**Microcontrollori a 8/32 bit:unità MCU (micro controller unit) per il tuo progetto PCB(Printed Circuit Board)**

In opposizione ai microprocessori e processori quali CPU o unità di elaborazione primaria con varie core (physical or virtual), differenti Clock, tempi di risposta e varie architetture usate in base a se sono per computer, workstation, server, smartphone, smartTV, smartwatches e altri sistemi come Alexa, Router wifi.

Esistono Microcontrollori MCU = Micro Controller Unit che sostanzialmente eseguono il lavoro che eseguirebbe un processore ma con dati specifici relativo a un sistema embedded.

Un sistema embedded

Per un sistema embedded quello che devo fare consiste nel realizzare il MCU e aggiungerlo in un PCB (Printed circuit Board) che può essere messo a paragone con la motherboard di un PC la quale contiene il processore e tutti i “hardware aggiuntivi” quali periferiche di I/O con usb, porta Ethernet, Bluetooth, WI-FI, BUS che collegano RAM e Memoria centrale come SSD o Hard DISK, relative ventole, porte VGA, porta HDMI, BUS che collega lettore CD/DVD e altri elementi in base all’ architettura usata.

For more differences between microprocessor and microcontroller reference:

<https://it.emcelettronica.com/differenze-tra-microprocessore-e-microcontrollore>

Per la realizzazione di questo PCB è stata usata un’interfaccia moderna e potente chiamata Altium Designer R .

Un interfaccia che permette la realizzazione e visualizzazione di ogni processo per la creazione di PCB (progettazione Elettronica) 🡺 quindi Hardware.

Nella PCB sostanzialmente sono io a decidere la disposizione di ogni elemento di elettronica visti nel corso di ELETTRONICA DIGITALE di Selmi. La parte digitale maggiormente di questo corso in cui…

(aggiunta lista elementi usati condensatore, Induttore , circuiti RLC, disposizioni serie e parallelo, diodi zener,..., CMOS, n-mos, p-mos, MOSFET, Analisi Circuitale, Potenza, Guadagno, Corrente Alternata non diretta (solo abitazioni) …. )

Sono quindi già definiti i requisiti per cui lavoreranno e la mole di lavoro che eseguiranno in tale PCB e relativo MCU. Inoltre si sa già a priori la quantità di dati che elaboreranno.

MCU unità di elaborazione primaria per una PCB (vedi anche modello 3D)

Microcontrolles are mostly pre-configured and on MPLAB R, hyou have already built boards and can buy Development Boards to test things or use them as controllers than add the with relative protection and after study where needed for the machinery or device.

E in base alle esigenze si possono avere a 8/16/32.. bit altre info da tenere a mente sono la memoria la velocità di risposta …

Sono quindi Microprocessori specializzati (Special purpouse) al contrario dei processori che montano computer (intel Core i-x con 1/2<x<9 ultima generazione, AMD Ryzen… ) che invece sono general purpouse e in base alle prestazioni possono essere usati per giochi, lavoro, server, playstation, cluster…

Tuttavia vi sono board come quella che ho usato io per avere prima familiarità con il sw scritto in C nel PIC/SP un connettore pin che funge da so per il board connesso nel DIP della board.

Board name: EasyPIC R V7 Connectivity

Board type: highly advanced board for Development with a variety on inputs, memory RAM or other type, USB ports, VGA port, EXT ICD port that I will use for my firmware interactions, seven Segment DISPLAY, LCD, GLCD and most importantly mikroProg USB 2.0 programmer-debugger (ICD in circuit debugger)

Reperibilità Dati tecnici Board: www.mikroe.com

For desription and specification on used board visit and add 3d view of board:

Reference from site: <https://www.mikroe.com/easypic-v7>

What is EASYPIC V7:

EasyPIC v7 is the seventh generation of our famous PIC development board. It is the product of accumulated knowledge over the past 10 years, and it’s state of the art of design, functionality and quality. It is loved by beginners and hobbyists, as well as professionals, and used in education and development by thousands. Board comes with [PIC18F45K22](http://www.microchip.com/wwwproducts/en/en546239) device.

It’ s modules,….

**With two mikroBUS™ sockets** you can add dozens of new functionality to your board with minimum hardware adjustments. Collection of mikroBUS™ compatible Click Boards is growing rapidly, and we have designed them with great care, so you don’t have to make any complicated settings. Just plug your board in, and you are good to go.

