

Sistemes encastats

(embedded systems)

d'escala petita



Jordi Binefa i Martínez

Responsable d'R+D+i



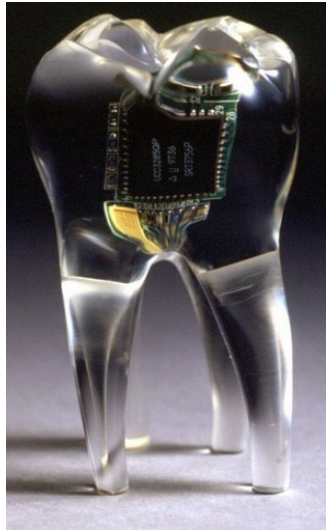
Professor de cicles formatius



Sistemes encastrats

(embedded systems)

d'escala petita



Jordi Binefa i Martínez

Responsable d'R+D+i

Professor de cicles formatius

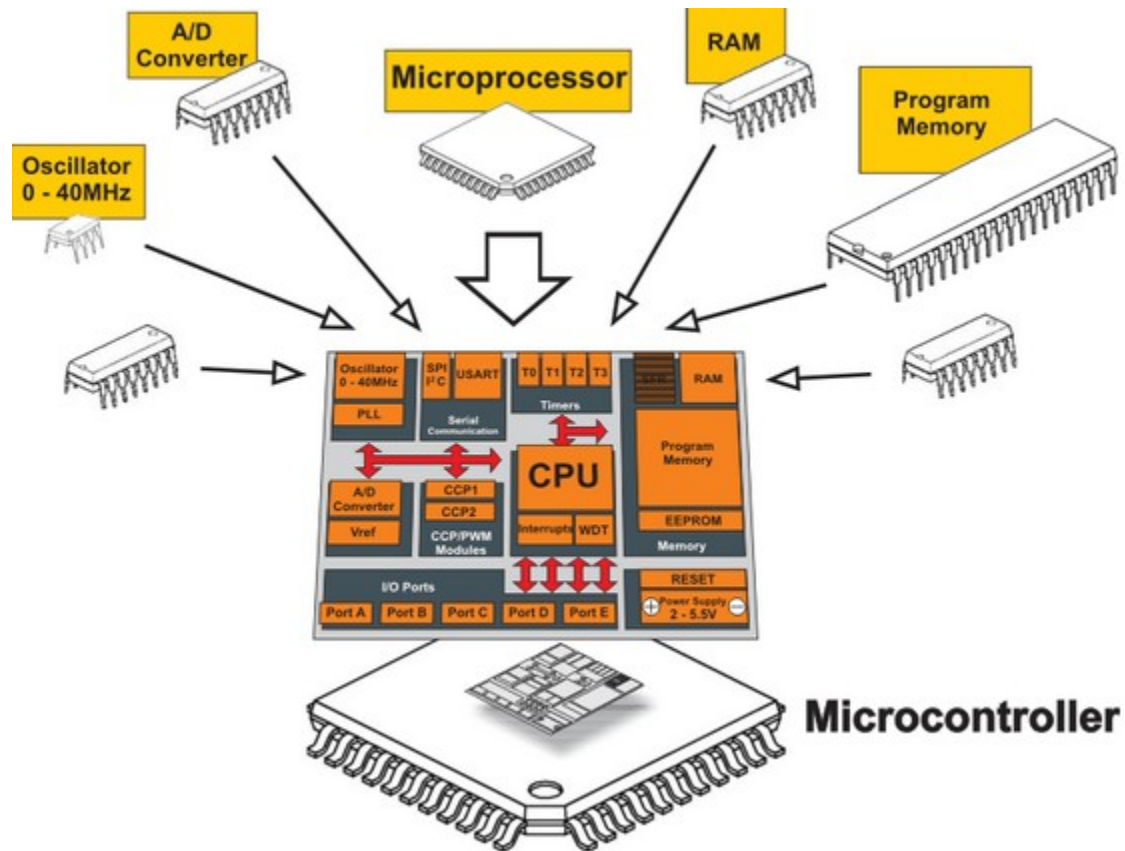


Microcontrolador (μ C, uC o MCU)³

És un petit ordinador en un sol circuit integrat especialitzat a controlar equips electrònics, i inclou una CPU, una petita quantitat de memòria, unitats d'E/S (Entrada/Sortida) i sol portar una memòria FRAM, NOR flash o una ROM per guardar el programa.

<https://ca.wikipedia.org/wiki/Microcontrolador>





Microcontrolador (μ C, uC o MCU)⁵

Els microcontroladors van ser dissenyats per a aplicacions per a sistemes encastrats en contraposició als microprocessadors utilitzats en els ordinadors personals per a aplicacions d'ús general.

<https://ca.wikipedia.org/wiki/Microcontrolador>



Sistema encastat

(embedded system)

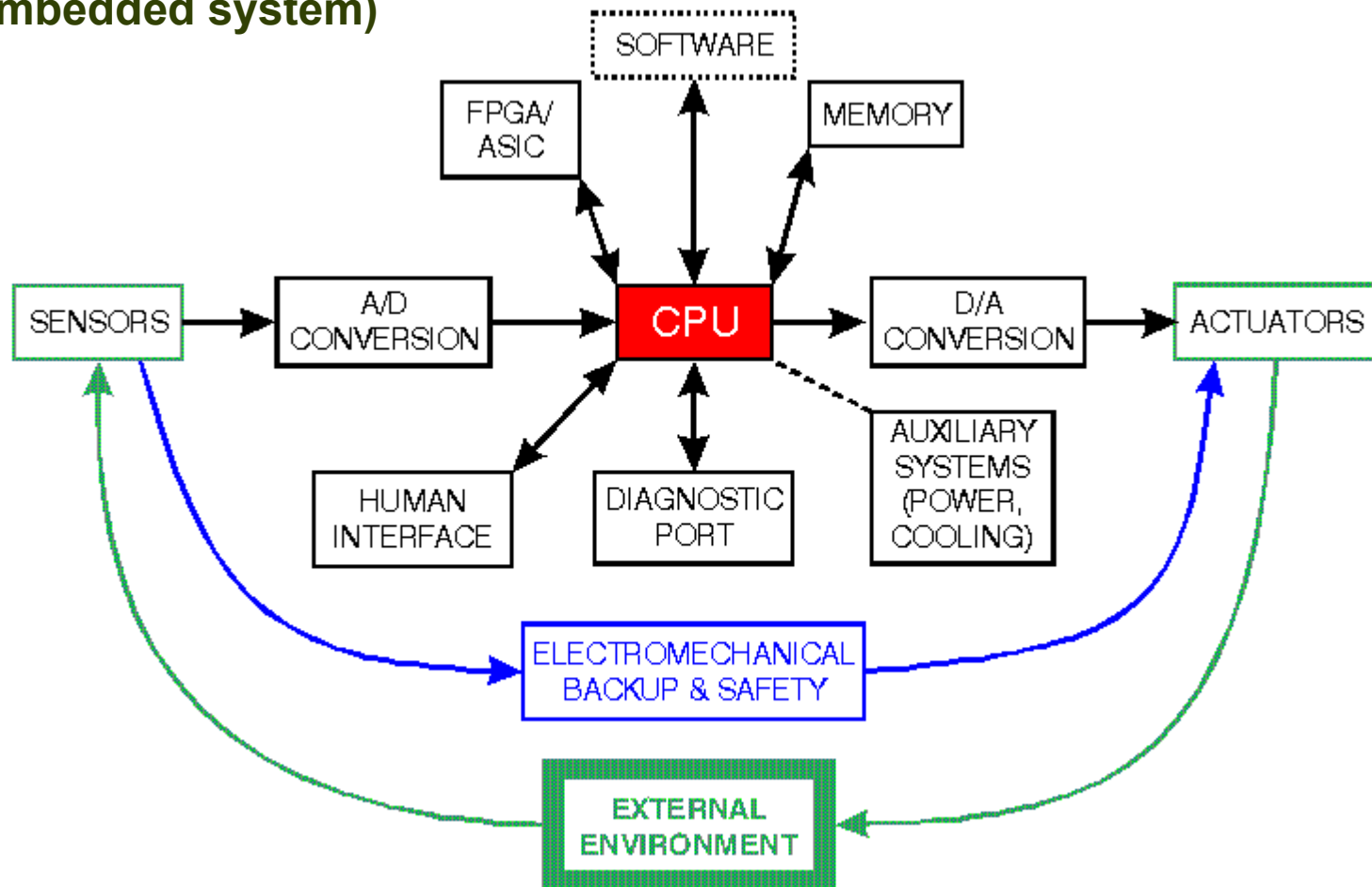
És un sistema electrònic dissenyat per realitzar una o poques funcions específiques normalment en un sistema de computació en temps real i que, és encapsulat totalment pel dispositiu que controla.

https://ca.wikipedia.org/wiki/Sistema_incrustat



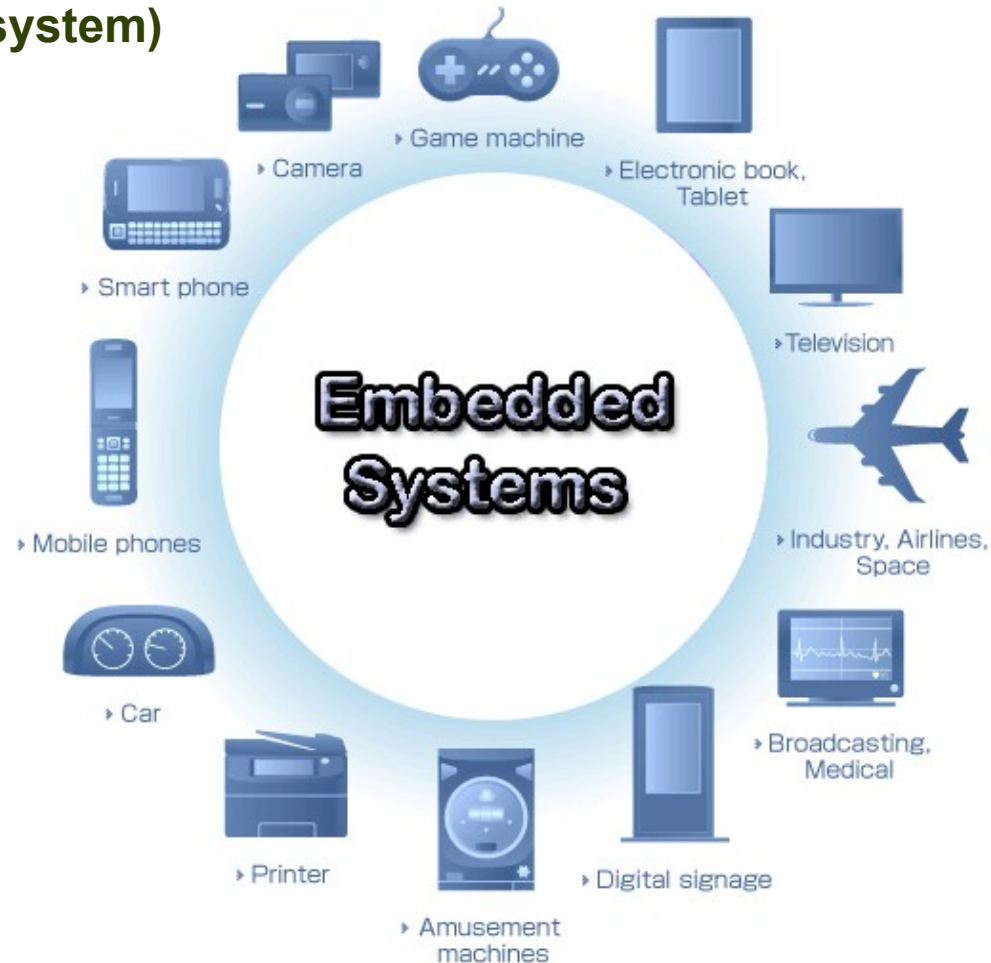
Sistema encastat

(embedded system)



