

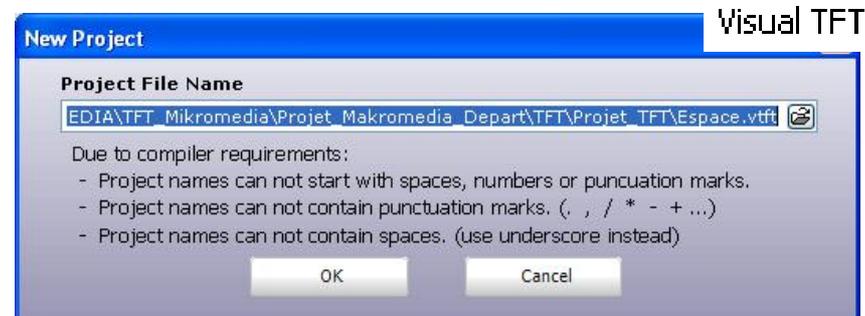
Intégration d'un projet Visual TFT dans PSoC Creator

1 Préparation des images

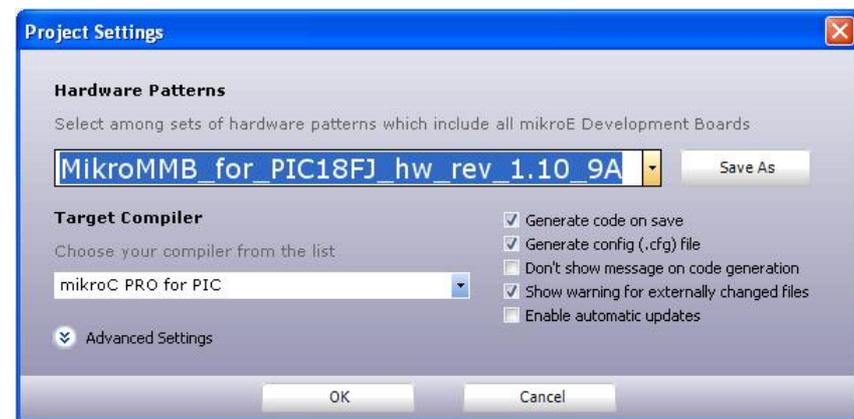
L'écran graphique est de dimension 320x240. Les images sont préparées avec cette dimension avec Paint Shop Pro.

2 Lancement du logiciel Visual TFT

↳ Choisir le dossier et le nom du projet.



↳ Sélectionner MikroMMB_for_PIC18FJ_hw_rev_1.10_9A comme microcontrôleur cible.



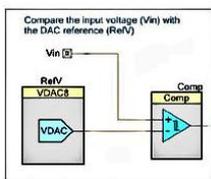
Création du GUI

Réaliser votre GUI en fonction des besoins du projet. Pour ajouter un écran supplémentaire cliquer sur la croix verte.



Renommer les boutons avec

ATTENTION les caractères accentués ne sont pas gérés.

