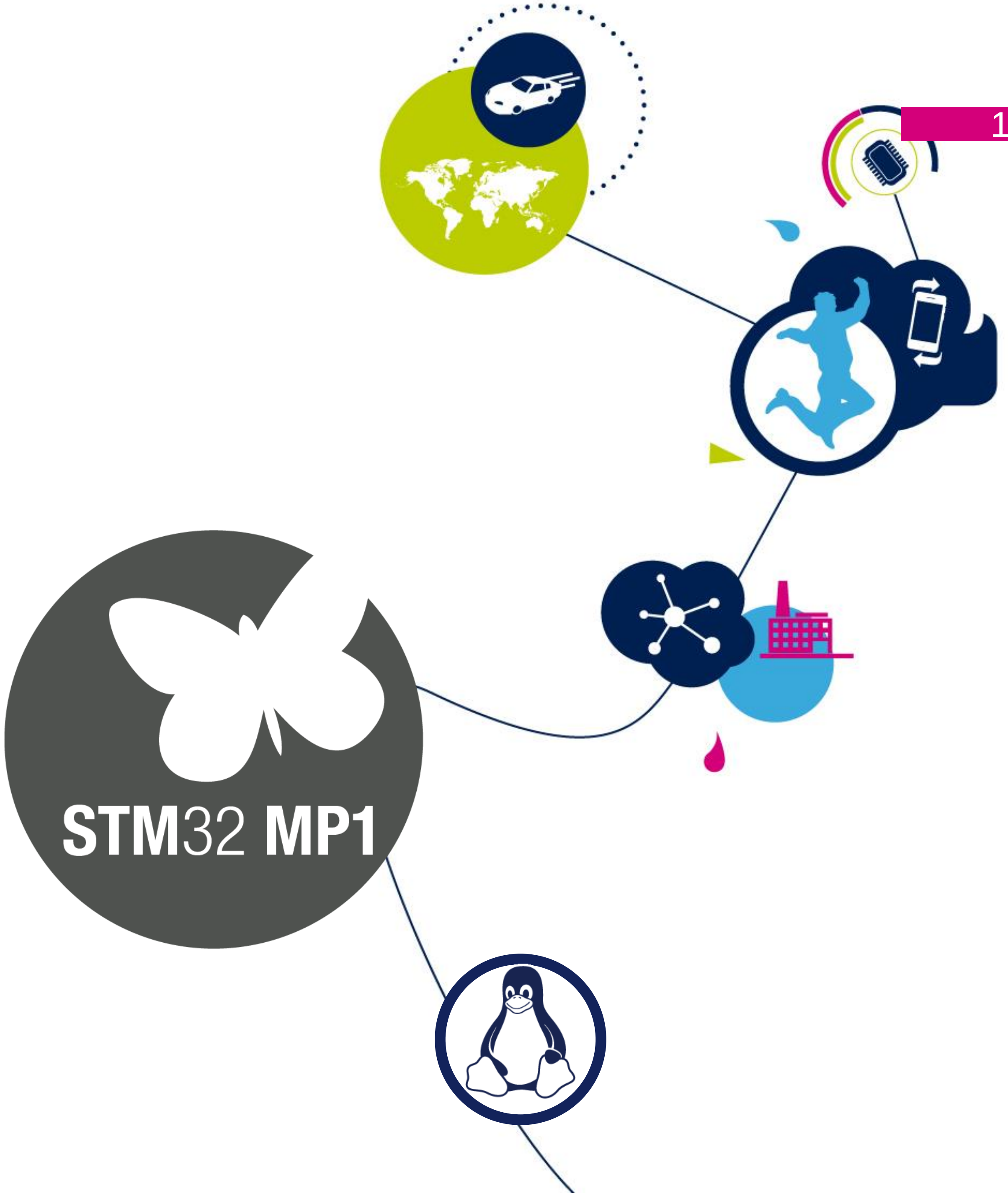
OpenSTLinux BSP pin Muxing

with STM32CubeMX for an easier HW configuration

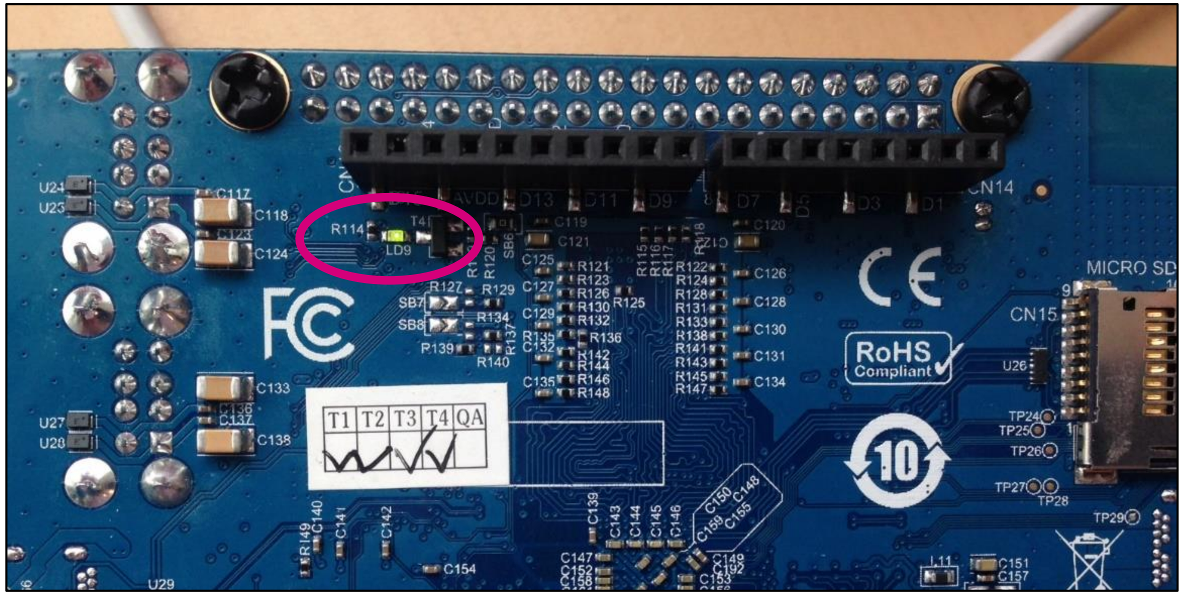


life.augmented

Device tree generation
PWM Timer 1

Lab Objective

- Set-up the OpenSTLinux timer driver with a specific PWM pin
- Start, a PWM Timer in Linux to blink a green LED every 1 sec



