

# Ingénierie de $\mu$ -contrôleurs, 4A MT

## 2<sup>ème</sup> séance : TP2 « affichage de vitesse »

Intervenant : Nikolay Smagin

Responsable pédagogique : Christophe Delebarre



2024

## Déroulement général de ces TPs

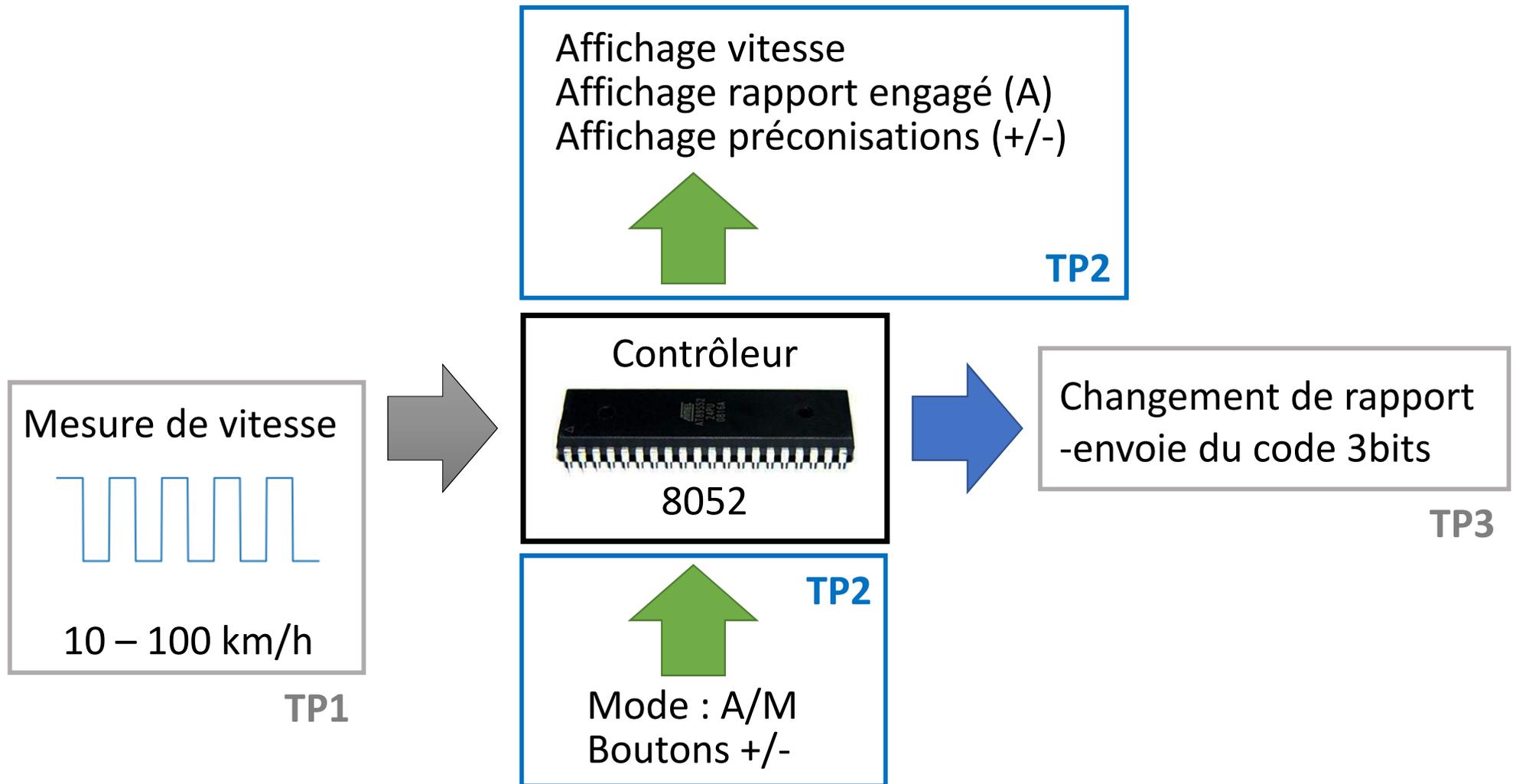
---

**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  $\mu$ 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

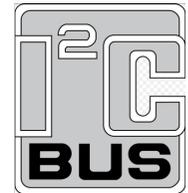
Price Range: \$61.92 - \$97.97

[Add to Compare](#)