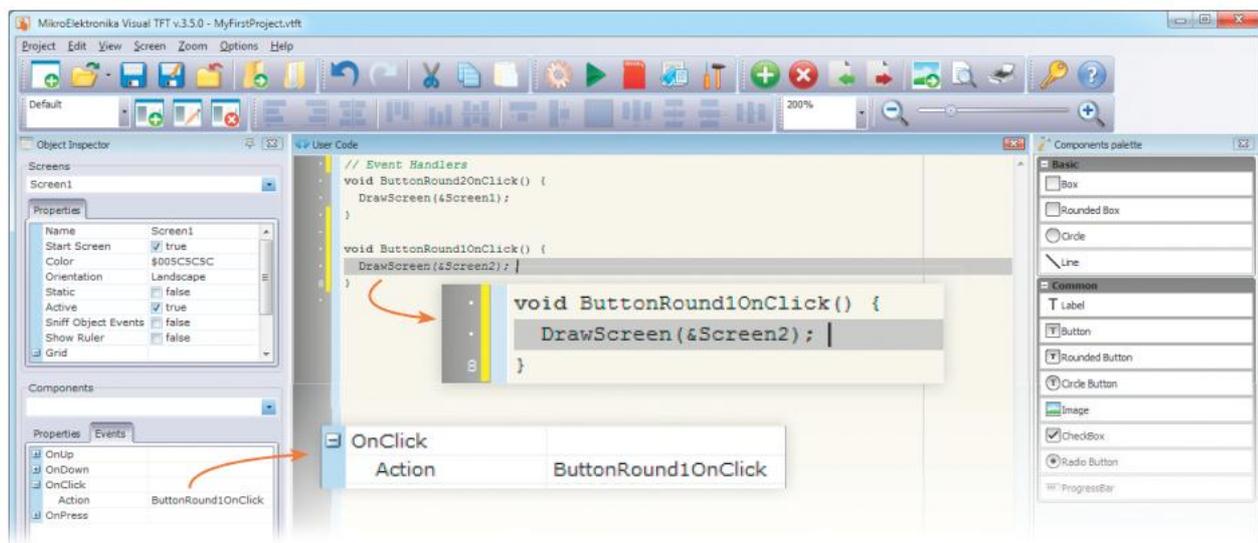


Les évènements sont ensuite ajoutés avec l'onglet event.

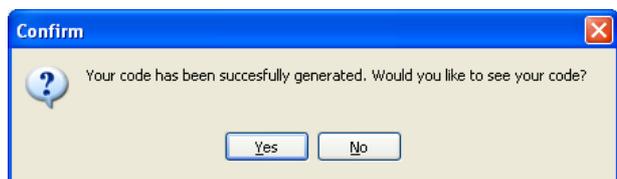
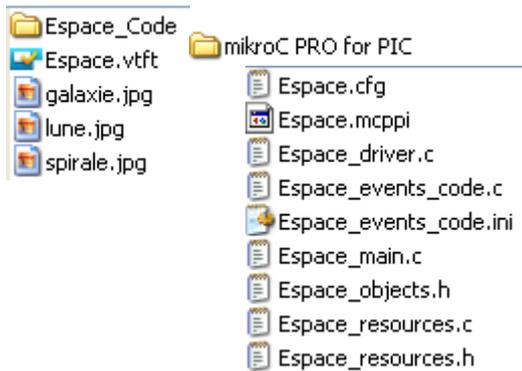
Add "OnClick" event code to ButtonRound2

It's time to specify the function of the buttons when clicked. In order to do that we will add **OnClick** events to both buttons. Locate the OnClick event of the **ButtonRound2** in the of the Events Tab of **Object Inspector**. Double click it. The **User Code window** will appear. It will contain the function

prototype that is automatically associated with the click event. In the function body just type the following line of code: **"DrawScreen(&Screen1);"**. This code will be executed when the button is clicked, thus invoking the drawing of Screen1.

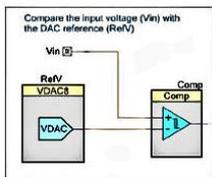


Génération du code du projet nommé espace



Le projet GUI est terminé

Graphical User Interface

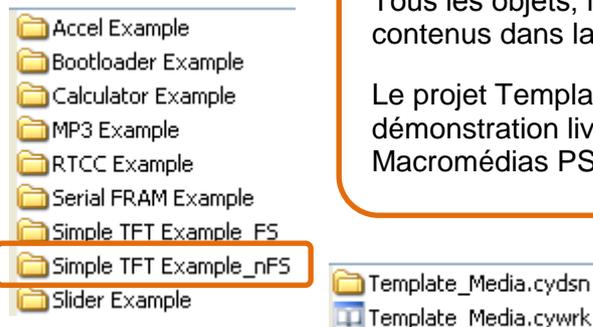


3 Intégration dans PSoC Creator

Nous allons intégrer les éléments du GUI dans un fichier de démo fourni par Mikroelectronica.