theory

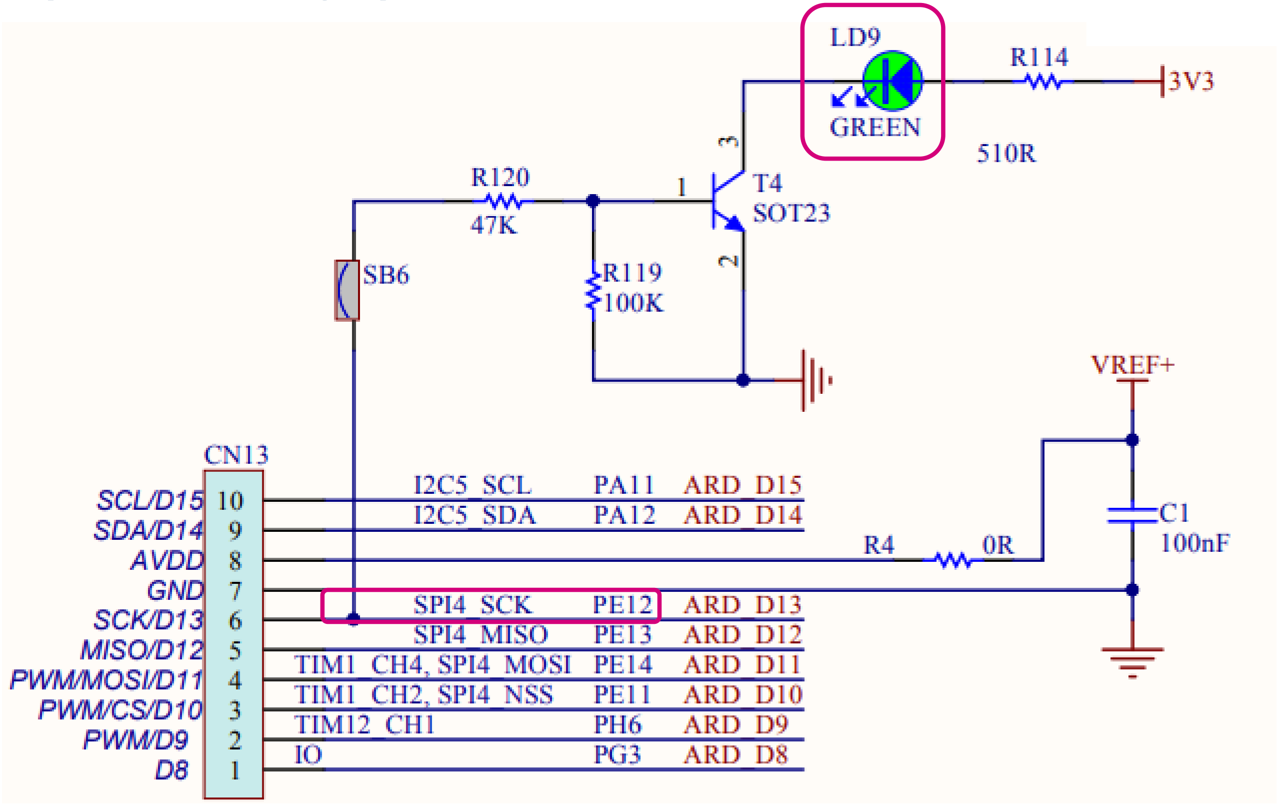
practice

Device tree generation
PWM Timer 1

STM32MP157C-DK2 hardware

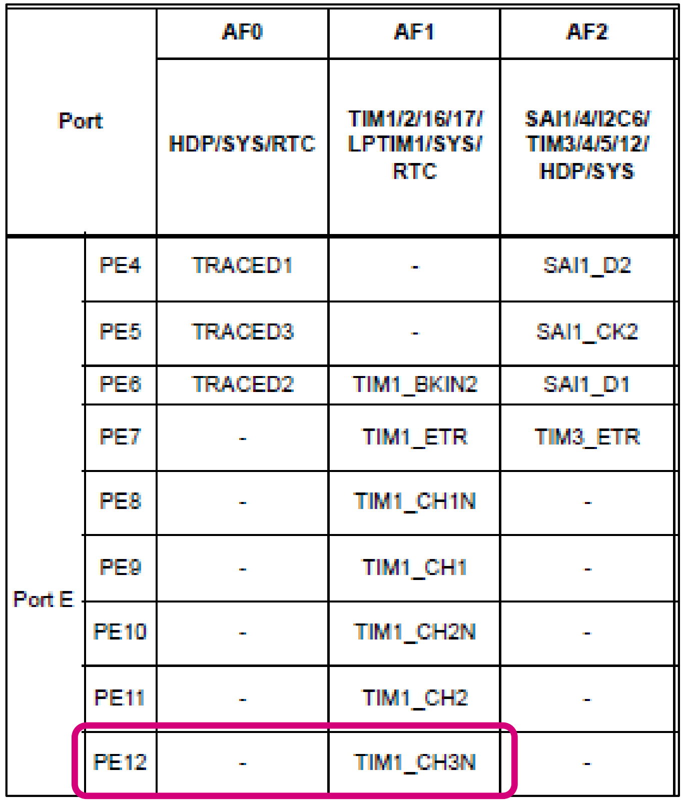
- STM32MP157C-DK2 schematics:

Green LD9 is controlled by the GPIO PE12 (Active High)



- STM32MP1 datasheet:

TIM1_CH3N is one of the PE12 alternate functions



Port		AF0	AF1	AF2
		HDP/SYS/RTC	TIM1/2/16/17/LPTIM1/SYS/RTC	SAI1/4/I2C6/TIM3/4/5/12/HDP/SYS
Port E	PE4	TRACED1	-	SAI1_D2
	PE5	TRACED3	-	SAI1_CK2
	PE6	TRACED2	TIM1_BKIN2	SAI1_D1
	PE7	-	TIM1_ETR	TIM3_ETR
	PE8	-	TIM1_CH1N	-
	PE9	-	TIM1_CH1	-
	PE10	-	TIM1_CH2N	-
	PE11	-	TIM1_CH2	-
	PE12	-	TIM1_CH3N	-

Device tree generation PWM Timer 1

Install on Linux host STM32CubeFW_MP1 for STM32CubeMX

- Install STM32CubeMP1 FW package on Linux host

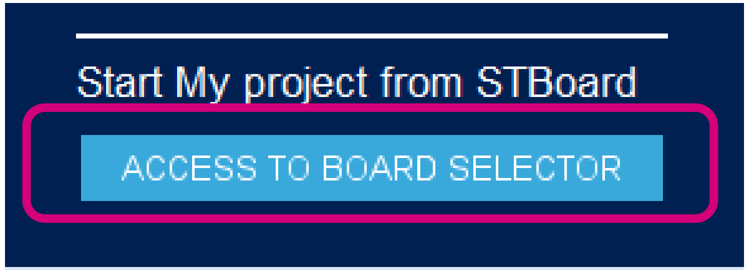
```
cd $HOME/Desktop/InputLabMaterial/Lab-Timer-A7/
mkdir -p $HOME/STM32Cube
unzip STM32Cube_FW_MP1_V1.0.1.zip -d $HOME/STM32Cube/Repository/
```

Note: assuming the lab material is already present at $HOME/Desktop/InputLabMaterial/Lab-Timer-A7
(otherwise available on the workshop USB key, if needed)

