

TP 4 Programme assembleur pour piloter les 4 LEDs du KIT de développement ATMEL

- 1- Réaliser une procédure delay(t) en assembleur (t : temps d'attente ou délai).
- 2- Réaliser un programme en assembleur permettant de faire clignoter les 4 LEDs de la carte AT91SAMS256 avec un délai variable.
- 3- Réaliser un programme qui affiche en binaire une séquence de 8 chiffres hexadécimaux codés en ASCII. Il affiche chiffre par chiffre sa représentation binaire en allumant les 4 LEDs. Le programme utilisera la procédure delay(t).

Exemple : '3B...' ●●☀☀ ☀☀●● ...

Annexes

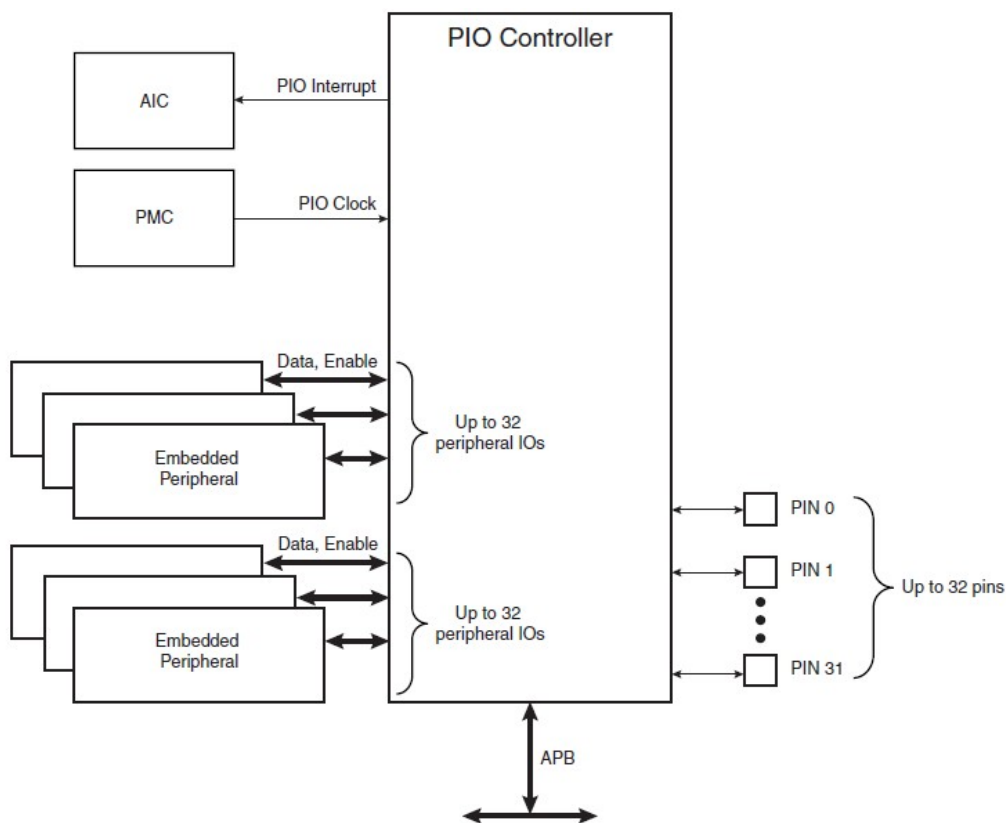


Figure 1 : Schéma synoptique du contrôleur PIO

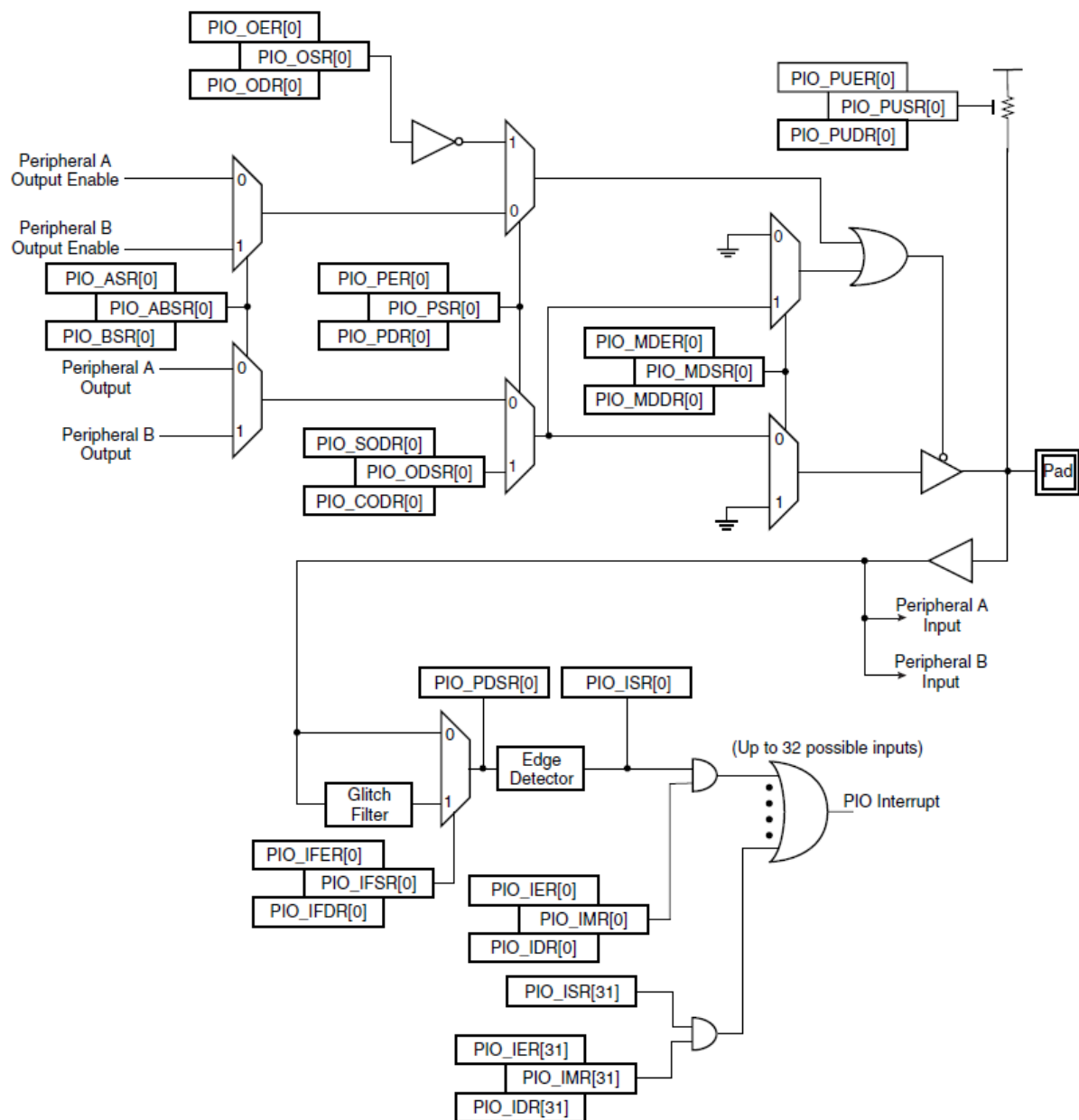


Figure 2 : Schéma fonctionnel du contrôleur PIO

Table 15-2. Register Mapping

| Offset | Register | Name | Access | Reset |
|--------|--------------------------------------|----------|----------------------------------|-------------|
| 0x0000 | PIO Enable Register | PIO_PER | Write-only | – |
| 0x0004 | PIO Disable Register | PIO_PDR | Write-only | – |
| 0x0008 | PIO Status Register | PIO_PSR | Read-only | (1) |
| 0x000C | Reserved | | | |
| 0x0010 | Output Enable Register | PIO_OER | Write-only | – |
| 0x0014 | Output Disable Register | PIO_ODR | Write-only | – |
| 0x0018 | Output Status Register | PIO_OSR | Read-only | 0x0000 0000 |
| 0x001C | Reserved | | | |
| 0x0020 | Glitch Input Filter Enable Register | PIO_IFER | Write-only | – |