Nous prendrons la démonstration

{PSoC Creator 3.0 SP1}\BootLoadable



_nFS no File System

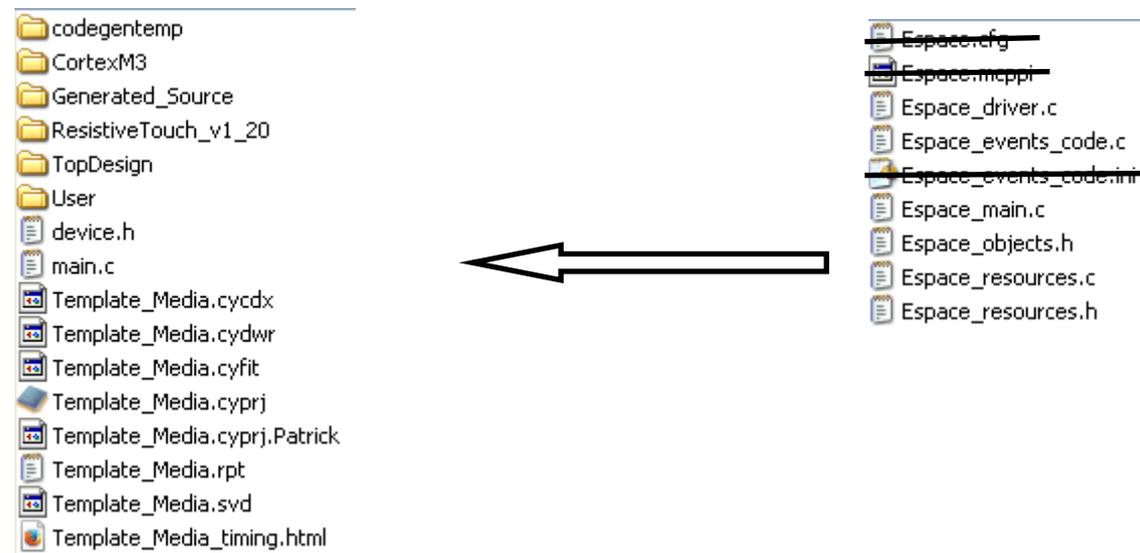
Tous les objets, images, ... sont contenus dans la mémoire du PSoC.

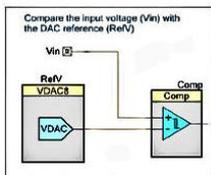
Le projet Template_Media est celui de démonstration livré avec les Macromédias PSoC.

! Il y a beaucoup de manipulations de fichiers donc faire une sauvegarde préalable, ne pas modifier les fichiers originaux.

Recopie des fichiers du GUI

↳ Recopier tous les fichiers *.c et *.h dans le répertoire du projet PSoC Creator.





Puis les intégrer au projet PSoC Creator dans Header Files pour les .h et dans Source Files pour les .c :

A ce stade le projet n'est pas compilable, il faut faire des modifications dans les fichiers insérés.

Préparation des fichiers

image_driver.c

↳ In `_driver.c` file, the following include directives need to be added.

```
#include <device.h>
#include "TFT.h"
#include "JPEG.h"
#include "Resist_Touch_Driver.h"
```

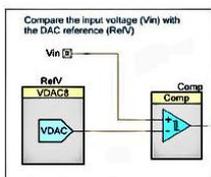
↳ Delete the following lines:

```
- #include "built_in"
- All pin definition lines starting with sbit - // TFT module connections
- const ADC_THRESHOLD = 750;
- char save_bled, save_bled_direction;
```

↳ Update the `Xcoord`, `Y`, variable declaration to `uint16`:

```
uint16 Xcoord, Ycoord;
```

