

Ingénierie de μ -contrôleurs, 4A MT

2^{ème} séance : TP2 « affichage de vitesse »

Intervenant : Nikolay Smagin

Responsable pédagogique : Christophe Delebarre



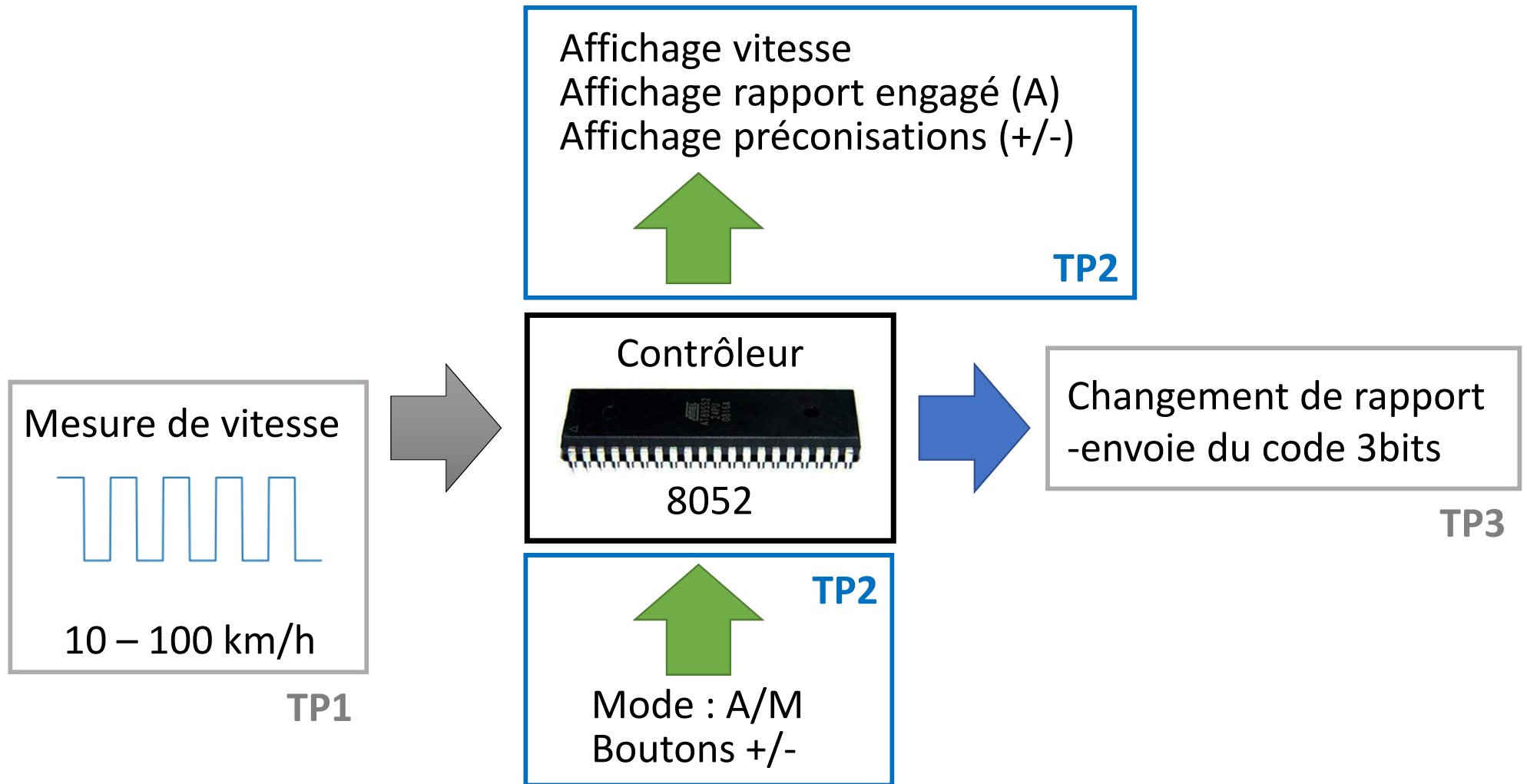
Déroulement général de ces TPs

TP1. Récupération de l'information sur vitesse dans une plage 10 – 100 km/h (utilisation des temporisateurs) ; un critère de 5% sera admis sur la précision.

