Microprocessor systems

EMISY

Semester 20L – Summer 2020
© Maciej Urbanski, MSc

email: M.Urbanski@elka.pw.edu.pl

Course organization – the thing the student needs to know

Lectures, projects: **Wednesdays 16:15 – 18:00**	**Maciej Urbanski**
Laboratory classes: **Mondays 14:15 – 20:00** **Thursdays 17:15-20:00**	**Bartosz Gąsowski** **Maciej Urbanski** **Stanislaw Hanasz**

Consultations (helpdesk) – when the student may reach us:

Maciej Urbanski:	Wednesdays, 18:30-19:30,	room 254 (need to knock the door)
Bartosz Gąsowski:	contact via email	
Stanislaw Hanasz:	contact via email	

Due to other lectures and laboratory classes the best way to communicate is via emails:

Email: M.Urbanski@elka.pw.edu.pl
Email: B.Gasowski@elka.pw.edu.pl
Email: shanasz@elektron.elka.pw.edu.pl

Course organization – the schedule

Lecture 1	26.02	- Organization, introduction
Lecture 2	04.03	- Architecture of microcontrollers
Lecture 3	11.03	- Memory arrangement in microcontrollers
Lecture 4	18.03	- Basic peripherals in microprocessor systems – clocks, I/O ports
Lecture 5	25.03	- Integrated and external peripherals in microprocessor systems, part 1
Lecture 6	01.04	- Principles of assembler language for 8051 microcontrollers and KEIL IDE
Lecture 7	08.04	- Test 1, Peripherals configuration in 8051 microcontrollers
Lecture 8	15.04	- Integrated and external peripherals in microprocessor systems, part 2
Lecture 9	22.04	- Integrated and external peripherals in microprocessor systems, part 3
Lecture 10	06.05	- A/D and D/A converters basics
Lecture 11	13.05	- Power consumption in microcontrollers. Power saving methods
Lecture 12	27.05	- Test 2, IDEs, programmers, debuggers, other programming languages, PROJECT DEADLINE
Lecture 13	03.06	- Examples of microprocessor systems
Lecture 14	10.06	- Makeup test, end of the course

How to pass the course and score well?

This course does not end with the final exam.

During the course student can score in total 100 points.

There will be two tests during the semester:

Test 1:	on Lecture 7 (8th April 2020)	all the topics up to Lecture 6	25 points
Test 2:	on Lecture 13 (27th May 2020)	topics from Lecture 7 up to Lecture 11	25 points

Tests will be based on the lecture material only.

There will be ONE makeup test, during Lecture 14 (12th June 2019)

This will be an opportunity to get better result from one of the tests (NOT BOTH).
The better result will be taken into final account.

In total student can score up to 50 points from tests.

How to pass the course and score well?

Projects

List of available project topics will be given during the next lecture.

Students are encouraged to suggest their own topics. In special cases (after Project Manager's approval) students' own projects can be realized using other than 8051 microcontrollers.

Each student has to select and apply for the topic to the Project Manager (**via email**).

Project task is to design a microprocessor based system that will **fulfil the project requirements**. Project must be done using an Intel 8051 family based microcontroller.
Any non 51' uC is prohibited and it's use will result in failing the project and thus the course !!!

Student can score up to 20 points with the project. Project score is based on the project report that has to be sent to Project Manager via email before the DEADLINE.

Students are requested to discuss the project topics and their ideas with the Project Manager.