↳ Remove `Init_MCU` and `Init_ADC` function definitions and corresponding calls in `void Start_TP()`



↳ supprimer la fonction InitializeTouchPanel elle fait double emploi avec celle qui est dans le main.c

```
static void InitializeTouchPanel() {
    TFT_Init_ILI9341_8bit(320, 240);
    TFT_BLED_Write(1);
    TP_TFT_Init(320, 240, 0, 1);
    TP_TFT_Set_ADC_Threshold(ADC_THRESHOLD);

    PenDown = 0;
    PressedObject = 0;
    PressedObjectType = -1;
}
```

↳ Supprimer la fonction Calibrate()

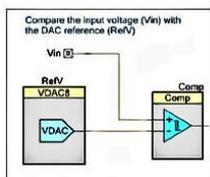
elle est déjà présente dans le main.c de PSoC Creator.

↳ supprimer les appels calibrate() et InitializeTouchPanel() de la fonction Start_TP()

```
void Start_TP() {
    CyDelay(1000);
    TFT_Fill_Screen(0);
    //TFT_Fill_Screen(0);
    InitializeObjects();
    display_width = Screen1.Width;
    display_height = Screen1.Height;
    DrawScreen(&Screen1);
}
```

↳ Add TFT_BLED_Write(1); call in InitializeTouchPanel() function, just below the display initialization call:

```
static void InitializeTouchPanel() {
    TP_TFT_Set_Default_Mode();
    TP_TFT_Init(320, 240);
    TFT_Set_Default_Mode();
    TFT_Init_ILI9341_8bit(320, 240);
    TFT_BLED_Write(1);
    PenDown = 0;
    PressedObject = 0;
    PressedObjectType = -1;
}
```



↳ In DrawScreen function replace first if/else-branch with following if/else-branch:

```
// --- Start of new if/else
if ((display_width != CurrentScreen->Width) || (display_height != CurrentScreen->Height)) {
    TFT_BLED_Write(0);
    TFT_Init_ILI9341_8bit(CurrentScreen->Width, CurrentScreen->Height);
    TP_TFT_Init(CurrentScreen->Width, CurrentScreen->Height);
    TFT_Fill_Screen(CurrentScreen->Color);
    display_width = CurrentScreen->Width;
    display_height = CurrentScreen->Height;
    TFT_BLED_Write(1);
}
else
    TFT_Fill_Screen(CurrentScreen->Color);
// --- End of new if/else
```

Préparation des fichiers

image_events_code.c

↳ In project_events_code.c
add #include "TFT.h" include directive.

Pour tous les fichiers

↳ For all files use find and replace to change:

- "const code" in "const"
- "Delay_ms" in "CyDelay"

↳ Add casting (uint8*) for following function calls:

- u TFT_Write_Text call - in front of text variable argument.

Example: `TFT_Write_Text((uint8*)"This example shows some basic features of Visual TFT software.", 10, 105);`

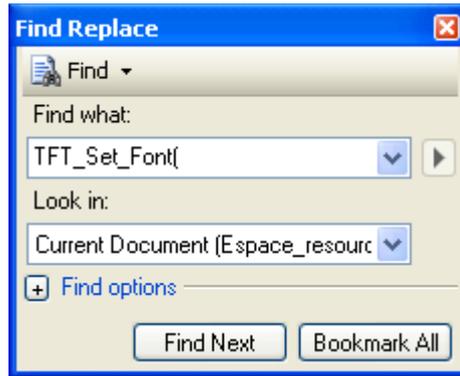
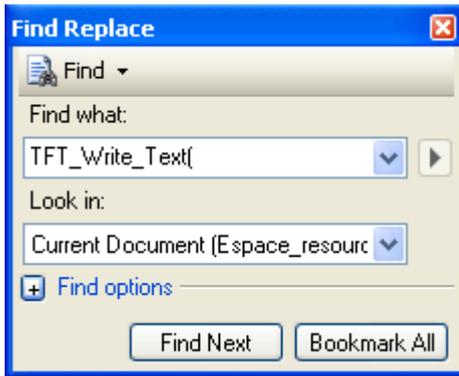
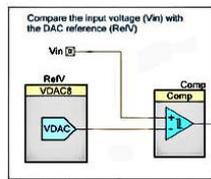
- TFT_Set_Font in front of font-name argument.

