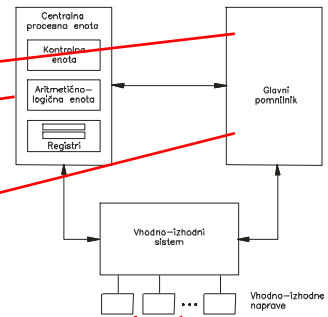
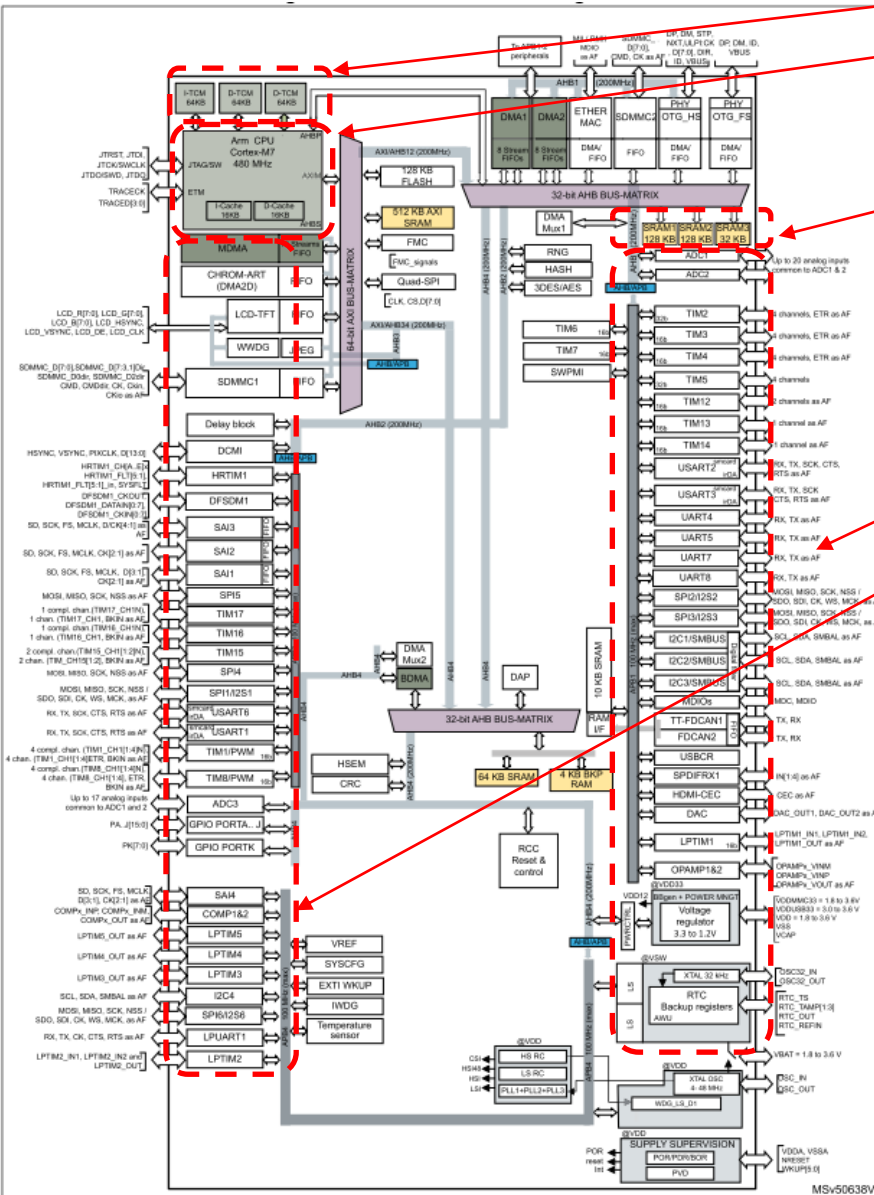


# STM32H7

*Vhodno / izhodne naprave*

*USART Serijska komunikacija*

# STM32H750XB



# Delo na STM32H7 razvojnem sistemu

## Priključitev :

- **Mikro USB** prikllop na **daljši stranici** (nad LCD, srednji !!!)

