



# ***Développez vos systèmes embarqués sur SoC FPGA***

***Comment embarquer Linux et développer en VHDL vos applicatifs dédiés au traitement d'image sans pénaliser le CPU.***

**Jean-Marie CODOL**  
Développeur

Submarine Open Technologies  
Montpellier

Avril 2022



# ***Développez vos systèmes embarqués sur SoC FPGA***

***Comment embarquer Linux et développer en VHDL vos applicatifs dédiés au traitement d'image sans pénaliser le CPU.***

***Partie 4***

# Plan

## Plan

JOUR 1

§001 Faire communiquer Linux avec un FPGA sur le SoC intel DE10-nano

- distribution fournie par Altera
- créer une image SD (en partie sur un serveur distant)
- **compilation croisée (eclipse CDT embedded)**
- activer les bridges HPS <-> FPGA par un device tree
- utiliser le bridge HPS2FPGA

J 2

§002 Partager une zone de RAM entre Linux et un FPGA sur le SoC intel DE10-nano

- projet de lecture/écriture en RAM sur FPGA
- projet de lecture/écriture en RAM sur CPU

J 3

§003 Cas pratique avec seuillage d'image

- projet openCV
- échange sur mémoire RAM

# §001 Faire communiquer Linux avec un FPGA

## Compilation croisée (eclipse CDT embedded)

Logiciels que l'on va installer en local :

```
Quartus 20.1  
SoC EDS 20.1 (déjà fait précédemment)  
Eclipse CDT embedded 2021-09 (4.21.0)
```

Suivre les instructions de :

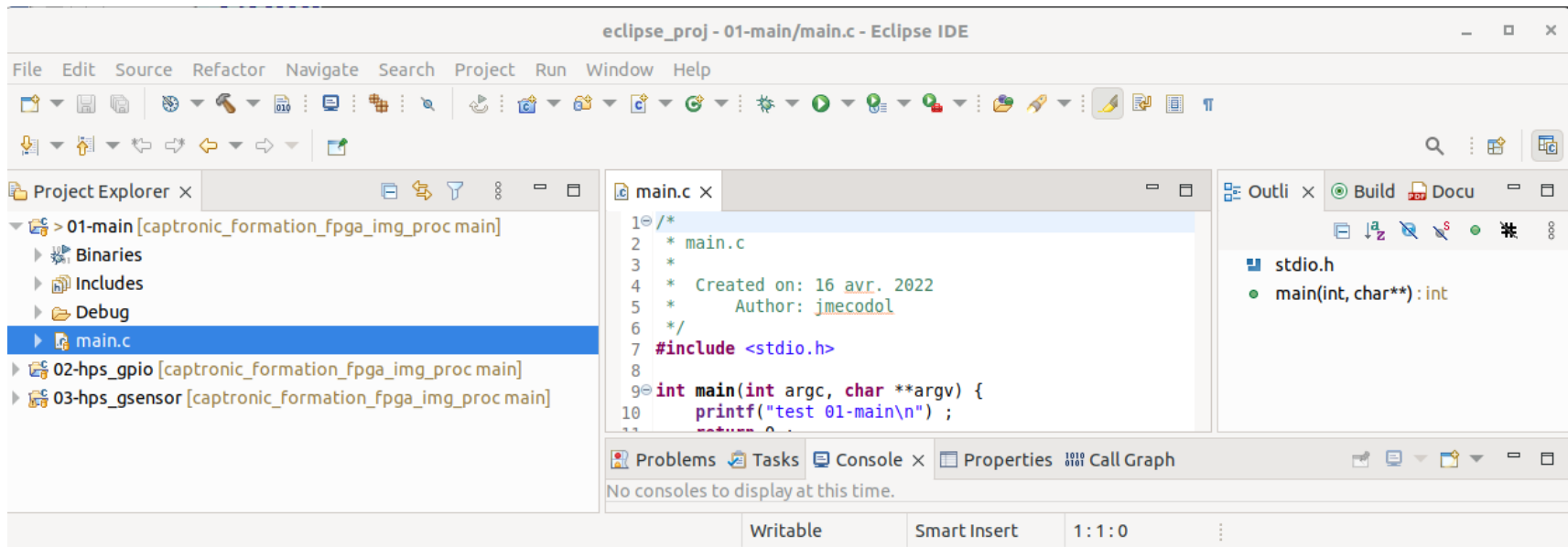
```
<captronic_formation_fpga_img_proc>/scripts/5-cross-compiler.sh
```

