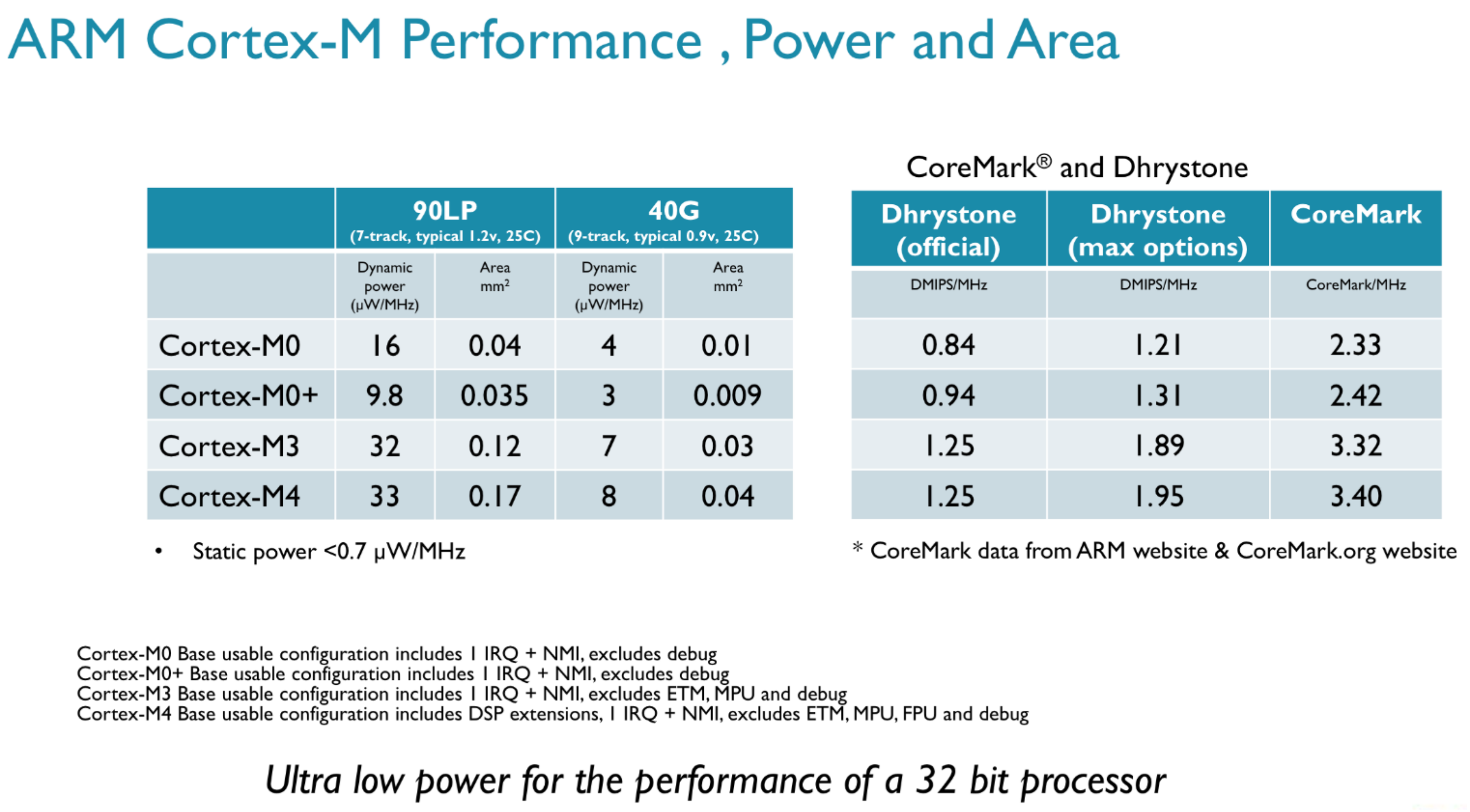
Further reading [ARM architecture]

ARM architecture - general





ARM Cortex M



ARM Cortex-M Performance , Power and Area

	90LP (7-track, typical 1.2v, 25C)		40G (9-track, typical 0.9v, 25C)	
	Dynamic power (µW/MHz)	Area mm^2	Dynamic power (µW/MHz)	Area mm^2
Cortex-M0	16	0.04	4	0.01
Cortex-M0+	9.8	0.035	3	0.009
Cortex-M3	32	0.12	7	0.03
Cortex-M4	33	0.17	8	0.04

- Static power <0.7 µW/MHz

CoreMark® and Dhrystone

Dhrystone (official)	Dhrystone (max options)	CoreMark
DMIPS/MHz	DMIPS/MHz	CoreMark/MHz
0.84	1.21	2.33
0.94	1.31	2.42
1.25	1.89	3.32
1.25	1.95	3.40

