

## Fonaments de Computadors: introducció amb AVR (II)

---

Joan Oliver

Departament de Microelectrònica i Sistemes Electrònics  
Universitat Autònoma de Barcelona  
(Joan.Oliver@uab.es)

## Índex

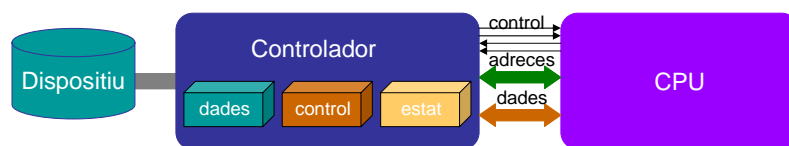
---

- Entrada/sortida
  - Perifèric
  - Entrada/sortida programada
  - Interrupció
  - Augment de prestacions d'E/S: DMA
- Comunicacions i busos
- ATmega8: perifèrics
  - L'entrada/sortida en AVR
  - Ports d'entrada/sortida
  - Timers/comptadors
  - Mòduls analògics/digitals
- Comunicacions
  - Comunicació sèrie
  - Comunicació SPI
- Bibliografia

## Entrada / sortida

- Objectiu de la unitat d'E/S
  - Realitzar la comunicació de perifèrics amb la CPU
- Tema complex en l'ús dels computadors. Ha de tenir en compte...
  - Mecanismes de sincronisme de la CPU amb els perifèrics
  - Interconnexió de perifèrics
  - Procediments del sistema operatiu per operar amb els perifèrics
  - La CPU s'ha de comunicar amb
    - Perifèrics lents. Per exemple la comunicació sèrie
    - Perifèrics ràpids. El tractament amb vídeo requereix comunicacions molt ràpides.
    - De paraula petita. Alguns treballen amb paraules d'1 byte, avui en dia molt petit. Etc
- Segons el tipus de transferència es parla de...
  - Transferència elemental
    - Transferència a la velocitat del sistema d'interconnexió (normalment el bus), no del perifèric
    - S'estableix comunicació física entre perifèric i CPU i es poden transmetre bits, bytes o paraules
    - Pot requerir la conversió a sèrie / paral·lel de dades
    - Es realitza per hardware
  - Transferència de bloc
    - Es transfereix un bloc de dades per sincronisme amb el perifèric
    - Es basa en transferències elementals a requeriment del perifèric
    - pot emmagatzemar temporalment informació
    - Es realitza per software o per DMA
  - Transferència d'E/S
    - Transfereix un o més blocs de dades
    - Gestiona les transferències de blocs i supervisa l'estat del perifèric
    - Pot requerir la conversió de dades
    - Es realitza per software

## Model de perifèric



- Per a comunicar-se amb la CPU el perifèric ha de disposar d'una part electrònica que ha d'adaptar la seva velocitat de procés i format de dades al de la CPU
- El mecanisme d'E/S establert per a cada perifèric ...
  - ... ha de sincronitzar la transferència de dades. Per exemple, la comunicació sèrie requereix de velocitats de transmissió entre 300 i 115200 bauds. La CPU treballa a velocitats molt superiors. mitjançant bucles d'espera i interrupcions s'adequa la velocitat de transmissió entre perifèric i CPU
  - ... ha de controlar el perifèric. Mitjançant els senyals de control i d'estat coneix la situació del perifèric i executa les ordres que cal per realitzar les operacions d'E/S.
    - Els senyals d'estat indiquen si hi ha nova dada, si s'ha acabat l'operació d'E/S, si el dispositiu d'entrada està en disposició d'efectuar transferències, etc.
    - Els senyals de control ordenen accions al perifèric com ara encendre o apagar, posicionar el capçal de la impressora, capturar una dada del port d'entrada, etc
- Tres mecanismes clàssics de comunicació entre perifèric i CPU són
  - E/S programada
  - E/S per interrupció
  - E/S per DMA

## E/S programada

- L'E/S programada realitza la transferència de dades entre la CPU i el perifèric per mitjà d'instruccions d'E/S
- En aquest cas
  - La transferència triga el que dura una instrucció
  - Hi ha d'haver instruccions per a realitzar la transferència
    - Poden ser concretes: IN, OUT
    - Poden ser les genèriques de transferència a memòria: ST, LD
  - Les ordres de recepció i transmissió de dada les ha de generar el perifèric a partir de la informació rebuda de la CPU
    - Adreça
      - Mitjançant un mecanisme de descodificació es selecciona el registre amb el que s'està treballant
      - Moltes CPU's no diferencien entre el mapa de memòria d'E/S i memòria principal
    - Tipus d'operació d'entrada/sortida.
      - Els senyals de control especifiquen el tipus de transferència i estableixen el camí per on han de passar les dades
    - Temporització.
      - La transferència pot ser síncrona o asíncrona
  - Prioritat. Quan molts perifèrics comparteixen la CPU cal establir un mecanisme de prioritat en la comunicació. La gestió pot ser:
    - Distribuïda. Els peticionaris es posen d'acord per determinar qui es queda amb el recurs
      - Exemple: per lògica distribuïda. La pròpia lògica del perifèric que es connecta al bus estableix, entre tots els concursants les prioritats d'accés.
    - Centralitzada. Un mestre dirigeix la cessió del recurs
      - Exemple: daisy-chain. tots els perifèrics tenen una línia comuna de petició de servei i una de concessió amb un mestre i aquest dona la cessió del bus de dades.

