Intel DE-10 Standard: HPS SoC Interfaces and Process Methodology

(A session under DE-10 SoC Track)

Dheemanth R Joshi

Research Assistant

Centre For Innovation and Entrepreneurship, PES University

After this session you will:

- Have a working understanding of the DE-10 board.
- Know some of the important interfaces and peripherals installed on the board.
- Have a brief idea about the Hard Processor System (HPS) installed on the board.
- Be able to run simple procedures on the board.

Pre-requisites

- Basic working understanding on Linux operating system.
- C-programming language basics.
- Please download the user manual from: https://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=165&No=1081&PartNo=4#contents

Today's session

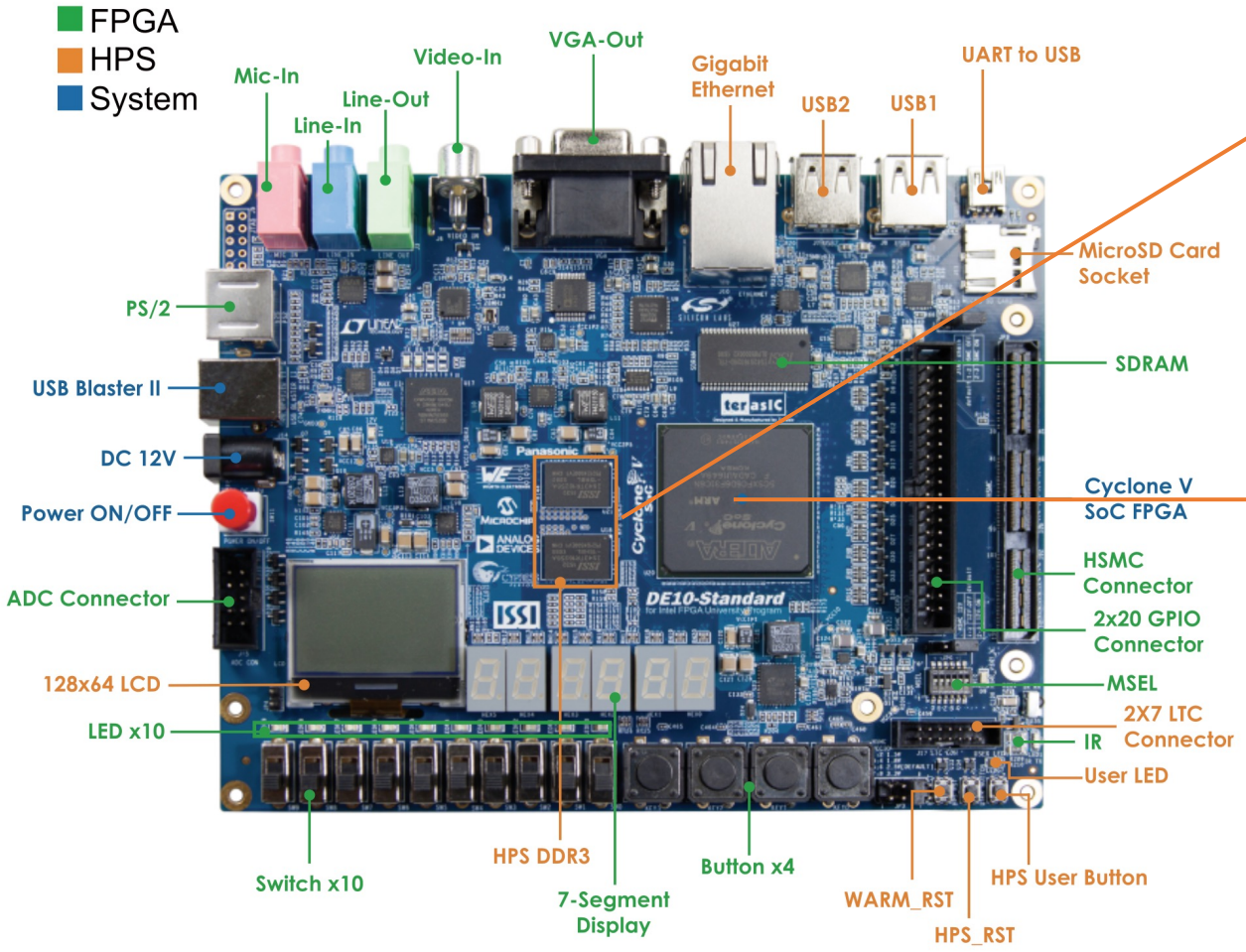
Interfaces/peripherals connecting HPS

- Basic architecture of the board
- MSEL setting
- Systems on the board
- Memory components
- Communication interfaces on chip

Running C procedures on chip

- Basic C program files structure
- Creation of executable file using EDS tool
- Transmission of the executable file to the chip
- Process on chip