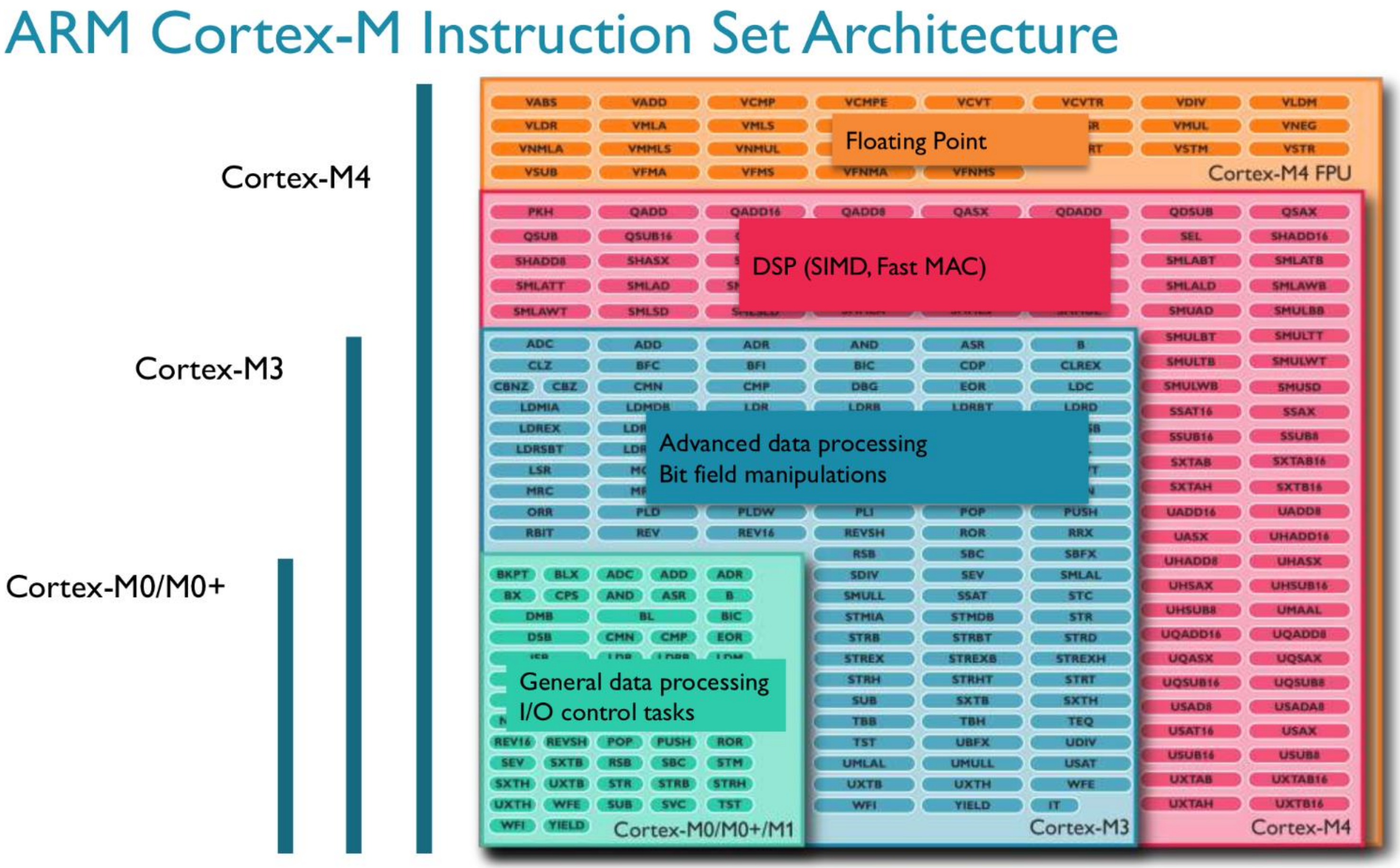
* CoreMark data from ARM website & CoreMark.org website

Cortex-M0 Base usable configuration includes 1 IRQ + NMI, excludes debug
Cortex-M0+ Base usable configuration includes 1 IRQ + NMI, excludes debug
Cortex-M3 Base usable configuration includes 1 IRQ + NMI, excludes ETM, MPU and debug
Cortex-M4 Base usable configuration includes DSP extensions, 1 IRQ + NMI, excludes ETM, MPU, FPU and debug

Ultra low power for the performance of a 32 bit processor

Further reading [ARM Cortex M]

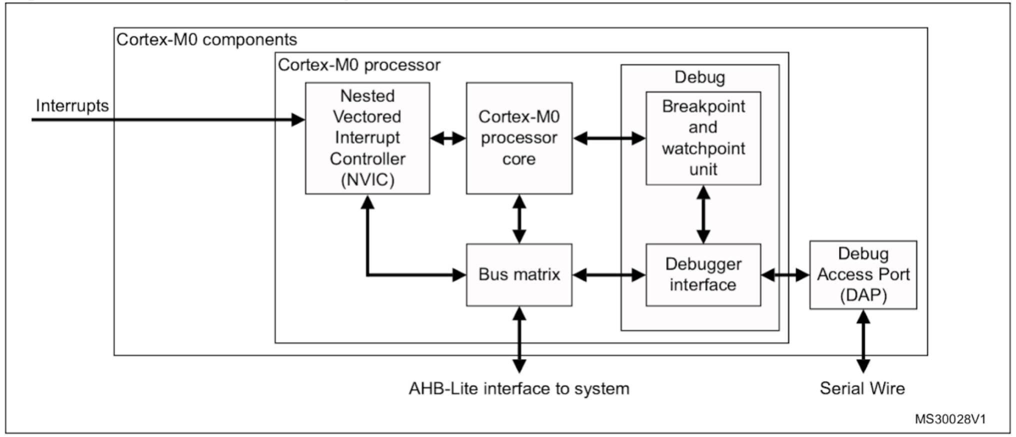
ARM Cortex M

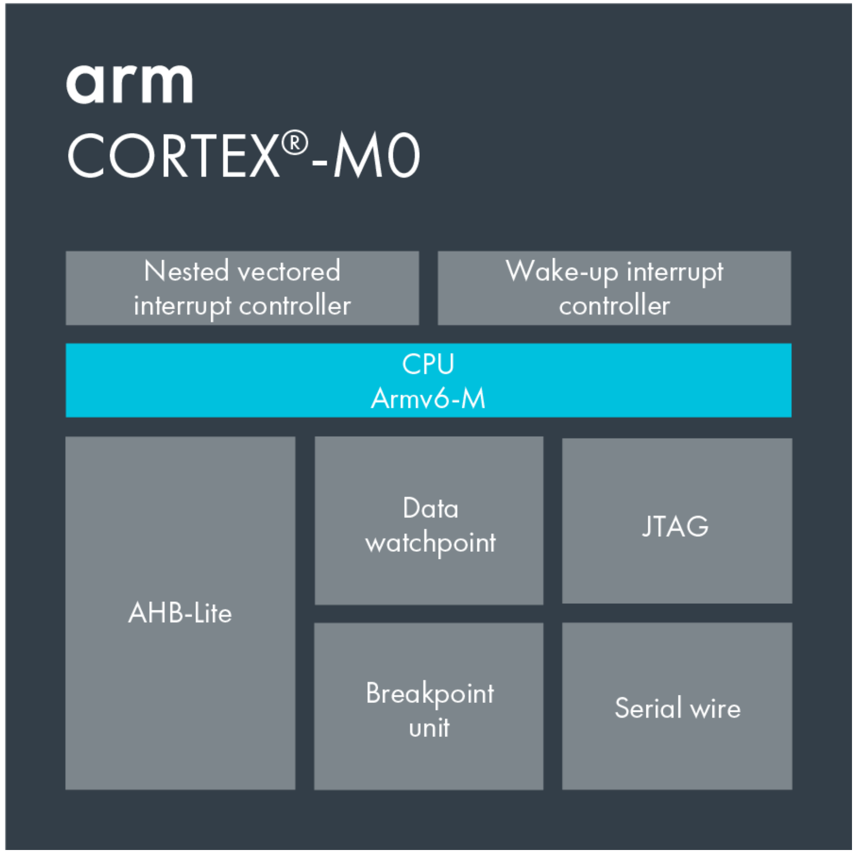


Further reading [ARM Cortex M]

ARM Cortex M0

Figure 1. STM32 Cortex-M0 implementation





➡ Further reading [ARM Cortex M]

Thank You!