## La interrupció

- La interrupció és una bifurcació externa al programa en execució provocada per senyals externs a la CPU
  - Fets externs importants demanen l'atenció de la CPU
  - O la produeix la CPU, per exemple quan s'ha produït una excepció
- La interrupció consta de dues parts
  - El cicle de concessió de la interrupció. La CPU ha de realitzar un salt en l'execució del programa degut a un fet no programat
  - La rutina de servei de la interrupció. Tros de programa que atén l'element que ha produït la interrupció
- La petició de la interrupció involucra els següents mecanismes
  - Identificació de qui ha produït la interrupció
  - Selecció de la rutina de tractament de la interrupció
  - Desactivació de la sol·licitud d'interrupció
  - Superposició d'interrupcions. S'ha de permetre que els dispositius amb mecanisme d'interrupció el puguin tenir activat simultàniament
  - Anidament d'interrupcions. S'ha d'establir una prioritat entre els dispositius amb interrupció.

## Augment de prestacions d'E/S: DMA (I)

- En general, les operacions d'E/S són operacions lentes enfront de la velocitat de procés de la CPU.
- Hi ha mecanismes que donen certa llibertat d'actuació (procés) als perifèrics permetent que aquests treballin en paral·lel amb la CPU, tot descarregant-la de tasques d'espera.
- Per exemple, en la lectura de dades de disquet la CPU pot informar al controlador corresponent per tal que arrenqui el dispositiu mentre la CPU continua fent tasques de procés. En acabar, el dispositiu interromp a la CPU per notificar-li que està preparat.
- Dos mecanismes genèrics de transferència de dades que permeten descarregar a la CPU del control de perifèrics són
  - L'accés directe a memòria o DMA
  - L'establiment de canals d'E/S.
    - En general, un canal és una unitat d'E/S que funciona per DMA.

## Augment de prestacions d'E/S: DMA (II)

- En el DMA el controlador del perifèric es comunica directament amb la memòria principal del computador i en porta el control de la transferència sense intervenció de la CPU
- La CPU informa al controlador de la tasca a realitzar. En acabar, el controlador interromp a la CPU per donar-li la informació sol·licitada.
- L'operació de DMA es fa per blocs de dades. Quan es parla de transferència es sol referir a l'intercanvi d'una paraula
- Per a realitzar les transferències el controlador de perifèric ha de conèixer:
  - L'adreça amb què ha de treballar en memòria principal, per a poder donar les ordres de control correctes
  - El tipus d'operació a realitzar, si d'entrada o sortida
  - El nombre de dades a transferir
  - L'adreça del perifèric
- Donat que l'accés a memòria es pot portar a terme per diferents usuaris, cal establir mecanismes de control de l'accés. Dos mecanismes bàsics són
  - Per control multiporta de memòria. La CPU es connecta a una de les múltiples portes de què disposa la memòria per poder ser accedida. Cada porta és un controlador d'accés a la memòria.
  - Per robatori de cicle. En aquest cas la memòria té una única porta d'accés que és compartida per tots els usuaris, i normalment amb control de la CPU. Aleshores, quan el perifèric necessiti realitzar una transferència a memòria haurà de dialogar amb la CPU per què li cedeixi un cicle o fase per a realitzar la transferència.

## Comunicacions i busos (I)

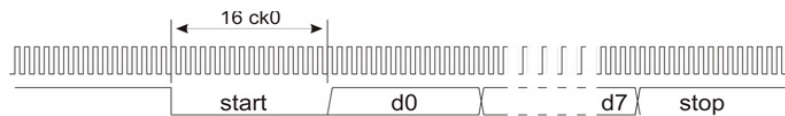
- En l'estructura d'un computador, apart dels elements bàsics de CPU, memòria i perifèrics, cal contemplar-hi els elements de comunicació, constituïts pels enllaços i els commutadors:
  - Els enllaços. Element que permet transferir informació entre diferents dispositius
  - Els commutadors. Element que encamina la informació entre dispositius
- El tipus més comú d'element de comunicació és el bus: és un camí que permet comunicar selectivament diferents dispositius (inclou, per tant, l'enllaç i el commutador)
- En la comunicació amb busos s'ha de considerar el mestre (inicia la comunicació) i l'esclau.
- L'operació bàsica del bus s'anomena cicle de bus. Permet realitzar una transferència elemental. En un cicle de bus es distingeixen les etapes d'adreçament i de transferència.
- Propietats dels busos
  - Segons el tipus de transferència els busos poden ser paral·lels, multiplexats o sèrie
  - Segons la funcionalitat els busos poden ser específics o genèrics
  - Segons el cicle, es poden tenir busos de cicle complet (realitza tots els passos d'adreçament i transferència de cop) o per paquets (cicle partit).
  - Segons la transferència pot ser síncron o asíncron
  - Un bus també pot tenir control d'errors

