

Module : Informatique Industrielle

Partie: Programmation de microcontrôleur



INSA Centre Val de Loire
Filière MRI 3^{ème} année
E. FERRERE

2020 - 2021

- Découverte des microcontrôleurs PSoC
- Initiation à la programmation des PSoC avec le logiciel “PSoC creator”
- Mise en œuvre du microcontrôleur PSoC 4 avec différents périphériques

Domaines d'application

- Systèmes embarqués
 - Téléphonie
 - Transports
 - Médical
 - Défense, etc.



Contraintes

- Taille réduite
- Consommation électrique
- Coût
- Dédiés Tâches spécifiques

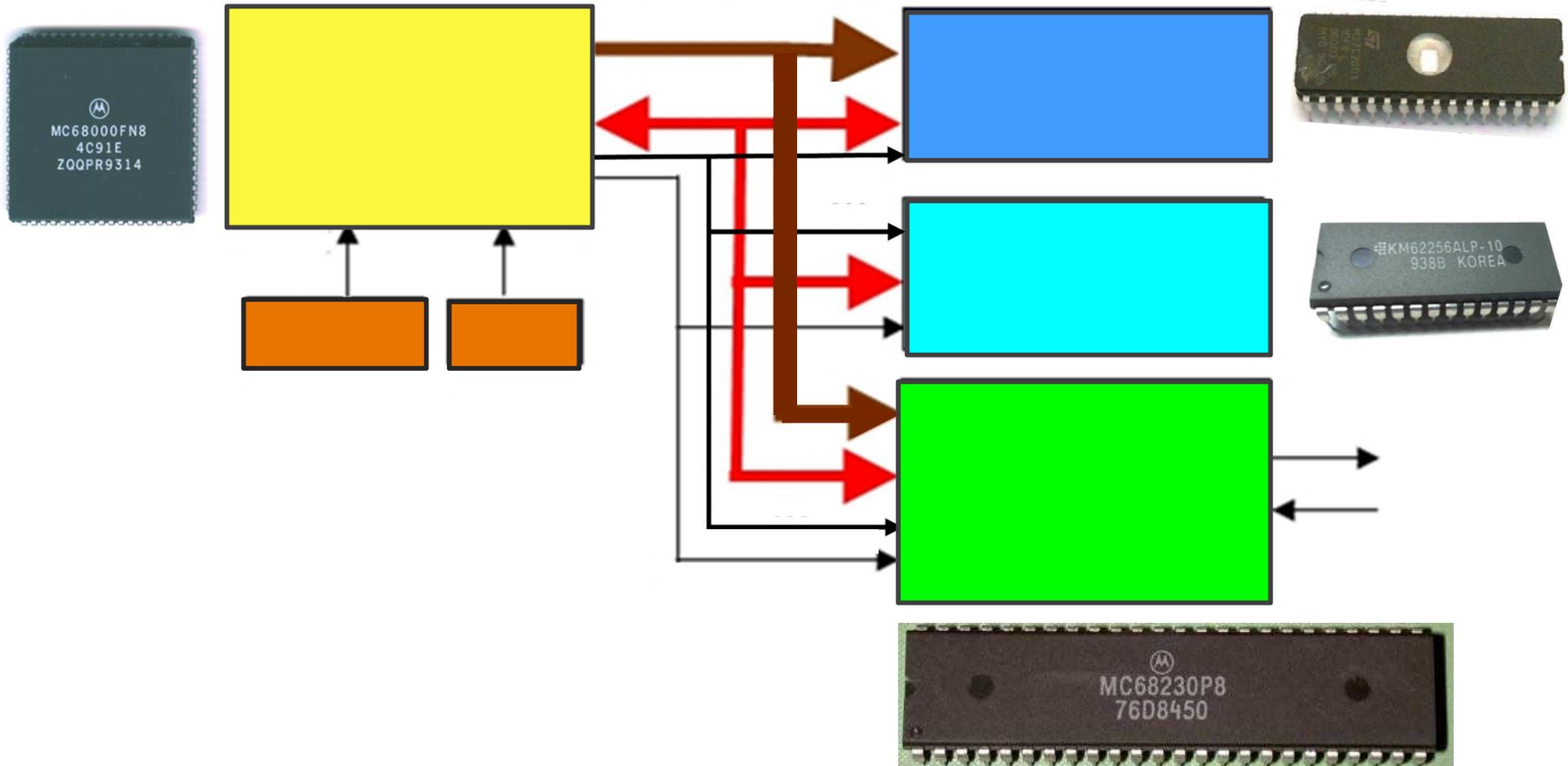


Communication avec l'extérieur

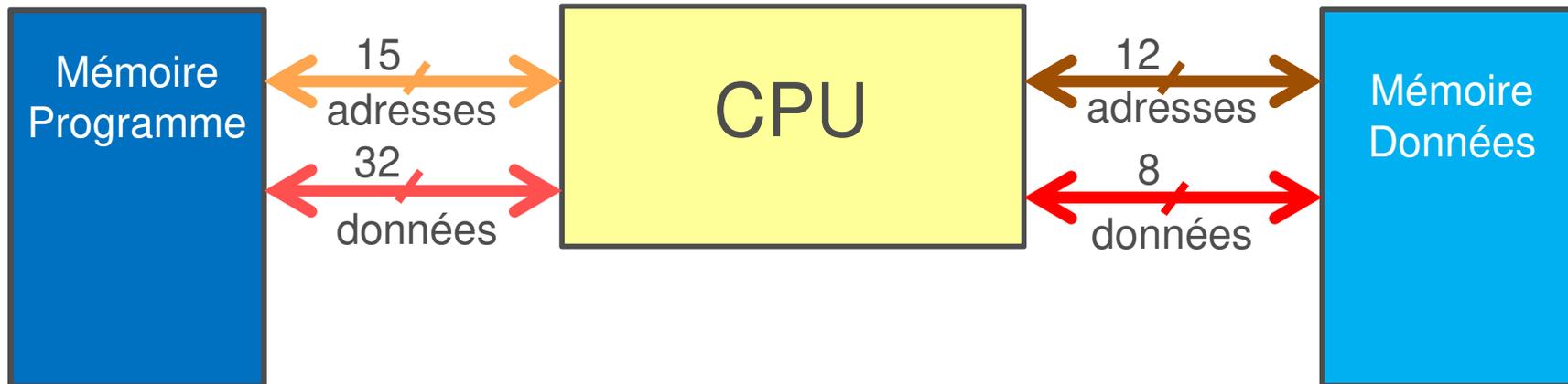
- Boutons, clavier
- Afficheurs, voyants
- Actionneurs
- Capteurs
- Communication avec d'autres systèmes



Architecture simplifiée type « Von Neumann »



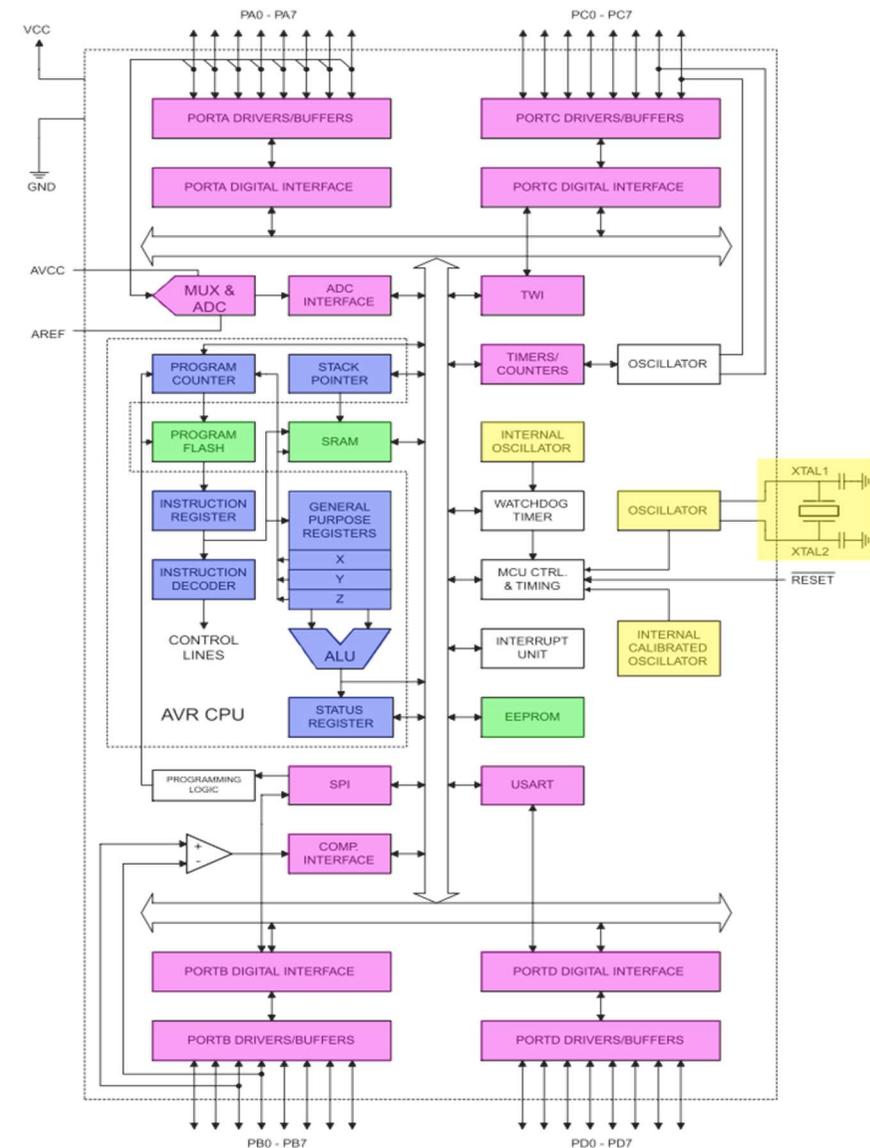
Architecture simplifiée type « Harvard »



Carte d'étude M168 : P90CE201

Composants d'un Microcontrôleur classique

- CPU (microprocesseur)
- Mémoires
 - EEPROM
 - SRAM
 - Flash
- Périphériques
 - Ports Entrées/Sorties Logiques
 - Ports Entrées/Sorties Analogiques
 - Timers, Compteurs
 - Chien de garde (watchdog)
 - Interfaces de communication série (UART, I2C, CAN, USB, etc.)
- Contrôleur d'interruptions
- Oscillateur
- Autres