Sistema encastat

(embedded system)



Sistema encastat

(embedded system)

A diferència d'un ordinador d'ús personal, que està dissenyat per tal de cobrir un ampli ventall de necessitats, els sistemes encastats cobreixen necessitats específiques.

https://ca.wikipedia.org/wiki/Sistema_incrustat

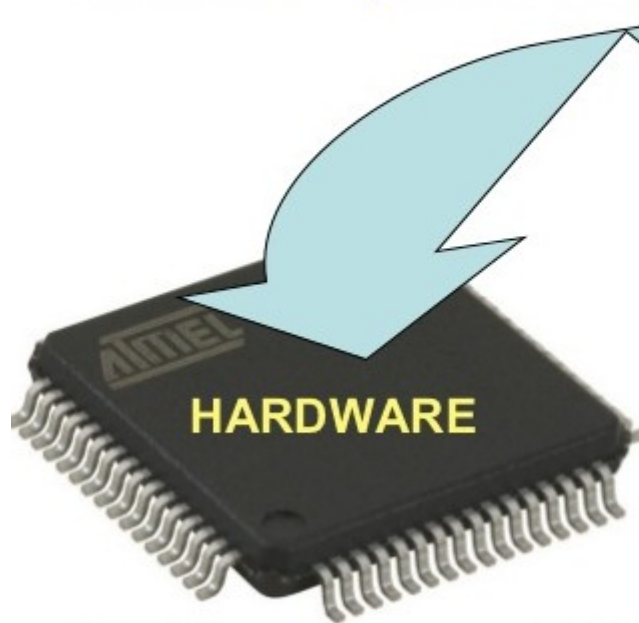


Sistema encastat

EMBEDDED SYSTEM

Definition: An Embedded System is one that has computer hardware with software embedded in it as one of its important components.

Its software embeds in ROM (Read Only Memory). It does not need secondary memories as in a computer



SOFTWARE PROGRAM

```
#include <16f876a.h>
#use delay (clock=2000000)
#byte PORTB=6
main()
{
  set_tris_b(0);
  portb=255;    //decimal
  delay_ms(1000);
  portb=0x55;   //hexadecimal
  delay_ms(1000);
  portb=0b10101010; //binary
  delay_ms(500);
}
```

03.01.09

murugan_m1@yahoo.com 996576

6



Programari

(software)

És el conjunt dels programes informàtics, procediments i documentació que fan alguna tasca a un ordinador. Comprèn el conjunt sistemàtic dels programes d'exploració i dels programes informàtics que serveixen per a aplicacions determinades.

<https://ca.wikipedia.org/wiki/Programari>



Maquinari

(hardware)

Components electrònics interconnectats que realitzen operacions lògiques i/o analògiques a la informació rebuda i emmagatzemada localment per produir una sortida, emmagatzemament o control de mecanismes actuadors.

https://en.wikipedia.org/wiki/Electronic_hardware



Microprogramari

(firmware)

És un bloc d'instruccions de programa per a propòsits específics, gravats en una memòria de tipus no volàtil (ROM, EEPROM, flash, etc.), que estableix la lògica de més baix nivell que controla els circuits electrònics d'un dispositiu de qualsevol tipus.

<https://ca.wikipedia.org/wiki/Microprogramari>



Microprogramari

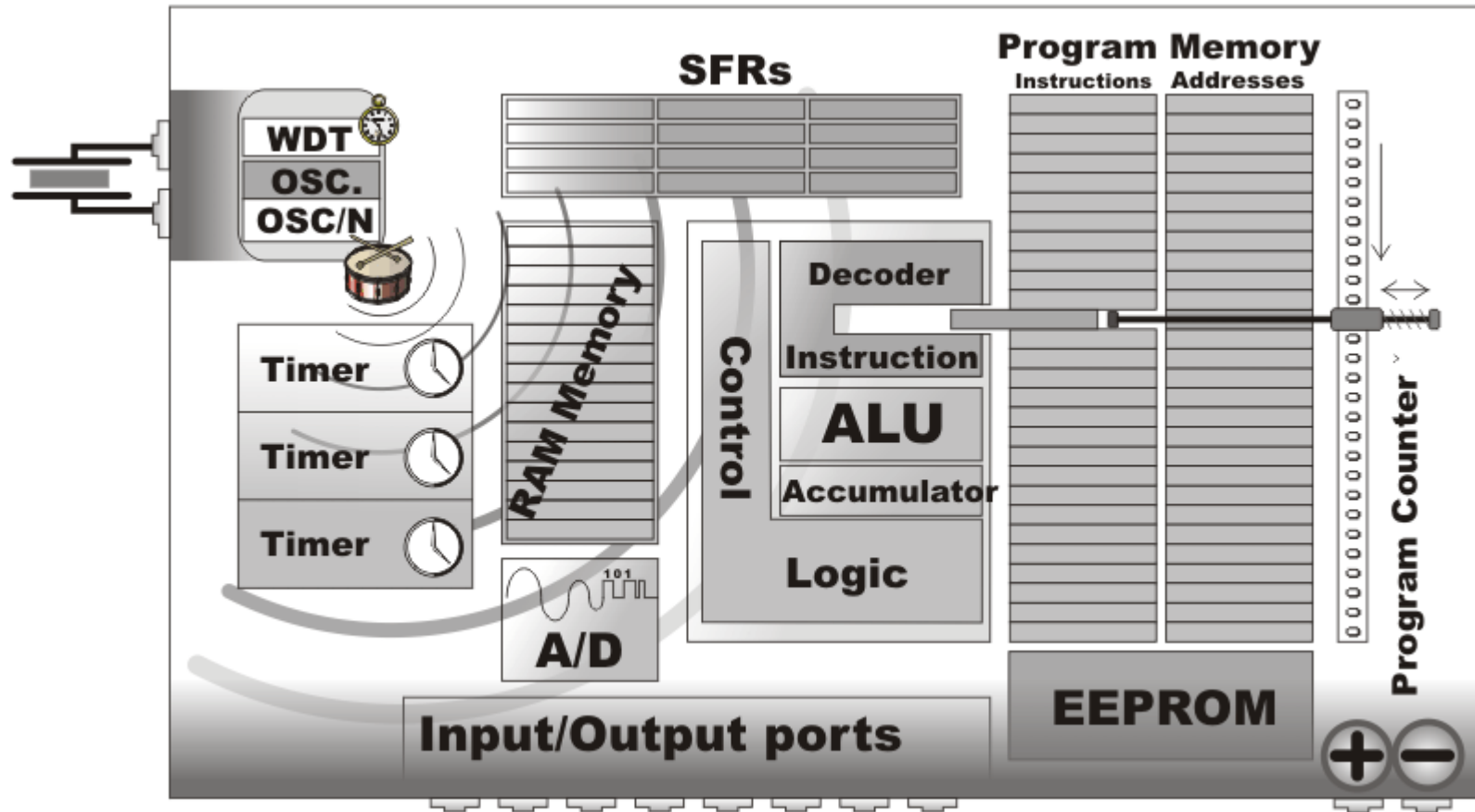
(firmware)

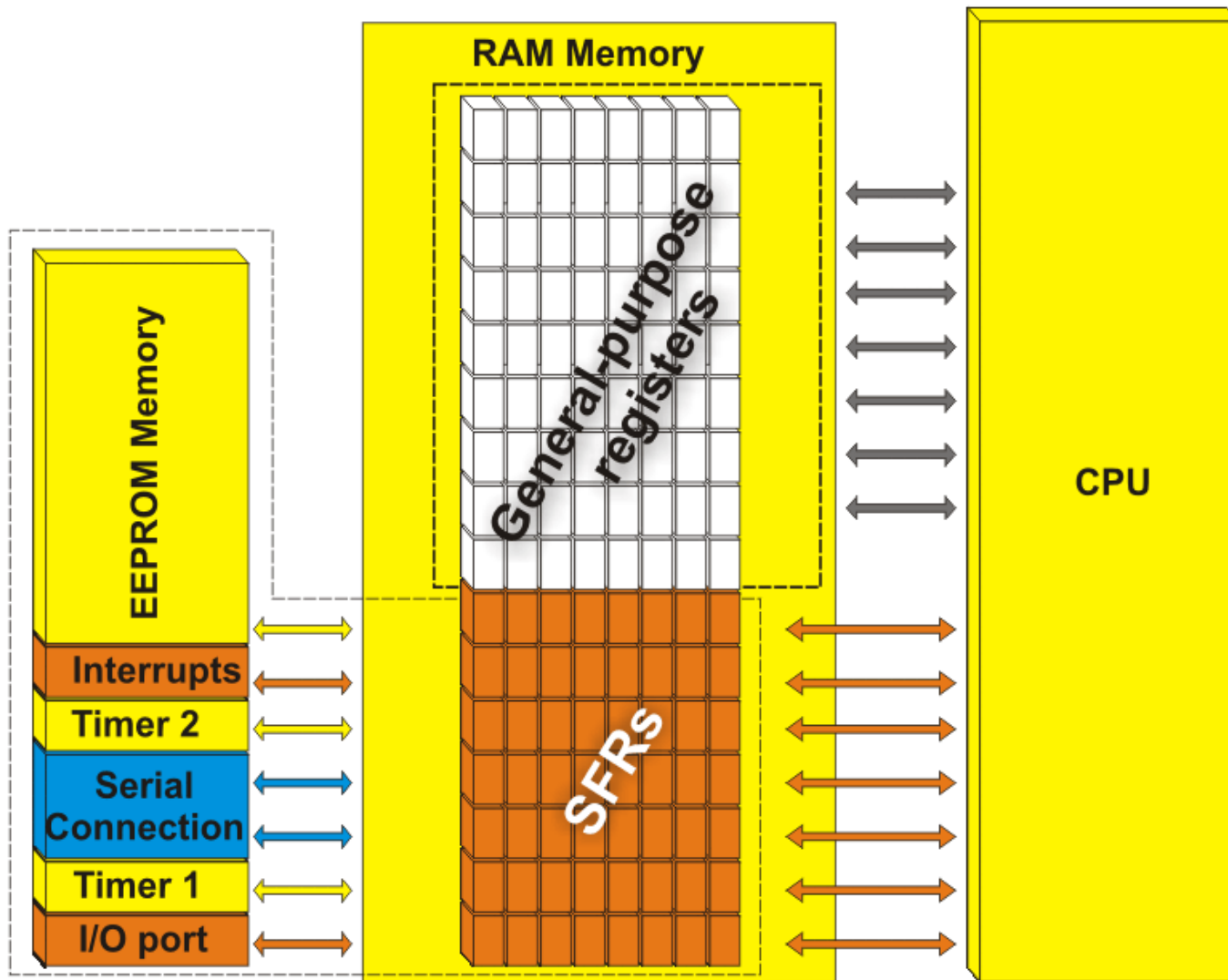
Com que està integrat a l'electrònica del dispositiu és en part maquinari, però també és programari, ja que proporciona lògica i es disposa en algun tipus de llenguatge de programació.