* Add microbus sockets:

<https://www.mikroe.com/blog/add-mikrobus-to-your-design>

* Add click Boards(GPS, Bluetooth, memory EEPROM, FRAM):

<https://www.mikroe.com/click-boards>

* Add various module with various purposes from mikroe to be checked:

<https://www.mikroe.com/blog/march>

* Microcontroller Distributors and upgraders industries:

<https://www.mikroe.com/distributors#europe>

* Books to learn From mostly PIC in c:

<https://www.mikroe.com/ebooks>

Add specification …

LPC board is very advanced programming language for boards like **LPCXpresso** and specifically like the **LPC1769 Develop. Board**

Now let’ s see the language itself

Self-taught language and possibly do certificate with the help of sub-relator Daniele, Simone at TENNECO corp.

Specifics C language libraries, drivers, different functionalities for Seven Segment Display and LCD Display and GLCD Display.

Describe Microcontrollers … architecture … and diff from cpu … link yt: <https://www.youtube.com/channel/UCGb-H3VKYA_CASQ18e2y0Dw>

Started with using PIC16F1778 microchip for controller used for writing firmware in c for the board so set of operations opening and closing periferical I/O like Temperature sensor click Board and other Click Boards previously Listed.

In particular read BOOK **PIC Microcontrollers Programming in C :**

[**https://www.mikroe.com/ebooks/pic-microcontrollers-programming-in-c**](https://www.mikroe.com/ebooks/pic-microcontrollers-programming-in-c)

For the PIC18F… that has 14\*2=28 PIN with diff functions watching videos from : (Good explanation of everything).

<https://www.youtube.com/watch?v=4UQ2_GodgoE&list=PLMkXE_yS8mSyEWxnLfI53Aw-Vpo1CxiNA>

Writing first program in MPLAB X for A LED OF 7 SEGMENT display …

History and elements using

What is mikroICD™? (debugger)

The on-board mikroProg™ programmer supports mikroICD™ - a highly effective tool for a Real-Time debugging on hardware level. The mikroICD™ debugger enables you to execute your program on the host PIC microcontroller and view variable values, Special Function Registers (SFR), RAM, CODE and EEPROM memory along with the mikroICD™ code execution on hardware. Whether you are a beginner, or a professional, this powerful tool, with intuitive interface and convenient set of commands will enable you to track down bugs quickly. mikroICD™ is one of the fastest, and most reliable debugging tools on the market

* OSC1: Crystal Oscillator XDZ 8.000 che fa da clock a frequenza 8.000 = 8MHz; X1 Quartz-crystal with relative pin connected in RA6 & RA7; Works with noth 3.3 V and 5 V.
* In DIP48 Socket we have: PIC16F1778;
* Board EasyPic v7;
* MPLAB ICD 3 used as debugger:

Starting to write first project in C to say like hello world with led small.

For the OSc configuration check out : <https://microchipdeveloper.com/8bit:emr-configuration-bits> .

For PIC 16F1778 detail consult <http://ww1.microchip.com/downloads/en/DeviceDoc/40001819B.pdf> .

For this particular MCU we have:

We have 3 port registers with n pins, MCLR pin, VCC, 2 GND, 2 OSC1:

* Port A: 6 pins/registers (RAx);
* Port B: 8 pins/resgisters (RBx); 0<x<pins
* Port C: 4 pins/registers (RCx);
* MCLR : 1 pin for resetting (riavvio) of PIC ON or OFF only as input;
* VCC :
* GND :
* OSC1 :

We have the first resistor (one for each port) that gives us direction of its port input or output:

* TRISBx (x= A, B, C in our case) to write in all pins of particular port:
  + TRSIBx = 0x00 🡺 for output 0x00
  + TRSIBx = 0xff 🡺 for input 0xff

To change specific pin direction 🡺 TRISCbits.TRISCx = …

Now we can write or read particular data from such port register with second Resistor (one for each port):

* PORTx (x= A, B, C) to write in all pins of particular port:
  + PORTx = 0x00 🡺 logic value
  + PORTx = 0xff 🡺 logic value

To change specific pin direction 🡺 PORTbits.Rxn = … or PORTbits.PORTxn=...

Third resistor associated with PIC microcontrollers is latch register (one for each port), it decides to read or write data(we rather use port):

* LATx
  + LATx = 0x00 🡺 logic value
  + LATx = 0xff 🡺 logic value

Writing first program (firstpj)

We can program to interface switches, relay, buzzers as follow:

Ricorda statement #pragma config … substitutes #include, #define, #if!defined so basically it creates a static header file.

Da aggiungere al log 2:51 per todos.