## Comunicacions i busos (II)

- En un computador s'estableix una gama àmplia de comunicacions que obliga a fer una distinció entre els busos, creant la jerarquia de busos:
  - Busos de tipus 0. Fa referència als busos interns al xip. No són accessibles a l'usuari, però en determinen la connexió del xip al sistema, ja que en determinen l'ample de paraula
  - Busos de tipus 1. És el bus d'interconnexió en una placa de circuit imprès. Les seves característiques venen determinades per les prestacions dels busos de xip i solen ser interns a la pròpia placa, de desenes de cm
  - Busos de tipus 2. Connecten diferents plaques en un mòdul, formant el *backplane*. A nivell de computador, avui en dia passa a ser el bus de sistema, arribant al centenar de senyals de control i de longitud menor a 1 m. Exemples: ISA, VME, PCI.
  - Busos de tipus 3. Connecten diferents mòduls i poden arribar als 10m de longitud, fet que obliga a tractar-lo com a línia de transmissió, emprant interconnexió mitjançant terminals o *buffers*.
  - Busos de tipus 4. És, per exemple, el bus paral·lel de connexió entre perifèrics, en què pot tenir una configuració molt diferent de la bus de sistema (com és el cas del port *Centronics* de comunicació amb impressores).  
Un port paral·lel empra un connector de 25 pins, transfereix paraules de 8 bits i arriba a velocitats de 2Mbit/s
  - Busos de tipus 5. Els busos sèrie constitueixen el tipus elemental de comunicació d'un sistema informàtic (per sota de la comunicació amb paquets). Sovint la informació es transmet en codis ASCII. Abans que les xarxes locals el substituïssin s'emprava per comunicar xarxes locals de baixa velocitat. La velocitat de transmissió és d'1 a 100Mbit/s, poden arribar a longituds de pocs Kms.  
Entre les normes més emprades hi ha la IEEE 802.  
La comunicació en el bus sèrie pot ser simple, semi-duplex (un fil, dos sentits anteraments) o full-duplex (dos fils).  
Entre les normes més comunes de comunicació sèrie entre terminals hi ha la RS-232-C. Permet connectar equips informàtics a mòdems, establint els mecanismes de sincronisme entre terminals, els nivells lògics de treball i el protocol. De fet, s'empra de forma usual per comunicar equips informàtics sense mòdem. La temporització ve definida per la velocitat de transmissió que pot agafar valors entre 110 i 921600 bauds.  
Actualment també s'ha popularitzat molt el bus USB, amb velocitats de 1.5Mb/s a 12Mb/s (480Mb/s – USB2). És un bus sèrie destinat a la connexió de perifèrics lents: ratolí, teclat, escàner, etc.

## El protocol RS232 (I)

- El protocol de comunicació sèrie full-dúplex RS232 asíncron permet establir una comunicació fàcil entre computador i perifèric.
- El estàndar EIA RS232-c especifica els detalls i prestacions que permet la comunicació RS232:
  - Nombre de bits i significat
  - Control de paritat
  - Estat de la línia en repòs i bit d'inici
  - Nombre de bits de stop
  - Velocitats de transmissió (bauds) basada sobre rellotge ràpid 16 cops més ràpid



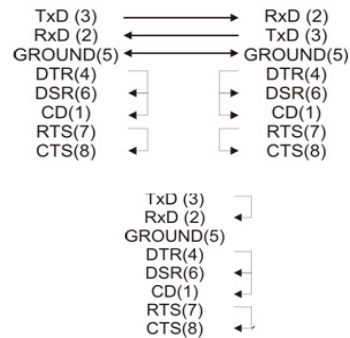
## El protocol RS232 (II)

- **Port sèrie**
- L'estàndar EIA RS232-c especifica un doble port de comunicació sèrie
- Pot emprar un connector de 25 pins o bé un de 9 pins.
- La taula dona les especificacions de pinout per 9 pins

Senyal		Funció	Pin
Transmit Data	TxD	Sortida sèrie	3
Receive Data	RxD	Entrada sèrie	2
Request to Send	RTS	Informa al mòdem que la UART està preparada per a intercanviar dades	7
Clear to Send	CTS	Mòdem preparat per intercanviar dades	8
Data Set Ready	DSR	Indica a la UART que el mòdem està preparat per a establir connexió	6
Signal Ground	GND		5
Carrier Detection	CD	S'activa quan el mòdem rep una portadora de l'altre mòdem	1
Data Terminal Ready	DTR	Informa al mòdem que la UART està preparada per a establir connexió	4
Ring Indicator	RI	S'activa quan el mòdem detecta una senyal indicadora de so	9

## El protocol RS232 (III)

- En sistemes de prototipat amb el PC o en comunicacions sèrie dedicades amb microcontroladors on els errors de comunicació són pràcticament nuls hi ha dues connexions típiques simples que eliminen el protocol de connexió
- Null mòdem
  - Només es necessiten tres connexions en la comunicació: TxD, RxD y GND, i donen a entendre al computador que està connectat amb un mòdem. La transmissió del PC es rep en la UART pel senyal de recepció. I al revés, la transmissió de la UART s'envia al bit de recepció del PC.
  - Per la seva simplicitat, la connexió *Null mòdem* és la que s'emprarà en les pràctiques.
- Connexió en loopback
  - La connexió loopback és útil quan es realitza un programa de depuració en el PC. El senyal de sortida es connecta al d'entrada. D'aquesta forma, tot senyal enviat pel PC pel pin de sortida és rebut en la línia d'entrada com si fos un eco del mòdem.