Name	George Serea
Position	Software Developer
Email address	George.serea@rinftech.com
Phone number	+407202588930
Website address	https://gserea.github.io/

www.facebook.com/RINF.Tech

www.linkedin.com/company/rinf

Bucharest, 4 Gara Herastrau
Street, Green Court Skanska

Software Development Life Cycle

Design

- Customer needs
- Concept study
- Requirements
- Product design

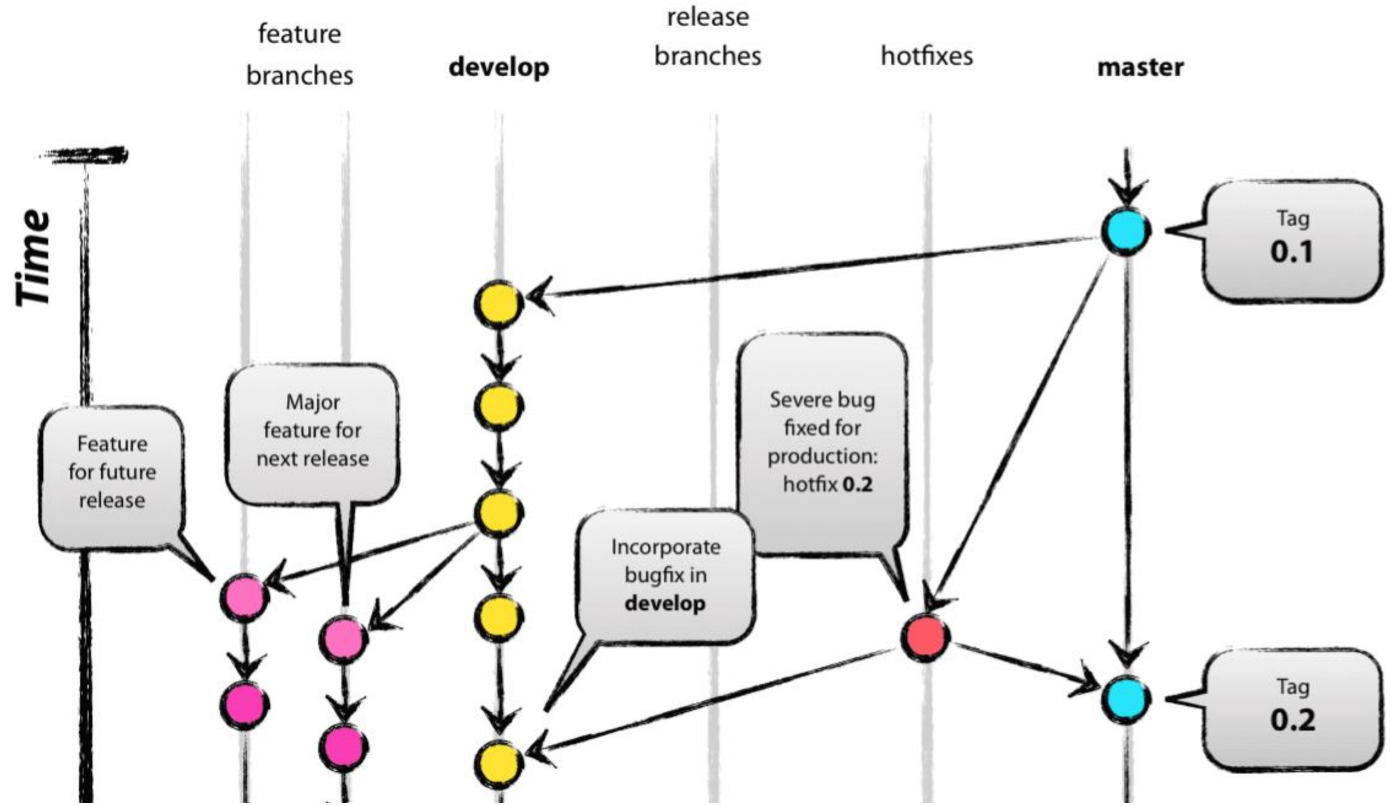
Develop

- Code development
