67 HAL TSC Generic Driver

67.1 TSC Firmware driver registers structures

67.1.1 TSC_InitTypeDef

Data Fields

- ***uint32_t CTPulseHighLength***
- ***uint32_t CTPulseLowLength***
- ***uint32_t SpreadSpectrum***
- ***uint32_t SpreadSpectrumDeviation***
- ***uint32_t SpreadSpectrumPrescaler***
- ***uint32_t PulseGeneratorPrescaler***
- ***uint32_t MaxCountValue***
- ***uint32_t IODefaultMode***
- ***uint32_t SynchroPinPolarity***
- ***uint32_t AcquisitionMode***
- ***uint32_t MaxCountInterrupt***
- ***uint32_t ChannelIOs***
- ***uint32_t ShieldIOs***
- ***uint32_t SamplingIOs***

Field Documentation

- ***uint32_t TSC_InitTypeDef::CTPulseHighLength***
  Charge-transfer high pulse length This parameter can be a value of ***TSC_CTPulseHL_Config***
- ***uint32_t TSC_InitTypeDef::CTPulseLowLength***
  Charge-transfer low pulse length This parameter can be a value of ***TSC_CTPulseLL_Config***
- ***uint32_t TSC_InitTypeDef::SpreadSpectrum***
  Spread spectrum activation This parameter can be a value of ***TSC_CTPulseLL_Config***
- ***uint32_t TSC_InitTypeDef::SpreadSpectrumDeviation***
  Spread spectrum deviation This parameter must be a number between Min_Data = 0 and Max_Data = 127
- ***uint32_t TSC_InitTypeDef::SpreadSpectrumPrescaler***
  Spread spectrum prescaler This parameter can be a value of ***TSC_SpreadSpec_Prescaler***
- ***uint32_t TSC_InitTypeDef::PulseGeneratorPrescaler***
  Pulse generator prescaler This parameter can be a value of ***TSC_PulseGenerator_Prescaler***
- ***uint32_t TSC_InitTypeDef::MaxCountValue***
  Max count value This parameter can be a value of ***TSC_MaxCount_Value***
- ***uint32_t TSC_InitTypeDef::IODefaultMode***
  IO default mode This parameter can be a value of ***TSC_IO_Default_Mode***
- ***uint32_t TSC_InitTypeDef::SynchroPinPolarity***
  Synchro pin polarity This parameter can be a value of ***TSC_Synchro_Pin_Polarity***
- ***uint32_t TSC_InitTypeDef::AcquisitionMode***
  Acquisition mode This parameter can be a value of ***TSC_Acquisition_Mode***
- ***uint32_t TSC_InitTypeDef::MaxCountInterrupt***
  Max count interrupt activation This parameter can be set to ENABLE or DISABLE.

- ***uint32_t TSC_InitTypeDef::ChannelIOs***
  Channel IOs mask
- ***uint32_t TSC_InitTypeDef::ShieldIOs***
  Shield IOs mask
- ***uint32_t TSC_InitTypeDef::SamplingIOs***
  Sampling IOs mask

67.1.2 TSC_IOConfigTypeDef

Data Fields

- ***uint32_t ChannelIOs***
- ***uint32_t ShieldIOs***
- ***uint32_t SamplingIOs***

Field Documentation

- ***uint32_t TSC_IOConfigTypeDef::ChannelIOs***
  Channel IOs mask
- ***uint32_t TSC_IOConfigTypeDef::ShieldIOs***
  Shield IOs mask
- ***uint32_t TSC_IOConfigTypeDef::SamplingIOs***
  Sampling IOs mask

67.1.3 TSC_HandleTypeDef

Data Fields

- ***TSC_TypeDef * Instance***
- ***TSC_InitTypeDef Init***
- ***__IO HAL_TSC_StateTypeDef State***
- ***HAL_LockTypeDef Lock***