Généralités

- PSoC: Programmable System on Chip
- Fabricant Cypress
- Architecture unique au monde
- Puissant
- Faible consommation

Constitution

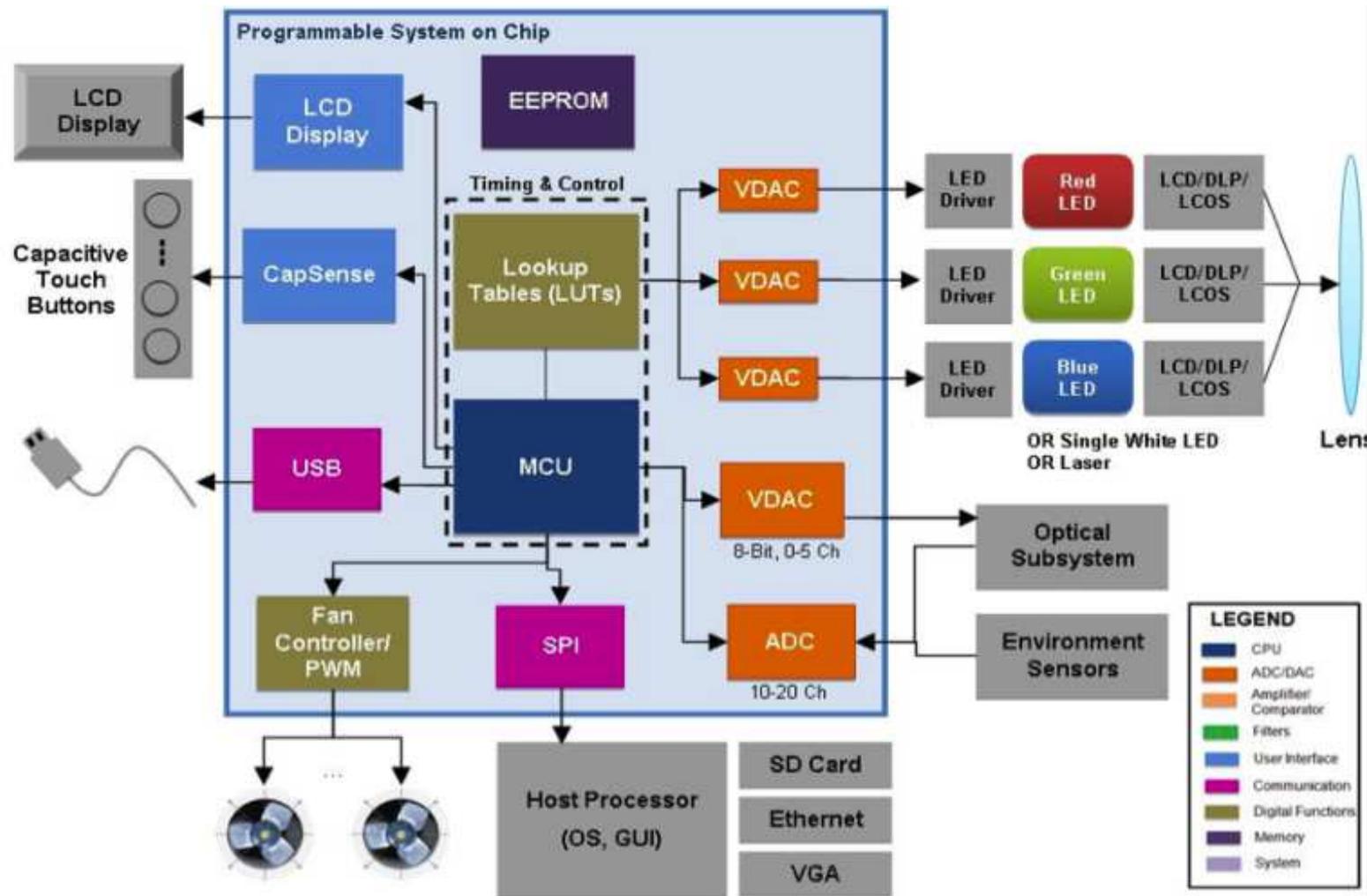
- Microcontrôleur 8 ou 32 bits
- Mémoire Flash, SRAM, EEPROM
- Structures logiques et analogiques programmables, configurables et interconnectables

Bibliothèque de fonctions logiques et analogiques

- ADC, DAC, UART, SPI, Timer, PWM, CAN...
- Contrôleurs de cartes mémoires (SD, miniSD...)
- Amplificateurs à gain programmable
- Amplificateurs opérationnels et d'instrumentation
- Comparateurs, filtres etc...



Exemple d'application: Vidéoprojecteur à leds



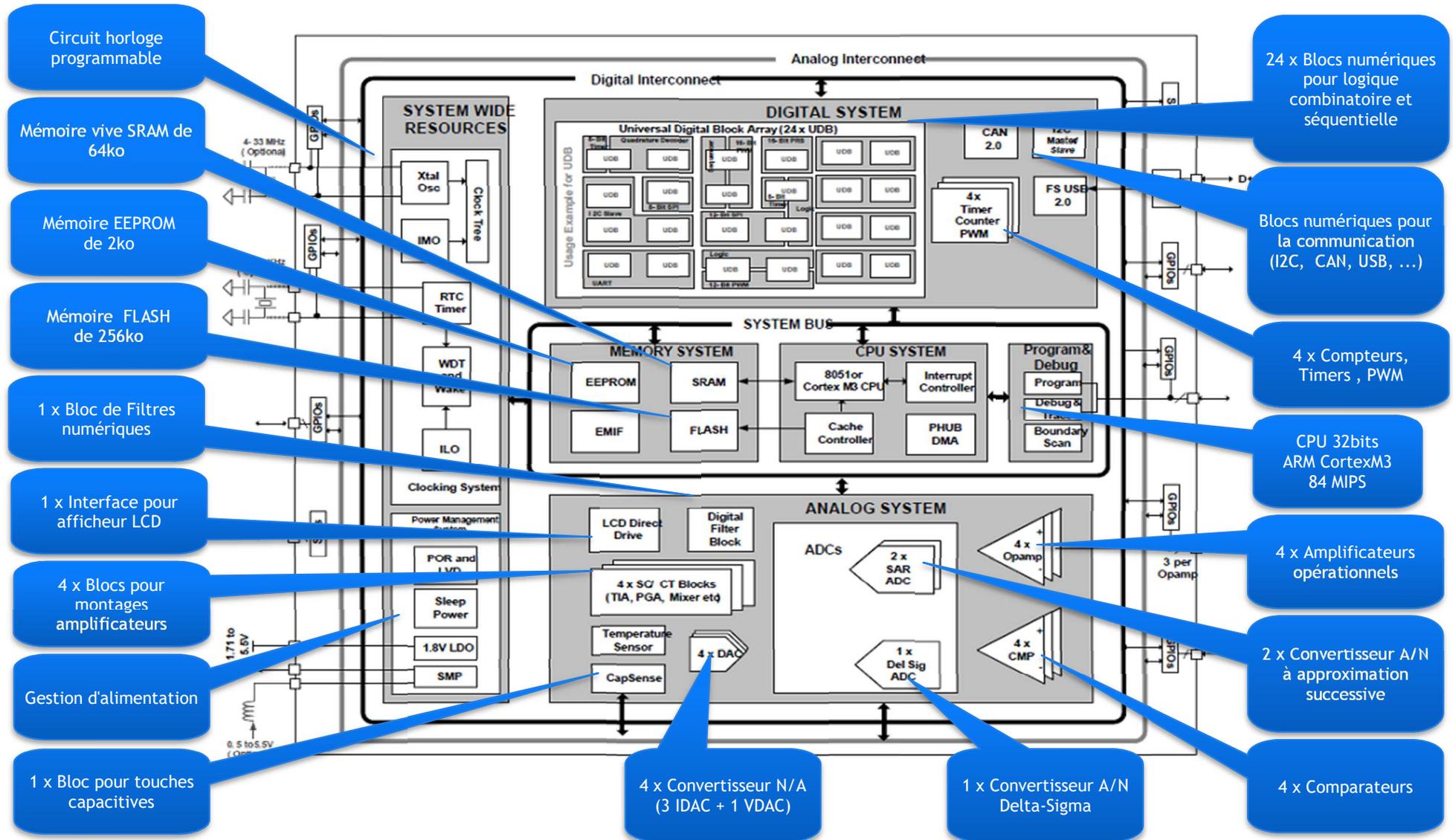
Dans cette application le microcontrôleur PSoC permet de :

- Contrôler l'intensité dans les LEDs
- Contrôler la vitesse des ventilateurs
- Mesurer la température
- Assurer le dialogue entre l'appareil et l'utilisateur (Touches capacitives et afficheur LCD)
- Communiquer avec le processeur hôte
- Assurer la liaison USB