Example:

```
TFT_Set_Font((uint8*)&Tahoma11x13_Regular, CL_AQUA, FO_HORIZONTAL);
```

```
TFT_Set_Font((uint8*)&Around_button->FontName, Around_button->Font_Color, FO_HORIZONTAL);
```

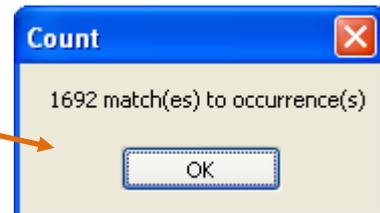
↳ également pour `TFT_Write_Text_Return_Pos(Abutton->Caption, Abutton->Left, Abutton->Top);`
`TFT_Write_Text_Return_Pos((uint8*)Abutton->Caption, Abutton->Left, Abutton->Top);`



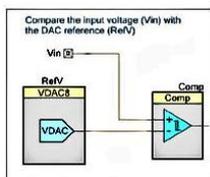
Pour les fichiers

`image_resource.c` `image_resource.h`

- Define size of resource arrays both in `project_resource.h` and `_resource.c` files.
- The easiest way to quickly count large resource arrays is to count occurrence of "0x".
- This can be done with notepad++ or similar text editor.



```
const char Tahoma11x13_Regular[] = (
    0x00,
    0x00,
    0x20, 0x00,
    0x7F, 0x00,
    0x0D,
    0x10,
    0x03, 0x88, 0x01, 0x00,
    0x02, 0x95, 0x01, 0x00,
    0x03, 0xA2, 0x01, 0x00,
    0x07, 0xAF, 0x01, 0x00,
    0x05, 0xBC, 0x01, 0x00,
    0x0A, 0xC9, 0x01, 0x00,
    0x07, 0xE3, 0x01, 0x00,
    0x01, 0xF0, 0x01, 0x00,
    0x03, 0xFD, 0x01, 0x00,
```

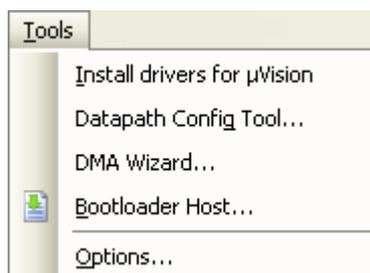



4 Compilation du projet

Compiler le projet il ne doit plus y avoir une erreur.

