

Les GPIO du PIC16F877

cours n°3
LI326

GPIO: General Purpose Input/Output

Selon Wikipedia:

- Les ports GPIO (General Purpose Input/Output, c'est-à-dire entrée/sortie pour un usage général) sont des ports d'entrée/sortie très utilisés dans le monde des microcontrôleurs, en particulier dans le domaine de l'électronique embarquée. Les périphériques GPIO comportent un ensemble de ports d'entrée/sortie qui peuvent être configurés pour jouer soit le rôle d'une entrée, soit le rôle d'une sortie.
- Lorsqu'un port GPIO est configuré en tant que sortie, on peut écrire dans un registre interne afin de modifier l'état d'une sortie. Lorsqu'il est configuré en tant qu'entrée, on peut détecter son état en lisant le contenu d'un registre interne.
- De plus, les périphériques GPIO peuvent produire des interruptions et des événements d'accès direct à la mémoire.

Plan

- Les GPIO du pic
- Lecture de la documentation
- Usages
 - Commande de LEDs
 - Afficheur 7-segments
 - Bouton poussoir
 - Clavier matriciel
 - Clavier 1 fil

Registres impliqués

File Address		File Address		File Address		File Address	
Indirect addr. ^(*)	00h	Indirect addr. ^(*)	80h	Indirect addr. ^(*)	100h	Indirect addr. ^(*)	180h
TMR0	01h	OPTION_REG	81h	TMR0	101h	OPTION_REG	181h
PCL	02h	PCL	82h	PCL	102h	PCL	182h
STATUS	03h	STATUS	83h	STATUS	103h	STATUS	183h
FSR	04h	FSR	84h	FSR	104h	FSR	184h
PORTA	05h	TRISA	85h		105h		185h
PORTB	06h	TRISB	86h	PORTB	106h	TRISB	186h
PORTC	07h	TRISC	87h		107h		187h
PORTD ⁽¹⁾	08h	TRISD ⁽¹⁾	88h		108h		188h
PORTE ⁽¹⁾	09h	TRISE ⁽¹⁾	89h		109h		189h
PCLATH	0Ah	PCLATH	8Ah	PCLATH	10Ah	PCLATH	18Ah
INTCON	0Bh	INTCON	8Bh	INTCON	10Bh	INTCON	18Bh
PIR1	0Ch	PIE1	8Ch	EEDATA	10Ch	EECON1	18Ch
PIR2	0Dh	PIE2	8Dh	EEADR	10Dh	EECON2	18Dh
TMR1L	0Eh	PCON	8Eh	EEDATH	10Eh	Reserved ⁽²⁾	18Eh
TMR1H	0Fh		8Fh	EEADRH	10Fh	Reserved ⁽²⁾	18Fh
T1CON	10h		90h		110h		190h
TMR2	11h	SSPCON2	91h		111h		191h
T2CON	12h	PR2	92h		112h		192h
SSPBUF	13h	SSPADD	93h		113h		193h
SSPCON	14h	SSPSTAT	94h		114h		194h
CCPR1L	15h		95h		115h		195h
CCPR1H	16h		96h		116h		196h
CCP1CON	17h		97h	General Purpose Register 16 Bytes	117h	General Purpose Register 16 Bytes	197h
RCSTA	18h	TXSTA	98h		118h		198h
TXREG	19h	SPBRG	99h		119h		199h
RCREG	1Ah		9Ah		11Ah		19Ah
CCPR2L	1Bh		9Bh		11Bh		19Bh
CCPR2H	1Ch		9Ch		11Ch		19Ch
CCP2CON	1Dh		9Dh		11Dh		19Dh
ADRESH	1Eh	ADRESL	9Eh		11Eh		19Eh
ADCON0	1Fh	ADCON1	9Fh		11Fh		19Fh
	20h		A0h		120h		1A0h
General Purpose Register 96 Bytes		General Purpose Register 80 Bytes		General Purpose Register 80 Bytes		General Purpose Register 80 Bytes	
	7Fh	accesses 70h-7Fh	EFh	accesses 70h-7Fh	16Fh	accesses 70h - 7Fh	1EFh
			F0h		170h		1F0h
			FFh		17Fh		1FFh
Bank 0		Bank 1		Bank 2		Bank 3	

- 5 ports: A,B,C,D,E
- Tous les ports ne sont pas égaux.
- Certains ports peuvent lever des interruptions.