# §001 Faire communiquer Linux avec un FPGA

## Compilation croisée (eclipse CDT embedded)

### Eclipse CDT embedded :

```
cd  
~/intelFPGA/20.1/embedded/embedded_command_shell.sh  
DEWD=$PWD/de10nano-wd  
cd $DEWD/eclipse/  
./eclipse/eclipse  
Espace de travail --> <captronic_formation_fpga_img_proc>/eclipse_proj2  
Créer et compiler pour arm (voir page suivante)
```



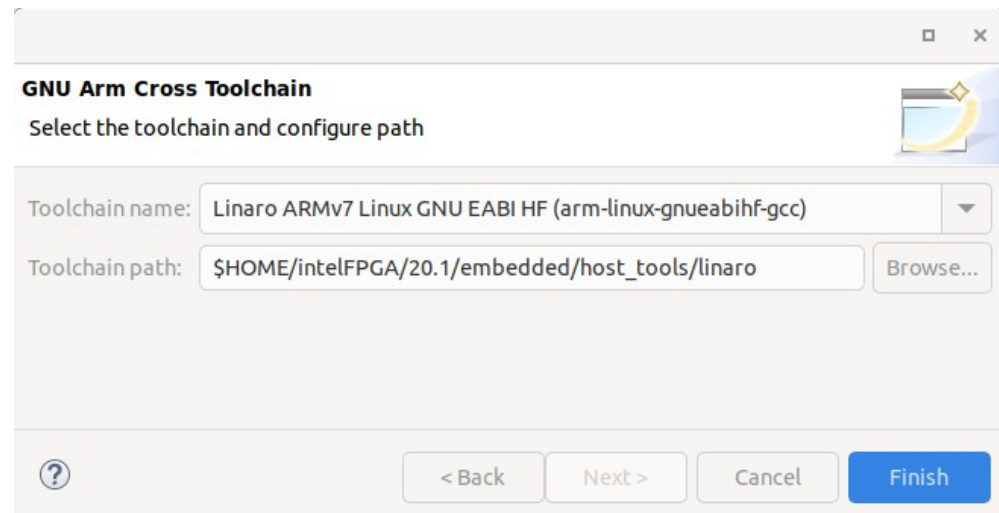
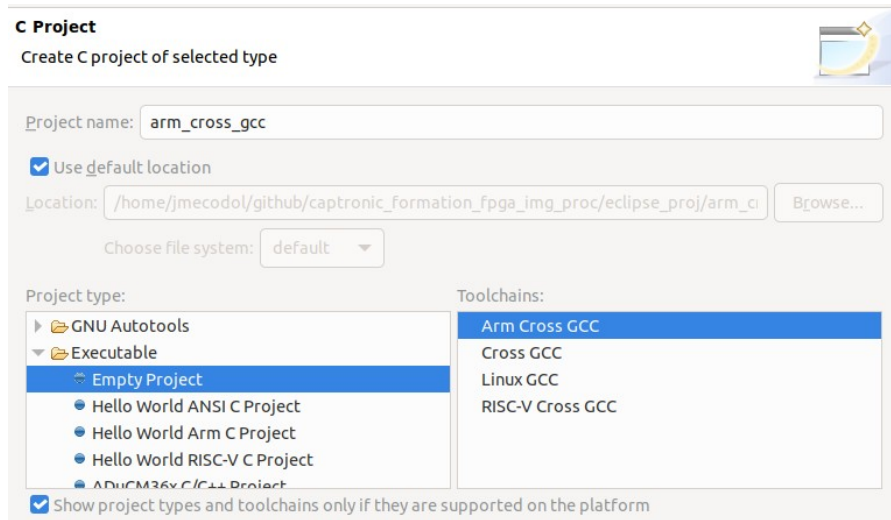


