

TIM with interrupt lab **15**

- Objective

- Learn how to setup TIM with Interrupt in CubeMX
- How to Generate Code in CubeMX and use HAL functions
- Indicate TIM interrupt with LED toggle

- Goal

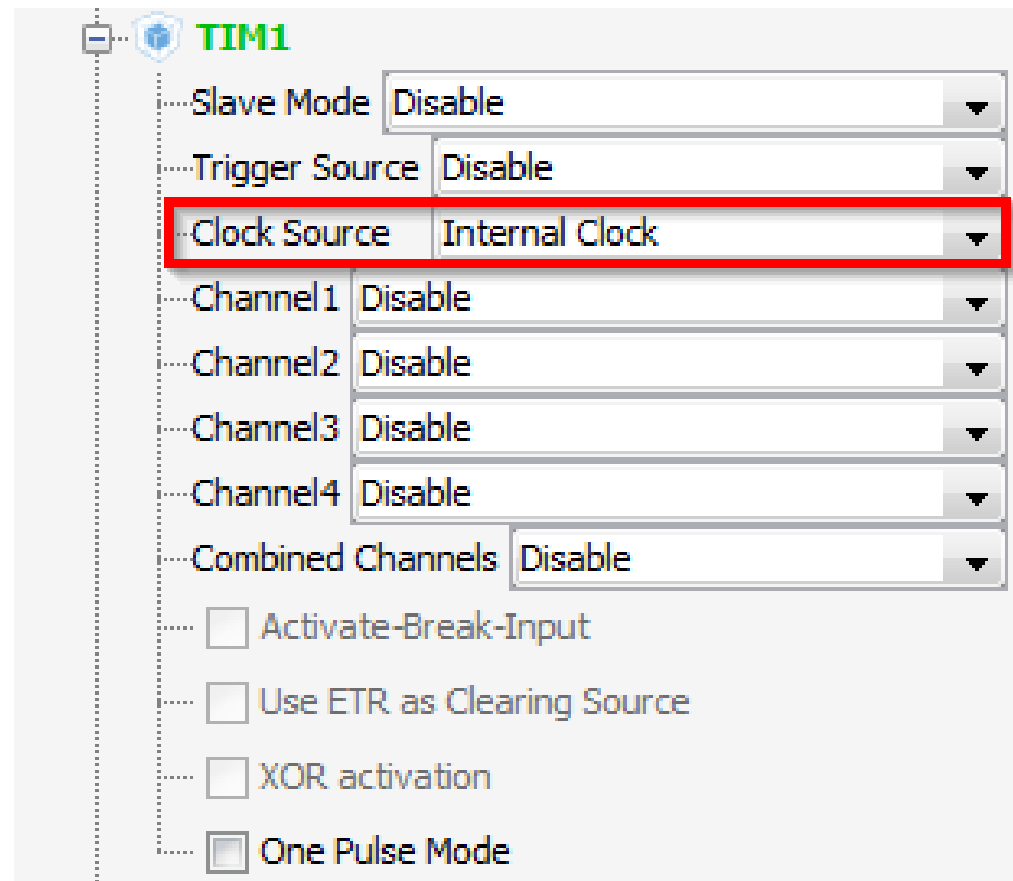
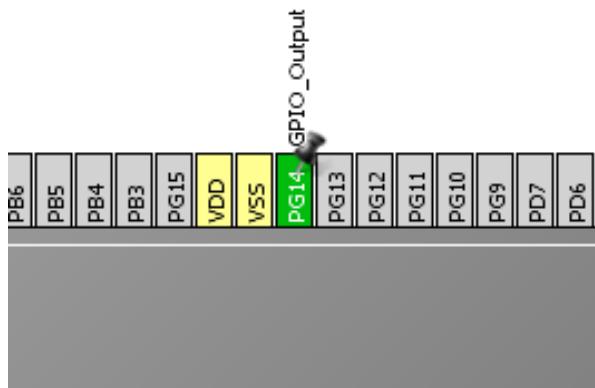
- Configure TIM in CubeMX and Generate Code
- Learn how start timer and handle interrupt
- Verify the correct functionality