La documentation officielle

Pour chaque port:

- Une description de la fonction
- Un schéma représentatif du comportement
- La liste des registres impliqués
 - La définition des registres spéciaux généraux se trouve aussi dans le chapitre sur la mémoire

PIC16F87X

3.0 I/O PORTS

Some pins for these I/O ports are multiplexed with an alternate function for the peripheral features on the device. In general, when a peripheral is enabled, that pin may not be used as a general purpose I/O pin.

Additional information on I/O ports may be found in the PICmicro™ Mid-Range Reference Manual, (DS33023).

3.1 PORTA and the TRISA Register

PORTA is a 6-bit wide, bi-directional port. The corresponding data direction register is TRISA. Setting a TRISA bit (= 1) will make the corresponding PORTA pin an input (i.e., put the corresponding output driver in a Hi-Impedance mode). Clearing a TRISA bit (= 0) will make the corresponding PORTA pin an output (i.e., put the contents of the output latch on the selected pin).

Reading the PORTA register reads the status of the pins, whereas writing to it will write to the port latch. All write operations are read-modify-write operations. Therefore, a write to a port implies that the port pins are read, the value is modified and then written to the port data latch.

Pin RA4 is multiplexed with the Timer0 module clock input to become the RA4/T0CKI pin. The RA4/T0CKI pin is a Schmitt Trigger input and an open drain output. All other PORTA pins have TTL input levels and full CMOS output drivers.

Other PORTA pins are multiplexed with analog inputs and analog VREF input. The operation of each pin is selected by clearing/setting the control bits in the ADCON1 register (A/D Control Register1).

Note: On a Power-on Reset, these pins are configured as analog inputs and read as '0'.

The TRISA register controls the direction of the RA pins, even when they are being used as analog inputs. The user must ensure the bits in the TRISA register are maintained set when using them as analog inputs.

EXAMPLE 3-1: INITIALIZING PORTA

```
BCF STATUS, RP0 ;
BCF STATUS, RP1 ; Bank0
CLRF PORTA      ; Initialize PORTA by
                ; clearing output
                ; data latches
BSF STATUS, RP0 ; Select Bank 1
MOVLW 0x06      ; Configure all pins
MOVWF ADCON1    ; as digital inputs
MOVLW 0xCF      ; Value used to
                ; initialize data
                ; direction
MOVWF TRISA     ; Set RA<3:0> as inputs
                ; RA<5:4> as outputs
                ; TRISA<7:6> are always
                ; read as '0'.
```

FIGURE 3-1: BLOCK DIAGRAM OF RA3:RA0 AND RA5 PINS

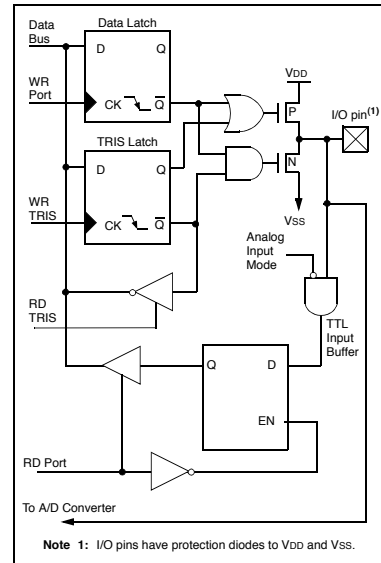
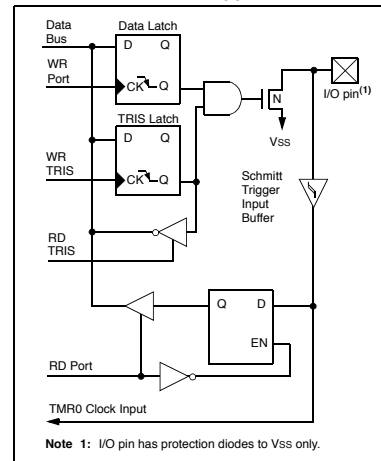


FIGURE 3-2: BLOCK DIAGRAM OF RA4/T0CKI PIN



PIC16F87X

TABLE 3-1: PORTA FUNCTIONS

Name	Bit#	Buffer	Function
RA0/AN0	bit0	TTL	Input/output or analog input.
RA1/AN1	bit1	TTL	Input/output or analog input.
RA2/AN2	bit2	TTL	Input/output or analog input.
RA3/AN3/VREF	bit3	TTL	Input/output or analog input or VREF.
RA4/T0CKI	bit4	ST	Input/output or external clock input for Timer0. Output is open drain type.
RA5/SS/AN4	bit5	TTL	Input/output or slave select input for synchronous serial port or analog input.

