Detailed Design of UART Driver

Team X

Contents

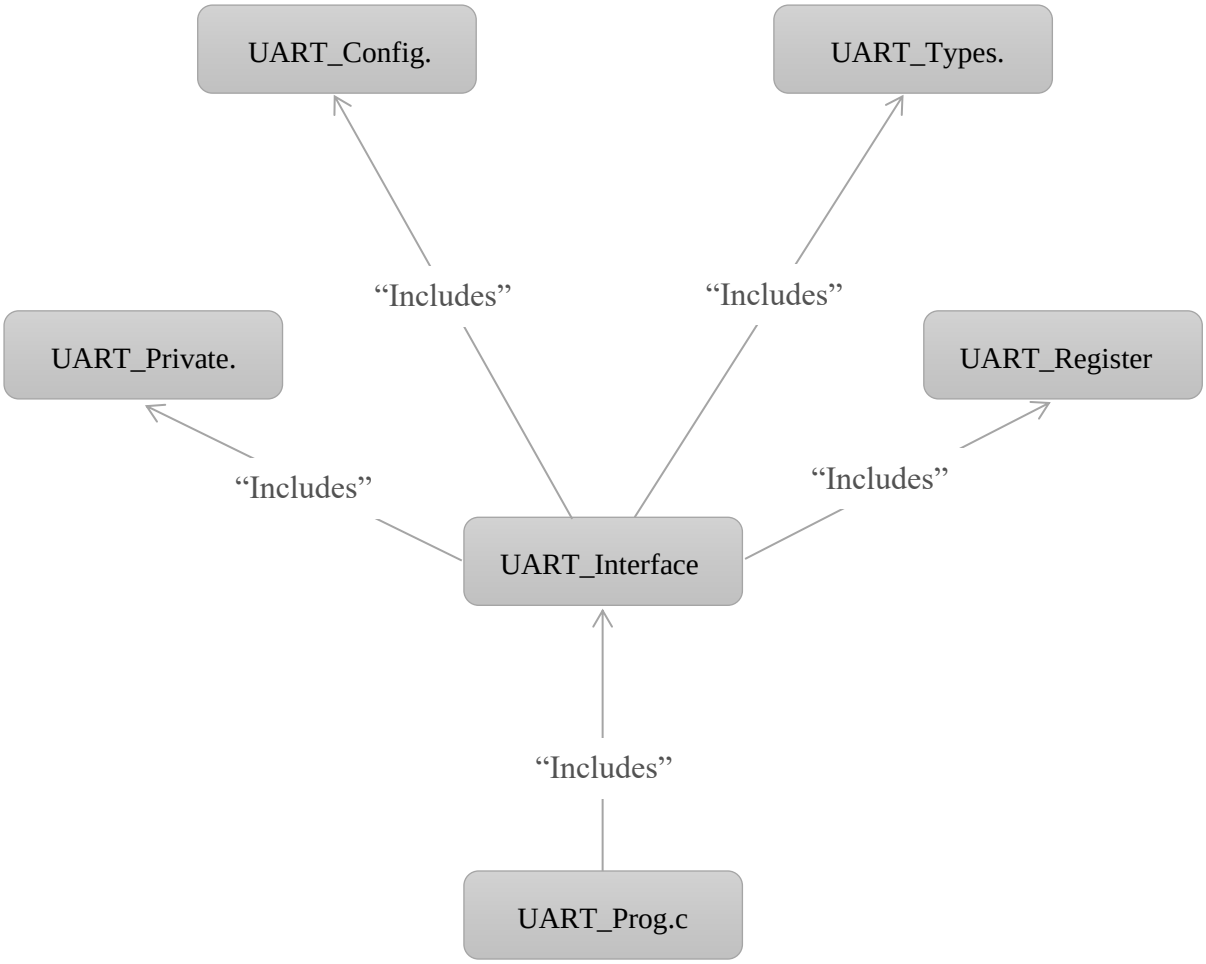
Introduction and functional overview:

This document specifies detailed design of the module UART Driver.

The UART driver initializes, sends character, number or string and Receive character, number or string.

