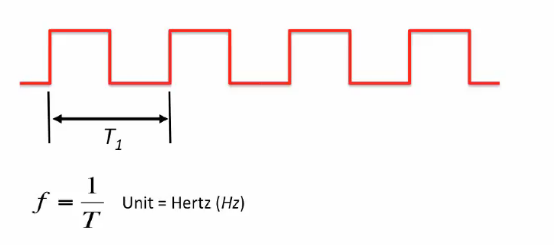
The **Pulse Width Modulation** (also known as **PWM**) is a digital modulation technique which uses duty-cycle of square waves to encode information

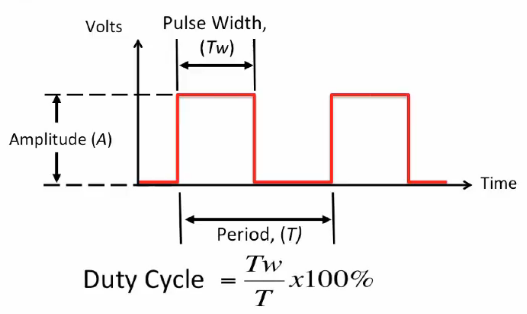
The PWM is a waveform which can basically switch between two states with a negligible raise/fall time and a constant period. The two states are usually two different voltage levels Vhigh and Vlow. In a general sense the two possible states can be named **active**and **idle.**

**STM32 Timer** (also abbreviated as **TIM**) is a peripheral which allows to generate PWM signals in hardware and this means once the Timer have been configured and started it can generate a PWM waveform on a certain output PIN without the intervention of the software.

When a Channel is configured **in PWM mode** the timer can generate a squarewave toggling its I\O line twice per period.



Duty Cycle = ratio of the pulse width to the period



0xFFFF = 65535

**typedef** **struct**

{

uint32\_t Prescaler; /\*!< Specifies the prescaler value used to divide the TIM clock.

This parameter can be a number between Min\_Data = 0x0000 and Max\_Data = 0xFFFF \*/

uint32\_t CounterMode; /\*!< Specifies the counter mode.

This parameter can be a value of @ref TIM\_Counter\_Mode \*/

uint32\_t Period; /\*!< Specifies the period value to be loaded into the active

Auto-Reload Register at the next update event.

This parameter can be a number between Min\_Data = 0x0000 and Max\_Data = 0xFFFF. \*/

uint32\_t ClockDivision; /\*!< Specifies the clock division.

This parameter can be a value of @ref TIM\_ClockDivision \*/

uint32\_t RepetitionCounter; /\*!< Specifies the repetition counter value. Each time the RCR downcounter

reaches zero, an update event is generated and counting restarts

from the RCR value (N).

This means in PWM mode that (N+1) corresponds to:

- the number of PWM periods in edge-aligned mode

- the number of half PWM period in center-aligned mode

GP timers: this parameter must be a number between Min\_Data = 0x00 and Max\_Data = 0xFF.

Advanced timers: this parameter must be a number between Min\_Data = 0x0000 and Max\_Data = 0xFFFF. \*/

uint32\_t AutoReloadPreload; /\*!< Specifies the auto-reload preload.

This parameter can be a value of @ref TIM\_AutoReloadPreload \*/

} TIM\_Base\_InitTypeDef;

**typedef** **struct**

{

uint32\_t OCMode; /\*!< Specifies the TIM mode.

This parameter can be a value of @ref TIM\_Output\_Compare\_and\_PWM\_modes \*/

uint32\_t Pulse; /\*!< Specifies the pulse value to be loaded into the Capture Compare Register.

This parameter can be a number between Min\_Data = 0x0000 and Max\_Data = 0xFFFF \*/

uint32\_t OCPolarity; /\*!< Specifies the output polarity.

This parameter can be a value of @ref TIM\_Output\_Compare\_Polarity \*/

uint32\_t OCNPolarity; /\*!< Specifies the complementary output polarity.

This parameter can be a value of @ref TIM\_Output\_Compare\_N\_Polarity

@note This parameter is valid only for timer instances supporting break feature. \*/

uint32\_t OCFastMode; /\*!< Specifies the Fast mode state.

This parameter can be a value of @ref TIM\_Output\_Fast\_State

@note This parameter is valid only in PWM1 and PWM2 mode. \*/

uint32\_t OCIdleState; /\*!< Specifies the TIM Output Compare pin state during Idle state.

This parameter can be a value of @ref TIM\_Output\_Compare\_Idle\_State

@note This parameter is valid only for timer instances supporting break feature. \*/

uint32\_t OCNIdleState; /\*!< Specifies the TIM Output Compare pin state during Idle state.

This parameter can be a value of @ref TIM\_Output\_Compare\_N\_Idle\_State

@note This parameter is valid only for timer instances supporting break feature. \*/

} TIM\_OC\_InitTypeDef;

\* @brief Starts the TIM Base generation in interrupt mode.

\* @param htim TIM Base handle

\* @retval HAL status

\*/

HAL\_StatusTypeDef HAL\_TIM\_Base\_Start\_IT(TIM\_HandleTypeDef \*htim)

/\*\*

\* @brief Starts the PWM signal generation in interrupt mode.

\* @param htim TIM PWM handle

\* @param Channel TIM Channel to be enabled

\* This parameter can be one of the following values:

\* @arg TIM\_CHANNEL\_1: TIM Channel 1 selected

\* @arg TIM\_CHANNEL\_2: TIM Channel 2 selected

\* @arg TIM\_CHANNEL\_3: TIM Channel 3 selected

\* @arg TIM\_CHANNEL\_4: TIM Channel 4 selected

\* @retval HAL status

\*/

HAL\_StatusTypeDef HAL\_TIM\_PWM\_Start\_IT(TIM\_HandleTypeDef \*htim, uint32\_t Channel)

/\*\*

\* @brief Stops the PWM signal generation in interrupt mode.

\* @param htim TIM PWM handle

\* @param Channel TIM Channels to be disabled

\* This parameter can be one of the following values:

\* @arg TIM\_CHANNEL\_1: TIM Channel 1 selected

\* @arg TIM\_CHANNEL\_2: TIM Channel 2 selected

\* @arg TIM\_CHANNEL\_3: TIM Channel 3 selected

\* @arg TIM\_CHANNEL\_4: TIM Channel 4 selected

\* @retval HAL status

\*/

HAL\_StatusTypeDef HAL\_TIM\_PWM\_Stop\_IT(TIM\_HandleTypeDef \*htim, uint32\_t Channel)