

ARM

Vhodno / izhodne naprave

DBGU – Debug Unit

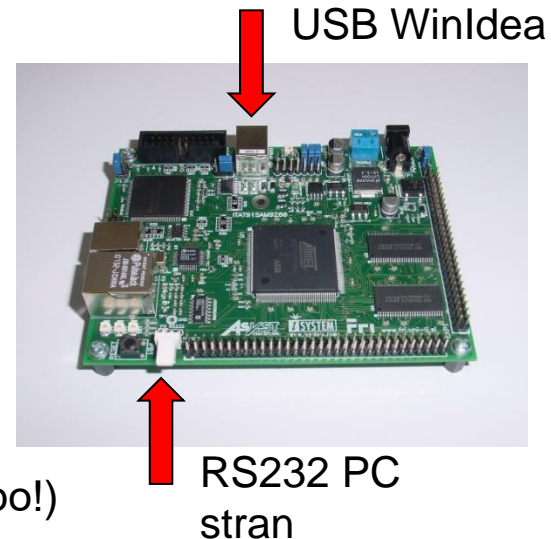
Delo na FRI-SMS razvojnem sistemu

Priključitev :

- **USB** priklp na **daljši stranici**, sveti **zelena LED** dioda

Poseben projekt za FRI-SMS (e-učilnica) :

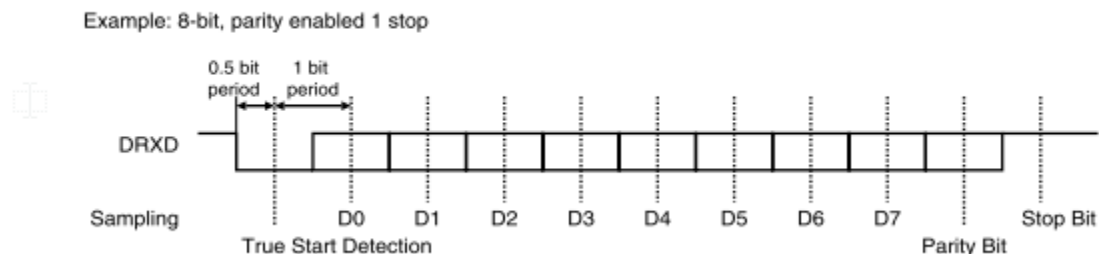
- **dodatne nastavitve** (informativno) :
 - frekvenca urinega signala (višja poveča porabo!)
 - vklop predpomnilnikov
 - inicializacija sklada oz. SP – kazalca na sklad
- **dodajanje vsebine (start.s):**
 - podatki/operandi:
 - dodamo v `/*constants*/` ,končamo z `.align`
 - program :
 - dodamo v `/* enter your code here */`
 - na koncu programa je mrtva zanka
 - podprograme dodamo za mrtvo zanko



DBGU ('debug unit' – razhroščevalna enota)

- Enota za dostop do vseh razhroščevalnih zmožnosti mikrokrmilnika
- V enoto je integriran UART (Universal Asynchronous Receiver Transmitter) z dvema priključkoma (sprejem – RX, oddaja – TX) in naslednjimi zmožnostmi:
 - dolžina znakov samo **8 bitov + bit parnosti**
 - **neodvisno delovanje** sprejemnika in oddajnika
 - generator hitrosti prenosa (baud rate generator) je skupni za oddajnik in sprejemnik
 - hitrost prenosa nastavljiva med **$MCK/(16*65536)$ in $MCK/16$**

Figure 28-5. Character Reception



DBGU ('debug unit' – razhroščevalna enota)

Potrebni koraki za nastavitev in delovanje UART (komunikacija z drugo napravo):

Nastavitev DBGU (UART) - potrebni koraki (dostopi 32 bitni): :