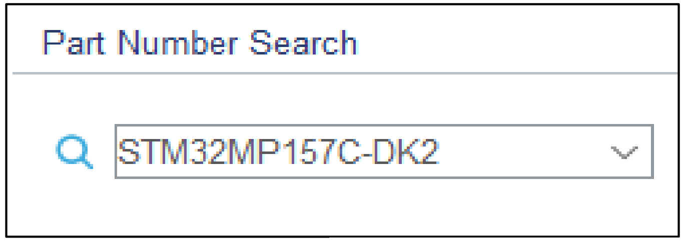
Device tree generation PWM Timer 1

Create a new STM32CubeMX project

1. Start a new project from the board selector

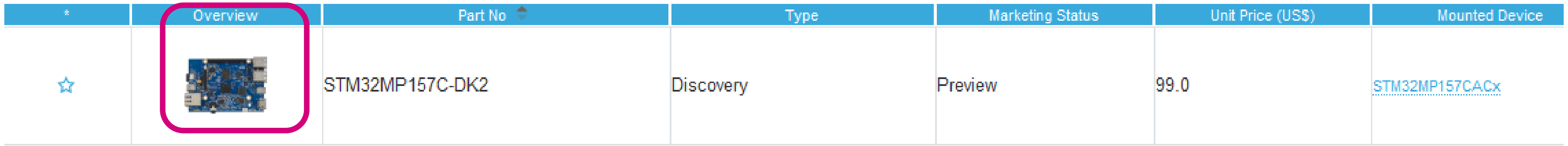


2. Select


Part Number Search

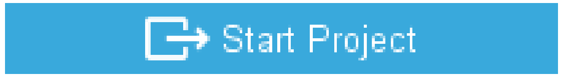
STM32MP157C-DK2

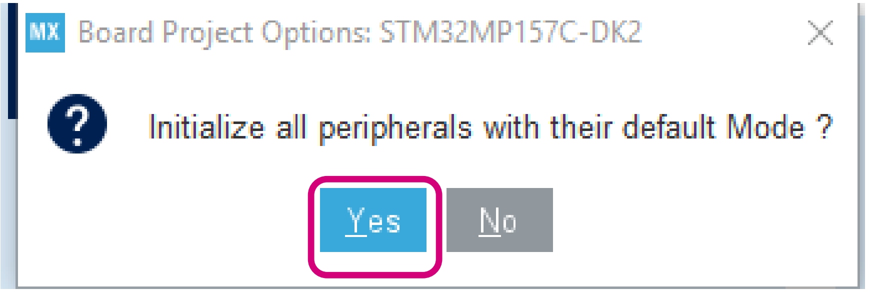
3. Click on overview



*	Overview	Part No	Type	Marketing Status	Unit Price (US$)	Mounted Device
☆		STM32MP157C-DK2	Discovery	Preview	99.0	STM32MP157CACx

4. Create project



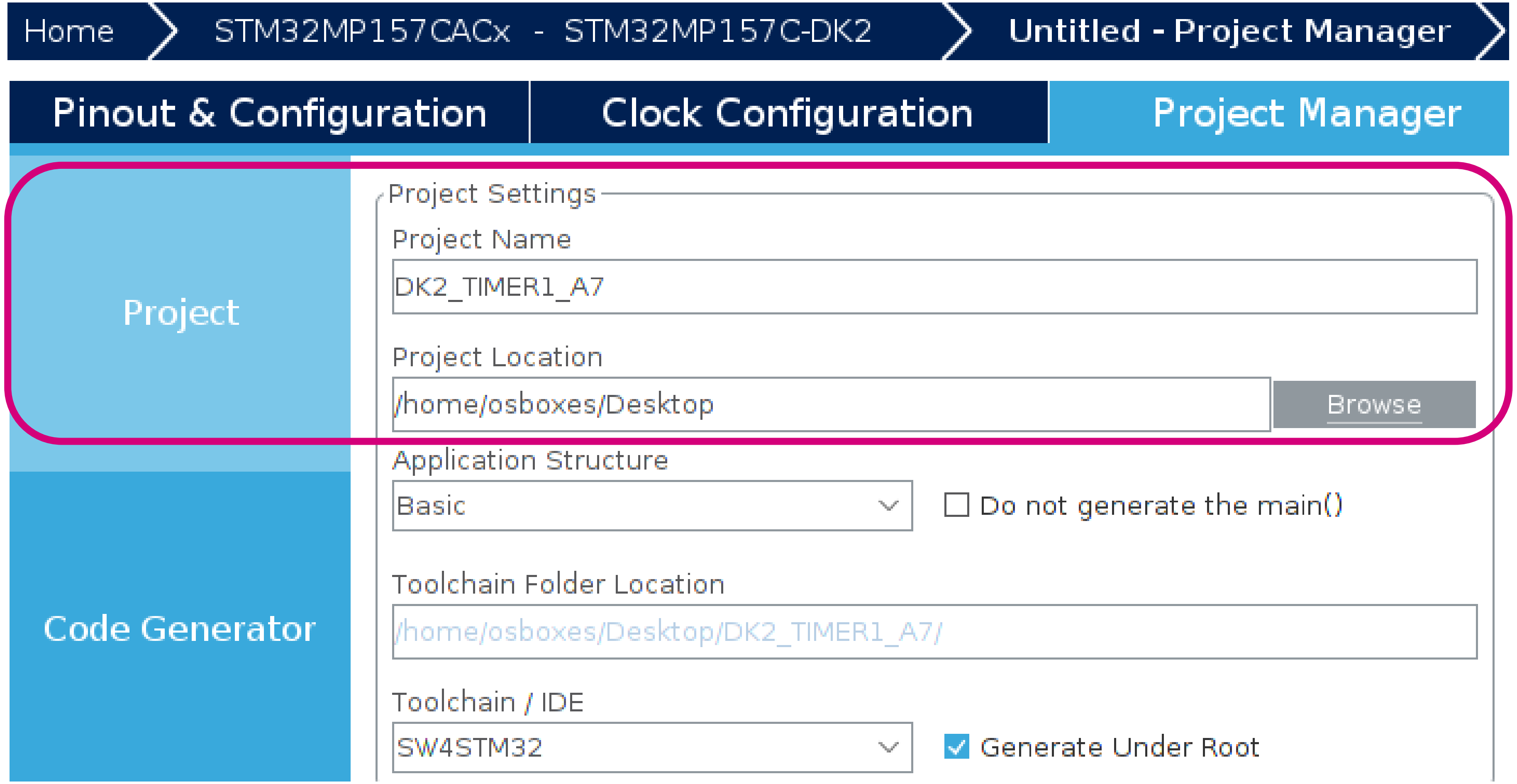

MX Board Project Options: STM32MP157C-DK2

Initialize all peripherals with their default Mode ?

Device tree generation PWM Timer 1

Configure Project paths

- In the Project Manager tab, add project name & location



Project Settings

Project Name: DK2_TIMER1_A7

Project Location: /home/osboxes/Desktop

Application Structure: Basic

□ Do not generate the main()

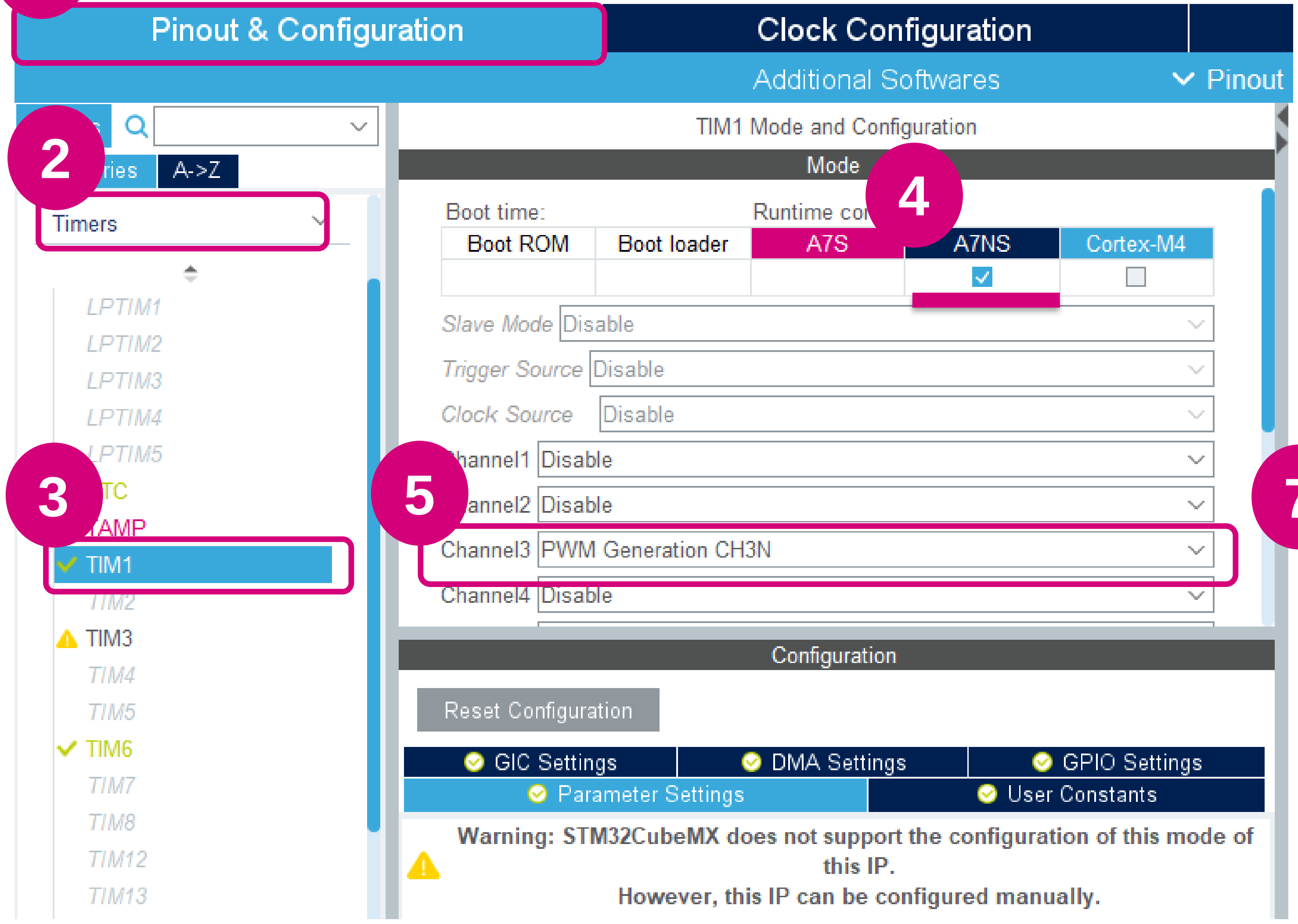
Toolchain Folder Location: /home/osboxes/Desktop/DK2_TIMER1_A7/