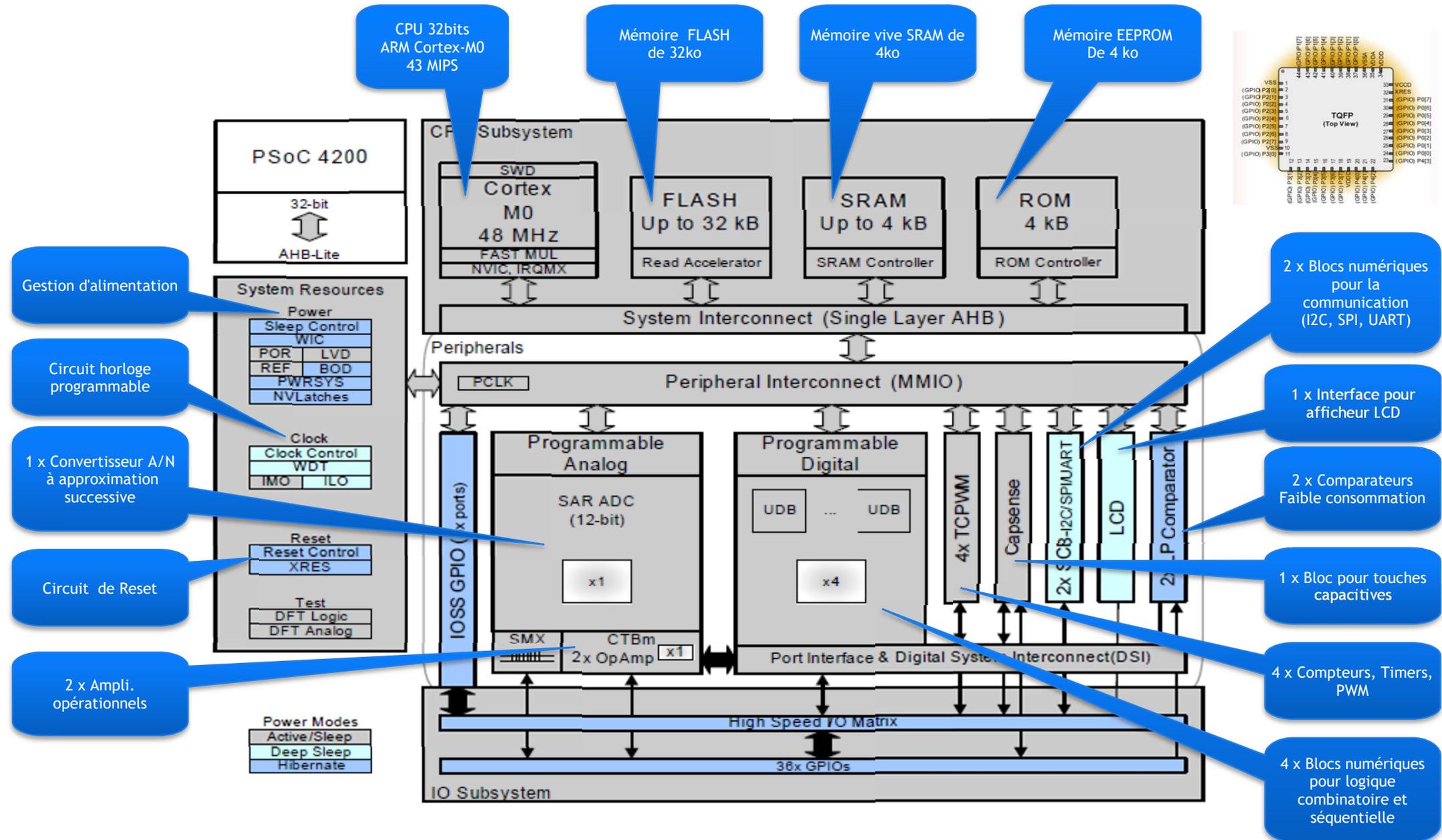
Caractéristiques principales des différentes familles

	PSoC 1	PSoC 3	PSoC 4	PSoC 5LP
Vitesse	24 MHz pour 4 MIPS	67 MHz pour 33 MIPS	24/48 MHz pour 43 MIPS	80 MHz pour 100 MIPS
Bus de données	8 bits	8 bits	32 bits	32 bits
Mémoire programme	Flash de 4 Kio à 32 Kio	Flash de 8 Kio à 64 Kio	Flash de 16 Kio à 32 Kio	Flash de 32 Kio à 256 Kio
Mémoire de données	SRAM de 256 o à 2 Kio	SRAM de 2 Kio à 8 Kio	SRAM de 4 Kio	SRAM de 16 Kio à 64 Kio
Alimentation	1,7 V à 5,25 V	0,5 V à 5,5 V	1,71 V à 5,5 V	de 1,71 V à 5,5 V
Consommation	Actif : 2 mA Veille : 3 µA	Actif : 0,8mA Veille : 1 µA Hibernation : 200 nA	Actif : 6,7mA à F=24MHz Veille : 1,7 mA Veille prolongée : 15µA Hibernation : 150 nA Arrêt : 20 nA	Actif : 3,1 mA à 6 MHz Veille : 2 µA hibernation : 300 nA
Conversion A/N	1 Delta-Sigma de 14 bits	1 Delta-Sigma de 20 bits	1 SAR de 12 bits	1 Delta-Sigma de 20 bits 2 SAR de 12 bits
Conversion N/A	2 de 8 bits	jusqu'à 4 de 8 bits	2 de 7 et 8 bits à sortie courant	jusqu'à 4 de 8 bits
Communication	USB 2.0, I2C, SPI, UART, LIN	USB 2.0, I2C, SPI, UART, CAN, LIN, I2S, JTAG	I2C, SPI, UART	USB 2.0, I2C, SPI, UART, LIN, I2S
Entrées/Sorties	64	72	32	72

Architecture interne PSoC 5LP



Architecture interne PSoC 4200



Cy8CKIT-001

- PSoC 1, 3, 4 et 5
- Boutons poussoirs
- LCD, LEDs
- Capsense
- RS232, USB, etc.
- Wireless (extension)
- Potentiomètre
- zone de prototypage rapide



Cy8CKIT-050

- PSoC 5, 5LP
- Boutons poussoirs
- LCD, LEDs
- Capsense
- RS232, CAN, I2C, USB
- Potentiomètre
- zone de prototypage rapide



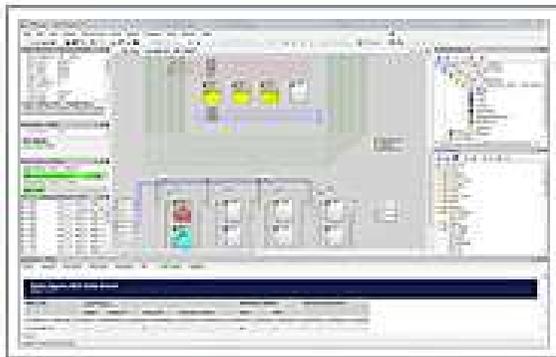
CY8CKIT-042

- PSoC 4200
- Bouton Poussoir
- Led RGB
- Capsense slider
- Compatible shields Arduino
- Compatible PMOD

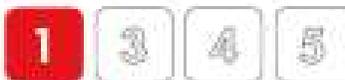


PSoC® Designer™

Integrated Design Environment



Supports
PSoC



PSoC 1

- Interconnexions matricielles entre blocs analogiques et logiques
- Programmation séquentielle définie en langage C

PSoC® Creator™

Integrated Design Environment



Supports
PSoC

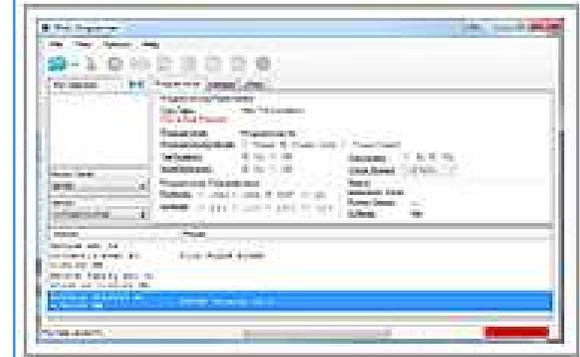


PSoC 3, 4, 5 et 6

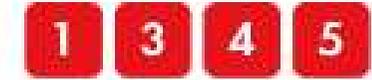
- Logiciel de saisie de schéma reliant les fonctions analogiques et logiques
- Programmation séquentielle assurée par un compilateur C