1. Nastavi **način delovanja** (normal mode) in parnost z vpisom v DBGU_MR
2. Nastavi **hitrost prenosa** z vpisom v DBGU_BRGR
vrednost, ki jo vpišemo izračunamo po formuli : $MCK/(16*BAUD_RATE)$
3. **Omogoči oddajnik in sprejemnik** z vpisom 1 v bita TXEN in RXEN v DBGU_CR

Delovanje (dostopi 32 bitni):

Oddaja znaka preko UART:

- preverimo, ali je oddajnik pripravljen na oddajo (bit TXRDY v DBGU_SR postavljen na 1)
- **vpišemo znak**, ki ga želimo poslati, vpišemo v spodnjih 8 bitov DBGU_THR

Sprejem znaka preko UART:

- ob sprejemu znaka se **na 1 postavi bit RXRDY v DBGU_SR**
- znak **preberemo v spodnjih 8 bitih DBGU_RHR**
 - **Branje DBGU_RHR zbriše zastavico RXRDY v DBGU_SR !!!**

DBGU ('debug unit' – razhroščevalna enota)

Potrebni koraki za nastavitve UART (komunikacija z drugo napravo):

1. Nastavi **način delovanja** (normal mode) in parnost z vpisom v DBGU_MR

28.5.2 Debug Unit Mode Register

Name: DBGU_MR

Access Type: Read-write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	CHMODE	–	–	–	PAR	–	–
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	–

• PAR: Parity Type

PAR			Parity Type
0	0	0	Even parity
0	0	1	Odd parity
0	1	0	Space: parity forced to 0
0	1	1	Mark: parity forced to 1
1	x	x	No parity

• CHMODE: Channel Mode

CHMODE		Mode Description
0	0	Normal Mode
0	1	Automatic Echo
1	0	Local Loopback
1	1	Remote Loopback

2. Nastavi **hitrost prenosa** z vpisom v DBGU_BRGR, vrednost, ki jo vpišemo izračunamo po formuli :

$$MCK/(16*BAUD_RATE) \quad \text{Primer: } 48M/(16*19200)=156.25\approx 156$$

3. **Omogoči oddajnik in sprejemnik** z vpisom 1 v bita TXEN in RXEN v DBGU_CR

28.5.1 Debug Unit Control Register

Name: DBGU_CR

Access Type: Write-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	RSTSTA
7	6	5	4	3	2	1	0
TXDIS	TXEN	RXDIS	RXEN	RSTTX	RSTRX	–	–

• RXEN: Receiver Enable

0 = No effect.

1 = The receiver is enabled if RXDIS is 0.

• TXEN: Transmitter Enable

0 = No effect.

1 = The transmitter is enabled if TXDIS is 0.

DBGU ('debug unit' – razhroščevalna enota)

Potrebni koraki za nastavitev in delovanje UART (testiranje brez povezave):

Nastavitev in Delovanje sta enaki kot pri priključeni napravi.

Edina sprememba je pri vklopu „Local Loopback“ testnega načina :

1. Nastavi **način delovanja** (Local Loopback mode) in vse ostalo enako kot pri priključeni napravi

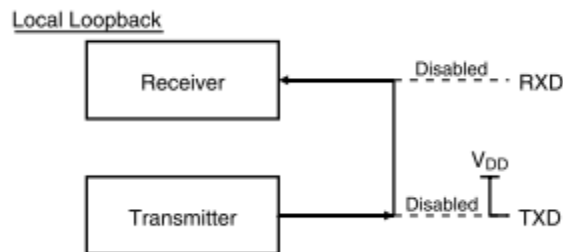
28.4.5 Test Modes

The Debug Unit supports three tests modes. These modes of operation are programmed by using the field CHMODE (Channel Mode) in the mode register DBGU_MR.

The Automatic Echo mode allows bit-by-bit retransmission. When a bit is received on the DRXD line, it is sent to the DTXD line. The transmitter operates normally, but has no effect on the DTXD line.