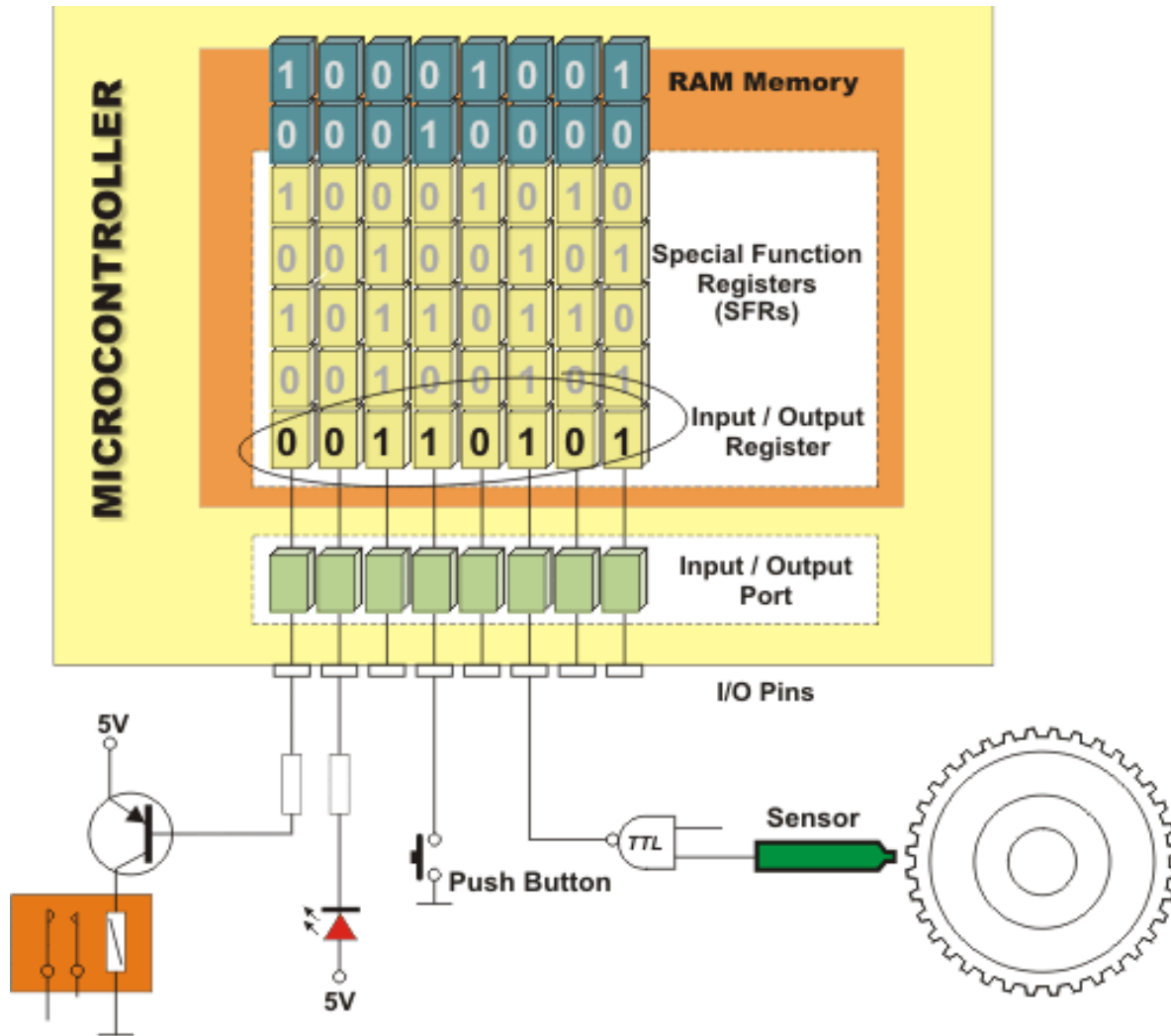
<https://ca.wikipedia.org/wiki/Microprogramari>

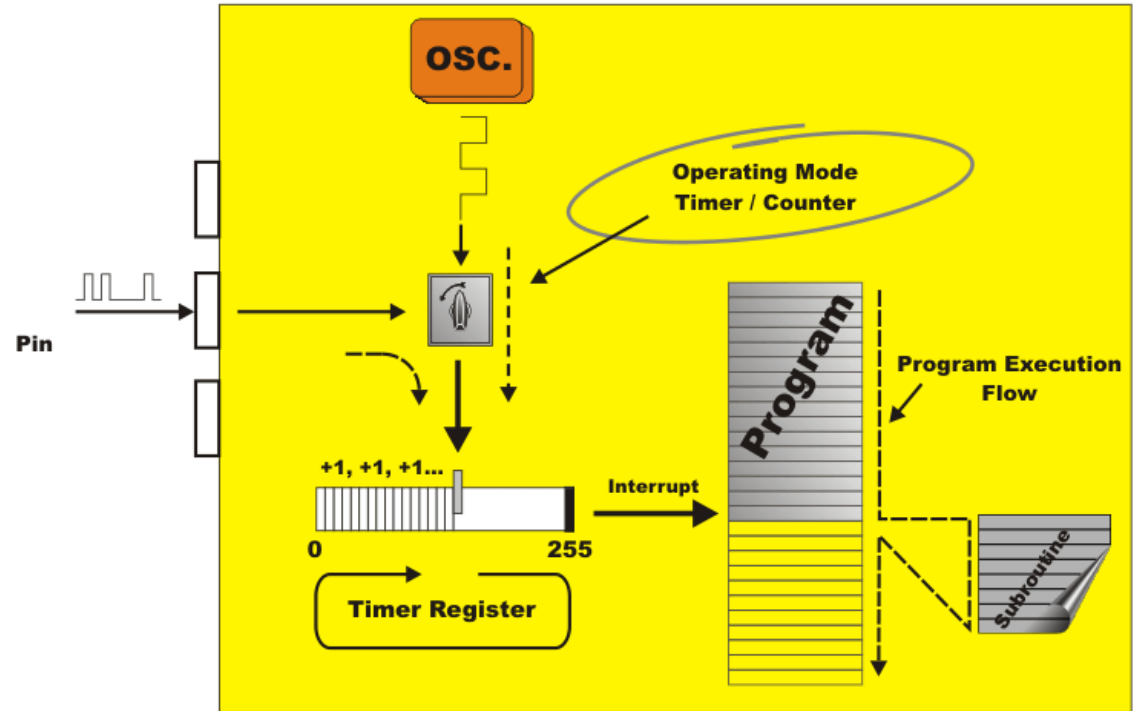
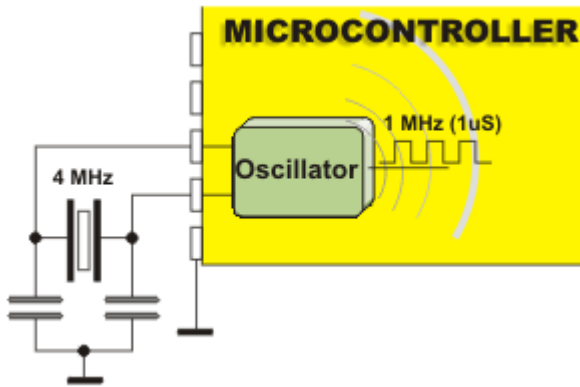


Com funciona un microcontrolador?