Legend: TTL = TTL input, ST = Schmitt Trigger input

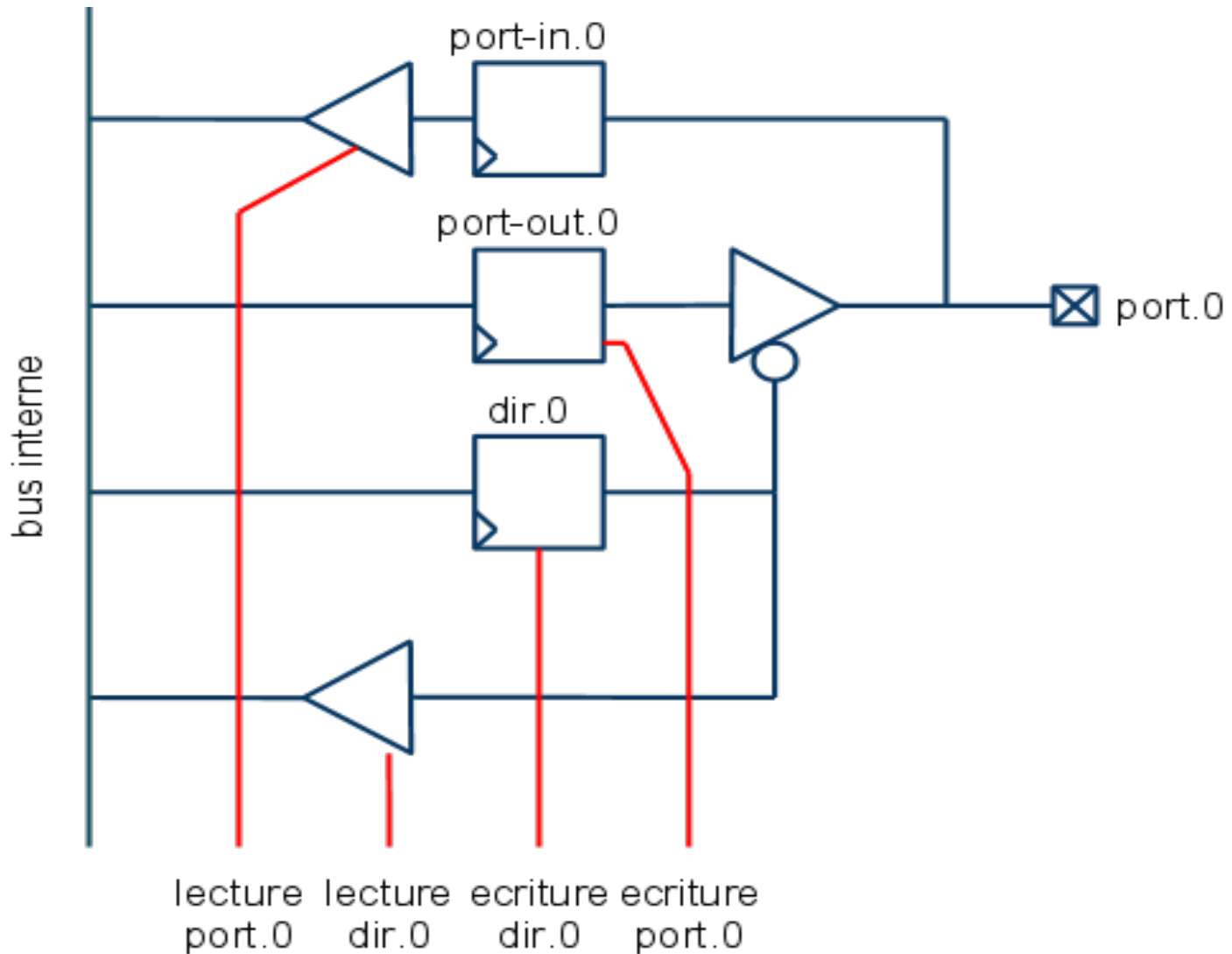
TABLE 3-2: SUMMARY OF REGISTERS ASSOCIATED WITH PORTA

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other RESETS
05h	PORTA	—	—	RA5	RA4	RA3	RA2	RA1	RA0	--0x 0000	--0u 0000
85h	TRISA	—	—	PORTA Data Direction Register						--11 1111	--11 1111
9Fh	ADCON1	ADFM	—	—	—	PCFG3	PCFG2	PCFG1	PCFG0	--0- 0000	--0- 0000