- Bug solving
- Incremental releases
- Integration & testing

Operate

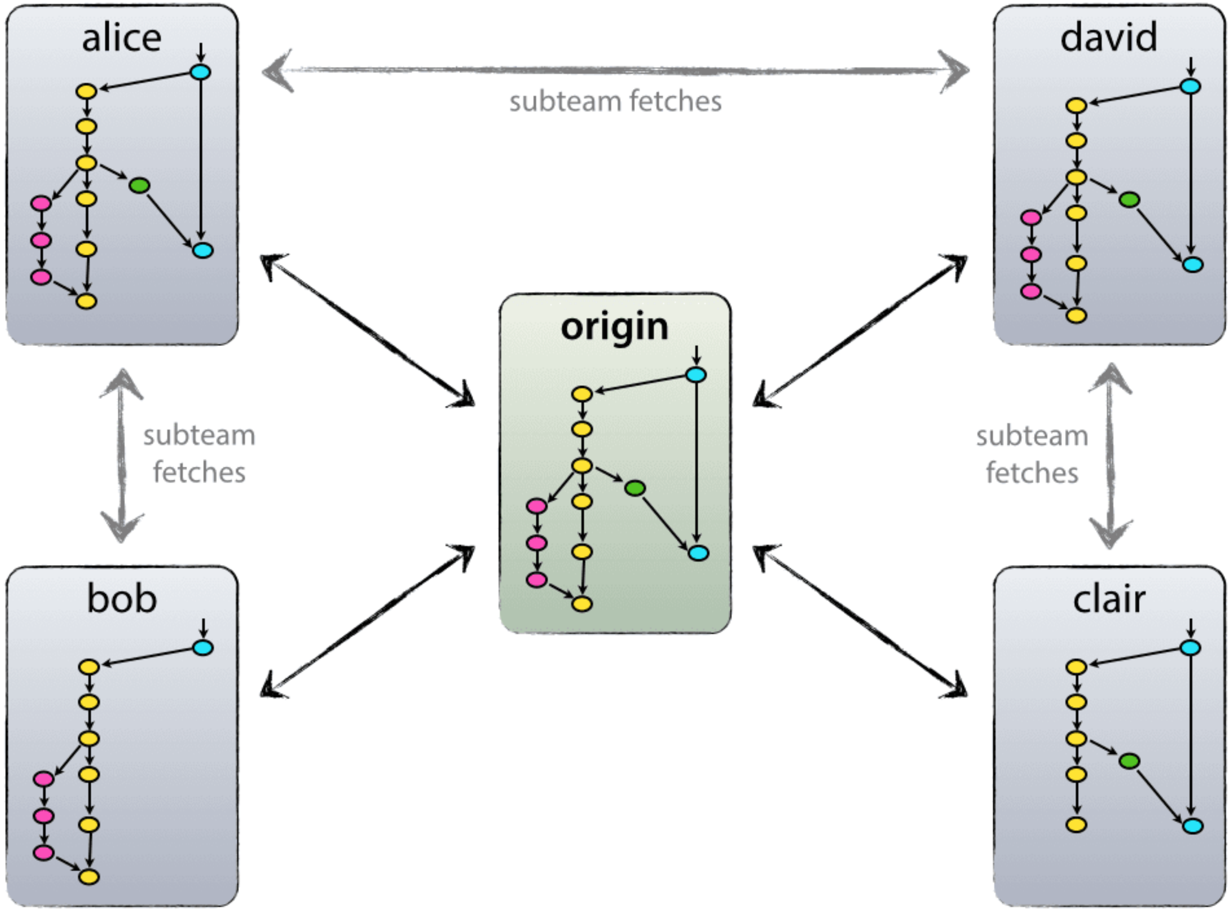
- Market releases
- Maintenance
- Client support
- Product phase-out

Software Versioning

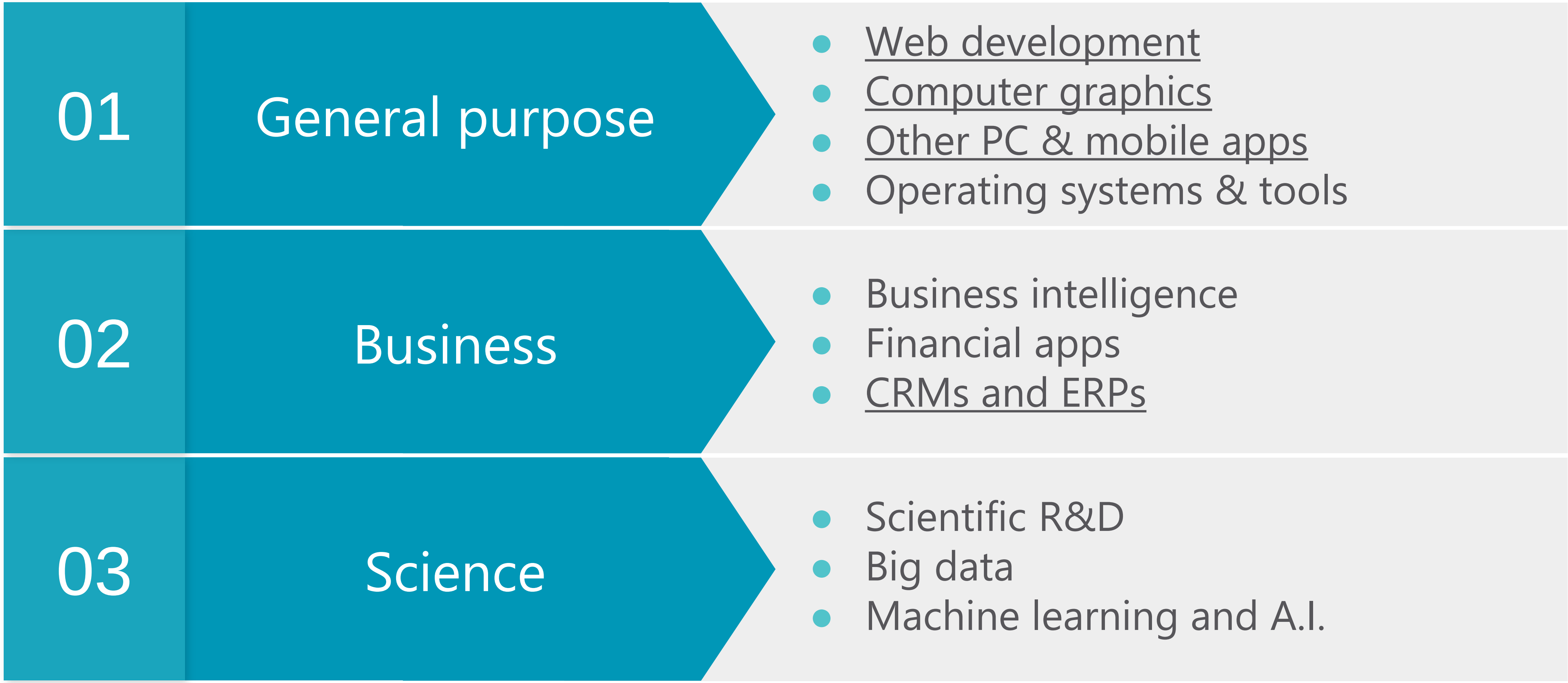


Further reading [Git Branching]

Software Versioning



Software Development Domains



01 General purpose

- Web development
- Computer graphics
- Other PC & mobile apps
- Operating systems & tools