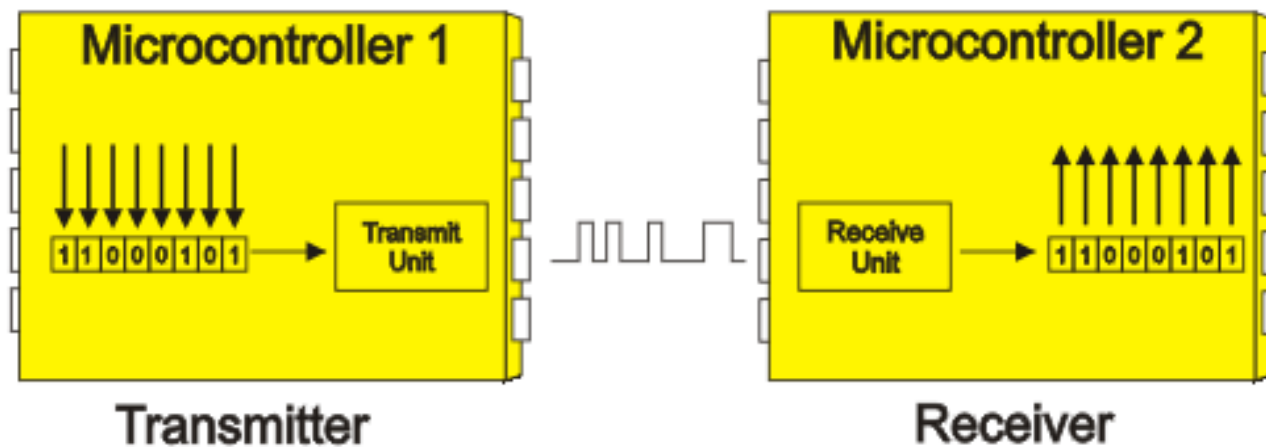




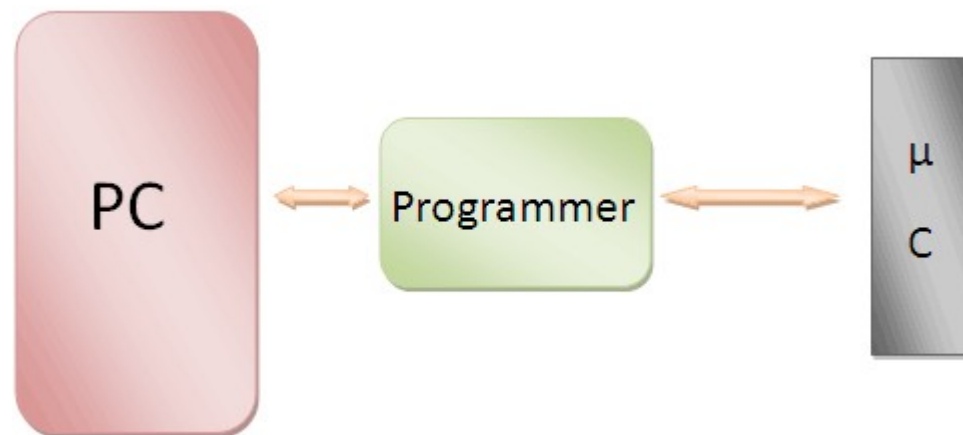




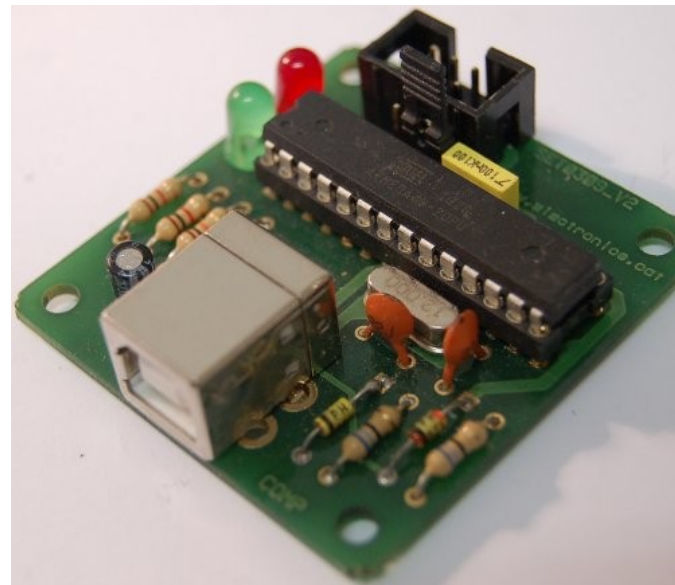
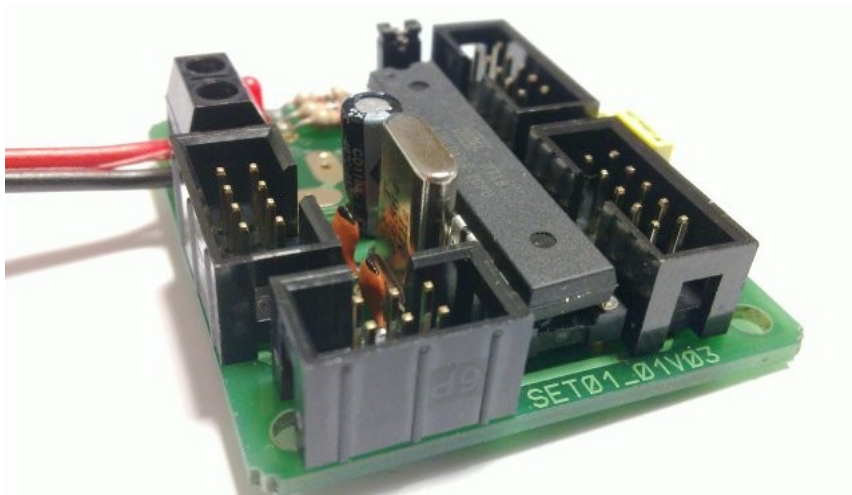
Comunicació entre microcontroladors

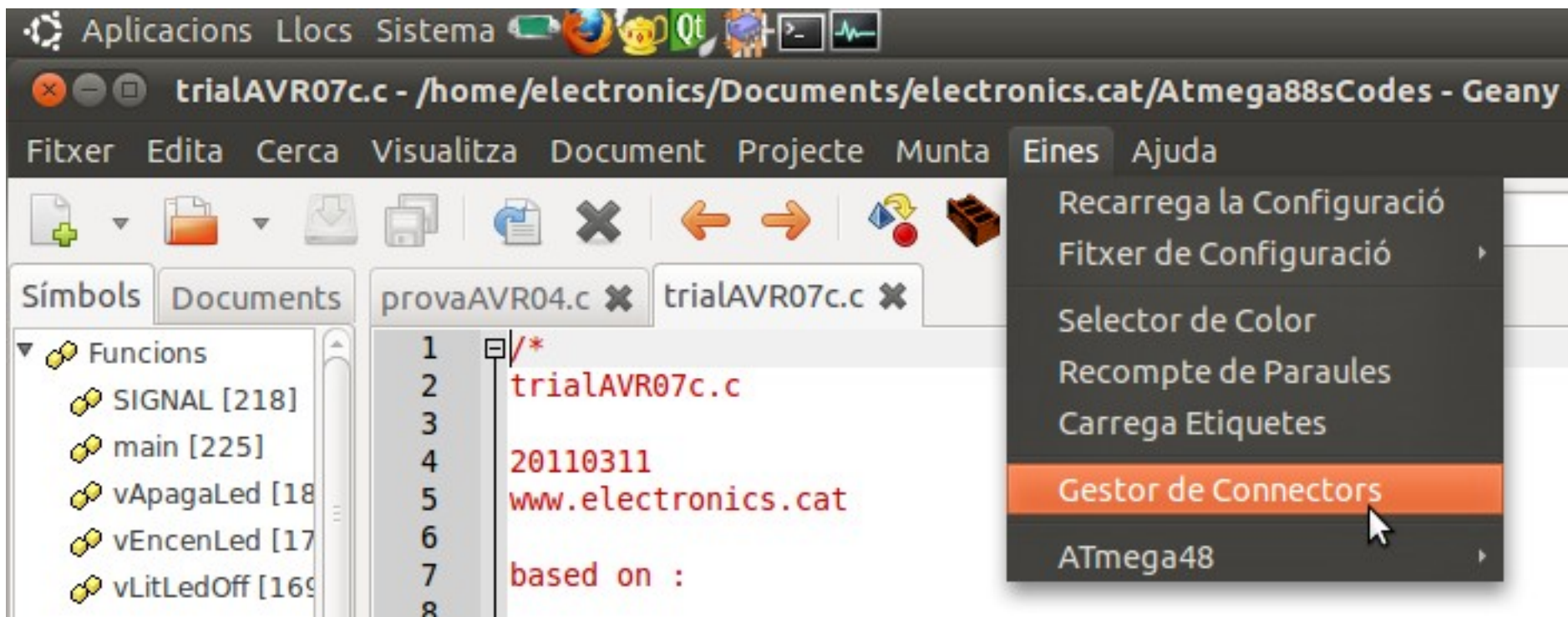


Com es programa un microcontrolador?



Com es programa un microcontrolador?





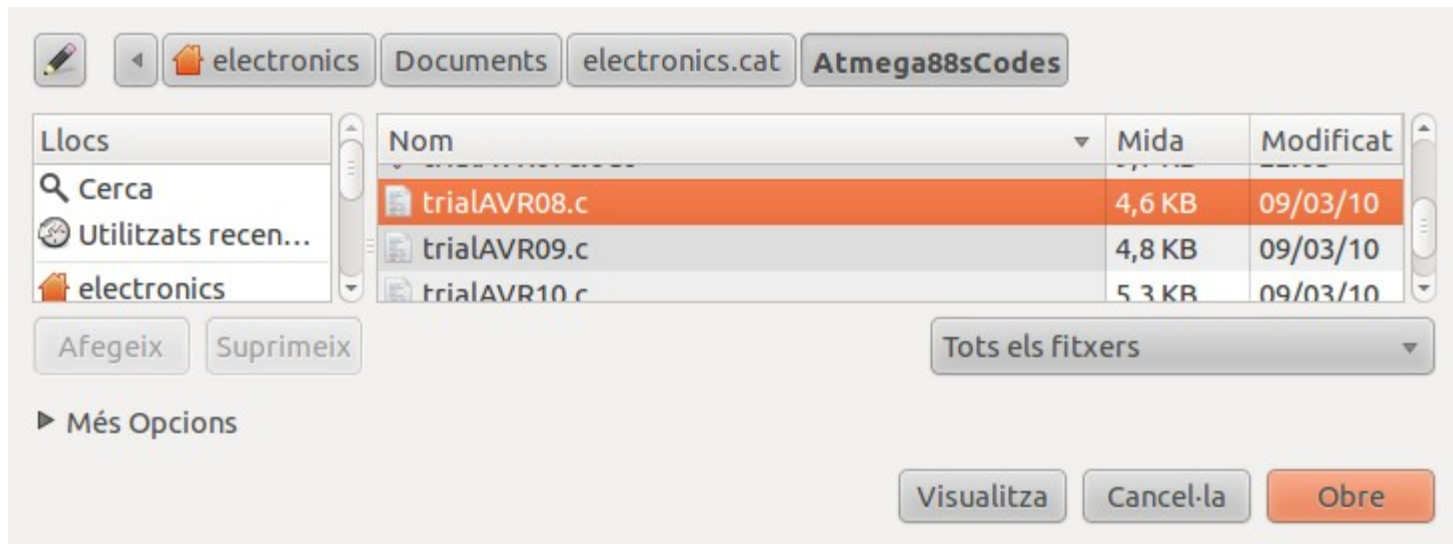
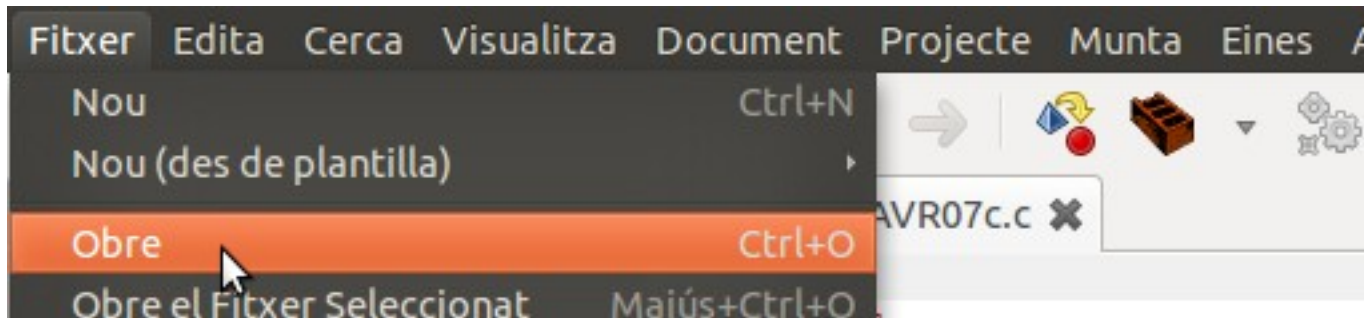
Trieu quins plugins carregar a l'inici:

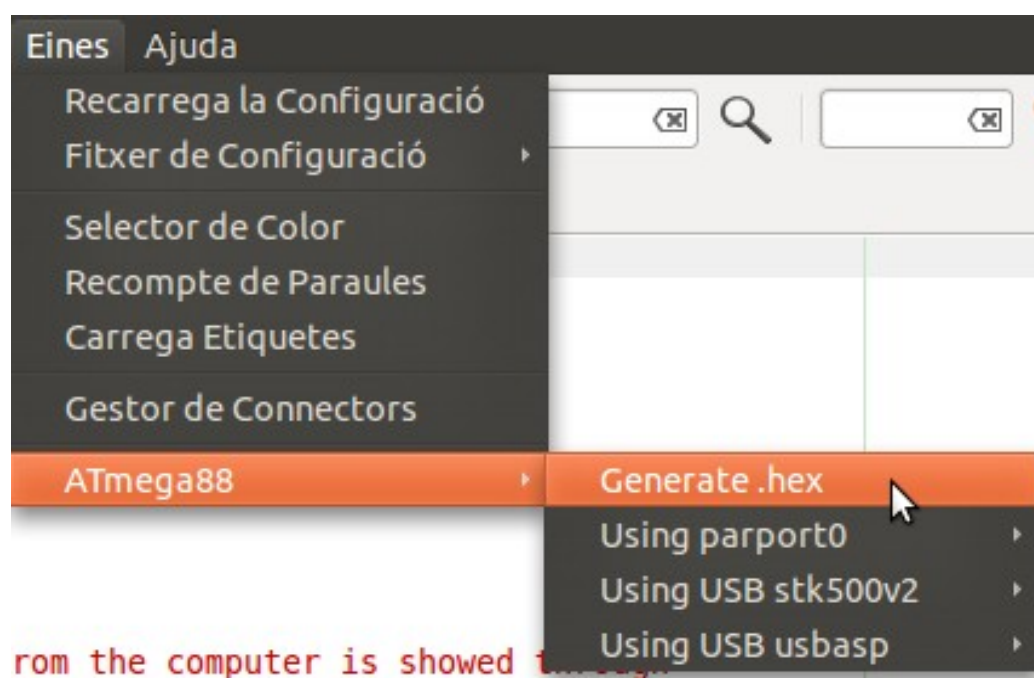
Actiu	Connector	Fitxer
<input type="checkbox"/>	Atmega48	/usr/lib/geany/geanyHex48.so
<input type="checkbox"/>	Atmega8	/usr/lib/geany/geanyHex8.so
<input type="checkbox"/>	Atmega8535	/usr/lib/geany/geanyHex8535.so
<input checked="" type="checkbox"/>	Atmega88	/usr/lib/geany/geanyHex88.so
<input type="checkbox"/>	Caràcters HTML	/usr/lib/geany/htmlchars.so
<input type="checkbox"/>	Constructor de Classes	/usr/lib/geany/classbuilder.so
<input type="checkbox"/>	Desa les Accions	/usr/lib/geany/saveactions.so
<input type="checkbox"/>	Divideix la Finestra	/usr/lib/geany/splitwindow.so
<input type="checkbox"/>	Explorador de Fitxers	/usr/lib/geany/filebrowser.so
<input type="checkbox"/>	Exporta	/usr/lib/geany/export.so

Detalls del connector: Ajuda Preferències

Connector:-Atmega88-0.0.14
 Descripció:-Geany's plugin to deal with Atmega88
 Autor(s):-Jordi Binefa <electronics.cat@gmail.com>

D'acord





En el fitxer inclòs des de /home/electronics/Documents/electronics.cat/Atmega88sCodes/trialAVR08.c:45:
 /usr/lib/gcc/avr/4.3.5/../../../../avr/include/avr/signal.h:36:2: avís: #warning "This header file is obsolete. Use <avr/interrupt.h>."

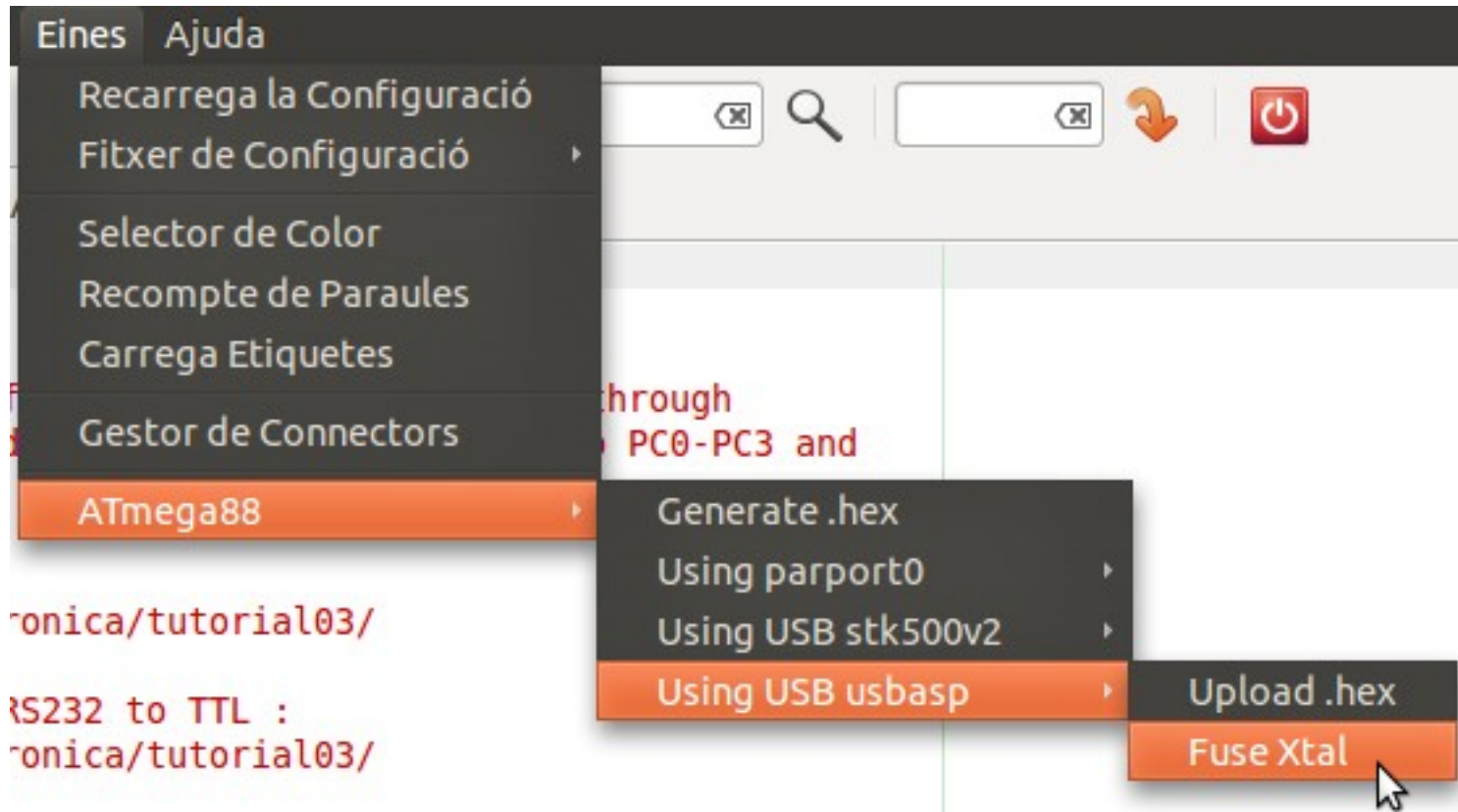
D'acord



File /home/electronics/Documents/electronics.cat/
 Atmega88sCodes/trialAVR08.hex has been created

