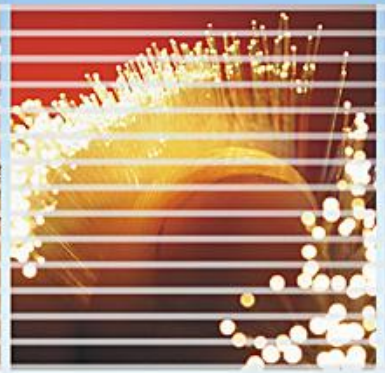
CYPRESS

Connecting From Last Mile to First Mile.™

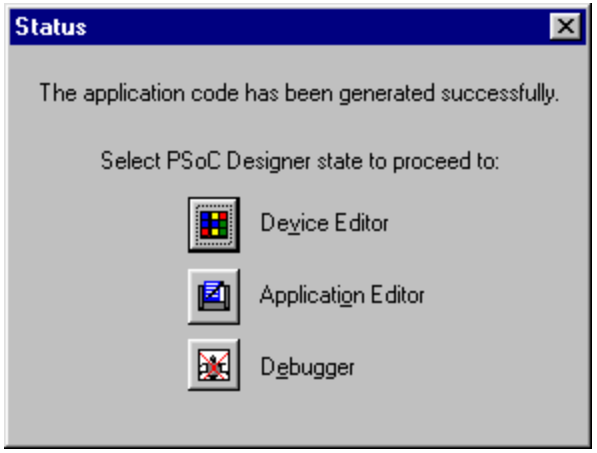
Capítol 2: Introducció l'entorn



PERSONAL ACCESS ENTERPRISE METRO CORE

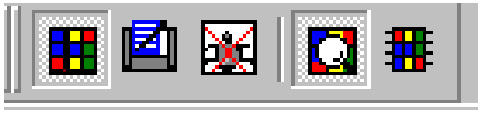
L’editor de dispositiu

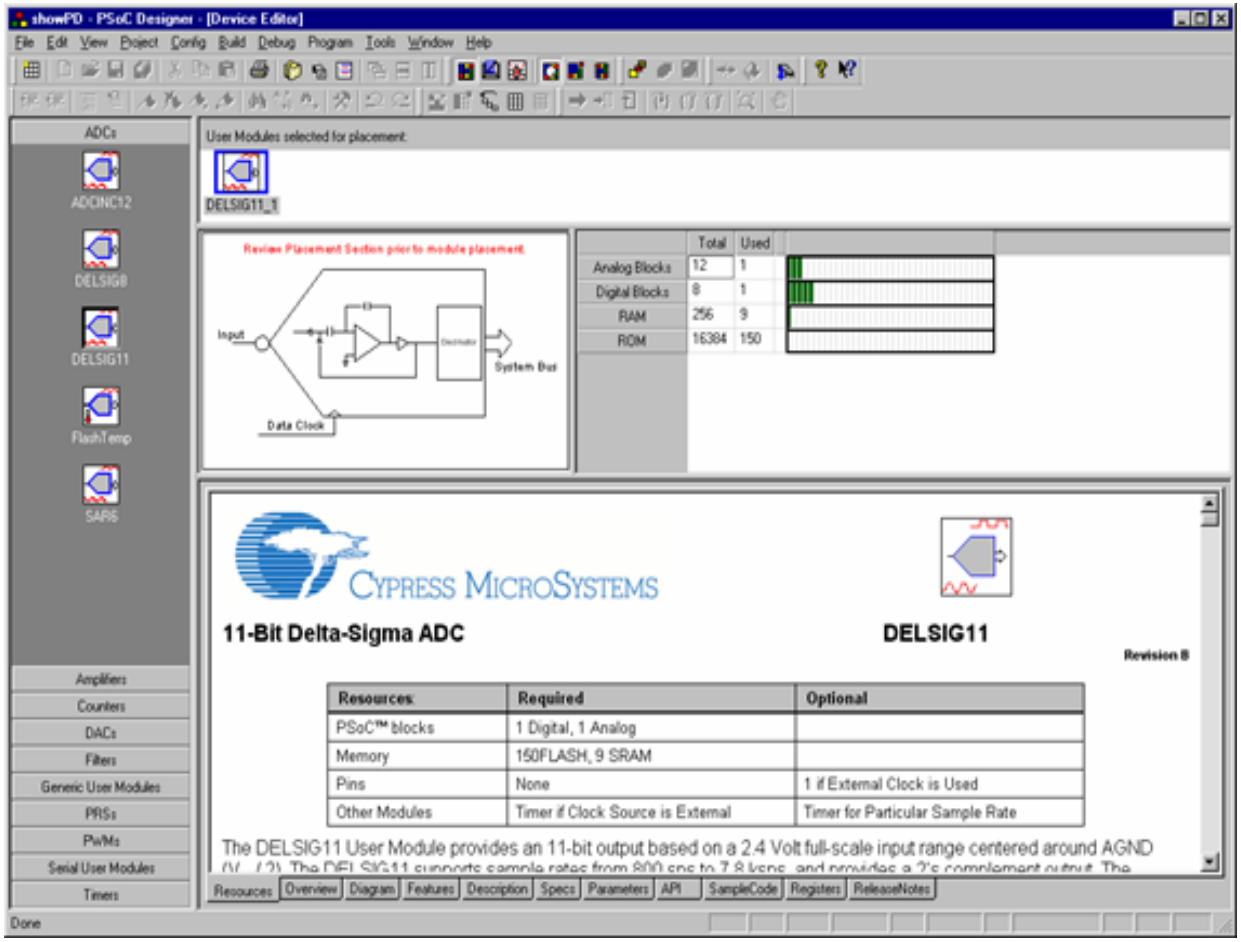
- Només començar un nou projecte, podeu navegar per les possibles parts del mateix:



- **Generar un floorplan amb els mòduls necessitats.**
- **Configurar els fitxers font per a la programació del dispositiu.**
- **Generar els fitxers font, fent servir el “aplication editor”.**
- **Crear el teu propi “datasheet” del xip.**
- **Executar pas a pas el codi.**