02 Business

- Business intelligence
- Financial apps
- CRMs and ERPs

03 Science

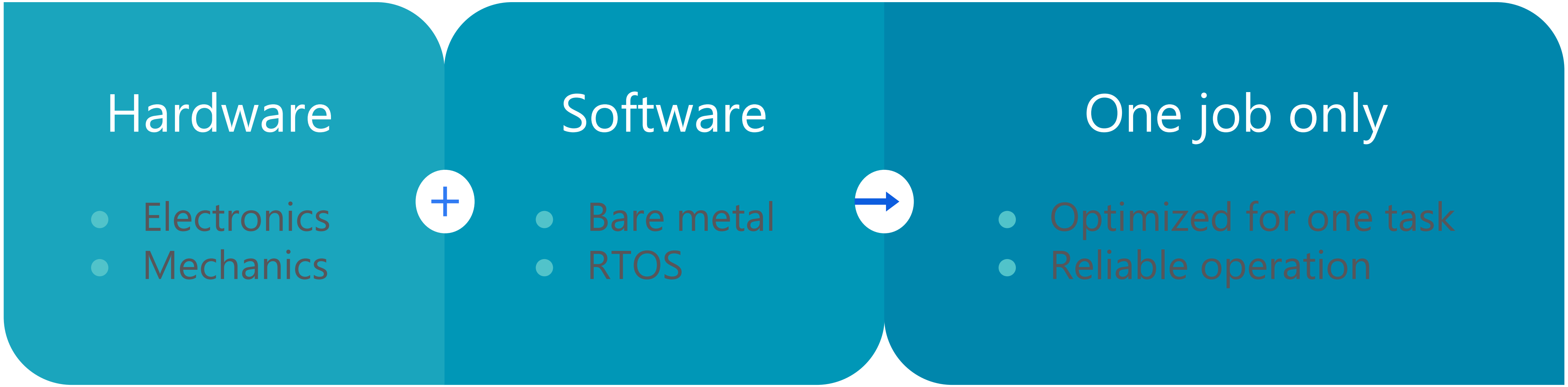
- Scientific R&D
- Big data
- Machine learning and A.I.

Software Development Domains

04 Embedded systems

- Automotive and transportation
- Medical devices
- Networking & telecom
- Mass market: IoT, camera, printer
- Energy production & distribution
- Space & Military devices
- Data collection and science R&D

Anatomy of an Embedded System



Hardware

- Electronics
- Mechanics

+

Software

- Bare metal
- RTOS

→

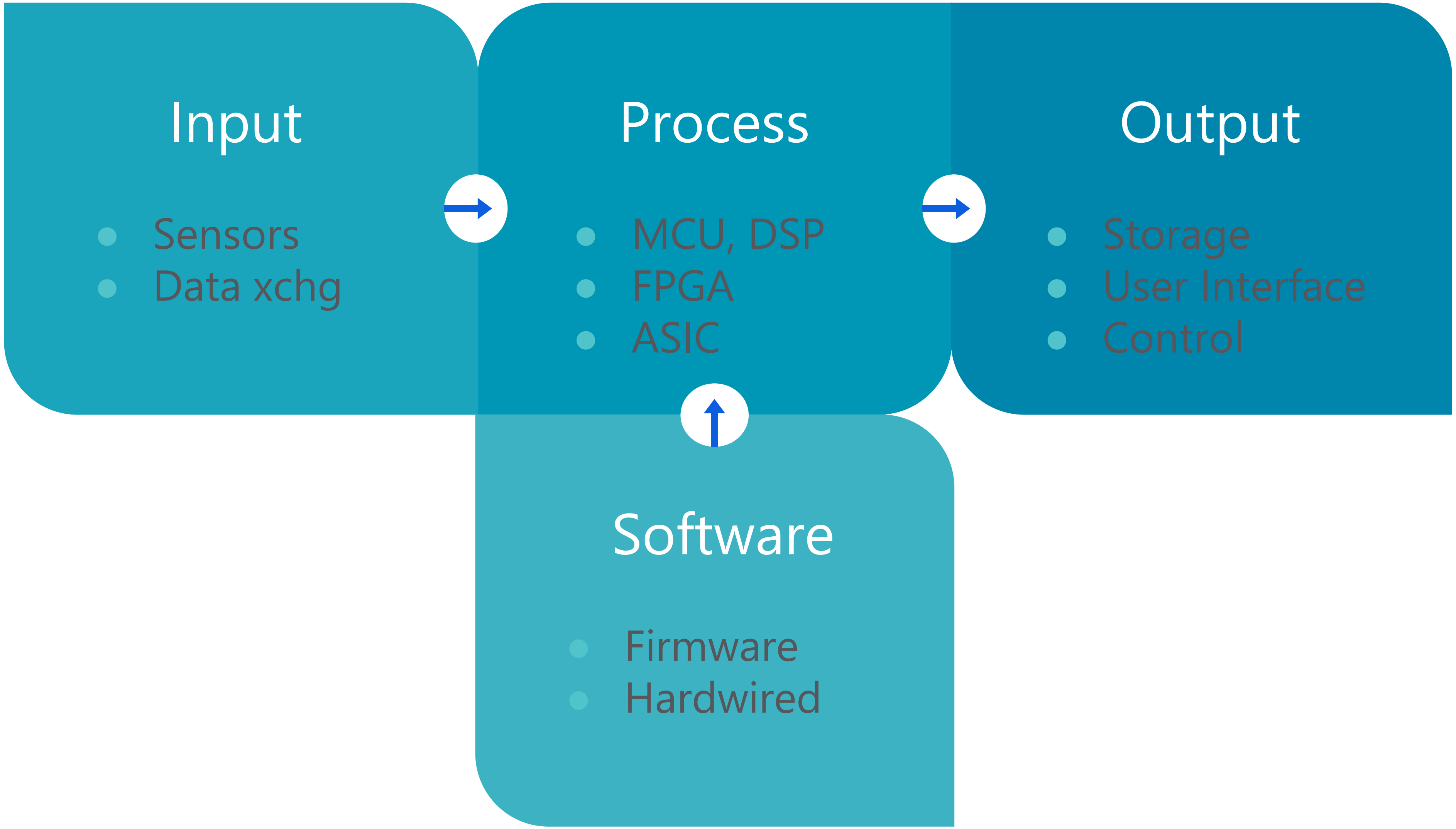
One job only

- Optimized for one task
- Reliable operation

Further reading [EmbeddedSystemCourse]