## Poseben začetni projekt (github) in info za STM32H7 (e-učilnica):

- **dodajanje vsebine (Main.s):**



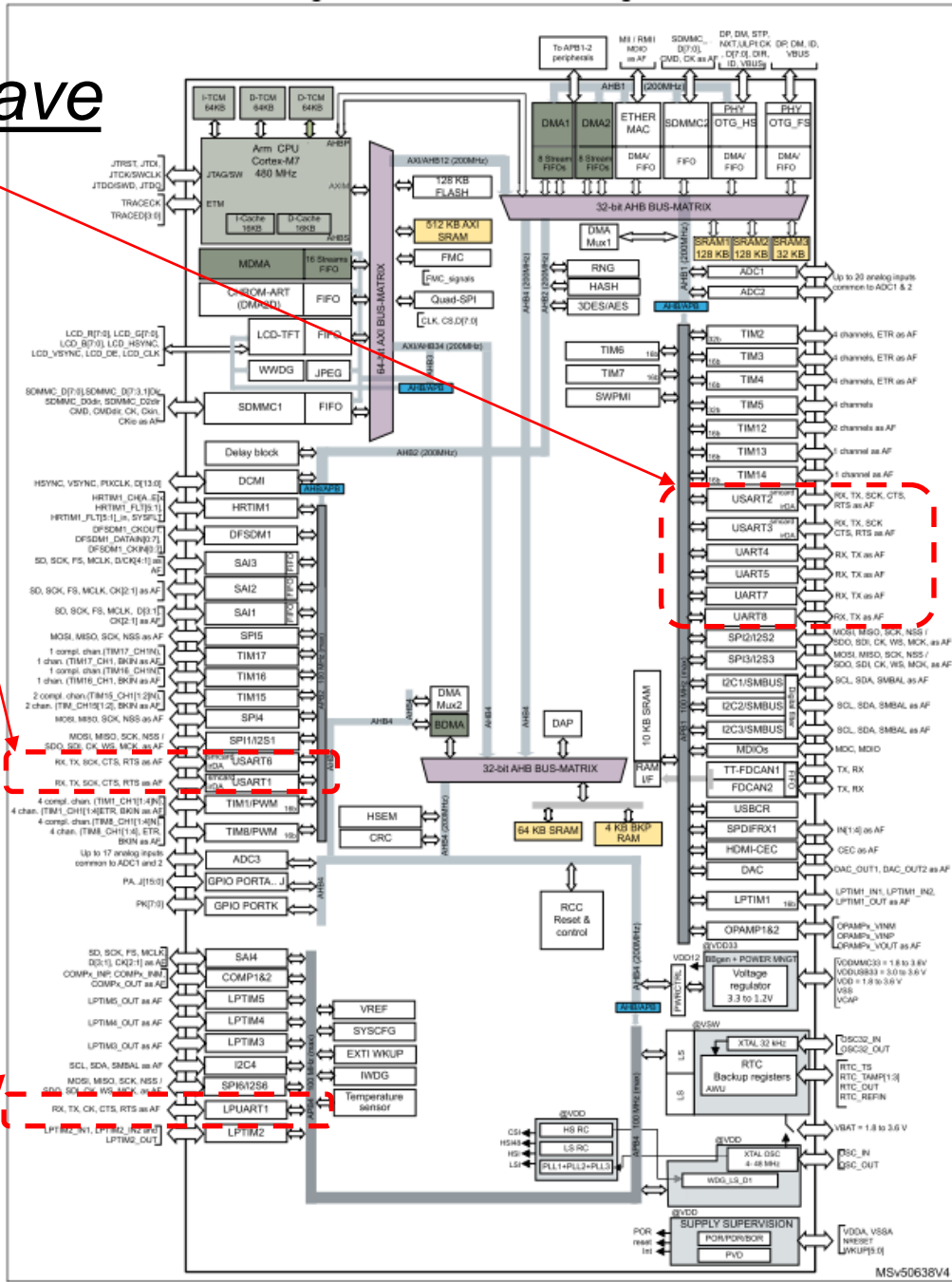
```
CubelDEWorkspace - stm32h7-asm/Core/Src/Main.s - STM32CubelDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer x
CubelDE_Workspace
  stm32f4-asm-qemu
  Delo
    ARM9Template
    stm32f4-asm (in STM32AsmTemplate)
    ARM9Template.zip
    Node_V4 (in node_v4)
    Sluzba
      CAN_IEX_Module
      ORLab-STM32H7
      stm32h7-asm
        Binaries
        Includes
        Core
          Src
            Main.s
          Startup
            startup_stm32h750xbhx.s
        Debug
        out
        makefile
        README.md
        STM32H750X.svd
        STM32H750XBHX_FLASH.ld
        STM32H750XBHX_RAM.ld
        README.md
      RALab-STM32H7
        stm32h7-asm_RA_LED
        README.md
      STM32_USB_Key_AdvDebug
      STM32_USB_Key_FreeRTOS_AdvDebug
      STM32CubelDE_Adv_Debug
      STM32F4_Discovery_VIN_Projects

Main.s x startup_stm32h750xbhx.s
12
13 ////////////////////////////////////////////////////////////////////
14 // Definitions
15 ////////////////////////////////////////////////////////////////////
16 // Definitions section. Define all the registers and
17 // constants here for code readability.
18
19 // Constants
20
21
22 // Start of data section|
23     .data
24
25     .align
26
27 STEV1: .word  0x10 // 32-bitna spr.
28 STEV2: .word  0x40 // 32-bitna spr.
29 VSOTA: .word  0 // 32-bitna spr.
30
31
32 // Start of text section
33     .text
34
35     .type main, %function
36     .global main
37
38     .align
39 main:
40     ldr r0, =STEV1 // Naslov od STEV1 -> r0
41     ldr r1, [r0] // Vsebina iz naslova v r0 -> r1
42
43     ldr r0, =STEV2 // Naslov od STEV1 -> r0
44     ldr r2, [r0] // Vsebina iz naslova v r0 -> r2
45
46     add r3,r1,r2 // r1 + r2 -> r3
47
48     ldr r0, =VSOTA // Naslov od STEV1 -> r0
49     str r3,[r0] // iz registra r3 -> na naslov v r0
50
51 __end: b __end
52
```

----- Razvojni sistem STM32H750-DK -----

- STM32H750B-DK Discovery kit with STM32H750XB MCU
- ORLab-STM32H7 - GitHub repozitorij
- User Manual Discovery kit stm32h750xb Uploaded 11/11/22, 10.15
- DataSheet\_stm32h750xb Uploaded 11/11/22, 10.16
- Reference Manual rm0433-stm32h750xb Uploaded 11/11/22, 10.17
- Programming\_Manual\_pm0253-stm32h750xb Uploaded 11/11/22, 10.17
- Errata\_es0396-stm32h750xb Uploaded 11/11/22, 10.19