Field Documentation

- ***TSC_TypeDef* TSC_HandleTypeDef::Instance***
  Register base address
- ***TSC_InitTypeDef TSC_HandleTypeDef::Init***
  Initialization parameters
- ***__IO HAL_TSC_StateTypeDef TSC_HandleTypeDef::State***
  Peripheral state
- ***HAL_LockTypeDef TSC_HandleTypeDef::Lock***
  Lock feature

67.2 TSC Firmware driver API description

67.2.1 TSC specific features

1. Proven and robust surface charge transfer acquisition principle
2. Supports up to 3 capacitive sensing channels per group
3. Capacitive sensing channels can be acquired in parallel offering a very good response time
4. Spread spectrum feature to improve system robustness in noisy environments
5. Full hardware management of the charge transfer acquisition sequence
6. Programmable charge transfer frequency
7. Programmable sampling capacitor I/O pin
8. Programmable channel I/O pin
9. Programmable max count value to avoid long acquisition when a channel is faulty

10. Dedicated end of acquisition and max count error flags with interrupt capability
11. One sampling capacitor for up to 3 capacitive sensing channels to reduce the system components
12. Compatible with proximity, touchkey, linear and rotary touch sensor implementation

67.2.2 How to use this driver

1. Enable the TSC interface clock using __HAL_RCC_TSC_CLK_ENABLE() macro.
2. GPIO pins configuration
   - Enable the clock for the TSC GPIOs using __HAL_RCC_GPIOx_CLK_ENABLE() macro.
   - Configure the TSC pins used as sampling IOs in alternate function output Open-Drain mode, and TSC pins used as channel/shield IOs in alternate function output Push-Pull mode using HAL_GPIO_Init() function.
3. Interrupts configuration
   - Configure the NVIC (if the interrupt model is used) using HAL_NVIC_SetPriority() and HAL_NVIC_EnableIRQ() and function.
4. TSC configuration
   - Configure all TSC parameters and used TSC IOs using HAL_TSC_Init() function.

TSC peripheral alternate functions are mapped on AF9.

Acquisition sequence

- Discharge all IOs using HAL_TSC_IODischarge() function.
- Wait a certain time allowing a good discharge of all capacitors. This delay depends of the sampling capacitor and electrodes design.
- Select the channel IOs to be acquired using HAL_TSC_IOConfig() function.
- Launch the acquisition using either HAL_TSC_Start() or HAL_TSC_Start_IT() function. If the synchronized mode is selected, the acquisition will start as soon as the signal is received on the synchro pin.
- Wait the end of acquisition using either HAL_TSC_PollForAcquisition() or HAL_TSC_GetState() function or using WFI instruction for example.
- Check the group acquisition status using HAL_TSC_GroupGetStatus() function.
- Read the acquisition value using HAL_TSC_GroupGetValue() function.

67.2.3 Initialization and de-initialization functions

This section provides functions allowing to:

- Initialize and configure the TSC.
- De-initialize the TSC.

This section contains the following APIs:

- ***HAL_TSC_Init()***
- ***HAL_TSC_DeInit()***
- ***HAL_TSC_MspInit()***
- ***HAL_TSC_MspDeInit()***

67.2.4 IO Operation functions

This section provides functions allowing to:

- Start acquisition in polling mode.
- Start acquisition in interrupt mode.
- Stop conversion in polling mode.
- Stop conversion in interrupt mode.

- Poll for acquisition completed.
- Get group acquisition status.
- Get group acquisition value.

This section contains the following APIs:

- ***HAL_TSC_Start()***
- ***HAL_TSC_Start_IT()***
- ***HAL_TSC_Stop()***
- ***HAL_TSC_Stop_IT()***
- ***HAL_TSC_PollForAcquisition()***
- ***HAL_TSC_GroupGetStatus()***
- ***HAL_TSC_GroupGetValue()***

67.2.5 Peripheral Control functions

This section provides functions allowing to:

- Configure TSC IOs
- Discharge TSC IOs

This section contains the following APIs:

- ***HAL_TSC_IOConfig()***
- ***HAL_TSC_IODischarge()***

67.2.6 State and Errors functions

This subsection provides functions allowing to

- Get TSC state.

