Understanding Microcontrollers

Developed by MECB Ltd

A Trainers Toolkit To Foster STEM Skills Using Microcontroller Applications

Co-funded by the Erasmus+ Programme of the European Union

Project No. 2019-1-RO01-KA202-063965

This project has been funded with support from the European Commission. The content reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Understanding Microcontrollers

Content

- History
- Working Principle
- Types of Microcontrollers
- Off-the-Shelf Controllers
- Summary

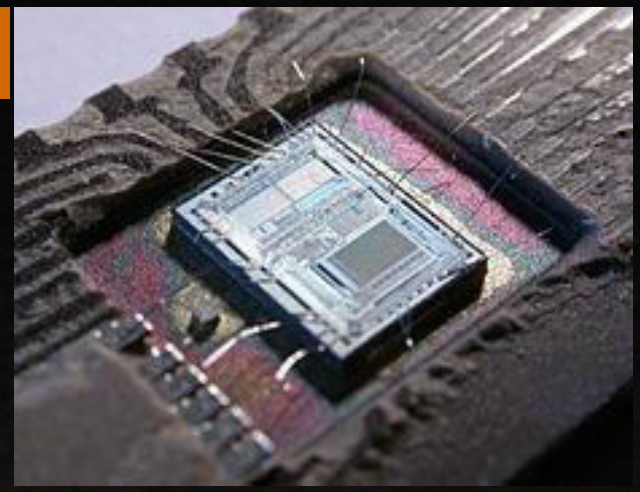
Project No. 2019-1-RO01-KA202-063965
This project has been funded with support from the European Commission. The content reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Co-funded by the Erasmus+ Programme of the European Union

History of Microcontrollers

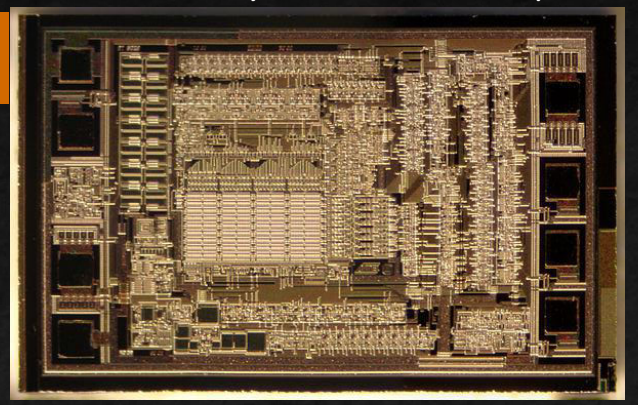
- Microcontrollers were first invented in the 1970s, this invention is considered to be the first since it incorporated the read-only memory, read/write memory and the processor onto one die/chip. From this, one of the most popular microcontrollers still being used today, the 8051 was developed by Intel.
- The second major development for the microcontroller occurred nearly right after, in the 1970s was the introduction of the EEPROM (Electrically Erasable Programmable Read-Only Memory), which allowed for the microcontroller to erase its memory with more compact electronics, allowing it to reduce in its size and cost over time and become more widely used as its application broadened.
- These have developed to a point where Humanity depends upon it to function since most modern technology would be impossible without them

1971



Die from Intel, by Ioan Sameli, Wikipedia

1972



EEPROM circuit, by Unknown, Wikipedia

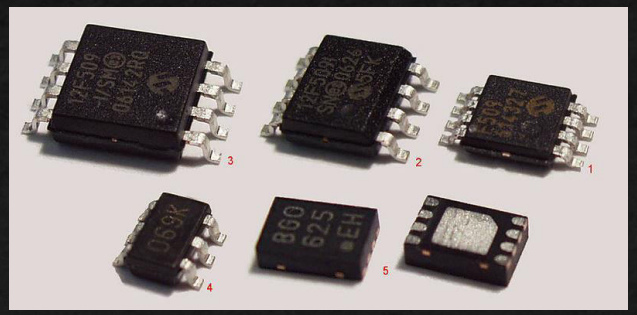
Project No. 2019-1-RO01-KA202-063965
This project has been funded with support from the European Commission. The content reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Co-funded by the Erasmus+ Programme of the European Union

What is a Microcontroller

- Microcontrollers are computers, which are made, designed and programmed for specific applications. These are used for controlling other parts of an electronic system, via inputs and outputs. These work exactly like a computer, with the difference that instead of something enormous, it can fit inside of your palm, or even down to your fingertip.
- Due to its size, it can be used in most applications, but due to this there is the obvious drawback that it has a smaller CPU, RAM, Input & Outputs, etc.. so careful though needs to be done when choosing the correct microcontroller