# U(S)ART naprave



# UART komunikacija

8-bit word length (M = 00), 1 Stop bit

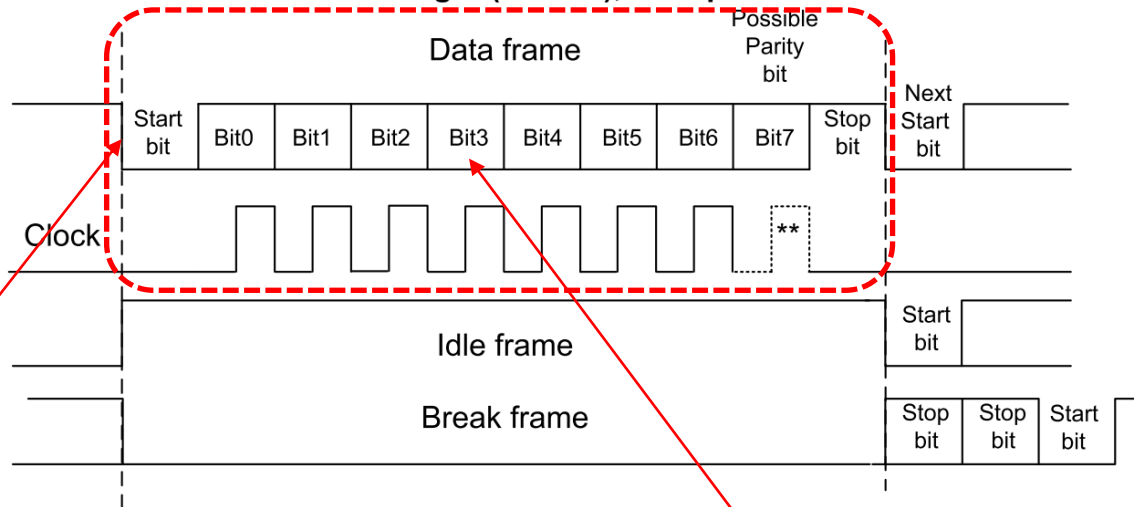


Figure 572. Start bit detection when oversampling by 16 or 8

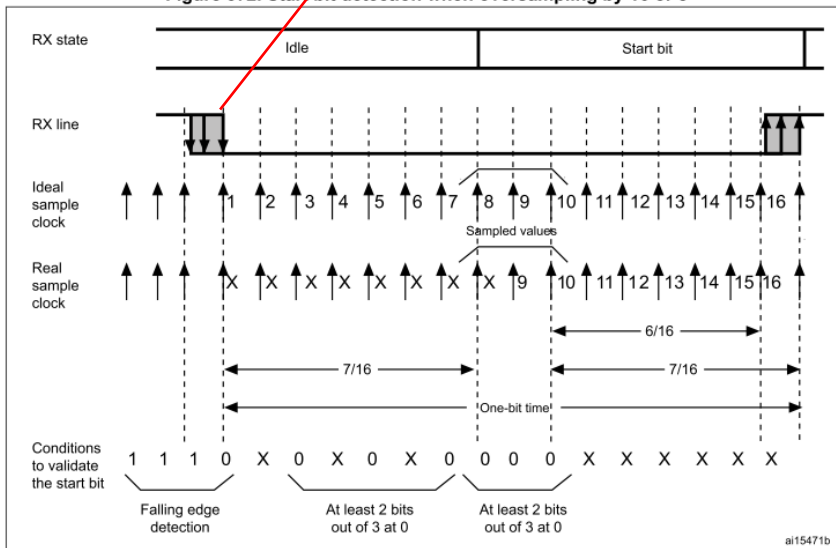
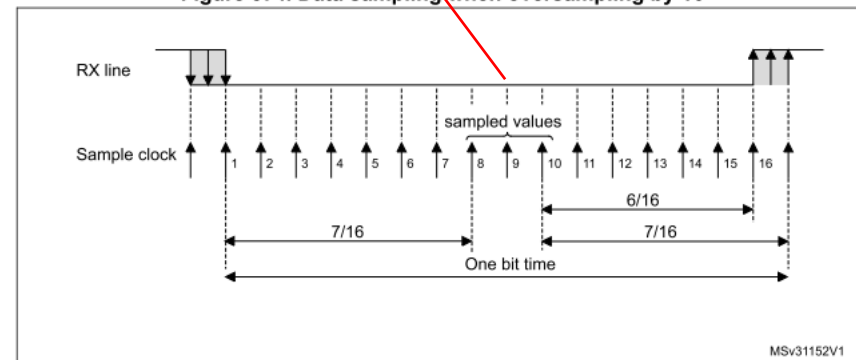


Figure 574. Data sampling when oversampling by 16



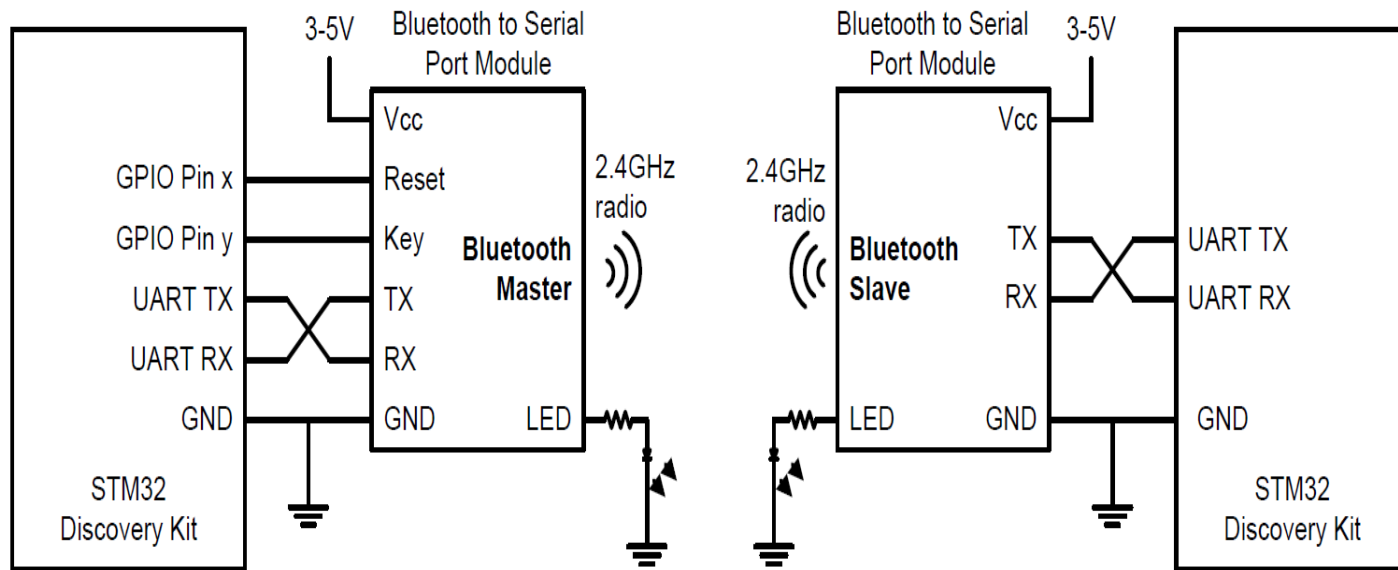
# UART – fizični nivoji (Voltage Levels)

Standard	Voltage signal	Max distance	Max speed	Number of devices supported per port
RS-232	Single end ( logic 1: +5 to +15V, logic 0: -5 to -15 V)	100 feet	115Kbit/s	1 master, 1 receiver
RS-422	Differential (-6V to +6V)	4000 feet	10Mbit/s	1 master, 10 receivers
RS-485	Differential (-7V to +12V)	4000 feet	10Mbit/s	32 masters, 32 receivers

Standards:

- How far can signal transfer?
- How fast can it transfer data?
- How many devices supported per port?
- How many masters allowed?