TP2. Affichage de la vitesse au port série du mode et du rapport enclenché. Contrairement à l'énoncé joint ci-dessous, l'afficheur LCD ne sera pas utilisée car il n'est pas accessible par défaut sous Keil μ Vision.

TP3. Gestion du passage automatique des rapports en fonction de la vitesse (mode auto).

Contrôleur de la boîte de vitesses



Il existe des afficheurs port série

[Home](#) » [Products](#) » [Serial LCD Displays](#) » [Character](#) » [Interface](#) » [RS232](#)

RS232 Character Serial LCD Displays



RS232 16x2 Character Module

Part #: CFA632YDIKS

Size: 108mm x 42mm x 19.9mm

Interfaces: RS232

Character Count by Lines: 16x2

Color: Yellow on Dark

Polarizer: Transmissive

Backlight: Yellow-Green LED

Type: Character Serial LCD

Price Range: \$42.11 - \$57.53

[Add to Compare](#)



16x2 RS232 Character LCD

Part #: CFA633TFHKS

Size: 110.5mm x 35mm x 25.6mm

Interfaces: RS232

Character Count by Lines: 16x2

Color: Dark on Light Gray

Polarizer: Transflective

Backlight: White LED

Type: Character Serial LCD

Price Range: \$35.62 - \$61.79

[Add to Compare](#)



20x4 RS232 Character LCD

Part #: CFA635TMLKS

Size: 142mm x 37mm x 20.8mm

Interfaces: RS232

Character Count by Lines: 20x4

Color: White on Blue

Polarizer: Transmissive

Backlight: White LED

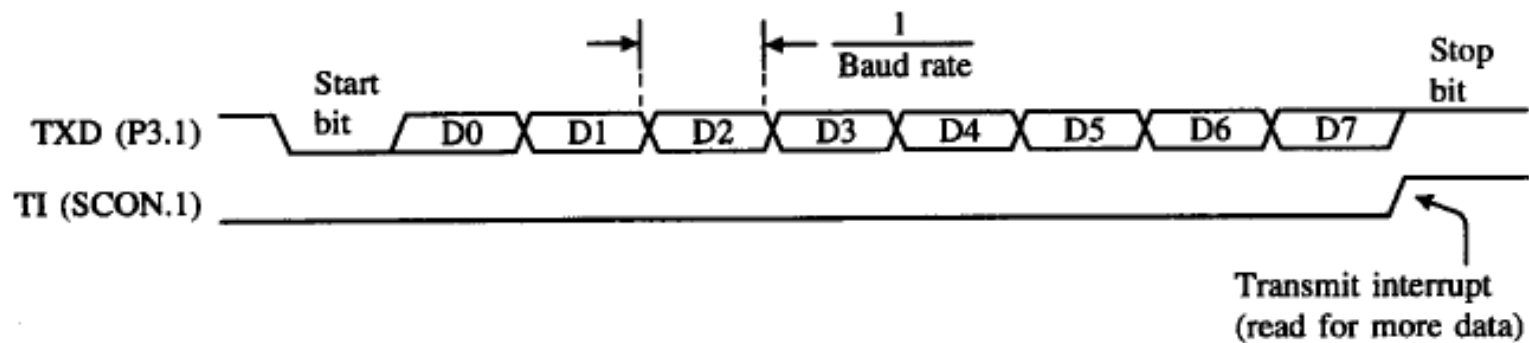
Type: Character Serial LCD

Price Range: \$61.92 - \$97.97

[Add to Compare](#)

Qu'est-ce qu'un port série ?