D'acord

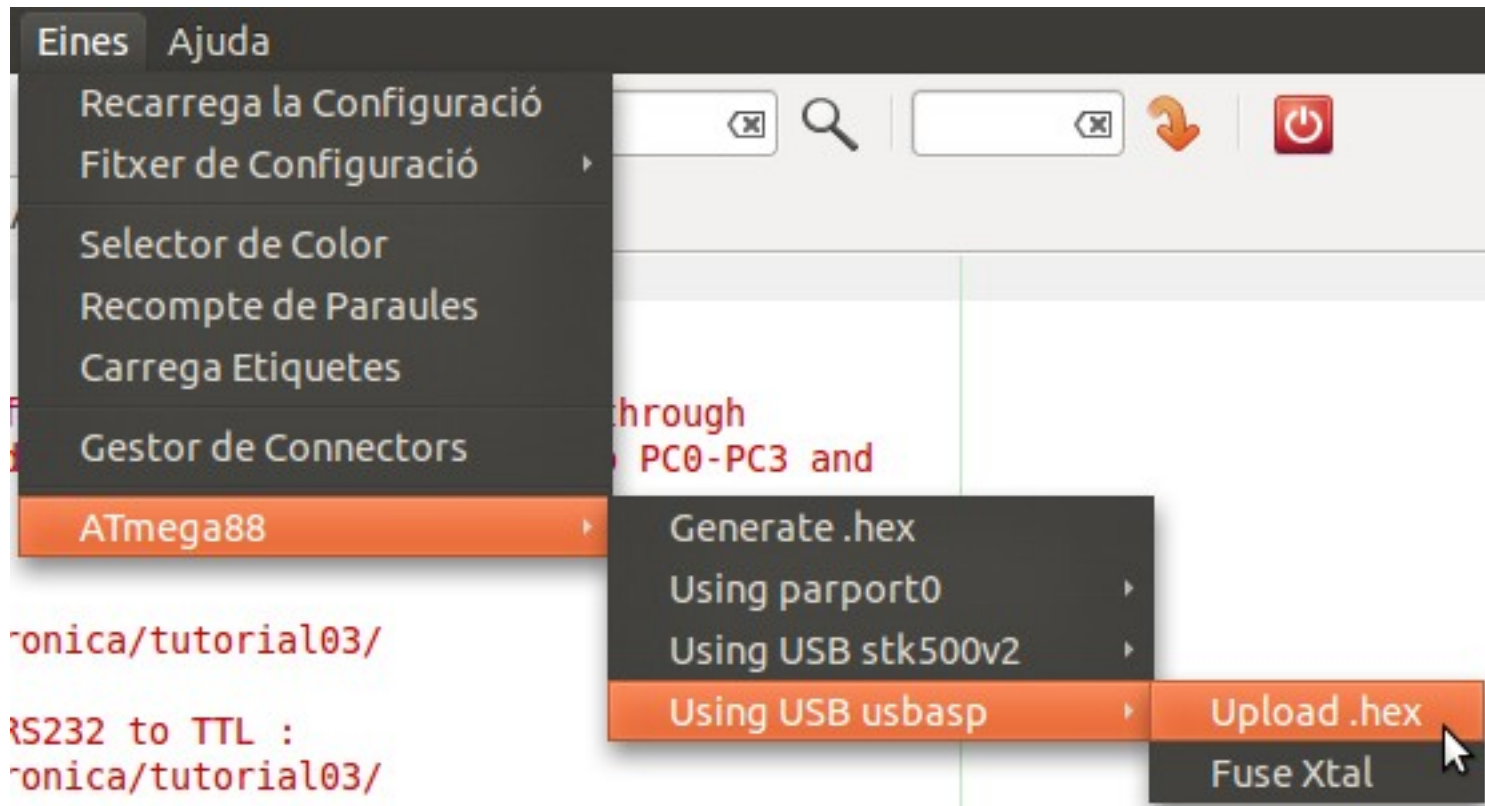




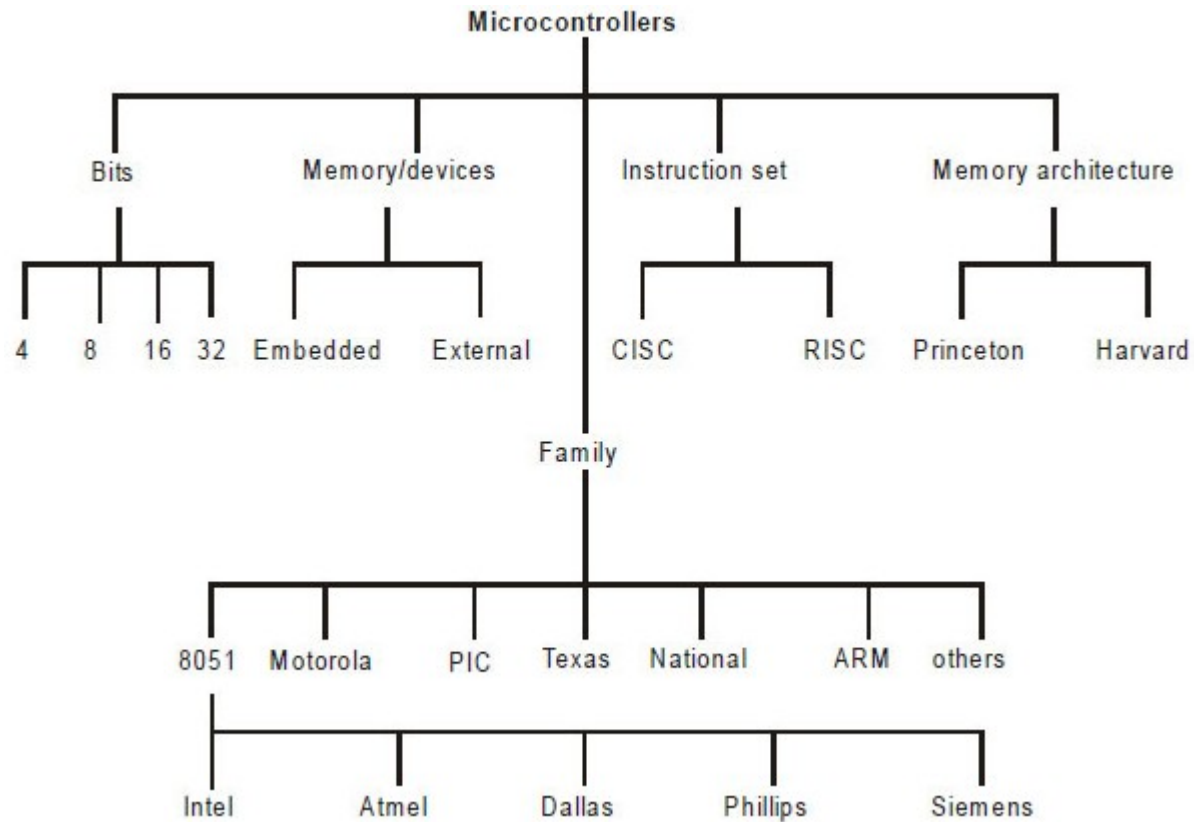
ronica/tutorial03/

RS232 to TTL :

ronica/tutorial03/



Tipus de microcontroladors



Conjunt d'instruccions

(Instruction set)

És una especificació que detalla les instruccions que una CPU d'un ordinador pot entendre i executar, o el conjunt de totes les ordres implementades per un disseny particular d'una CPU.

https://ca.wikipedia.org/wiki/Conjunt_d%27instruccions



Conjunt d'instruccions

(Instruction set)

El terme descriu els aspectes del processador generalment visibles a un programador, incloent-hi els tipus de dades nadiues, les instruccions, els registres, l'arquitectura de memòria i les interrupcions, entre altres aspectes.

https://ca.wikipedia.org/wiki/Conjunt_d%27instruccions



Conjunt d'instruccions RISC

RISC (Reduced Instruction Set Computer) és un tipus de microprocessador que reconeix un nombre típicament reduït d'instruccions de codi màquina. Basat en el model modern de Von Neumann.

https://ca.wikipedia.org/wiki/Conjunt_d%27instruccions



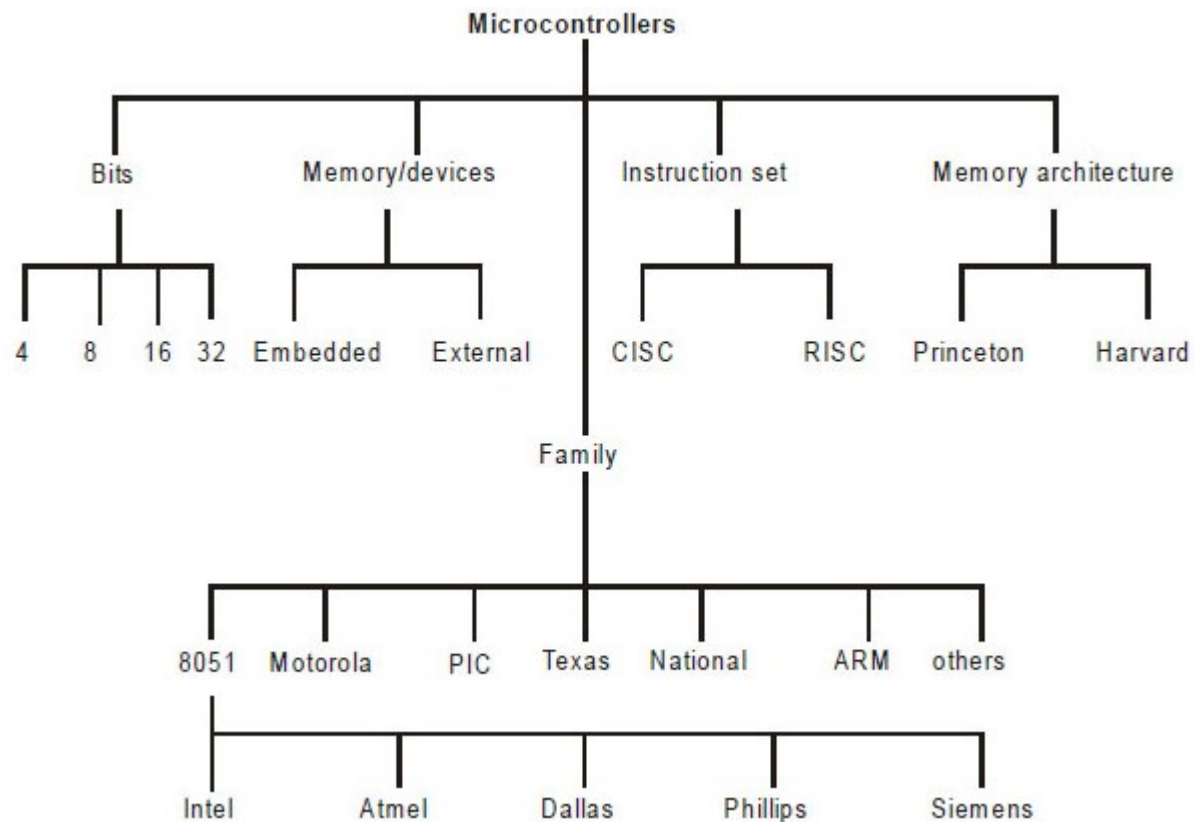
Conjunt d'instruccions CISC

CISC (de l'anglès Complex Instruction Set Computer) és un model d'arquitectura de computadors (PC).

CISC és una nomenclatura despectiva per part dels defensors/creadors processadors RISC

https://ca.wikipedia.org/wiki/Conjunt_d%27instruccions





Feature	ATmega328	ATtiny85	ATtiny84
Flash	32KB	8KB	8KB
SRAM	2048B	512B	512B
EEPROM	1024B	512B	512B
Pkg pins	28	8	14
Cost*	\$1.608	\$0.72	\$0.768
I/O pins	20 (or 23)	5 (or 6)	11 (or 12)
Analog	6	3 (or 4)	8

4 Bit Microcontrollers

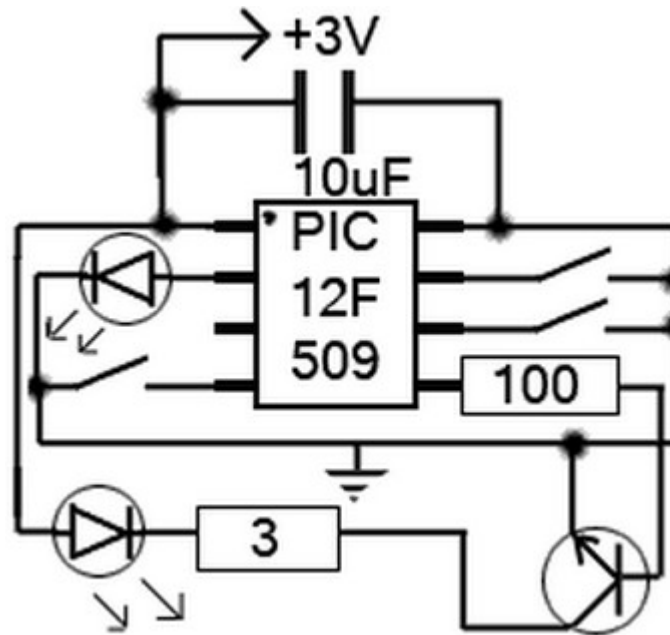
Model (Manufacturer)	I/O	Pins	RAM (bytes)	ROM (bytes)	Counters	Extra Features
COP400 Family (National)	23	28	64	1K	1	Serial bit I/O
HMCS40 (Hitachi)	10	28	32	512	-	10-bit ROM
TMS 1000 (Texas Instruments)	23	28	64	1K	-	LED display

8 Bit Microcontrollers

Model (Manufacturer)	I/O	Pins	RAM (bytes)	ROM (bytes)	Counters	Extra Features
8048 (Intel)	27	40	64	1K	1	8k External memory
8051 (Intel)	32	40	128	4K	2	128k External memory, Boolean processing, serial port
COP800 Family (National)	24	28	64	1K	1	Serial bit I/O, 8-channel A/D converter
6805 (Motorola)	20	28	64	1K	1	PLL frequency synthesizer,
68hc11 (Motorola)	40	52	256	8K	2	A/D, PWM generator, pulse accumulator
TMS370 (Texas)	55	68	256	4K	2	watchdog timer, Instruments) Serial ports, A/D (8 bit, 8 channel)
PIC (Micro Chip)	12	18	25	1K	0	small pin count, very low power consumption

Processor	16-bit RISC	32-bit ARM Cortex M3	16-bit H8/300H
Vendor	TI	ST	Renesas
Manufacturer part #	MSP430F2617TPMR	STM32F103CBT6	DF38099FP10V
Max speed	16 MHz	72 MHz	10 MHz
Flash memory	92 kbytes	128 kbytes	128 kbytes
RAM	8 kbytes	20 kbytes	4 kbytes
Speed & voltage	2.2 V @ 1 MHz	2.4 V @ 8 MHz	2.7 V @ 4 MHz
Active mode current	365 μ A	8000 μ A	3000 μ A
Sleep mode current	0.5 μ A	2.8 μ A	1.5 μ A
Wake-up time	1 μ s	1.8 μ s	Not specified
Package	64-LQFP	48-LQFP	100-LQFP
Dimensions (mm ²)	10 by 10 mm	7 by 7 mm	14 by 14 mm
IO Pins	48	37	75
On-chip ADC	12 bits, 8 channels	12 bits, 10 channels	10 bit, 8 channels
On-chip LCD driver	No	No	Yes
Compiler Vendor	IAR	Keil or IAR	Renesas
Compiler Cost	\$2500	Free (Keil)	Free