15

Use TIM with interrupt

262

- Create project in CubeMX
 - Menu > File > New Project
 - Select STM32F4 > STM32F429/439 > LQFP144 > STM32F439ZITx
- CubeMX TIM selection
 - Select TIM clock source Internal clock
 - Enable GPIO for LED PG14

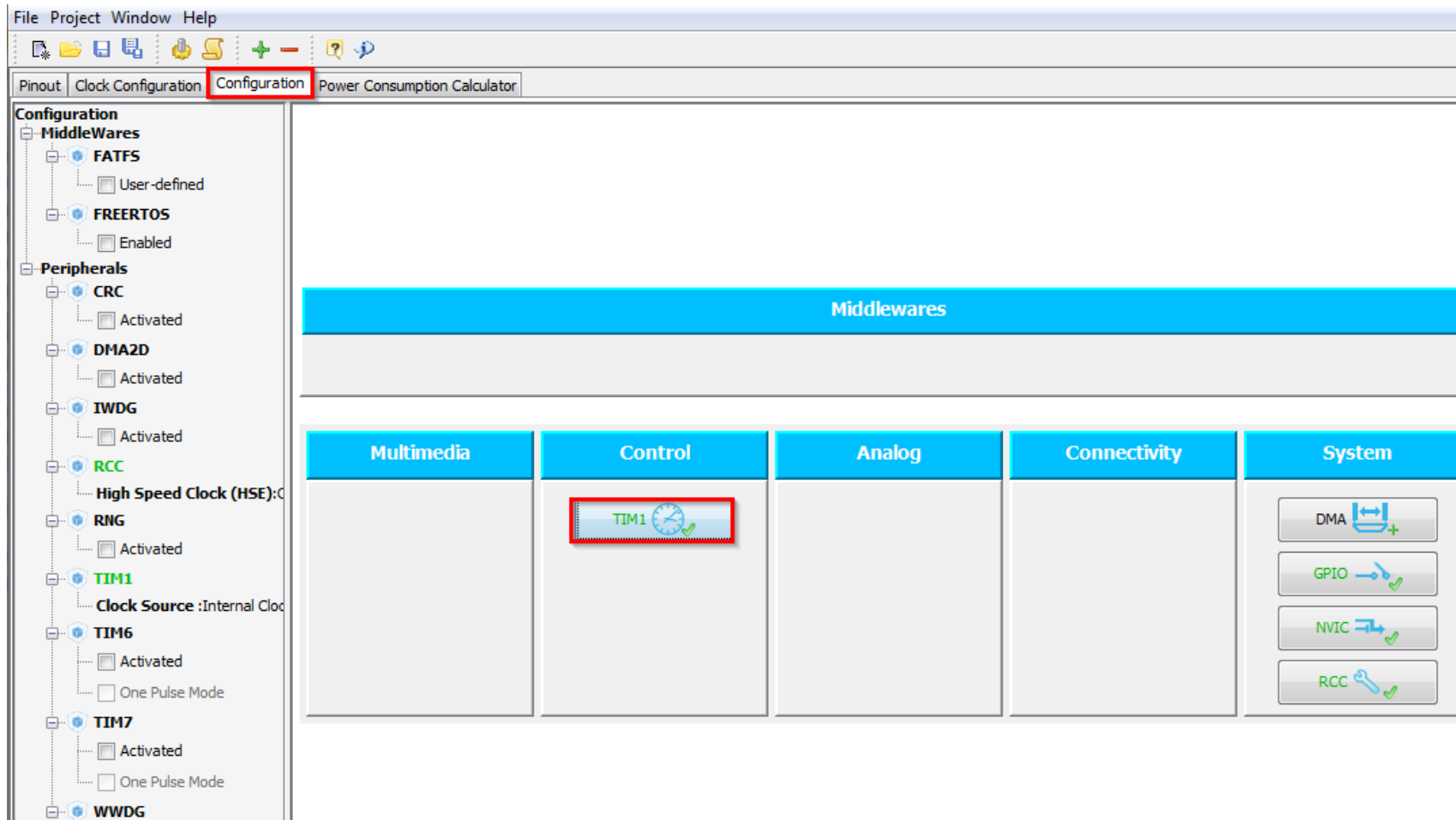


15

Use TIM with interrupt

263

- CubeMX TIM configuration
 - Tab>Configuration>Control>TIM1
 - Check the settings