## Perifèrics en ATmega8

- Els microcontroladors AVR disposen d'un conjunt variable de perifèrics amb els que es pot treballar.
- Entre diferents dispositius cal fer referència a:
  - **Ports d'entrada/sortida.**
    - Un conjunt de ports variable segons el component permeten la comunicació del processador amb l'exterior
    - Cada pin d'entrada/sortida pot tenir una altra funcionalitat (per exemple com a sortida PWM d'un timer, sortida RxD, TxD en comunicació sèrie, etc) segons la programació interna que se'n realitzi. Comptadors/timers
    - El nombre de comptadors/timers depèn del dispositiu (el ATmega8 en té 3), i poden tenir diferents funcionalitats: com a temporitzadors, sortides PWM, comptadors amb comparació, ...
  - **Comptadors/timers**
    - El nombre de comptadors/timers depèn del dispositiu (el ATmega8 en té 3), i poden tenir diferents funcionalitats: com a temporitzadors, sortides PWM, comptadors amb comparació, ...
  - **Mòduls analògics**
    - Disposa de comparadors i convertidors ADC
  - **Comunicacions**
    - Disposen de mòduls de comunicació SPI i USART
  - **Mòduls d'inicialització i de programació**
    - Mòduls de programació i inicialització del microcontrolador

## Entrada / sortida en ATmega8 (I)

- Els microcontroladors AVR realitzen la seva entrada/sortida majoritàriament a través de ports d'E/S.
- La comunicació és per E/S i es pot emprar la interrupció
- Els pins de cada port poden ser adreçats de forma conjunta o per separat.
- Cada pin té una implementació bidireccional amb pull-up's interns que, opcionalment, poden ser activats.
- El control de cada pin es porta a través de tres registres:
  - El registre de control de direcció, Data Direction Register - DDR<sub>xn</sub> (x representa el port i n el pin; p.e., DDRB3)
  - El registre de dades del port, Port Data Register – PORT<sub>xn</sub>
  - I el registre d'entrada del port, Input Pin Address - PIN<sub>xn</sub>.
- Aleshores, la configuració de cada port ve donada per:
  - Les entrades es poden configurar en tri-state o amb pull-up
  - Cada pin d'entrada/sortida té associada una doble funcionalitat: la de pin d'entrada/sortida (configuració per defecte) o com a entrada/sortida d'algun perifèric intern del microcontrolador

## Entrada / sortida en ATmega8 (II)

- Es configura el PORTB com a sortida:
 

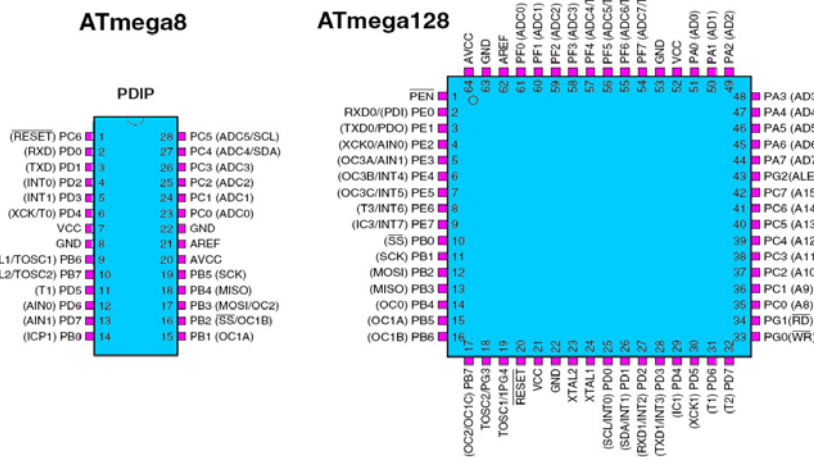
```
ser r16          ; posar el registre r16 a 1
out DDRB, r16   ; posar el PORTB com a sortida
out PORTB, r16  ; posar 1's a la sortida del PORTB
nop             ; sincronisme per poder llegir immediatament
in r16, PORTB
```
- Es configura el PORTB com a entrada amb pull-up's:
 

```
clr r16         ; posar el registre r16 a 0
out DDRB, r16   ; posar el PORTB com a entrada
ser r16         ; registre r16 a 1
out PORTB, r16  ; activar pull-up's a les entrades
nop             ; sincronisme per llegir immediatament
in r16, PORTB
```
- Es configura els pins PORTB7:4 d'entrada i els PORTB3:0 de sortida:
 

```
ldi r16, 0x0F   ; pins 7 a 4 del PORTB com entrades i 3 a 0 com sortides
out DDRB, r16
ldi r16, (1<<PB7)|(1<<PB6) ; pull-up's en les entrades 7 i 6, 0's a les sortides
out PORTB, r16  ; activar els pull-up's a les entrades
nop             ; sincronisme per poder llegir immediatament
in r16, PINB    ; es llegeix el valor del PORTB
```