PSoC® Programmer™

Integrated Programming Application



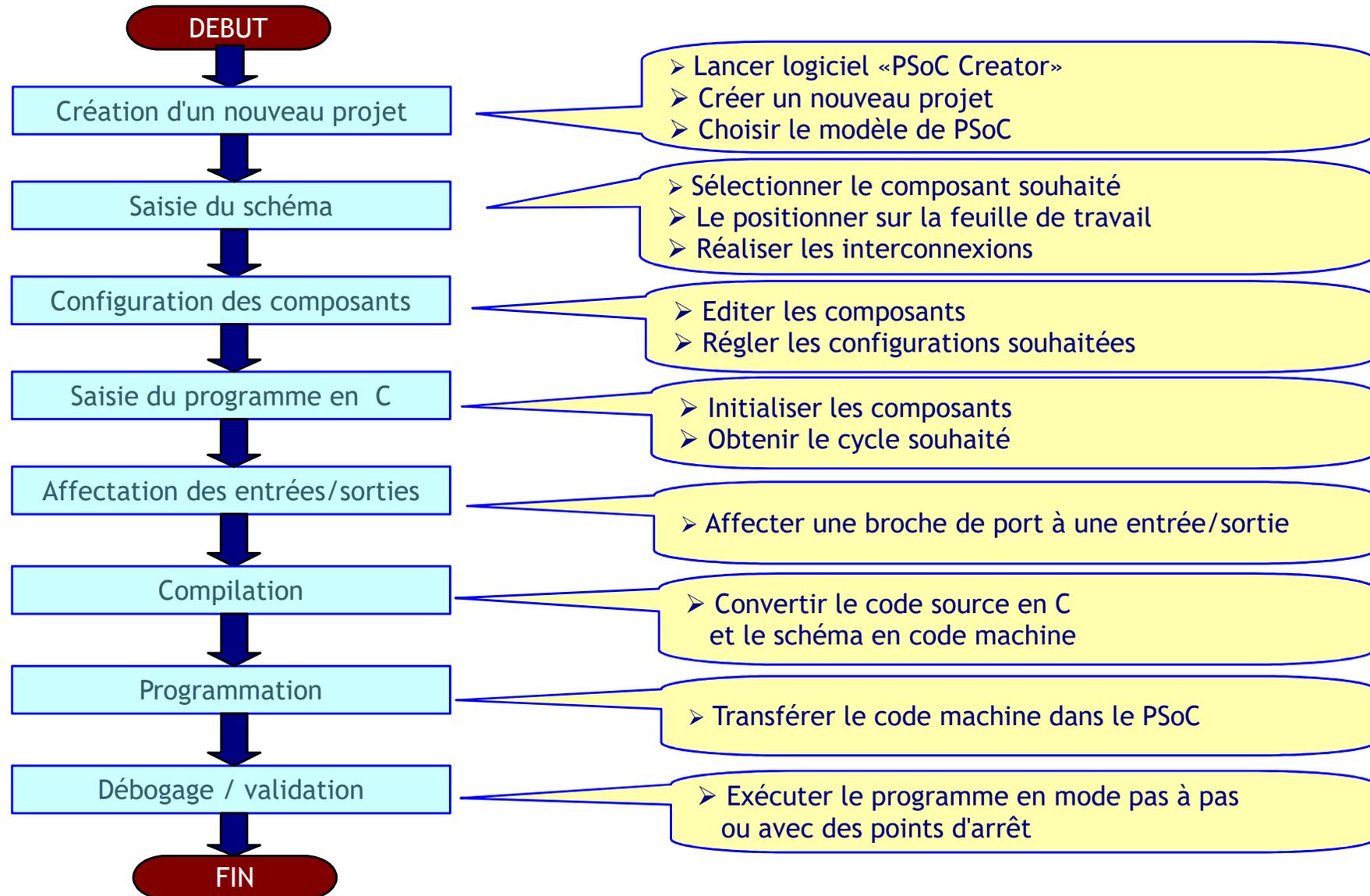
Supports
PSoC



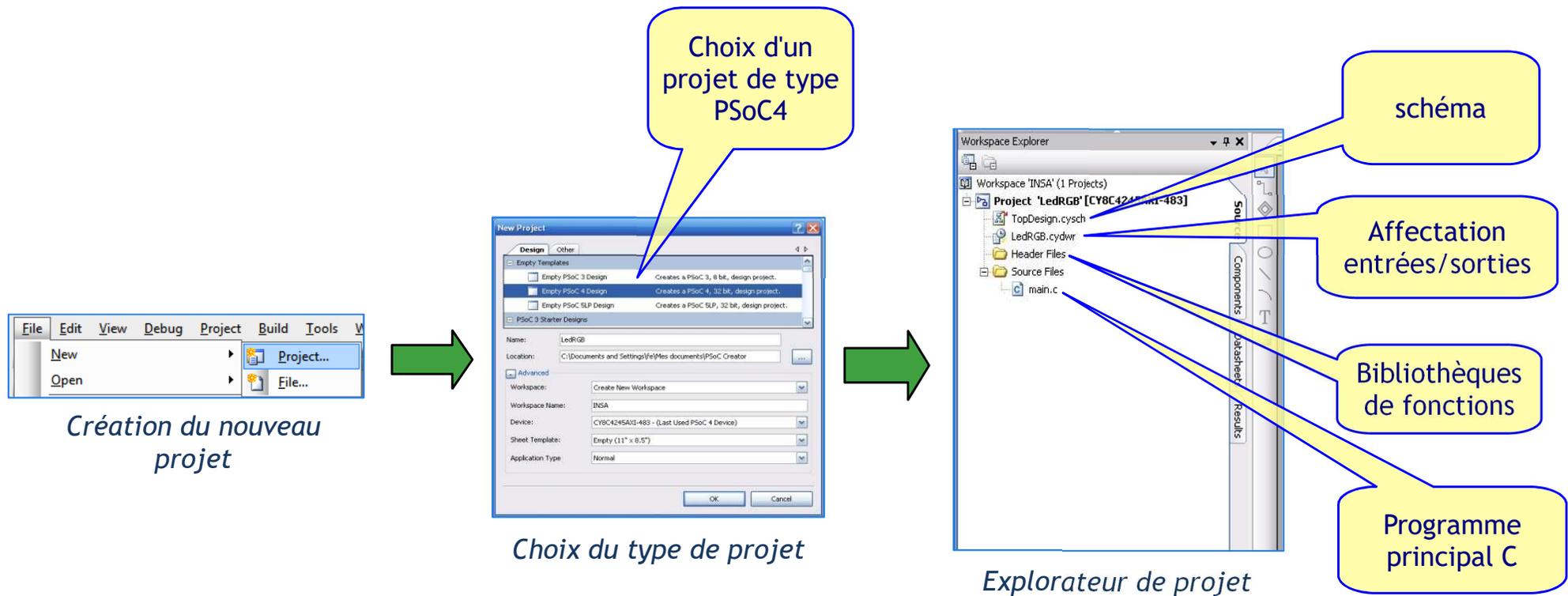
PSoC 1, 3, 4, 5 et 6

- Logiciel permettant de programmer et de configurer les PSoC sur cible

Démarche de développement d'un projet

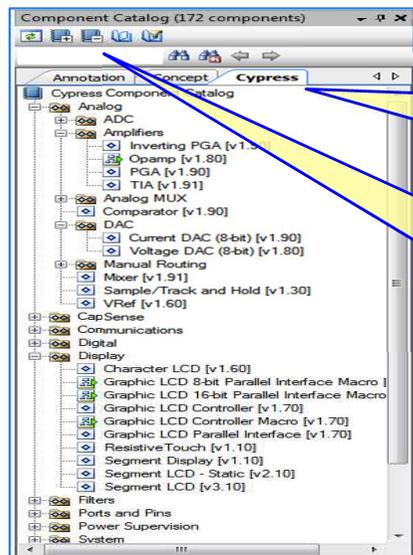


Création d'un nouveau projet



- Un double-clic sur un fichier du projet permet de l'éditer
- Les fichiers principalement éditables sont:
 - le schéma
 - l'affectation des entrées/sorties
 - le code source en C

Saisie du schéma et configuration des composants



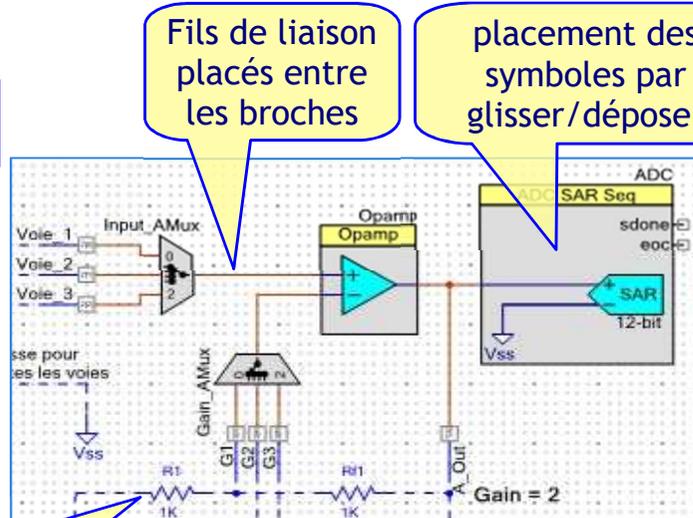
Catalogue des composants

Onglet pour les symboles des composants

Onglet pour les symboles d'annotation (off chip)



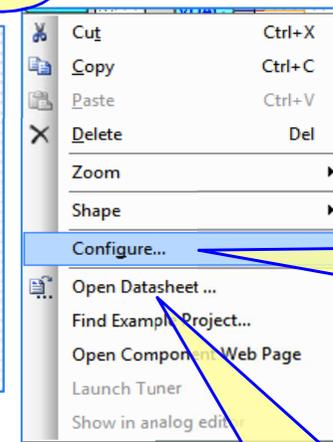
Schéma d'annotation en bleu



Feuille de schéma

Fils de liaison placés entre les broches

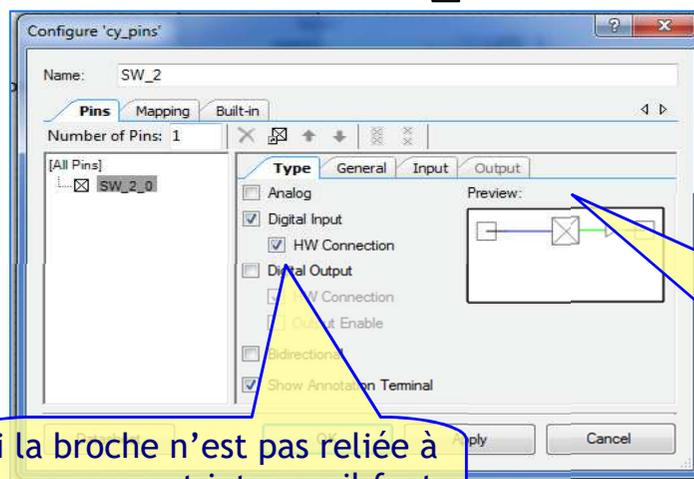
placement des symboles par glisser/déposer



Configuration du composant

Documentation du composant en « pdf »

La configuration sous forme de boîte de dialogue permet de générer automatiquement le code d'initialisation d'un composant.
Par la suite, dans l'édition du programme, il faut appeler la fonction associée dans la partie configuration.

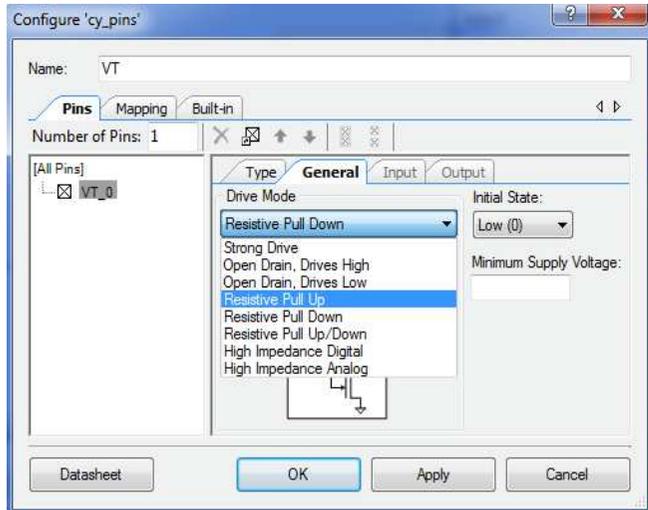


Si la broche n'est pas reliée à un composant interne, il faut décocher la case

La configuration du composant se fait avec une boîte de dialogue disposant:

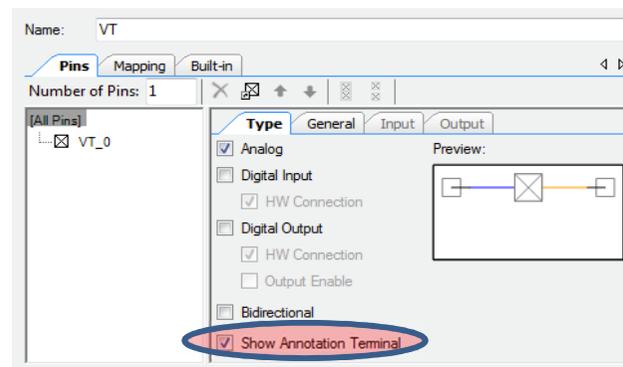
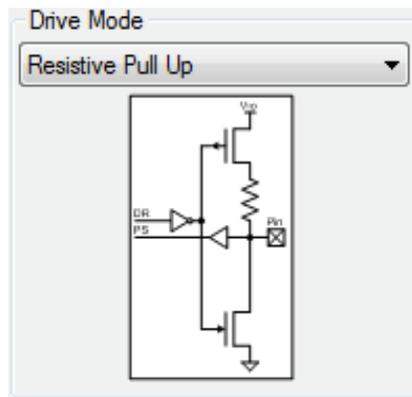
- cases à cocher
- listes déroulantes
- zones de saisie

Configuration des entrées / sorties



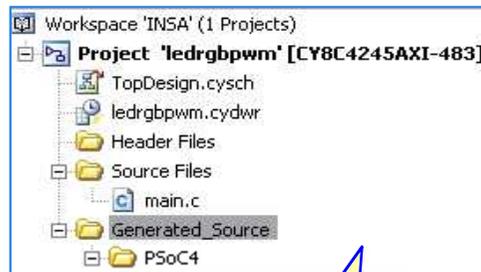
Type de configuration de l'entrée	Choix « Drive mode »
Niveau bas au repos	Resistive pull-down
Niveau haut au repos	Resistive pull-up
Niveau intermédiaire (VDD / 2)	Resistive pull-up / down
Non défini	High impedance digital / analog

Type de configuration de la sortie	Choix « Drive mode »
Imposer le niveau bas	Open drain, drive low
Imposer le niveau haut	Open drain, drive high
Imposer les niveaux haut et bas	Strong



Sur le schéma il est possible de rajouter des symboles d'annotation afin d'indiquer les connexions externes des broches du PSoC. Pour que cela se fasse correctement il est possible de configurer les broches en mode « Show annotation terminal ».