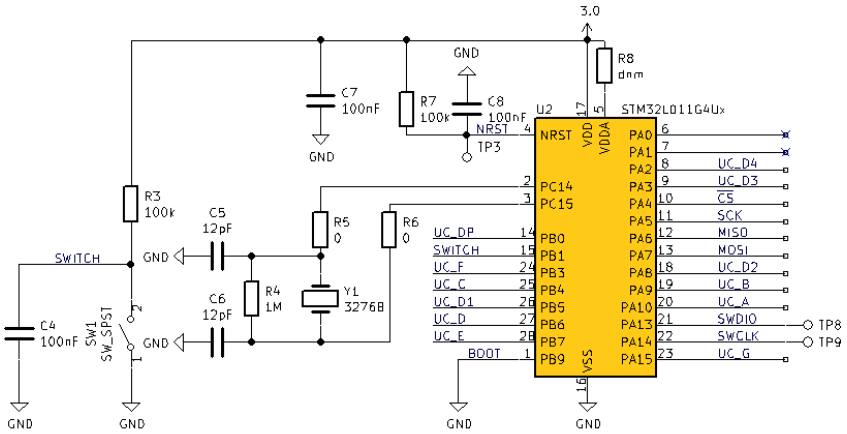
Student has to prepare detailed documentation (block diagram with descriptions, functional description, electrical schematic and draft of algorithm for microprocessor's firmware).
Project does not have to be done in hardware.

Project DEADLINE passes on 27.05 (after Lecture 12). Projects given after WILL NOT BE CHECKED and will cause failing the course.

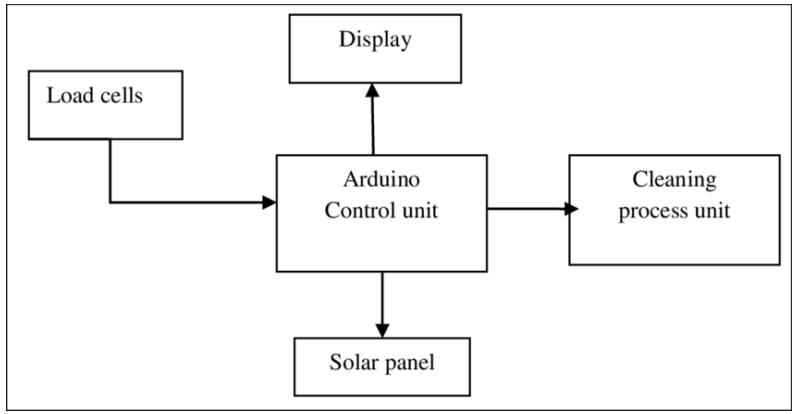
How to pass the course and score well?

Project report contents

- Detailed circuit diagram
  - Values/types of all the components
  - Components' selection explanation
- No black boxes allowed
- Power supplies do not have to be designed (unless the project explicitly requires that)
- Diagram description, including all the details needed to make the project
- Analysis of the project
  - Discussion of the project requirements
  - Discussion of the solution – **showing the way of thinking**
- Draft of the microcontroller firmware
  - Block diagram for the algorithm



Detailed circuit diagram
(STM32L011G4U6 basic application schematic)



Definitely not a detailed circuit diagram

How to pass the course and score well?

Laboratory classes

There will be 3 laboratory classes, realized in groups of 16 student
Labs are realized by pairs of students.
Labs will take place Mondays 14:15-17:00, 17:15-20:00,
Students have to sign up for desired laboratory classes timeslot.

Lab 1: LCD display

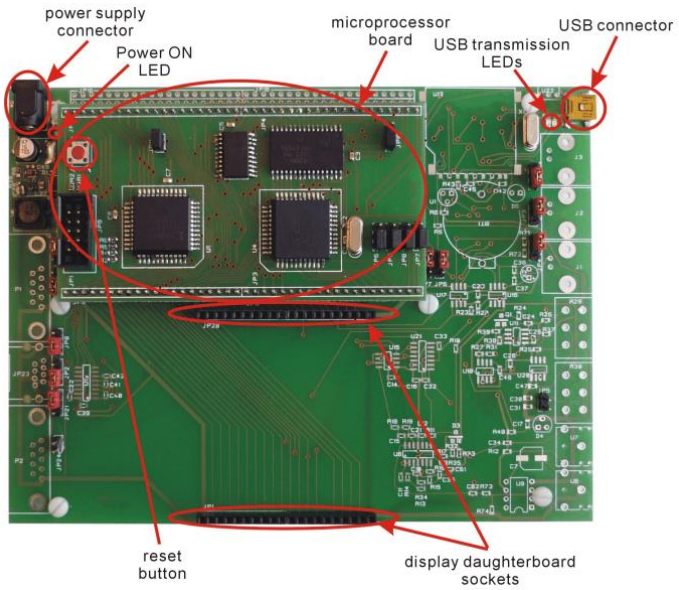
Write a program that will display your name on the LCD display (first name in the upper line, family name in the bottom line of the display). The program must perform software LCD initialization.