The **Local Loopback** mode allows the transmitted characters to be received. **DTXD and DRXD pins are not used** and the **output of the transmitter is internally connected to the input of the receiver.** The DRXD pin level has no effect and the DTXD line is held high, as in idle state.

The Remote Loopback mode directly connects the DRXD pin to the DTXD line. The transmitter and the receiver are disabled and have no effect. This mode allows a bit-by-bit retransmission.



DBGU ('debug unit' – razhroščevalna enota)

Potrebni koraki za nastavitve UART (testiranje brez povezave):

1. Nastavi **način delovanja** (normal mode) in parnost z vpisom v DBGU_MR

28.5.2 Debug Unit Mode Register

Name: DBGU_MR

Access Type: Read-write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	CHMODE	–	–	–	PAR	–	–
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	–

• PAR: Parity Type

PAR			Parity Type
0	0	0	Even parity
0	0	1	Odd parity
0	1	0	Space: parity forced to 0
0	1	1	Mark: parity forced to 1
1	x	x	No parity

• CHMODE: Channel Mode

CHMODE		Mode Description
0	0	Normal Mode
0	1	Automatic Echo
1	0	Local Loopback
1	1	Remote Loopback

2. Nastavi **hitrost prenosa** z vpisom v DBGU_BRGR, vrednost, ki jo vpišemo izračunamo po formuli :

$$MCK/(16*BAUD_RATE) \quad \text{Primer: } 48M/(16*19200)=156.25 \approx 156$$

3. **Omogoči oddajnik in sprejemnik** z vpisom 1 v bita TXEN in RXEN v DBGU_CR

28.5.1 Debug Unit Control Register

Name: DBGU_CR

Access Type: Write-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	RSTSTA
7	6	5	4	3	2	1	0
TXDIS	TXEN	RXDIS	RXEN	RSTTX	RSTRX	–	–

• RXEN: Receiver Enable

0 = No effect.

1 = The receiver is enabled if RXDIS is 0.

• TXEN: Transmitter Enable

0 = No effect.

1 = The transmitter is enabled if TXDIS is 0.

DBGU ('debug unit' – razhroščevalna enota)

Oddaja znaka preko UART:

- preverimo, ali je oddajnik pripravljen na oddajo (bit TXRDY v DBGU_SR postavljen na 1)

28.5.6 Debug Unit Status Register

Name: DBGU_SR

Access Type: Read-only

31	30	29	28	27	26	25	24
COMMRX	COMMTX	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	RXBUF	TXBUFE	-	TXEMPTY	-
7	6	5	4	3	2	1	0
PARE	FRAME	OVRE	ENDTX	ENDRX	-	TXRDY	RXRDY

- **RXRDY: Receiver Ready**

0 = No character has been received since the last read of the DBGU_RHR or the receiver is disabled.

1 = At least one complete character has been received, transferred to DBGU_RHR and not yet read.

- **TXRDY: Transmitter Ready**

0 = A character has been written to DBGU_THR and not yet transferred to the Shift Register, or the transmitter is disabled.

1 = There is no character written to DBGU_THR not yet transferred to the Shift Register.

- vpišemo znak, ki ga želimo poslati, vpišemo v spodnjih 8 bitov DBGU_THR

Primer čakanja na zastavico na b₂

```
ldr r1, =BAZNI_NASLOV_IO_NAPRAVE
CAKAJ:
ldr r2, [r1, #ODMIK_REGISTRA]
tst r2, #0b0100 @ b2=1 ?
beq CAKAJ @ ne, cakaj
```

DBGU ('debug unit' – razhroščevalna enota)

Sprejem znaka preko UART:

- ob sprejemu znaka se na 1 postavi bit RXRDY v DBGU_SR

28.5.6 Debug Unit Status Register

Name: DBGU_SR

Access Type: Read-only

31	30	29	28	27	26	25	24
COMMRX	COMMTX	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	RXBUFF	TXBUFE	–	TXEMPTY	–
7	6	5	4	3	2	1	0
PARE	FRAME	OVRE	ENDTX	ENDRX	–	TXRDY	RXRDY

- **RXRDY: Receiver Ready**

0 = No character has been received since the last read of the DBGU_RHR or the receiver is disabled.