Toolchain / IDE: SW4STM32

☑ Generate Under Root

Device tree generation
PWM Timer 1

Configure on A7 Timer 1 Channel 3 in PWM mode



TIM1 Mode and Configuration

Mode

Boot time:		Runtime co		
Boot ROM	Boot loader	A7S	A7NS	Cortex-M4
			☑	☐

Slave Mode Disable

Trigger Source Disable

Clock Source Disable

hannel1 Disable

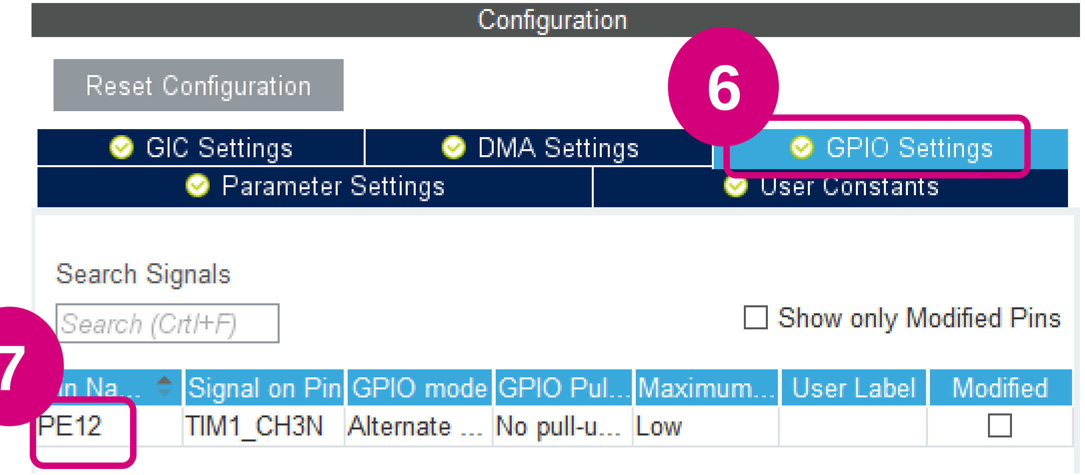
annel2 Disable

Channel3 PWM Generation CH3N

Channel4 Disable

Configuration

Warning: STM32CubeMX does not support the configuration of this mode of this IP.
However, this IP can be configured manually.



Search Signals

Search (Crtl+F)

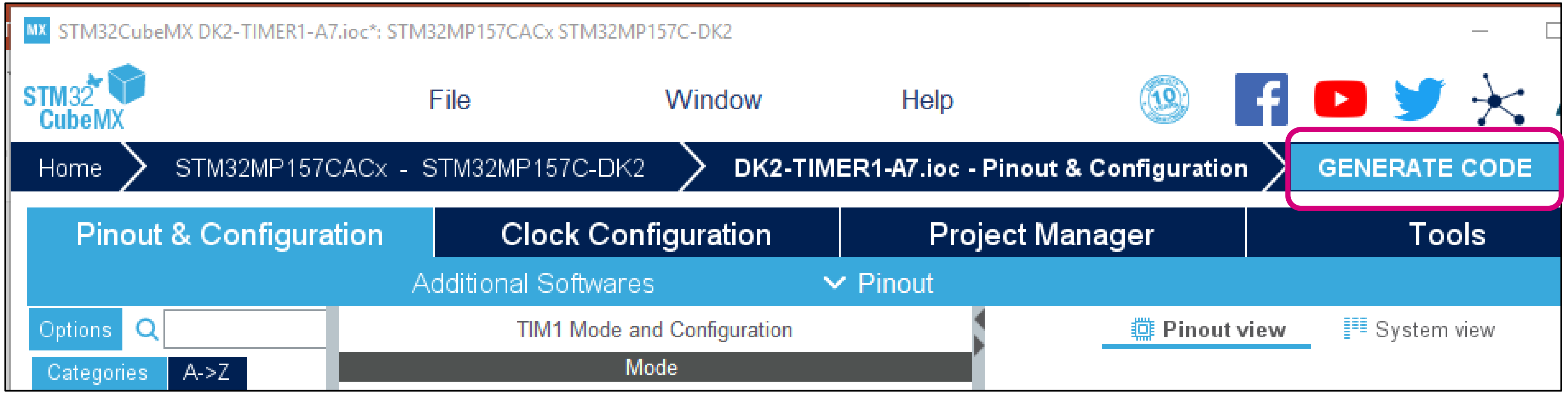
☐ Show only Modified Pins

n Na...	Signal on Pin	GPIO mode	GPIO Pul...	Maximum...	User Label	Modified
PE12	TIM1_CH3N	Alternate ...	No pull-u...	Low		☐