Affectation des entrées/sorties

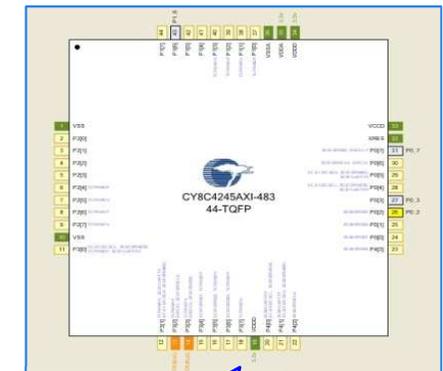


Explorateur de projet

Alias	Name /	Port	Pin	Lock
PO_2	PO[2]	SCB0:SPI:SS3	26	<input checked="" type="checkbox"/>
PO_3	PO[3]		27	<input checked="" type="checkbox"/>
PO_7	PO[7]	SCB1:SPI:SS0, WAKEUP	31	<input checked="" type="checkbox"/>
P_6	P1[6]		43	<input checked="" type="checkbox"/>

Nom de broche
saisie dans le schéma

Référence de port
du PSoC



Plan de brochage
du PSoC

Dans l'explorateur de projet, on sélectionne le fichier ayant l'extension «.cydwr» afin de définir les affectations des broches d'entrées/sorties.

Dans le tableau qui s'affiche, on choisit un port spécifique et distinct pour chaque nom de broche saisie précédemment sur le schéma.

On peut observer cette affectation sur le plan de brochage du PSoC.

Saisie du code source

```

17 #define IC 3600
18 int VITESSE=0;
19
20 /* INTERRUPTION COMPTEUR 1 */
21 CY_ISR(IntCounter2)
22 {
23     LED1_Write(1);
24     VITESSE=Counter_1_ReadCounter();
25     Counter_1_WriteCounter(0);
26     LED1_Write(0);
27 }
28
29 void main()
30 {
31     //Initialisation compteur 1 et 2
32     Counter_1_Start();
33     Counter_2_Start();
34     Counter_2_WritePeriod(1000);
35
36     //Initialisation affichage
37     LCD_Char_1_Start();
38     LCD_Char_1_PrintString("***ANEMOMETRE***");
39     LCD_Char_1_Position(1,0);
40     LCD_Char_1_PrintString("*** INSA-CVL ***");
41     CyDelay(1000);
42     LCD_Char_1_ClearDisplay();
43     LCD_Char_1_PrintString("V :    km/h    ");
44
45     // Initialisation interruption compteur 2
46     isr_1_Start();
47     isr_1_Disable();
48     isr_1_SetVector(IntCounter2);
49     isr_1_Enable();
50     CyGlobalIntEnable;
51
52     for(;;)
53     {
54         LCD_Char_1_Position(0,4);
55         LCD_Char_1_PrintString("    ");
56         LCD_Char_1_Position(0,4);
57         LCD_Char_1_PrintNumber(VITESSE);
58         CyDelay(100);
59     }
60 }
    
```

ss-prg d'interruption lié à un composant (timer)
Mot clé associé est «CY_ISR» avec entre parenthèses le nom du vecteur

Initialisation des composants configurés dans le schéma grâce aux boîtes de dialogue

Instructions exécutées une seule fois dans la durée de vie du programme

Initialisation interruptions

Boucle principale

L'interruption est un programme s'exécutant lors d'un événement particulier comme le changement d'état d'une entrée ou le débordement d'un compteur

Génération de l'application

- Elaboration de la conception
- Génération du code HDL
- Synthèse des structures logiques
- Mise en place, routage et génération de code des modules analogiques
- Placement, routage et génération de code des modules logiques
- Génération API
- Compilation



Icone permettant de lancer la compilation

Compilateurs supportés

- PSoC 3: Cypress-Edition Keil™ CA51 Compiler Kit
- PSoC 4, 5 et 6: GNU/CodeSourcery Sourcery G++™ Lite
- Aucune restriction d'utilisation

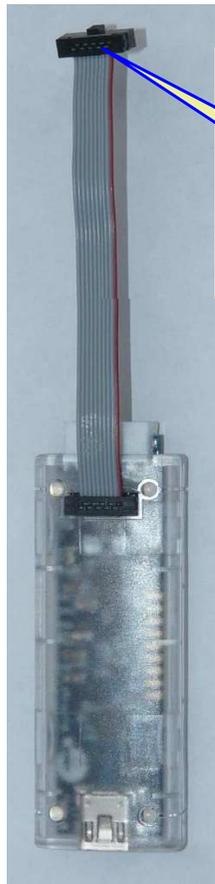


```
Output
Show output from: All
arm-none-eabi-gcc.exe -I. -I./Generated_Source/PSoC4 -Wno-main -mcpu=cortex-m0 -mthumb -Wall -g -Wa,-alh=.\\CortexMO\\ARM_GCC_473\\Debug\\Vc
arm-none-eabi-gcc.exe -I. -I./Generated_Source/PSoC4 -Wno-main -mcpu=cortex-m0 -mthumb -Wall -g -Wa,-alh=.\\CortexMO\\ARM_GCC_473\\Debug\\Vc
arm-none-eabi-gcc.exe -I. -I./Generated_Source/PSoC4 -Wno-main -mcpu=cortex-m0 -mthumb -Wall -g -Wa,-alh=.\\CortexMO\\ARM_GCC_473\\Debug\\Vc
arm-none-eabi-ar.exe -rs .\\CortexMO\\ARM_GCC_473\\Debug\\Acq&AmpliAuto.a .\\CortexMO\\ARM_GCC_473\\Debug\\cyfitter_cfg.o .\\CortexMO\\ARM_GCC_473\\
arm-none-eabi-ar.exe: creating .\\CortexMO\\ARM_GCC_473\\Debug\\Acq&AmpliAuto.a
arm-none-eabi-gcc.exe -mthumb -march=armv6-m -T .\\Generated_Source\\PSoC4\\cm0gcc.ld -g -Wl,-Map,.\\CortexMO\\ARM_GCC_473\\Debug\\Acq&AmpliAuto
cyelftool.exe -C "C:\\Documents and Settings\\fe\\Mes documents\\PSoC Creator\\Design03\\Acq&AmpliAuto.cydsn\\CortexMO\\ARM_GCC_473\\Debug\\Acq&Ampli
cyelftool.exe -S "C:\\Documents and Settings\\fe\\Mes documents\\PSoC Creator\\Design03\\Acq&AmpliAuto.cydsn\\CortexMO\\ARM_GCC_473\\Debug\\Acq&Ampli
Flash used: 6342 of 32768 bytes (19,4 %).
SRAM used: 1572 of 4096 bytes (38,4 %).
----- Build Succeeded: 04/23/2014 22:15:36 -----
```

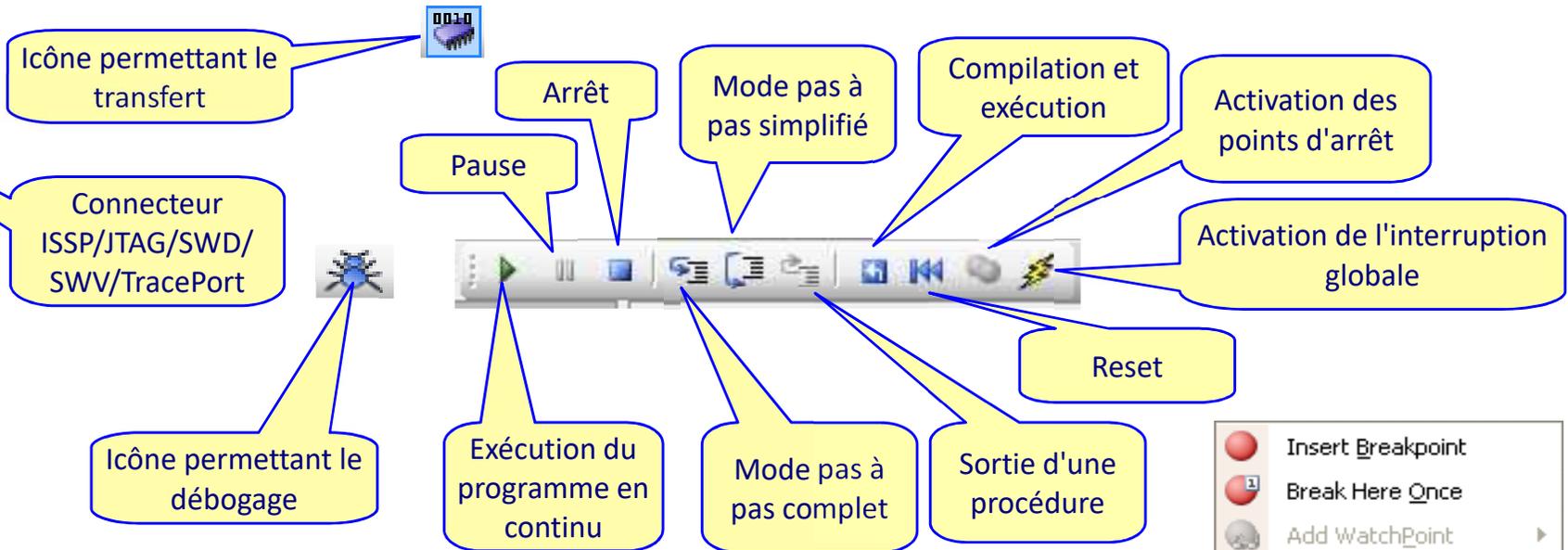
La compilation est une opération relativement longue pour le PSoC. Cette durée est due à la configuration des composants qui correspond à un routage des différents blocs élémentaires présents dans le PSoC.