Embedded System

Anatomy of an Embedded System



Input

- Sensors
- Data xchg

Process

- MCU, DSP
- FPGA
- ASIC

Output

- Storage
- User Interface
- Control

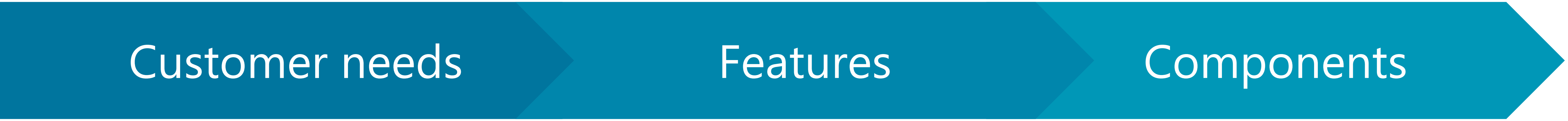
Software

- Firmware
- Hardwired

Some architecture choices

01	Cost	Just the resources needed
02	Reliability	Simple solution => few bugs
03	Time to market	Choose popular, easy to use & well supported solutions
04	Availability	Manufacturer can supply needed number for life-span duration of product

Design an embedded system



Customer needs

Features

Components

Describe what is the task the product needs to perform.

Describes what characteristics the product needs to possess to do the task.

Describes what actual components need to be present in the product to offer the needed capabilities (features).

Customer needs

Greenhouse monitor

I need a device that

- Measures temperature in my greenhouse
- Displays average temperature
- Lets me set a threshold temperature
- Gives alarm if temperature drops below threshold
- Monitors and shows total hours of sunlight

Home automation

I need a device that

- Displays the room temperature
- Lets me set desired temperature
- Starts the heater or AC when temperature goes below or above threshold
- Turns on a light after sunset when I enter the front door

Features

Greenhouse monitor

- Measure temperature
- Display various data
- Allows setting parameters
- Measure light intensity
- Gives an alarm (visual? audio?)

Home automation

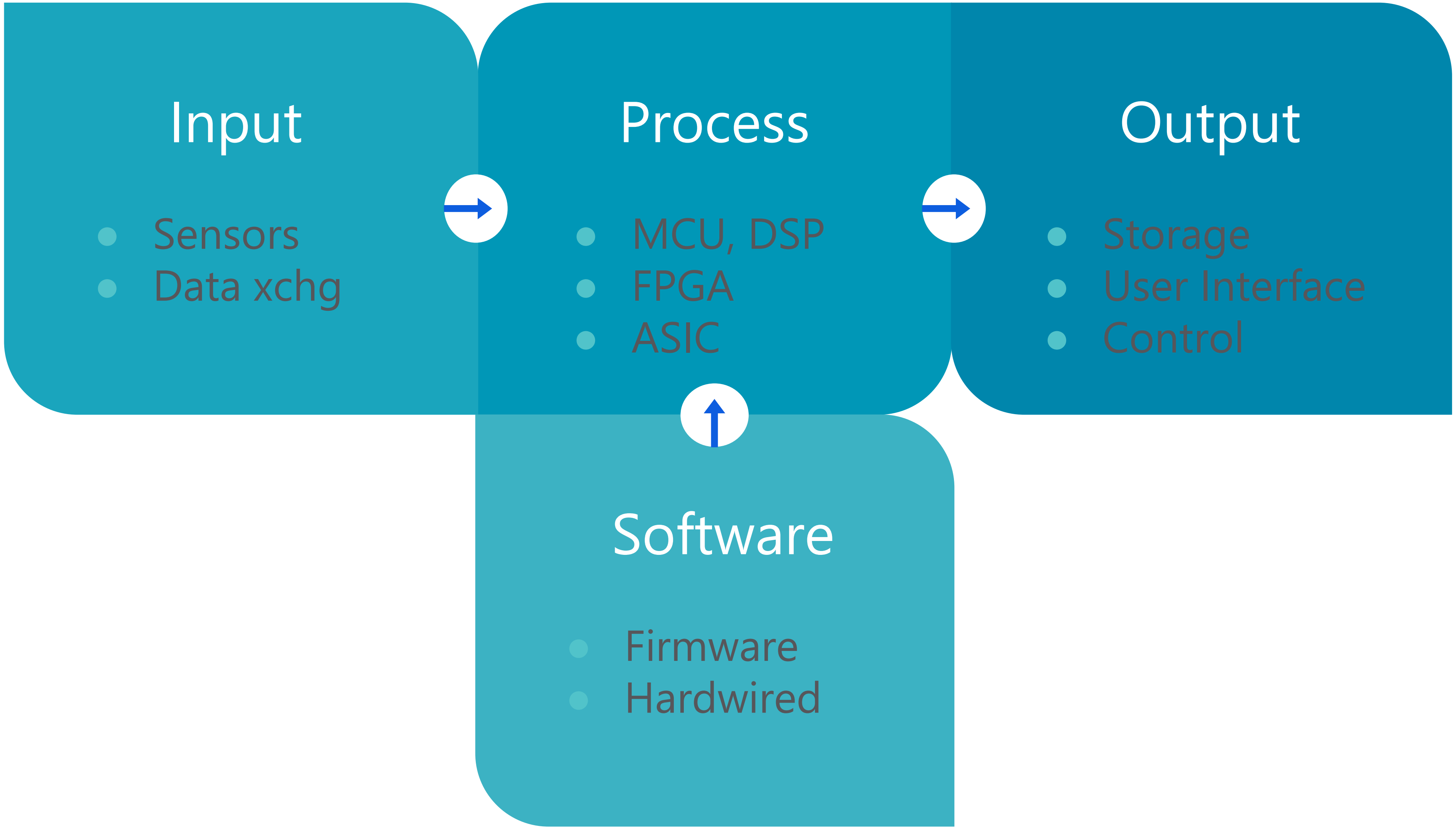
- Measure temperature
- Display various data
- Allows setting parameters
- Measure light intensity
- Control AC, heater, light