Device tree generation PWM Timer 1

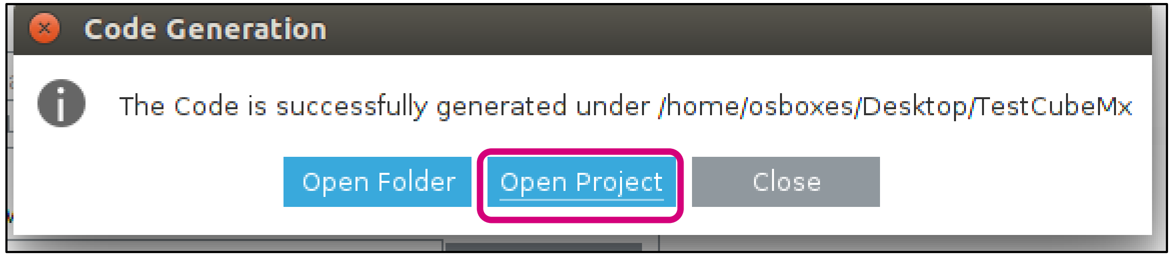
PWM Timer 1 initialization in Linux

1) Generate the kernel device tree



STM32CubeMX DK2-TIMER1-A7.ioc*: STM32MP157CACx STM32MP157C-DK2

2) Open Project



Code Generation

The Code is successfully generated under /home/osboxes/Desktop/TestCubeMx

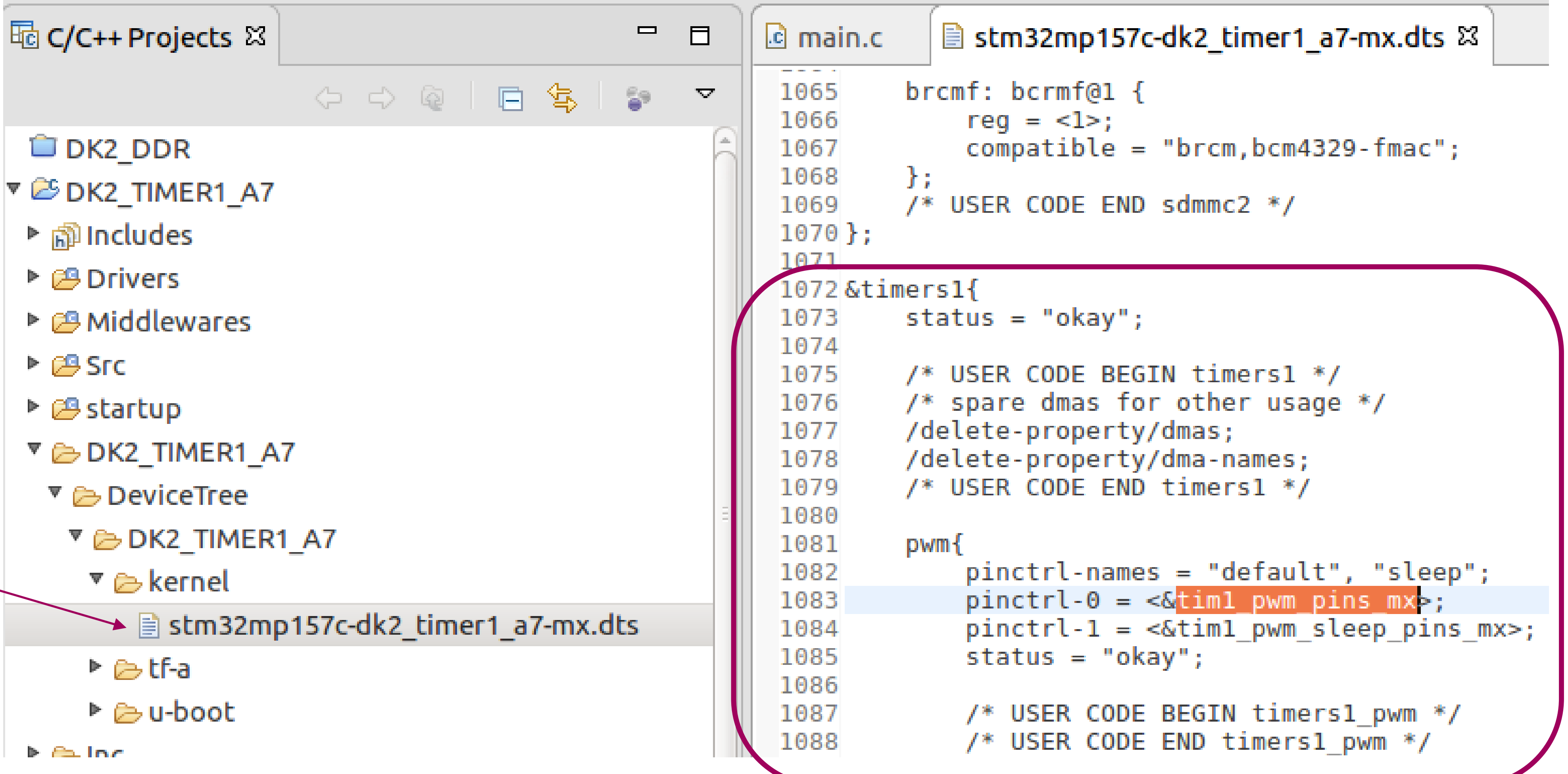
Device tree generation PWM Timer 1

PWM Timer 1 initialization in Linux

- Observe the Timer 1 device tree node

expand

double-click



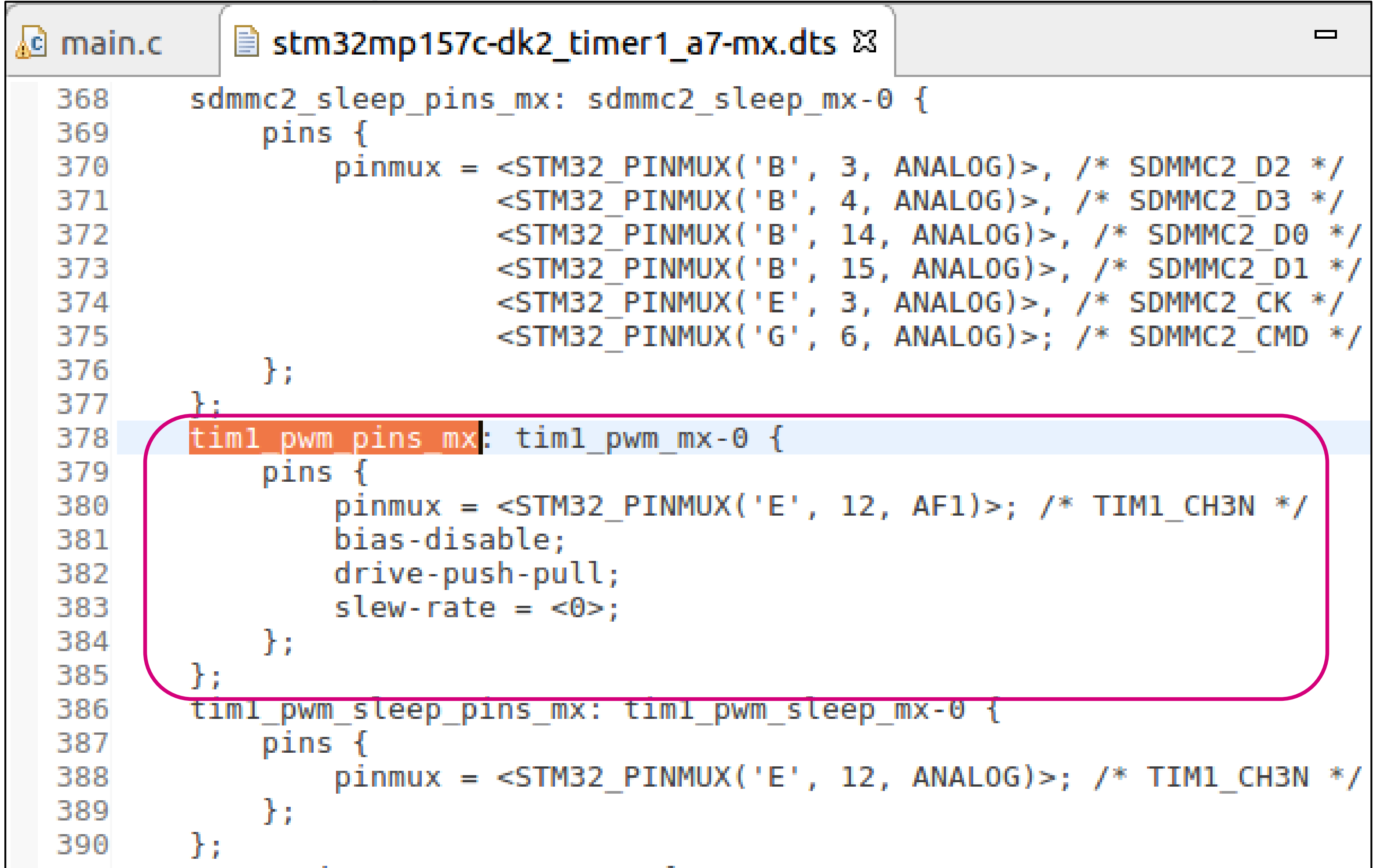
```
    brcmf: bcrmf@1 {
        reg = <1>;
        compatible = "brcm,bcm4329-fmac";
    };
    /* USER CODE END sdmmc2 */
};

&timers1{
    status = "okay";

    /* USER CODE BEGIN timers1 */
    /* spare dmas for other usage */
    /delete-property/dmas;
    /delete-property/dma-names;
    /* USER CODE END timers1 */

    pwm{
        pinctrl-names = "default", "sleep";
        pinctrl-0 = <&tim1_pwm_pins_mx>;
        pinctrl-1 = <&tim1_pwm_sleep_pins_mx>;
        status = "okay";

        /* USER CODE BEGIN timers1_pwm */
        /* USER CODE END timers1_pwm */
```