# UART – fizični nivoji -> Bluetooth



# Viri: User & Programming manuals, vezalna shema



UM2488



RM0433

Reference manual

User manual

STM32H742, STM32H743/753 and STM32H750 Value line advanced Arm®-based 32-bit MCUs

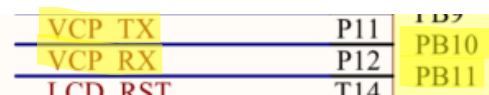
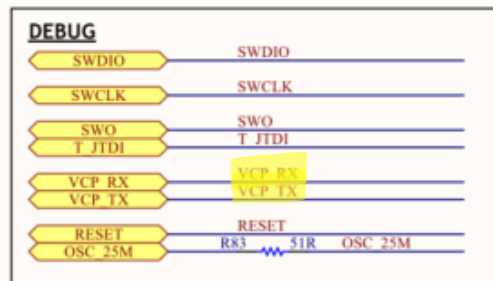
Discovery kits with STM32H745XI and STM32H750XB microcontrollers

## 6.14 Virtual COM port

The USART3 serial interface is directly available as a virtual COM port of the PC, connected to the STLINK-V3 USB connector (CN14). The virtual COM port settings as the following:

- 115200 baud
- 8-bit data
- No parity
- 1 stop bit
- no flow control

## Vezalna shema



48	Universal synchronous/asynchronous receiver transmitter (USART/UART) .....	2039
49	Low-power universal asynchronous receiver transmitter (LPUART) .....	2127

Table 8. Register boundary addresses<sup>(1)</sup> (continued)

Boundary address	Peripheral	Bus	Register map
0x40005000 - 0x400053FF	UART5	APB1 (D2)	Section 48.7: USART registers
0x40004C00 - 0x40004FFF	UART4		Section 48.7: USART registers
0x40004800 - 0x40004BFF	USART3		Section 48.7: USART registers
0x40004400 - 0x400047FF	USART2		Section 48.7: USART registers

Table 395. USART/LPUART features

USART modes/features <sup>(1)</sup>	USART1/2/3/6	UART4/5/7/8	LPUART1
Hardware flow control for modem	X	X	X
Continuous communication using DMA	X	X	X
Multiprocessor communication	X	X	X
Synchronous mode (Master/Slave)	X	-	-
Smartcard mode	X	-	-
Single-wire Half-duplex communication	X	X	X
IrDA SIR ENDEC block	X	X	-
LIN mode	X	X	-
Dual clock domain and wakeup from low-power mode	X	X	X
Receiver timeout interrupt	X	X	-
Modbus communication	X	X	-
Auto baud rate detection	X	X	-
Driver Enable	X	X	X
USART data length		7, 8 and 9 bits	
Tx/Rx FIFO	X	X	X
Tx/Rx FIFO size		16	

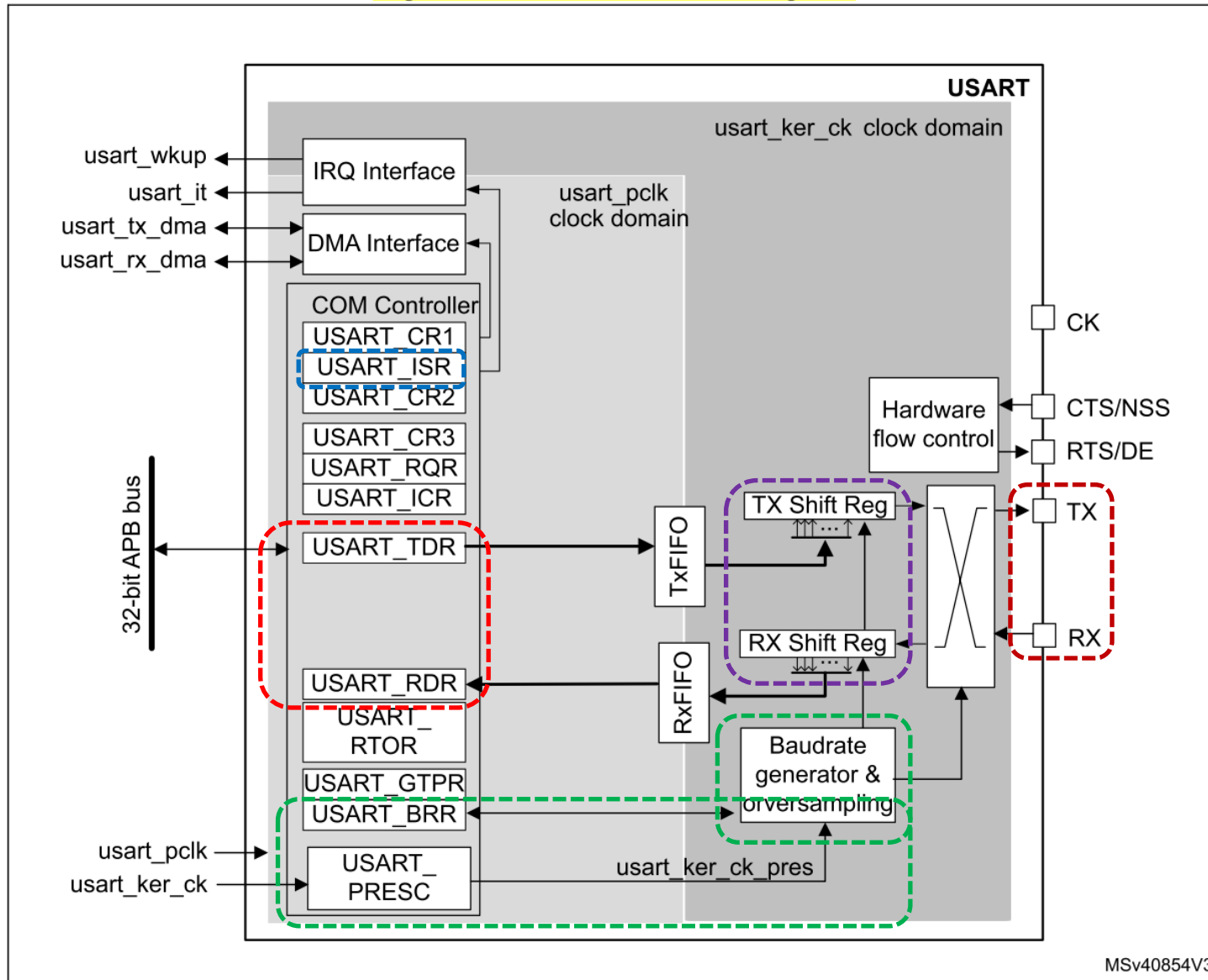
1. X = supported.