## Entrada / sortida en ATmega8 (III)



## Comptadors en ATmega8 (I)

- Els microcontroladors AVR disposen de comptadors per a realitzar diferents tasques de temporització

- El ATmega8 té tres comptadors:

- **Timer/Counter0:**

- Comptador simple de 8 bits que pot treballar com a generador de seqüència
- Té un prescaler de 10 bits
- Pot ser controlat a partir d'un senyal extern
- Amb interrupció per overflow

- **Timer/Counter1:**

- Comptador de 16 bits amb dues unitats de comparació que permeten dues sortides PWM independents (per mitjà de dos pins d'entrada/sortida) i amb diferents modalitats
- Té un prescaler de 10 bits
- Pot ser controlat a partir d'un senyal extern
- Té interrupcions en overflow i en comparació

- **Timer/Counter2:**

- Comptador de 8 bits amb unitat de comparació que pot treballar com a generador de seqüència o com a modulador PWM
- Té un prescaler de 10 bits
- Pot tenir com a rellotge un oscil·lador extern a 32KHz

**Nota als comparadors:** si s'empra el senyal d'overflow cal recordar que s'activa en la transició a 0 i que, per desactivar-lo, cal tenir activada la rutina de servei d'interrupció per overflow, o que cal escriure-hi un 1 un cop activat.

## Comptadors en ATmega8 (II)

- Modes de treball dels comptadors
  - Cal tenir sempre a prop el datasheet del microcontrolador per comprendre bé els modes de funcionament dels comptadors. Cal diferenciar 4 modes (segons el timer)
    - Mode normal
      - El comptador compta amunt i es genera un overflow en passar del màxim a 0
      - En aquest mode no es recomana emprar la unitat de comparació
    - Mode CTC: Clear Timer on Compare Match
      - El comptador passa a 0 quan el valor del comptador (TCNTx) s'iguala al valor del comparador (OCRx)
      - En comparació es genera una interrupció (d'estar activada) per comparació
    - Mode Fast PWM
      - El comptador compta amunt i es genera un overflow en passar del màxim a 0.
      - Quan arriba al valor màxim el comptador salta a 0
      - En passar el comptador pel valor del registre de comparació es pot commutar la sortida generant un PWM
    - Mode PWM amb correcció de fase
      - És el mode més complex
      - Un cop el comptador arriba al valor màxim canvia el sentit de la pendent, decrementant el comptador fins arribar a 0, moment en què torna a anar amunt.
      - La sortida pot canviar de valor a cada cop que el timer s'iguali amb el comparador
  - I la sortida?
    - Pot treballar en mode normal (pin desconnectat)
    - Commutant en comparació amb OCx
    - Posant-se a 0 en comparació (i a 1 en TOP, si és el cas)
    - Posant-se a 1 en comparació (i a 0 en TOP, si és el cas)

## Comptadors en ATmega8 (III)

- Registres que actuen sobre els comptadors (per exemple, el timer/counter1):
  - **Registre comptador, TCNT1.**  
Registre de 16 bits que pot ser llegit i escrit i que conté el valor del comptador. Consta de TCNT1H i TCNT1L
  - **Registres comparadors de sortida A i B, OCR1A/OCR1B (16 bits)**  
Contenen valors de comparació amb el valor del comptador. Permeten generar formes d'ona arbitrària o PWM en els pins de sortida OC1A/OC1B. Quan comparador i comptador coincideixen, s'activa el flag OCF1A/OCF1B, que és esborrat per software (escrivint-hi un 1), o en entrar en la ISR  
L'activació d'interrupció per comparació s'habilita amb OCIE1A/OCIE1B (TIMSK)
  - **Registre de captura d'entrada ICR1.**  
Registre de captura del comptador davant un fet extern (entrada ICP1 o bé comparador)
  - **Registres de control d'operació del comptador, TCCR1A/TCCR1B**
    - TCCR1B2:0 - CS12:0. Prescaler: divisió de la freqüència base per 1, 8, 64, 256, 1024, no funcionament i per rellotge extern (pin T1)
    - TCCR1A1:0 i TCCR1B 4-3 - WGM13:0. Selecció de mode de funcionament: normal, Clear on Compare Match, PWM ràpid, de correcció de fase en PWM i de correcció de fase i període en PWM.
    - TCCR1A 7-6/5-4, COM1A1:0/COM1B1:0 - Mode de sortida de comparació del canal A/B (segons mode de funcionament del comptador)
    - TCCR1B:7 - ICNC1 - Bit de cancel·lació de soroll del comptador 1. Bit d'activació de filtre per reduir el soroll en el pin ICP1.
    - TCCR1B6 - ICES1 - Bit de d'especificació de flanc de captura del pin ICP1 del comptador 1.
  - **Registre màscara d'interrupció, TIMSK.**
    - És el registre d'activació d'interrupció dels comptadors. Els bits que afecten al comptador1 són:
      - Bit 2 - TOIE1. Bit d'habilitació d'interrupció per overflow del comptador.
      - Bit 3 - OCIE1A. Bit d'habilitació d'interrupció per comparació amb el registre A.
      - Bit 4 - OCIE1B. Bit d'habilitació d'interrupció per comparació amb el registre B.
      - Bit 5 - TICIE1. Bit d'habilitació d'interrupció per captura externa.
  - **Registre de flag's d'interrupció del comptador, TIFR.**
    - És el registre de status del què passa en el comptador, comú als altres comptadors. Conté els flags referents al comptador 1:
      - Bit 2 - TOV1. És el flag d'overflow del comptador.
      - Bit 3 - OCF1B. És el flag de comparació amb el registre B.
      - Bit 4 - OCF1A. És el flag de comparació amb el registre A.
      - Bit 5 - ICF1. És el flag de captura pel pin ICP1.