Basic System Architecture



Hard Processor System (HPS)

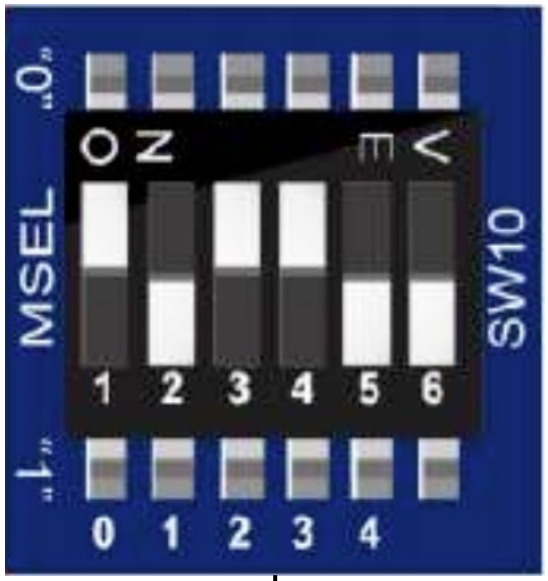
- Dual Core ARM Cortex-A9 MPCore processor
- Normal ARM architecture
- C procedures run here
- Various peripherals on chip to enable non-native system communication

FPGA

- Altera Cyclone V SoC FPGA
- Verilog synthesized netlist run here
- Like HPS SoC, there are various interfaces to communicate with external systems
- (user reconfigurable)

System peripherals have been marked in the figure
Important ones will be discussed separately

Setting the board configuration: MSEL pins



- MSEL configuration for AS mode (“10010”)
- Configured from EPCS

FPGA can be configured via 2 modes:

1) Active Serial (AS) mode: FPGA configured from EPCS (default)
2) FPPx32 mode: FPGA configured from HPS software : (through Linux OS)

Configuration state MSEL[4:0] for each mode is:

1) AS mode: “10010”
2) FPPx32 mode: “01010”

Please Note:
The following configurations must be strictly followed which if failed can lead to catastrophic failure of the chip

Board reset elements: HPS_RESET_N

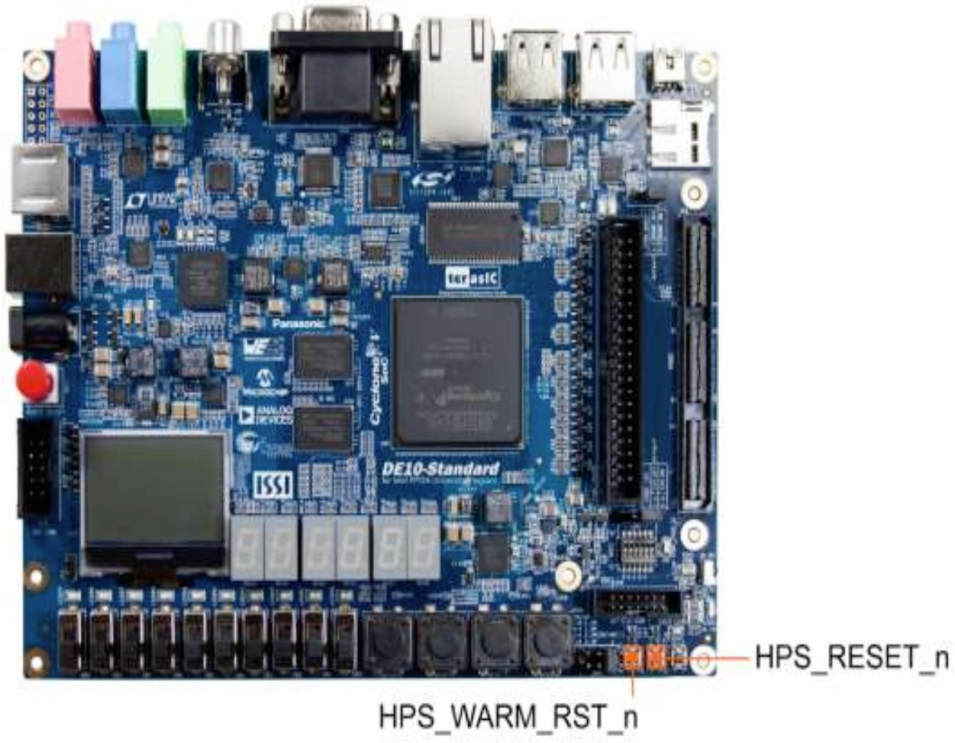


Figure indicating the location of HPS Reset Buttons on the board

HPS Reset Buttons

HPS_RESET_N
- Cold reset to the HPS, Ethernet PHY and USB host device
- Used in functional mode