1 = At least one complete character has been received, transferred to DBGU_RHR and not yet read.

- **TXRDY: Transmitter Ready**

0 = A character has been written to DBGU_THR and not yet transferred to the Shift Register, or the transmitter is disabled.

1 = There is no character written to DBGU_THR not yet transferred to the Shift Register.

Pozor: Če je register **DBGU_RHR** v izpisu (Watch ali SFR okno), se lahko zastavica RXRDY briše (zaradi branja)!

- znak preberemo v spodnjih 8 bitih DBGU_RHR

Primer čakanja na zastavico na b₂

```
ldr r1, =BAZNI_NASLOV_IO_NAPRAVE
CAKAJ:
ldr r2, [r1, #ODMIK_REGISTRA]
tst r2, #0b0100 @ b2=1 ?
beq CAKAJ @ ne, cakaj
```

DBGU ('debug unit' – razhroščevalna enota)

Sprejem / oddaja znakov preko UART:

- ob sprejemu znaka se na 1 postavi bit RXRDY v DBGU_SR
- znak preberemo v spodnjih 8 bitih DBGU_RHR (ob branju se RXRDY zbriše)
- znak, ki ga želimo poslati, vpišemo v spodnjih 8 bitov DBGU_THR in sicer takrat, ko je oddajnik pripravljen na oddajo (bit TXRDY v DBGU_SR postavljen na 1)

Naslovi registrov (dostopi 32 bitni):

.equ DBGU_BASE, 0xFFFFF200	/* Debug Unit Base Address */
.equ DBGU_CR, 0x00	/* DBGU Control Register */
.equ DBGU_MR, 0x04	/* DBGU Mode Register*/
.equ DBGU_IER, 0x08	/* DBGU Interrupt Enable Register*/
.equ DBGU_IDR, 0x0C	/* DBGU Interrupt Disable Register */
.equ DBGU_IMR, 0x10	/* DBGU Interrupt Mask Register */
.equ DBGU_SR, 0x14	/* DBGU Status Register */
.equ DBGU_RHR, 0x18	/* DBGU Receive Holding Register */
.equ DBGU_THR, 0x1C	/* DBGU Transmit Holding Register */
.equ DBGU_BRGR, 0x20	/* DBGU Baud Rate Generator Register */

DBGU ('debug unit' – razhroščevalna enota)

Sprejem / oddaja znakov preko lokalne zanke (Local Loopback):

- znak, ki ga želimo poslati, **vpišemo v spodnjih 8 bitov DBGU_THR** in sicer takrat, ko je oddajnik pripravljen na oddajo (bit **TXRDY v DBGU_SR** postavljen na 1)
- ob **sprejemu znaka** se na 1 postavi bit **RXRDY v DBGU_SR** in znak preberemo v **spodnjih 8 bitih DBGU_RHR** (ob branju se **RXRDY** zbriše)

Naslovi registrov (dostopi 32 bitni):

<code>.equ DBGU_BASE, 0xFFFFF200</code>	<code>/* Debug Unit Base Address */</code>
<code>.equ DBGU_CR, 0x00</code>	<code>/* DBGU Control Register */</code>
<code>.equ DBGU_MR, 0x04</code>	<code>/* DBGU Mode Register*/</code>
<code>.equ DBGU_IER, 0x08</code>	<code>/* DBGU Interrupt Enable Register*/</code>
<code>.equ DBGU_IDR, 0x0C</code>	<code>/* DBGU Interrupt Disable Register */</code>
<code>.equ DBGU_IMR, 0x10</code>	<code>/* DBGU Interrupt Mask Register */</code>
<code>.equ DBGU_SR, 0x14</code>	<code>/* DBGU Status Register */</code>
<code>.equ DBGU_RHR, 0x18</code>	<code>/* DBGU Receive Holding Register */</code>
<code>.equ DBGU_THR, 0x1C</code>	<code>/* DBGU Transmit Holding Register */</code>
<code>.equ DBGU_BRGR, 0x20</code>	<code>/* DBGU Baud Rate Generator Register */</code>