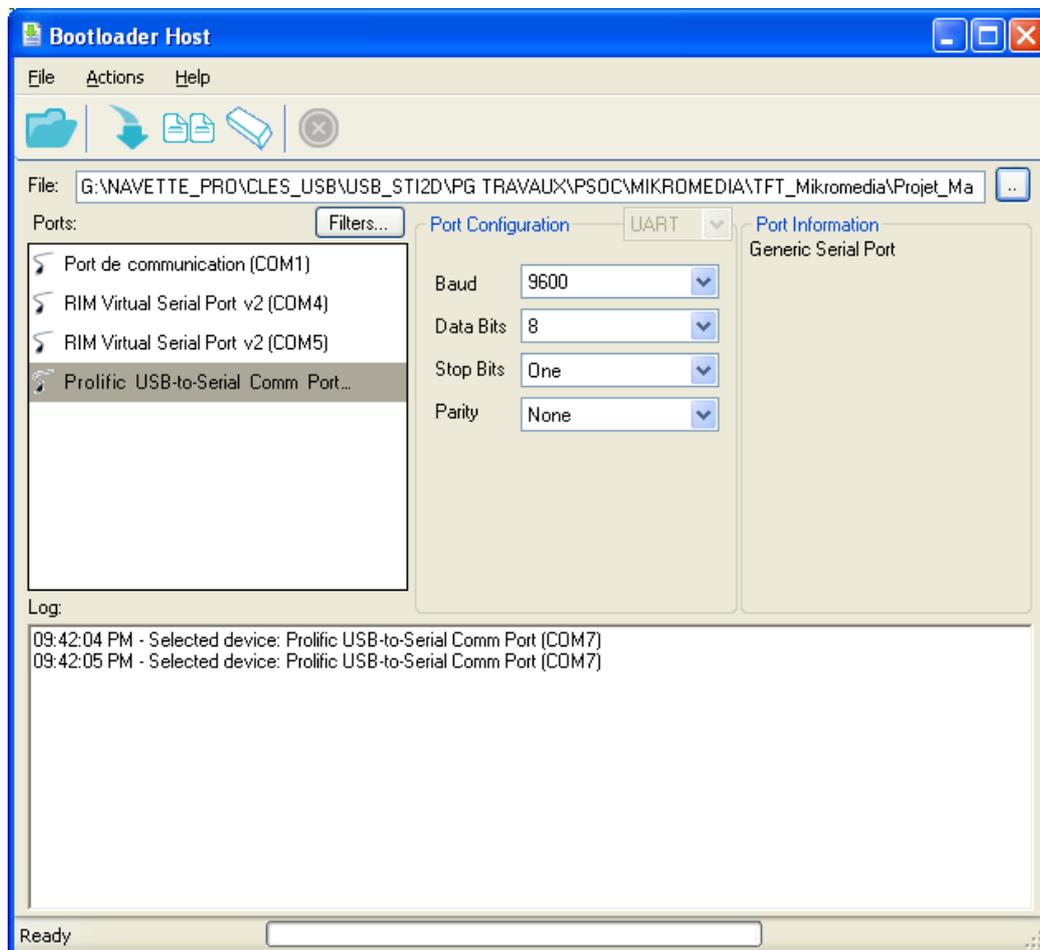
5 Programmation avec le bootloading

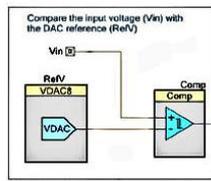
Lancer le programme de bootloader

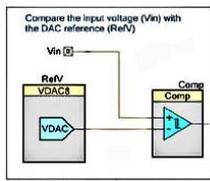


Bien vérifier que le fichier de programmation est le bon :