File structure:

The UART module shall comply with the following file structure:



Requirement's traceability:

Requirement	Description	Satisfied by
[SRS_UART_3301]	The UART driver shall provide a service that that initializes the UART.	[DD_UART_3305]
[SRS_UART_3302]	The UART driver shall provide a service that transmit character or number to another MCU	[DD_UART_3306]
[SRS_UART_3303]	The UART driver shall provide a service that send a string to another MCU.	DD_UART_3308]
[SRS_UART_3304]	The UART driver shall provide a service that receive character, number or string from another MCU.	[DD_UART_3310]
[SRS_UART_3305]	The UART driver shall provide a service that set call back function	[DD_UART_3312] [DD_UART_3301] [DD_UART_3302] [DD_UART_3303] [DD_UART_3304]
[SRS_UART_3306]	The UART driver shall provide a service that enable transmit complete interrupt.	[DD_UART_3307]
[SRS_UART_3307]	The UART driver shall provide a service that enable Enable USART data register empty	[DD_UART_3309]
[SRS_UART_3308]	The UART driver shall provide a service that enable receive complete	[DD_UART_33011

API specification:

Type definitions:

1- [DD_UART_3301]

Name:	(*ptr2fun_Data_register_empty)
Type	static volatile void
Range:	-----
Description:	A pointer to a function that doesn't take argument sand returns void
Covered requirements:	[SRS_UART_3305]

2- [DD_UART_3302]

Name:	(*ptr2fun_transmit_complete)
Type	static volatile void
Range:	-----

Description:	A pointer to a function that doesn't take argument sand returns void
Covered requirements:	[SRS_UART_3305]

3- [DD_UART_3303]

Name:	(*ptr2fun_receive_data)
Type	static volatile void
Range:	-----
Description:	A pointer to a function that doesn't take argument sand returns void
Covered requirements:	[SRS_UART_3305]

4- [DD_UART_3304]

Name:	MODES
Type	enum
Range:	-----
Description:	Type helps me when select mode of UART these modes are Data_Register_Empty ,Transmit_Complete and receive_complete.
Covered requirements:	[SRS_UART_3305]

Function definitions:

1- [DD_UART_3305]

Service name:	UART initialization
Syntax:	E_Errore_State_UART MCAL_UART_init(uint32 baudRate)
Sync/Async:	ASynchronous
Re-entrancy:	Re-entrant
Parameters (in):	baud_rate
Parameters (out):	none
Parameters (inout):	none
Return type:	E_Errore_State_UART
Description:	Function that initializes the UART
Covered requirements:	[SRS_UART_3301]

2- [DD_UART_3306]

Service name:	UART transmit character or number.
Syntax:	E_Errore_State_UART MCAL_UART_send_sysch(uint8 Data)
Sync/Async:	Synchronous
Re-entrancy:	Re-entrant
Parameters (in):	Data | data is sent by UART.
Parameters (out):	none
Parameters (inout):	none
Return type:	E_Errore_State_UART
Description:	Function that sends character or number to another MCU
Covered requirements:	[SRS_UART_3302]

3- [DD_UART_3307]

Service name:	Enable Transmit complete interrupt
Syntax:	E_Errore_State_UART MCAL_UART_TXC_Enable()
Sync/Async:	ASynchronous
Re-entrancy:	Re-entrant
Parameters (in):	Data | data is sent by UART.
Parameters (out):	none
Parameters (inout):	none
Return type:	E_Errore_State_UART
Description:	Enable Transmit complete interrupt TXCIE
Covered requirements:	[SRS_UART_3306]

4- [DD_UART_3308]

Service name:	UART send string
Syntax:	E_Errore_State_UART MCAL_UART_send_str_sysch(char*Str)
Sync/Async:	Synchronous
Re-entrancy:	Re-entrant
Parameters (in):	none
Parameters (out):	none
Parameters (inout):	*Str … String is sent by UART
Return type:	E_Errore_State_UART

Description:	Function that sends string to another MCU
Covered requirements:	[SRS_UART_3303]

5- [DD_UART_3309]

Service name:	Enable USART data register empty interrupt
Syntax:	E_Errore_State_UART MCAL_UART_UDRE_Enable()
Sync/Async:	ASynchronous
Re-entrancy:	Re-entrant
Parameters (in):	none
Parameters (out):	none
Parameters (inout):	none
Return type:	E_Errore_State_UART
Description:	Enable USART data register empty interrupt UDRIE
Covered requirements:	[SRS_UART_3307]

6- [DD_UART_3310]