1. Série : envoi/réception **bit par bit**
2. **2 fils** pour envoie réception, on peut avoir des fils complémentaires pour le synchro, contrôle, etc.
3. Fil d'horloge ou Baude rate identique pour l'émetteur et le récepteur
4. Le **baud** (symbole **Bd**) est l'unité de rapidité de modulation en télégraphie, l'inverse de la durée du plus court élément du signal.



Port série : à quoi sert-il ?

I²C (Inter-Integrated Circuit, en anglais), **TTL**



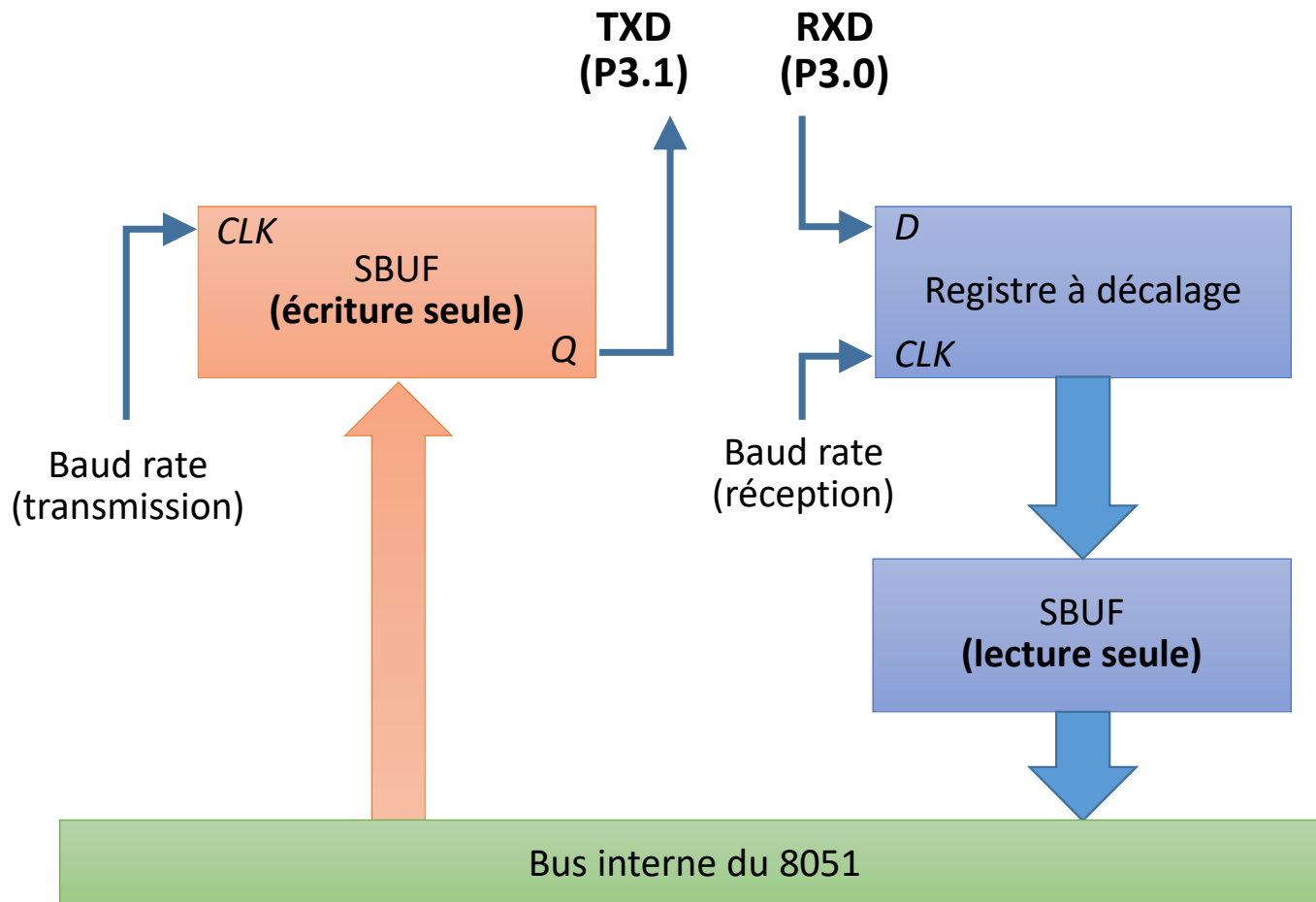
RS-232 (le « **port série** »), **-15V ; +15V**