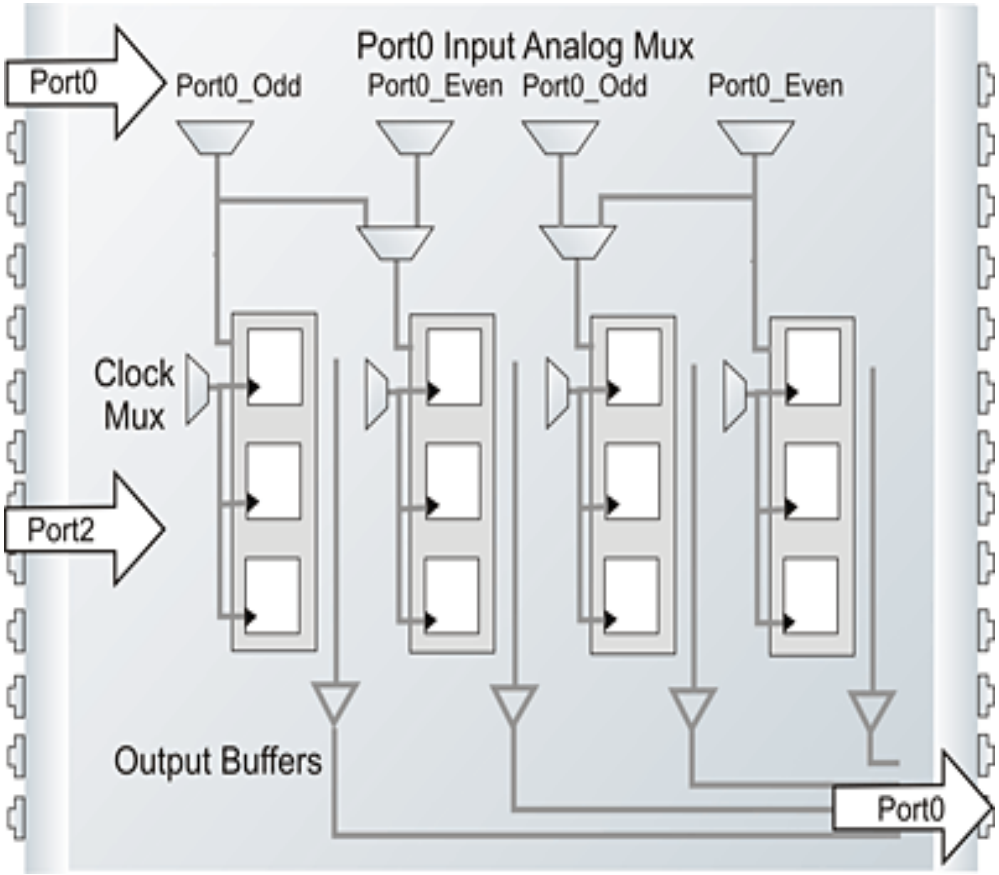
•Selecció dels mòduls d'usuari

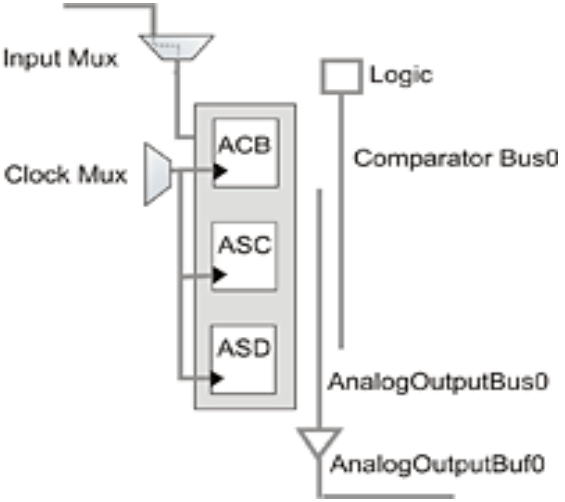




- Veure la llista dels catàleg de mòduls d'usuari.
- Veure el datasheet de cada mòdul.
- Selecció dels mòduls i inclusió dins el projecte.
- Veure la totalitat de recursos de sistema.

Bloques Análogos

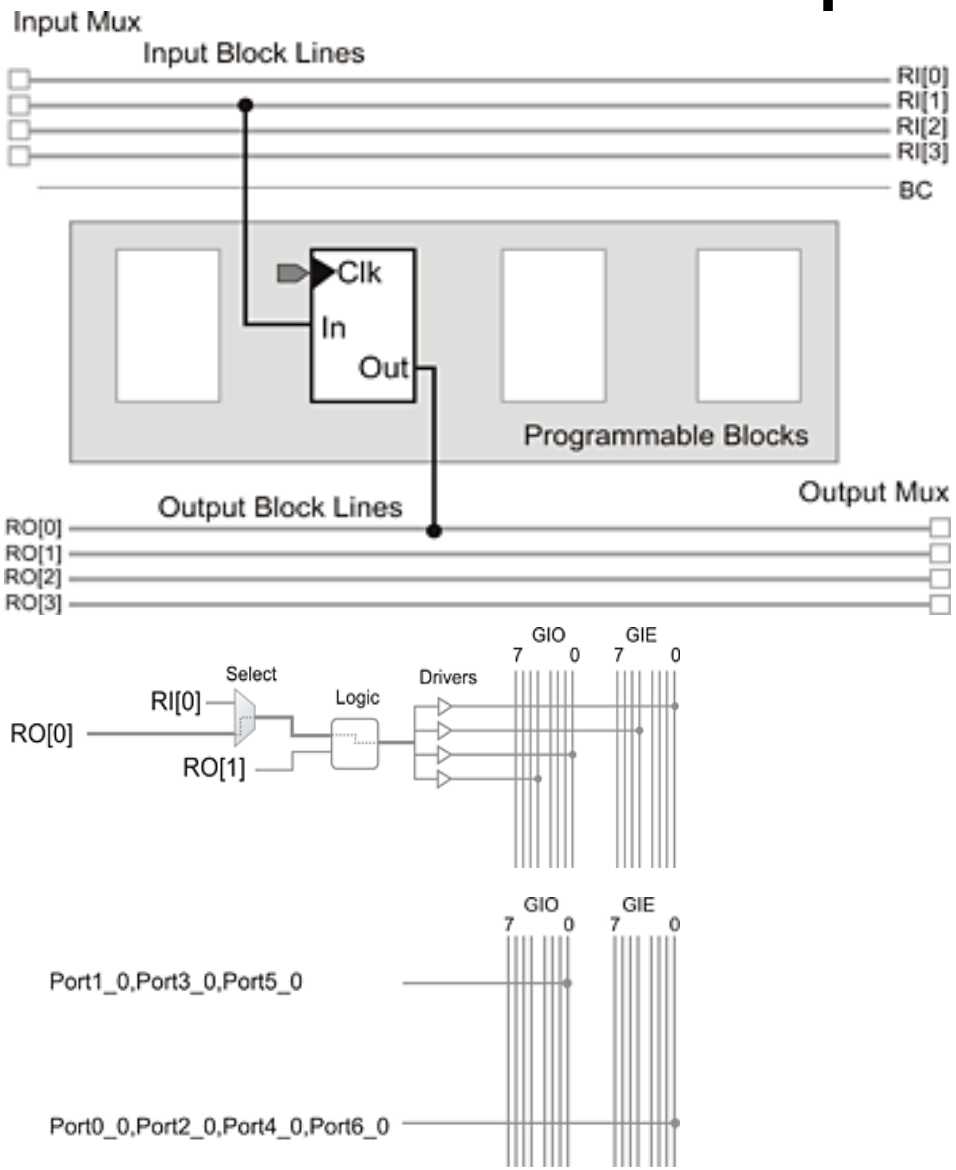




Tipos de bloques:

- **ACB**
- **ASC**
- **ASD**

Bloques Digitales



Tipos de bloques:

- **DCB**
- **DBB**

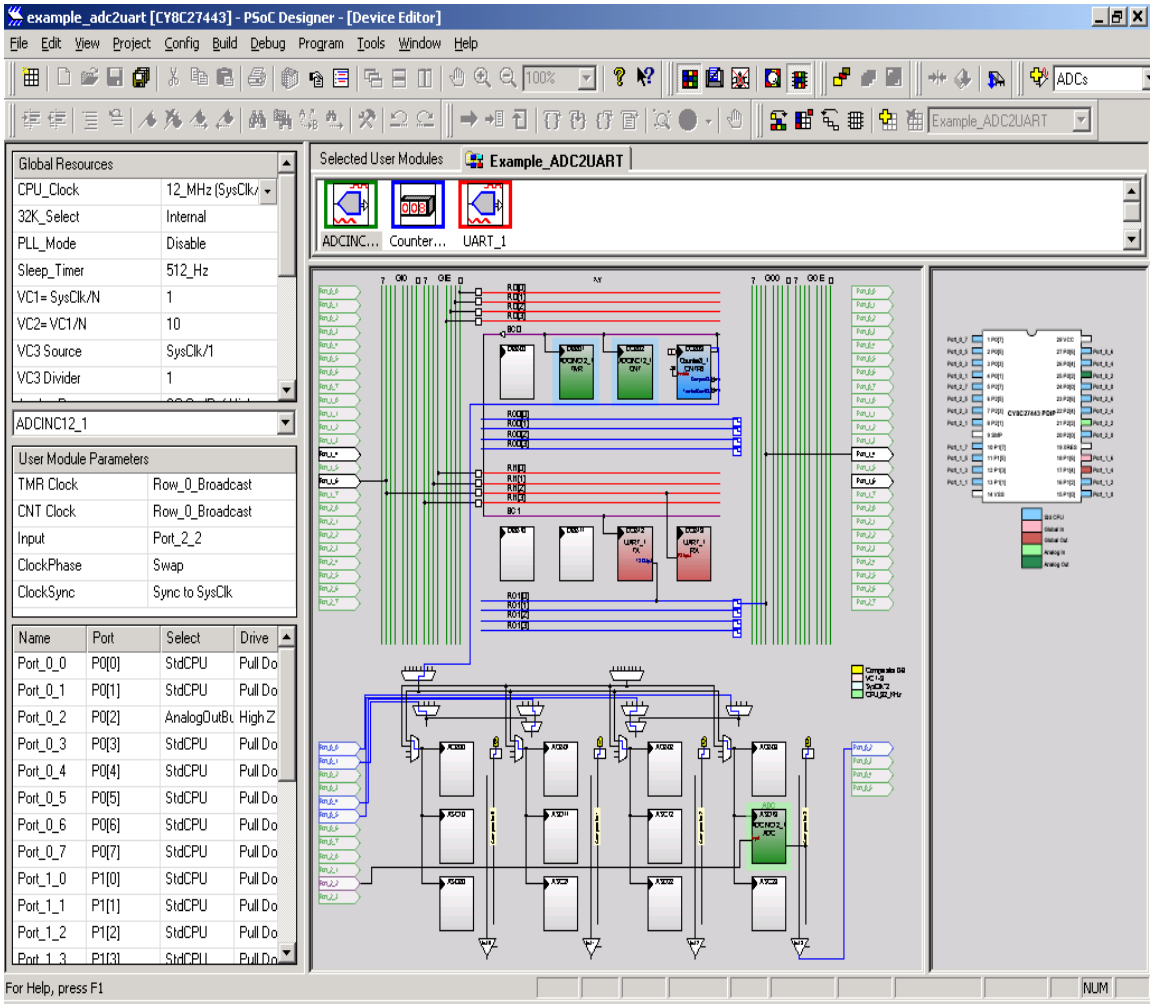
Estos bloques tienen como agregado el hecho que pueden programar su salida mediante bloques lógicos, a través de operaciones lógicas AND, OR, XOR

(GIO, GIE)Son las líneas globales de entrada para la comunicación de sistemas digitales entre las líneas de entrada y los multiplexores. Estas se encuentran divididas en 2 grupos las cuales se separan en las líneas pares (GIO) tales como P1(2) y líneas impares (GIE), como por ejemplo, P1(3)

PSoC Designer

Editor combinat amb el plànol del xip

Ubicació dels mòduls d'usuari

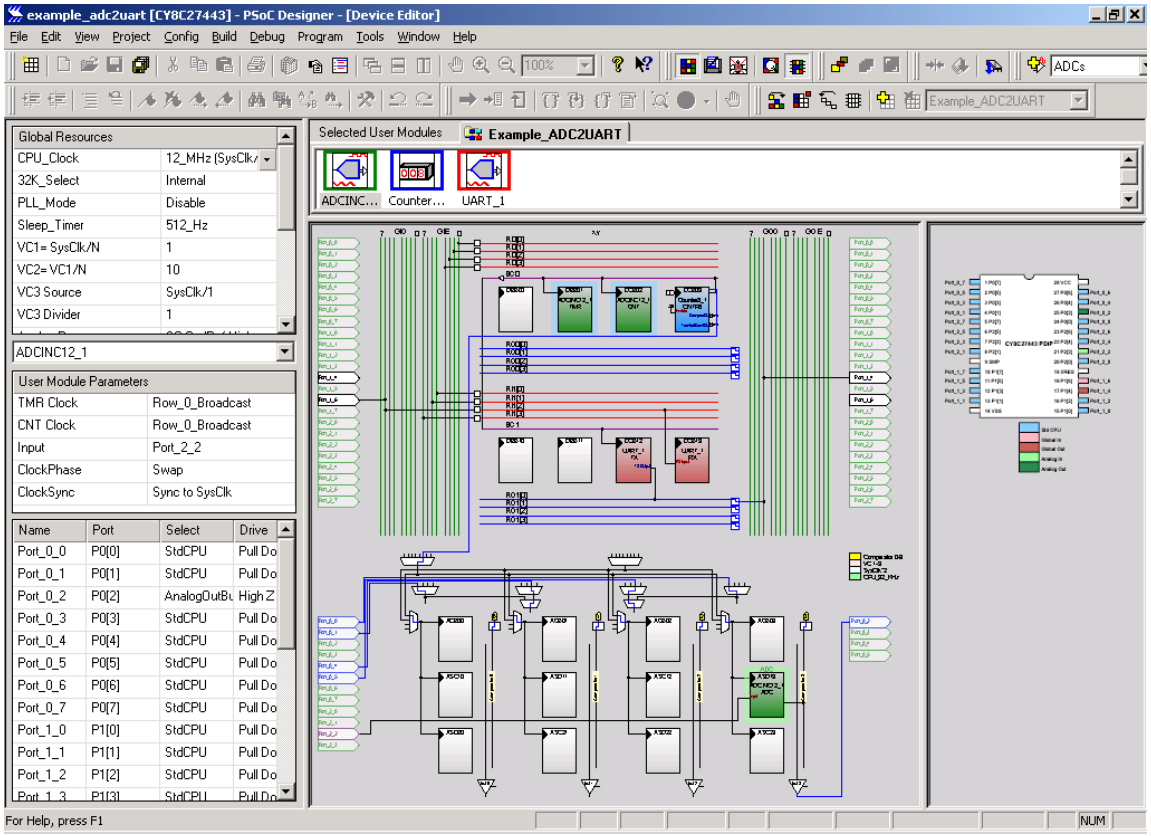


- **Veure l'arquitectura del sistema, per blocs combinant mòduls d'usuari i el traçament dels bussos i pistes.**
- **Conèixer els millors llocs per a ubicar els UM segons les interconnexions.**
- **Seleccionar i configurar els UM i els recursos globals del sistema.**
- **Definir el clock per als Um.**

PSoC Designer

Editor combinat amb el plànol del xip

- Especificació del pin-out



- Escollir les opcions del pin, del mòdul escollit.
- Fer les connexions des dels pins, fins als mòduls d'usuari.
- Escollir el mode, i el comandament dels GPIO pins.