Service name:	UART receive data
Syntax:	uint8 MCAL_UART_receive_sysch()
Sync/Async:	Synchronous
Re-entrancy:	Re-entrant
Parameters (in):	none
Parameters (out):	none
Parameters (inout):	Function return uint8 | This is data received from another MCU.
Return type:	uint8.
Description:	Function that receives data from another MCU.
Covered requirements:	[SRS_UART_3304]

7- [DD_UART_3311]

Service name:	Enable USART receive complete
Syntax:	E_Errore_State_UART MCAL_UART_RXC_Enable();
Sync/Async:	ASynchronous
Re-entrancy:	Re-entrant

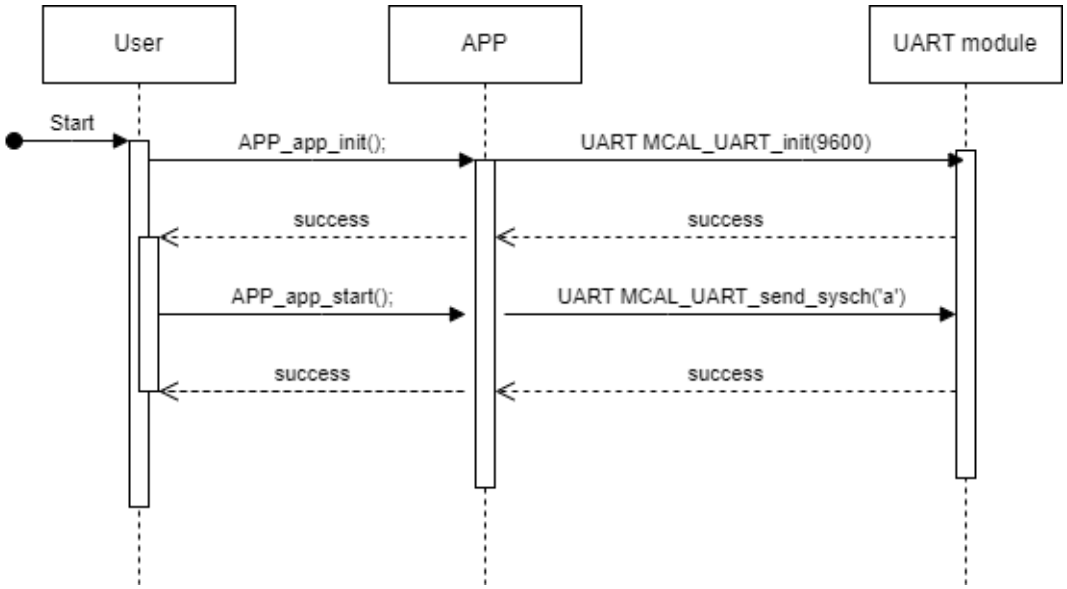
Parameters (in):	none
Parameters (out):	none
Parameters (inout):	none
Return type:	E_Errore_State_UART
Description:	Enable USART receive complete RXCIE
Covered requirements:	[SRS_UART_3308]

8- [DD_UART_3312]

Service name:	Set call back
Syntax:	E_Errore_State_UART MCAL_UART_setCAllBACK(void (*p2f)(void),MODES mode)
Sync/Async:	Synchronous
Re-entrancy:	Re-entrant
Parameters (in):	void (*p2f)(void) | pointer to function , mode| to select mode
Parameters (out):	none
Parameters (inout):	none
Return type:	E_Errore_State_UART
Description:	Function that takes address of the function and sends this function to ISR
Covered requirements:	[SRS_UART_3305]