15

Use TIM with interrupt

264

- CubeMX TIM configuration
 - Tab>Parameter Settings
 - Prescaler to 18000
 - Counter period to 10000
 - Together with 180MHz TIMER1 clock we get period 1Hz

TIM1 Configuration

✓ Parameter Settings ✓ NVIC Settings ✓ DMA Settings

Configure the below parameters :

[-] Counter Settings

Prescaler (PSC - 16 bits value)	18000
Counter Mode	Up
Counter Period (AutoReload Register - 16 bits value)	10000
Internal Clock Division (CKD)	No Division
Repetition Counter (RCR - 8 bits value)	0

[-] Trigger Output (TRGO) Parameters

Master/Slave Mode	Disable (no sync between this TIM (Master) and its Slaves)
Trigger Event Selection	Reset (UG bit from TIMx_EGR)

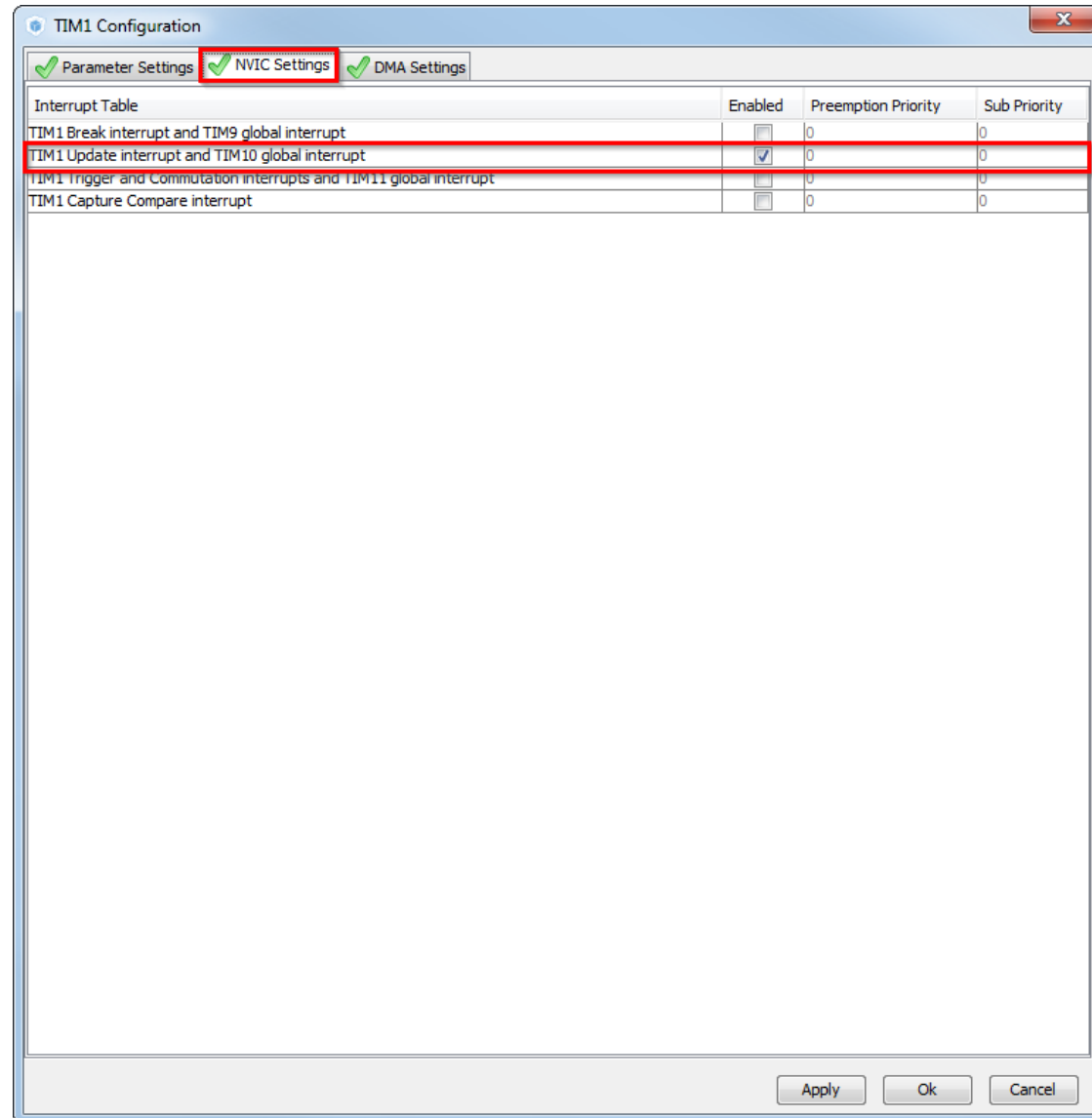
Apply Ok Cancel

15

Use TIM with interrupt

265

- CubeMX TIM configuration
 - Tab>NVIC Settings
 - Enable TIM1 Update interrupt
 - Button OK