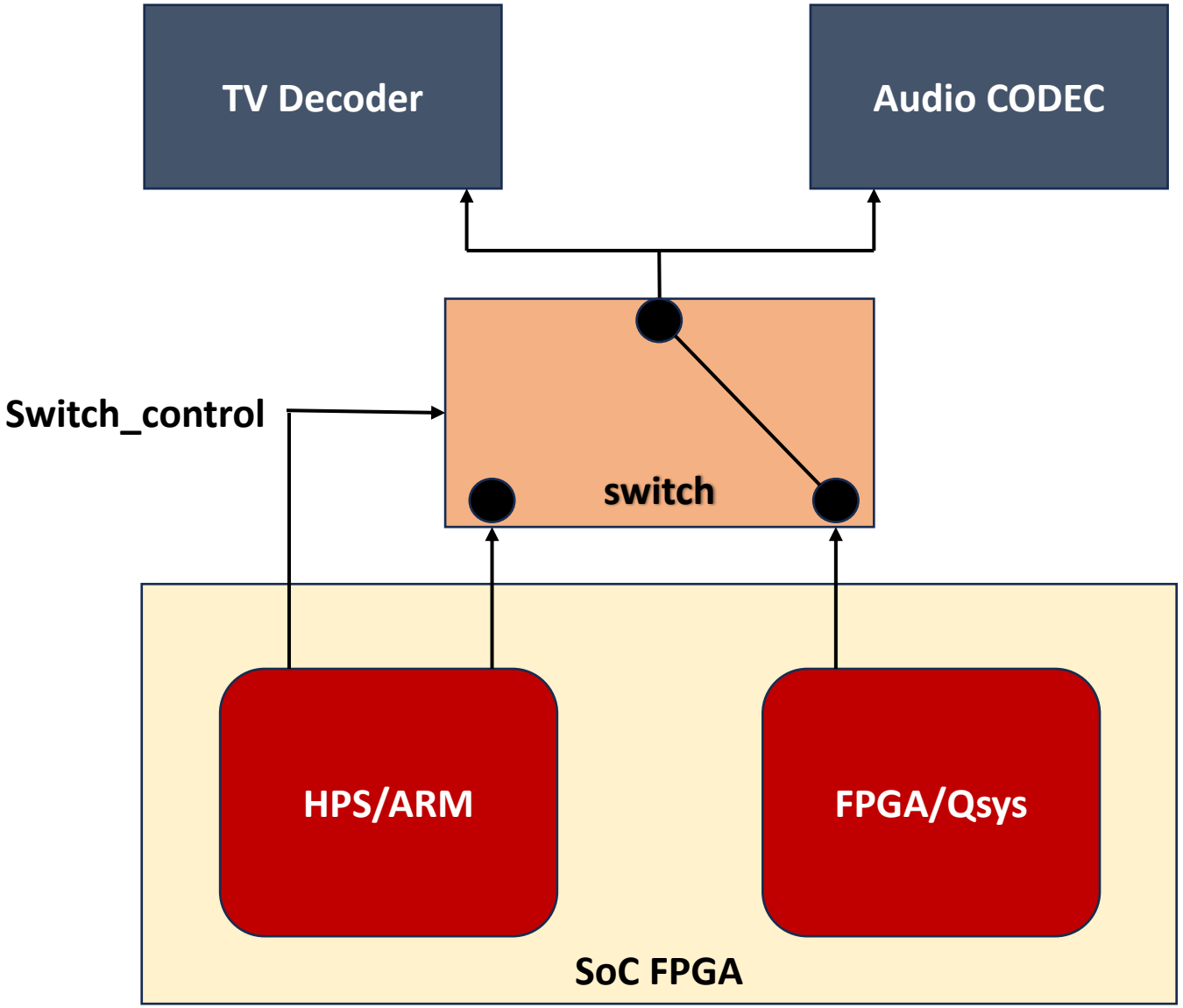
HPS_WARM_RST_N
- Warm reset to the HPS block
- Used in test/ debug mode

- Focus only on “Functional mode”
- Section on clock circuitry has been skipped

Peripherals Connected to FPGA

- **User Push Buttons, Switches and LED's**
- 7-segment display
- 2x20 GPIO Expansion Header
- HSMC connector
- **24-bit Audio CODEC**
- **VGA Output; TV decoder**
- IR receiver/ emitter
- SDRAM memory
- A2D converter and 2x5 header

Resource sharing between SoC and FPGA: I2C multiplexer



The board implements an I2C multiplexer for HPS to access the I2C bus originally owned by FPGA.

Through this control mechanism, HPS can access Audio CODEC and TV decoder peripherals independently.

Peripherals Connected to HPS

- **User Push Buttons and LEDs:** Similar to the FPGA, the HPS also has its set of switches, buttons, LEDs, and other interfaces connected exclusively. Users can control these interfaces to monitor the status of HPS.