## Port série : à quoi sert-il ?

---

**I<sup>2</sup>C** (Inter-Integrated Circuit, en anglais), **TTL, 0-5 V**



**RS-232** (le « **port série** »), **-15V ; +15V : utilisation d'un driver!**

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

**1-Wire** (Dallas Semiconductor) : **0-5 V**

**CAN** (Controller Area Network) : **0-5 V**

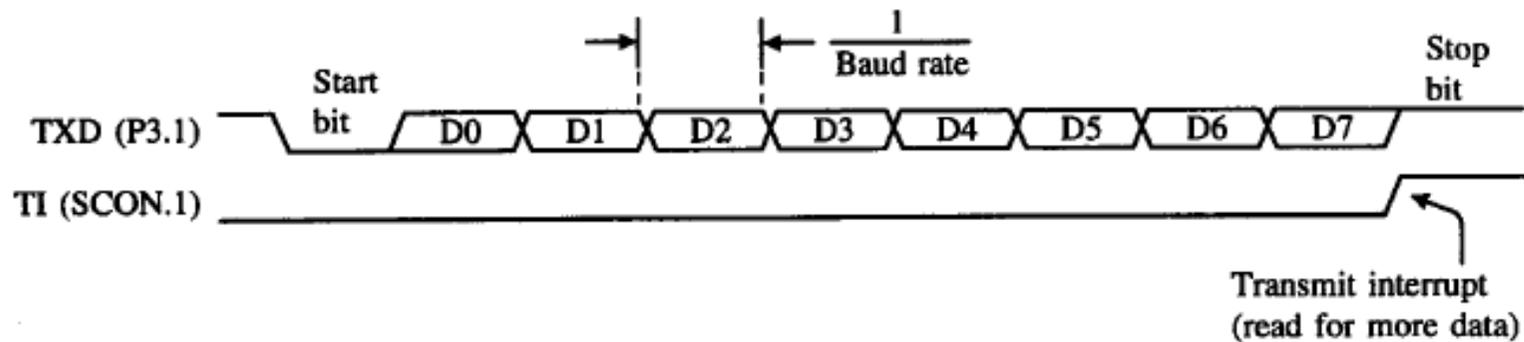
---

Baud rate

---

# 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.

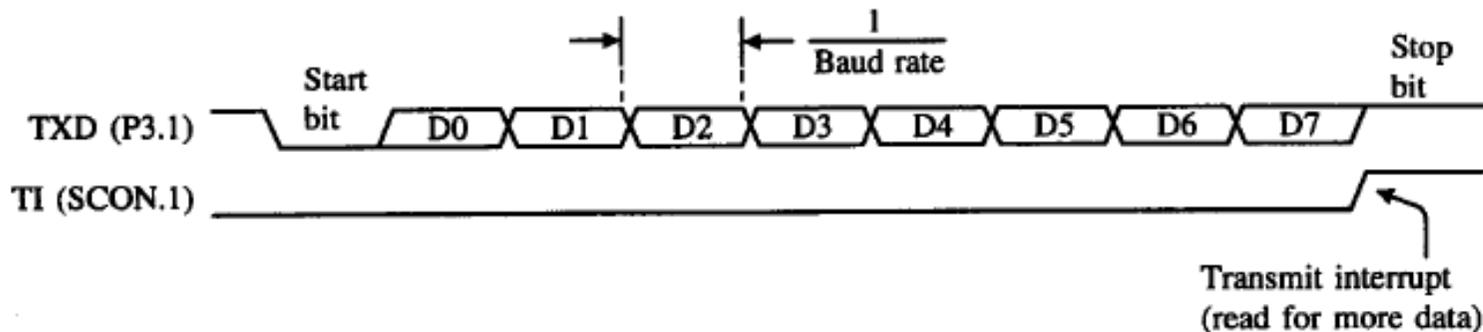


# Bauds réglementés

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

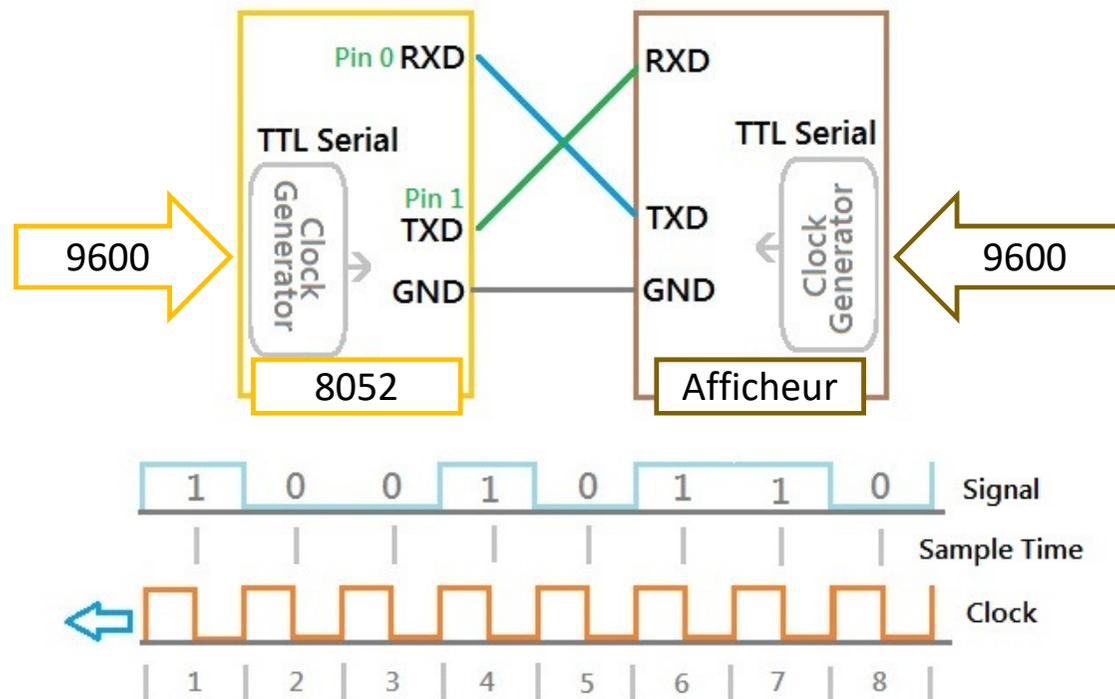
Pour 8 UART 1 caractère est codé sur 10 bits, alors la vitesse de transmission (en Hertz) d'un symbole est égale à **[Baud rate]/10**

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



# Bauds réglementés

Emetteur et récepteur doivent être réglés au même baud rate pour une communication correcte



---

# Communication port série avec TIMER1

---

# Vitesse de communication avec TIMER2

---

## **Utilisation de TIMER2 :**

La configuration de TIMER2 en générateur de rythme est réalisée à l'aide des bits RCLK et TCLK du registre T2CON. La vitesse de communication est donnée par :

$$VC = \frac{\text{fréquence de débordement}}{16}$$

D'une manière générale, TIMER2 est utilisé en mode Auto-Reload, dans ce cas, la vitesse de transmission est définie par :