| 0x0024 | Glitch Input Filter Disable Register | PIO_IFDR | Write-only | – |
| 0x0028 | Glitch Input Filter Status Register | PIO_IFSR | Read-only | 0x0000 0000 |
| 0x002C | Reserved | | | |
| 0x0030 | Set Output Data Register | PIO_SODR | Write-only | – |
| 0x0034 | Clear Output Data Register | PIO_CODR | Write-only | |
| 0x0038 | Output Data Status Register | PIO_ODSR | Read-only or(2) Read-write | – |
| 0x003C | Pin Data Status Register | PIO_PDSR | Read-only | (3) |
| 0x0040 | Interrupt Enable Register | PIO_IER | Write-only | – |
| 0x0044 | Interrupt Disable Register | PIO_IDR | Write-only | – |
| 0x0048 | Interrupt Mask Register | PIO_IMR | Read-only | 0x00000000 |
| 0x004C | Interrupt Status Register(4) | PIO_ISR | Read-only | 0x00000000 |
| 0x0050 | Multi-driver Enable Register | PIO_MDER | Write-only | – |
| 0x0054 | Multi-driver Disable Register | PIO_MDDR | Write-only | – |
| 0x0058 | Multi-driver Status Register | PIO_MDSR | Read-only | 0x00000000 |
| 0x005C | Reserved | | | |
| 0x0060 | Pull-up Disable Register | PIO_PUDR | Write-only | – |
| 0x0064 | Pull-up Enable Register | PIO_PUER | Write-only | – |
| 0x0068 | Pad Pull-up Status Register | PIO_PUSR | Read-only | 0x00000000 |
| 0x006C | Reserved | | | |

Figure 3 : Table d’adressage des différents registres du PIO

Fonctionnement des LEDs

Les 4 LEDs sont connectées aux PIO conformément au schéma électrique suivant :

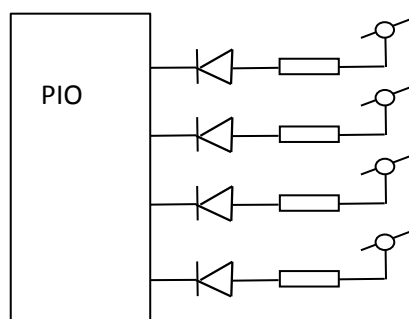



Figure 4 : Schéma de connexion des 4 LEDs