Name	Arduino Uno	Raspberry Pi	BeagleBone
Model Tested	R3	Model B	Rev A5
Price	\$29.95	\$35	\$89
Size	2.95"x2.10"	3.37"x2.125"	3.4"x2.1"
Processor	ATMega 328	ARM11	ARM Cortex-A8
Clock Speed	16MHz	700MHz	700MHz
RAM	2KB	256MB	256MB
Flash	32KB	(SD Card)	4GB(microSD)
EEPROM	1KB		
Input Voltage	7-12v	5v	5v
Min Power	42mA (.3W)	700mA (3.5W)	170mA (.85W)
Digital GPIO	14	8	66
Analog Input	6 10-bit	N/A	7 12-bit
PWM	6		8
TWI/I2C	2	1	2
SPI	1	1	1
UART	1	1	5
Dev IDE	Arduino Tool	IDLE, Scratch, Squeak/Linux	Python, Scratch, Squeak, Cloud9/Linux
Ethernet	N/A	10/100	10/100
USB Master	N/A	2 USB 2.0	1 USB 2.0
Video Out	N/A	HDMI, Composite	N/A
Audio Output	N/A	HDMI, Analog	Analog



Eines de disseny



Hardware Design Tools

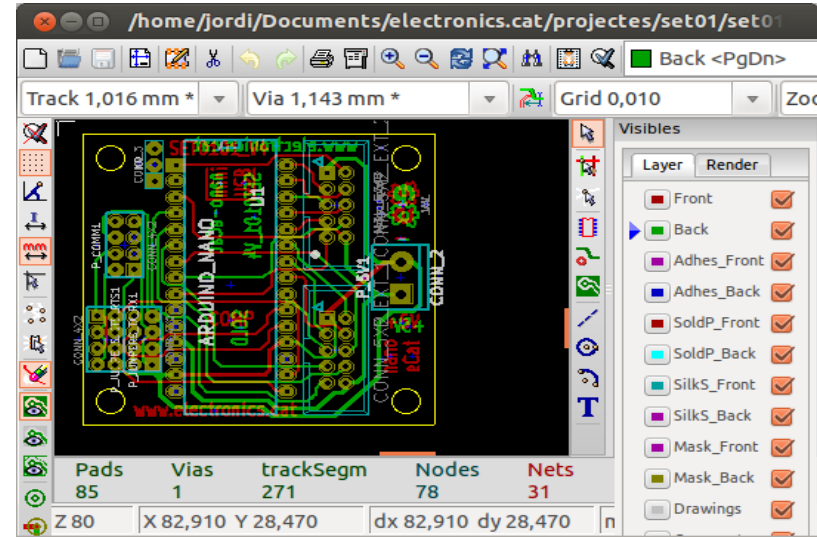
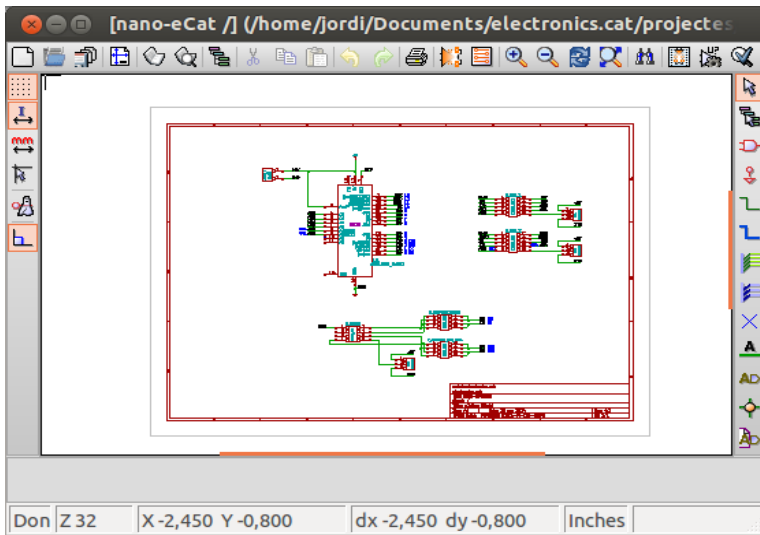
Electronic Design Automation (EDA)/Computer Aided Design (CAD)

Licensed

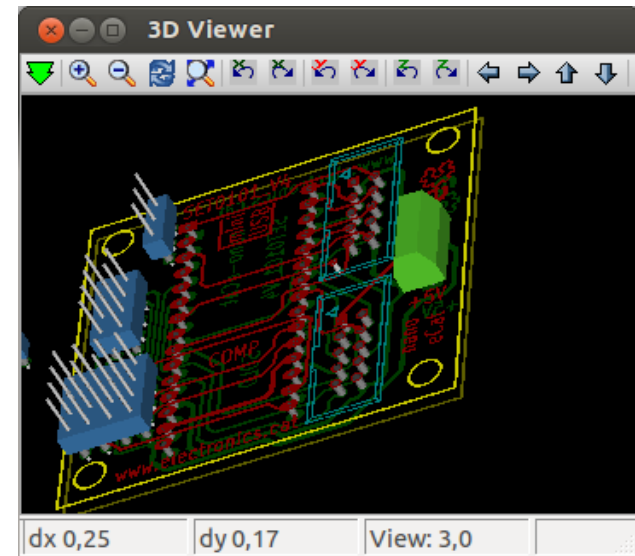
- Cadence OrCAD™
- Labcenter Proteus™
- CadSoft EAGLE
- Novarm DipTrace®
- ... and many more

Open Source

- KiCAD (win/linux/mac)
- gEDA (linux/mac)
- Fritzing (win/linux/mac)
- FreePCB (win)
- ... and a few more



KiCad



123D Circuits / Circuits Components Circuit Scribe

Shop Search Sign up Sign in

Simulate Arduino online and easily create custom circuit boards

Design and simulate circuit boards with our breadboard, schematic and PCB editor. Share your designs and collaborate as a team.

Get started designing electronics online!

Show me an example!

Draw Conductive Traces

Easily create and simulate Circuit Scribe sketches online.

<http://123d.circuits.io/>



A on fer prototipus?





Arxius de fabricació (gerbers)

Nom	Mida	Tipus
▼ gerber_v1_1	7 elements	carpeta
JXXIII.drl	3,6 kB	subtítols MPSub
JXXIII-B_Cu.pho	48,7 kB	document de text pla
JXXIII-B_Mask.pho	4,6 kB	document de text pla
JXXIII-Edge_Cuts.pho	749 bytes	document de text pla
JXXIII-F_Cu.pho	43,1 kB	document de text pla
JXXIII-F_Mask.pho	5,9 kB	document de text pla
JXXIII-F_Silks.pho	2,2 MB	document de text pla
▼ pdf_v1_1	7 elements	carpeta
JXXIII.drl	3,6 kB	subtítols MPSub
▪ JXXIII-B_Cu.pdf	23,2 kB	document PDF
▪ JXXIII-B_Mask.pdf	11,1 kB	document PDF
□ JXXIII-Edge_Cuts.pdf	2,8 kB	document PDF
▪ JXXIII-F_Cu.pdf	21,5 kB	document PDF
▪ JXXIII-F_Mask.pdf	11,8 kB	document PDF
▪ JXXIII-F_Silks.pdf	127,4 kB	document PDF



PCB Prototype the Easy Way
Full feature custom PCB prototype service.

Home My Account PCB Instant Quote PCB Assembly Orders List Coupons Delivery Address

Order Detail

Order Detail

OrderNO. :	W01310AS26	GerberFile :	[Download]W01310AS26_JXXIII_v11_ind.zip		
Length :	73 mm	Width :	74.7 mm	Quantity :	50
Layers :	2 layers	Thickness :	1.6 mm	Surface Finish :	HASL lead free
Finished Copper :	1 oz Cu	Solder Mask :	Red	Silkscreen :	White
CreateTime :	10/11/2015 3:22:06 AM	Delivery Type :	3-4 days	Delivery Date :	2015-10-14
Manufacturing :					
					Total: US \$53.00

[View Detail](#)

73 X 74.7 mm Layers: 2 , Material: FR-4 , Finished Copper: 1 oz Cu , S...



AH19DS DA2F

[OVERVIEW](#)[SPECIFICATIONS](#)[TECHNICAL SUPPORT](#)

Lift system with electrical engine activated through integrated push buttons in the cover plate, external GPI controls (contact closer) or AH-Net Protocol (RS422) through CAT 5 cable with loop-through and addressing. Embedded LINUX PC with special software for Signature Name, updated from external server by RJ45 network (TCP-IP).

[Download info](#)

<http://www.arthurholm.com/>



Programari allotjat a sourceforge.net :

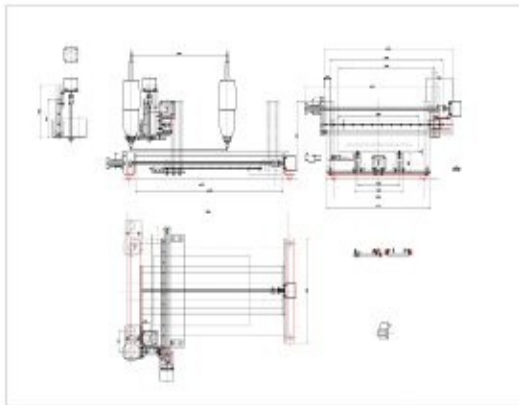
(<http://sourceforge.net/projects/qtnc/files/>)

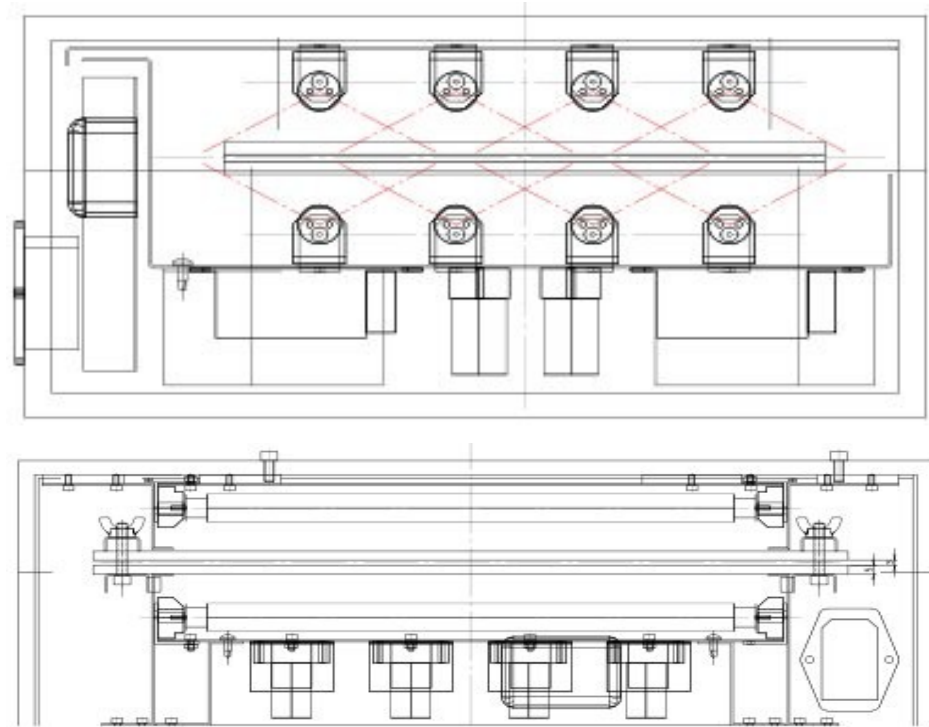
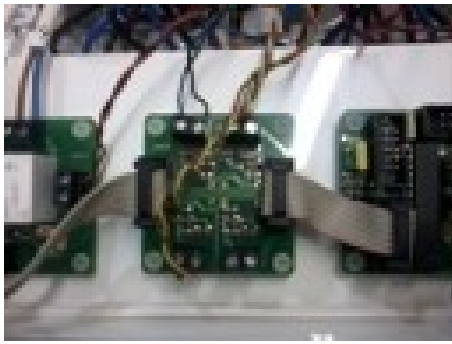