DBGU Priključitev na PC strani

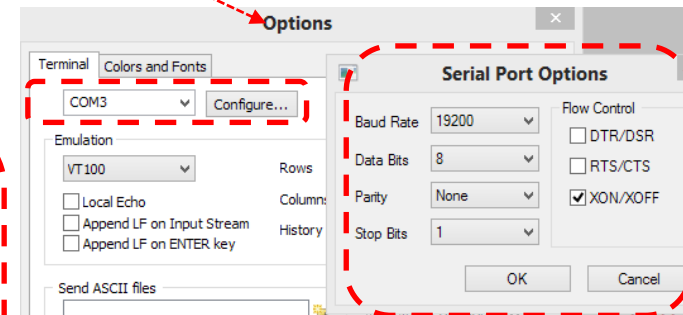
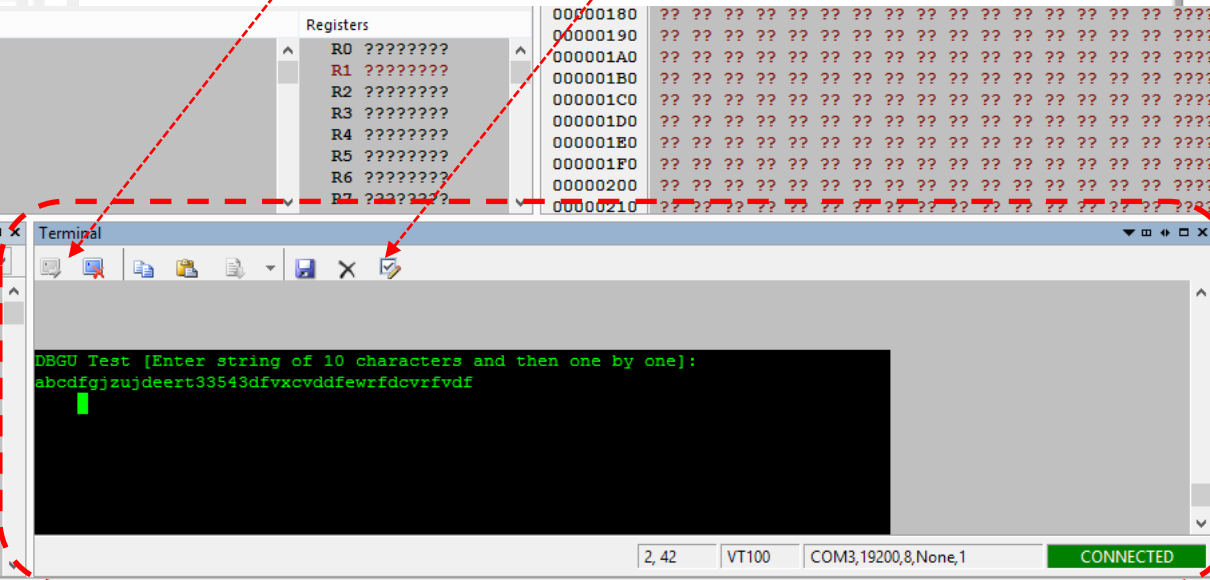
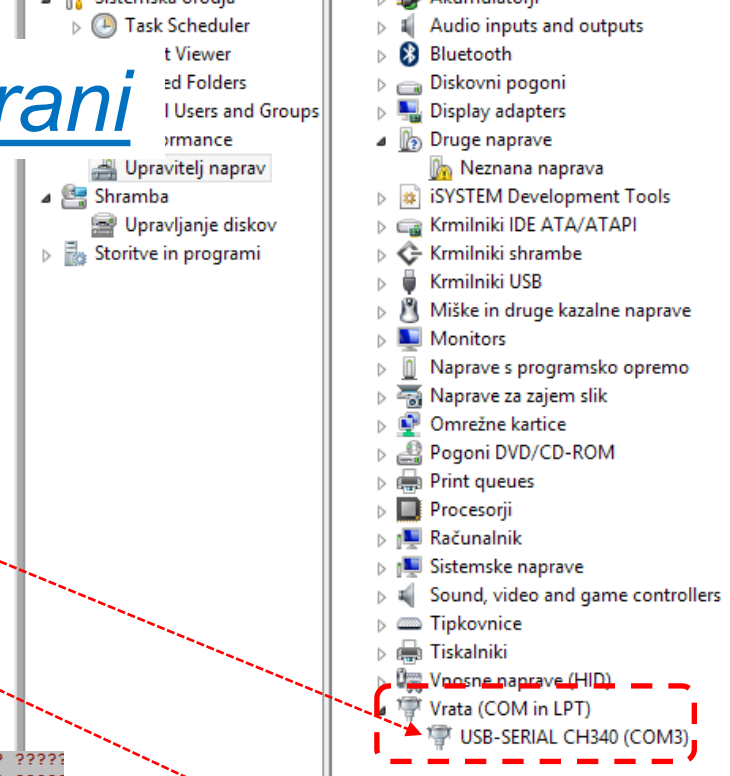
1. Ugotovimo številko serijskega vmesnika (COM porta):