Principe de fonctionnement

Lorsque la sortie du port parallèle 'PIO' est égale à '0' la LED s'allume, car la tension aux bornes de la diode est supérieure à 0,7V. Pour éteindre la diode la sortie du PIO doit être à 1.

| | | | |
|---|-----|----------------------------|---|
| 48 | 30h | Set Output Data Register | les bits à 1 imposent une sortie à 1 (éteindre LED) |
|  | 34h | Clear Output Data Register | les bits à 1 imposent une sortie à 0 (allumer LED) |

Avant de pouvoir écrire dans les registres du PIO, il faut initialiser le PIO :

- Validation de la sortie
- Direction = sortie

Exemple de programmation

Les instructions suivantes doivent être utilisées pour initialiser le PIO :

```
; Constantes
ADR_GPIO      EQU 0xFFFFF400 ; adresse de base du PIO
GPIO_OUTPUT   EQU 0xF        ; '1' la direction du PIO est initialisée en sortie
LED1          EQU 0x8        ; DS4 la LED est connectée au PIO n° 4
LED2          EQU 0x4        ; DS3
LED3          EQU 0x2        ; DS2
LED4          EQU 0x1        ; DS1

; Code

      LDR R0,=ADR_GPIO        ; R0 adresse de base GPIO
      LDR R1,=GPIO_OUTPUT     ; R1 direction de sortie des LEDs
      STR R1,[R0,#0]          ; valider le GPIO (voir annexe jointe)
      STR R1,[R0,#16]         ; Valider le GPIO en Sortie

; Exemple : allumer puis éteindre une diode

      LDR R1,=LED1            ; R1 = LED1
      STR R1,[R0,#48]         ; 30h Eteindre LED1
      STR R1,[R0,#52]         ; 34h Allumer LED1
```