

Manipulations de registres



Systèmes Numériques

Isn @ hepia

Fabien Vannel

Accès aux périphériques

- Le processeur peut utiliser différentes méthodes pour contrôler les registres des les périphériques.
 - Ecriture directe de tous les bits du registre
 - Modification de certains bits
- Tous les périphériques du LPC1769 sont accessibles en 32 bits.
- Les périphériques disposent de plusieurs registres pouvant être en lecture uniquement (RO), écriture uniquement (WO) ou lecture/écriture (R/W)

Ecriture dans un registre

- Il est possible d'écrire l'intégralité des 32 bits.
- De modifier certains bit en relisant le contenu du registre
 - Attention : ceci ne peut être fait que pour les registre du type R/W

1 – Bases sur les nombres en C

Affectation d'une constante à une variable ou à un registre de périphérique:

On utilisera toujours la méthode la plus lisible pour le programmeur selon les besoins. *Pour rappel l'ordinateur ne connaît que 0 et 1*

Exemple pour affecter une valeur à une variable :

```
uint32_t  a ;  
a = 1073741843;  
ou      a = 2^31 + 19;
```

Pour affecter une valeur à un registre de périphérique :
toujours du type uint32_t :

```
FIO2DIR = 0x40000013;  
ou      FIO1SET = 0b110;
```

1 – Bases sur les opérateurs en C

Pour cet exemple

`uint8_t a;`

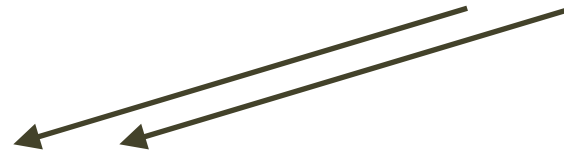
Opération	Description	symbole	exemple	Résultat
NOT	Inverse bit à bit	<code>~</code>	<code>a = ~ 0x11;</code>	<code>0b11101110</code>
AND	ET bit à bit	<code>&</code>	<code>a = 0x11 & 0xF;</code>	<code>0x1;</code>
OR	OU bit à bit	<code> </code>	<code>a = 0x11 0xF;</code>	<code>0x1F;</code>
SHIFT ←	Décalage à gauche	<code><<</code>	<code>a = 0x1 << 4;</code>	<code>0x10;</code>
SHIFT →	Décalage à droite	<code>>></code>	<code>a = 0x10 >> 4;</code>	<code>0x1;</code>

1 - Opérations binaires

- Affectation d'un seul bit à 1

0x1

31	30	29	...	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	1



0x1<< 4

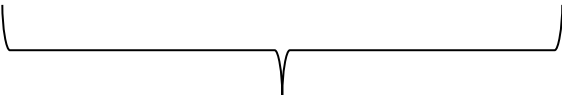
31	30	29	...	5	4	3	2	1	0
0	0	0	0	0	1	0	0	0	0

2 – Suite de plusieurs bits contigus à l'état 1

- P1.20 à P1.15 doivent être des sorties. Les autres pattes du port P1 restent des entrées.

FIO1DIR

31	...	21	20	19	18	17	16	15	14	...	0
0	0	0	1	1	1	1	1	1	0	0	0



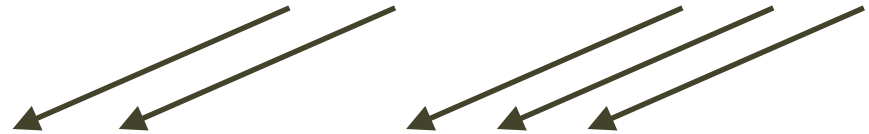
6 bits consécutifs à 1 = 0b111111 (ou 0x3F)

2 - Opérations binaires

- Affectation de 6 bits consécutifs à 1

0x3F

31	...	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	1	1	1	1	1	1



0x3F << 15

31	...	21	20	19	18	17	16	15	14	...	0
0	0	0	1	1	1	1	1	1	0	0	0

FIO1DIR = 0x3F << 15; 

3 – Plusieurs bits (non contigus) à l'état 1

- P1.20 à P1.15 doivent être des sorties ainsi que P1.0 Les autres pattes du port P1 restent des entrées.

FIO1DIR

31	...	21	20	19	18	17	16	15	14	...	0
0	0	0	1	1	1	1	1	1	0	0	1

6 bits consécutifs à 1 = 0x3F

3 - Opérations binaires

P1.20 à P1.15 en sorties

$0x3F \ll 15$

31	...	21	20	19	18	17	16	15	14	...	0
0	0	0	1	1	1	1	1	1	0	0	0

P1.0 en sortie

$0x1$

31	...	21	20	19	18	17	16	15	14	...	0
0	0	0	0	0	0	0	0	0	0	0	1

$FIO1DIR = (0x3F \ll 15) | 0x1;$



4 – Modification d'un bit à l'état 1

- P1.20 à P1.15 ainsi que P1.0 sont déjà configurés en tant que sorties. Les autres pattes du port P1 étant des entrées.
- Modifier P1.1 en tant que sortie