Desktop Computer vs Microcontroller, by Robot shop, https://www.robotshop./community/tutorials/show/how-to-make-a-robcomot-lesson-4-understanding-microcontrollers



Some Smallest Microcontrollers, by Unknown, Wikimedia

Project No. 2019-1-RO01-KA202-063965
This project has been funded with support from the European Commission. The content reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Co-funded by the Erasmus+ Programme of the European Union

Programming a Microcontroller

- All Controllers run on Assembly language, but it is tedious to program in this language so higher level programs are used to speed up the process such as C#, Java, Python, etc...
- Since the controller needs to run on assembly language, a Compiler (a program which converts from high-level language to low-level) is used to do this. This is a very important step since electronic systems run on Machine code not on high-level programming.
- The obvious advantage is that there is less time wasted upon writing the code but the other drawback is that typically a lot of extra code will be put into the microcontroller which slows it down typically.

```
MONITOR FOR 6802 1.4          9-14-80  TSC ASSEMBLER  PAGE    2

C000                    ORG     ROM+$0000 BEGIN MONITOR
C000 8E 00 70  START    LDS     #STACK

                        ****************************************
                        * FUNCTION: INITA - Initialize ACIA
                        * INPUT: none
                        * OUTPUT: none
                        * CALLS: none
                        * DESTROYS: acc A

0013                    RESETA  EQU     %00010011
0011                    CTLREG  EQU     %00010001

C003 86 13      INITA   LDA A   #RESETA   RESET ACIA
C005 B7 80 04           STA A   ACIA
C008 86 11              LDA A   #CTLREG   SET 8 BITS AND 2 STOP
C00A B7 80 04           STA A   ACIA

C00D 7E C0 F1           JMP     SIGNON    GO TO START OF MONITOR

                        ****************************************
                        * FUNCTION: INCH - Input character
                        * INPUT: none
                        * OUTPUT: char in acc A
                        * DESTROYS: acc A
                        * CALLS: none
                        * DESCRIPTION: Gets 1 character from terminal

C010 B6 80 04   INCH    LDA A   ACIA      GET STATUS
C013 47                 ASR A             SHIFT RDRF FLAG INTO CARRY
C014 24 FA              BCC     INCH      RECIEVE NOT READY
C016 B6 80 05           LDA A   ACIA+1    GET CHAR
C019 84 7F              AND A   #$7F      MASK PARITY
C01B 7E C0 79           JMP     OUTCH     ECHO & RTS

                        ****************************************
                        * FUNCTION: INHEX - INPUT HEX DIGIT
                        * INPUT: none
                        * OUTPUT: Digit in acc A
                        * CALLS: INCH
                        * DESTROYS: acc A
                        * Returns to monitor if not HEX input

C01E 8D F0      INHEX   BSR     INCH      GET A CHAR
C020 81 30              CMP A   #'0       ZERO
C022 2B 11              BMI     HEXERR    NOT HEX
C024 81 39              CMP A   #'9       NINE
C026 2F 0A              BLE     HEXRTS    GOOD HEX
C028 81 41              CMP A   #'A
C02A 2B 09              BMI     HEXERR    NOT HEX
C02C 81 46              CMP A   #'F
C02E 2E 05              BGT     HEXERR
C030 80 07              SUB A   #7        FIX A-F
C032 84 0F      HEXRTS  AND A   #$0F      CONVERT ASCII TO DIGIT
C034 39                 RTS

C035 7E C0 AF   HEXERR  JMP     CTRL      RETURN TO CONTROL LOOP
```