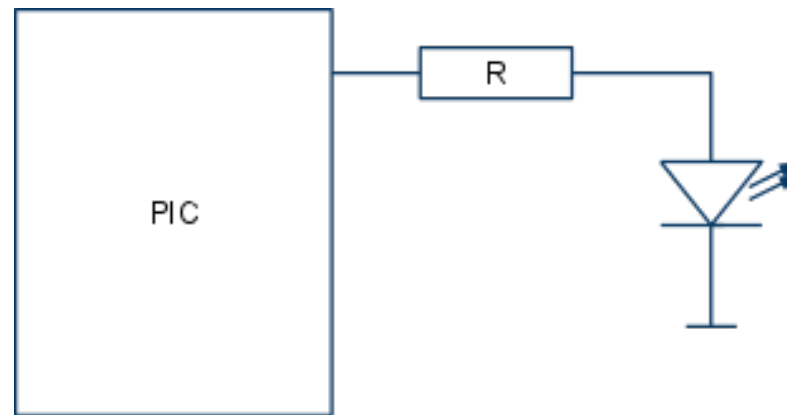
Legend: x = unknown, u = unchanged, - = unimplemented locations read as '0'.
Shaded cells are not used by PORTA.

Note: When using the SSP module in SPI Slave mode and SS enabled, the A/D converter must be set to one of the following modes, where PCFG3:PCFG0 = 0100, 0101, 011x, 1101, 1110, 1111.

GPIO exemple du port D

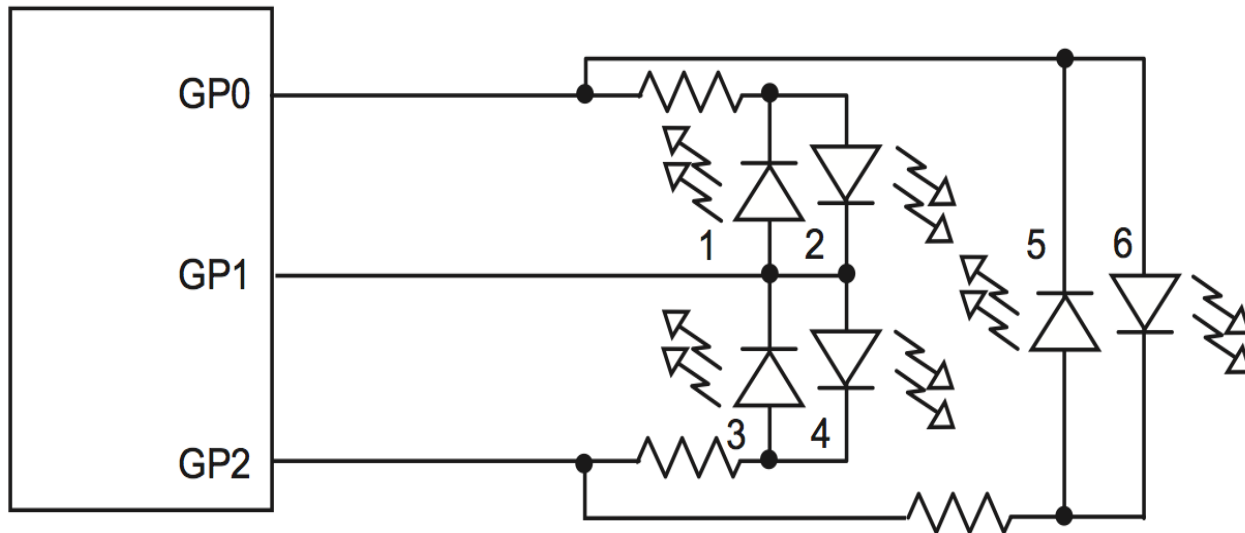


Affichage par LEDs



- 1 sur la broche allume la LED
- 0 éteint la LED
- La résistance permet de limiter le courant (entre 2 et 20mA)

Commandes 6 LEDs



GPx			LEDs					
0	1	2	1	2	3	4	5	6
0	0	0	0	0	0	0	0	0
0	1	Z	1	0	0	0	0	0
1	0	Z	0	1	0	0	0	0
Z	0	1	0	0	1	0	0	0
Z	1	0	0	0	0	1	0	0
0	Z	1	0	0	0	0	1	0
1	Z	0	0	0	0	0	0	1