Lab 2: LED display

Write a program responsible for controlling the dynamic LED display so that it would display a text given by the teacher.

Lab 3: Keyboard

Write a program that would scan the keyboard and display number of the pressed key (hex digit 0..F if a key is pressed, blank display if all the keys are released) on the display.



Labkit, featuring AT89S52 microcontroller

Lab classes start after Lecture 7.
Student's task is to prepare a simple program, written in **assembler.**
The code will be rated by the Lab Manager during lab classes or shortly after (within two days).
Student can score up to 10 points per lab, in total 30 points to be scored with laboratory classes

It is possible to pass the course with the labs failed, the lab result has an impact on the final score of the course.

Student has to be able to explain how the prepared code works. It's forbidden to use the code that student has not developed – **detected trials of such behaviour will cause the failure of ALL lab classes.**

How to pass the course and score well?

Labs schedule

There are 4 groups (10 students per group):
This will be confirmed during next lecture.

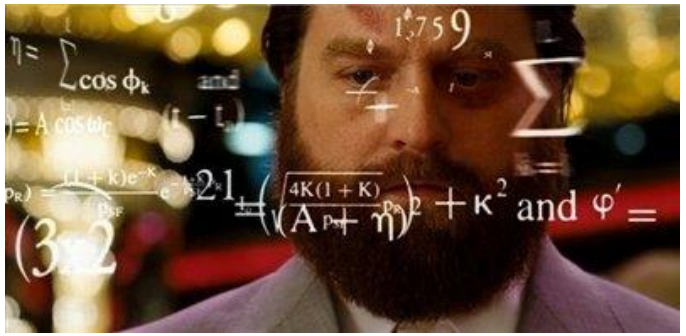
	Group A: 14:15-17:00	Group B: 17:15-20:00	Group C: 14:15-17:00	Group D: 17:15-20:00	Group E: 17:15-20:00
L1	**06.04**	**06.04**	**20.04**	**20.04**	**09.04**
L2	**27.04**	**27.04**	**04.05**	**04.05**	**30.04**
L3	**11.05**	**11.05**	**25.05**	**25.05**	**14.05**
Extra	**01.06**	**01.06**	**08.06**	**08.06**	**04.06**

Extra lab gives an opportunity to correct one selected lab (improve result), or take missed lab (due to unexpected events).
Students must apply for the extra lab to the lecture manager for approval.

Final scores and how they will be calculated...

Final scores

Lecture tests:	50 points
Project:	20 points
Lab classes:	30 points
TOTAL:	100 points



The primary requirement for passing the course is to score minimum 25 points from lecture tests.
The secondary requirement is to score minimum 10 points from project.
If any of these requirement is not met the result is failing the course.

Laboratory classes are not required to pass the course, but they affect the final grade significantly.

Final grades:

Less than 50 points total	-	2.0	-	COURSE FAILED
In range from 50 to 60	-	3.0	-	COURSE PASSED
In range from 60 to 70	-	3.5	-	COURSE PASSED
In range from 70 to 80	-	4.0	-	COURSE PASSED
In range from 80 to 90	-	4.5	-	COURSE PASSED
In range from 90 to 100	-	5.0	-	COURSE PASSED, RESPECT !