Assembly Language, by Michael Holley, From Wikimedia

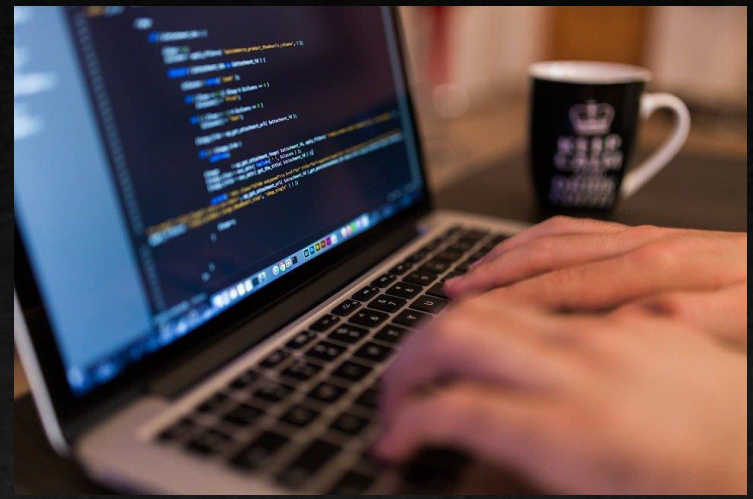
Project No. 2019-1-RO01-KA202-063965
This project has been funded with support from the European Commission. The content reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Co-funded by the Erasmus+ Programme of the European Union

Programming a Microcontroller

Typically, the process can be broken down into 4 simple steps;

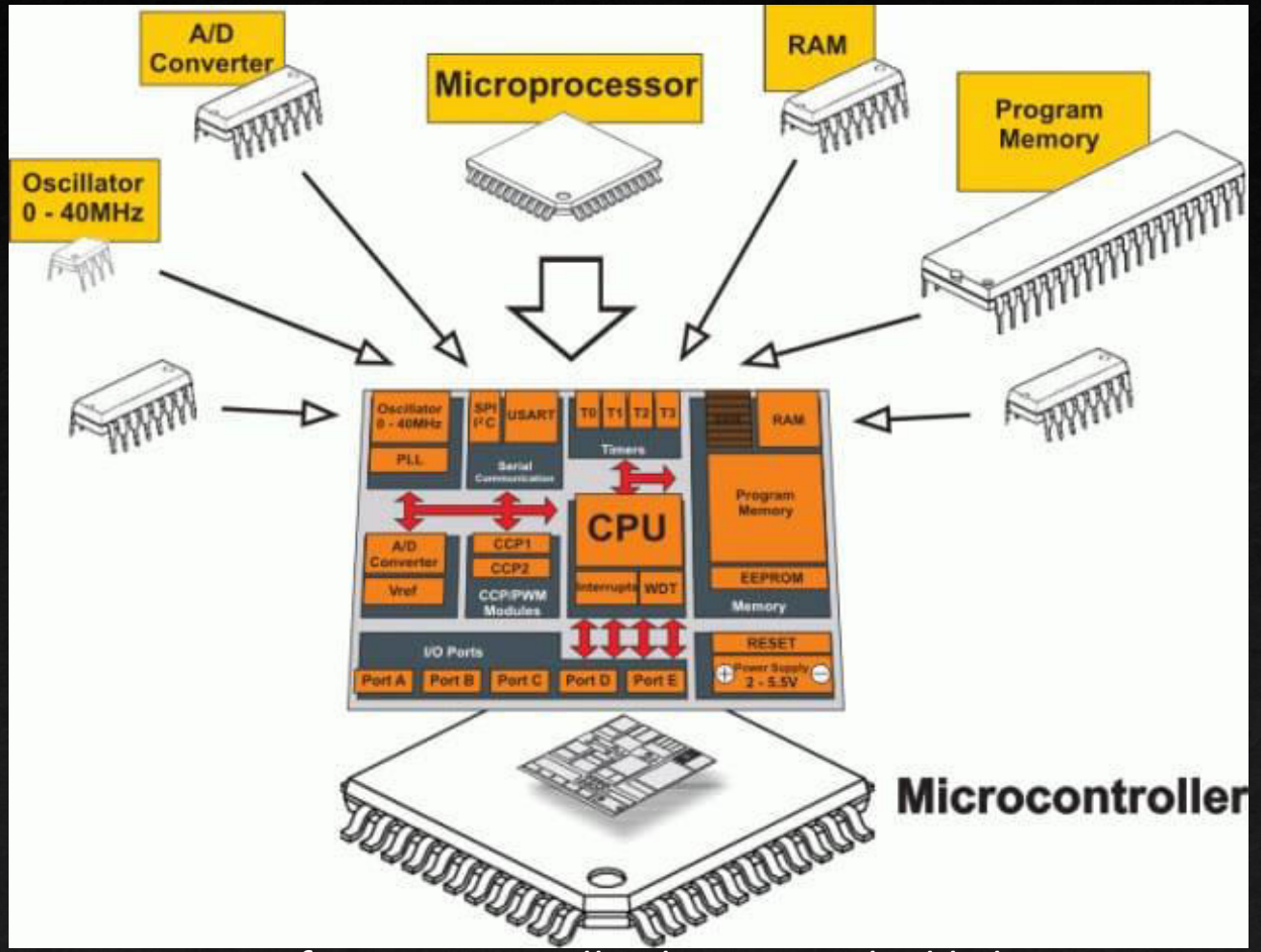
1. Write the program code onto a computer
2. Compile the code for the microcontroller that you are using
3. Connect your microcontroller to your computer
4. Upload the compiled version of the program to your microcontroller (saved onto the Program Memory)