- **Gigabit Ethernet:** RJ-45 interface for external communication has been provided. The HPS is connected to the RJ-45 with a PHY chip integrated with Ethernet transceiver which implements MAC protocol.

- **DDR3 Memory:** The board supports 1GB of DDR3 SDRAM comprising of two x16 bit DDR3 devices on HPS side. The signals are connected to the dedicated Hard Memory Controller for HPS I/O banks and the target speed is 400MHz.

- **UART to USB:** The board has one UART interface connected for communication with the HPS. This interface doesn't support HW flow control signals.

Peripherals Connected to HPS

- **Micro SD card socket:** The board supports Micro SD card interface with x4 data lines. It serves not only an external storage for the HPS, but also an alternative boot option for DE10-Standard board. (Operating system **(LXDE)** is booted through this interface)

- **2-Port USB host:** useful to connect monitor, camera, mouse etc..

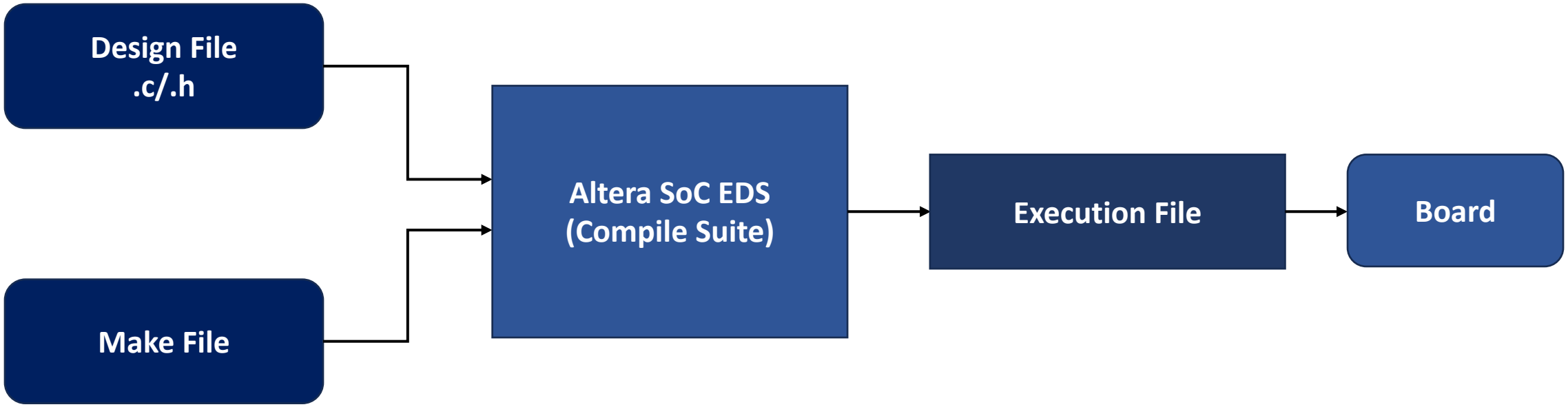
- **Accelerometer (G-sensor):** This G-sensor is a small, thin, ultralow power assumption 3-axis accelerometer with high-resolution measurement.

- **LTC Connector:** The board has a 14-pin header, which is originally used to communicate with various daughter cards from Linear Technology.

- **128x64 LCD:** The board equips an LCD Module with 128x64 dots for display capabilities. The LCD module uses serial peripheral interface to connect with the HPS.

Executing Procedures On Chip: EDS

- **Software Development Flow:**



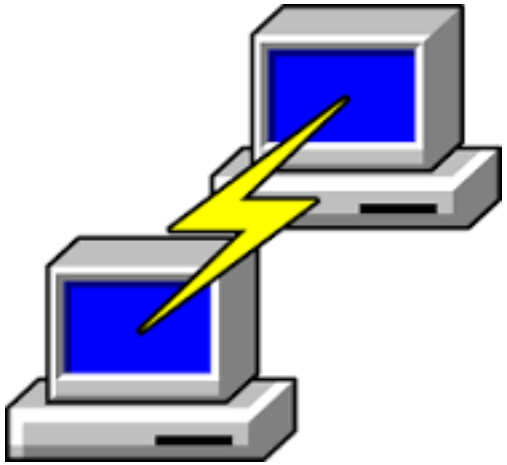
Note: your PC will perform these steps until execution file is created, later the file is transferred to the board via Ethernet.