Zone de message indiquant le résultat de la compilation

Transfert du programme et débogage



MiniProg3

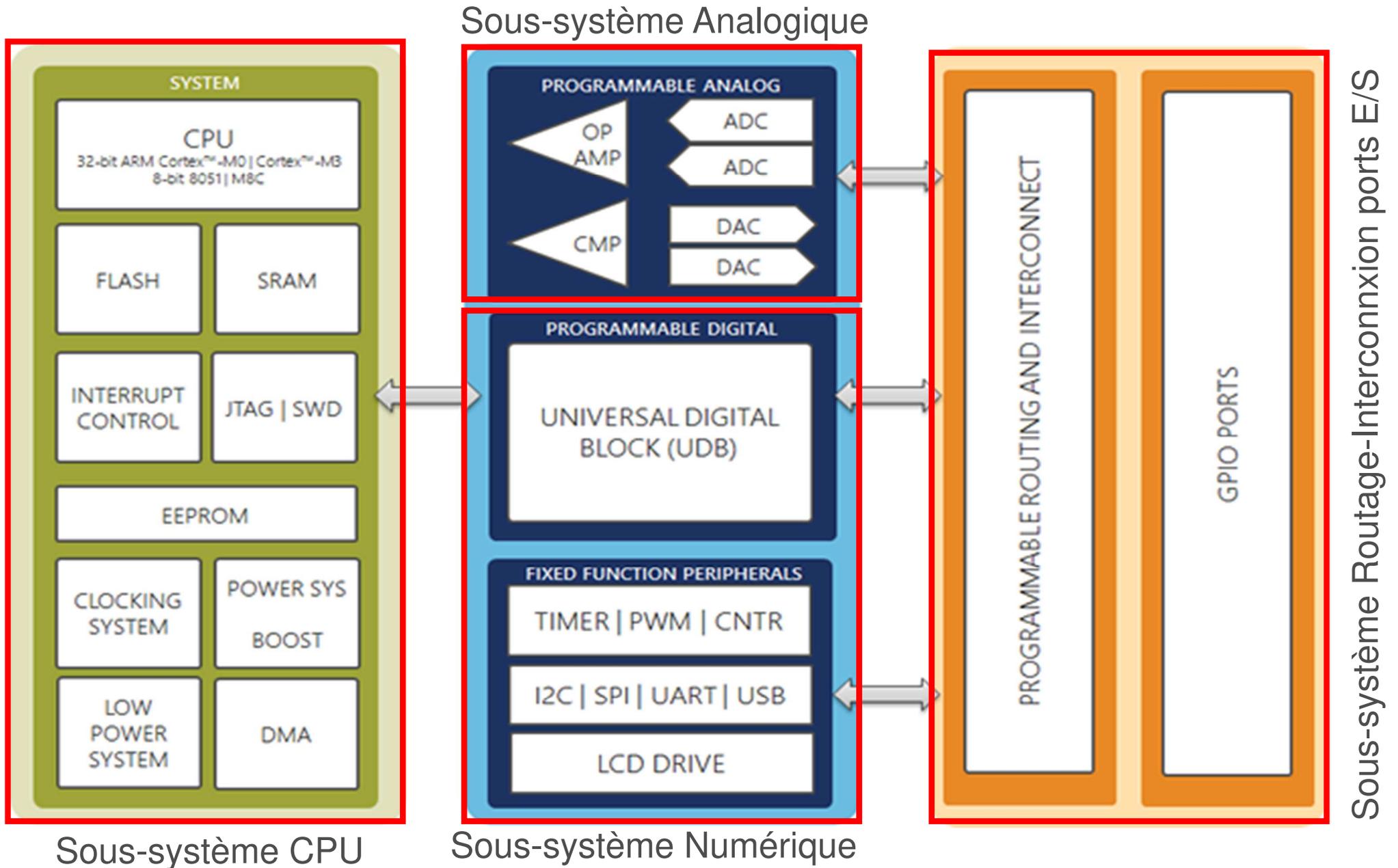


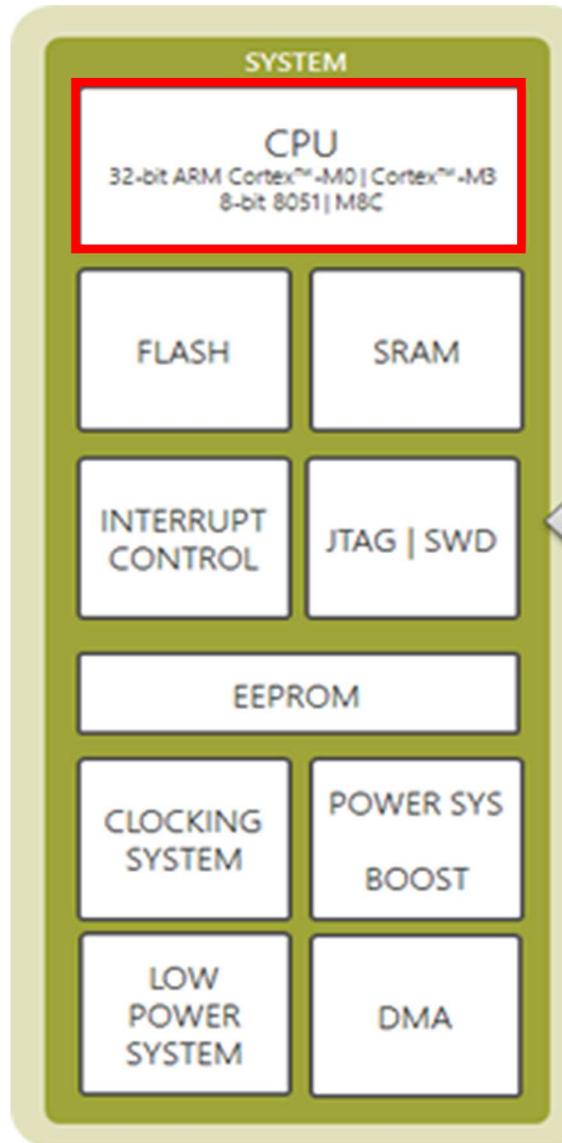
Locals				
Name	Value	Address	T...	Radix
period	0x0	000000F4 (XData)	int	Default
duty	0x4	000000F6 (XData)	int	Default

	Insert <u>B</u> reakpoint
	Break Here <u>O</u> nce
	Add <u>W</u> atchPoint
	Add <u>W</u> atch
	<u>R</u> un To Cursor
	<u>S</u> et Next Instruction
	<u>U</u> ndo Ctrl+Z
	<u>R</u> edo Ctrl+Y
	<u>C</u> ut Ctrl+X
	<u>C</u> opy Ctrl+C
	<u>P</u> aste Ctrl+V
	<u>D</u> elete Del
	Select <u>A</u> ll Ctrl+A

Le débogage consiste à exécuter le programme en mode pas à pas ou avec des points d'arrêt

Architecture générale PSoC





ARM Cortex-M3

- CPU leader dans l'industrie de l'embarqué
- Exemples d'application et de code très répandus
- Fréquence: 80 MHz; 100 DMIPS (PSoC 5LP)
- Architecture ARM v7:
 - Jeu d'instructions Thumb2
 - Instructions sur 16 et 32 bits
 - ALU sur 32 bits; Multiplication et division matérielle
- 1 cycle d'horloge à 3-stage pipeline; Architecture Harvard

ARM Cortex-M0

- Fréquence: 48 MHz; 43 DMIPS
- Architecture ARM v6

8051

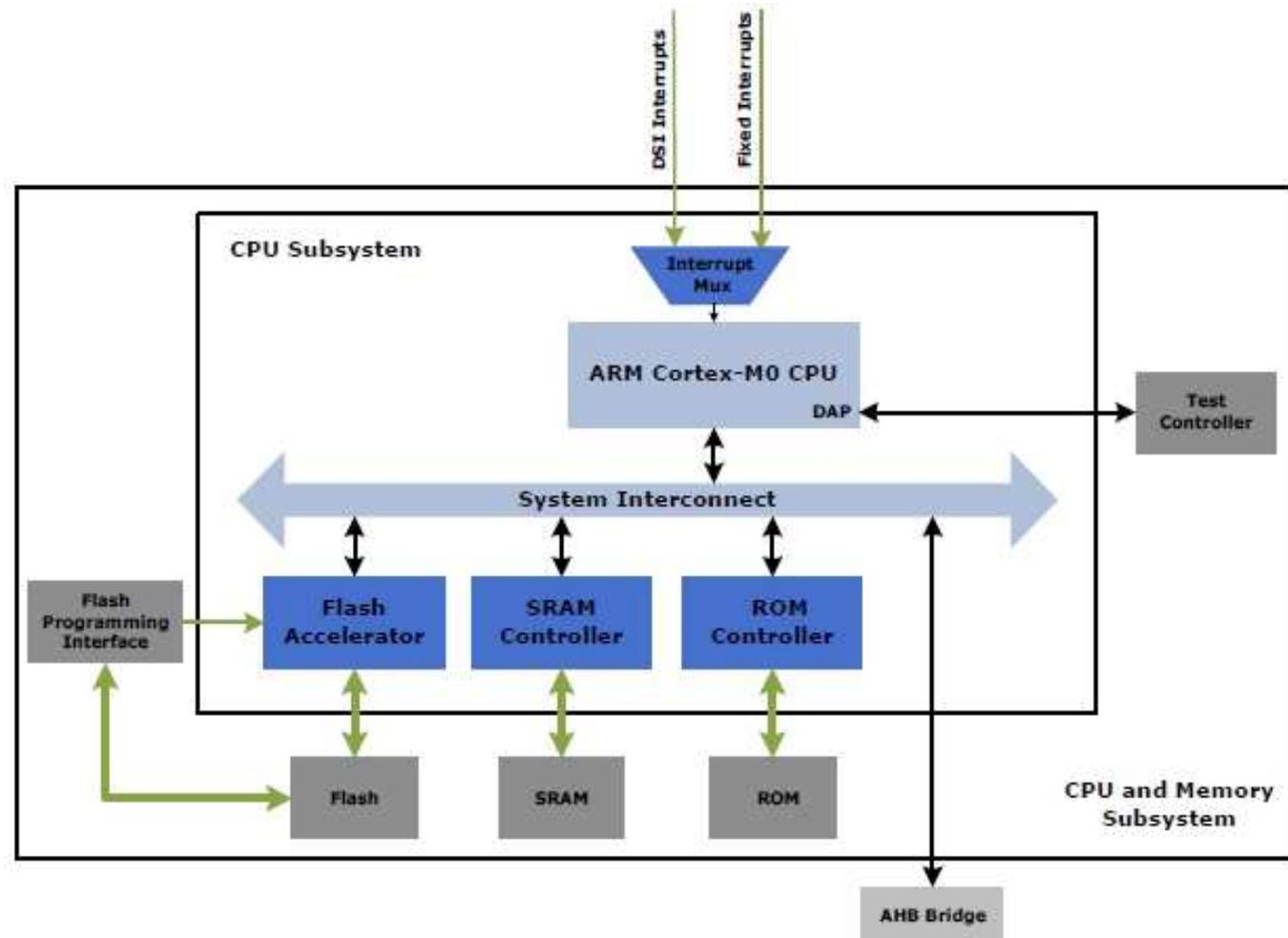
- Fréquence: 67 MHz; 33 MIPS
- Exécution des instructions en 1 cycle d'horloge