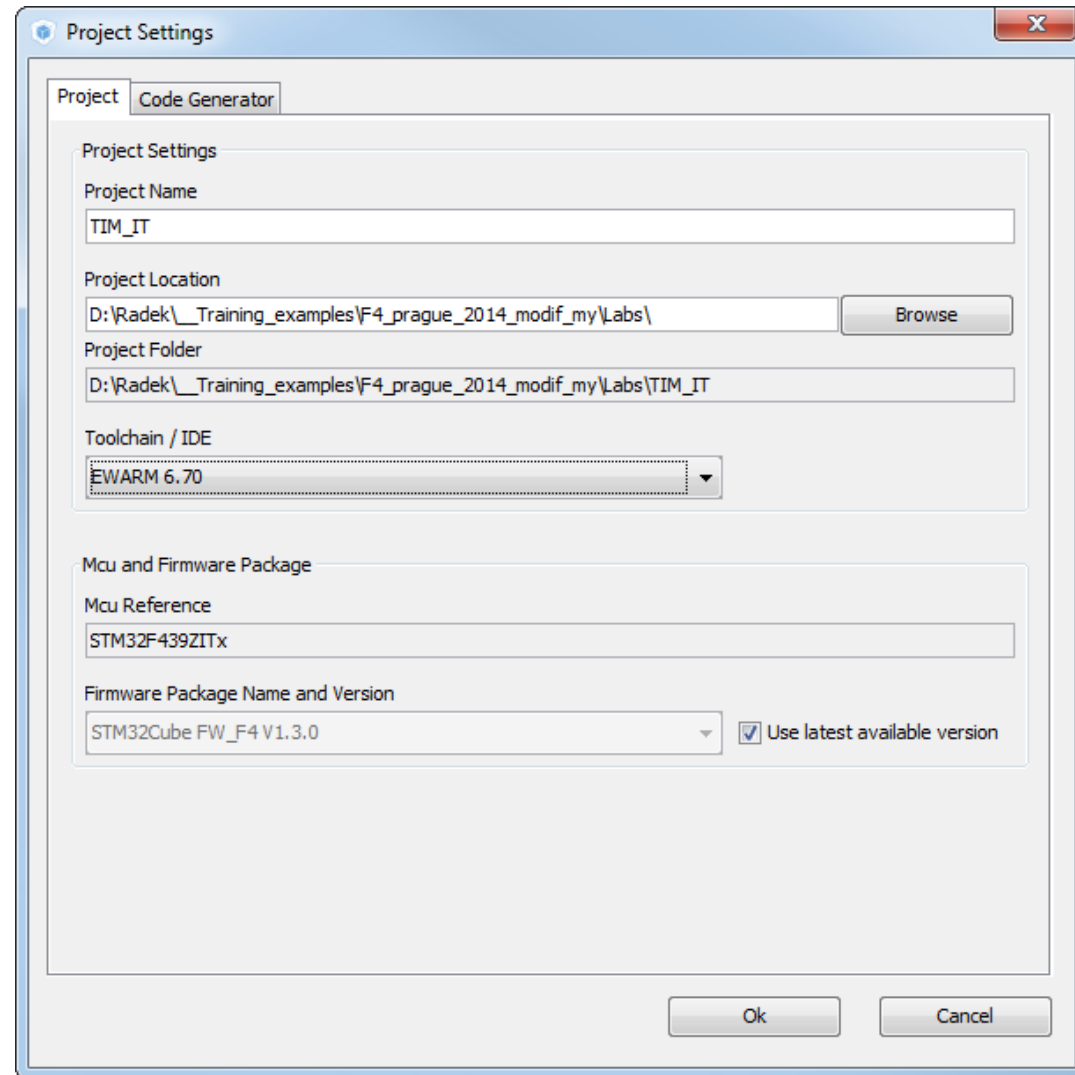


- Now we set the project details for generation

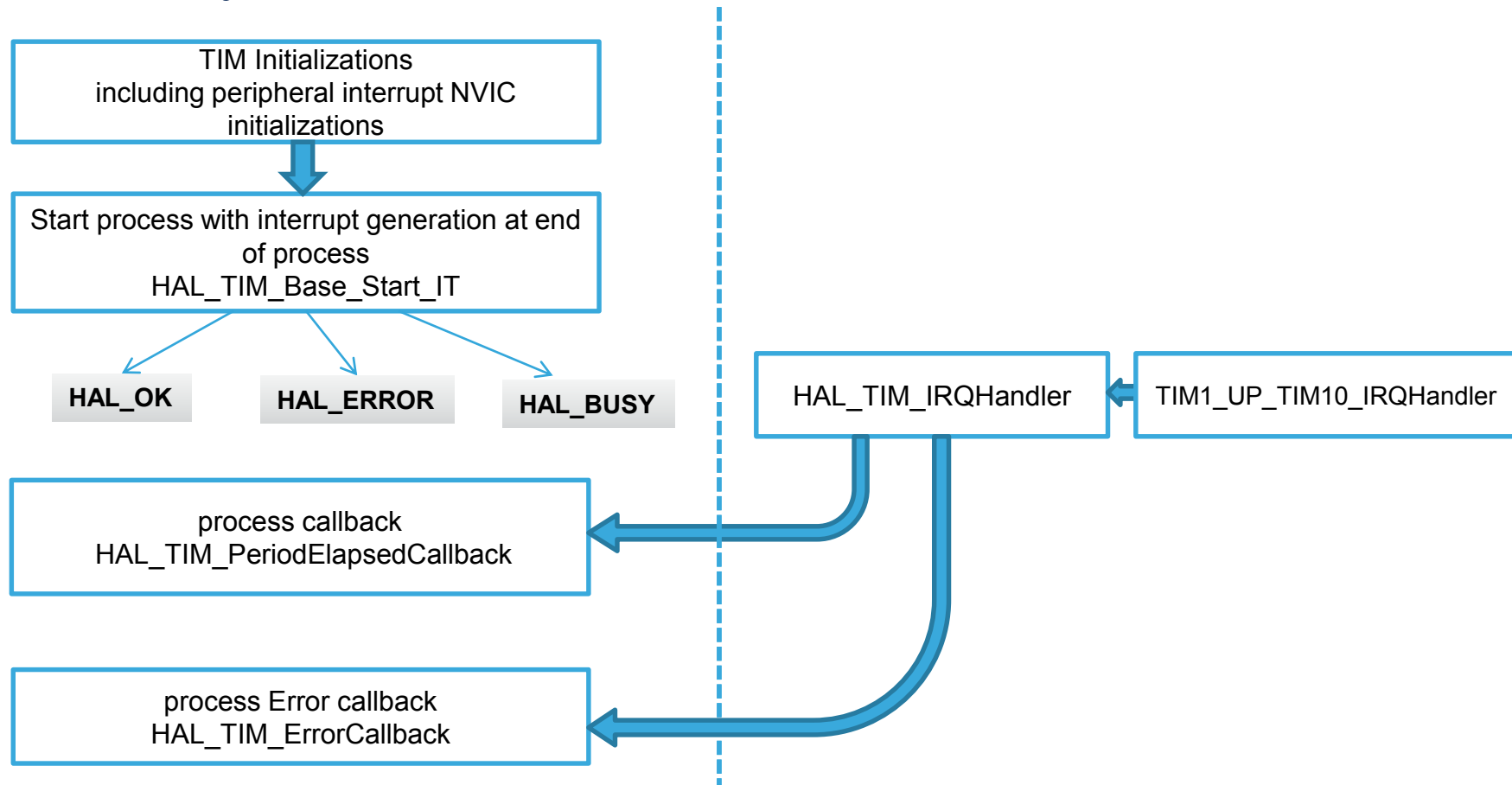
- Menu > Project > Project Settings
- Set the project name
- Project location
- Type of toolchain