„Upravitelj naprav“

2. Okno Terminal v WinIdea (PC stran):

- Nastavitve COM porta (Options)
- Prikllop/Odklop

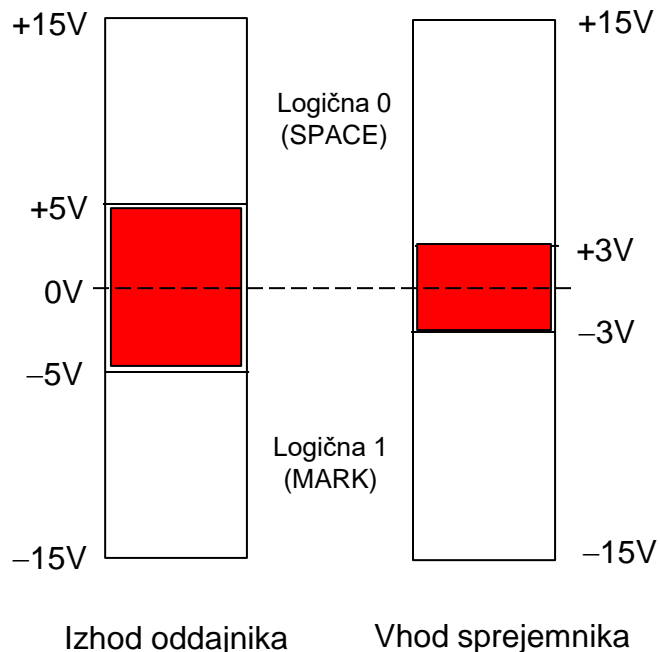
tukaj se kažejo znaki, ki jih pošlje FRI-SMS



Nasveta:

USB-RS232 kable priklopite na USB2
Pogosto ti kabli ne delujejo pod Win10

- **Električni del standarda (iz predmeta VIN – Vh.izh. naprave):**
 - Napetostna in logična nivoja