..\Template_Media.cydsn\CortexM3\ARM_GCC_473\Debug\Template_Media.cyacd



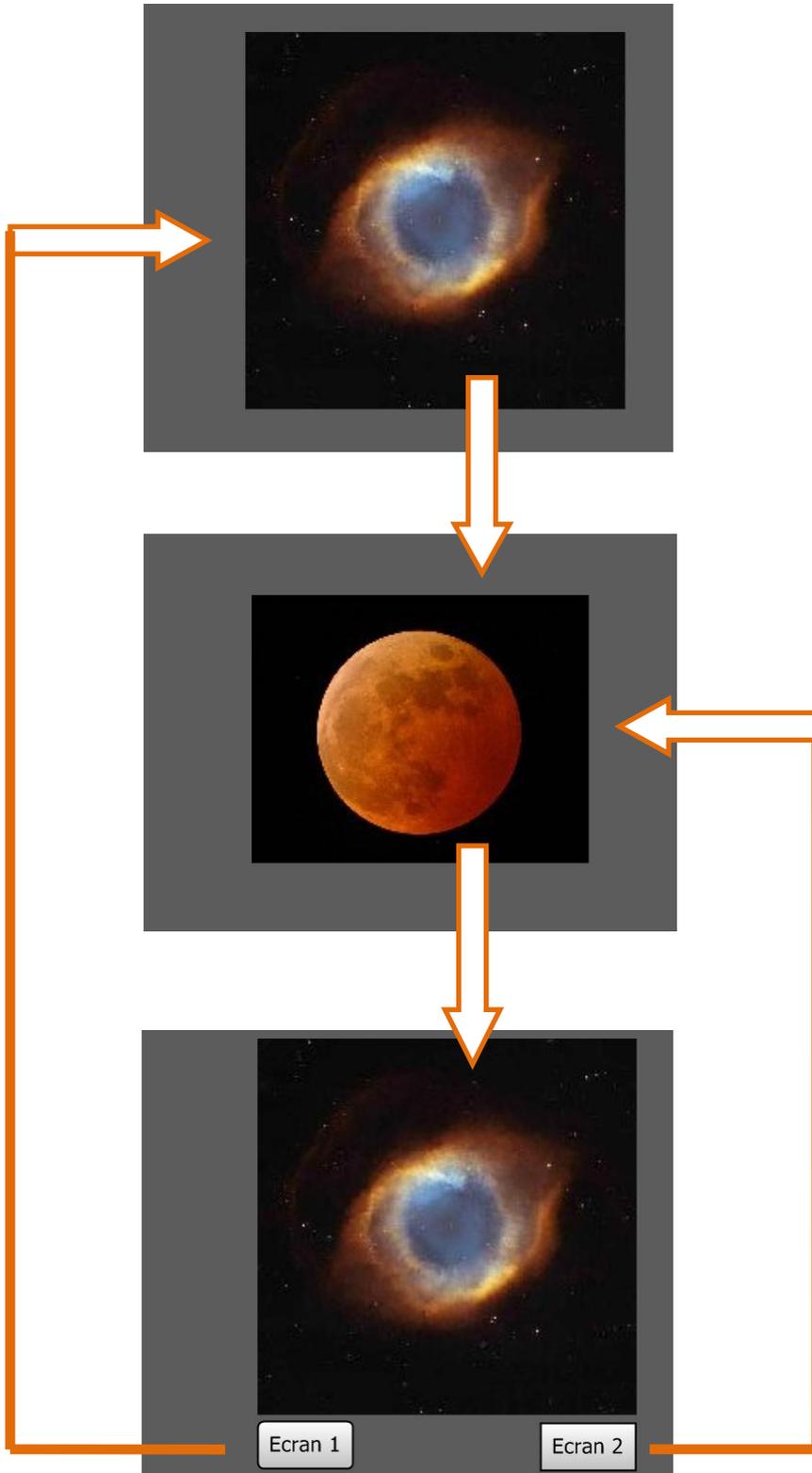
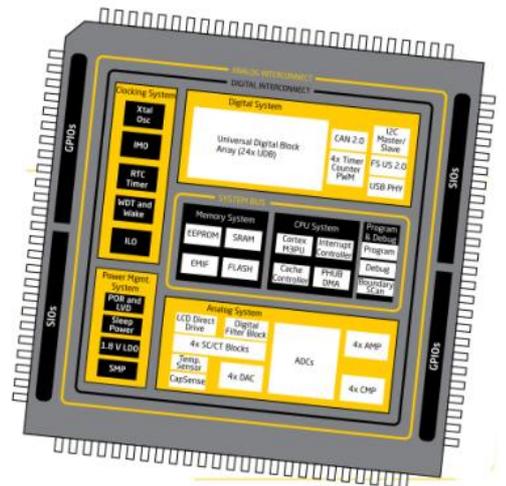


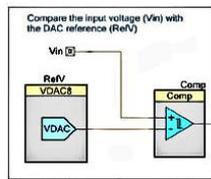


Le projet final TFT PSoC5LP



3.5.0
VISUAL TFT
 always one step ahead





 Indique un document ressource

 Retour au sommaire

 Retour à la page courante