UART (Universal Asynchronous Receiver Transmitter), **TTL**

1-Wire (Dallas Semiconductor)

CAN (Controller Area Network)

C'est confus : mais il y a 2 registres SBUF



T2	P1.0	1	40	VCC
T2EX	P1.1	2	39	P8.8 AD8
	P1.2	3	38	P8.1 AD1
	P1.3	4	37	P8.2 AD2
	P1.4	5	36	P8.3 AD3
	P1.5	6	35	P8.4 AD4
	P1.6	7	34	P8.5 AD5
	P1.7	8	33	P8.6 AD6
	RST	9	32	P8.7 AD7
RXD	P3.0	10	31	EA/Vpp*
TXD	P3.1	11	30	ALE/PROG
INT0	P3.2	12	29	PSEN
INT1	P3.3	13	28	P2.7 A15
T0	P3.4	14	27	P2.6 A14
T1	P3.5	15	26	P2.5 A13
WR	P3.6	16	25	P2.4 A12
RD	P3.7	17	24	P2.3 A11
XTAL2		18	23	P2.2 A10
XTAL1		19	22	P2.1 A 9
VSS		20	21	P2.0 A 8

DIP 40

Communication port série par le 8051

Registre	Nom complet	Description
SBUF	Serial port buffer	8 bits à écrire/lire
SCON	Serial port control	Mode de fonctionnement, flags d'interruption, 9 ^{ème} bits
IE	Interrupt enable	Bit ES (Enable serial)
IP	Interrupt priority	Priorité Temporisateurs/Port série
PCON	Power control	Bit SMOD = 0/1 pour régler le multiplicateur ($\div 32/16$) de Baud Rate
TIMERS 0,1 et 2 (8052)		Baud Rate: Débordement $\div 32$ (SMOD = 0) ou $\div 16$ (SMOD = 1)

Byte address	Bit address	
FF		
F0	F7 F6 F5 F4 F3 F2 F1 F0	B
E0	E7 E6 E5 E4 E3 E2 E1 E0	ACC
D0	D7 D6 D5 D4 D3 D2 - D0	PSW
B8	- - - BC BB BA B9 B8	IP
B0	B7 B6 B5 B4 B3 B2 B1 B0	P3
A8	AF - - AC AB AA A9 A8	IE
A0	A7 A6 A5 A4 A3 A2 A1 A0	P2
99	not bit addressable	SBUF
98	9F 9E 9D 9C 9B 9A 99 98	SCON
90	97 96 95 94 93 92 91 90	P1
8D	not bit addressable	TH1
8C	not bit addressable	TH0
8B	not bit addressable	TL1
8A	not bit addressable	TL0
89	not bit addressable	TMOD
88	8F 8E 8D 8C 8B 8A 89 88	TCON
87	not bit addressable	PCON
83	not bit addressable	DPH
82	not bit addressable	DPL
81	not bit addressable	SP
80	87 86 85 84 83 82 81 80	P0