Project No. 2019-1-RO01-KA202-063965
This project has been funded with support from the European Commission. The content reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Co-funded by the Erasmus+ Programme of the European Union

Internal Components of a Microcontroller



Parts of a microcontroller, by Max Embedded,
https://www.arrow.com/en/research-and-events/articles/engineering-basics-what-is-a-microcontroller

Project No. 2019-1-RO01-KA202-063965
This project has been funded with support from the European Commission. The content reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

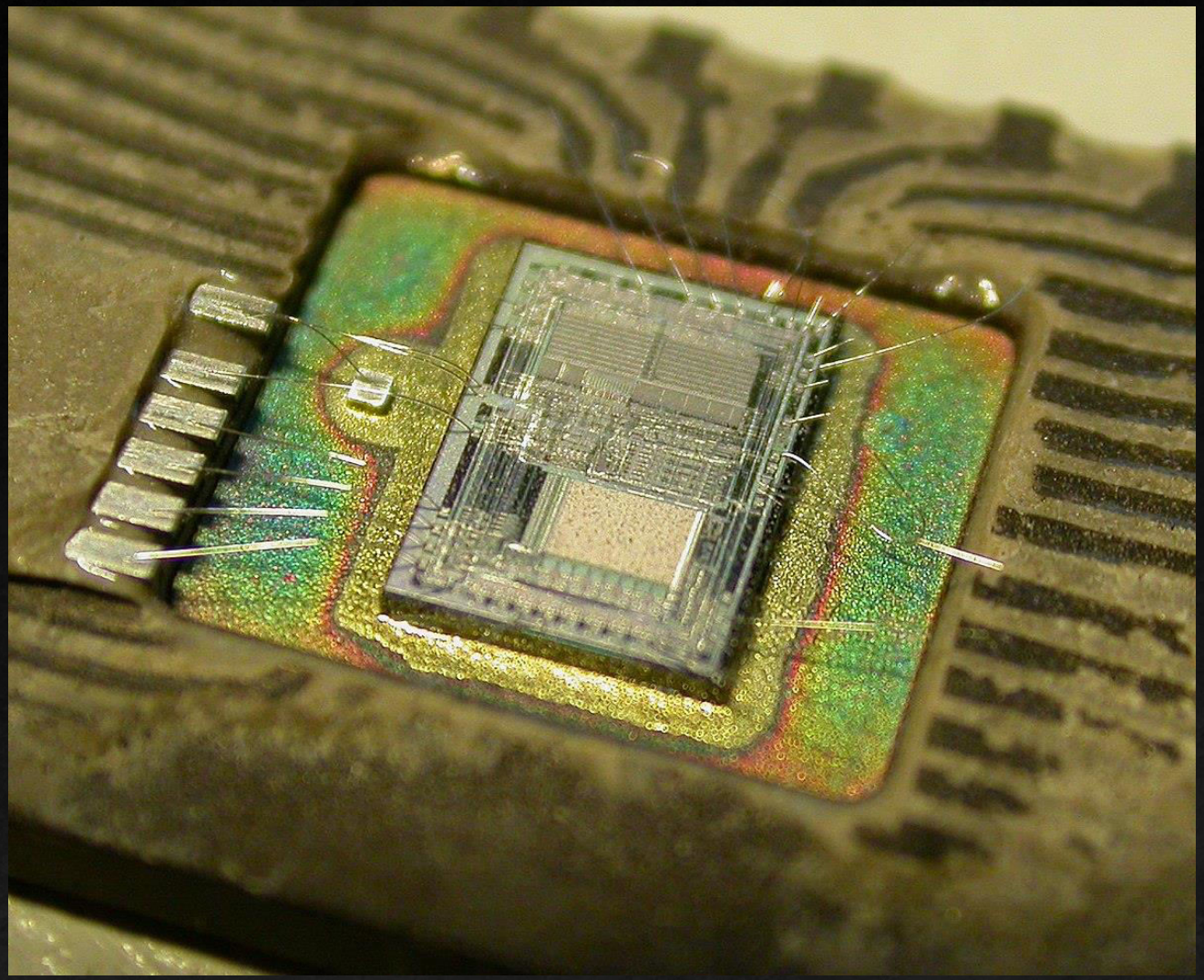
Co-funded by the Erasmus+ Programme of the European Union

