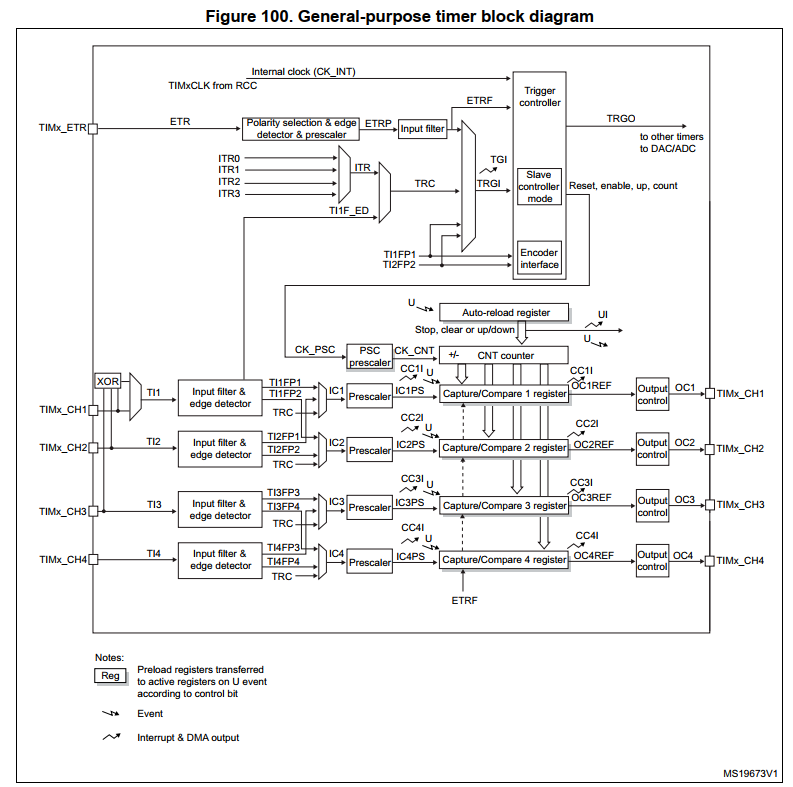
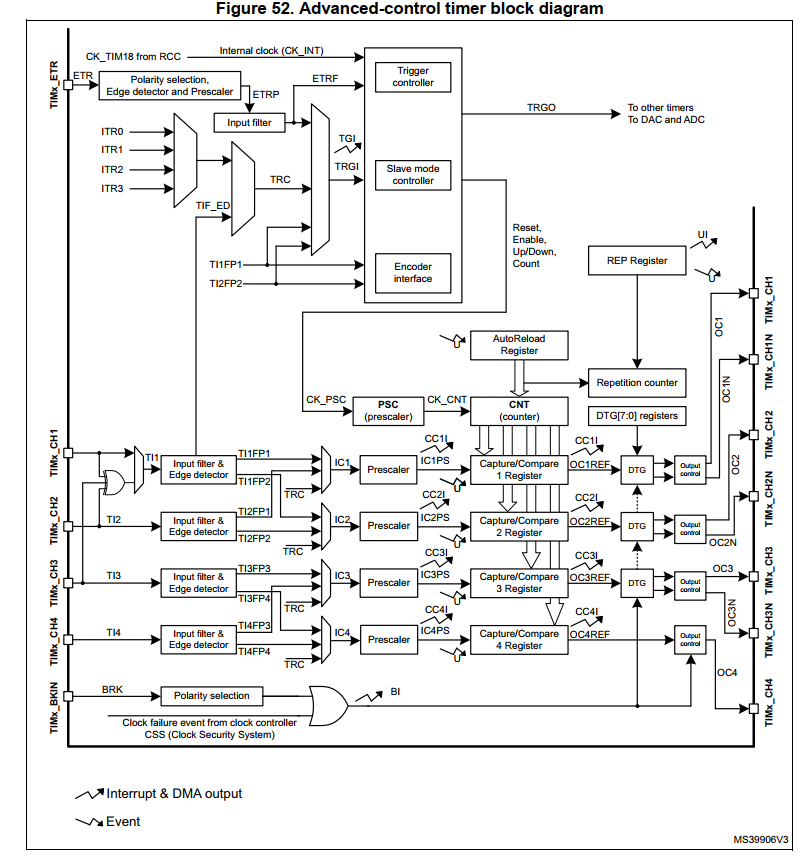
## Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Revision** | **Author** | **Changes** |
| 2022/2/22 | 1.0 | ZhangSheng | Initial version. |

## IP Description: (讲清楚IP的工作方式)

TIM1 ( Advanced timer )

TIM2~4 ( General-purpose timer )



## Feature set: （功能列表）

* 16bit up,dowm, up/down auto-reload counter.
* 16-bit programmable prescaler allowing dividing (also “on the fly”) the counter clock

frequency either by any factor between 1 and 65536.