Home

Name ▾	Modified ▾	Size ▾	Downloads ▾
Desktop Software	2011-12-21		
FirmwareAtmega	2011-12-21		

Totals: 2 Items

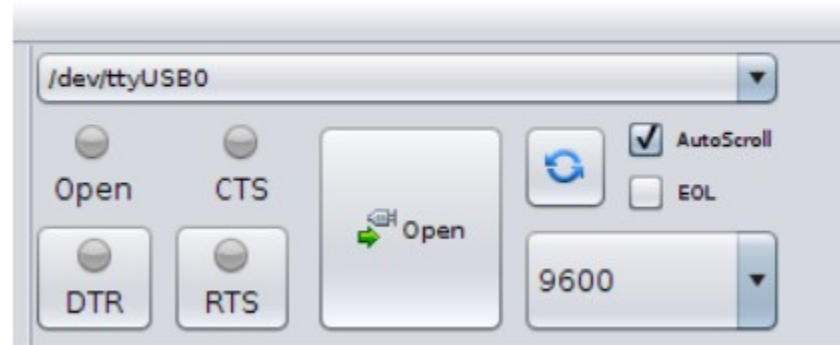




Procés per a posar en marxa l'ESP-12 amb ESPlorer

Executeu :

```
$ java -jar ESPlorer.jar
```



Procés per a posar en marxa l'ESP-12 amb ESPlorer

The screenshot displays the ESPlorer IDE interface. On the left, the code editor shows a Lua script for a web server on an ESP8266. The script sets the WiFi mode to STATION, configures the SSID and password, and sets up two LEDs. It then creates a TCP server on port 80 and listens for connections. The server responds with an HTML page that says "ESP8266 Web Server".

```

1  wifi.setmode(wifi.STATION)
2  wifi.sta.config("ONOF5AC", "2445418455")
3  print(wifi.sta.getip())
4  led1 = 4
5  led2 = 5
6  gpio.mode(led1, gpio.OUTPUT)
7  gpio.mode(led2, gpio.OUTPUT)
8  srv=net.createServer(net.TCP)
9  srv:listen(80,function(conn)
10     conn:on("receive", function(client,request)
11         local buf = "";
12         local _, _, method, path, vars = string.find(request,
13             "^(GET|POST|HEAD|OPTIONS|PUT|DELETE|PATCH|TRACE|CONNECT) ", 1)
14         if method == nil then
15             _, _, method, path = string.find(request, "([A-Z]+) ")
16         end
17         local _GET = {}
18         if (vars ~= nil) then
19             for k, v in string.gmatch(vars, "(%w+)=(%w+)&*") do
20                 _GET[k] = v
21             end
22             buf = buf.."<h1> ESP8266 Web Server</h1>";

```

The right pane shows the terminal output of the upload process. It indicates that the firmware cannot be autodetected, but the upload of the Lua scripts was successful. The terminal also shows the output of the Lua code, displaying the IP address and the response to a GET request.

```

Can't autodetect firmware, because proper answer not r
00000000
00000000 g0

NodeMCU 0.9.6 build 20150704 powered by Lua 5.1.4
lua: cannot open init.lua
> Uploading to ESP file closeTcp.lua...Success
> Uploading to ESP file xpi.lua...Success
> dofile("xpi.lua")
nil
> =wifi.sta.getip()
192.168.1.43 255.255.255.0 192.168.1.1
> =wifi.sta.getip()
192.168.1.43 255.255.255.0 192.168.1.1
>

```

The bottom of the IDE shows various control buttons such as "Save & Run", "View on ESP", "Run", and "Upload...". The status bar at the bottom indicates the current file being edited: `=wifi.sta.getip()`.

Procés per a posar en marxa l'ESP-12 amb ESPlorer

```
> dofile("xpi.lua")
192.168.1.43 255.255.255.0 192.168.1.1
xpi.lua:8: only one tcp server allowed
> Uploading to ESP file closeTcp.lua...Success

> dofile("closeTcp.lua")
> dofile("xpi.lua")
nil
> =wifi.sta.getip()
192.168.1.43 255.255.255.0 192.168.1.1
>
```



```
1  wifi.setmode(wifi.STATION)
2  wifi.sta.config("ONOF5AC","2445418455")
3  print(wifi.sta.getip())
4  led1 = 4
5  led2 = 5
6  gpio.mode(led1, gpio.OUTPUT)
7  gpio.mode(led2, gpio.OUTPUT)
8  srv=net.createServer(net.TCP)
9  srv:listen(80,function(conn)
10     conn:on("receive", function(client,request)
11         local buf = "";
12         local _, _, method, path, vars = string.find(request, "([A-Z]+) (.+)?(.+) HTTP");
13         if(method == nil)then
14             _, _, method, path = string.find(request, "([A-Z]+) (.+) HTTP");
15         end
16         local _GET = {}
17         if (vars ~= nil)then
18             for k, v in string.gmatch(vars, "(%w+)=(%w+)&*") do
19                 _GET[k] = v
20             end
21         end
22         buf = buf.."<h1> ESP8266 Web Server</h1>";
23         buf = buf.."<p>GPIO0 <a href=?pin=ON1?><button>ON</button></a>&nbsp;<a href=?pin=OFF1?><button>OFF</button></a></p>";
24         buf = buf.."<p>GPIO2 <a href=?pin=ON2?><button>ON</button></a>&nbsp;<a href=?pin=OFF2?><button>OFF</button></a></p>";
25         local _on,_off = "", ""
26         if(_GET.pin == "ON1")then
27             gpio.write(led1, gpio.HIGH);
28         elseif(_GET.pin == "OFF1")then
29             gpio.write(led1, gpio.LOW);
30         elseif(_GET.pin == "ON2")then
31             gpio.write(led2, gpio.HIGH);
32         elseif(_GET.pin == "OFF2")then
33             gpio.write(led2, gpio.LOW);
34         end
35         client:send(buf);
36         client:close();
37         collectgarbage();
38     end)
39 end)
```

closeTcp.lua ✕

```
1  srv.close(srv)
2  srv=nil
```



```
jordi@eCat:~/Documents/electronics.cat/conferencies/20151028/code/python$ python esp8266server.py
/dev/ttyUSB0 ONOF5AC 2445418455
Sending command: AT
[Vendor:www.ai-thinker.com Version:0.9.2.4]

ready
Command result: ready
Sending command: AT+CWMODE=1
no change
Command result: no change
Sending command: AT+CWLAP
+CWLAP:(2,"MOVISTAR_9FCA",-89,"f8:8e:85:d9:9f:cb",1)
+CWLAP:(2,"MOVISTAR_F73D",-91,"f8:8e:85:e6:f7:3e",1)
+CWLAP:(4,"ONOF5AC",-55,"c0:3f:0e:c2:f5:ac",2)
+CWLAP:(4,"Valledupar",-77,"84:9c:a6:47:a7:33",2)
+CWLAP:(4,"INT_U9T7H9",-80,"10:fe:ed:9c:16:6a",2)
+CWLAP:(0,"_AUTO_ONOWiFi",-56,"c2:3f:0e:c2:f5:ad",2)
+CWLAP:(0,"_ONOWiFi",-56,"c2:3f:0e:c2:f5:ae",2)
+CWLAP:(3,"TRAC",-93,"28:94:0f:f9:be:b6",6)
+CWLAP:(4,"Orange-C341",-84,"88:03:55:aa:c3:43",7)
+CWLAP:(2,"vodafonejpl",-81,"20:2b:c1:37:22:c3",8)
+CWLAP:(3,"Nincols",-85,"20:c9:d0:1b:2d:69",11)
+CWLAP:(4,"InOutTravel",-93,"0a:18:d6:0b:61:f1",11)
+CWLAP:(0,"Kubi_CityWifi_Fast_Internet",-76,"a4:6c:2a:68:ea:71",11)
+CWLAP:(0,"Barcelona WiFi",-76,"a4:6c:2a:68:ea:73",11)
+CWLAP:(1,"TechSales",-59,"e0:cb:4e:59:d9:23",13)

OK
Command result: OK
Sending command: AT+CWJAP="ONOF5AC","2445418455"
Command result: AT+CWJAP="ONOF5AC","2445418455"

Sending command: AT+CIFSR
busy p...

OK
Command result: OK
Sending command: AT+CIPMUX=1

OK
Command result: OK
Sending command: AT+CIPSERVER=1,80

OK
Command result: OK
Sending command: AT+CIFSR
192.168.1.44

OK
Command result: OK
```

Codi de l'Arduino Nano per a connectar l'ESP8266

```
1  #include <SoftwareSerial.h>
2
3  SoftwareSerial mySerial(10, 11); // RX, TX
4
5  void setup(){
6      Serial.begin(9600);
7      while (!Serial) {
8          ; // wait for serial port to connect. Needed for Leonardo only
9      }
10     mySerial.begin(9600);
11 }
12
13 void loop(){
14     if (mySerial.available())
15         Serial.write(mySerial.read());
16     if (Serial.available())
17         mySerial.write(Serial.read());
18 }
19
```

```
82 if len(sys.argv) != 4:
83     print "Usage: esp8266test.py port ssid password"
84     sys.exit()
85
86 port = sys.argv[1]
87 speed = 9600
88 ssid = sys.argv[2]
89 pwd = sys.argv[3]
90 p = 80
91
92 ser = serial.Serial(port,speed)
93 if ser.isOpen():
94     ser.close()
95 ser.open()
96 ser.isOpen()
97
98 send_cmd( "AT" )
99 send_cmd( "AT+CWMODE=1" ) # set device mode (1=client, 2=AP, 3=both)
100 send_cmd( "AT+CWLAP", 30) # scan for WiFi hotspots
101 send_cmd( "AT+CWJAP=\""+ssid+"\", \""+pwd+"\" , 5 ) # connect
102 send_cmd( "AT+CIFSR", 5) # check IP address
103
104 send_cmd( "AT+CIPMUX=1" ) # multiple connection mode
105 send_cmd("AT+CIPSERVER=1," + str(p))
106 send_cmd( "AT+CIFSR", 5) # check IP address
107
108 # process requests
109 while (1):
110     process_request("electronics.cat (" + str(datetime.datetime.now()) + ")")
111     sleep(0.3)
112
113 ser.close()
```

Torn de preguntes ...



... i sessió pràctica.

COMIAT

Presentació descarregable a :

<http://binefa.cat/blog/?p=256>

Correu electrònic de contacte : jordibinefa@electronics.cat

twitter



<https://twitter.com/electronicscat>

• <https://twitter.com/JordiBinefa>

LinkedIn

<http://es.linkedin.com/pub/jordi-binefa/13/717/90b>

Moltes gràcies per la vostra assistència