Executing Procedures On Chip: EDS

Software/Tools/API for on chip execution

Altera SoC EDS (Embedded design file suite)

- Can be downloaded from intel website
- Set of files to transform design and makefile to executable file
- Done through built in GCC compiler which is called on virtual linux terminal executed as a batch file



PuTTY Software and Virutal Com Drivernano:

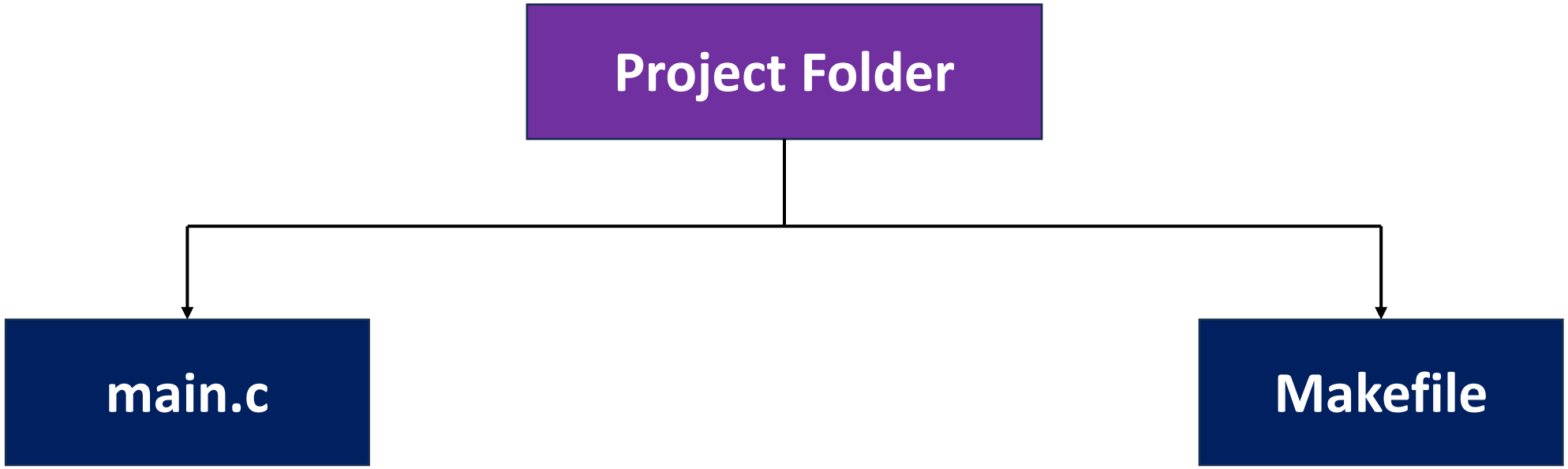
- PuTTY is a free open source terminal emulator
- Supports various Communication protocols such as SCP, SSH, Telnet etc…
- Used for communicating between PC and the board

Executing Procedures On Chip

- **Step-1: Create a project folder (On PC)**
- **A project usually includes the design files .c/.h and a make file. These files are generally stored under the same folder.**
- **Developer can create a folder under the installed Altera SoC EDS installation folder.**

Executing Procedures On Chip

- **Step-2 : Create a Design file and a make file (On PC)**



Executing Procedures On Chip

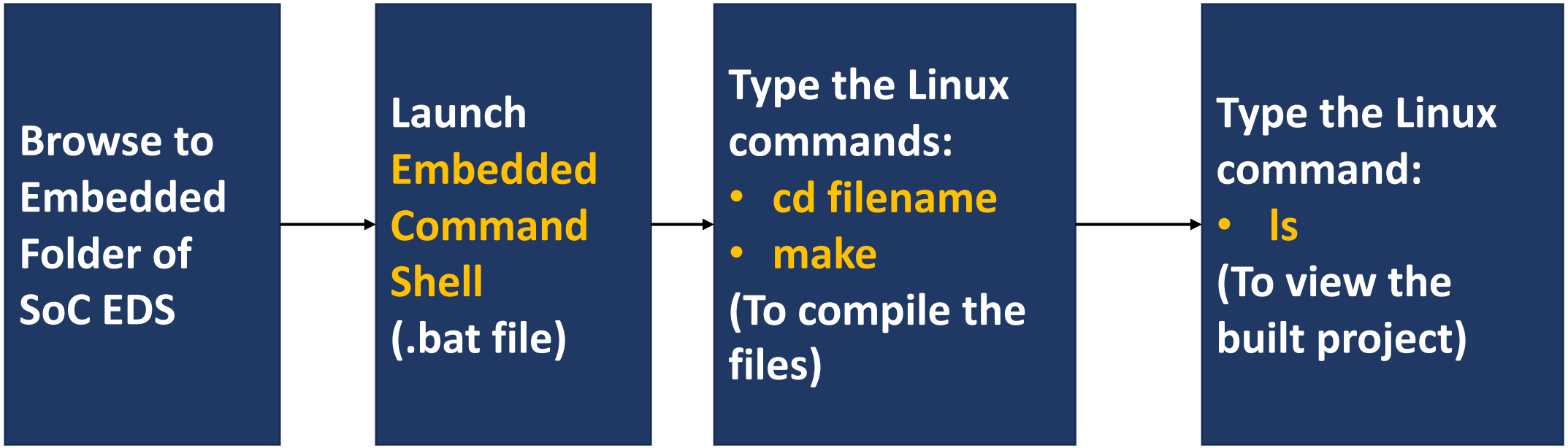
```
#include <stdio.h>

int main(int argc, char **argv) {

        printf("Hello World!\r\n");

        return( 0 );
```