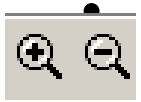
PSoC Designer

Opcions de visibilitat

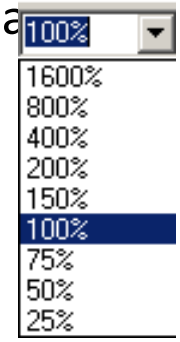
- Barra de View 

- Original
  - Recupera el format original del disseny.

- Moure el disseny



- Zoom
  - Pujar / Baixar Zoom
  - Zoom seleccionat



PSoC Designer

Opcions de visibilitat

- Accessos directes a modes de visibilitat

- Control + clic
  - zoom in
  - + shift: zoom out

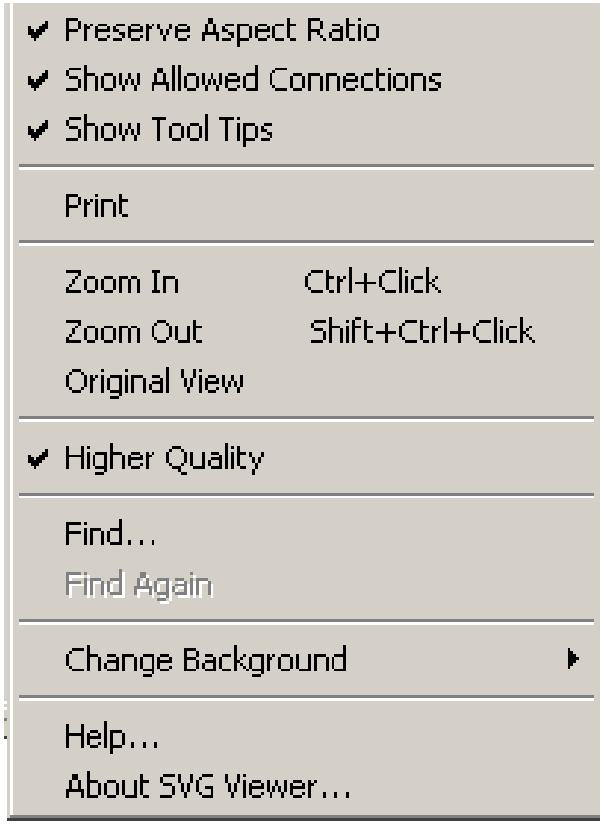
- Alt + clic
  - Permet moure't pel disseny

PSoC Designer

Opcions de visibilitat

- Menú

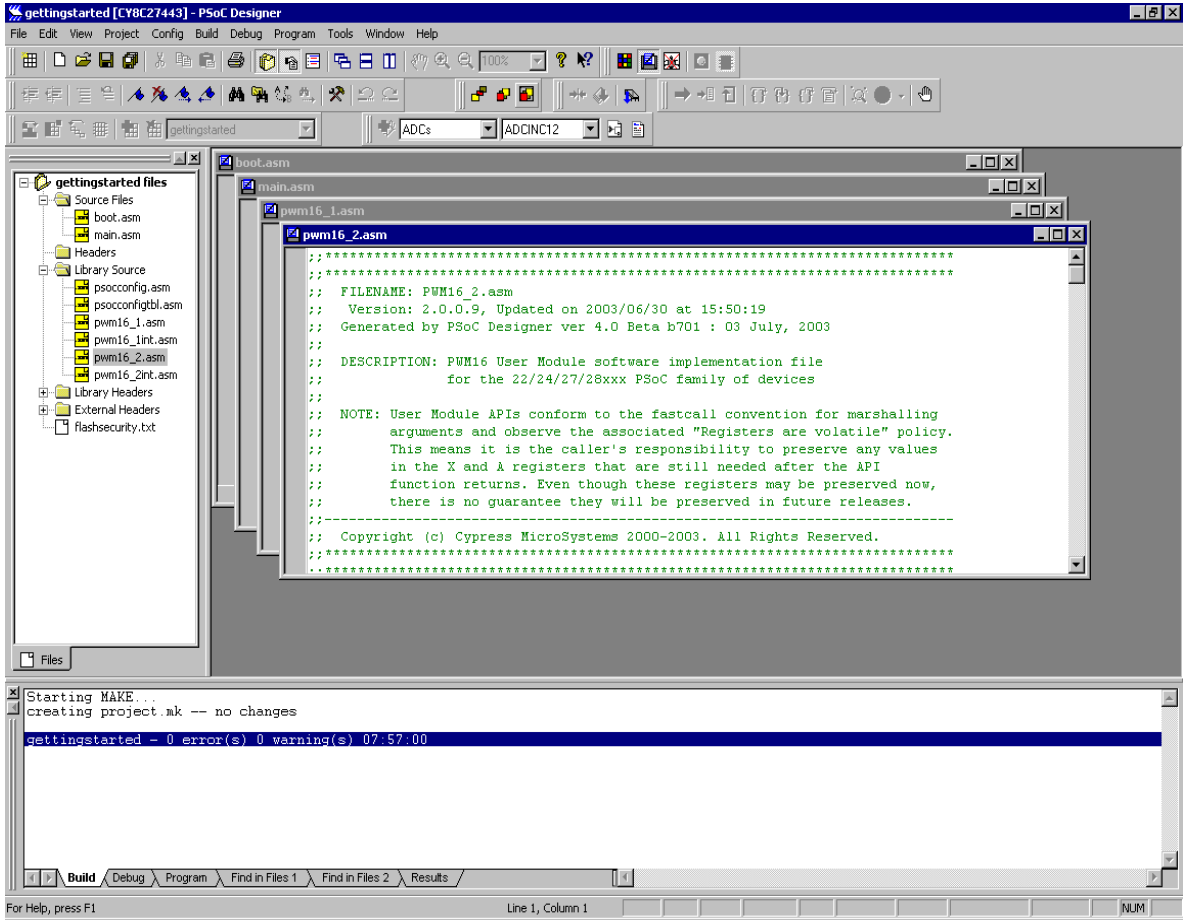
- Botó dret a un espai buit per al menú d'ajuda:
  - Zoom In
  - Zoom Out
  - Original View
  - Qualitat dels gràfics
  - Fons dels gràfics



PSoC Designer

Editor d'aplicacions

Per escriure codis en assembler / c:



- **Manipular codis amb els dos llenguatges.**
- **Tractar amb el compilador, i amb les llibreries.**

Compilador de C del PSoC Designer

- El compilador de C, CY3202-C d'iMAGEcraft, és un component opcional al Psoc designer.
- Permet fer fitxers fonts amb C, i depurar-los.
- Inclou les següents característiques:

Compilador d'ANSI C.

Es pot incorporar codis d'assembler.