$$VC = \frac{f_{osc}}{32 \times (65536 - RCAP2)}$$

RCAP2 (16 bit) -> T2HIGH (8 bit) & T2LOW (8 bit) ; VC = 9600

**Pour info :** les erreurs de de la vitesse de communication en utilisant TIMER1 (**nous utiliserons TIMER2**)

**TABLE 5-3**

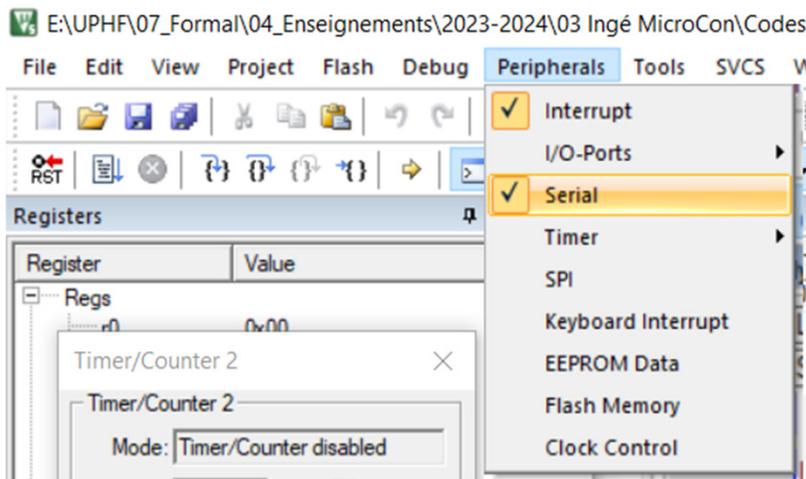
Baud rate summary

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

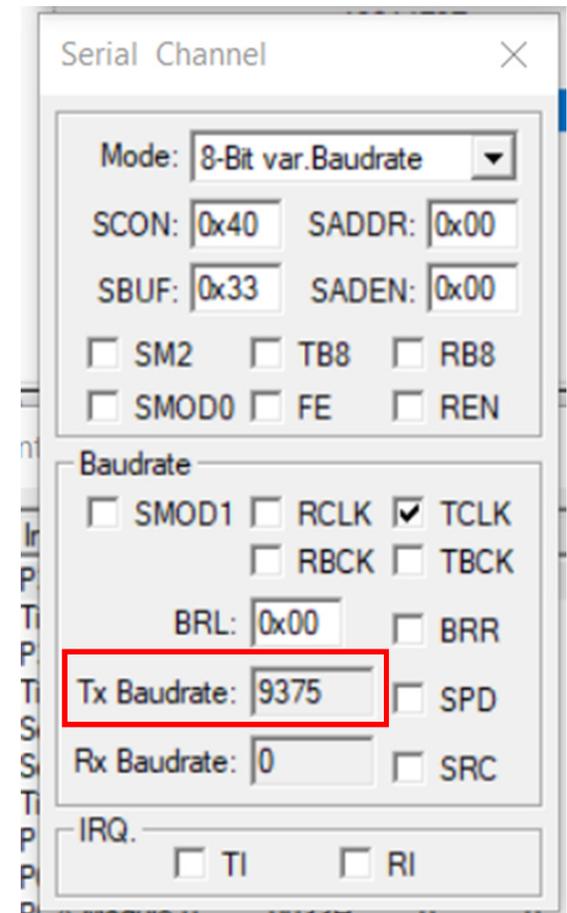
Nous utilisons un oscillateur quartz à 12 MHz, donc une erreur de baud rate est inévitable

# Comment s'assurer de baud rate sur Keil uVision

## Menu -> Peripherals -> Serial



A savoir : avec le quartz de 12 MHz, il y aura toujours une erreur de définition de baud rate



---

# Taches à accomplir pour le TP2

---

# Taches à accomplir pour le TP2

---

Télécharger le projet pour le TP2

Tout d'abord, recopiez le code du TP1 (mesures de vitesses dans la VITHEX et éventuellement, les fréquences additionnelles correspondant aux autres vitesses ajoutées (optionnel du TP1)) dans le nouveau projet de TP2.