References

Books:

- J.W. Coffron, W.E. Long – *Practical Interfacing Techniques for Microprocessor Systems*
- J.W. Coffron, *Getting started with 8080, 8085, Z80 and 6800 Microprocessors*
- R.J. Tocci, N.S. Widmer – *Digital Systems: Principles and Applications*
- W. Kleitz – *Digital and Microprocessor Fundamentals: Theory and Applications*
- J.W. Stewart, K.X. Miao – *The 8051 Microcontroller: Hardware, Software and Interfacing*
- C. Noviello – *Mastering STM32*
- P. Horowitz, W. Hill – *The Art of Electronics, third edition*
- Y. Zhu – *Embedded Systems with ARM Cortex-M Microcontrollers in Assembly Language*
- S. Barrett, D. Pack – *Microcontroller Programming and Interfacing: Texas Instruments MSP430*
- T. Grace – *Programming and Interfacing ATMEL's AVRs*
- C. Kuhnel – *AVR RISC Microcontroller Handbook*

Others:

- Databooks (Analog Devices, Atmel, Microchip, Freescale, Intel, NXP, Silicon Labs, Texas Instruments, STMicroelectronics)
- Application notes for various microprocessors and microcontrollers
- EMISY bundle: http://www.ise.pw.edu.pl/impuls/emisy1.html
- Studia3 server (English version) : http://studia3.elka.pw.edu.pl/

Not recommended:

- Wikipedia
- Various amateur designs published on the Internet
- Radio-amateur magazines, etc.

Microprocessor systems

EMISY

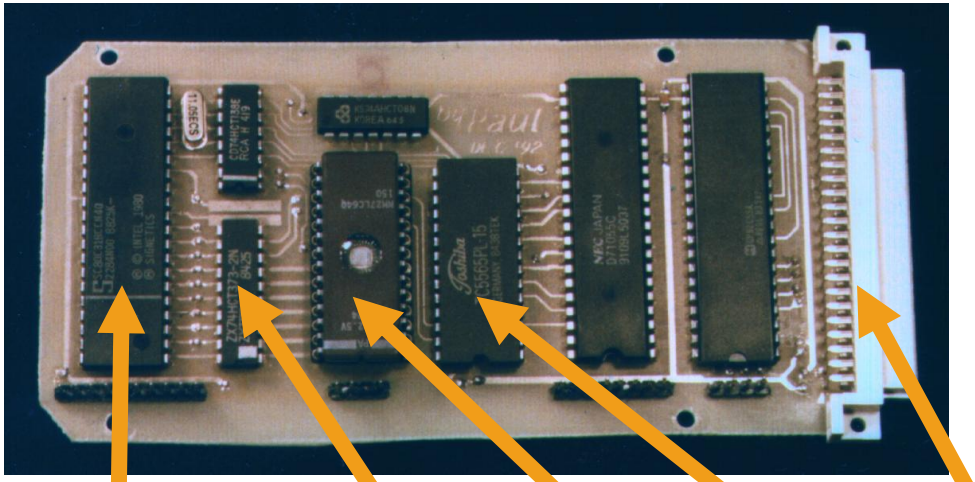
Introduction
Lecture 1

Semester 20L – Summer 2020
© Maciej Urbanski, MSc

email: M.Urbanski@elka.pw.edu.pl

A bit of history and why do we need the microcontrollers and microprocessors?