[https://st-onlinetraining.s3.amazonaws.com/STM32H7-Peripheral-USART-interface\\_%28USART%29/index.html](https://st-onlinetraining.s3.amazonaws.com/STM32H7-Peripheral-USART-interface_%28USART%29/index.html)



# USART shema

Figure 568. USART block diagram



MSv40854V3

# USART (Registri za nastavitve urinega signala – vklop naprave)

## 8.7.45 RCC APB1 Clock Register (RCC\_APB1LENR)

This register can be accessed via two different offset address.

Table 70. RCC\_APB1ENR address offset and reset value

Register Name	Address Offset	Reset Value
RCC_APB1LENR	0x0E8	0x0000 0000
RCC_C1_APB1LENR	0x148	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UART8EN	UART7EN	DAC12EN	Res.	CECEN	Res.	Res.	Res.	I2C3EN	I2C2EN	I2C1EN	UART5EN	UART4EN	USART3EN	USART2EN	SPDIFRXEN
r/w	r/w	r/w		r/w				r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI3EN	SPI2EN	Res.	Res.	Res.	Res.	LPTIM1EN	TIM14EN	TIM13EN	TIM12EN	TIM7EN	TIM6EN	TIM5EN	TIM4EN	TIM3EN	TIM2EN
r/w	r/w					r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

Bit 18 **USART3EN**: USART3 Peripheral Clocks Enable

Set and reset by software.

0: USART3 peripheral clocks disable (default after reset)

1: USART3 peripheral clocks enabled

The peripheral clocks of the **USART3** are: the kernel clock selected by USART234578SEL and provided to **usart\_ker\_ck** input, and the **rcc\_pclk1** bus interface clock.

Table 8. Register boundary addresses<sup>(1)</sup>

Boundary address	Peripheral	Bus	Register map
0x58027000 - 0x580273FF	RAMECC3		<a href="#">Section 3.4: RAMECC registers</a>
0x58026400 - 0x580267FF	HSEM		<a href="#">Section 10.4: HSEM registers</a>
0x58026000 - 0x580263FF	ADC3		<a href="#">Section 25.7: ADC common registers</a>
0x58025800 - 0x58025BFF	DMAMUX2		<a href="#">Section 17.6: DMAMUX registers</a>
0x58025400 - 0x580257FF	BDMA		<a href="#">Section 16.6: BDMA registers</a>
0x58024C00 - 0x58024FFF	CRC		<a href="#">Section 21.4: CRC registers</a>
0x58024800 - 0x58024BFF	PWR		<a href="#">Section 6.8: PWR register description</a>
0x58024400 - 0x580247FF	RCC		<a href="#">Section 8.7: RCC register description</a>

# USART (Registri za nastavitve GPIO AF)

## 8.7.43 RCC\_AHB4 Clock Register (RCC\_AHB4ENR)

This register can be accessed via two different offset address.

Table 68. RCC\_AHB4ENR address offset and reset value

Register Name	Address Offset	Reset Value
RCC_AHB4ENR	0x0E0	0x0000 0000
RCC_C1_AHB4ENR	0x140	

Enable GPIOB

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	BKPRAMEN	Res.	Res.	HSEIMEN	ADC3EN	Res.	Res.	BDMAEN	Res.	CRCEN	Res.	Res.	Res.
			rw			rw	rw			rw		rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	GPIOKEN	GPIOJEN	GPIOIEN	GPIOHEN	GPIOGEN	GPIOFEN	GPIOEEN	GPIODEN	GPIOCEN	GPIOBEN	GPIOAEN
					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

# USART (Registri za nastavitve GPIO AF)



**STM32H750VB STM32H750ZB**  
**STM32H750IB STM32H750XB**

32-bit Arm® Cortex®-M7 480MHz MCUs, 128 Kbyte Flash,  
 1 Mbyte RAM, 46 com. and analog interfaces, crypto

Datasheet - production data

PB10, PB11  
USART3 = AF7

**Table 9. Port B alternate functions (continued)**

Port	AF0	AF1	AF2	AF3	AF4	AF5	AF6	AF7	AF8	AF9	AF10	AF11	AF12
	SYS	TIM1/2/16/ 17/LPTIM1/ HRTIM1	SAI1/TIM3/ 4/5/12/ HRTIM1	LPUART/ TIM8/ LPTIM2/3/4 5/HRTIM1/ DFSDM1	I2C1/2/3/4/ USART1/ TIM15/ LPTIM2/ DFSDM1/ CEC	SPI1/2/3/4/5/ 6/CEC	SPI2/3/SAI1 3/I2C4/ UART4/ DFSDM1	SPI2/3/6/ USART1/2/3 6/UART7/S DMMC1	SPI6/SAI2/ 4/UART4/5/ 8/LPUART/ SDMMC1/ SPDIFRX1	SAI4/ FDCAN1/2/ TIM13/14/ QUADSPI/ FMC/ SDMMC2/ LCD/ SPDIFRX1	SAI2/4/ TIM8/ QUADSPI/ SDMMC2/ OTG1_HS/ OTG2_FS/ LCD	I2C4/ UART7/ SWPMI1/ TIM1/8/ DFSDM1/ SDMMC2/ MDIOS/ ETH	TIM1/8/FMC /SDMMC1/ MDIOS/ OTG1_FS/ LCD
PB8	-	TIM16_CH1	TIM4_CH3	DFSDM1_ CKIN7	I2C1_SCL	-	I2C4_SCL	SDMMC1_ CKIN	UART4_RX	FDCAN1_ RX	SDMMC2_ D4	ETH_MII_ TXD3	SDMMC1_ D4
PB9	-	TIM17_CH1	TIM4_CH4	DFSDM1_ DATIN7	I2C1_SDA	SPI2_NSS/ I2S2_WS	I2C4_SDA	SDMMC1_ CDIR	UART4_TX	FDCAN1_ TX	SDMMC2_ D5	I2C4_ SMBA	SDMMC1_ D5
<b>PB10</b>	-	TIM2_CH3	HRTIM_ SCOUT	LPTIM2_IN 1	I2C2_SCL	SPI2_SCK/ I2S2_CK	DFSDM1_ DATIN7	<b>USART3_ TX</b>	-	QUADSPI_ BK1_NCS	OTG_HS_ ULPI_D3	ETH_MII_ RX_ER	-
<b>PB11</b>	-	TIM2_CH4	HRTIM_ SCIN	LPTIM2_ ETR	I2C2_SDA	-	DFSDM1_ CKIN7	<b>USART3_ RX</b>	-	-	OTG_HS_ ULPI_D4	ETH_MII_ TX_EN/ ETH_RMII_ TX_EN	-
PB12	-	TIM1_BKIN	-	-	I2C2_SMBA	SPI2_NSS/ I2S2_WS	DFSDM1_ DATIN1	USART3_ CK	-	FDCAN2_ RX	OTG_HS_ ULPI_D5	ETH_MII_ TXD0/ETH_ RMII_TXD0	OTG_HS_ ID
PB13	-	TIM1_CH1N	-	LPTIM2_ OUT	-	SPI2_SCK/ I2S2_CK	DFSDM1_ CKIN1	USART3_ CTS/ USART3_ NSS	-	FDCAN2_ TX	OTG_HS_ ULPI_D6	ETH_MII_ TXD1/ETH_ RMII_TXD1	-
PB14	-	TIM1_CH2N	TIM12_ CH1	TIM8_ CH2N	USART1_TX	SPI2_MISO/ I2S2_SDI	DFSDM1_ DATIN2	USART3_ RTS/ USART3_ DE	UART4_ RTS/ UART4_ DE	SDMMC2_ D0	-	-	OTG_HS_ DM
PB15	RTC_ REFIN	TIM1_CH3N	TIM12_ CH2	TIM8_ CH3N	USART1_RX	SPI2_MOSI/ I2S2_SDO	DFSDM1_ CKIN2	-	UART4_ CTS	SDMMC2_ D1	-	-	OTG_HS_ DP