Assurez-vous que tout fonctionne comme avant (affichage correct de la variable VITHEX)

Utiliser le point d'arrêt placé dans le EXT1ISR

# Taches à accomplir pour le TP2

---

## 1

1. Employer **TIMER2** pour écriture port série à un taux de Baud standard ( $F_{osc} = 12 \text{ MHz}$ ):
  - 1.1. Définir le baud rate **9600** pour la communication en mode Auto-Reload.
  - 1.2. Monter dans le CR que le Baud rate est respecté.

# Taches à accomplir pour le TP2

## 2

2. Corriger la fonction **Hex2CDU** pour décomposer la variable VITHEX en centaines, dizaines et unités (votre TP de 3<sup>ème</sup> année)

```
167 ;*****
168 Hex2CDU:
169     MOV     V2,  VITHEX ; Remplacez cette ligne pour afficher les centaines
170     MOV     V1,  VITHEX ; Remplacez cette ligne pour afficher les dizaines
171     MOV     V0,  VITHEX ; Remplacez cette ligne pour afficher les unités
172     RET
173
174 ;*****
```

3. Corriger la fonction **HEXASCII** pour afficher les valeurs correctes par UART#1 (encodage ASCII)

# ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	\$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[END OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

## Taches à accomplir pour le TP2

---

# 3

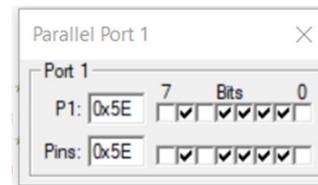
4. Utiliser les définitions des broches pour le mode A/M et '+' et '-'.
5. Employer la scrutation des broches dans la boucle principale
6. Employer le changement d'affichage de mode : 'A/M'. Il s'agit juste d'affichage, le fonctionnement sera développé lors du TP3

# Scrutation des broches

Modifier la boucle principale pour intégrer la scrutation des broches définies par :

```
18 MD      BIT      P1.0
19 R_PLUS  BIT      P1.1
20 R_MOINS BIT      P1.2
```

*Les broches sont déjà définies*



*Avant :*

```
108 ; waiting main loop
109     SJMP $           ; ZzZzZzZzZz time for a nap
110
```

*Après :*

```
110 MAIN_LOOP:           ; DEBUT BOUCLE PRINCIPAL
111
112     CALL    CHANGE_MODE
113     CALL    RAPPORT_MANUEL
114     SJMP    MAIN_LOOP   ; FIN BOUCLE PRINCIPAL
```

```
CHANGE_MODE:
    JB      MD, AUTO
    ; mettre "M"
RET

AUTO:
    ; mettre "A"
RET
```

# Scrutation des broches

---

En appuyant sur la broche MD, l'affichage bascule entre « A » et « M ». Il n'y a pas d'effet ressort (Toggle Button)

En appuyant sur les broches R\_PLUS et R\_MOINS le rapport R est incrémenté/décrémenté

Le rapport varie dans les limites 1 .. 6

## TP2 : affichage de vitesse par le port série

---

*Bon courage !*

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

Au supplément du compte-rendu il est nécessaire de joindre les projets entiers Keil  $\mu$ Vision :