Figure on top: main.c
Figure on right: Makefile

```
#
TARGET = my_first_hps

ALT_DEVICE_FAMILY ?= soc_cv_av
SOCEDS_ROOT ?= $(SOCEDS_DEST_ROOT)
HWLIBS_ROOT = $(SOCEDS_ROOT)/ip/altera/hps/altera_hps/hwlib
CROSS_COMPILE = arm-linux-gnueabihf-
CFLAGS = -g -Wall -D$(ALT_DEVICE_FAMILY) \
            -I$(HWLIBS_ROOT)/include/$(ALT_DEVICE_FAMILY) \
            -I$(HWLIBS_ROOT)/include/
LDFLAGS = -g -Wall
CC = $(CROSS_COMPILE)gcc
ARCH = arm

build: $(TARGET)

$(TARGET): main.o
        $(CC) $(LDFLAGS) $^ -o $@

%.o : %.c
        $(CC) $(CFLAGS) -c $< -o $@

.PHONY: clean
clean:
        rm -f $(TARGET) *.a *.o *~
```

Executing Procedures On Chip

- **Step-3: Compile the project (On PC)**



Executing Procedures On Chip

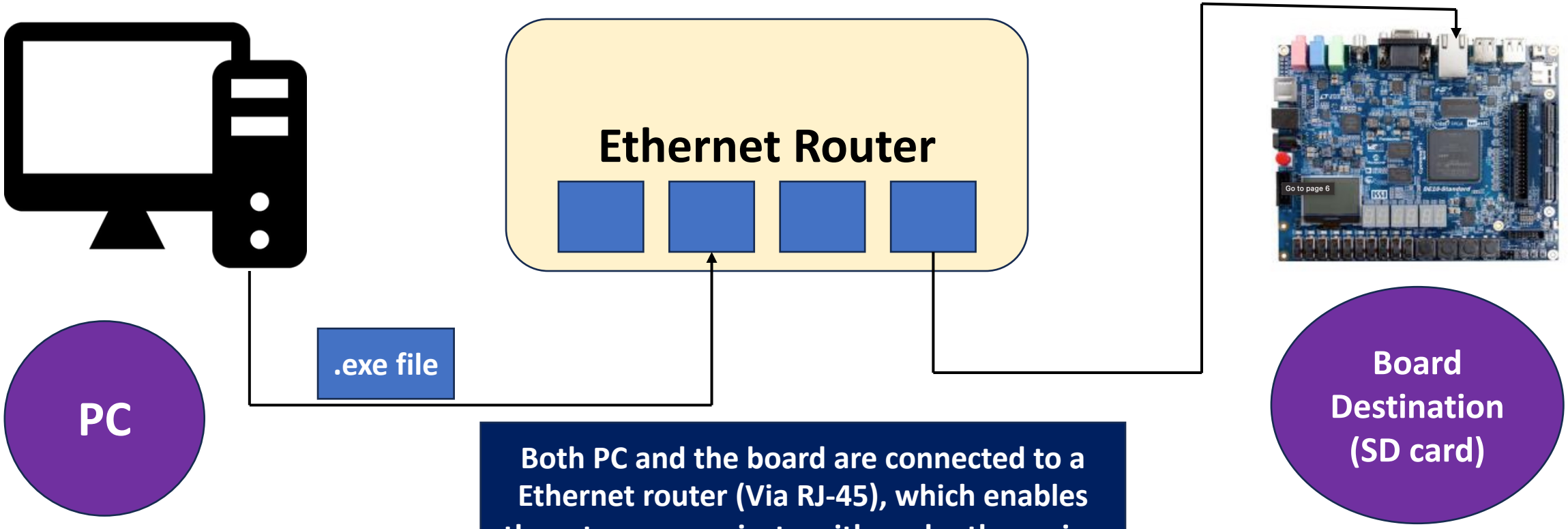
Figure indicating the Linux : Altera Embedded Command Shell on the PC

Executing Procedures On Chip: File transfer to chip

- **Once the execution file is ready through EDS, it must be transferred to the board (bootable SD card) for execution.**
- **This is achieved by SCP communication protocol handled by PuTTY API.**
- **Communication is done through ethernet port.**