* Up to 4 independent channels for:
  + Input capture
  + Output compare
  + PWM generation (Edge and Center-aligned Mode)
  + One-pulse mode output
* Complementary outputs with programmable dead-time.
* Synchronization circuit to control the timer with external signals and to interconnect

several timers together.

* Repetition counter to update the timer registers only after a given number of cycles of

the counter.

* Break input to put the timer’s output signals in reset state or in a known state.
* Interrupt/DMA generation on the following events:
  + Update: counter overflow/underflow, counter initialization (by software or

internal/external trigger).

* + Trigger ‘’(counter start, stop, initialization or count by internal/external trigger).
  + Input capture.
  + Output compare.
  + Break input.
* Supports incremental (quadrature) encoder and hall-sensor circuitry for positioning

Purposes.

* Trigger input for external clock or cycle-by-cycle current management.

## Feature comparison with STM32: （跟STM32的功能比较）

**Chip-Platform:**

|  |  |  |
| --- | --- | --- |
| 符号描述：   1. ✅支持 2. ❌不支持 3. ❌硬件有，但SDK计划不支持 | At103 | STM32 |
| 输入捕获/输出比较/PWM/基本定时 | ✅ | ✅ |
| 刹车功能 | ✅ | ✅ |
| 多种时钟模式（外部时钟，内部时钟） | ✅ | ✅ |
| 死区可编程的互补输出 | ✅ | ✅ |
| 霍尔接口 | ✅ | ✅ |
| 中断/DMA | ✅ | ✅ |

## API Design: （分类列出会实现的API, API尽量一样）

|  |  |  |
| --- | --- | --- |
|  | 103 | STM32 |
| init/de-init | TIM\_DeInit  TIM\_TimeBaseInit  TIM\_OCxInit (x =1 2 3 4)  TIM\_ICInit  TIM\_TimeBaseStructInit  TIM\_OCStructInit  TIM\_ICStructInit  TIM\_BDTRStructInit |  |
| Config | TIM\_PWMIConfig  TIM\_BDTRConfig  TIM\_ITConfig  TIM\_DMAConfig  TIM\_InternalClockConfig  TIM\_ITRxExternalClockConfig  TIM\_TIxExternalClockConfig  TIM\_ETRClockMode1Config  TIM\_ETRClockMode2Config  TIM\_ETRConfig  TIM\_PrescalerConfig  TIM\_CounterModeConfig  TIM\_EncoderInterfaceConfig  TIM\_ForcedOCxConfig (x =1 2 3 4)  TIM\_ARRPreloadConfig  TIM\_OCxPreloadConfig (x =1 2 3 4)  TIM\_OCxFastConfig (x =1 2 3 4)  TIM\_OCxPolarityConfig (x =1 2 3 4)  TIM\_OCxNPolarityConfig （x=1 2 3）  TIM\_UpdateDisableConfig  TIM\_UpdateRequestConfig |  |
| Set/Clear/Get/Ctrl | TIM\_SelectHallSensor  TIM\_SelectOnePulseMode  TIM\_SelectSlaveMode  TIM\_SelectMasterSlaveMode  TIM\_SelectOutputTrigger  TIM\_SelectInputTrigger  TIM\_SelectCOM  TIM\_SelectCCDMA  TIM\_SelectOCxM  TIM\_CtrlPWMOutputs  TIM\_CCPreloadControl  TIM\_GenerateEvent  TIM\_SetCounter  TIM\_SetAutoreload  TIM\_SetComparex (x =1 2 3 4)  TIM\_SetICxPrescaler (x =1 2 3 4)  TIM\_SetClockDivision  TIM\_GetCapturex (x =1 2 3 4)  TIM\_GetCounter  TIM\_GetPrescaler  TIM\_ClearOCxRef (x =1 2 3 4) |  |
| Cmd（Enable/Disable） | TIM\_Cmd  TIM\_DMACmd  TIM\_CCxCmd  TIM\_CCxNCmd |  |
| FlagStatus | TIM\_GetFlagStatus  TIM\_ClearFlag  TIM\_GetITStatus  TIM\_ClearITPendingBit |  |

## Data structure design:

typedef struct

{

  uint16\_t TIM\_Prescaler;         /\*!< Specifies the prescaler value used to divide the TIM clock.

