

```

/*****
/*          CLAVIER  HEXADECIMAL          */
/*-----*/
/*      test si appui toutes les 20 ms      */
/*      par interruption du timer 16 bits    */
/*-----*/
/*      (c) J.N. Martin le 21/12/00        */
/*****

/* clavier hexadecimal sur PORTC */
/* interruptions du timer 16 bits */
/* affichage par printf NoICE */

/*----- fichiers entetes -----*/
#include <typesiut.h> // definit les types TOctet TMot Bool
#include <hc11.h>     // registres 68HC11
#include <i2chc.h>    // bus I2C
#include <stdio.h>    // pour printf NoICE

/*----- equivalence constantes symboliques -----*/
#define SENSLICO 0x0F // P74 in et P30 out
#define TESTLICO 0xF0 // test 4 lignes
#define BASE_VECTIT 0xC000 // base table vecteurs IT
#define EN_IT1 0x80 // autorise IT bit7 TMSK1
#define IT_OCL 0x80 // acquitte IT bit7 TFLG1
#define VINGTMS 40000 // base 20 ms horloge 500 ns

/*----- mise en place des iTs -----*/
#pragma interrupt_handler ItTimerClav

/*----- prototypes des fonctions -----*/
void BloqueMcu(void);
void InitClavierPOC(void);
void InitITimer(void);
void ItTimerClav(void);
Bool bAppui(void);

/*----- variable globale -----*/
Bool bAppClav; // echange principale - iT
TOctet oImagePORTC;

/*- definition type pointeur sur fonction vide --*/
typedef void (*TptFct)(void);
TptFct *TabVectiT = (TptFct *)BASE_VECTIT; // declaration + init

/*----- fonction principale -----*/
void main(void)
{
    InitClavierPOC(); // P74 entrent P30 sortent
    InitI2c();
    InitITimer();
    do{
        if(bAppClav == VRAI)
            printf(" Appui sur clavier\n\r");
    }while(VRAI);
}

```

```

/*----- initialise le sens de PORTC -----*/
void InitClavierPOC(void)
{
    DDRC = SENSLICO; // P74 entrent P30 sortent
}

/*----- fonction test si appui -----*/
Bool bAppui(void)
{
    Bool bTappui;
    oImagePORTC = TESTLICO;
    PORTC = oImagePORTC; // ecriture PORTC
    oImagePORTC = PORTC; // lecture PORTC

    if((oImagePORTC & TESTLICO) == TESTLICO)
        bTappui = FAUX;
    else
        bTappui = VRAI;
    return bTappui;
}

/*----- Initialise IT du timer -----*/
void InitITimer (void)
{
    INTR_OFF (); // interdit IT pdt initialisation
    TabVectiT[9] = ItTimerClav; // vecteur IT timer en 0xC009
    TMSK1 = TMSK1 | EN_IT1; // force b7 = 1 sur TMSK1
    TOC1 = VINGTMS;
    bAppClav = FAUX; // pas appui
    INTR_ON (); // autorise les ITs valides
}

/*----- fonction IT -----*/
void ItTimerClav (void)
{
    bAppClav = bAppui(); // positionne variable clavier
    TFLG1 = TFLG1 | IT_OCL; // acquittement IT
    TOC1 = TOC1 + VINGTMS; // base 20 ms
}

```