- Now we can Generate Code

- Menu > Project > Generate Code



HAL Library TIM with IT flow



- Open the project in our IDE
 - The functions we want to put into main.c
 - Between */* USER CODE BEGIN 2 */* and */* USER CODE END 2 */* tags
- For TIM start use function
 - `HAL_TIM_Base_Start_IT(TIM_HandleTypeDef *htim)`
- TIM callback
 - `void TIM1_UP_TIM10_IRQHandler(void)`
- GPIO LED toggle
 - `HAL_GPIO_TogglePin(GPIO_TypeDef* GPIOx, uint16_t GPIO_Pin)`

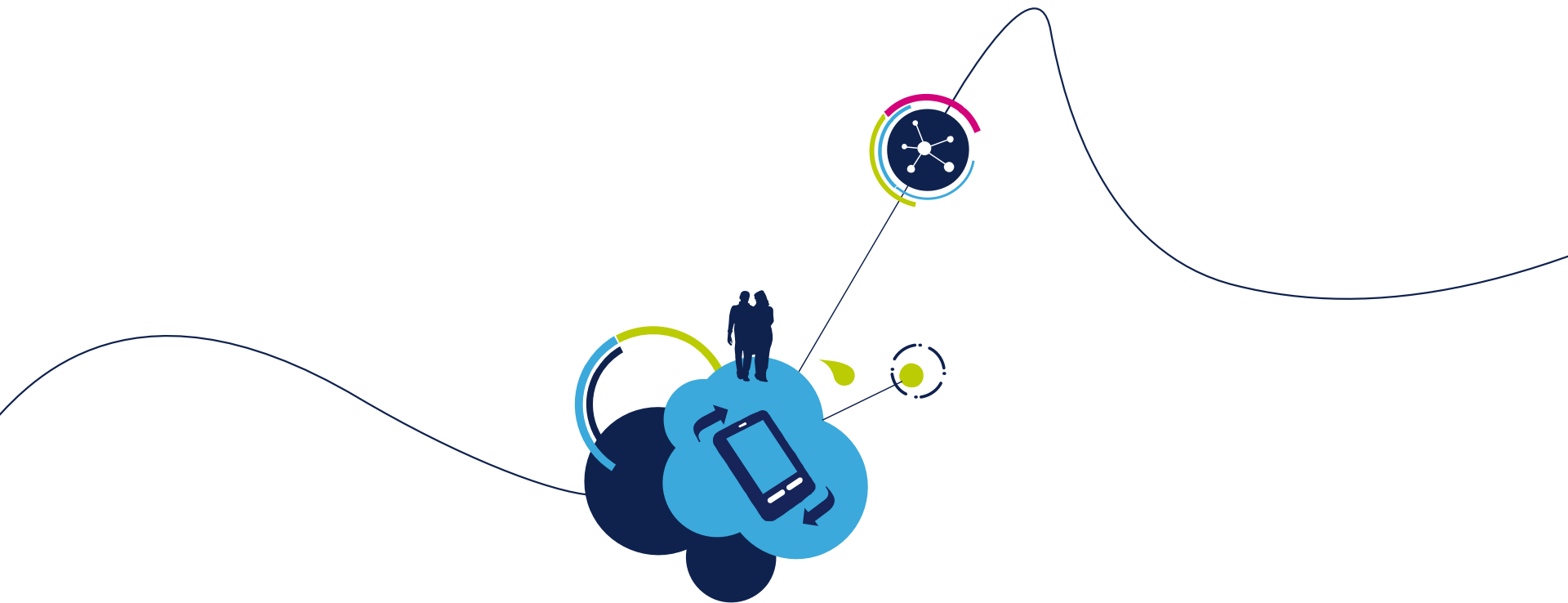
- Solution

- TIM start

```
/* USER CODE BEGIN 2 */  
HAL_TIM_Base_Start_IT(&htim1);  