Components

01	Measure Temperature	Temperature Sensor
02	Display Data	Display (user interface)
03	Change Parameters	Buttons (user interface)
04	Measure Light Intensity	Light Sensor
05	Control light, heater, AC	Relays

Embedded System

Anatomy of an Embedded System



Input

- Sensors
- Data xchg

Process

- MCU, DSP
- FPGA
- ASIC

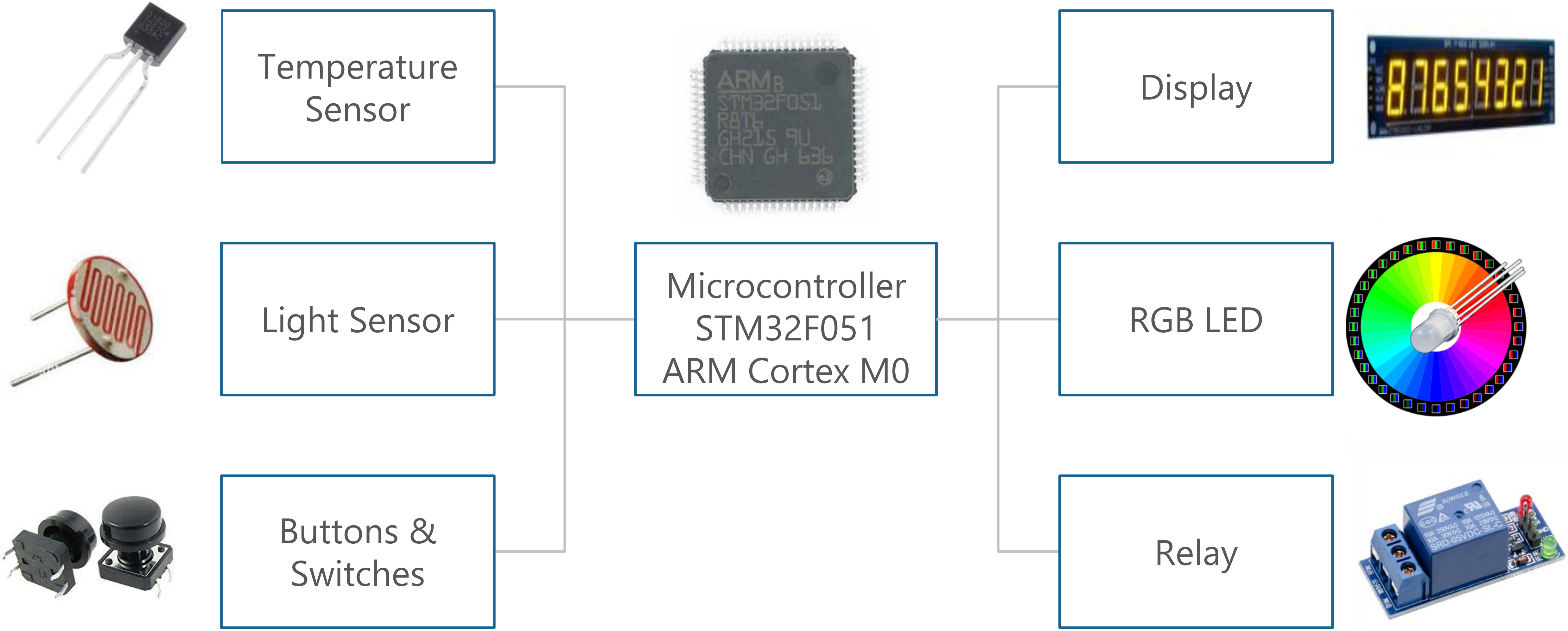
Output

- Storage
- User Interface
- Control

Software

- Firmware
- Hardwired

Our Embedded System



Temperature Sensor

Light Sensor

Buttons & Switches

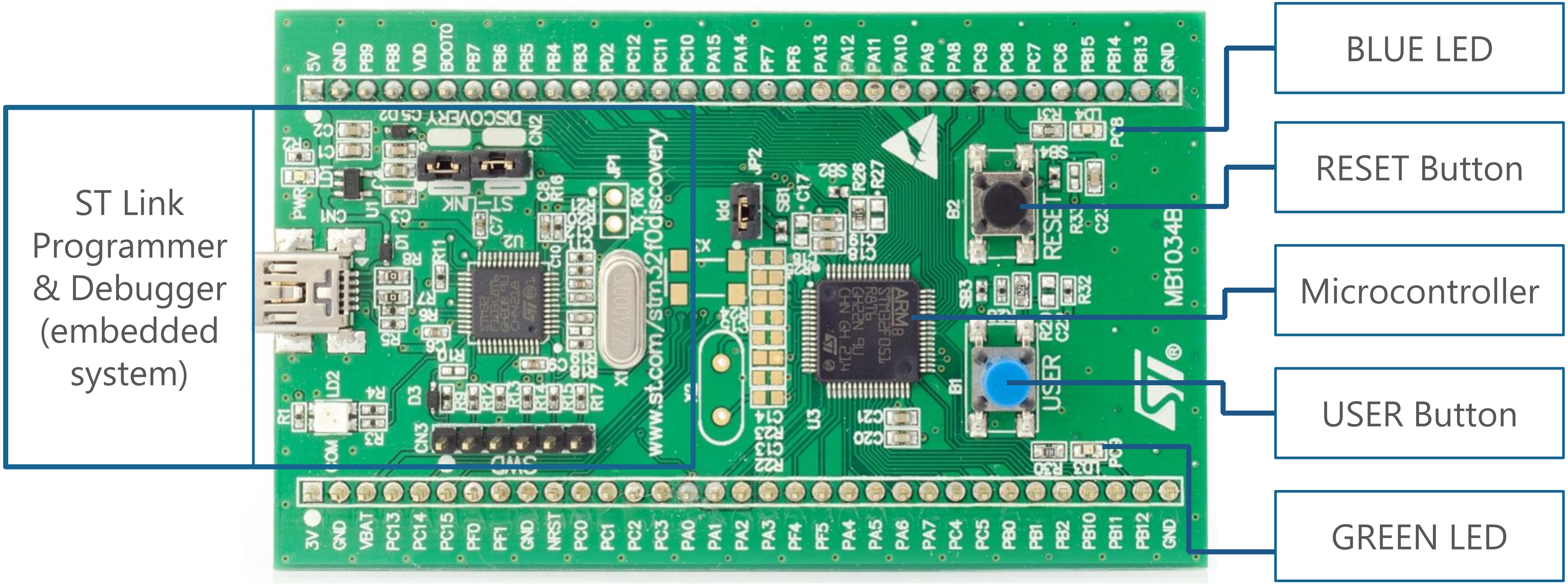
Microcontroller STM32F051 ARM Cortex M0

Display

RGB LED

Relay

Our Embedded System



ST Link Programmer & Debugger (embedded system)