FIO1DIR

31	...	21	20	19	18	17	16	15	14	...	1	0
0	0	0	1	1	1	1	1	1	0	0	1	1

4 - Opérations binaires

Etat actuel de FIO1DIR

$(0x3F \ll 15) | 0x1$

31	...	21	20	19	18	17	16	15	14	...	1	0
0	0	0	1	1	1	1	1	1	0	0	0	1

$0x1 \ll 1$
P1.1 en sortie

0	0	0	0	0	0	0	0	0	0	0	1	0
---	---	---	---	---	---	---	---	---	---	---	---	---

OU (bit à bit)

0	0	0	1	1	1	1	1	1	0	0	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---

$FIO1DIR = FIO1DIR | (0x1 \ll 1);$

$FIO1DIR |= (0x1 \ll 1);$



5 - Modification de bits contigus à l'état 0

- P1.20 à P1.15 ainsi que P1.0 sont déjà configurés en tant que sorties. Les autres pattes du port P1 étant des entrées.
- Modifier à présent P1.18 à P1.17 en entrée

FIO1DIR

31	...	21	20	19	18	17	16	15	14	...	0
0	0	0	1	1	0	0	1	1	0	0	1

Solution intuitive : ~~$\text{FIO1DIR} |= (\text{0b00} \ll 17);$~~

Car équivalent à : $\text{FIO1DIR} |= 0;$

Ce qui ne modifie rien !

5 - Opérations binaires

Etat actuel de FIO1DIR

$(0x3F \ll 15) | 0x1$

31	...	21	20	19	18	17	16	15	14	...	1	0
0	0	0	1	1	1	1	1	1	0	0	0	1

$0x3 \ll 17$

0	0	0	0	0	1	1	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---

$\sim(0x3 \ll 17)$

1	1	1	1	1	0	0	1	1	1	1	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---

ET (bit à bit)

0	0	0	1	1	0	0	1	1	0	0	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---

$FIO1DIR = FIO1DIR \& \sim(0x3 \ll 17);$

$FIO1DIR \&= \sim(0x3 \ll 17);$



6 - Modification de bits non contigus à l'état 0

- P1.20 à P1.15 ainsi que P1.0 sont déjà configurés en tant que sorties. Les autres pattes du port P1 étant des entrées.
- Exercice : Modifier à présent P1.18 à P1.17 en entrée ainsi que P1.0

FIO1DIR

31	...	21	20	19	18	17	16	15	14	...	0
0	0	0	1	1	0	0	1	1	0	0	0

Réponse :

Modifications de bits dans un registre

- Pour certains type de registre il est possible de modifier leur contenu au travers d'autre registres
- Exemple : FIOPIN peut être manipulé par écriture des 32 bits simultanés (similaires aux transparents précédent)
- Le registre FIOSET permet de modifier certains bits à l'état 1 de FIOPIN et de ne pas modifier l'état des autres bits.
- Le registre FIOCLR permet de modifier certains bits à l'état 0 de FIOPIN et de ne pas modifier l'état des autres bits.

Mise à 1 de bits spécifiques

Etat actuel de FIO2PIN

31	...	10	9	8	7	6	5	4	3	2	1	0
0	..	0	1	1	0	0	1	0	0	1	0	1

Écriture dans FIO2SET

0	0	0	0	0	1	1	0	0	0	1	1	0
---	---	---	---	---	---	---	---	---	---	---	---	---

Nouvel état de FIO2PIN

0	..	0	1	1	1	1	1	0	0	1	1	1
---	----	---	---	---	---	---	---	---	---	---	---	---

Mise à 0 de bits spécifiques

Etat actuel de FIO2PIN

31	...	10	9	8	7	6	5	4	3	2	1	0
0	..	0	1	1	1	1	1	0	0	1	1	1

Écriture dans FIO2CLR

0	0	0	0	1	1	0	0	0	1	0	1	0
---	---	---	---	---	---	---	---	---	---	---	---	---

Nouvel état de FIO2PIN

0	..	0	1	0	0	1	1	0	0	1	0	1
---	----	---	---	---	---	---	---	---	---	---	---	---

7 – Isolement de bit (les masques)

Est-ce que l'entrée P1.8 est à l'état 1

Etat actuel de FIO1PIN

31	...	11	10	9	8	7	6	5	4	...	1	0
0	1	0	1	0	1	1	1	1	0	0	1	1

$1 \ll 8$

0	0	0	0	0	1	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---

ET (bit à bit)

0	0	0	0	0	1	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---

```
if ( FIO1PIN & (1 << 8))  
    {...}
```

7 – Isolement de bit (les masques)

Tester si les valeurs des entrée P1.8 à P1.6 sont à l'état

0

Etat actuel de FIO1PIN

31	...	11	10	9	8	7	6	5	4	...	1	0
0	1	0	1	0	0	0	0	1	0	0	1	1

0b1111 << 6

0	0	0	0	0	1	1	1	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---

ET (bit à bit)

0	0	0	0	0	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---

if !(FIO1PIN & (0b111 << 6))
{...}