Important Parts of a Microcontroller

- RAM: This is where the microcontroller stores its information while it is working. This is a depository where none of the information is stored if it is turned off (Volatile)
- CPU: Is the place where the computer executes the instructions given to it. The oscillator is used as an internal clock
- Programmable Memory: This is where the microcontroller stores the program that is programmed which is typically put in after manufacturing of the device. This type of memory needs to be a Non-Volatile since it contains the instructions onhow the device should operate which without it would seize to function completely.
- I/O Ports: This is where the microcontroller receives input and executes outputs. Typically these are Digital, i.e. either High or Low though these can also be Analog pins which can give a range of signals.

Project No. 2019-1-RO01-KA202-063965
This project has been funded with support from the European Commission. The content reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.
Co-funded by the Erasmus+ Programme of the European Union

Microcontroller Types



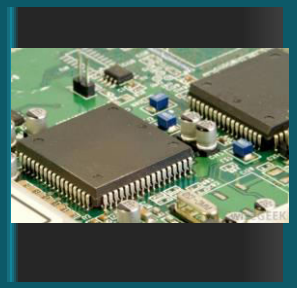
Microcontroller Die, by Unknown, Wikimedia

- There are tens of thousands of different microcontrollers nowadays in the market but the breakdown of these can be separated into three subcategories, which are;
  - 1. Embedded Microcontrollers
  - 2. 8 to 32 bit microcontrollers
  - 3. Digital Signal Processors
- These can be separated since these type of microcontrollers, although have similar functions, have different internal designs so that these can perform better in that respective field.

Project No. 2019-1-RO01-KA202-063965
This project has been funded with support from the European Commission. The content reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

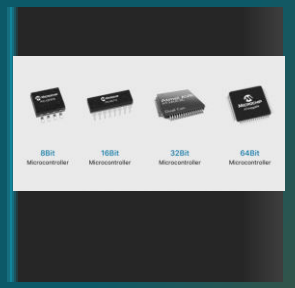
Co-funded by the Erasmus+ Programme of the European Union

Different Types of Microcontrollers



Embedded Microcontrollers

Digital Signal Processors



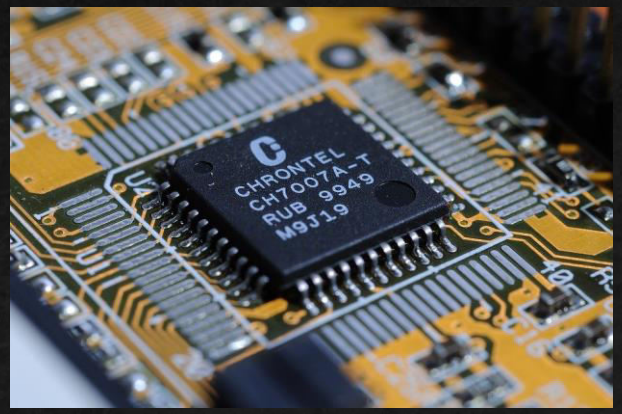
8 to 32 bit Microcontrollers

Project No. 2019-1-RO01-KA202-063965
This project has been funded with support from the European Commission. The content reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Co-funded by the Erasmus+ Programme of the European Union

Embedded Microcontrollers

- These are the most common type of microcontrollers which are common nowadays. These are designed for a specific function and would not work if implemented in a different scenario than what they were programmed for. Since the applications that these are used for are highly specific, therefore typically the costs for this type of microcontrollers is typically very low since these don't require much components to function. These are very useful for everyday objects like calculators, washing machines, ATMs, Fairy Lights, Remote Control, etc…
- Due to this a series of microcontrollers have been developed such as the 8051, PIC, STM32, etc…



Project No. 2019-1-RO01-KA202-063965
This project has been funded with support from the European Commission. The content reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Co-funded by the Erasmus+ Programme of the European Union

