|  |  |  |  |
| --- | --- | --- | --- |
| 4 | Fiche de synthèse | Port d’entrées / sorties | Mots clé : Microcontrôleur, ports d’entrées/sorties,Buffer, bidirectionnel, configurer, écrire, lire, DDRx, PORTx, PINx. |

Bascule n appartenant au registre de direction des données (**DDRx)**

**PXn**

Dn Qn

**DDRXn**

D’n Q’n

**PORTXn**

Bascule n appartenant au registre des données (**PORTx)**

B

U

S

D

E

D

O

N

N

E

E

S

B

U

S

D

E

C

O

N

T

R

O

L

Structure logique partielle de la **cellule d’E/S**

Buffer d’écriture

**Buffer de lecture**

**Les bus sont contrôlés par le programme implanté dans le µC.**

Schéma partiel de la structure logique reliée à une broche d’E/S

Les **PORT d’ENTREES/SORTIES** permettent aux µC d’échanger des informations avec leur environnement. Un port x contient un registre de direction des données appelé **DDRx et** un registre de sortie appelé **PORTx**. Un registre est constitué de **BASCULES.**

Le **sens** du transfert des informations sur les broches du composant est contrôlé par le registre de direction de donnée. Si la sortie Q d’une bascule est à l’état logique « 1 », la broche qui lui est associée est positionnée en **SORTIE**. Si elle est à l’état logique « 0 » la broche est positionnée en **ENTREE**.

La **CONFIGURATION** des broches des ports en entrée ou en sortie s’effectue par une commande :

**DDRx = <valeur> ;**

Exemple : DDRA = 0b00001111 ; // les quatre bits de poids faible du port A sont positionnés en sortie,

// les quatre bits de poids fort sont positionnés en entrée

L’**ECRITURE** d’une donnée numérique dans le registre de sortie d’un port s’effectue par une commande :

**PORTx = <valeur> ;**

Exemple : PORTA = 0b00001010 ; // broches PA1 et PA3 au niveau logique 1 et les broches PA0 et PA2 au niveau logique 0.

La **LECTURE** d’une donnée numérique présente à l’entrée d’un port s’effectue par une commande :

**<variable> = PINx ;**

Exemple : Capteur = PINA ; // Mémorisation de l’état logique présent sur les broches du Port A dans la **variable** Capteur.

**Ce qu’il faut retenir sur le masquage**

Pour isoler une ou plusieurs informations binaires présentes sur un port d’E/S, il est nécessaire d’effectuer une opération de masquage entre ce port et une constante appelée « **masque »**. La mémorisation du résultat de l’opération de masquage est réalisée en l’affectant à une variable.



**<nom\_variable> ← <Lecture\_Contenu du Port> ET\_bit\_à\_bit <Mot de masquage>**

soit en **langage C** (cross compilateur CVAVR) :

<nom\_variable> = PINx & <Mot de masquage>