Microprocessor is a digital integrated circuit that is able to perform a set of operations, according to given set of instructions. Together with external peripheral circuits, like GPIOs (General Purpose Input/Output circuits), memories, buffers, etc., it creates a **microprocessor system.**



Microprocessor Buffers EPROM memory RAM memory (8 kB) I/O connector

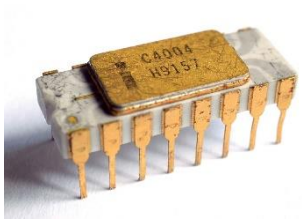
1992 microprocessor system board

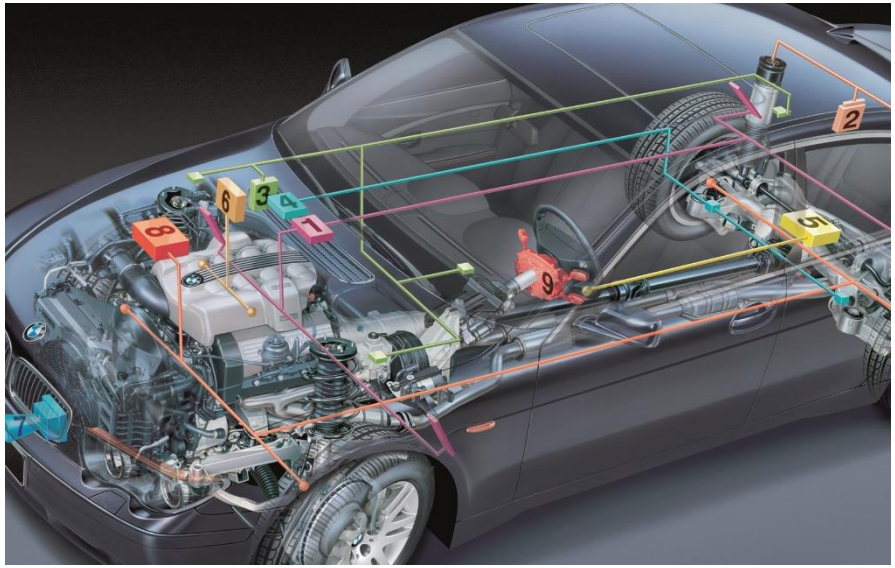
that's how it was done in the past...

A bit of history and why do we need the microcontrollers and microprocessors?

First designs in the 1970s – 4 bit processors MP944 (Garret AiResearch), clock speed 375 kHz, designed for CADC system for F-14 Tomcat Naval Interceptor US jet fighter

1971 – Intel 4004 CPU, first commercially Available microprocessor by Intel
1972 – Intel 8008 CPU
1974 – Motorola 6800
1976 – Zilog Z80
1993 – Intel Pentium
2000 – AMD Athlon XP
2017 – AMD Ryzen and lots of
Intel I3, I5, I7, etc.

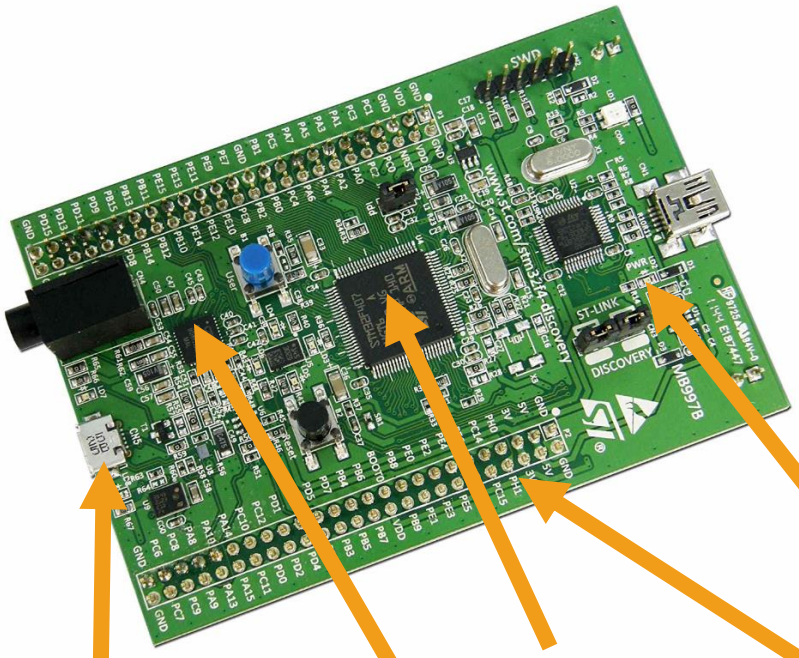




Not to forget about the softcores, like Intel NIOS, Xilinx Microblaze, or Lattice Mico

A bit of history and why do we need the microcontrollers and microprocessors?