# §001 Faire communiquer Linux avec un FPGA

## Compilation croisée (eclipse CDT embedded)

### Installer Eclipse CDT embedded :

Fichier -> nouveau projet -> C/C++ / C managed build /  
Project name : arm\_cross\_gcc  
Executable / Empty Project / Arm Cross GCC  
Toolchain name : Linaro ARMv7 Linux GNU EABI HF (**arm-linux-gnueabi-hf-gcc**)  
Toolchain path : \$HOME/intelFPGA/20.1/embedded/host\_tools/linaro



# §001 Faire communiquer Linux avec un FPGA

## Compilation croisée (eclipse CDT embedded)

### Installer Eclipse CDT embedded :

Propriétés du projet :

C/C++ Build

Settings

Target Processor

**cortex-a9**

Gnu Arm Cross Gcc Compiler

Preprocessor >> -D >> soc\_cv\_av

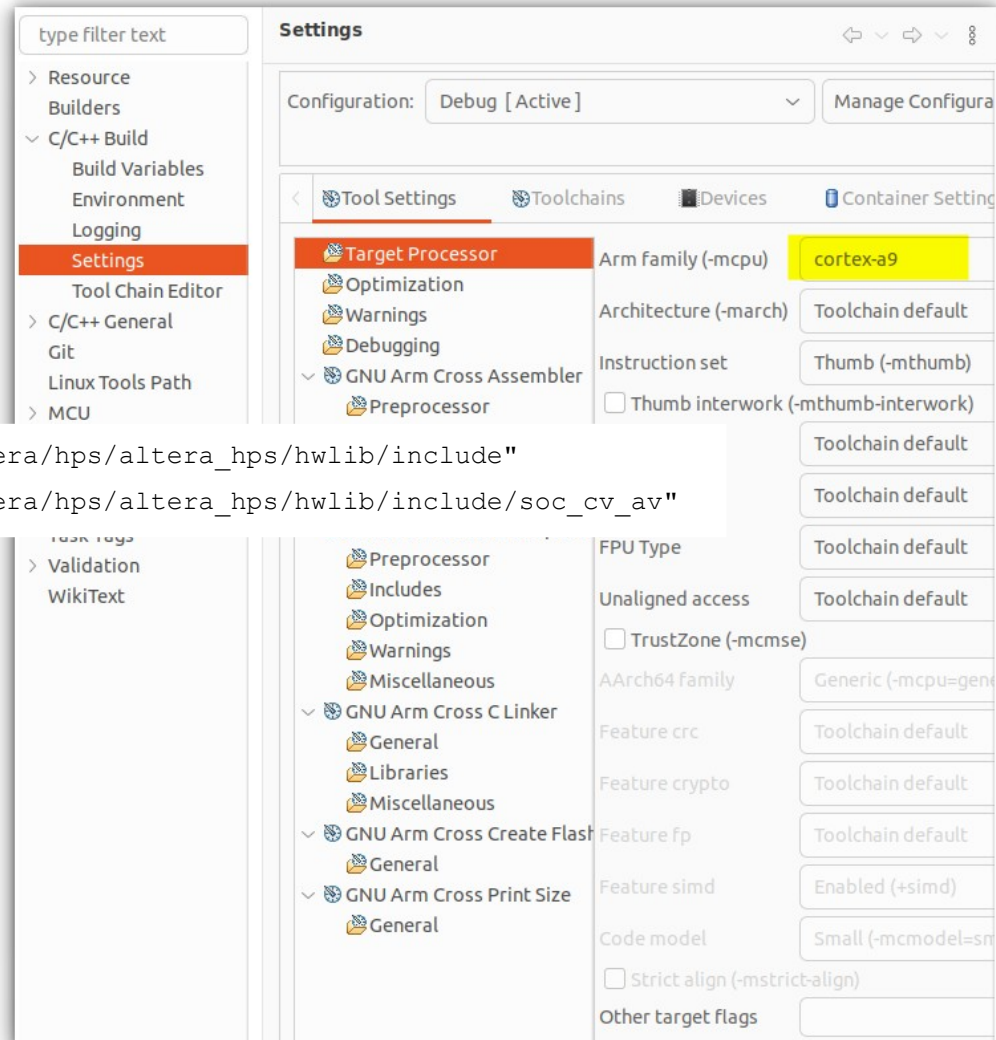
Includes >>

-I >> "\${HOME}/intelFPGA/20.1/embedded/ip/altera/hps/altera\_hps/hwlib/include"

-I >> "\${HOME}/intelFPGA/20.1/embedded/ip/altera/hps/altera\_hps/hwlib/include/soc\_cv\_av"