This section contains the following APIs:

- ***HAL_TSC_GetState()***

67.2.7 Detailed description of functions

HAL_TSC_Init

Function name	**HAL_StatusTypeDef HAL_TSC_Init (TSC_HandleTypeDef * htsc)**
Function description	Initialize the TSC peripheral according to the specified parameters in the TSC_InitTypeDef structure and initialize the associated handle.
Parameters	• **htsc:** TSC handle
Return values	• **HAL:** status

HAL_TSC_DeInit

Function name	**HAL_StatusTypeDef HAL_TSC_DeInit (TSC_HandleTypeDef * htsc)**
Function description	Deinitialize the TSC peripheral registers to their default reset values.
Parameters	• **htsc:** TSC handle

Return values
- **HAL:** status

HAL_TSC_MspInit

Function name **void HAL_TSC_MspInit (TSC_HandleTypeDef * htsc)**

Function description Initialize the TSC MSP.

Parameters
- **htsc:** pointer to a TSC_HandleTypeDef structure that contains the configuration information for the specified TSC.

Return values
- **None:**

HAL_TSC_MspDeInit

Function name **void HAL_TSC_MspDeInit (TSC_HandleTypeDef * htsc)**

Function description DeInitialize the TSC MSP.

Parameters
- **htsc:** pointer to a TSC_HandleTypeDef structure that contains the configuration information for the specified TSC.

Return values
- **None:**

HAL_TSC_Start

Function name **HAL_StatusTypeDef HAL_TSC_Start (TSC_HandleTypeDef * htsc)**

Function description Start the acquisition.

Parameters
- **htsc:** pointer to a TSC_HandleTypeDef structure that contains the configuration information for the specified TSC.

Return values
- **HAL:** status

HAL_TSC_Start_IT

Function name **HAL_StatusTypeDef HAL_TSC_Start_IT (TSC_HandleTypeDef * htsc)**

Function description Start the acquisition in interrupt mode.

Parameters
- **htsc:** pointer to a TSC_HandleTypeDef structure that contains the configuration information for the specified TSC.

Return values
- **HAL:** status.

HAL_TSC_Stop

Function name **HAL_StatusTypeDef HAL_TSC_Stop (TSC_HandleTypeDef * htsc)**

Function description Stop the acquisition previously launched in polling mode.

Parameters
- **htsc:** pointer to a TSC_HandleTypeDef structure that contains the configuration information for the specified TSC.

Return values
- **HAL:** status

HAL_TSC_Stop_IT

Function name **HAL_StatusTypeDef HAL_TSC_Stop_IT (TSC_HandleTypeDef * htsc)**

Function description Stop the acquisition previously launched in interrupt mode.

Parameters
- **htsc:** pointer to a TSC_HandleTypeDef structure that contains the configuration information for the specified TSC.

Return values
- **HAL:** status

HAL_TSC_PollForAcquisition

Function name **HAL_StatusTypeDef HAL_TSC_PollForAcquisition (TSC_HandleTypeDef * htsc)**

Function description Start acquisition and wait until completion.

Parameters
- **htsc:** pointer to a TSC_HandleTypeDef structure that contains the configuration information for the specified TSC.

Return values
- **HAL:** state

Notes
- There is no need of a timeout parameter as the max count error is already managed by the TSC peripheral.

HAL_TSC_GroupGetStatus

Function name **TSC_GroupStatusTypeDef HAL_TSC_GroupGetStatus (TSC_HandleTypeDef * htsc, uint32_t gx_index)**

Function description Get the acquisition status for a group.