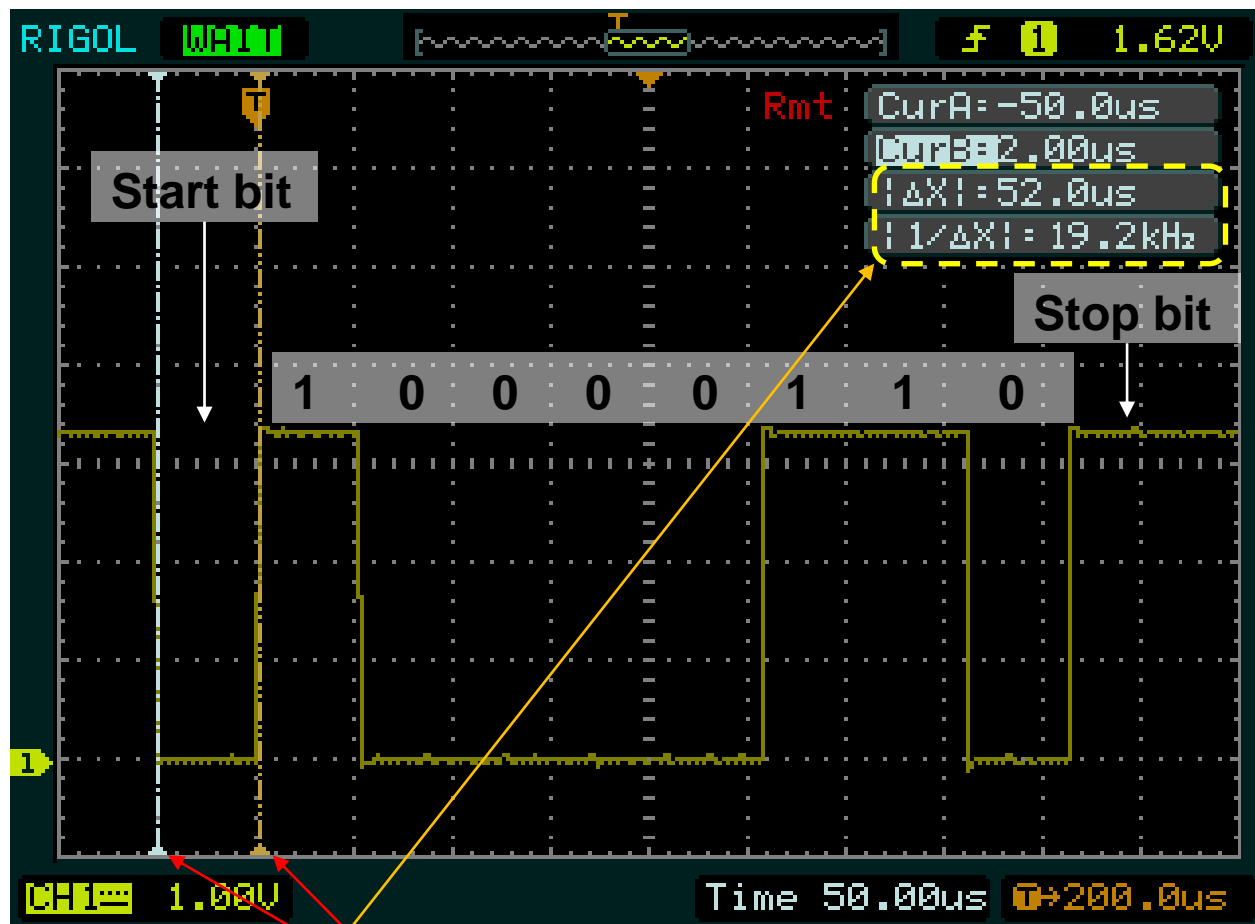


Šumna imuniteta je 2 V ($5V - 3V = 2V$)

Podatkovna signala RxDin TxD:
Logična 1: neg. napetostni nivo
Logična 0: poz. napetostni nivo

Kontrolni signali
Signal aktiven: poz. napetostni nivo
Signal neaktiven: neg. napetostni nivo

Zaslona osciloskopa – meritev DTXD (TTL)