ARM Cortex-M0

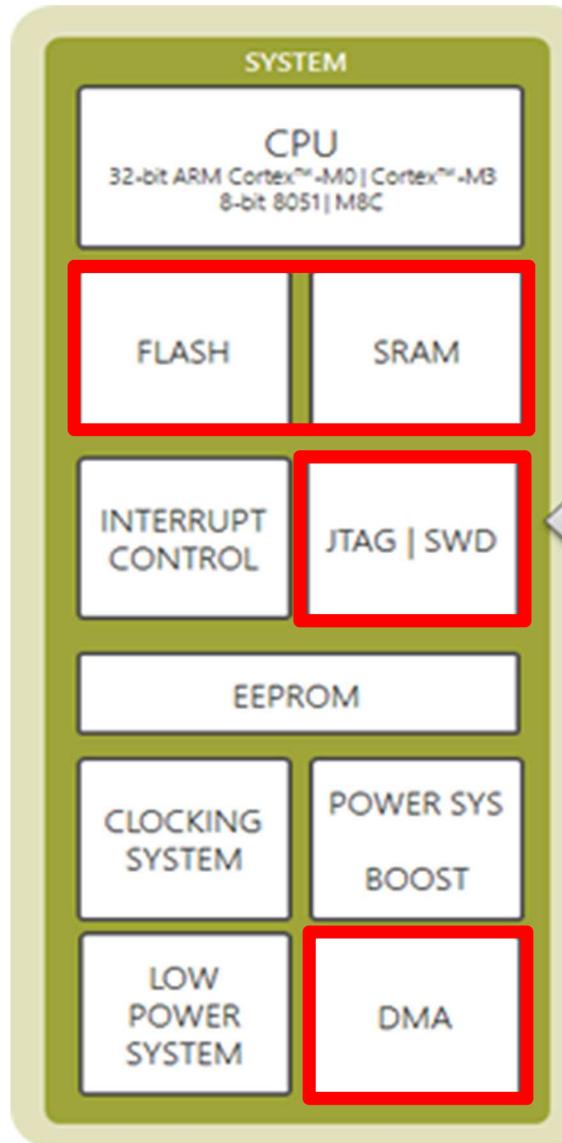
- CPU: 32 bits
- RISC
- Fréquence: 48 MHz
- Performance: 0,9 DMIPS/MHz
- Architecture ARM v6
 - Instructions 16 et 32 bits
 - Jeu d'instructions Thumb et 2
- 3-stage pipeline (1 cycle horloge)
- Espace adressable: 4 Go
- Architecture Harvard
- Faible consommation

Mémoires

- Flash: 16 à 32 ko
- SRAM: 4 ko
- ROM: 4 ko



Architecture générale sous-système CPU PSoC 4



Mémoire haute performance

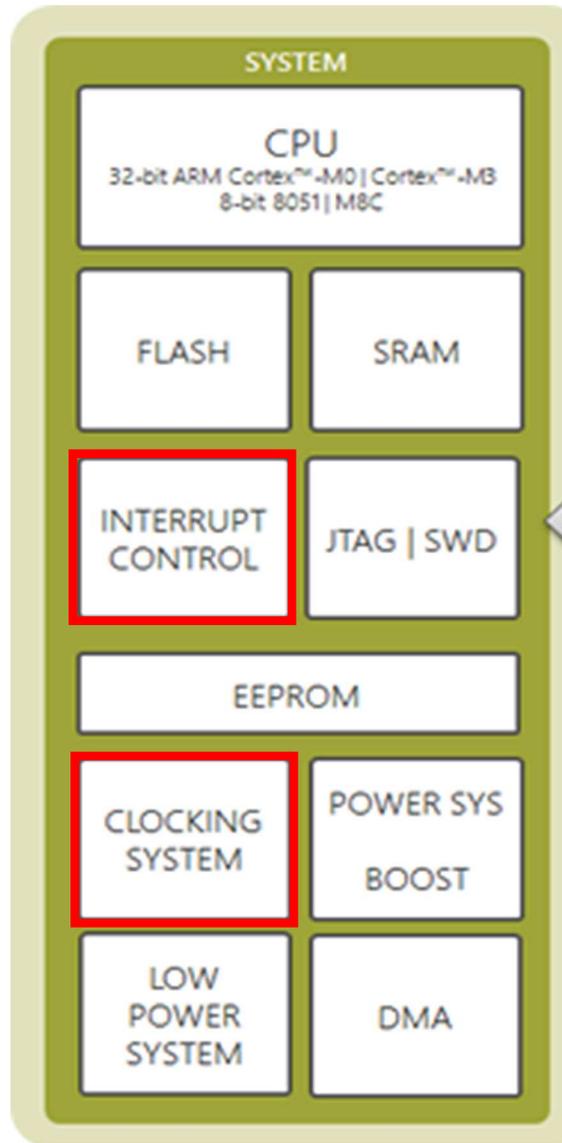
- Mémoire Flash avec ECC
- Transfert rapide SRAM vers flash
- EEPROM

Système performant DMA

- DMA à 24 canaux
- Transfert par DMA pour tous les périphériques Logiques et Analogiques
- Accès simultané à la SRAM par la CPU et le contrôleur DMA

Mise au point (Debug) sur la puce

- Standard JTAG/SWD (Serial Wire Debug)
- Traçage direct sur la puce
- ICE inutile



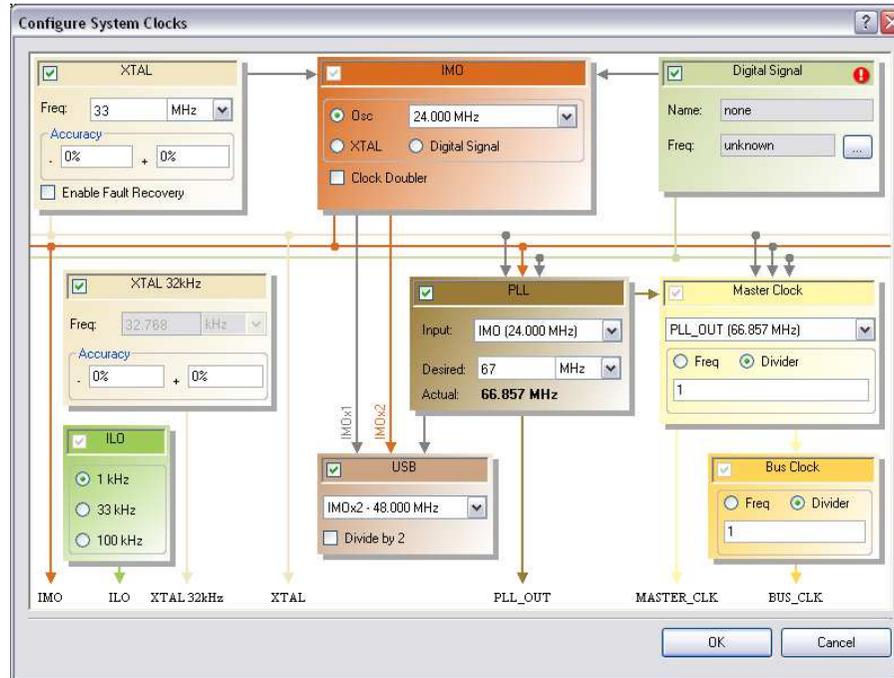
Système d'horloge

- Nombreuses sources d'horloge
 - Oscillateur principal interne
 - Entrée d'horloge externe à base d'un quartz
 - Entrée d'horloge externe à base d'un oscillateur
 - Sortie d'un doubleur d'horloge
 - Oscillateur interne basse fréquence
 - Entrée d'horloge externe à base d'un quartz 32kHz
 - Horloge dédiée USB à 48 MHz
 - Sortie PLL
- Diviseurs de fréquence 16 bits
 - 8 numériques
 - 4 Analogiques
- Assistant de configuration PSoC Creator

Contrôleur d'interruption

- 32 lignes d'interruption
- 8 niveaux d'interruption par ligne (4 pour PSoC 4)