## Comptadors en ATmega8 (IV)

**Exemple: Creació d'un retard d'1 ms emprant el timer0 en mode CTC**

```

;FREQ 1Hz AMB COMPTADOR EN MODE CTC (ATMEGA128)
; Joan Oliver (Universitat Autònoma de Barcelona)
; Retard d'1Hz.
; Treballant amb Timer1, mode CTC, sortida per OC1A (PORTB.5)
INCLUDE "m8def.inc"
.DEF temp = R16 ; declaració de registres

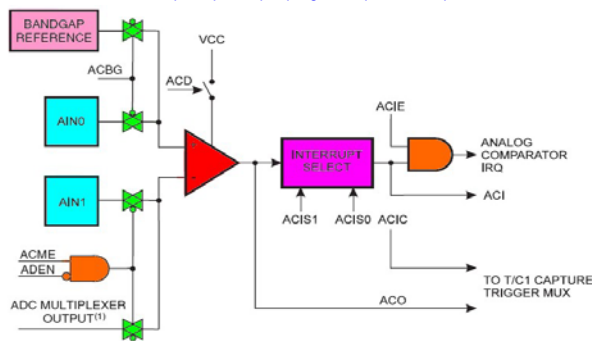
.ORG 0x0 rjmp inici ; salt a inici de programa
; -----INICI PROGRAMA -----
inici:
sbi DDRB,1 ; inicialitzar PORTB.5 -> sortida OC1A
; -----SETUP COMPTADOR-----
ldi temp, (1<<COM1A0) ; sortida en mode toggle on compare
out TCCR1A, temp
ldi temp, (1<<WGM12)|(1<<COM1A0)|(1<<CS11) ; mode comptador CTC i prescaler a 256
out TCCR1B, temp
ldi temp, high(62500) ; registre de comparació a 31250
out OCR1AH, temp
ldi temp, low(62500)
out OCR1AL, temp

; -----PROGRAMA PRINCIPAL-----
main: nop
rjmp main ; ...no cal fer res!
    
```

## Entrada / sortida: mòduls analògics (I)

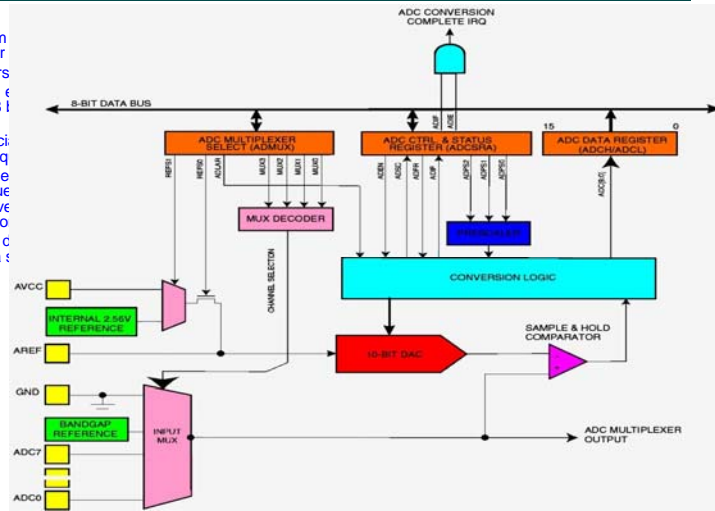
**Comparador:**

- El funcionament del comparador és simple. Compara la diferència de potencial que hi ha entre les entrades AIN0 i AIN1. Si és positiva la sortida ACO és un 1-lògic. En cas contrari es té un 0-lògic.
- Apart d'aquest funcionament bàsic,
- El comparador admet com a entrada per AIN0 una tensió de referència fixa donada pel microcontrolador
- L'entrada AIN1 pot ser una de les entrades multiplexades que entren al convertidor analògic-digital.
- Es pot emprar la sortida de comparació com a entrada de rellotge pel Timer1.
- I disposa d'una unitat de control d'interrupció que es pot programar per flanc o per canvi de nivell.