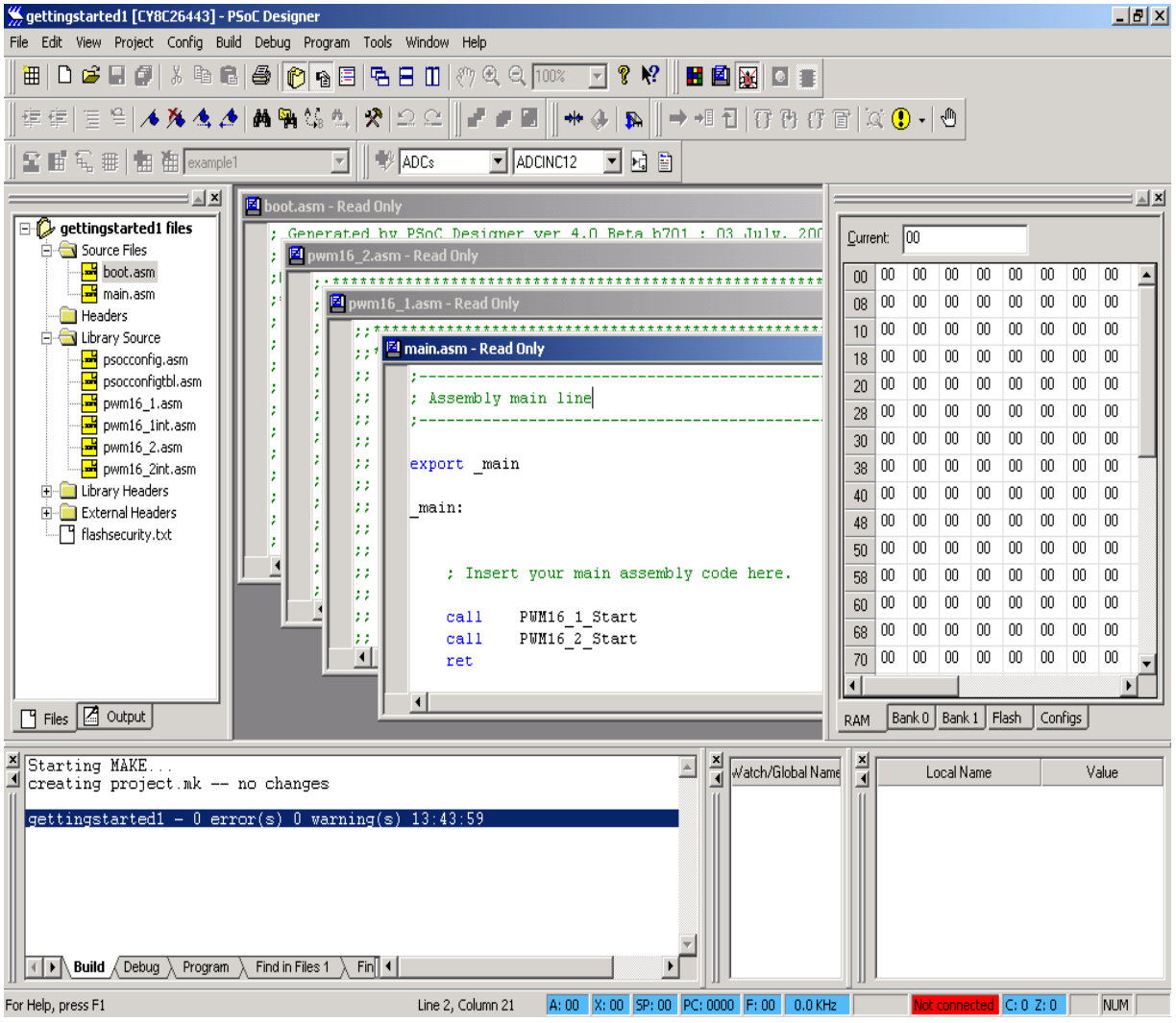
Compressor de codis integrat.

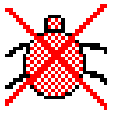
7 tipus de dades incloent variables en. coma flotant compatible amb IEEE 32.

Llibreries de matemàtiques i d'strings.

Rutines d'interrupció amb C.

Depurador





- **Interfaz amb el IDE.**
- **Visualitzar els continguts, i manipulació dels registres i mapa de memòria.**
- **Run/Halt /Single Step.**
- **Ubicar breakpoints.**

PSoC Designer 5.0

Altres característiques

- Design Rule Checker

- Reducció del tamany del codi
  - Opció de lincar només les APIS que es fan servir.
  - Compressió del codi amb C entre el 2% i el 10%.
- Permet diferents modes de lincat.

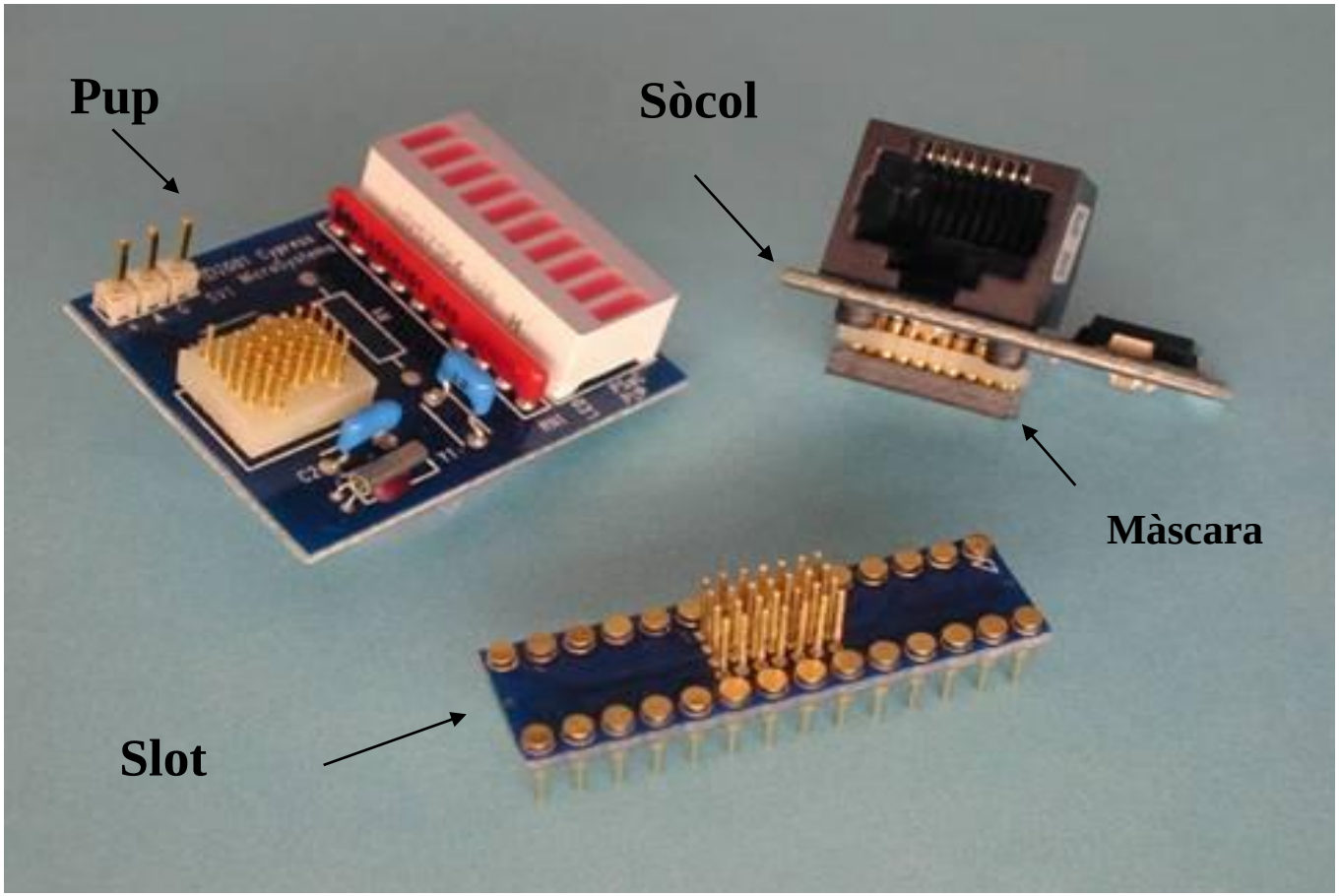
- Característiques millorades del depurador
  - Depurar pas a pas, assembler i C.
  - Veure els continguts dels vectors, byte a byte.