Les GPIOs peuvent avoir 3 états : 0, 1 et Z

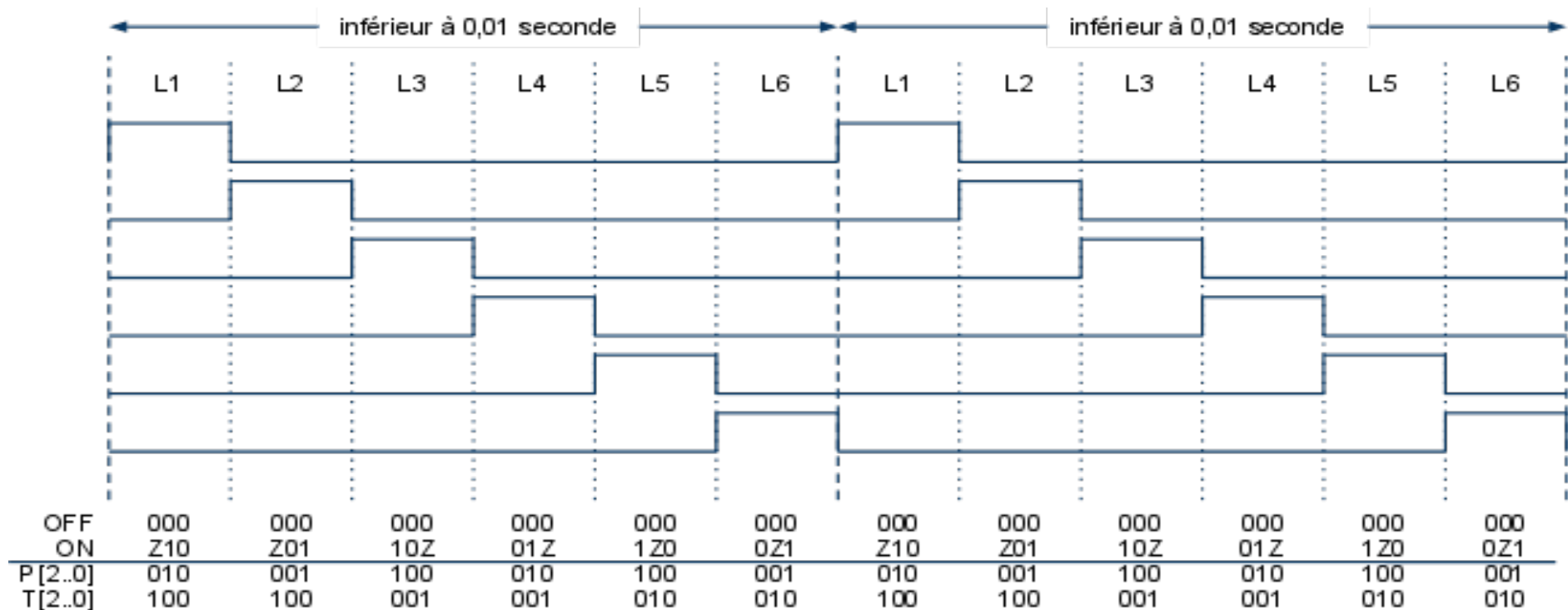
Ici, on peut allumer 1 led parmi les 6.

En généralisant, avec GP ports, on peut commander $GP*(GP-1)$ LEDs

- 2 LEDs avec 2 GPs, 6 avec 3, 12 avec 4, 20 avec 5, ...

Multiplexage temporel

- Les LEDs s'allument et s'éteignent instantanément.
- L'œil humain ne distingue pas les clignotements $>$ à 100Hz.
- Si on allume les LEDs à tour de rôle à une fréquence supérieure à 100Hz on peut "voir" plusieurs leds allumées.
- avec 6 leds:



Algorithme d'affichage

soit

- Vaff une valeur sur 6 bits à afficher
- Baff le numéro du bit à afficher (sera compris entre 0 et 5)
- Paff un tableau de 6 cases avec les codes à mettre sur les Ports pour afficher 1
Paff [6] = {010,001,100,010,100,001};
- Taff un tableau de 6 cases avec les codes à mettre sur les Dir (Tris) pour afficher 1
Taff [6] = {100,100,001,001,010,010};
- P La valeur du port
- T La direction du port