SPECIAL FUNCTION REGISTERS

Registre SCON

MOV SCON, #01000010B ; |SM0|SM1|SM2|REN|TB8|RB8|TI |RI |

Bit	Symbole	Adresse	Description
SCON.7	SM0	9FH	(Serial mode 0). Bit 0 de choix du mode de port série
SCON.6	SM1	9EH	(Serial mode 1). Bit 1 de choix du mode de port série
SCON.5	SM2	9DH	(Serial mode 2). Bit 2 de choix du mode de port série. Utilisé pour des communications entre plusieurs unités en modes 2 et 3
SCON.4	REN	9CH	(Receiver Enable). Ce bit doit être en état 1 afin de recevoir des caractères
SCON.3	TB8	9BH	(Transmit bit 8). 9 ^{ème} bit émis en modes 2 et 3. Il faut le gérer par le code.
SCON.2	RB8	9AH	(Receive bit 8). 9 ^{ème} bit reçu.
SCON.1	TI	99H	(Transmit interrupt flag). Ce flag est mis à la fin de transmission de la séquence d'envoi. Il faut le gérer par le code.
SCON.0	RI	98H	(Receive interrupt flag). Ce flag est mis à la fin de réception d'un caractère. Il faut le gérer par le code.

Les modes de port serie

SM0	SM1	Mode	Description	Baud Rate
0	0	0	Registre à décalage	Fixé par la fréquence de l'oscillateur ($\div 12$)
0	1	1	UART 8 bit	Variable, défini par le temporisateur TIMER1 [ou TIMER2 (pour RXD, si les deux sont utilisés)]
1	0	2	UART 9 bit	Fixé par la fréquence de l'oscillateur ($\div 32$ ou $\div 64$)
1	1	3	UART 9 bit	Variable, défini par le temporisateur TIMER1 [ou TIMER2 (pour RXD, si les deux sont utilisés)]

Baud rate fixé par la fréquence de l'oscillateur \Rightarrow communication entre des composants (capteur, numériseur, etc.), souvent se trouvant sur la même carte

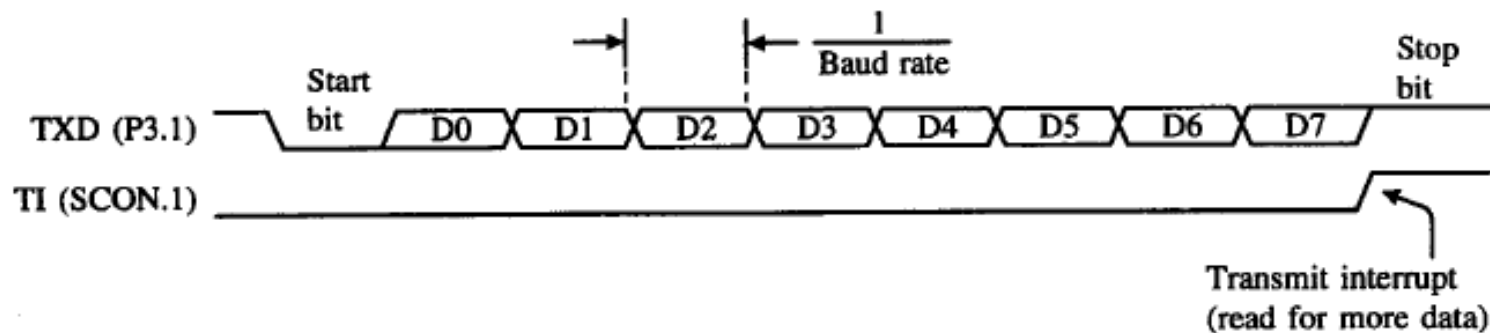
Baud rate déterminé par temporisateurs \Rightarrow communication entre des dispositifs (régulateur, afficheur, etc.), souvent éloignés les uns des autres

Bauds réglementés

Afin de faciliter l'interopérabilité entre périphériques (PC, microcontrôleur, modem...) des vitesses de transmission sont normalisées par multiples et sous-multiples de 9600 baud

Pour 8 UART 1 caractère est codé sur 10 bits, alors la vitesse de transmission (en Hertz) d'un symbole est égale à **[Baud rate]/10**