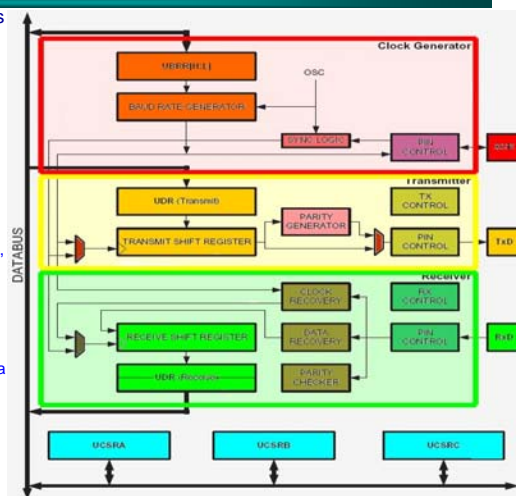
## Entrada / sortida: mòduls analògics (II)

- Conversor analògic-digital:
  - El valor de conversió mínim externa o (es pot connectar)
  - Quan no s'empra el conversor
  - La conversió és de 10 bits, necessita no és major de 8 bits significatius.
  - La conversió del ADC s'inicia amb una interrupció del ADC indica que ha acabat la conversió i continua l'entrada que s'ha convertit.
  - El conversor pot funcionar a una freqüència de 50KHz i 200KHz. Una conversió inicialitza la circuiteria de conversió.
  - El resultat de conversió ve donat a través de l'ADC. Vref la tensió de referència.



## Comunicacions amb USART (I)

- Els microcontroladors AVR tenen dos modes principals de comunicació sèrie:
  - Comunicació SPI
  - Comunicació amb USART
- Arquitectura de la USART
  - El microcontrolador ATmega8 disposa d'un mòdul de transmissió/recepció sèrie síncron/asíncron (USART) amb característiques com:
    - Full dúplex. Té registres de transmissió i recepció independents que el permeten enviar i rebre al mateix temps.
    - Comunicació amb múltiples baudrate's i possibilitat de comunicació a doble velocitat.
    - Suporta diferents formats de protocol: de 5 fins a 9 bits, amb paritat o no i amb 1 o 2 bits de stop.
    - Detecció d'errors.
    - Possibilitat d'emprar la interrupció com a mitjà de millorar la transmissió/recepció.
    - El mòdul de transmissió disposa del seu registre de desplaçament, lògica de control i de generació de paritat i d'un buffer d'escriptura que li permet efectuar la transmissió contínua de paraules sense retards enmig.
    - El mòdul de recepció conté la lògica de sincronisme en recepció asíncrona, un registre de desplaçament, lògica de control i de paritat i un doble buffer de recepció que li permet operar com a FIFO i donant-li una major seguretat en la recepció de dades.



## Comunicacions amb USART (II)

- La USART és un mòdul de certa complexitat en el microcontrolador i necessita una lectura acurada del full d'especificacions.
  - **Registre d'E/S de la USART, UDR.**  
Els buffers de transmissió i de recepció comparteixen la mateixa adreça d'E/S, UDR. El buffer de transmissió és el buffer on es posa la dada a enviar i el de recepció és el d'arribada de dades en la recepció.
  - **Registre A de status i control, UCSRA.** Els bits més importants són:
    - Bit 1 - U2X. Bit de duplicació de velocitat de transmissió
    - Bits 2:4 - PE-DOR:FE. Bits d'errors: paritat, *overflow*, i error de format
    - Bit 5 - UDRE. USART Data Register Empty. Quan és 1 indica que el buffer de transmissió és buit i pot rebre una nova dada. Pot generar la interrupció UDRIE.
    - Bit 6 - TXC. USART Transmit Complete. Es posa a 1 quan s'ha enviat tot el frame. S'esborra de forma automàtica en entrar en la rutina de servei TXCIE.
    - Bit 7 - RXC. USART Receive Complete. Es posa a 1 quan hi ha una dada no entrada en el buffer i es posa a 0 en llegir-la. Pot generar una interrupció RXCIE.
  - **Registre B de status i control, UCSRB.**
    - Bit 0 - TXB8. És el 9è bit de transmissió quan es treballa amb 9 bits
    - Bit 1 - RXB8. És el 9è bit de recepció quan es treballa amb 9 bits
    - Bit 2 - UCSZ2. Va amb els bits UCSZ1:0 del registre UCSRC.
    - Bit 3 - TXEN. Habilita la transmissió en la USART
    - Bit 4 - RXEN. Habilita la recepció en la USART
    - Bit 5 - UDRIE. S'habilita la interrupció sobre el flag UDRE
    - Bit 6 - TXCIE. Habilita la interrupció sobre el flag de transmissió TXC
    - Bit 7 - RXCIE. Habilita la interrupció sobre el flag de recepció RXC
  - **Registre C de status i control, UCSRC.**
    - Bit 2:1 - UCSZ1:0. Juntament amb UCSZ2 (UCSRB), UCSZ2:0 estableix el nombre de bits en la comunicació. Quan val 3 s'envien 8 bits.
    - Bit 3 - USBS. Indica el nombre de bits de stop. Quan és zero indica 1 bit
    - Bit 5:4 - UPM1:0. Estableix la paritat. A 0 quan no s'empra paritat.
    - Bit 7 - URSEL. Veure URSEL en registre UBRRH. Quan és 1 selecciona el registre UCSRC
  - **Registres de baud rate, UBRRH (UBRRH,UBRRL)**
    - Bit 11:0 - UBRRL1:0. Són els bits d'establiment de la velocitat de comunicació.
    - Bit 15 - URSEL. És un bit complicat! com que UBRRH comparteix adreça d'E/S amb UCSRC. Ha de ser 0 quan s'escriu a UBRRH.
- La velocitat de transmissió depèn del baud rate seleccionat, de si s'ha activat el bit de comunicació de doble velocitat i de la freqüència de rellotge base del microcontrolador. Per això, per establir la velocitat de comunicació cal consultar les taules del manual d'especificacions.

