**Utilisation d’un clavier de PC avec la carte SSI**

**Mise en œuvre du sous programme d’interruption INT0**

**Réception**

|  |  |  |
| --- | --- | --- |
| **Producteur** | **Variable globale** | **Consommateur** |
| **Clavier de PC (IBM)** | **Prog sur carte SSI** |
| **SP INT0**  ***CODE ASCII***  **DATA (trame PS2 IBM 11 bits)**  **Prog princ**  **↑clk**  ***bit*** | **📭** <***g\_key\_ASCII***>  **🏳** <**g\_flag\_char**> |  |

**Emission**

|  |  |  |
| --- | --- | --- |
| **Producteur** | **Variable globale** | **Consommateur** |
| **Prog sur carte SSI** |  | **Clavier** |
| **↑clk**  **SP INT0**  ***Entier (10 bits)***  **Prog princ** | 📭 <**g\_trame\_TX**> | **DATA (trame Code IBM)**  **(Clavier PC)** |

**Dans le programme principal:**

* **Déclarer les bibliothèques ci-dessous**

// Ces fichiers d'en-tête et les bibliothèques associées doivent être placés dans le répertoire du projet

//-------------------------------------------------------------------------------------------------------

#include "LitClav102.h" // Sous-programme de désérialisation et de décodage de la

// trame issue du clavier de PC (102T)

#include "ssi2.h" // Sous-programme Affiche\_LCD, Lire\_BP

* **Déclarer les variables ci-dessous dans le champ « variable globale »**

//========================================================================================================

// Declare your global variables here

//--------------------------------------------------------------------------------------------------------

/\* Communication avec le clavier PS2

//--------------------------------------------------------------------------------------------------------

Le programme principal se synchronise avec la routine d'interruption ext\_int0\_isr(void) située dans la bibliothèque "LitClav.lib" grâce aux variables ci-dessous (voir le fichier d'en-tête litClav102.h)

--------------------------------------------------------------------------------------------------------\*/

unsigned char g\_key\_ASCII=0x01;// Contient le code ASCII de la touche pressée (à initialiser avec un caractère

// n'existant pas sur le clavier utilisé)

bit g\_flag\_char = FAUX; // g\_flag\_char = 1 lorsqu'une touche est appuyée

// g\_flag\_char = 0 lorsqu'une touche a été relâchée

bit g\_Sens\_transfert = RECEPTION ; // Indique le sens de la communication

// g\_Sens\_transfert = 0 => µC <--- clavier

// g\_Sens\_transfert = 1 => µC ---> clavier

unsigned int g\_Trame\_TX; // Contient la valeur à envoyer au clavier

//--------------------------------------------------------------------------------------------------------

* **Paramétrer l’IRQ0 (éventuellement avec Wizard)**

// External Interrupt(s) initialization

// INT0: On

// INT0 Mode: Falling Edge

// INT1: Off

// INT2: Off

GICR|=0x40;

MCUCR=0x02;

MCUCSR=0x00;

GIFR=0x40;

* **Autoriser les interruptions avant le début de la partie exécutive**

// Global enable interrupts

#asm("sei")

while (1)

…