Kit de desenvolupament

CY3205-DK

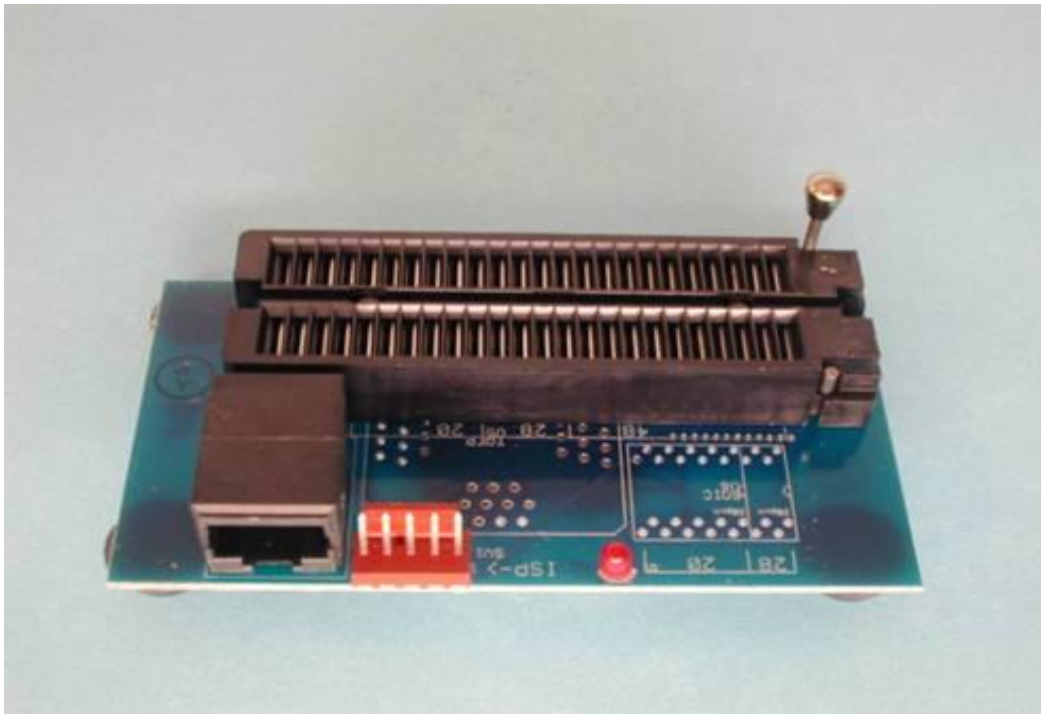
Conté tot ho necessari per a donar suport al model d'empaquetament 28-pin PDIP.

Connector del Psoc amb plaques externes



Programador de Xips

- **Existeix un tipus de programador per a cada model d'encapsulat.**



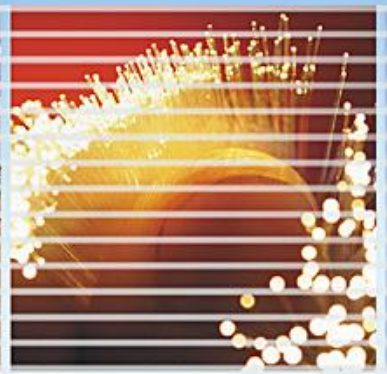
Programador

Model Xip	sòcol programador	Empaquetat
CY8C27143-24PI	CY3205-PR	8 PDIP
CY8C27243-24SI	CY3205-S2	20 SOIC
CY8C27243-24PVI	CY3205-S3	20 SSOP
CY8C29466-24PVXI	CY3205-PR	28 PDIP
CY8C27443-24SI	CY3205-S2	28 SOIC
CY8C27443-24PVI	CY3205-S3	28 SSOP
CY8C27543-24AI	CY3205-S5	44 TQFP
CY8C2643-24PVI	CY3205-S4	48 SSOP

CYPRESS

Connecting From Last Mile to First Mile.™

Introducció al PSoCEval



PERSONAL ACCESS ENTERPRISE METRO CORE

Estructura PSoCEval

- El PSoCEval és un kit de desenvolupament que permet entrenar tota la família de PSoC 26XX, 27XX i 29XX.
- La placa està formada per:

- El Sòcol d'entrenament.
- La Board de desenvolupament.
- Zona de senyalització (4 leds).
- Zona d'actuació (1 polsador).
- Zona analògica (potenciòmetre de regulació de 0 a 5v).
- Zona de comunicació amb perifèric extern (RS232).
- Zona d'alimentació (donada una tensió d'alimentació gran, ens la transforma amb 5v constants).
- Zona de programació (kit de programació).