- 110 baud
- 220 baud
- 300 baud
- 1 200 baud
- 2 400 baud
- 4 800 baud
- **9 600 baud**
- 19 200 baud
- 38 400 baud



Réglage de taux de Baud

Baud Rate	Crystal Frequency	SMOD	TH1 Reload Value	Actual Baud Rate	Error
9600	12.000 MHz	1	-7 (0F9H)	8923	7%
2400	12.000 MHz	0	-13 (0F3H)	2404	0.16%
1200	12.000 MHz	0	-26 (0E6H)	1202	0.16%
19200	11.059 MHz	1	-3 (0FDH)	19200	0
9600	11.059 MHz	0	-3 (0FDH)	9600	0
2400	11.059 MHz	0	-12 (0F4H)	2400	0
1200	11.059 MHz	0	-24 (0E8H)	1200	0

Utilisation de TIMER1 :

Dans le cas de TIMER1, la vitesse de communication est définie par :

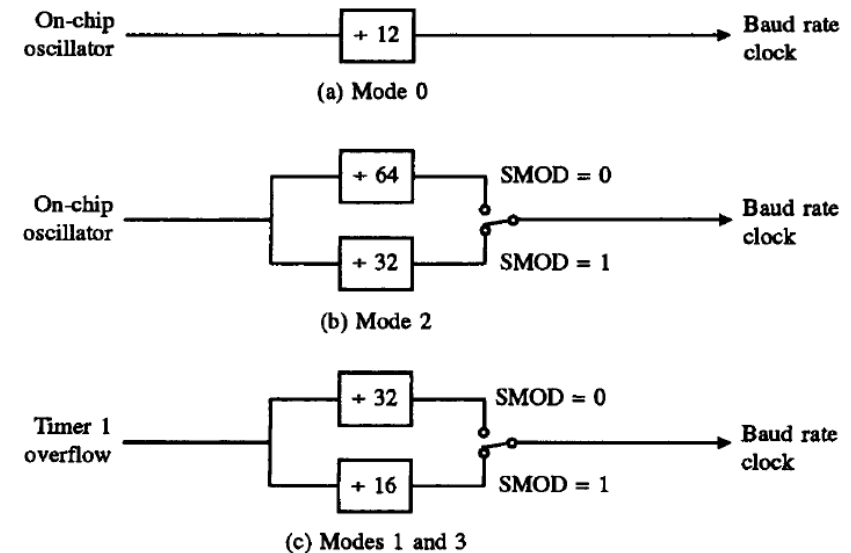
$$VC = \frac{2^{SMOD}}{32} \times \text{fréquence de débordement}$$

D'une manière générale, TIMER1 est utilisé en mode Auto-Reload, dans ce cas, la vitesse de communication est définie par :

$$VC = \frac{2^{SMOD}}{32} \times \frac{f_{osc}}{12 \times (256 - TH1)}$$

Vitesse de communication	Fréquence de l'oscillateur	SMOD	TH1
19200	11.059 MHz	1	FDh
9600	11.059 MHz	0	FDh
4800	11.059 MHz	0	FAh
2400	11.059 MHz	0	F4h
1200	11.059 MHz	0	E8h

Le tableau ci-contre donne la valeur de TH1 pour les vitesses de communication couramment utilisées et ceci pour un quartz de 10.059 MHz



Beaucoup d'exemples sont disponibles dans le livre

Assembly language:

```
OUTCHR:  MOV C,P           ;PUT PARITY BIT IN C FLAG
          CPL C           ;CHANGE TO ODD PARITY
          MOV ACC.7,C     ;ADD TO CHARACTER
AGAIN:   JNB TI,AGAIN     ;TX EMPTY?
          CLR TI         ;YES: CLEAR FLAG AND
          MOV SBUF,A      ; SEND CHARACTER
          CLR ACC.7      ;STRIP OFF PARITY BIT AND
          RET            ; RETURN
OUTSTR:  MOV A,@DPTR     ;GET CHARACTER
          JZ EXIT        ;IF 0, DONE
          CALL OUTCHR    ;OTHERWISE SEND IT
          INC DPTR       ;INCREMENT POINTER
          SJMP OUTSTR    ; AND GET NEXT CHARACTER
EXIT:    RET
```