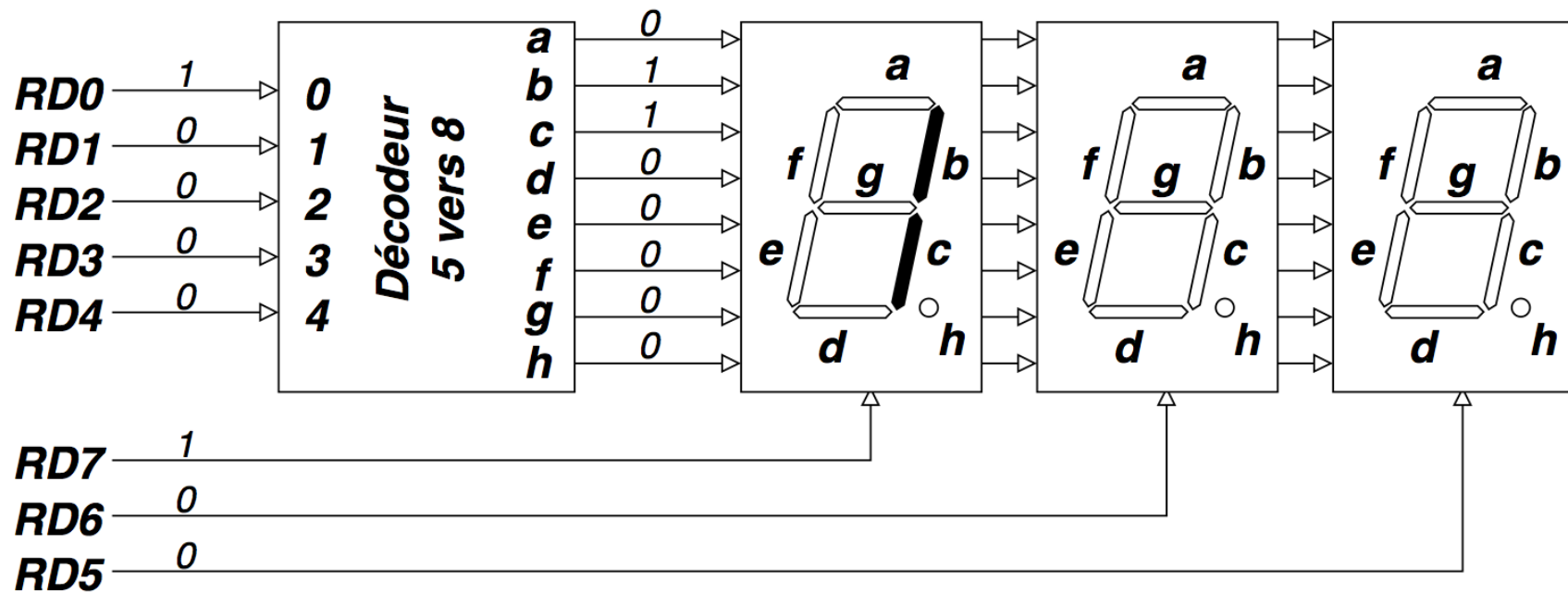
algo:

```
faire périodiquement (p. ex. 10ms)
  P = 0; T = 0;
  si ((Vaff >> Baff) & 1) alors
    T = Taff [ Baff ];
    P = Paff [ Baff ];
  finsi
  Baff = ( Baff + 1 ) % 6;
finfaire
```

Afficheurs 7 segments

commande de 3 afficheurs 7-segments (branchés sur le port D)
grâce au multiplexage temporel.