Cròquis del disseny

-

Alimentació

Comunicació

Sòcol de programació

Connector RS232

Botó de rèset

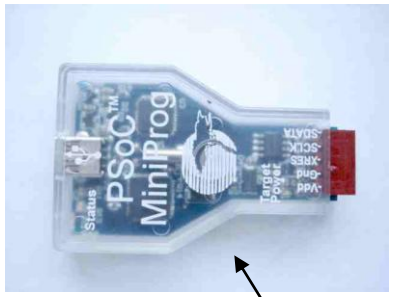
Potenciòmetre

Leds

Zona de disseny

Polsador entrada

Contrast LCD



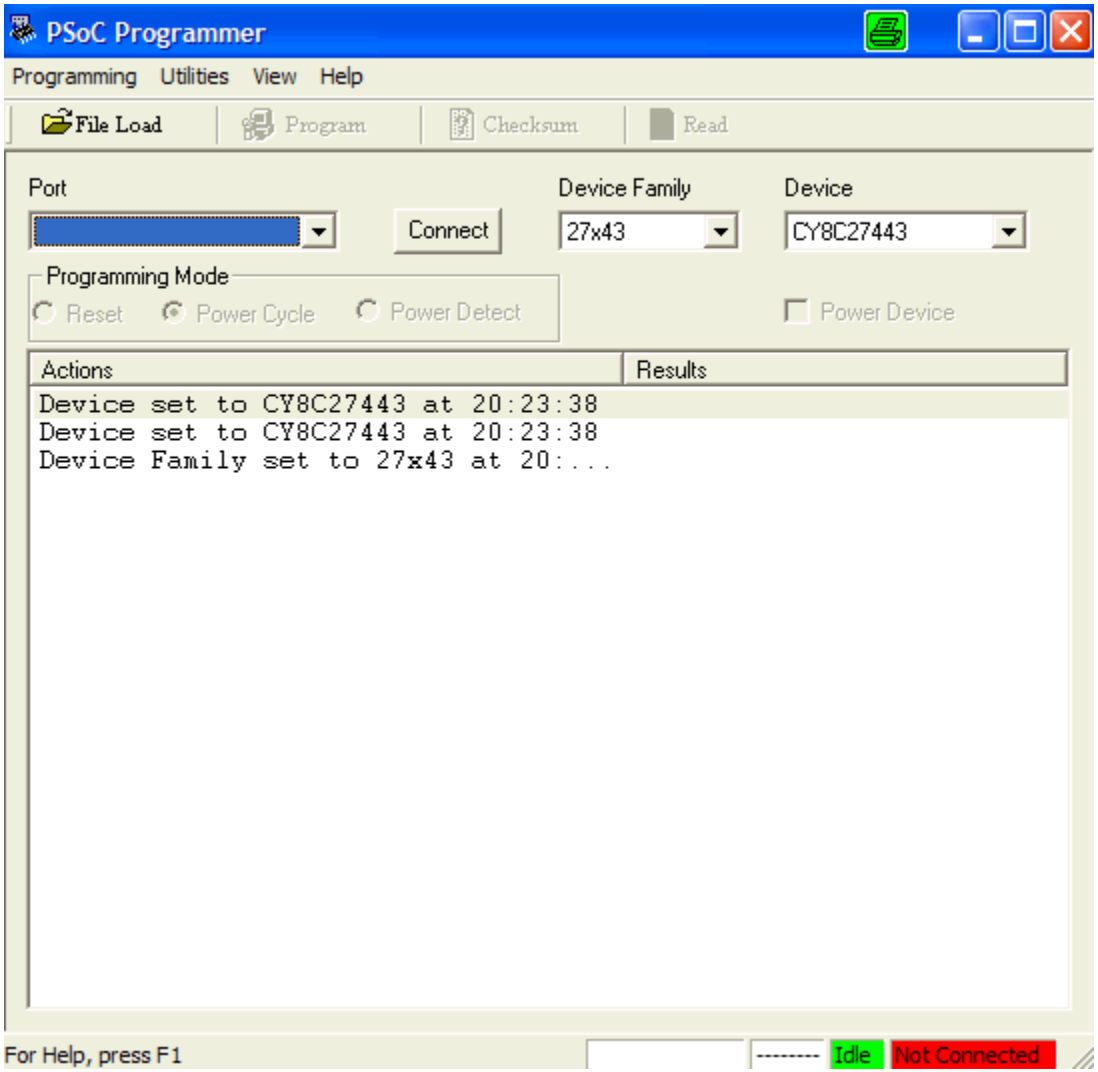
Programador

Pantalla LCD

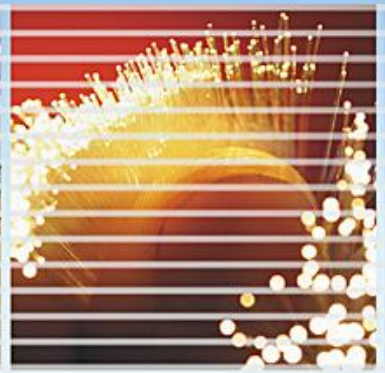
Consideracions

- El PSoCEval fa servir la alimentació del propi PC a través de l'USB.
- ÉS IMPORTANTISSIM NO FER CAP CORTCIRCUIT AMB ELS DISSENYS QUE IMPLEMENTAREM. PER A QUALSEVOL DUBTE DEMANEU SEMPRE PEL PROFESSOR.
- El projecte que farem al final del Curs, farà servir tots el recursos d'aquesta placa.
- Una vegada muntat el circuit cal que el professor el verifiqui.
- Tot seguit connectarem el programador al port USB del PC, que és aparell blanc que es connecta al sòcol de programació, on les lletres miren cap a la placa (consulteu al professor).
- Ara engeguem el programa PSoC Programmer, seleccionant el Device Family 29X46 i el Device CY8C29466. L'aplicació s'ha de posar en mode Connected.
- Seguidament, cal tenir el programador en mode Not powered. Si està en mode powered no deixarà programar el PSoC. Per activar/desactivar l'alimentació cal clicar sobre l'option Power Device.

PSoC Programmer



Capítol 3:
Mini manual de C



Elements a presentar

- **Tipus de variables**
- **Operacions**
- **funcions memòria - string**
- **funcions matemàtiques**

Tipus de variables

Type	Bytes	Description	Range
char	1	A single byte of memory that defines characters	[1] unsigned 0…255 signed -128…127
int	2	Used to define integer numbers	unsigned 0…65535 [1] signed -32768…32767
short	2	Standard type specifying 2-byte integers	unsigned 0…65535 [1] signed -32768…32767
long	4	Standard type specifying the largest integer entity	unsigned 0…4294967295 [1] signed -2147483648…2147483647
float	4	Single precision floating point number in IEEE format	1.175e-38…3.40e+38
double	4	Single precision floating point number in IEEE format	1.175e-38…3.40e+38
enum	1 if enum < 256 2 if enum > 256	Used to define a list of aliases that represent integers.	0…65535

Operacions

Pre.	Op.	Function	Group	Form	Description
1	++	Postincrement		a ++	
1	--	Postdecrement		a --	
1	[ ]	Subscript		a[b]	
1	( )	Function Call		a(b)	
1	.	Select Member		a.b	
1	->	Point at Member		a->b	
2	sizeof	Sizeof		sizeof a	
2	++	Preincrement		++ a	
2	--	Predecrement		-- a	
2	&	Address of		&a	
2	*	Indirection		*a	
2	+	Plus		+a	
2	-	Minus		-a	
2	~	Bitwise NOT	Unary	~ a	1's complement of a
2	!	Logical NOT		!a	

Operacions II

2	(decla-ration)	Type Cast			(declaration)a
3	*	Multiplication	Binary	a * b	a multiplied by b
3	/	Division	Binary	a / b	a divided by b
3	%	Modulus	Binary	a % b	Remainder of a divided by b
4	+	Addition	Binary	a + b	a plus b
4	-	Subtraction	Binary	a - b	a minus b
5	<<	Left Shift	Binary	a << b	Value of a shifted b bits left
5	>>	Right Shift	Binary	a >> b	Value of a shifted b bits right
6	<	Less		a < b	a less than b
6	<=	Less or Equal		a <= b	a less than or equal to b
6	>	Greater		a > b	a greater than b
6	>=	Greater or Equal		a >= b	a greater than or equal to b
7	==	Equals		a == b	