life.augmented

Device tree generation
PWM Timer 1

PWM Timer 1 initialization in Linux

- Observe the PWM Timer 1 pin controller node



```
    sdmmc2_sleep_pins_mx: sdmmc2_sleep_mx-0 {
        pins {
            pinmux = <STM32_PINMUX('B', 3, ANALOG)>, /* SDMMC2_D2 */
                     <STM32_PINMUX('B', 4, ANALOG)>, /* SDMMC2_D3 */
                     <STM32_PINMUX('B', 14, ANALOG)>, /* SDMMC2_D0 */
                     <STM32_PINMUX('B', 15, ANALOG)>, /* SDMMC2_D1 */
                     <STM32_PINMUX('E', 3, ANALOG)>, /* SDMMC2_CK */
                     <STM32_PINMUX('G', 6, ANALOG)>; /* SDMMC2_CMD */
        };
    };
    tim1_pwm_pins_mx: tim1_pwm_mx-0 {
        pins {
            pinmux = <STM32_PINMUX('E', 12, AF1)>; /* TIM1_CH3N */
            bias-disable;
            drive-push-pull;
            slew-rate = <0>;
        };
    };
    tim1_pwm_sleep_pins_mx: tim1_pwm_sleep_mx-0 {
        pins {
            pinmux = <STM32_PINMUX('E', 12, ANALOG)>; /* TIM1_CH3N */
        };
    };
```

Device tree generation PWM Timer 1

Timer 1 PWM initialization in Linux

- To save time, the device tree *stm32mp157c-dk2-timer1-A7.dts* has been compiled in ST developer package environment
- The resulted binary file is *stm32mp157c-dk2.dtb*
- *stm32mp157c-dk2.dtb* binary is already on the USB key under InputLabMaterial/Lab-Timer-A7
- Thanks to “*stm32mp157c-dk2.dtb*”, Timer 1 driver is loaded at boot time
- We will copy it on the target

INFORMATION ONLY

Device tree generation PWM Timer 1

How to Compile a device tree in Developer Package

❑ Copy the updated .dts file into kernel source :

```
cd /local/STM32MP15-Ecosystem-v1.0.0/Developer-Package/stm32mp1-openstlinux-4.19-thud-mp1-19-02-
20/sources/arm-openstlinux_weston-linux-gnueabi/linux-stm32mp-4.19-r0/linux-
4.19.9/arch/arm/boot/dts

cp $HOME/Desktop/DK2_TIMER1_A7/DeviceTree/DK2_TIMER1_A7/kernel/stm32mp157c-dk2-timer1-a7-mx.dts .
```

❑ Add it in arch/arm/boot/dts/Makefile

```
dtb-$(CONFIG_ARCH_STM32) += \
        ....
        stm32mp157c-ev1-m4-examples.dtb \
        stm32mp157c-dk2-timer1-a7-mx.dtb
```

life.augmented

INFORMATION ONLY

Device tree generation PWM Timer 1

How to Compile a device tree in Developer Package

❑ Rebuild device tree and copy and rename on target (scp network copy shown here)

```
cd /local/STM32MP15-Ecosystem-v1.0.0/Developer-Package/stm32mp1-openstlinux-4.19-
thud-mp1-19-02-20/sources/arm-openstlinux_weston-linux-gnueabi/linux-stm32mp-4.19-
r0/linux-4.19.9

source $SDK_ROOT/SDK/environment-setup-cortexa7t2hf-neon-vfpv4-openstlinux_weston-
linux-gnueabi
make dtbs
cd arch/arm/boot/dts/

scp stm32mp157c-dk2-timer1-a7-mx.dtb root@192.168.7.2:/boot/stm32mp157c-dk2.dtb
```