Configuration dans Psoc Creator

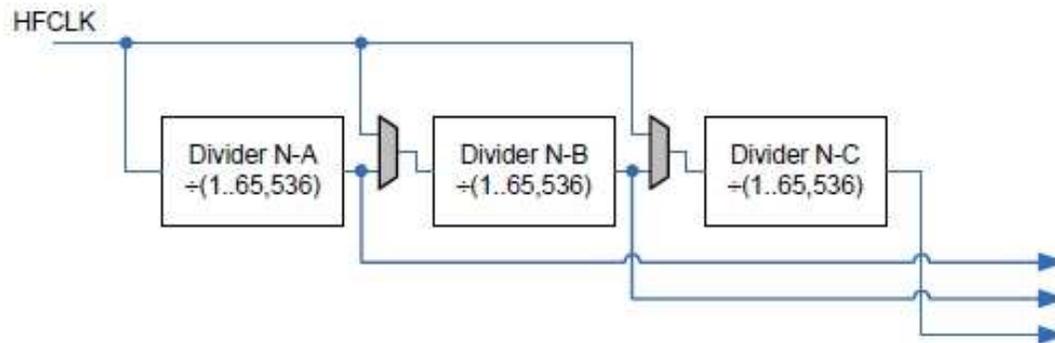
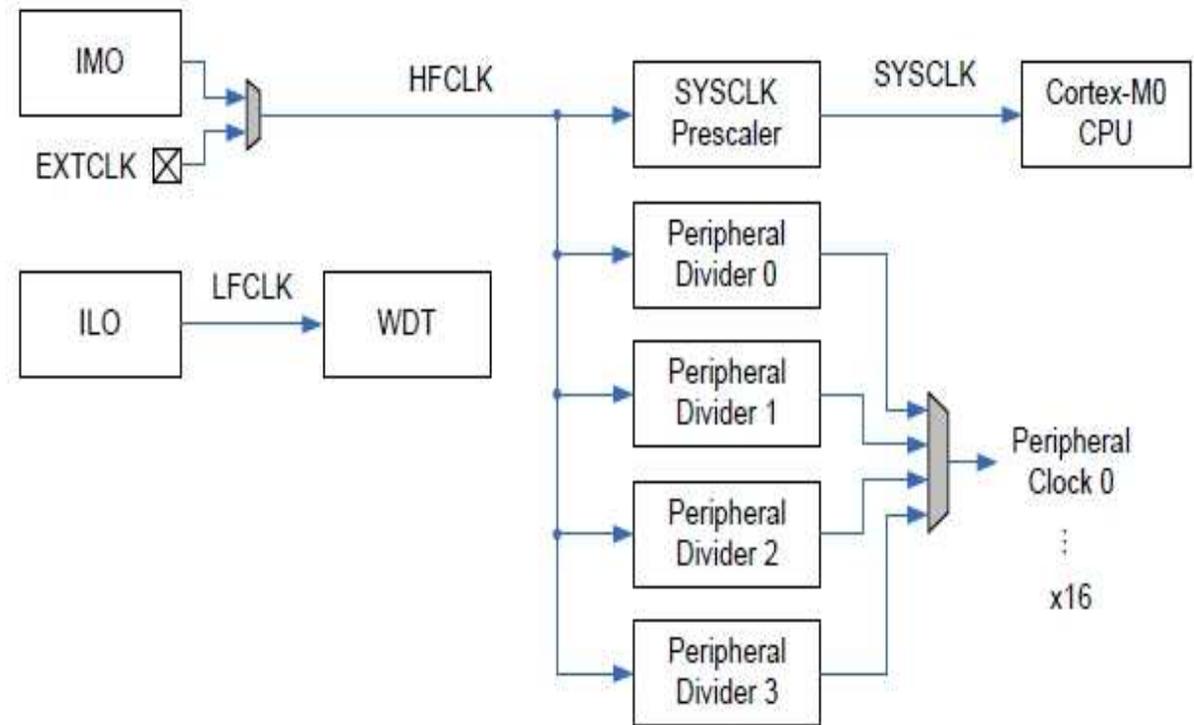


Type	Name	Domain	Desired Frequency	Nominal Frequency	Accuracy (%)	Tolerance (%)	Divider	Start on Reset	Source Clock
System	Digital_Signal	DIGITAL	? MHz	? MHz	±0	-	0	<input type="checkbox"/>	
System	ILO	DIGITAL	? MHz	1.000 kHz	±20	-	0	<input checked="" type="checkbox"/>	
System	XTAL_32KHZ	DIGITAL	32.768 kHz	32.768 kHz	±0.000999999974737875	-	0	<input checked="" type="checkbox"/>	
System	IMO	DIGITAL	? MHz	24.000 MHz	±4	-	0	<input checked="" type="checkbox"/>	
System	XTAL	DIGITAL	33.000 MHz	33.000 MHz	±0.0009999999974737875	-	0	<input checked="" type="checkbox"/>	
System	USB_CLK	DIGITAL	48.000 MHz	48.000 MHz	±4	-	1	<input checked="" type="checkbox"/>	IM0x2
System	BUS_CLK	DIGITAL	? MHz	66.857 MHz	±4	-	1	<input checked="" type="checkbox"/>	MASTER_CLK
System	MASTER_CLK	DIGITAL	? MHz	66.857 MHz	±4	-	1	<input checked="" type="checkbox"/>	PLL_OUT
System	PLL_OUT	DIGITAL	67.000 MHz	66.857 MHz	±4	-	0	<input checked="" type="checkbox"/>	IMO
Local	Clock_1	DIGITAL	24.000 MHz	24.000 MHz	±4	±5	1	<input checked="" type="checkbox"/>	Auto: IMO

Ressources

- 2 sources internes:
 - IMO de 3 à 48 MHz
 - ILO à 32 kHz
- EXTCLK: Entrée d'horloge externe
- HFCLK: Horloge haute fréquence
 - IMO ou EXTCLK
- SYSCLK: horloge du processeur

- 4 x blocs diviseurs de fréquence comportant chacun 3 compteurs de 16 bits



Interruptions - Exceptions

- Interruption:
 - Evénements liés au matériel
(*Timers, CAN, Ligne E/S, etc.*)
- Exception:
 - Evénements liés à la CPU
(*division par zéro, adressage mémoire, etc.*)
- Contrôleur d'interruption (NVIC) intégré au processeur
- 32 lignes d'interruption
- Temps de latence très court
- Table des vecteurs
- 4 niveaux de priorité par interruption (0 à 3)
- Déclenchement par niveau ou par impulsion sur front montant ou descendant

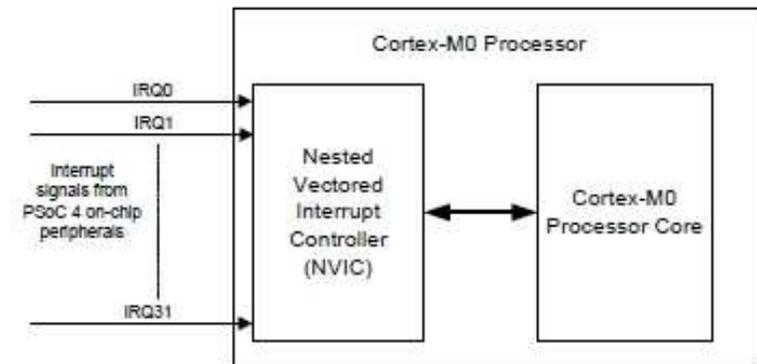
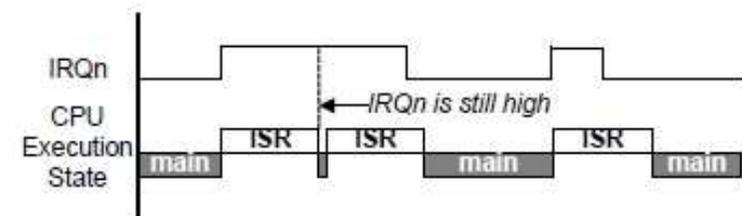
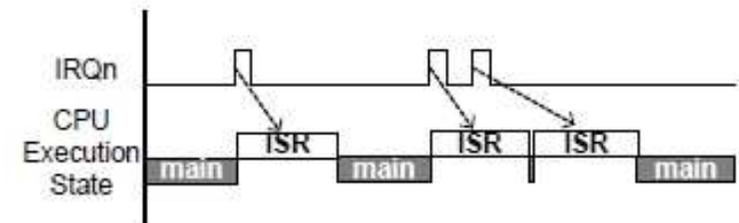


Schéma de principe des interruptions



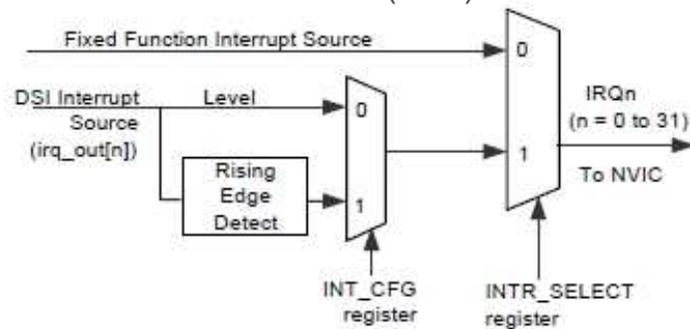
Déclenchement par niveau



Déclenchement par impulsion

Sources des interruptions

- 2 sources possibles pour chaque ligne d'interruption
- Périphériques internes
- Lignes issues du système d'interconnexion (DSI)



Multiplexage des sources d'interruption

Etats des interruptions

- Inactive
- Pending (en attente d'être prise en compte)
- Active (prise en compte par SP-IT, acquittement)
- Active & Pending

Interrupt No.	Cortex-M0 Exception No.	Fixed Function Interrupt Source	DSI Interrupt Source
NMI	2	-	irq_out[0]
IRQ0	16	GPIO Interrupt - Port 0	irq_out[1]
IRQ1	17	GPIO Interrupt - Port 1	irq_out[1]
IRQ2	18	GPIO Interrupt - Port 2	irq_out[2]
IRQ3	19	GPIO Interrupt - Port 3	irq_out[3]
IRQ4	20	GPIO Interrupt - Port 4	irq_out[4]
IRQ5	21	<DSI-only>	irq_out[5]
IRQ6	22	<DSI-only>	irq_out[6]
IRQ7	23	<DSI-only>	irq_out[7]
IRQ8	24	LPCOMP (low-power comparator)	irq_out[8]
IRQ9	25	WDT (Watchdog timer)	irq_out[9]
IRQ10	26	SCB1 (Serial Communication Block 1)	irq_out[10]
IRQ11	27	SCB2 (Serial Communication Block 2)	irq_out[11]
IRQ12	28	SPC (System Performance Controller)	irq_out[12]
IRQ13	29	PWR (Power Manager)	irq_out[13]
IRQ14	30	SAR (Successive Approximation ADC)	irq_out[14]
IRQ15	31	CSD (CapSense block counter overflow interrupt)	irq_out[15]
IRQ16	32	TCPWM0 (Timer/Counter/PWM 0)	irq_out[16]
IRQ17	33	TCPWM1 (Timer/Counter/PWM 1)	irq_out[17]
IRQ18	34	TCPWM2 (Timer/Counter/PWM 2)	irq_out[18]
IRQ19	35	TCPWM3 (Timer/Counter/PWM 3)	irq_out[19]
IRQ20	36	<DSI-only>	irq_out[20]
IRQ21	37	<DSI-only>	irq_out[21]
IRQ22	38	<DSI-only>	irq_out[22]
IRQ23	39	<DSI-only>	irq_out[23]
IRQ24	40	<DSI-only>	irq_out[24]
IRQ25	41	<DSI-only>	irq_out[25]
IRQ26	42	<DSI-only>	irq_out[26]
IRQ27	43	<DSI-only>	irq_out[27]
IRQ28	44	<DSI-only>	irq_out[28]
IRQ29	45	<DSI-only>	irq_out[29]
IRQ30	46	<DSI-only>	irq_out[30]
IRQ31	47	<DSI-only>	irq_out[31]