Digital Signal Processors

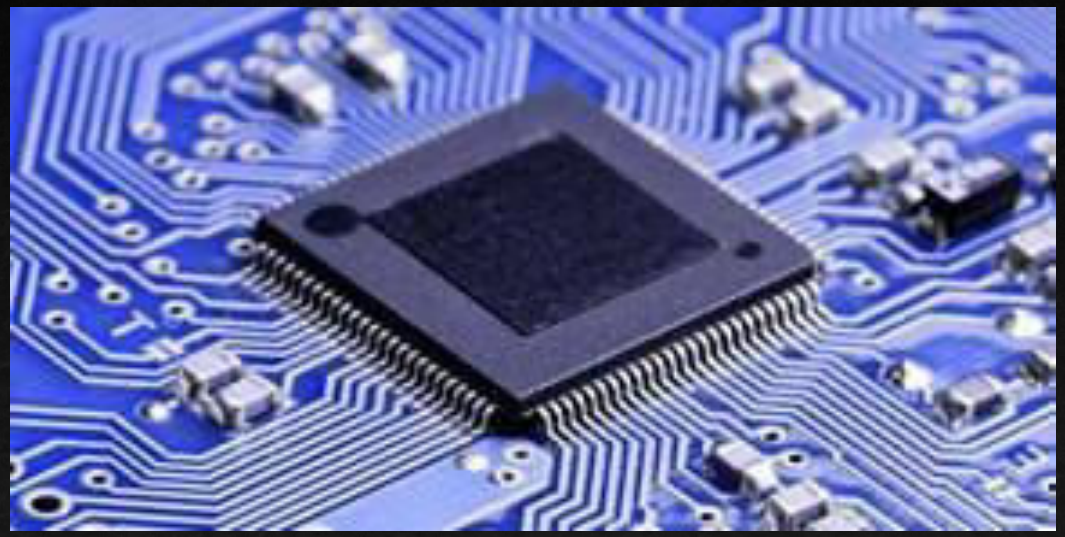
- DSP's are microchips which have been designed to optimize real-word signals into useful information i.e. applying mathematical formula to process the signal. Due to this, the design of the microchip is specific for improving the speed of processing the signal making it react faster to the input of a signal when compared to any other typical microcontroller.
- The most typical operations done by DSP's are typically "Minus", "Plus", "Multiply" and "Divide". This is extremely useful since with it Monitors, Microphones, Modems, etc.. can operate much faster than before.

Project No. 2019-1-RO01-KA202-063965
This project has been funded with support from the European Commission. The content reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Co-funded by the Erasmus+ Programme of the European Union

8 to 32 Bit Microcontrollers

- These Microcontrollers are typically designed to operate in a multitude of different situations, making them useful for applications that need to be versatile. These typically contain within themselves all of the required components needed to function i.e. the Ram, EEPROM, etc…
- Since costs have come down, the application for this type of microcontroller has increased dramatically, and due to their versatility and ease of use, they have entered most of our everday lives. E.g. Smartwatches, Robots, PLC's, etc…
- Further Reading:



Project No. 2019-1-RO01-KA202-063965
This project has been funded with support from the European Commission. The content reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Co-funded by the Erasmus+ Programme of the European Union

Considerations for choosing a Microcontroller

- There are five main components which are looked at when designing Microcontroller. These are;
  1. CPU, which will determine how fast the microcontroller will execute the functions
  2. I/O, which will determine how many components it can handle
  3. Memory, which will determine how complex the function it will be performing
  4. Special Functions, where any other necessary components are needed for it to function like Interruprs, Timers, etc….
  5. Physical Dimensions, which will determine how big it will be but also all of the other 4 components mentioned beforehand

Project No. 2019-1-RO01-KA202-063965
This project has been funded with support from the European Commission. The content reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.
Co-funded by the Erasmus+ Programme of the European Union

Off the Shelf Microcontrollers

Most Popular Microcontrollers, by The Engineering Projects,
https://www.theengineeringprojects.com/2018/03/introduction-to-microcontrollers.html

Project No. 2019-1-RO01-KA202-063965
This project has been funded with support from the European Commission. The content reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Co-funded by the Erasmus+ Programme of the European Union

Understanding Microcontrollers

Topic Summary

This slide concludes upon Understanding Microcontrollers

So far you should have learned something about the following topics:

1. What Microcontrollers are
2. A general idea about how they function
3. The different applications that they are designed for
4. Considerations when choosing a microcontroller

Project No. 2019-1-RO01-KA202-063965
This project has been funded with support from the European Commission. The content reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Co-funded by the Erasmus+ Programme of the European Union