Parameters
- **htsc:** pointer to a TSC_HandleTypeDef structure that contains the configuration information for the specified TSC.
- **gx_index:** Index of the group

Return values
- **Group:** status

HAL_TSC_GroupGetValue

Function name **uint32_t HAL_TSC_GroupGetValue (TSC_HandleTypeDef * htsc, uint32_t gx_index)**

Function description Get the acquisition measure for a group.

Parameters
- **htsc:** pointer to a TSC_HandleTypeDef structure that contains the configuration information for the specified TSC.
- **gx_index:** Index of the group

Return values
- **Acquisition:** measure

HAL_TSC_IOConfig

Function name **HAL_StatusTypeDef HAL_TSC_IOConfig (TSC_HandleTypeDef * htsc, TSC_IOConfigTypeDef * config)**

Function description Configure TSC IOs.

Parameters
- **htsc:** pointer to a TSC_HandleTypeDef structure that contains the configuration information for the specified TSC.

- **config:** pointer to the configuration structure.

Return values
- **HAL:** status

HAL_TSC_IODischarge

Function name **HAL_StatusTypeDef HAL_TSC_IODischarge (TSC_HandleTypeDef * htsc, uint32_t choice)**

Function description Discharge TSC IOs.

Parameters
- **htsc:** pointer to a TSC_HandleTypeDef structure that contains the configuration information for the specified TSC.
- **choice:** enable or disable

Return values
- **HAL:** status

HAL_TSC_GetState

Function name **HAL_TSC_StateTypeDef HAL_TSC_GetState (TSC_HandleTypeDef * htsc)**

Function description Return the TSC handle state.

Parameters
- **htsc:** pointer to a TSC_HandleTypeDef structure that contains the configuration information for the specified TSC.

Return values
- **HAL:** state

HAL_TSC_IRQHandler

Function name **void HAL_TSC_IRQHandler (TSC_HandleTypeDef * htsc)**

Function description Handle TSC interrupt request.

Parameters
- **htsc:** pointer to a TSC_HandleTypeDef structure that contains the configuration information for the specified TSC.

Return values
- **None:**

HAL_TSC_ConvCpltCallback

Function name **void HAL_TSC_ConvCpltCallback (TSC_HandleTypeDef * htsc)**

Function description Acquisition completed callback in non-blocking mode.

Parameters
- **htsc:** pointer to a TSC_HandleTypeDef structure that contains the configuration information for the specified TSC.

Return values
- **None:**

HAL_TSC_ErrorCallback

Function name **void HAL_TSC_ErrorCallback (TSC_HandleTypeDef * htsc)**

Function description Error callback in non-blocking mode.

Parameters
- **htsc:** pointer to a TSC_HandleTypeDef structure that contains the configuration information for the specified TSC.

Return values
- **None:**

67.3 TSC Firmware driver defines

67.3.1 TSC

Acquisition Mode

TSC_ACQ_MODE_NORMAL

TSC_ACQ_MODE_SYNCHRO

CTPulse High Length

TSC_CTPH_1CYCLE

TSC_CTPH_2CYCLES

TSC_CTPH_3CYCLES

TSC_CTPH_4CYCLES

TSC_CTPH_5CYCLES

TSC_CTPH_6CYCLES

TSC_CTPH_7CYCLES

TSC_CTPH_8CYCLES

TSC_CTPH_9CYCLES

TSC_CTPH_10CYCLES

TSC_CTPH_11CYCLES

TSC_CTPH_12CYCLES

TSC_CTPH_13CYCLES

TSC_CTPH_14CYCLES

TSC_CTPH_15CYCLES

TSC_CTPH_16CYCLES

CTPulse Low Length

TSC_CTPL_1CYCLE

TSC_CTPL_2CYCLES

TSC_CTPL_3CYCLES

TSC_CTPL_4CYCLES

TSC_CTPL_5CYCLES

TSC_CTPL_6CYCLES

TSC_CTPL_7CYCLES

TSC_CTPL_8CYCLES

TSC_CTPL_9CYCLES

TSC_CTPL_10CYCLES

TSC_CTPL_11CYCLES

TSC_CTPL_12CYCLES

TSC_CTPL_13CYCLES

TSC_CTPL_14CYCLES

TSC_CTPL_15CYCLES

TSC_CTPL_16CYCLES

TSC Exported Macros

__HAL_TSC_RESET_HANDLE_STATE

Description:

- Reset TSC handle state.