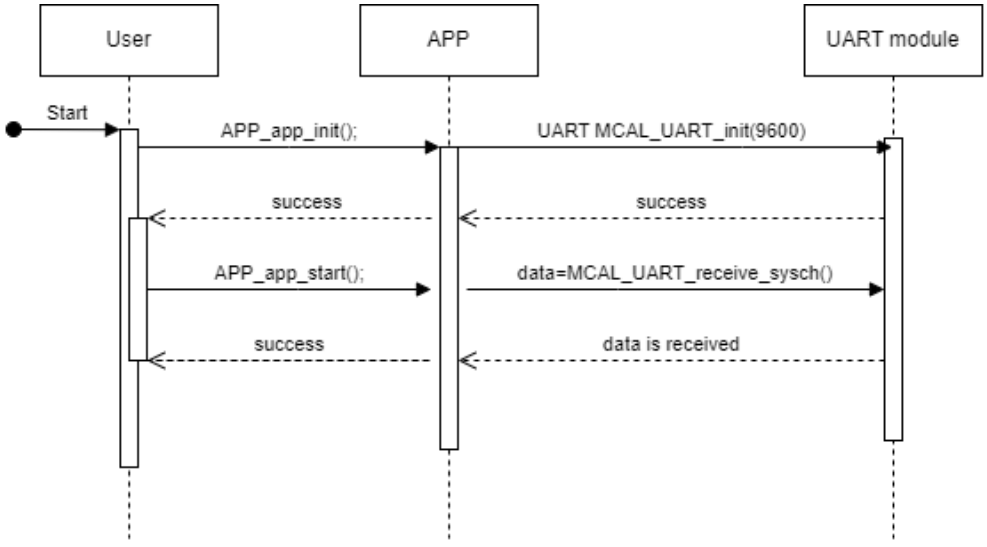
Sequence Diagram:

UART sends character a by synchronous way:



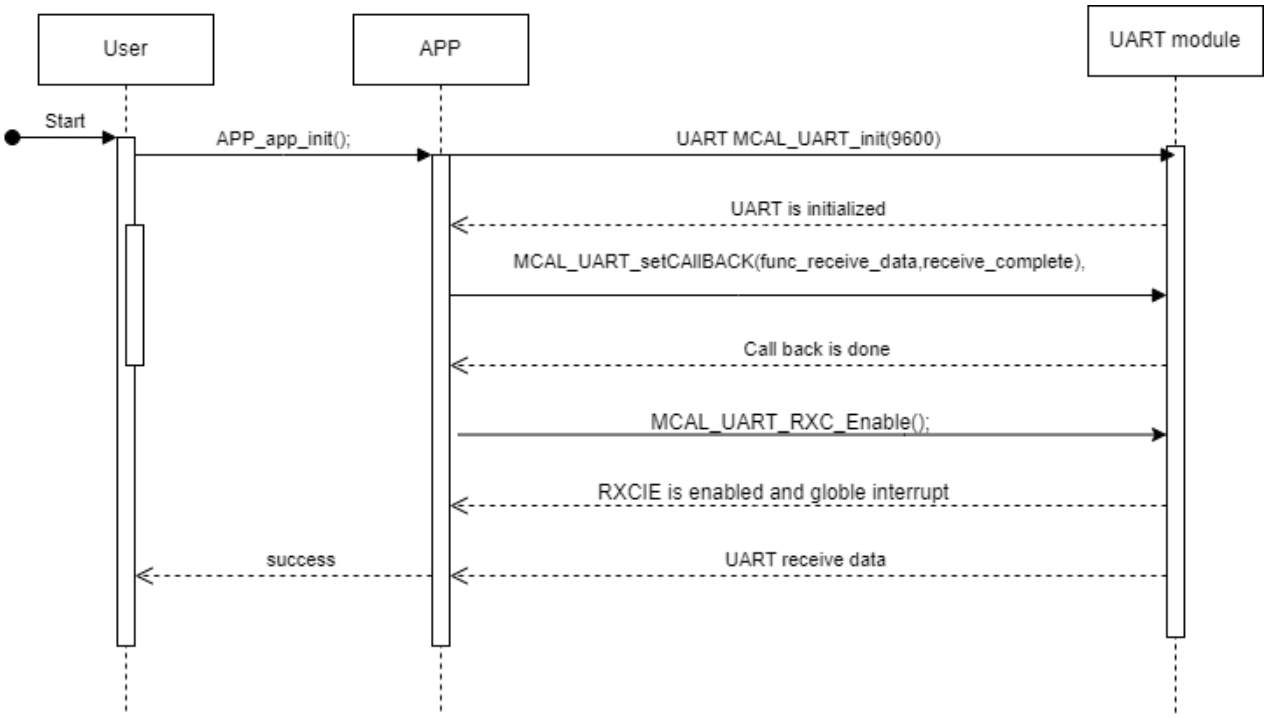
User wants app to initialize when app initialize the UART initialize. User wants the app to starts when the app starts the UART sends character a by synchronous way.