Cette configuration trouve son origine dans les fichiers Makefile qui se trouvent dans les exemples du CDROM fournis avec le SoC

```
1 #
2 TARGET = hps_gpio
3
4 #
5 ALT_DEVICE_FAMILY ?= soc_cv_av
6
7 HWLIBS_ROOT = $(SOCEDS_DEST_ROOT)/ip/altera/hps/altera_hps/hwlib
8
9 CFLAGS = -g -Wall -Werror -IS $(HWLIBS_ROOT)/include -IS $(HWLIBS_ROOT)/include/$(ALT_DEVICE_FAMILY) -D$(ALT_DEVICE_FAMILY)
10 LDFLAGS = -g -Wall -Werror
11
12 CROSS_COMPILE = arm-linux-gnueabihf-
13 CC = $(CROSS_COMPILE)gcc
14 LD = $(CROSS_COMPILE)gcc
15 ARCH = arm
16
17 .PHONY: build
18 build: $(TARGET)
19
20 $(TARGET): main.o
21 $(LD) $(LDFLAGS) $^ -o $@
22
23 %.o : %.c
24 $(CC) $(CFLAGS) -c $< -o $@
```



## §001 Faire communiquer Linux avec un FPGA

### Compilation croisée (eclipse CDT embedded)

Créer un projet : 01-test avec un hello World

L'exécuter sur cible DE10-nano (copier avec scp et lancer).

Créer un projet : 02-hps\_gpio avec le contenu du CDROM :