Parameters:

- __HANDLE__: TSC handle

Return value:

- None

__HAL_TSC_ENABLE

Description:

- Enable the TSC peripheral.

Parameters:

- __HANDLE__: TSC handle

Return value:

- None

__HAL_TSC_DISABLE

Description:

- Disable the TSC peripheral.

Parameters:

- __HANDLE__: TSC handle

Return value:

- None

__HAL_TSC_START_ACQ

Description:

- Start acquisition.

Parameters:

- __HANDLE__: TSC handle

Return value:

- None

__HAL_TSC_STOP_ACQ

Description:

- Stop acquisition.

Parameters:

- __HANDLE__: TSC handle

Return value:

- None

__HAL_TSC_SET_IODEF_OUTPPLOW

Description:

- Set IO default mode to output push-pull low.

Parameters:

- __HANDLE__: TSC handle

Return value:

- None

__HAL_TSC_SET_IODEF_INFLOAT

Description:

- Set IO default mode to input floating.

Parameters:

- __HANDLE__: TSC handle

Return value:

- None

__HAL_TSC_SET_SYNC_POL_FALL

Description:

- Set synchronization polarity to falling edge.

Parameters:

- __HANDLE__: TSC handle

Return value:

- None

__HAL_TSC_SET_SYNC_POL_RISE_HIGH

Description:

- Set synchronization polarity to rising edge and high level.

Parameters:

- __HANDLE__: TSC handle

Return value:

- None

__HAL_TSC_ENABLE_IT

Description:

- Enable TSC interrupt.

Parameters:

- __HANDLE__: TSC handle
- __INTERRUPT__: TSC interrupt

Return value:

- None

__HAL_TSC_DISABLE_IT

Description:

- Disable TSC interrupt.

Parameters:

- __HANDLE__: TSC handle
- __INTERRUPT__: TSC interrupt

Return value:

- None

__HAL_TSC_GET_IT_SOURCE

Description:

- Check whether the specified TSC interrupt source is enabled or not.

Parameters:

- __HANDLE__: TSC Handle
- __INTERRUPT__: TSC interrupt

Return value:

- SET: or RESET

__HAL_TSC_GET_FLAG

Description:

- Check whether the specified TSC flag is set or not.

Parameters:

- __HANDLE__: TSC handle
- __FLAG__: TSC flag

Return value:

- SET: or RESET

__HAL_TSC_CLEAR_FLAG

Description:

- Clear the TSC's pending flag.

Parameters:

- __HANDLE__: TSC handle
- __FLAG__: TSC flag

Return value:

- None

__HAL_TSC_ENABLE_HYSTERESIS

Description:

- Enable schmitt trigger hysteresis on a group of IOs.

Parameters:

- __HANDLE__: TSC handle
- __GX_IOY_MASK__: IOs mask

Return value:

- None

__HAL_TSC_DISABLE_HYSTERESIS

Description:

- Disable schmitt trigger hysteresis on a group of IOs.

Parameters:

- __HANDLE__: TSC handle
- __GX_IOY_MASK__: IOs mask

Return value:

- None

__HAL_TSC_OPEN_ANALOG_SWITCH

Description:

- Open analog switch on a group of IOs.

Parameters:

- __HANDLE__: TSC handle
- __GX_IOY_MASK__: IOs mask

Return value:

- None

__HAL_TSC_CLOSE_ANALOG_SWITCH

Description:

- Close analog switch on a group of IOs.

Parameters:

- __HANDLE__: TSC handle
- __GX_IOY_MASK__: IOs mask

Return value:

- None

__HAL_TSC_ENABLE_CHANNEL

Description:

- Enable a group of IOs in channel mode.