Microcontroller is a single integrated circuit that has embedded peripheral circuits. It's structure allows to perform arithmetical operations based on given instruction set. It consists of a CPU, memory, GPIOs, timers, watchdogs, communication modules (I2C, SPI, USB, Ethernet), and many others, all withing a single chip. System created with a microcontroller is **also a microprocessor system.**

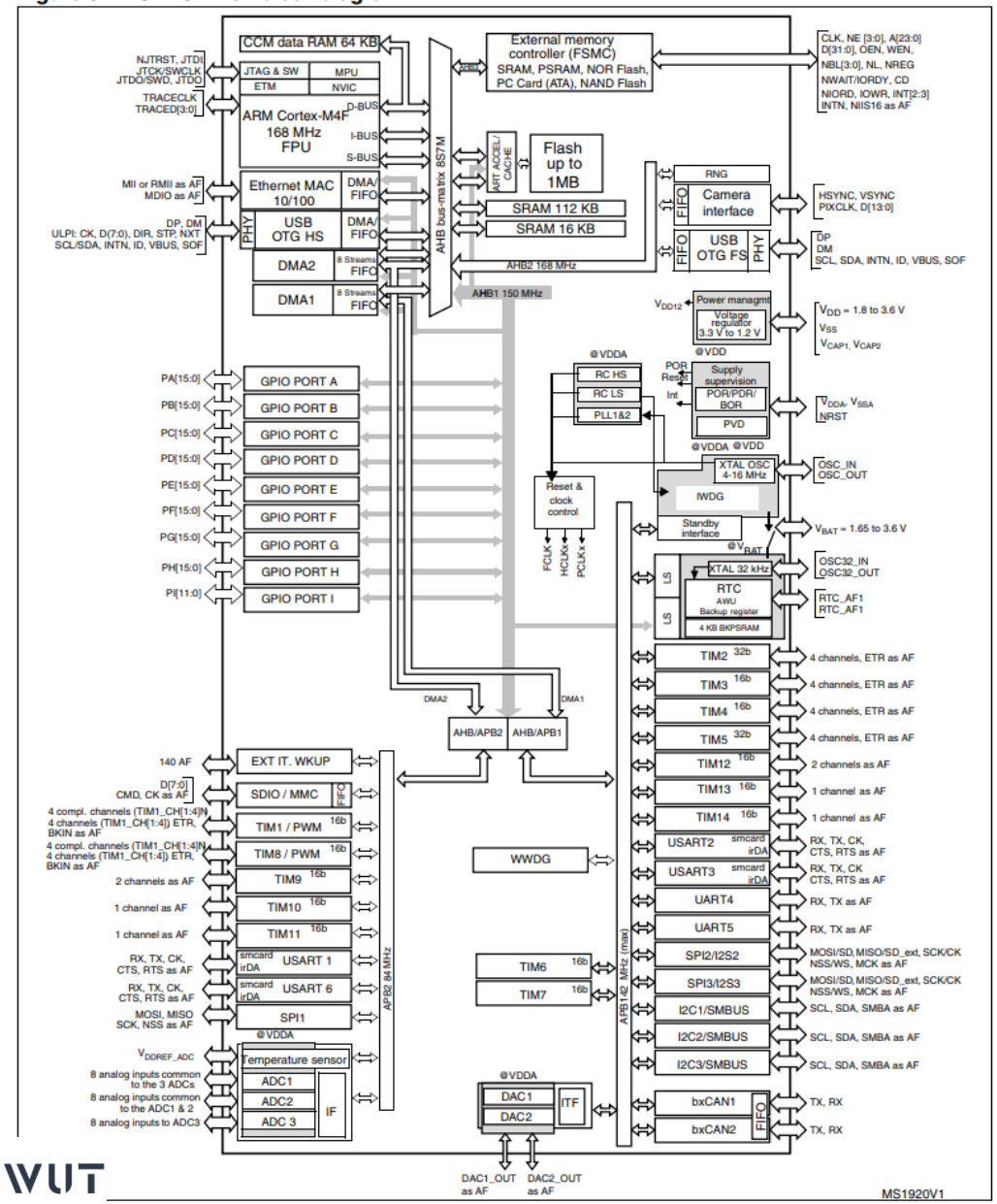


STMicroelectronics STM32F4 Discovery Evaluation board

An example of modern simple microprocessor system, based on ARM microcontroller

Internal structure of modern microcontroller

Figure 5. STM32F40x block diagram



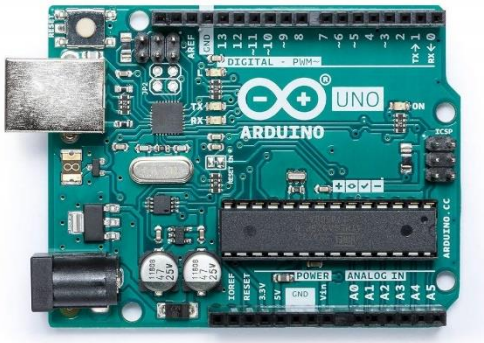
The most important elements of the microcontroller:

- CPU – a core of the microcontroller, it coordinates the operation of all other modules and circuits. During its operation it addresses the memory, input/output circuits and is responsible for the external control signals. Inside the CPU core there are:
  - **Registers** – temporary memory, storage for small amounts of data used during CPU operations
  - **ALU** – arithmetical logic unit – performs all the arithmetical and logic operations.