BLUE LED

RESET Button

Microcontroller

USER Button

GREEN LED

5V GND PB9 PB8 VDD BOOT0 PB7 PB6 PB5 PB4 PB3 PD2 PC12 PC11 PC10 PA15 PA14 PF7 PF6 PA13 PA12 PA11 PA10 PA9 PA8 PC9 PC8 PC7 PC6 PB15 PB14 PB13 GND
3V GND VBAT PC13 PC14 PC15 PF0 PF1 GND NRST PC0 PC1 PC2 PC3 PA0 PA1 PA2 PA3 PF4 PF5 PA4 PA5 PA6 PA7 PC4 PC5 PB0 PB1 PB2 PB10 PB11 PB12 GND
DISCOVERY
ST-LINK
www.st.com/stm32f0discovery
SWD
RESET
USER
MB1034B
PC8
PC9

Environment: Windows

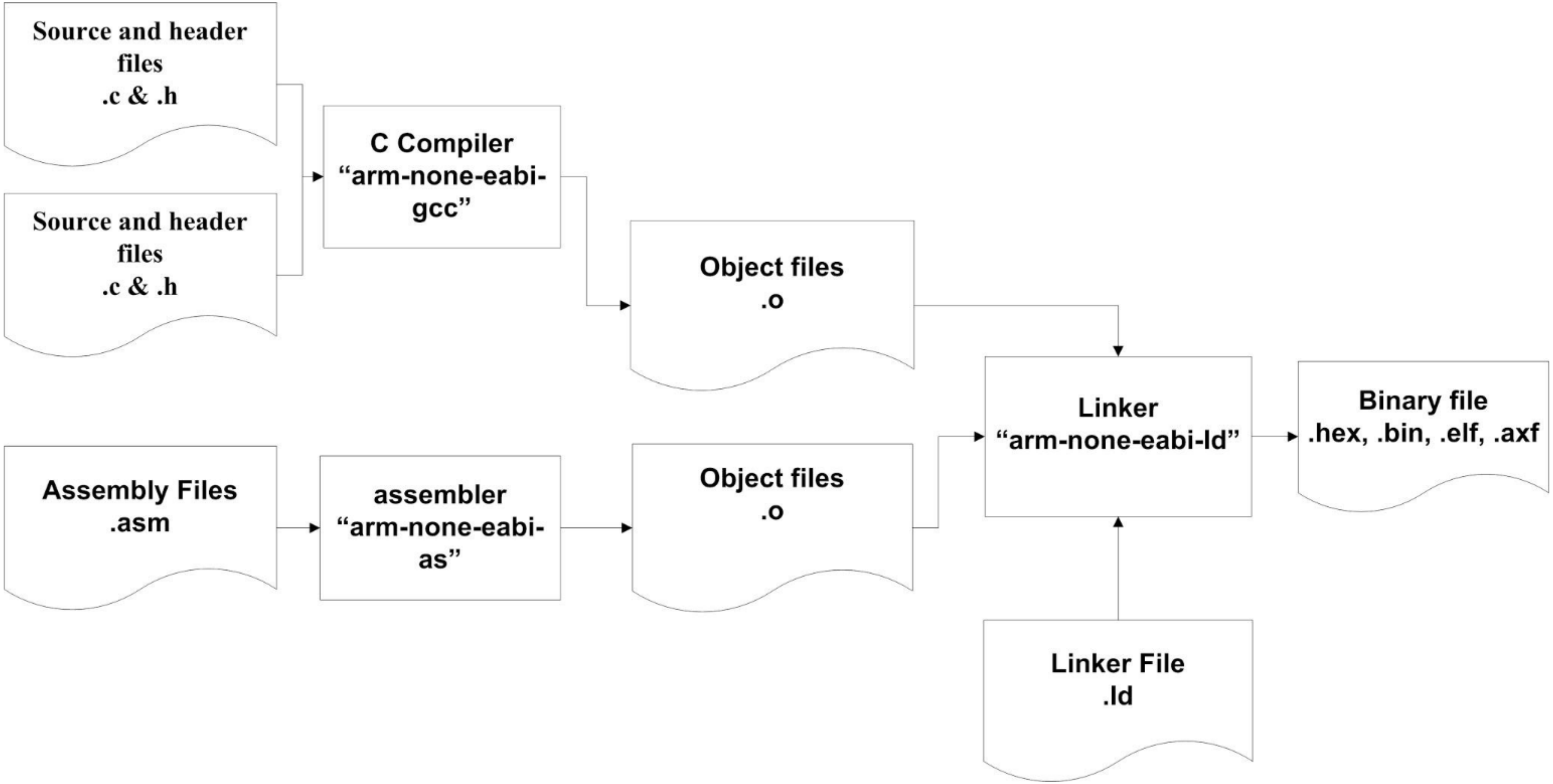
Open Windows setup - STM32F0Discovery
and follow the steps

Open Linux setup - STM32F0Discovery
and follow the steps

C language

- Some C important concepts
- Memory mapping

C toolchain


Source and header files
.c & .h
Source and header files
.c & .h
C Compiler
“arm-none-eabi-gcc”
Object files
.o
Assembly Files
.asm
assembler
“arm-none-eabi-as”
Object files
.o
Linker
“arm-none-eabi-ld”
Binary file
.hex, .bin, .elf, .axf
Linker File
.ld

Hardware

- Basic hardware notions
- Analog circuits
- Digital circuits

Microcontrollers

- Meet the microcontroller datasheet

Conclusion

- Our selected project
- Microcontroller architecture
- Next steps, homework
- Q & A