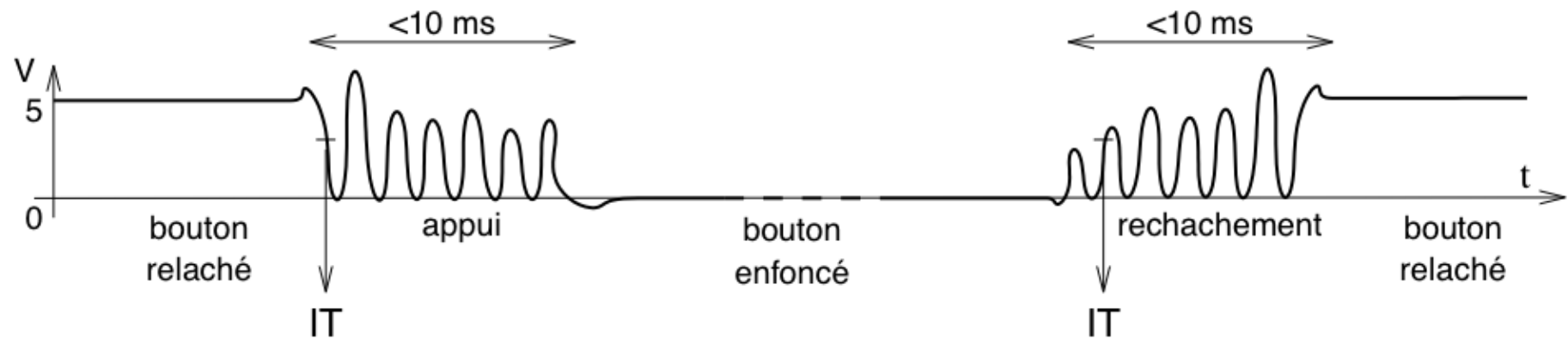
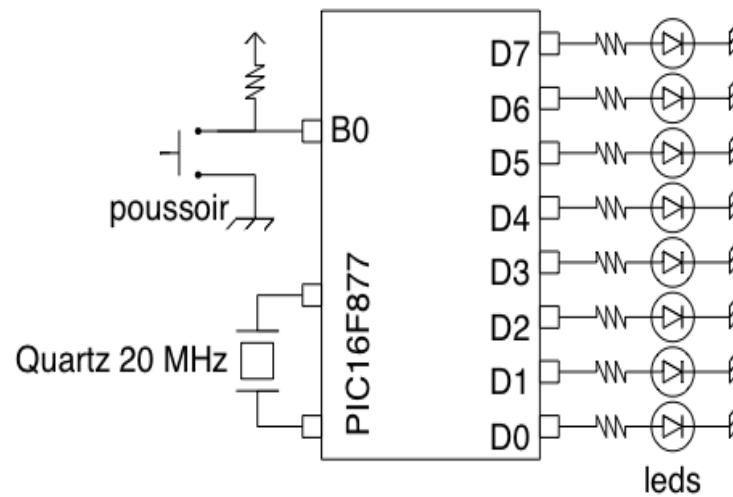
Par exemple, si le PORTD présente la valeur B' 10000001', cela affiche « 1... »



Le bouton poussoir

Une entrée sur 1 bit numérique:

- relaché : 1
- appuyé : 0



Traitement des rebonds

Il suffit d'échantillonner à une période supérieure aux rebonds, mais inférieure à la durée d'appui

soit

- B : une variable
- E : la valeur du bouton poussoir
- appui : un drapeau à 1 si appui
- relache : un drapeau à 1 si relachement

faire périodiquement

```
B = (B << 1) & E
```

```
si (B & 3) == 2 alors
```

```
  appui = 1
```

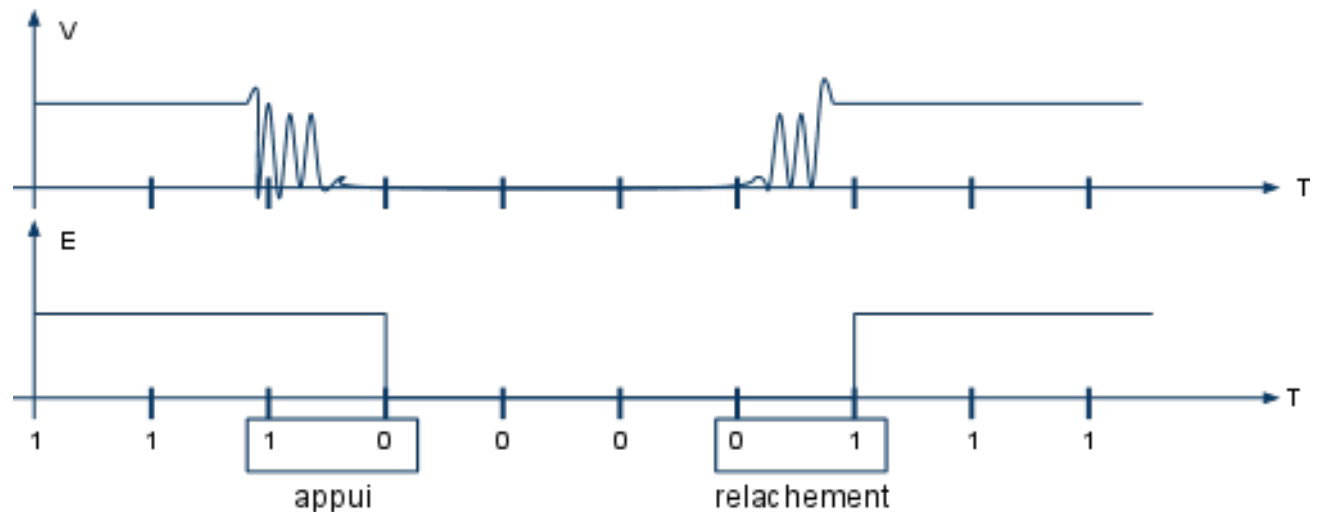
```
finsi
```

```
si (B & 3) == 1 alors
```

```
  relache == 1
```

```
finsi
```