/* USER CODE END 2 */
```

- Callback handling

```
/* USER CODE BEGIN 4 */  
void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim)  
{  
    HAL_GPIO_TogglePin(GPIOD,GPIO_PIN_14);  
}  
/* USER CODE END 4 */
```



TIM with PWM output lab **16**

- Objective

- Learn how to setup TIM with PWM out in CubeMX
- How to Generate Code in CubeMX and use HAL functions
- Indicate TIM PWM on LED

- Goal

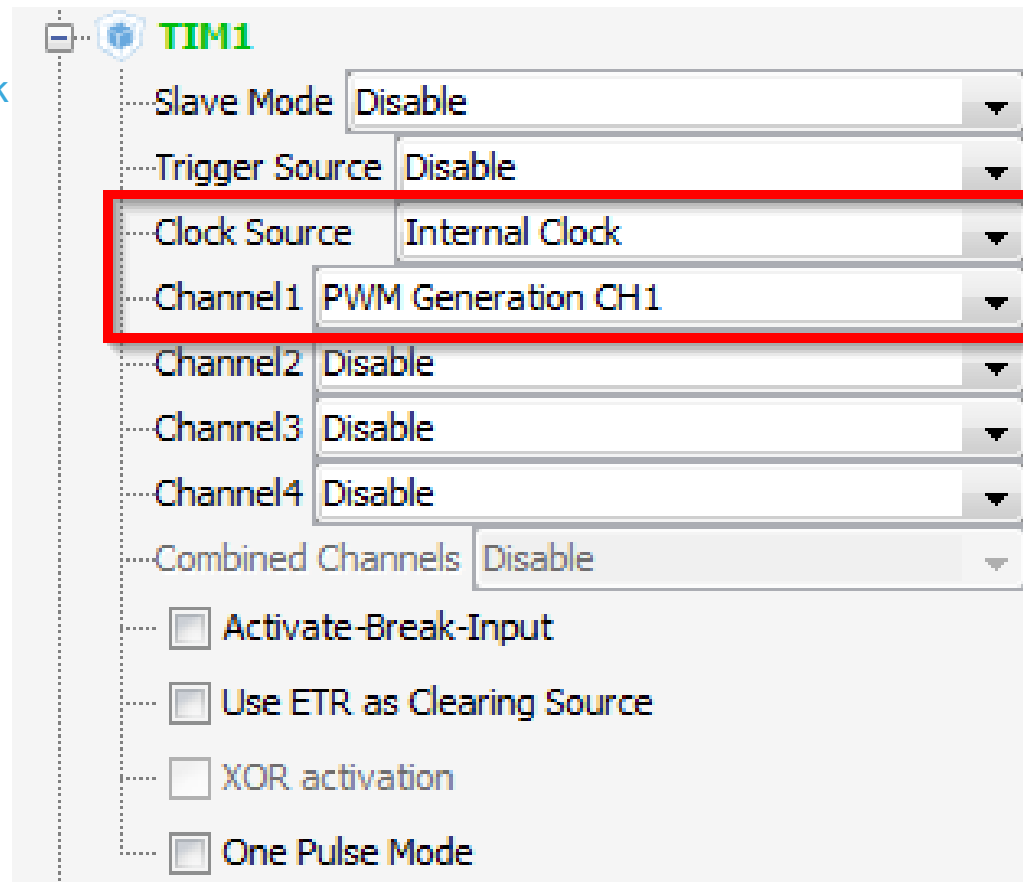
- Configure TIM in CubeMX and Generate Code
- Learn how start timer and set PWM out
- Verify the correct functionality with LED