❑ Sync and Reboot

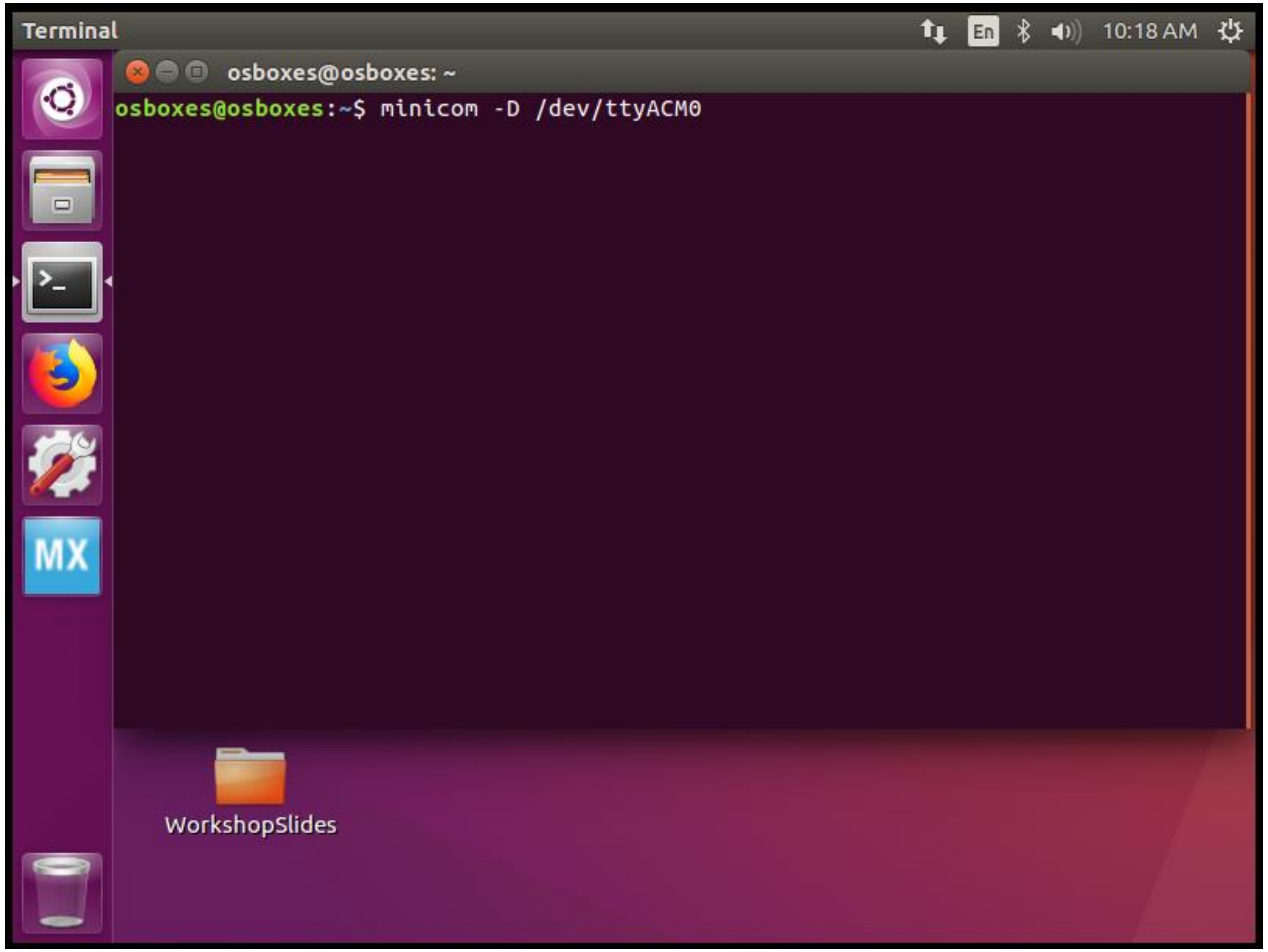
root@stm32mp1:/#

```
sync
reboot
```

life.augmented

Device tree generation PWM Timer 1

Start minicom



```
osboxes@osboxes:~$ minicom -D /dev/ttyACM0
```

- Start a new terminal and open minicom in order to establish a serial connection between Linux host and the target

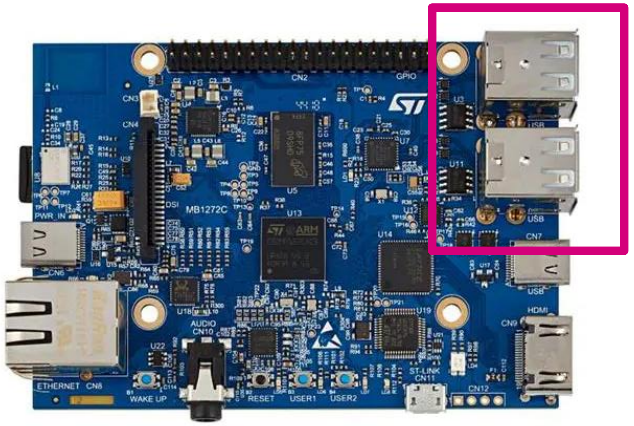
```
minicom -D /dev/ttyACM0
```

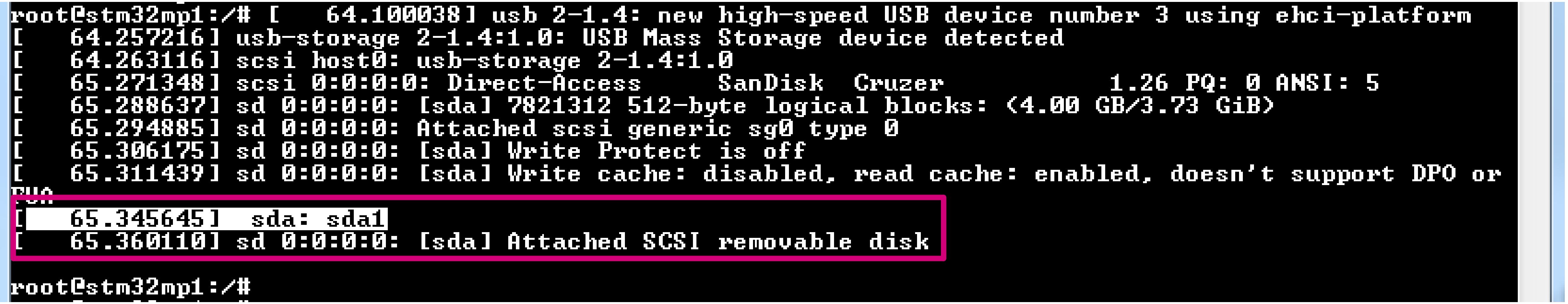
Device tree generation
PWM Timer 1

Plug in the USB stick

1) Plug the USB stick into one of the 4 USB host ports on the discovery board

2) Observe the log in the minicom console indicating that the USB stick has been recognized





```
root@stm32mp1:/# [   64.100038] usb 2-1.4: new high-speed USB device number 3 using ehci-platform
[   64.257216] usb-storage 2-1.4:1.0: USB Mass Storage device detected
[   64.263116] scsi host0: usb-storage 2-1.4:1.0
[   65.271348] scsi 0:0:0:0: Direct-Access     SanDisk  Cruzer           1.26 PQ: 0 ANSI: 5
[   65.288637] sd 0:0:0:0: [sda] 7821312 512-byte logical blocks: (4.00 GB/3.73 GiB)
[   65.294885] sd 0:0:0:0: Attached scsi generic sg0 type 0
[   65.306175] sd 0:0:0:0: [sda] Write Protect is off
[   65.311439] sd 0:0:0:0: [sda] Write cache: disabled, read cache: enabled, doesn't support DPO or
FUA
[   65.345645]  sda: sda1
[   65.360110] sd 0:0:0:0: [sda] Attached SCSI removable disk

root@stm32mp1:/#
```

life.augmented

Mount the USB stick on the Linux host

1) Mount the USB stick file system on the Linux host

```
root@stm32mp1:/#
```

```
mount /dev/sda1 /mnt
```

Note:
You can specify the file system type and enable verbose messaging with…

```
mount -t vfat /dev/sda1 /mnt -v
```

Copy the dtb to the boot directory

1) Mount the USB stick file system on the Linux host

```
root@stm32mp1:/#
```

```
mount /dev/sda1 /mnt
```

2) Copy the dtb to the boot directory

```
root@stm32mp1:/#
```

```
cp /mnt/InputLabMaterial/Lab-Timer-A7/stm32mp157c-dk2.dtb /boot
```

Sync

1) Mount the USB stick file system on the Linux host

```
root@stm32mp1:/#
```

```
mount /dev/sda1 /mnt
```

2) Copy the dtb to the boot directory

```
root@stm32mp1:/#
```

```
cp /mnt/InputLabMaterial/Lab-Timer-A7/stm32mp157c-dk2.dtb /boot
```

3) Sync

```
root@stm32mp1:/#
```

```
sync
```

Device tree generation PWM Timer 1

Reboot the target with the new device tree

- Reboot with the new device tree to initialize the Linux Timer 1 driver

```
root@stm32mp1:/#
```

```
reboot
```

life.augmented

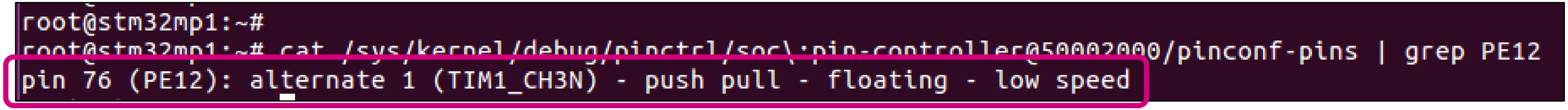
Device tree generation
PWM Timer 1

Check PWM Timer 1 Linux driver is loaded

1) Observe GPIO PE12 is assigned to Timer1 CH3N

root@stm32mp1:/#

```
cat /sys/kernel/debug/pinctrl/soc\:pin-controller@50002000/pinconf-pins | grep PE12
```