Executing Procedures On Chip: File transfer to chip

Setup for file transfer



Executing Procedures On Chip: File transfer to chip

- **Step-4: Configure the network on Board (On chip)**

Login as Root user

Type the Command:

- **udhcpc**

To query an ip address from the DHCP server,
Type the command

- **Ipconfig**

To view the assigned ip address to the board

```
root@DE10_STANDARD:~# udhcpc
root@DE10_STANDARD:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 9e:77:86:70:9b:0f
          inet addr:192.168.21.134  Bcast:192.168.21.255  Mask:255.255.255.0
          inet6 addr: fe80::8bf5:80f8:866f:a9de/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:11 errors:0 dropped:0 overruns:0 frame:0
          TX packets:13 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2166 (2.1 KB)  TX bytes:2334 (2.3 KB)
          Interrupt:39 Base address:0x8000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:160 errors:0 dropped:0 overruns:0 frame:0
          TX packets:160 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:11840 (11.8 KB)  TX bytes:11840 (11.8 KB)

root@DE10_STANDARD:~#
```

LXDE User interface displaying Linux Shell on chip

Executing Procedures On Chip: File transfer to chip

- **Step 5: Connect the PC and the board (On PC)**
- **Assuming ip address= 192.168.21.134**

On Altera SoC command shell, type the command:

- scp file_name@ip_address:/home/root

- This command transfers the executable file to the board's root directory.
- Reply "yes" to the prompt message asking "are you sure you want to continue connecting (yes/no)?"

- you will be asked for a password for the root user, enter the password and continue.
- Default password for the root directory is terasic

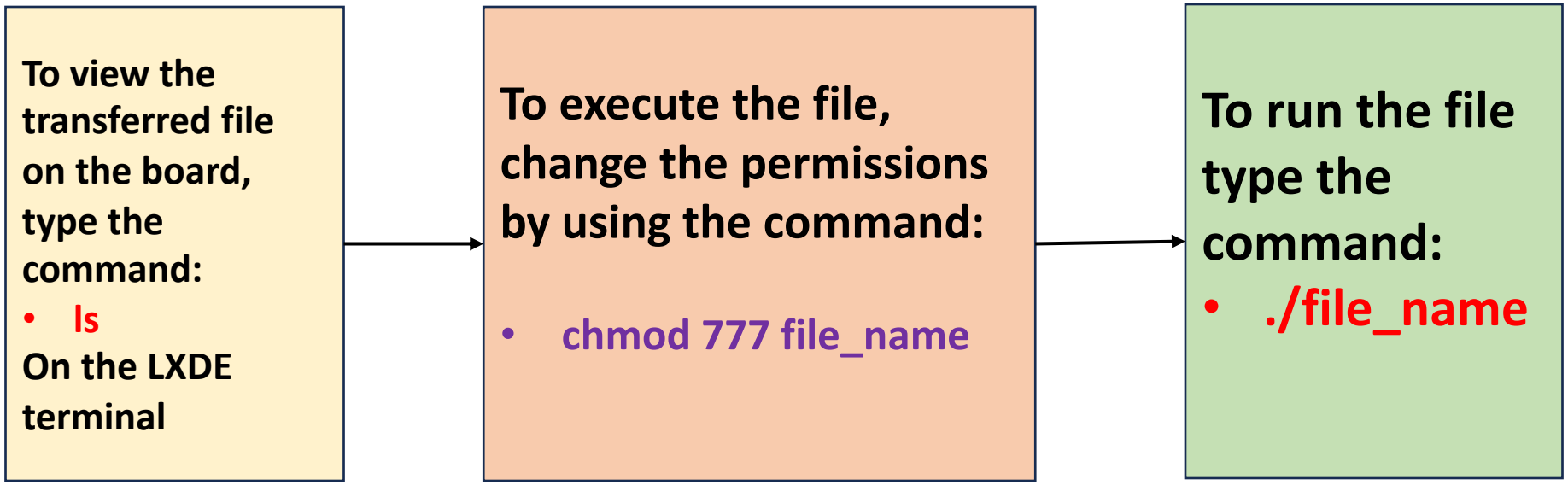
Executing Procedures On Chip: File transfer to chip

```
johnny@johnny-wh-pc /cygdrive/C/intelFPGA/16.1/embedded
$ cd my_first_hps/

johnny@johnny-wh-pc /cygdrive/C/intelFPGA/16.1/embedded/my_first_hps
$ make
arm-linux-gnueabihf-gcc -g -Wall -Werror -IC:/intelFPGA/16.1/embedded/ip/altera/
hps/altera_hps/hwlib/include -IC:/intelFPGA/16.1/embedded/ip/altera/hps/altera_h
ps/hwlib/include/soc_cv_av -Dsoc_cv_av -c main.c -o main.o
arm-linux-gnueabihf-gcc -g -Wall -Werror   main.o -o my_first_hps

johnny@johnny-wh-pc /cygdrive/C/intelFPGA/16.1/embedded/my_first_hps
$ scp my_first_hps root@192.168.21.134:/home/root
Could not create directory '/home/johnny/.ssh'.
The authenticity of host '192.168.21.134 (192.168.21.134)' can't be established.

ECDSA key fingerprint is SHA256:YAVGTDiDJ5Pwbx1o4bkeYZtfVcVyKJIiTZwZVRnIJP4.
Are you sure you want to continue connecting (yes/no)? yes
Failed to add the host to the list of known hosts (/home/johnny/.ssh/known_hosts
).
root@192.168.21.134's password:
my_first_hps                                  100% 7149     7.0KB/s   00:00

johnny@johnny-wh-pc /cygdrive/C/intelFPGA/16.1/embedded/my_first_hps
$
```