Parameters:

- __HANDLE__: TSC handle
- __GX_IOY_MASK__: IOs mask

Return value:

- None

__HAL_TSC_DISABLE_CHANNEL

Description:

- Disable a group of channel IOs.

Parameters:

- __HANDLE__: TSC handle
- __GX_IOY_MASK__: IOs mask

Return value:

- None

__HAL_TSC_ENABLE_SAMPLING

Description:

- Enable a group of IOs in sampling mode.

Parameters:

- __HANDLE__: TSC handle

- __GX_IOY_MASK__: IOs mask

Return value:

- None

__HAL_TSC_DISABLE_SAMPLING

Description:

- Disable a group of sampling IOs.

Parameters:

- __HANDLE__: TSC handle
- __GX_IOY_MASK__: IOs mask

Return value:

- None

__HAL_TSC_ENABLE_GROUP

Description:

- Enable acquisition groups.

Parameters:

- __HANDLE__: TSC handle
- __GX_MASK__: Groups mask

Return value:

- None

__HAL_TSC_DISABLE_GROUP

Description:

- Disable acquisition groups.

Parameters:

- __HANDLE__: TSC handle
- __GX_MASK__: Groups mask

Return value:

- None

__HAL_TSC_GET_GROUP_STATUS

Description:

- Gets acquisition group status.

Parameters:

- __HANDLE__: TSC Handle
- __GX_INDEX__: Group index

Return value:

- SET: or RESET

Flags definition

TSC_FLAG_EOA

TSC_FLAG_MCE

Group definition

TSC_NB_OF_GROUPS

TSC_GROUP1

TSC_GROUP2
TSC_GROUP3
TSC_GROUP4
TSC_GROUP5
TSC_GROUP6
TSC_GROUP7
TSC_GROUP8
TSC_ALL_GROUPS
TSC_GROUP1_IDX
TSC_GROUP2_IDX
TSC_GROUP3_IDX
TSC_GROUP4_IDX
TSC_GROUP5_IDX
TSC_GROUP6_IDX
TSC_GROUP7_IDX
TSC_GROUP8_IDX
TSC_GROUP1_IO1
TSC_GROUP1_IO2
TSC_GROUP1_IO3
TSC_GROUP1_IO4
TSC_GROUP1_ALL_IOS
TSC_GROUP2_IO1
TSC_GROUP2_IO2
TSC_GROUP2_IO3
TSC_GROUP2_IO4
TSC_GROUP2_ALL_IOS
TSC_GROUP3_IO1
TSC_GROUP3_IO2
TSC_GROUP3_IO3
TSC_GROUP3_IO4
TSC_GROUP3_ALL_IOS
TSC_GROUP4_IO1
TSC_GROUP4_IO2
TSC_GROUP4_IO3
TSC_GROUP4_IO4
TSC_GROUP4_ALL_IOS

TSC_GROUP5_IO1
TSC_GROUP5_IO2
TSC_GROUP5_IO3
TSC_GROUP5_IO4
TSC_GROUP5_ALL_IOS
TSC_GROUP6_IO1
TSC_GROUP6_IO2
TSC_GROUP6_IO3
TSC_GROUP6_IO4
TSC_GROUP6_ALL_IOS
TSC_GROUP7_IO1
TSC_GROUP7_IO2
TSC_GROUP7_IO3
TSC_GROUP7_IO4
TSC_GROUP7_ALL_IOS
TSC_GROUP8_IO1
TSC_GROUP8_IO2
TSC_GROUP8_IO3
TSC_GROUP8_IO4
TSC_GROUP8_ALL_IOS
TSC_ALL_GROUPS_ALL_IOS