16

Use TIM with PWM output

272

- Create project in CubeMX
 - Menu > File > New Project
 - Select STM32F4 > STM32F429/439 > LQFP144 > STM32F439ZITx
- CubeMX TIM selection
 - Select TIM clock source - Internal clock
 - Set Channel1 to PWM generation



16

Use TIM with PWM output

273

- CubeMX TIM configuration

- TAB>Configuration
>Control>TIM1
- TAB>Parameter settings
- Prescaler to 18000
- Counter period to 10000
- Together with 180MHz TIMER1 clock we get period 1Hz
- PWM pulse to 5000 this give us 1Hz blinking frequency

TIM1 Configuration

✓ Parameter Settings ✓ NVIC Settings ✓ DMA Settings ✓ GPIO Settings

Configure the below parameters :

Counter Settings

Prescaler (PSC - 16 bits value)	18000
Counter Mode	Up
Counter Period (AutoReload Register - 16 bits value)	10000
Internal Clock Division (CKD)	No Division
Repetition Counter (RCR - 8 bits value)	0

Trigger Output (TRGO) Parameters

Master/Slave Mode	Disable (no sync between this TIM (Master) and its Slaves)
Trigger Event Selection	Reset (UG bit from TIMx_EGR)

Break And Dead Time management

Break State	Disable
Break Polarity	High
Automatic Output State	Disable
Off State Selection for Run Mode (OSSR)	Disable
Off State Selection for Idle Mode (OSSR)	Disable
Lock Configuration	Off

PWM Generation Channel 1

Mode	PWM mode 1
Pulse (16 bits value)	5000
Fast Mode	Disable
CH Polarity	High
CH Idle State	Reset

Apply Ok Cancel

16

Use TIM with PWM output

274

- Now we set the project details for generation

- Menu > Project > Project Settings
- Set the project name
- Project location
- Type of toolchain

- Now we can Generate Code

- Menu > Project > Generate Code

Project Settings

Project Code Generator

Project Settings

Project Name
TIM_PWM

Project Location
D:\Radek__Training_examples\F4_prague_2014_modif_my\Labs\ Browse

Project Folder
D:\Radek__Training_examples\F4_prague_2014_modif_my\Labs\TIM_PWM

Toolchain / IDE
EWARM 6.70

Mcu and Firmware Package

Mcu Reference
STM32F439ZITx

Firmware Package Name and Version
STM32Cube FW_F4 V1.3.0 ☒ Use latest available version

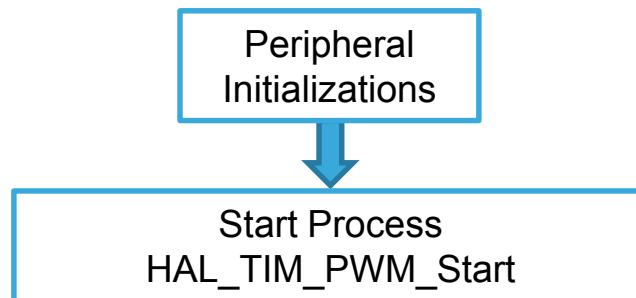
Ok Cancel

16

Use TIM with PWM output

275

- Start process TIM with PWM(same for DMA, ADC)
 - Non blocking start process



16

Use TIM with PWM output

276

- Open the project in our IDE
 - The functions we want to put into main.c
 - Between */* USER CODE BEGIN 2 */* and */* USER CODE END 2 */* tags
- For TIM start use function
 - `HAL_TIM_PWM_Start(TIM_HandleTypeDef *htim, uint32_t Channel)`
- GPIO LED toggle
 - We wire the Channel1 PE9 with LED PG14

16

Use TIM with PWM output

277

- Solution

- TIM PWM start

```
/* USER CODE BEGIN 2 */  
HAL_TIM_PWM_Start(&htim1,TIM_CHANNEL_1);  
/* USER CODE END 2 */
```

- TIM1 Channel 1 and LED connection