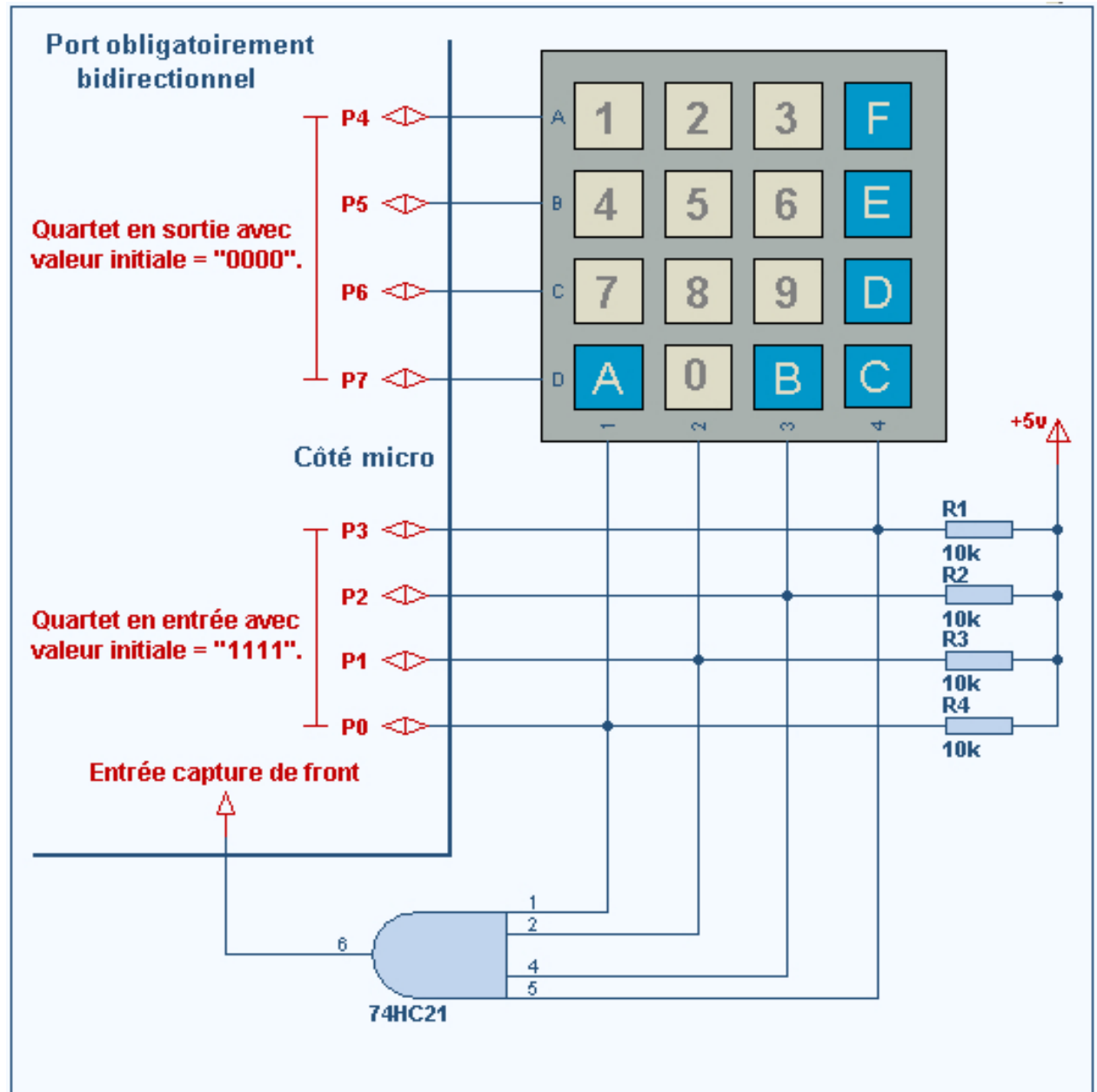
```
finfaire
```



Clavier matriciel

P4 à P7 à 0 permet de détecter un appui

puis on place une des lignes à 0 les autres en entrée et on lit P0 à P3



clavier 1 fil

- La charge d'une capacité est "proportionnelle" à la résistance de charge :
si on double la resistance on double le temps de charge

- Algorithme

GP2 = 0

compteur = 0

GP2 est mis en entrée

tantque GP2 != 1

compteur++

fantantque

touche = fct(compteur)

*garder les 4 bits de poids
forts du compteur*