**le dossier entier compressé incluant les fichiers**

**\*.uvproj et \*.a**

---

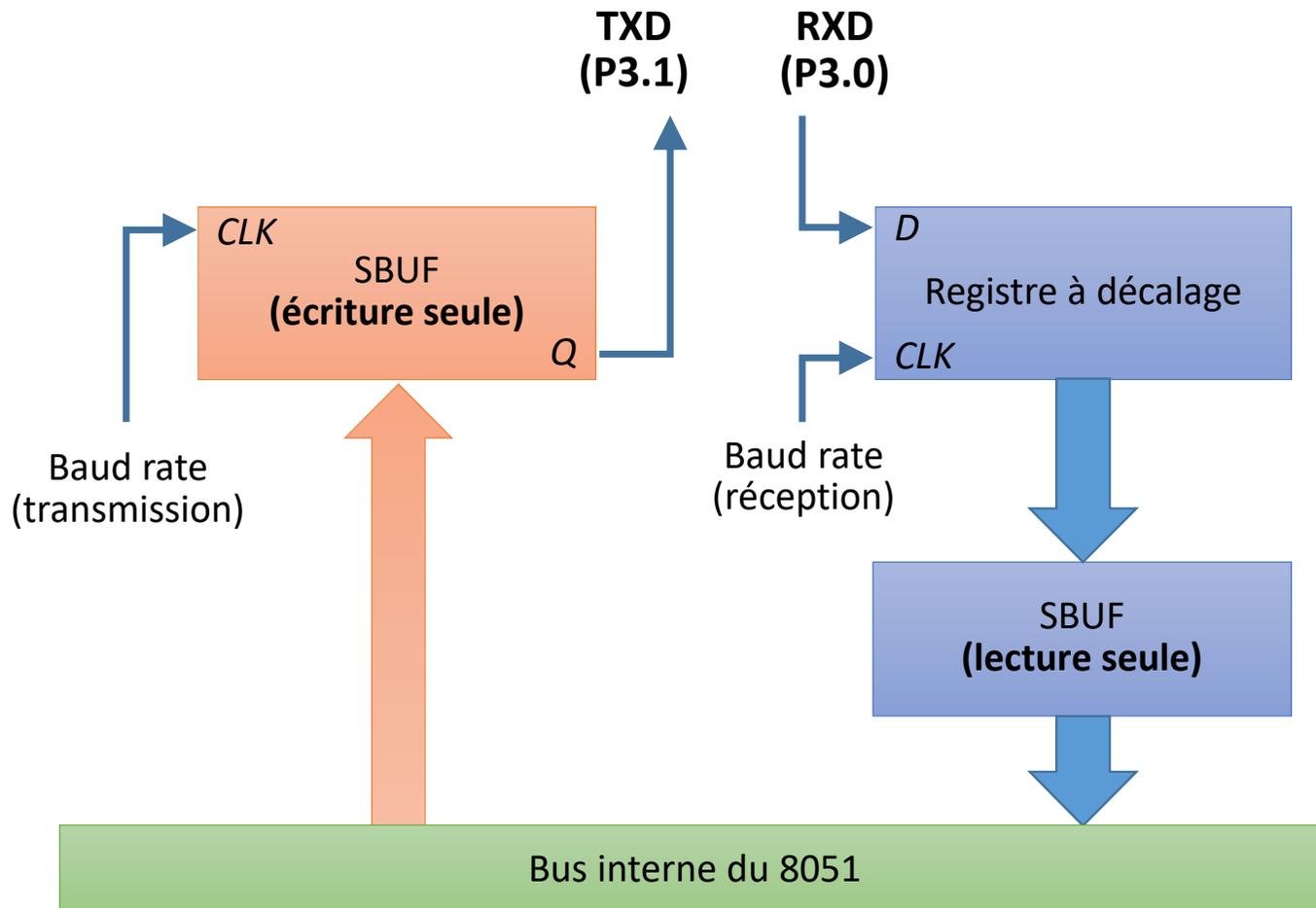
# Annexe

---

# Caractères de contrôle pour un afficheur port série

Sequence	Description
<b>From Target → μVision</b>	
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
<b>From μVision ⇒ Target</b>	
<ESC>[A	Cursor Key UP
<ESC>[B	Cursor Key DOWN
<ESC>[C	Cursor Key RIGHT
<ESC>[D	Cursor Key LEFT

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

# Communication port série par le 8051

Registre	Nom complet	Description
<b>SBUF</b>	Serial port buffer	8 bits à écrire/lire
<b>SCON</b>	Serial port control	Mode de fonctionnement, flags d'interruption, 9 <sup>ème</sup> 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 <sup>ème</sup> bit émis en modes 2 et 3. Il faut le gérer par le code.
SCON.2	RB8	9AH	(Receive bit 8). 9 <sup>ème</sup> 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 série

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