                                       This parameter can be a number between 0x0000 and 0xFFFF \*/

  uint16\_t TIM\_CounterMode;       /\*!< Specifies the counter mode.

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

  uint16\_t TIM\_Period;            /\*!< Specifies the period value to be loaded into the active

                                       Auto-Reload Register at the next update event.

                                       This parameter must be a number between 0x0000 and 0xFFFF.  \*/

  uint16\_t TIM\_ClockDivision;     /\*!< Specifies the clock division.

                                      This parameter can be a value of @ref TIM\_Clock\_Division\_CKD \*/

  uint8\_t TIM\_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

                                       This parameter must be a number between 0x00 and 0xFF.

                                       @note This parameter is valid only for TIM1 and TIM8. \*/

} TIM\_TimeBaseInitTypeDef;

typedef struct

{

  uint16\_t TIM\_OCMode;        /\*!< Specifies the TIM mode.

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

  uint16\_t TIM\_OutputState;   /\*!< Specifies the TIM Output Compare state.

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

  uint16\_t TIM\_OutputNState;  /\*!< Specifies the TIM complementary Output Compare state.

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

                                   @note This parameter is valid only for TIM1 and TIM8. \*/

  uint16\_t TIM\_Pulse;         /\*!< Specifies the pulse value to be loaded into the Capture Compare Register.

                                   This parameter can be a number between 0x0000 and 0xFFFF \*/

  uint16\_t TIM\_OCPolarity;    /\*!< Specifies the output polarity.

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

  uint16\_t TIM\_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 TIM1 and TIM8. \*/

  uint16\_t TIM\_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 TIM1 and TIM8. \*/

  uint16\_t TIM\_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 TIM1 and TIM8. \*/

} TIM\_OCInitTypeDef;

typedef struct

{

  uint16\_t TIM\_Channel;      /\*!< Specifies the TIM channel.

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

  uint16\_t TIM\_ICPolarity;   /\*!< Specifies the active edge of the input signal.

                                  This parameter can be a value of @ref TIM\_Input\_Capture\_Polarity \*/

  uint16\_t TIM\_ICSelection;  /\*!< Specifies the input.

                                  This parameter can be a value of @ref TIM\_Input\_Capture\_Selection \*/

  uint16\_t TIM\_ICPrescaler;  /\*!< Specifies the Input Capture Prescaler.

                                  This parameter can be a value of @ref TIM\_Input\_Capture\_Prescaler \*/

  uint16\_t TIM\_ICFilter;     /\*!< Specifies the input capture filter.

                                  This parameter can be a number between 0x0 and 0xF \*/

} TIM\_ICInitTypeDef;

typedef struct

{

  uint16\_t TIM\_OSSRState;        /\*!< Specifies the Off-State selection used in Run mode.

                                      This parameter can be a value of @ref OSSR\_Off\_State\_Selection\_for\_Run\_mode\_state \*/

  uint16\_t TIM\_OSSIState;        /\*!< Specifies the Off-State used in Idle state.

                                      This parameter can be a value of @ref OSSI\_Off\_State\_Selection\_for\_Idle\_mode\_state \*/

  uint16\_t TIM\_LOCKLevel;        /\*!< Specifies the LOCK level parameters.

                                      This parameter can be a value of @ref Lock\_level \*/

  uint16\_t TIM\_DeadTime;         /\*!< Specifies the delay time between the switching-off and the

                                      switching-on of the outputs.

                                      This parameter can be a number between 0x00 and 0xFF  \*/

  uint16\_t TIM\_Break;            /\*!< Specifies whether the TIM Break input is enabled or not.

                                      This parameter can be a value of @ref Break\_Input\_enable\_disable \*/

  uint16\_t TIM\_BreakPolarity;    /\*!< Specifies the TIM Break Input pin polarity.

                                      This parameter can be a value of @ref Break\_Polarity \*/

  uint16\_t TIM\_AutomaticOutput;  /\*!< Specifies whether the TIM Automatic Output feature is enabled or not.

                                      This parameter can be a value of @ref TIM\_AOE\_Bit\_Set\_Reset \*/

} TIM\_BDTRInitTypeDef;

## Known behavior and issue.