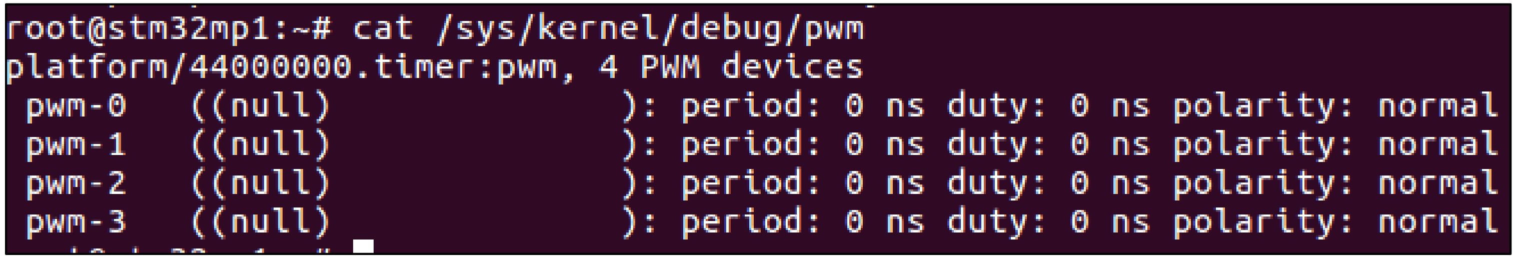


```
root@stm32mp1:~#
root@stm32mp1:~# cat /sys/kernel/debug/pinctrl/soc\:pin-controller@50002000/pinconf-pins | grep PE12
pin 76 (PE12): alternate 1 (TIM1_CH3N) - push pull - floating - low speed
```

2) Observe driver of Timer 1 (@44000000) - CH3N in mode PWM is not started

root@stm32mp1:/#

```
cat /sys/kernel/debug/pwm
```



```
root@stm32mp1:~# cat /sys/kernel/debug/pwm
platform/44000000.timer:pwm, 4 PWM devices
 pwm-0   ((null)                ): period: 0 ns duty: 0 ns polarity: normal
 pwm-1   ((null)                ): period: 0 ns duty: 0 ns polarity: normal
 pwm-2   ((null)                ): period: 0 ns duty: 0 ns polarity: normal
 pwm-3   ((null)                ): period: 0 ns duty: 0 ns polarity: normal
```

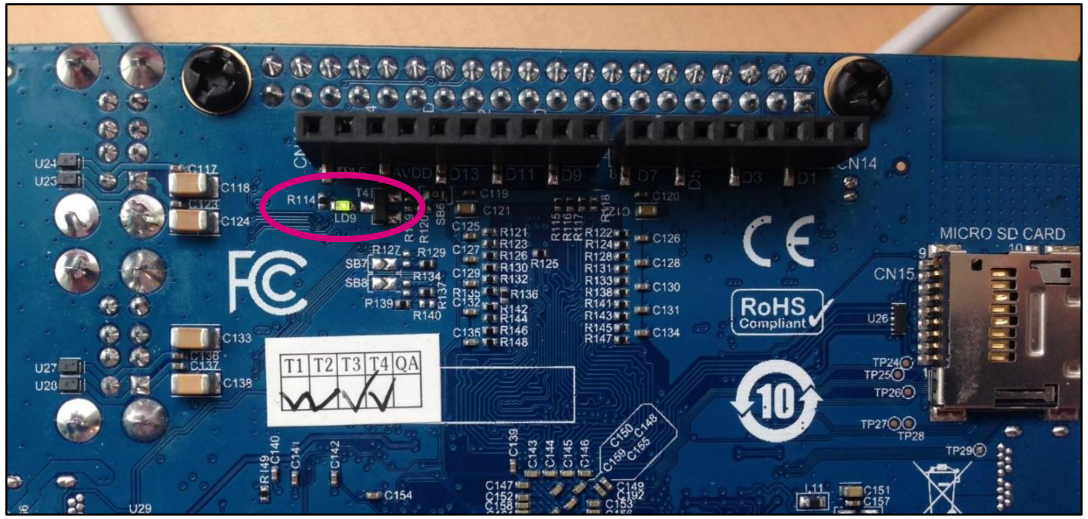
Device tree generation
PWM Timer 1

PWM Timer 1 configure and start

- Configure and start PWM Timer 1 CH3N TIM1_CH3 is pwm2 directory, period 1s (in ns)

```
root@stm32mp1:/#
```

```
cd /sys/class/pwm/pwmchip0
```



- Enable channel 3, configure period and duty cycle

```
root@stm32mp1:/#
```

```
echo 2 > export
echo 1000000000 > pwm2/period
echo 900000000 > pwm2/duty_cycle
```

Device tree generation PWM Timer 1

PWM Timer 1 configure and start

- Start PWM

```
root@stm32mp1:/#
```

```
echo 1 > pwm2/enable
```

- Stop PWM

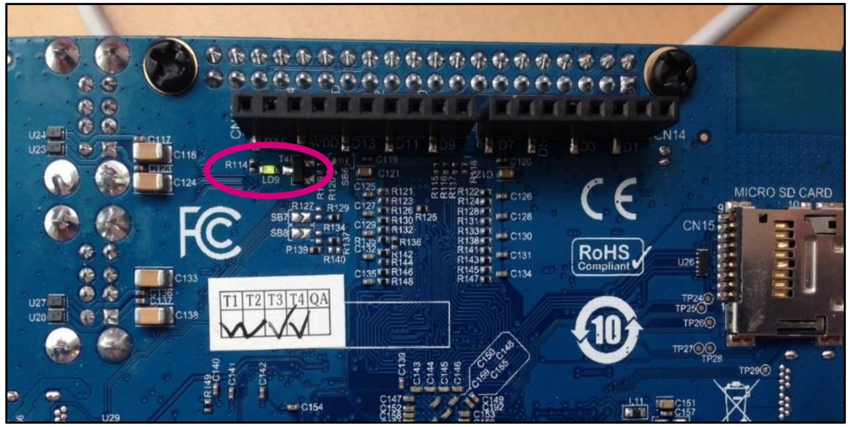
```
root@stm32mp1:/#
```

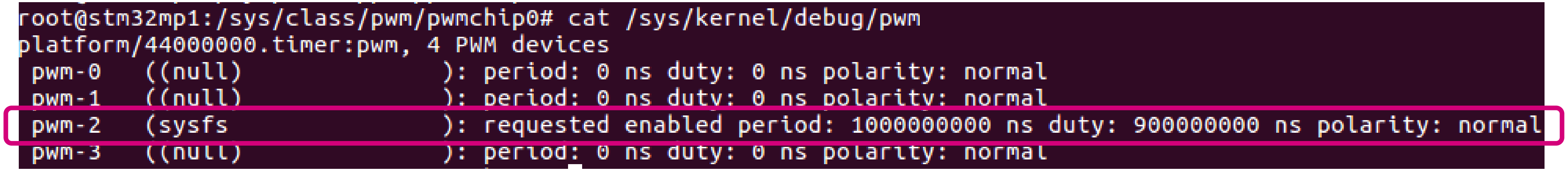
```
echo 0 > pwm2/enable
```

- Observe the green LED on the back of the board and timer driver state as below

```
root@stm32mp1:/#
```

```
cat /sys/kernel/debug/pwm
```





```
root@stm32mp1:/sys/class/pwm/pwmchip0# cat /sys/kernel/debug/pwm
platform/44000000.timer:pwm, 4 PWM devices
 pwm-0   ((null)               ): period: 0 ns duty: 0 ns polarity: normal
 pwm-1   ((null)               ): period: 0 ns duty: 0 ns polarity: normal
 pwm-2   (sysfs                ): requested enabled period: 1000000000 ns duty: 900000000 ns polarity: normal
 pwm-3   ((null)               ): period: 0 ns duty: 0 ns polarity: normal
```

Thank you