# USART (Registri za nastavitve GPIO AF)

## 11.4.1 GPIO port mode register (GPIOx\_MODER) (x =A to K)

Address offset: 0x00

Reset value: 0xABFF FFFF for port A

Reset value: 0xFFFF FEBF for port B

Reset value: 0xFFFF FFFF for other ports

PB10, PB11

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MODER15[1:0]		MODER14[1:0]		MODER13[1:0]		MODER12[1:0]		MODER11[1:0]		MODER10[1:0]		MODER9[1:0]		MODER8[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MODER7[1:0]		MODER6[1:0]		MODER5[1:0]		MODER4[1:0]		MODER3[1:0]		MODER2[1:0]		MODER1[1:0]		MODER0[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:0 **MODER[15:0][1:0]**: Port x configuration I/O pin y (y = 15 to 0)

These bits are written by software to configure the I/O mode.

00: Input mode

01: General purpose output mode

**10: Alternate function mode**

11: Analog mode (reset state)

**Table 8. Register boundary addresses<sup>(1)</sup>**

Boundary address	Peripheral	Bus	Register map
0x58020400 - 0x580207FF	GPIOB		<a href="#">Section 11.4: GPIO registers</a>
0x58020000 - 0x580203FF	GPIOA		<a href="#">Section 11.4: GPIO registers</a>

# USART (Registri za nastavitve GPIO AF)

## 11.4.10 GPIO alternate function high register (GPIOx\_AFRH) (x = A to J)

Address offset: 0x24

Reset value: 0x0000 0000

PB10, PB11

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AFR15[3:0]				AFR14[3:0]				AFR13[3:0]				AFR12[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AFR11[3:0]				AFR10[3:0]				AFR9[3:0]				AFR8[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:0 **AFR[15:8][3:0]**: Alternate function selection for port x I/O pin y (y = 15 to 8)

These bits are written by software to configure alternate function I/Os.

0000: AF0  
0001: AF1  
0010: AF2  
0011: AF3  
0100: AF4  
0101: AF5  
0110: AF6  
**0111: AF7**  
1000: AF8  
1001: AF9  
1010: AF10  
1011: AF11  
1100: AF12  
1101: AF13  
1110: AF14  
1111: AF15

# USART – stanje, nastavitve

Table 401. USART register map and reset values

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	USART_CR1 FIFO disabled	Res.	Res.	FIFOEN	M1	EOBIE	RTOIE	DEAT[4:0]				DEDT[4:0]				OVER8	CMIE	MME	M0	WAKE	PCE	PS	PEIE	TXEIE	TCIE	RXNEIE	IDLEIE	TE	RE	UESM	UE		
	Reset value			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x04	USART_CR2	ADD[7:0]							RTOEN	ABRMOD[1:0]		ABREN	MSBFIRST	DATAINV	TXINV	RXINV	SWAP	LINEN	STOP[1:0]	CLKEN	CPOL	CPHA	LBCL	Res.	LBDIE	LBDL	ADDM7	DIS_NSS	Res.	Res.	SLVEN		
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x0C	USART_BRR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BRR[15:0]																
	Reset value																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x1C	USART_ISR FIFO mode disabled	Res.	Res.	Res.	Res.	Res.	Res.	TCBGT	Res.	Res.	REACK	TEACK	WUF	RWU	SBKF	CMF	BUSY	ABRF	ABRE	UDR	EOBF	RTOF	CTS	CTSIF	LBDF	TXE	TC	RXNE	IDLE	ORE	NE	FE	PE
	Reset value							0			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
0x24	USART_RDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RDR[8:0]								
	Reset value																									0	0	0	0	0	0	0	0
0x28	USART_TDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TDR[8:0]								
	Reset value																									0	0	0	0	0	0	0	0

Ctrl1 reg.

Ctrl2 reg.

BRate reg.

Status reg.

Data Rx/TX reg.

## Osnovni registri za delovanje USART naprave:

### USART\_ISR

TXE=1, pošlji znak | RXNE=1 prejet znak

USART\_RDR, USART\_TDR : Data Registra: RX, TX

USART\_RDR : Received Char | USART\_TDR : Send Char

USART\_BRR : BaudRate Register

delilnik frekvence za baudrate

USART\_CR1 : Control Register 1

ENABLE USART (UE), TX (TE), RX (RE) bits

# USART (Registri za nastavitve delovanja)

## 48.7.2 USART control register 1 [alternate] (USART\_CR1)

Address offset: 0x00

Reset value: 0x0000 0000

The same register can be used in FIFO mode enabled (previous section) and FIFO mode disabled (this section).

### FIFO mode disabled

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	FIFO EN	M1	EOBIE	RTOIE	DEAT[4:0]				DEDT[4:0]					
		r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OVER8	CMIE	MME	M0	WAKE	PCE	PS	PEIE	TXEIE	TCIE	RXNEIE	IDLEIE	TE	RE	UESM	UE
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

Bit 0 **UE**: USART enable

When this bit is cleared, the USART prescalers and outputs are stopped immediately, and all current operations are discarded. The USART configuration is kept, but all the USART\_ISR status flags are reset. This bit is set and cleared by software.