## Comunicacions amb USART: exemple (I)

- Inicialització
  - Abans de realitzar cap comunicació la USART s'ha d'inicialitzar. La inicialització inclou l'establiment del baud rate, posar el format de transmissió (nombre de bits, paritat, bits de stop) i l'habilitació de la transmissió i recepció i si es fa per interrupcions.
  - Un codi exemple base de comunicació pot ser:

```

;-----RUTINA DE SETUP DE LA UART-----
; a 4800 bauds i frame format: 8data,1stop bit
setup_UART:
    ldi temp0,0 ;baud rate a 4800
    out UBRRH,temp0
    ldi temp0,12
    out UBRRL,temp0
    ldi temp0,(1<<RXEN)|(1<<TXEN)                ;enable recepció i transmissió
    out UCSRB,temp0
    ldi temp0,(1<<URSEL)|(1<<UCSZ1)|(1<<UCSZ0)    ;frame format: 8 bits, 1 stop, no paritat
    out UCSRC,temp0
    ret
    
```

## Comunicacions: RS-232 (III)

- Funcionament de la transmissió
  - La transmissió s'habilita a través del bit TXEN del registre UCSRB. La seva habilitació sobreseu el funcionament del bit PORTD.1 que passa a ser el pin de sortida de la transmissió, TxD.
  - La transmissió s'inicia carregant el buffer de transmissió (escriptura al registre d'entrada/sortida UDR) amb la dada a transmetre. La dada s'enviarà al registre de desplaçament per a ser enviada un cop aquest estigui lliure (immediatament si no té cap dada, o un cop hagi transmès la dada que té).
  - Per tant, si no es vol perdre cap dada en la transmissió, no es pot enviar lliurement una dada al buffer de transmissió, si no que s'ha d'esperar a què estigui lliure. Això es pot conèixer per polling o per interrupció.
  - L'exemple mostra una rutina de polling sobre el buffer UDR preguntant pel bit UDRE, *USART Data Register Empty*.
 

```

;-----RUTINA DE TRANSMISSIÓ-----
;Molt senzillament, aquesta rutina envia un byte cada cop que es crida
;No es pot enviar fins que el registre de transmissió queda buit
transmit: sbis UCSRA, UDRE
          rjmp transmit
          out udr, dada
          ret
          
```
  - Evidentment l'ús de la rutina de polling pot significar importants retards de procés quan hi ha moltes dades per enviar. En aquests casos, quan no n'hi ha prou incrementant al màxim permès la velocitat de transmissió, és recomanable emprar una cua i interrupcions per conèixer si una nova dada pot ser carregada al buffer de transmissió.

## Comunicacions: RS-232 (III)

- Funcionament de la recepció
  - El funcionament de la recepció és similar al de transmissió. La recepció s'habilita a través del bit RXEN del registre UCSRB. La seva habilitació sobreseu el funcionament del bit PORTD.0 que passa a ser el pin d'entrada de la recepció, RxD.
  - La recepció s'inicia quan detecta un bit de recepció vàlid. Cada nou bit que es reb es va carregant en el registre de desplaçament de recepció fins que es reb el primer bit de stop, moment en què passa la dada al buffer de recepció. Llegint el registre d'E/S UDR es carrega la dada en el registre pertinent.
  - L'exemple mostra una rutina de polling sobre el bit RXC per determinar si ha arribat una nova dada. El flag RXC es manté activat mentre no es llegeix la dada del registre UDR.

```

;-----RUTINA DE RECEPCIÓ-----
;Molt senzillament, aquesta rutina espera l'arribada d'una dada
;Aleshores la carrega en el registre temporal
recepcio: sbis UCSRA, RXC
          rjmp recepcio
          in temp, udr
          ret
          
```

## Bibliografia

---

- Pedro de Miguel Anasagasti. Fundamentos de los Computadores. Edit Thomson. 2004.
- ATmega8 Datasheet i ATmega128 Datasheet
- [http://www.cannic.uab.es/Docencia/FCwebAVR/FC\\_AVR.htm](http://www.cannic.uab.es/Docencia/FCwebAVR/FC_AVR.htm)