  - Instruction decoder
  - Interrupts controller
- Memory – this is where the firmware is stored and from where the CPU gets the instruction set to realize.
- Internal busses – they connect the modules of the microcontroller with the CPU
- Input/output ports – they allow the CPU to communicate with the external circuits.

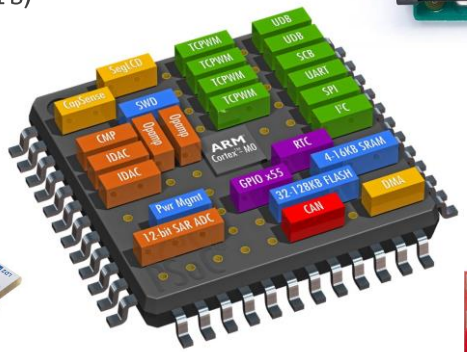
Most popular microcontroller families (and evaluation kits)

Microchip microcontrollers:
- Atmel AVR – most famous for the Arduino kits
- AVR32
- AT91SAM
- PIC microcontroller series

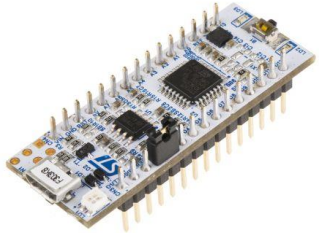
Cypress Semiconductors PSoCs (Programmable System-on-Chip)



Intel 8051, manufactured by many companies
(Atmel/Microchip, Infineon, NXP, Silicon Labs and many others)



Freescale:
- C08
- ColdFire
- 68HC11 (based on Motorola 6800)



Texas Instruments:
- MSP430
- MSP432

ARM core based microcontrollers:
- STMicroelectronics
  - STM32 family, with Discovery and Nucleo kit
- NXP Semiconductors
  - LPC1000, 2000 and 4000 microcontrollers
- Freescale Kinetis series
- Many, many others, including Raspberry Pi, Orange
  Odroid, Beaglebone, etc.

The biggest and most important companies in the market