Operacions III

Pre.	Op.	Function	Group	Form	Description
7	!=	Not Equals		a != b	
8	&	Bitwise AND	Bitwise	a & b	Bitwise AND of a and b
9	^	Bitwise Exclusive OR	Bitwise	a ^ b	Bitwise Exclusive OR of a and b
10	|	Bitwise Inclusive OR	Bitwise	a | b	Bitwise OR of a and b
11	&&	Logical AND		a && b	
12	||	Logical OR		a || b	
13	? :	Conditional		c?a:b	
14	=	Assignment		a = b	
14	*=	Multiply Assign		a *= b	
14	/=	Divide Assign		a /= b	
14	%=	Remainder Assign		a %= b	
14	+=	Add Assign		a += b	
14	-=	Subtract Assign		a -= b	
14	<<=	Left Shift Assign		a <<= b	
14	>>=	Right Shift Assign		a >>= b	

Operacions de flux

Tipus

- if - else
- Switch
- while
- do
- for
- goto
- continue
- break
- return
- struct
- typedef
- punters

Funcions memòria - string

Function	Prototype	Description	Header
abs	int abs(int);		stdlib.h
atof	double atof(CONST char *);		stdlib.h
atoi	int atoi(CONST char *);		stdlib.h
atol	long atol(CONST char *);		stdlib.h
itoa	void itoa(char *string, unsigned int value, int base);		stdlib.h
ltoa	void ltoa(char *string, unsigned long value, int base);		stdlib.h
ftoa	char *ftoa(float f, int *status); /* ftoa function */ #define _FTOA_TOO_LARGE -2 /* |input| > 2147483520 */ #define _FTOA_TOO_SMALL -1 /* |input| < 0.0000001 */ /* ftoa returns static buffer of ~15 chars. If the input is out of * range, *status is set to either of the above #define, and 0 is * returned. Otherwise, *status is set to 0 and the char buffer is * returned.	* This version of the ftoa is fast but cannot handle values outside * of the range listed. Please contact us if you need a (much) larger * version that handles greater ranges. * Note that the prototype differs from the earlier version of this * function. */	stdlib.h

Funcions memòria - string II

rand	int rand(void);		stdlib.h
srand	void srand(unsigned);		stdlib.h
strtol	long strtol(CONST char *, char **, int);		stdlib.h
strtoul	unsigned long strtoul(CONST char *, char **, int);		stdlib.h
cstrcat	char *cstrcat(char *, const char *);		string.h
cstrcmp	int cstrcmp(const char *cs, char *);		string.h
cstrcpy	char *cstrcpy(char *, const char *cs);		string.h
cstrlen	size_t cstrlen(const char *cs);		string.h
memchr	void *memchr(void *, int, size_t);		string.h
memcmp	int memcmp(void *, void *, size_t);		string.h
memcpy	void *memcpy(void *, void *, size_t);		string.h

Funcions memòria - string III

memmove	void *memmove(void *, void *, size_t);		string.h
memset	void *memset(void *, int, size_t);		string.h
strcat	char *strcat(char *, CONST char *);		string.h
strcmp	int strcmp(CONST char *, CONST char *);		string.h
strcoll	int strcoll(CONST char *, CONST char *);		string.h
strcpy	char *strcpy(char *, CONST char *);		string.h
strcspn	size_t strcspn(CONST char *, CONST char *);		string.h
strlen	size_t strlen(CONST char *);		string.h
strncat	char *strncat(char *, CONST char *, size_t);		string.h
strncmp	int strncmp(CONST char *, CONST char *, size_t);		string.h
strncpy	char *strncpy(char *, CONST char *, size_t);		string.h
strpbrk	char *strpbrk(CONST char *, CONST char *);		string.h
strrchr	char *strrchr(CONST char *, int);		string.h
strspn	size_t strspn(CONST char *, CONST char *);		string.h
strstr	char *strstr(CONST char *, CONST char *);		string.h

Funcions matemàtiques

float modf(float y, float *i);	modf splits the double val apart into an integer part and a fractional part, returning the fractional part and storing the integer. The fractional part is returned. Each result has the same sign as the supplied argument val.
float floor(float y);	floor finds the nearest integer less than or equal to x. floor returns the integer result as a double.
float ceil(float y);	ceil finds the nearest integer greater than or equal to x. ceil returns the integer result as a double.
float fround(float d);	Produce a quotient that has been rounded to the nearest mathematical integer; if the mathematical quotient is exactly halfway between two integers, (that is, it has the form integer+1/2), then the quotient has been rounded to the even (divisible by two) integer.
float tan(float x);	The function returns the tangent of x for x in radians. If x is large the value returned might not be meaningful, but the function reports no error.
float acos(float x);	acos computes the inverse cosine (arc cosine) of the input value. Arguments to acos must be in the range -1 to 1. The function returns the angle whose cosine is x, in the range [0, pi] radians.
float exp(float x);	exp calculates the exponential of x, that is, the base of the natural system of logarithms, approximately 2.71828). The function returns the exponential of x, e^x.

Funcions matemàtiques II

float log(float x);	Return the natural logarithm of x, that is, its logarithm base e (where e is the base of the natural system of logarithms, 2.71828...). The function returns the natural logarithm of x.
float pow(float x,float y);	pow calculates x raised to the exp1.0nt y. On success, pow returns the value calculated.
float sinh(float x);	sinh computes the hyperbolic sine of the argument x. The function returns the hyperbolic sine of x.
float cosh(float x);	cosh computes the hyperbolic cosine of the argument x. The function returns the hyperbolic cosine of x.
float fabs(float x);	fabs calculates the absolute value (magnitude) of the argument x, by direct manipulation of the bit representation of x. Return the absolute value of the floating point number x.
float frexp(float x, int *eptr);	All non zero, normal numbers can be described as m * 2**p. frexp represents the double val as a mantissa m and a power of two p. The resulting mantissa will always be greater than or equal to 0.5, and less than 1.0 (as long as val is nonzero). The power of two will be stored in *exp. Return the mantissa and exponent of x as the pair (m, e). m is a float and e is an integer such that x == m * 2**e. If x is zero, returns (0.0, 0), otherwise 0.5 <= abs(m) < 1.
float tanh(float x);	The function returns the hyperbolic tangent of x.
float sin(float x);	Return the sine of x.
float atan(float x);	The function returns the angle whose tangent is x, in the range [-pi/2, +pi/2] radians.

Funcions matemàtiques III

float atan2(float y, float x);	The function returns the angle whose tangent is y/x, in the full angular range [-pi, +pi] radians.
float asin(float x);	The function returns the angle whose sine is x, in the range [-pi/2, +pi/2] radians.
float exp10(float x);	Returns 10 raised to the specified real number.
float log10(float x);	log10 returns the base 10 logarithm of x. It is implemented as log(x) / log(10).
float fmod(float y, float z);	The fmod function computes the floating-point remainder of x/y (x modulo y). The fmod function returns the value for the largest integer i such that, if y is nonzero, the result has the same sign as x and magnitude less than the magnitude of y.
float sqrt(float x);	The function returns the square root of x, x^(1/2).
float cos(float x);	The function returns the cosine of x for x in radians. If x is large the value returned might not be meaningful, but the function reports no error.
float ldexp(float d, int n);	ldexp calculates the value that it takes and returns float rather than double values. ldexp returns the calculated value.