Interrupts definition

TSC_IT_EOA
TSC_IT_MCE

IO Default Mode

TSC_IODEF_OUT_PP_LOW
TSC_IODEF_IN_FLOAT

IO Mode

TSC_IOMODE_UNUSED
TSC_IOMODE_CHANNEL
TSC_IOMODE_SHIELD
TSC_IOMODE_SAMPLING

Max Count Value

TSC_MCV_255
TSC_MCV_511
TSC_MCV_1023

TSC_MCV_2047

TSC_MCV_4095

TSC_MCV_8191

TSC_MCV_16383

Pulse Generator Prescaler

TSC_PG_PRESC_DIV1

TSC_PG_PRESC_DIV2

TSC_PG_PRESC_DIV4

TSC_PG_PRESC_DIV8

TSC_PG_PRESC_DIV16

TSC_PG_PRESC_DIV32

TSC_PG_PRESC_DIV64

TSC_PG_PRESC_DIV128

Spread Spectrum Prescaler

TSC_SS_PRESC_DIV1

TSC_SS_PRESC_DIV2

Synchro Pin Polarity

TSC_SYNC_POLARITY_FALLING

TSC_SYNC_POLARITY_RISING

68 HAL UART Generic Driver

68.1 UART Firmware driver registers structures

68.1.1 UART_InitTypeDef

Data Fields

- ***uint32_t BaudRate***
- ***uint32_t WordLength***
- ***uint32_t StopBits***
- ***uint32_t Parity***
- ***uint32_t Mode***
- ***uint32_t HwFlowCtl***
- ***uint32_t OverSampling***
- ***uint32_t OneBitSampling***
- ***uint32_t ClockPrescaler***

Field Documentation

- ***uint32_t UART_InitTypeDef::BaudRate***
  This member configures the UART communication baud rate. The baud rate register is computed using the following formula: UART: =====If oversampling is 16 or in LIN mode, Baud Rate Register = ((uart_ker_ckpres) / ((huart->Init.BaudRate)))If oversampling is 8, Baud Rate Register[15:4] = ((2 * uart_ker_ckpres) / ((huart->Init.BaudRate)))[15:4] Baud Rate Register[3] = 0 Baud Rate Register[2:0] = (((2 * uart_ker_ckpres) / ((huart->Init.BaudRate)))[3:0]) >> 1 LPUART: ======= Baud Rate Register = ((256 * lpuart_ker_ckpres) / ((huart->Init.BaudRate))) where (uart/lpuart)_ker_ck_pres is the UART input clock divided by a prescaler
- ***uint32_t UART_InitTypeDef::WordLength***
  Specifies the number of data bits transmitted or received in a frame. This parameter can be a value of ***UARTEx_Word_Length***.
- ***uint32_t UART_InitTypeDef::StopBits***
  Specifies the number of stop bits transmitted. This parameter can be a value of ***UART_Stop_Bits***.
- ***uint32_t UART_InitTypeDef::Parity***
  Specifies the parity mode. This parameter can be a value of ***UART_Parity***
  Note: When parity is enabled, the computed parity is inserted at the MSB position of the transmitted data (9th bit when the word length is set to 9 data bits; 8th bit when the word length is set to 8 data bits).
- ***uint32_t UART_InitTypeDef::Mode***
  Specifies whether the Receive or Transmit mode is enabled or disabled. This parameter can be a value of ***UART_Mode***.
- ***uint32_t UART_InitTypeDef::HwFlowCtl***
  Specifies whether the hardware flow control mode is enabled or disabled. This parameter can be a value of ***UART_Hardware_Flow_Control***.
- ***uint32_t UART_InitTypeDef::OverSampling***
  Specifies whether the Over sampling 8 is enabled or disabled, to achieve higher speed (up to f_PCLK/8). This parameter can be a value of ***UART_Over_Sampling***.
- ***uint32_t UART_InitTypeDef::OneBitSampling***
  Specifies whether a single sample or three samples' majority vote is selected. Selecting the single sample method increases the receiver tolerance to clock deviations. This parameter can be a value of ***UART_OneBit_Sampling***.