Uart receive character 'a' by synchronous way:



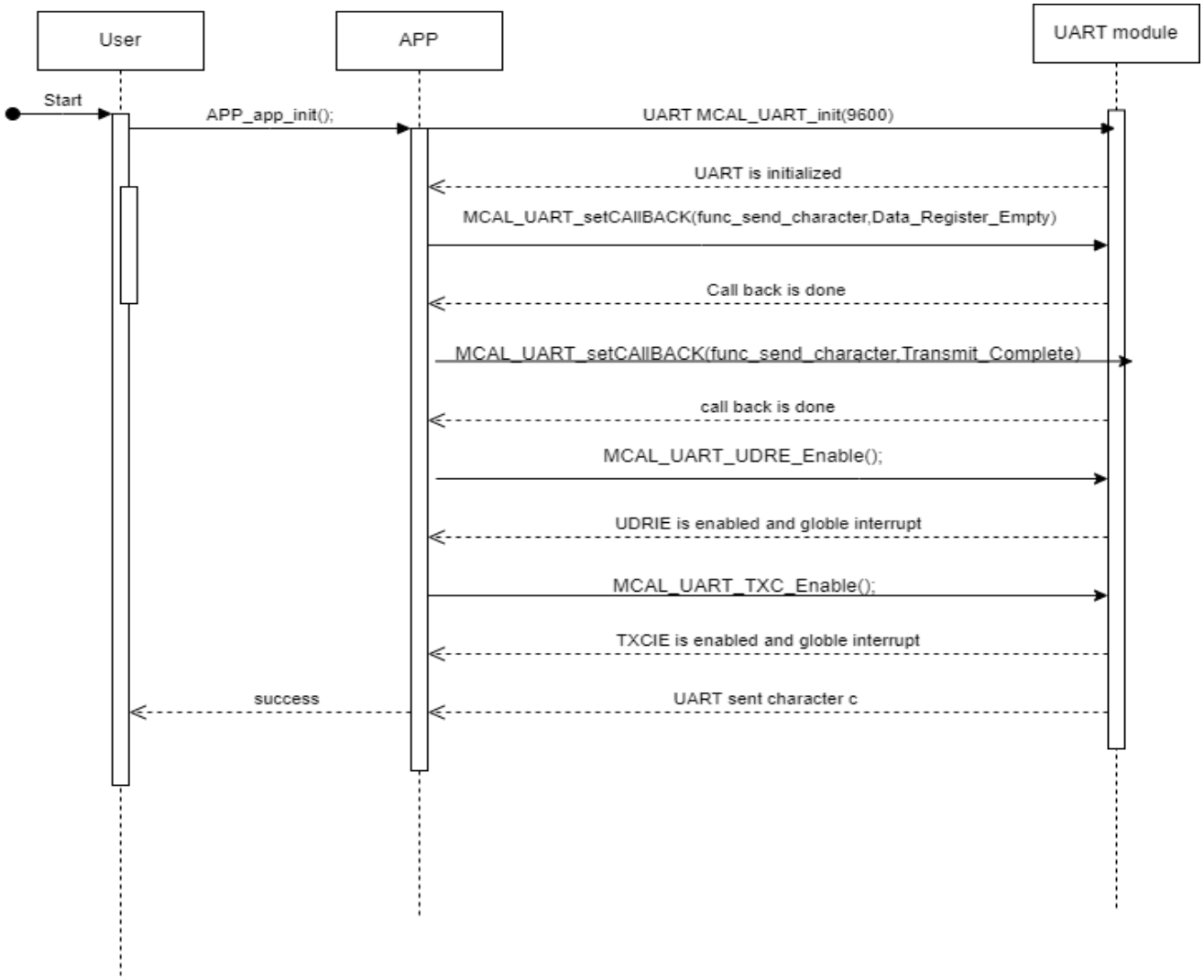
User wants app to initialize when app initialize the UART initialize. User wants the app to starts when the app starts the UART receives character a by synchronous way.

UART receive character c by asynchronous way:



User wants app to initialize when app initialize the UART initialize, function call back is set and RXCIE and global interrupt are enabled then UART will receive data.

UART sends character c by asynchronous way:



User wants app to initialize when app initialize the UART initialize, function call back is set and UDRIE and TXCIE and global interrupt are enabled then UART will send character c.