Également sur la page 292

Caractères de contrôle pour un afficheur

Sequence	Description
From Target → μVision	
CR ('\r')	Carriage Return
LF ('\n')	Line Feed
TAB ('\t')	Tabulator
BS (0x08)	Back Space
BEL (0x07)	Bell
<ESC>[C	Move cursor (position) 1 to the right
<ESC>[D	Move cursor (position) 1 to the left
<ESC>[K	Delete line right to cursor
<ESC>[J	Clear screen
<ESC>[y;xf	Set cursor to Row y, Column x; (x,y) are numeric ASCII values
From μVision ⇒ Target	
<ESC>[A	Cursor Key UP
<ESC>[B	Cursor Key DOWN
<ESC>[C	Cursor Key RIGHT
<ESC>[D	Cursor Key LEFT

Exemple n°1 : Ecriture port série de nombre d'appui

Logic Analyzer

Setup... Load... Save... Min Time 57.93308 s Max Time 79.70699 s Grid 1 s Zoom In Out All Min/Max Auto Undo Update Screen Stop Clear Transition Prev Next Jump to Code Trace Signal Info Amplitude Show Cycles Cursor

T1
IE0
P1
P1

59.02178 s 59.82178 s 75.02178

Disassemble

Serial

44 LOOP
45
46
47 SJMP LOOP
48
49
50

Parallel Port 3

Port 3
P3: 0xED 7 Bits 0
Pins: 0xED

Interrupt System

Int Source	Vector	Mode	Req	Ena	Pri
P3.2/Int0	0003H	1	0	1	0
Timer 0	000BH	0	1	0	
P3.3/Int1	0013H	1	0	1	0
Timer 1	001BH	1	0	0	
Serial Rcv.	0023H	0	0	0	
Serial Xmit.	0023H	1	0	0	
Timer 2	002BH	0	0	0	
P1.1/T2EX	002BH	0	0	0	
PCA Timer	0033H	0	0	0	

Command

Load "E:\UPHF\07_Forma1\04_Enseignements\2020
WS 2, `COMPT0
WS 2, `COMPT1
WS 2, `R0
WS 2, `R1

UART #1

```
====COMPTEURS 8051====  
N° d'appui INTO : 9  
N° d'appui INT1 : 5  
=====
```

Examinez le fichier [...] \[04. Serial_2press_buttons]\Serial_2press_buttons.uvproj

Exemple : Ecriture port série de nombre d'appui

Problème avec les attentes des flags TI \Rightarrow les interruptions du temporisateur1 s'arrête
 \Rightarrow **Il faut régler les priorités des interruptions !**

La scrutation d'une seule broche de P3 se fait par le carry flag C (A pour 8 bits, C pour 1 bit)

```
59 LOOP_MAIN:
60     MOV     C, P3.4 ; Operation 1 bit par le carry flag (drapeau de report)
61     MOV     PT0, C
62     SJMP   LOOP_MAIN
```

Autre approche possible : interruptions port série

- SPIR se produise lorsque **TI** ou **RI** est activé.
- Une interruption d'émission se produit lorsque la transmission du caractère précédent écrit dans SBUF est terminée. Une interruption de réception se produit lorsqu'un caractère a été complètement reçu et attend d'être lu dans SBUF.
- L'indicateur qui provoque l'interruption de port série n'est pas effacé par le matériel lorsque l'unité centrale de traitement (UC) se positionne sur l'interruption.
- La raison en est qu'il existe deux sources pour une interruption de port série, TI ou RI. La source de l'interruption doit être déterminée dans l'ISR et le drapeau d'interruption effacé par le code.