0: USART prescaler and outputs disabled, low-power mode

1: USART enabled

Bit 2 **RE**: Receiver enable

This bit enables the receiver. It is set and cleared by software.

0: Receiver is disabled

1: Receiver is enabled and begins searching for a start bit

Bit 15 **OVER8**: Oversampling mode

0: Oversampling by 16

1: Oversampling by 8

This bit can only be written when the USART is disabled (UE=0).

Note: In LIN, IrDA and Smartcard modes, this bit must be kept cleared.

Bit 3 **TE**: Transmitter enable

This bit enables the transmitter. It is set and cleared by software.

0: Transmitter is disabled

1: Transmitter is enabled

Note: During transmission, a low pulse on the TE bit ('0' followed by '1') sends a preamble (idle line) after the current word, except in Smartcard mode. In order to generate an idle character, the TE must not be immediately written to '1'. To ensure the required duration, the software can poll the TEACK bit in the USART\_ISR register.

In Smartcard mode, when TE is set, there is a 1 bit-time delay before the transmission starts.

Bit 28 **M1**: Word length

This bit must be used in conjunction with bit 12 (M0) to determine the word length. It is set or cleared by software.

M[1:0] = '00': 1 start bit, 8 Data bits, n Stop bit

M[1:0] = '01': 1 start bit, 9 Data bits, n Stop bit

M[1:0] = '10': 1 start bit, 7 Data bits, n Stop bit

This bit can only be written when the USART is disabled (UE=0).

Note: In 7-bits data length mode, the Smartcard mode, LIN master mode and Auto baud rate (0x7F and 0x55 frames detection) are not supported.

Bit 12 **M0**: Word length

This bit is used in conjunction with bit 28 (M1) to determine the word length. It is set or cleared by software (refer to bit 28 (M1) description).

This bit can only be written when the USART is disabled (UE=0).



# USART (Registri za nastavitve delovanja)

## 48.7.3 USART control register 2 (USART\_CR2)

Address offset: 0x04

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADD[7:0]								RTOEN	ABRMOD[1:0]		ABREN	MSBFIRST	DATAINV	TXINV	RXINV
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWAP	LINEN	STOP[1:0]		CLKEN	CPOL	CPHA	LBCL	Res.	LBDIE	LBDL	ADDM7	DISNSS	Res.	Res.	SLVEN
rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw			rw

Bits 13:12 **STOP[1:0]**: stop bits

These bits are used for programming the stop bits.

00: 1 stop bit

01: 0.5 stop bit.

10: 2 stop bits

11: 1.5 stop bits

This bitfield can only be written when the USART is disabled (UE=0).

# USART (Registri za nastavitve delovanja)

## 48.7.5 USART baud rate register (USART\_BRR)

This register can only be written when the USART is disabled (UE=0). It may be automatically updated by hardware in auto baud rate detection mode.

Address offset: 0x0C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BRR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:16 Reserved, must be kept at reset value.

Bits 15:0 **BRR[15:0]**: USART baud rate

**BRR[15:4]**

BRR[15:4] = USARTDIV[15:4]

**BRR[3:0]**

When **OVER8 = 0**, **BRR[3:0] = USARTDIV[3:0]**.

When **OVER8 = 1**:

BRR[2:0] = USARTDIV[3:0] shifted 1 bit to the right.

BRR[3] must be kept cleared.

## 48.5.7 USART baud rate generation

The baud rate for the receiver and transmitter (Rx and Tx) are both set to the value programmed in the USART\_BRR register.

**Equation 1: baud rate for standard USART (SPI mode included) (OVER8 = '0' or '1')**

In case of **oversampling by 16**, the baud rate is given by the following formula:

$$\text{Tx/Rx baud} = \frac{\text{usart\_ker\_ckpres}}{\text{USARTDIV}}$$

In case of oversampling by 8, the baud rate is given by the following formula:

$$\text{Tx/Rx baud} = \frac{2 \times \text{usart\_ker\_ckpres}}{\text{USARTDIV}}$$

## How to derive USARTDIV from USART\_BRR register values

### Example 1

To obtain 9600 baud with usart\_ker\_ck\_pres= 8 MHz:

- In case of oversampling by 16:  
USARTDIV = 8 000 000/9600  
BRR = USARTDIV = 833d = 0341h
- In case of oversampling by 8:  
USARTDIV = 2 \* 8 000 000/9600  
USARTDIV = 1666,66 (1667d = 683h)  
BRR[3:0] = 3h >> 1 = 1h  
BRR = 0x681

### Example 2

To obtain 921.6 Kbaud with usart\_ker\_ck\_pres = 48 MHz:

- In case of oversampling by 16:  
USARTDIV = 48 000 000/921 600  
BRR = USARTDIV = 52d = 34h
- In case of oversampling by 8:  
USARTDIV = 2 \* 48 000 000/921 600  
USARTDIV = 104 (104d = 68h)  
BRR[3:0] = USARTDIV[3:0] >> 1 = 8h >> 1 = 4h  
BRR = 0x64

# USART (Registri za nastavitve delovanja)

## 48.7.5 USART baud rate register (USART\_BRR)

This register can only be written when the USART is disabled (UE=0). It may be automatically updated by hardware in auto baud rate detection mode.

Address offset: 0x0C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BRR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:16 Reserved, must be kept at reset value.

Bits 15:0 **BRR[15:0]**: USART baud rate

**BRR[15:4]**

BRR[15:4] = USARTDIV[15:4]

**BRR[3:0]**

When OVER8 = 0, BRR[3:0] = USARTDIV[3:0].

When OVER8 = 1:

BRR[2:0] = USARTDIV[3:0] shifted 1 bit to the right.

BRR[3] must be kept cleared.

$$\underline{64M/115200 = 555.55 = 556}$$

In case of **oversampling by 16**, the baud rate is given by the following formula:

$$\text{Tx/Rx baud} = \frac{\text{usart\_ker\_ckpres}}{\text{USARTDIV}}$$

# USART (Registri za nastavitve delovanja)

## 48.7.10 USART interrupt and status register [alternate] (USART\_ISR)

Address offset: 0x1C

Reset value: 0x0000 00C0

The same register can be used in FIFO mode enabled (previous section) and FIFO mode disabled (this section).