Figure Shows the execution of scp command to establish connection with the board

Executing Procedures On Chip: Execution of the Program on Chip

- **The program is now stored on board's SD card**
- **Step 6: Execute the program on board (on Chip)**



Executing Procedures On Chip: Execution of the Program on Chip

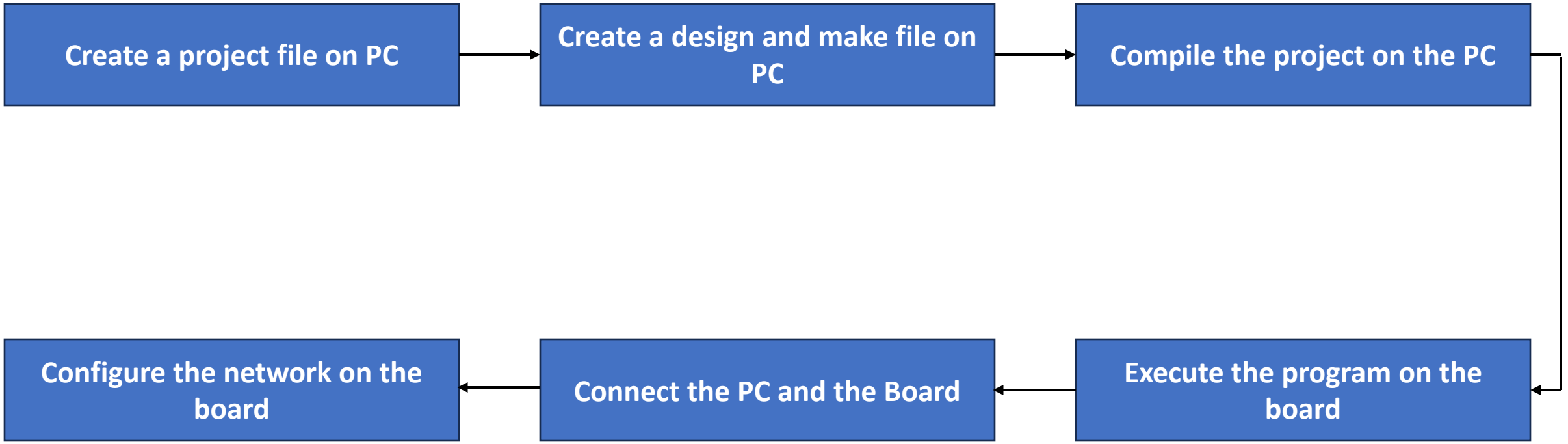
```
root@DE10_STANDARD:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 9e:77:86:70:9b:0f
          inet addr:192.168.21.134  Bcast:192.168.21.255  Mask:255.255.255.0
          inet6 addr: fe80::8bf5:80f8:866f:a9de/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:310 errors:0 dropped:0 overruns:0 frame:0
          TX packets:62 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:41303 (41.3 KB)  TX bytes:8329 (8.3 KB)
          Interrupt:39 Base address:0x8000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:160 errors:0 dropped:0 overruns:0 frame:0
          TX packets:160 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:11840 (11.8 KB)  TX bytes:11840 (11.8 KB)

root@DE10_STANDARD:~# chmod 777 my_first_hps
root@DE10_STANDARD:~#
```

```
          RX packets:160 errors:0 dropped:0 overruns:0 frame:0
          TX packets:160 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:11840 (11.8 KB)  TX bytes:11840 (11.8 KB)


root@DE10_STANDARD:~# chmod 777 my_first_hps
root@DE10_STANDARD:~# ./my_first_hps
Hello World!
root@DE10_STANDARD:~#
```

Figure on left: chmod command

Figure on right: running the file and displaying the output on chip

Executing Procedures On Chip: flowchart



QNA

Thanks

- **Dheemanth R Joshi**
- **Research Assistant**
- **Centre For Innovation and Entrepreneurship, PES University**