demos/SoC/hps\_gpio/main.c

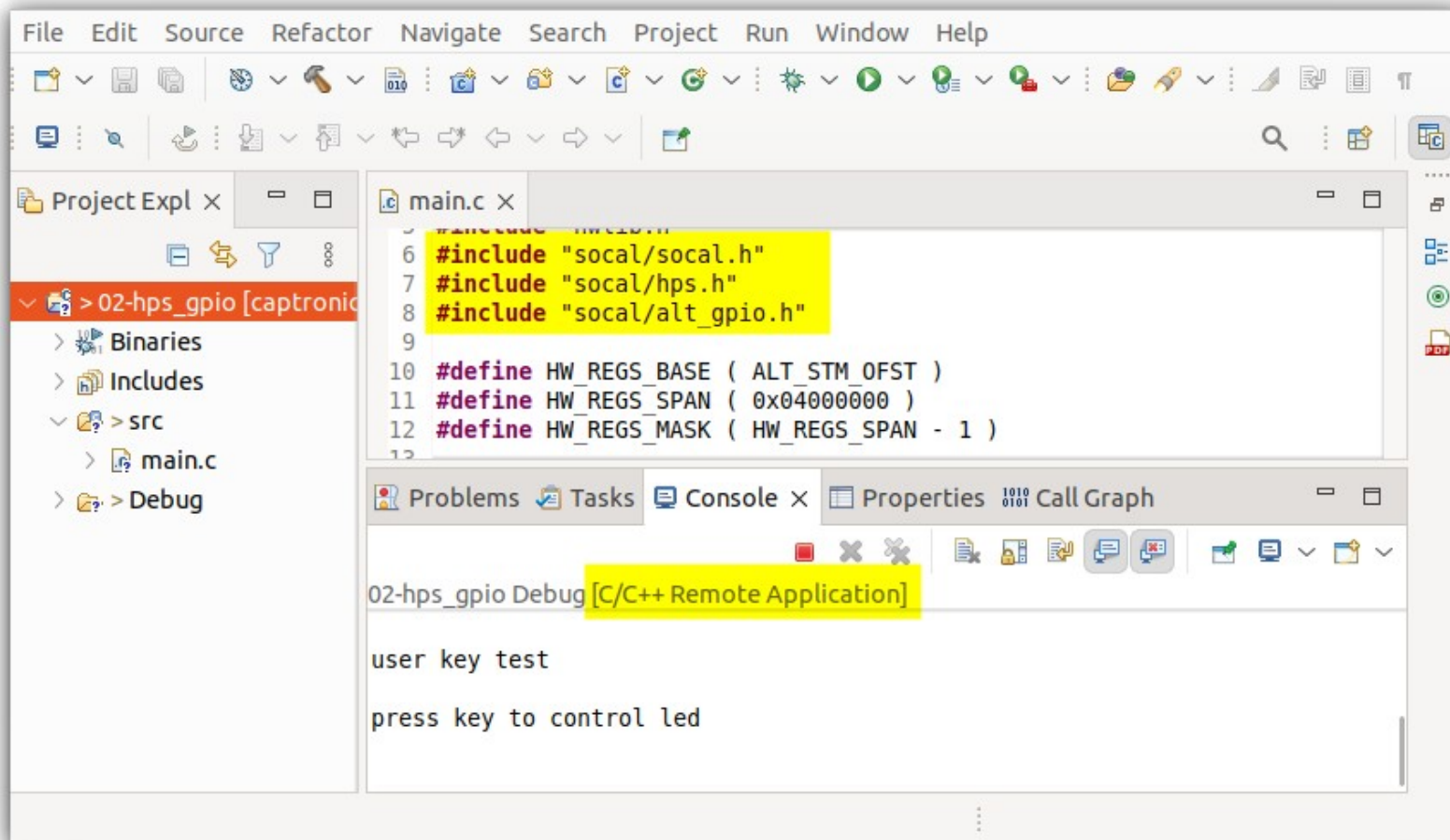
L'exécuter sur cible DE10-nano

Utiliser l'execution distante du logiciel Eclipse.