### FIFO mode disabled

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	TCBGT	Res.	Res.	RE ACK	TE ACK	WUF	RWU	SBKF	CMF	BUSY
						r			r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABRF	ABRE	UDR	EOBF	RTOF	CTS	CTSIF	LBDIF	TXE	TC	RXNE	IDLE	ORE	NE	FE	PE
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

Bit 5 **RXNE**: Read data register not empty

RXNE bit is set by hardware when the content of the USART\_RDR shift register has been transferred to the USART\_RDR register. It is cleared by reading from the USART\_RDR register. The RXNE flag can also be cleared by writing 1 to the RXFRQ in the USART\_RQR register.

An interrupt is generated if RXNEIE=1 in the USART\_CR1 register.

0: Data is not received

1: Received data is ready to be read.

Bit 7 **TXE**: Transmit data register empty

TXE is set by hardware when the content of the USART\_TDR register has been transferred into the shift register. It is cleared by writing to the USART\_TDR register. The TXE flag can also be set by writing 1 to the TXFRQ in the USART\_RQR register, in order to discard the data (only in Smartcard T=0 mode, in case of transmission failure).

An interrupt is generated if the TXEIE bit =1 in the USART\_CR1 register.

0: Data register full

1: Data register not full

# USART (Registri za nastavitve delovanja)

## 48.7.12 USART receive data register (USART\_RDR)

Address offset: 0x24

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	RDR[8:0]								
							r	r	r	r	r	r	r	r	r

Bits 31:9 Reserved, must be kept at reset value.

Bits 8:0 **RDR[8:0]**: Receive data value

Contains the received data character.

The RDR register provides the parallel interface between the input shift register and the internal bus (see [Figure 568](#)).

When receiving with the parity enabled, the value read in the MSB bit is the received parity bit.

## 48.7.13 USART transmit data register (USART\_TDR)

Address offset: 0x28

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	TDR[8:0]								
							r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

Bits 31:9 Reserved, must be kept at reset value.

Bits 8:0 **TDR[8:0]**: Transmit data value

Contains the data character to be transmitted.

The USART\_TDR register provides the parallel interface between the internal bus and the output shift register (see [Figure 568](#)).

When transmitting with the parity enabled (PCE bit set to 1 in the USART\_CR1 register), the value written in the MSB (bit 7 or bit 8 depending on the data length) has no effect because it is replaced by the parity.

Note: This register must be written only when TXE/TXFNF=1.

# USART – krmiljenje

## Potrebni koraki za krmiljenje USART naprave:

1. **Vklop USART3 naprave**
  - **RCC\_APB1LENR** :  $b_{18}=1$  (USART3 Enable Clock)
2. **Nastavitev GPIOB priključkov 10,11 na AF7(Alt. Function7)**
  - **RCC\_AHB4ENR** (Peripheral Clock Register):  $b_1=1$  .. Port B Enable
  - **GPIOB\_MODER** (Mode Register): 0b10, AF on pins PB10,PB11
  - **GPIOB\_AFRH** (AF Register): 0x07700 AF7 on pins PB10,PB11
3. **Nastavitev hitrosti delovanja (BaudRate)**
  - **USART3\_BRR** (BaudRate Register):  $64M/115200 \approx 556$
4. **Sprožitev delovanja**
  - **USART3\_CR1** (Control Register 1): 0b1101 TX, RX, USART Enable bits

## 5. Delovanje

Oddaja znaka:

- **USART3\_ISR**: ko  $TXE=1$ , vpis znaka v **USART3\_TDR**

Sprejem znaka:

- **USART3\_ISR**: ko  $RXNE=1$ , preberi znak iz **USART3\_RDR**

# USART – krmiljenje

## Naslovi registrov:

```
// AHB4ENR register offset is 0xE0
.equ    RCC_AHB4ENR,    0x580244E0 // RCC AHB4
peripheral clock reg
// RCC base address is 0x58024400
.equ    RCC_BASE,      0x58024400 // RCC base reg

// APB1LENR register offset is 0xE8
.equ    RCC_APB1LENR,  0xE8 // RCC APB1LENR
peripheral clock reg

// GPIOB base address is 0x58020400
.equ    GPIOB_BASE,    0x58020400 // GPIOB base
address)

// AFRH register offset is 0x24
.equ    GPIOx_AFRH,    0x24 //
.equ    GPIO_AFRH_VALUE, 0x7700 // AF7 on
PB10,11
```

```
// USART3 base address is 0x40004800
.equ    USART3_BASE,   0x40004800 // USART3 base
address)

// CRx registers
// CR1 register
.equ    USART_CR1,     0x00 // CR1 register
.equ    USART_CR1_VAL, 0b1101 // CR1 register value



















// CRx registers
.equ    USART_CR2,     0x04 // CR2 register
.equ    USART_CR3,     0x08 // CR3 register

// BRR register
.equ    USART_BRR,     0x0C // BRR register
.equ    USART_BRR_VAL,556 // BRR register 64 000 000 /
115 200 = 555.55 = 556

// ISR register
.equ    USART_ISR,     0x1c // ISR register

// Data registers
.equ    USART_RDR,     0x24 // Receive Data Register
.equ    USART_TDR,     0x28 // Transmit Data Register
```

# CubeIDE – Registers okno

Name	Value
▼  0101 General Registers	
 0101 r0	0x0
 0101 r1	0x0
 0101 r2	0x0
 0101 r3	0x0
 0101 r4	0x0
 0101 r5	0x1000
 0101 r6	0x40020c14
 0101 r7	0x0
 0101 r8	0x0
 0101 r9	0x0
 0101 r10	0x0
 0101 r11	0x0
 0101 r12	0x0
 0101 sp	0x20020000
 0101 lr	0xffffffff
 0101 pc	0x800002a
 0101 xpsr	0x41000000



# CubeIDE – SFR okno

(x)= Variables   Breakpoints   Expressions   Registers   Live Expressions   Peripherals   SFRs ×

type filter text

Register	Address	Value
> DFSDM		
> TIM16		
> TIM17		
> TIM15		
> USART1		
> USART2		
✓ USART3		
> CR1	0x40004800	0xd
> CR2	0x40004804	0x0
> CR3	0x40004808	0x0
> BRR	0x4000480c	0x22c
> GTPR	0x40004810	0x0
> RTOR	0x40004814	0x0
> RQR	0x40004818	
> ISR	0x4000481c	0x600d0
> ICR	0x40004820	
> RDR	0x40004824	0x0
> TDR	0x40004828	0x6c
> PRESC	0x4000482c	0x0
> UART4		
> UART5		
> USART6		
> UART7		
> UART8		