Exemple n°2 : Ecriture port série de nombre d'appui (SPISR)

The screenshot displays the Logic Analyzer interface with the following components:

- Logic Analyzer Window:** Shows waveforms for signals T1, IE0, P1, and P1. The time range is from 59.02178 s to 75.02178 s. A cursor is positioned at 59.82178 s.
- Serial Window:** Shows assembly code for a serial port. The code includes a loop that writes to port 3 (0xED) and jumps back to the start of the loop.
- Interrupt System Window:** Shows a table of interrupt sources and their configurations.
- Command Window:** Shows the command to load the project file.
- UART #1 Window:** Shows the output of the serial port, displaying the number of button presses for INT0 and INT1.

Int Source	Vector	Mode	Req	Ena	Pri
P3.2/Int0	0003H	1	0	1	0
Timer 0	000BH	0	1	0	0
P3.3/Int1	0013H	1	0	1	0
Timer 1	001BH	1	0	0	0
Serial Rcv.	0023H	0	0	0	0
Serial Xmit.	0023H	1	0	0	0
Timer 2	002BH	0	0	0	0
P1.1/T2EX	002BH	0	0	0	0
PCA Timer	0023H	0	0	0	0

```
Load "E:\\UPHF\\07_Forma1\\04_Enseignements\\2020\\WS 2, `COMPT0
WS 2, `COMPT1
WS 2, `R0
WS 2, `R1
```

```
====COMPTEURS 8051====
N° d'appui INT0 : 9
N° d'appui INT1 : 5
```

Examinez le fichier [..]\\[Serial_2press_buttons_interrupts]\\05.Serial_2press_buttons_interrupts.uvproj

Taches à accomplir pour le TP2

1. Employer TIMER2 pour écriture port série à un taux de Baud standard (les registres de TIMER2 sont déjà définis dans les projets d'exemples, mais ils ne sont pas utilisés).
 1. Monter dans le CR que le Baud rate est respecté.
2. Choisir le mode d'écriture au port série : (1) par fonctions ou (2) par interruptions.
 1. Démontrer dans le CR que le timing est respecté si le mode (1) a été choisi.
3. Dessiner la partie statique de « tableau de bord ». Le format est arbitraire, mais les éléments suivants doivent y figurer : VITESSE (km/h); RAPPORT; MODE; CONSIGNE.

```
|====TABLEAU DE BORD :  
| VITESSE = XXX km/h  
| RAPPORT = X  
| MODE = A/M  
| CONSIGNE : X
```

Taches à accomplir pour le TP2

4. Ecrire la valeur de la vitesse dans le tableau de bord.
5. Définir les broches pour le mode et '+' et '-'.
6. Employer la scrutation des broches dans la boucle principale
7. Employer le changement d'affichage de mode : 'A/M'.

TP2 : affichage de vitesse par le port série

Bon courage !

Vous pouvez juste incrémenter vos comptes rendus en ajoutant des descriptions et en intégrant le code du TP1

Au supplément du compte-rendu pensez à m'envoyer les projets entiers Keil μ Vision : le dossier entier compressé incluant les fichiers *.uvproj et *.a

Annexe : organisation possible du programme

Initialisation

;=====

BOUCLE_PRINCIPALE:

LANCEMENT TIMER pour comptage de vitesse

CONVERSION temps -> vitesse

... CHANGEMENT RAPPORT

ECRITURE port série

SCRUTATION DE BROCHES

JMP BOUCLE_PRINCIPALE

;=====

FONCTIONS

ISRs

Annexe : réglage de SMOD

```
MOV  A,PCON          ;GET CURRENT VALUE OF PCON
SETB ACC.7           ;SET BIT 7 (SMOD)
MOV  PCON,A          ;WRITE VALUE BACK TO PCON
```

Comme PCON n'est pas adressable par bit, le réglage de SMOD sans modifier les autres bits de PCON nécessite une opération de "lecture-modification-écriture". Les instructions suivantes configurent SMOD :