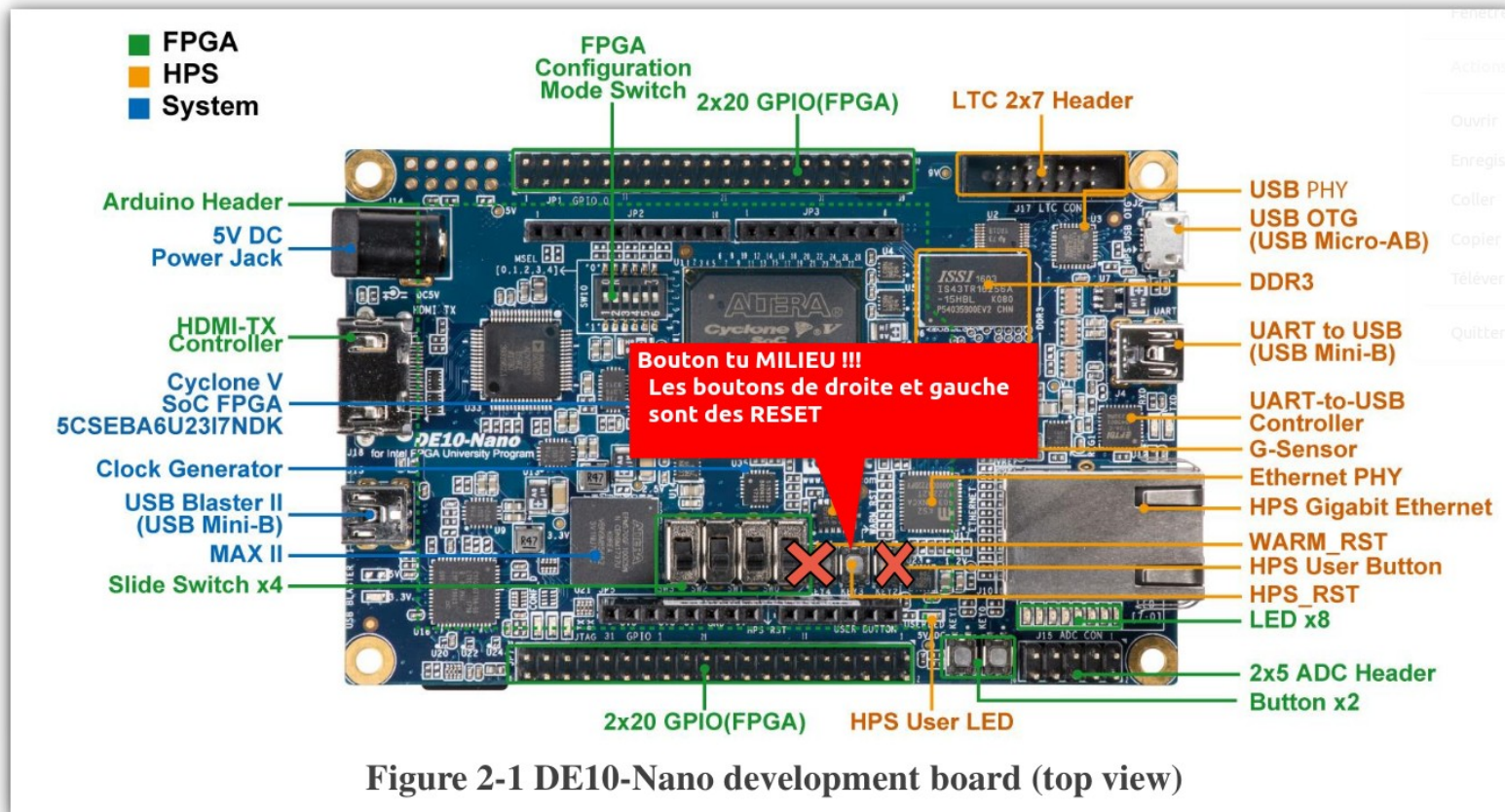
# §001 Faire communiquer Linux avec un FPGA

## Compilation croisée (eclipse CDT embedded)



# §001 Faire communiquer Linux avec un FPGA

Compilation croisée (eclipse CDT embedded)



# Plan

## Plan

JOUR 1

§001 Faire communiquer Linux avec un FPGA sur le SoC intel DE10-nano

- distribution fournie par Altera
- créer une image SD (en partie sur un serveur distant)
- compilation croisée (eclipse CDT embedded)
- activer les bridges HPS <-> FPGA par un device tree
- **utiliser le bridge HPS2FPGA**

J 2

§002 Partager une zone de RAM entre Linux et un FPGA sur le SoC intel DE10-nano

- projet de lecture/écriture en RAM sur FPGA
- projet de lecture/écriture en RAM sur CPU

J 3

§003 Cas pratique avec seuillage d'image

- projet openCV
- échange sur mémoire RAM