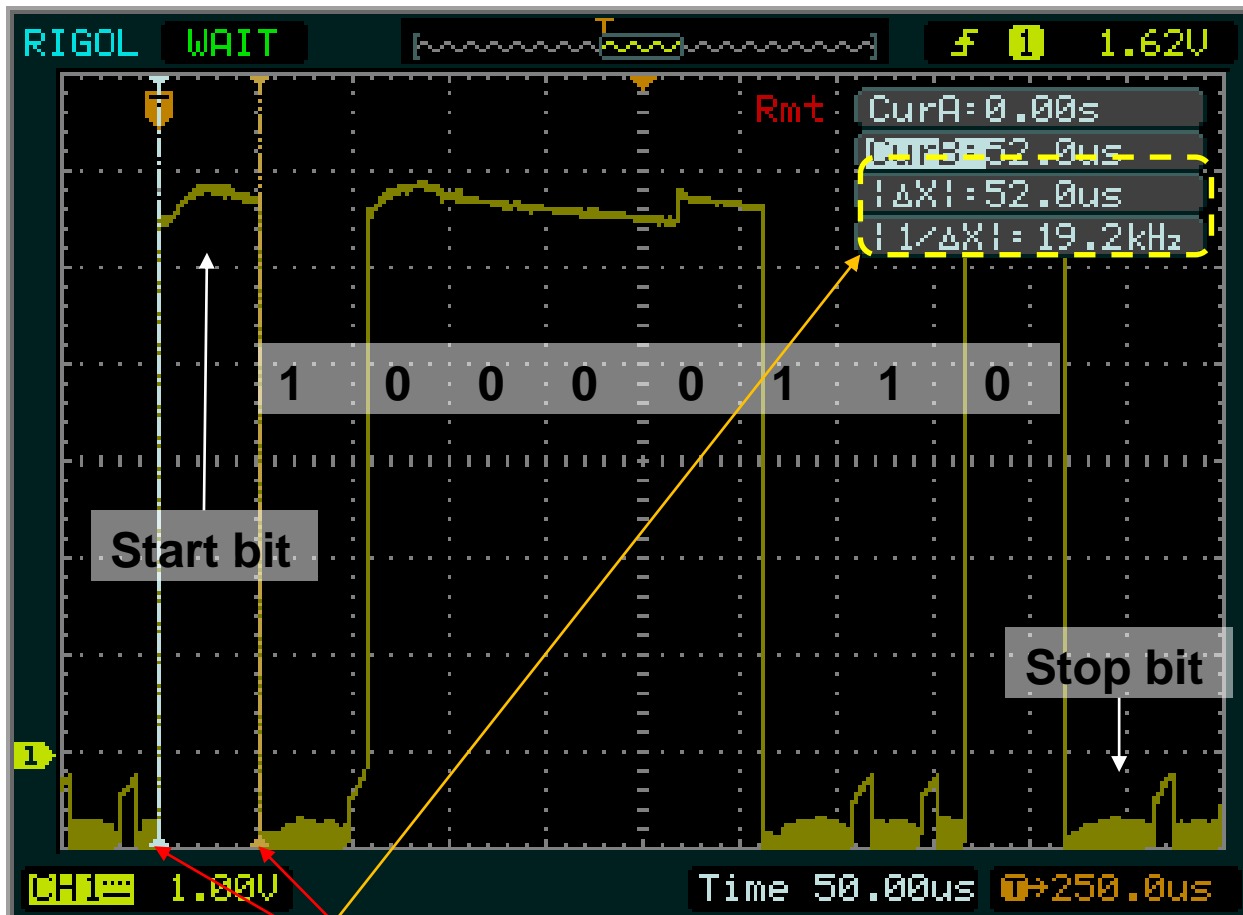
‘a’ :

- ascii koda :
97=0b01100001
- DTXD :
 - Start bit
 - 10000110
 $b_0 \dots b_7$
 - Stop bit

Meritev periode signala (1 baud):

- $52\mu\text{s} \approx 19200 \text{ baud}$

Zaslona osciloskopa – meritev RS232



‘a’ :

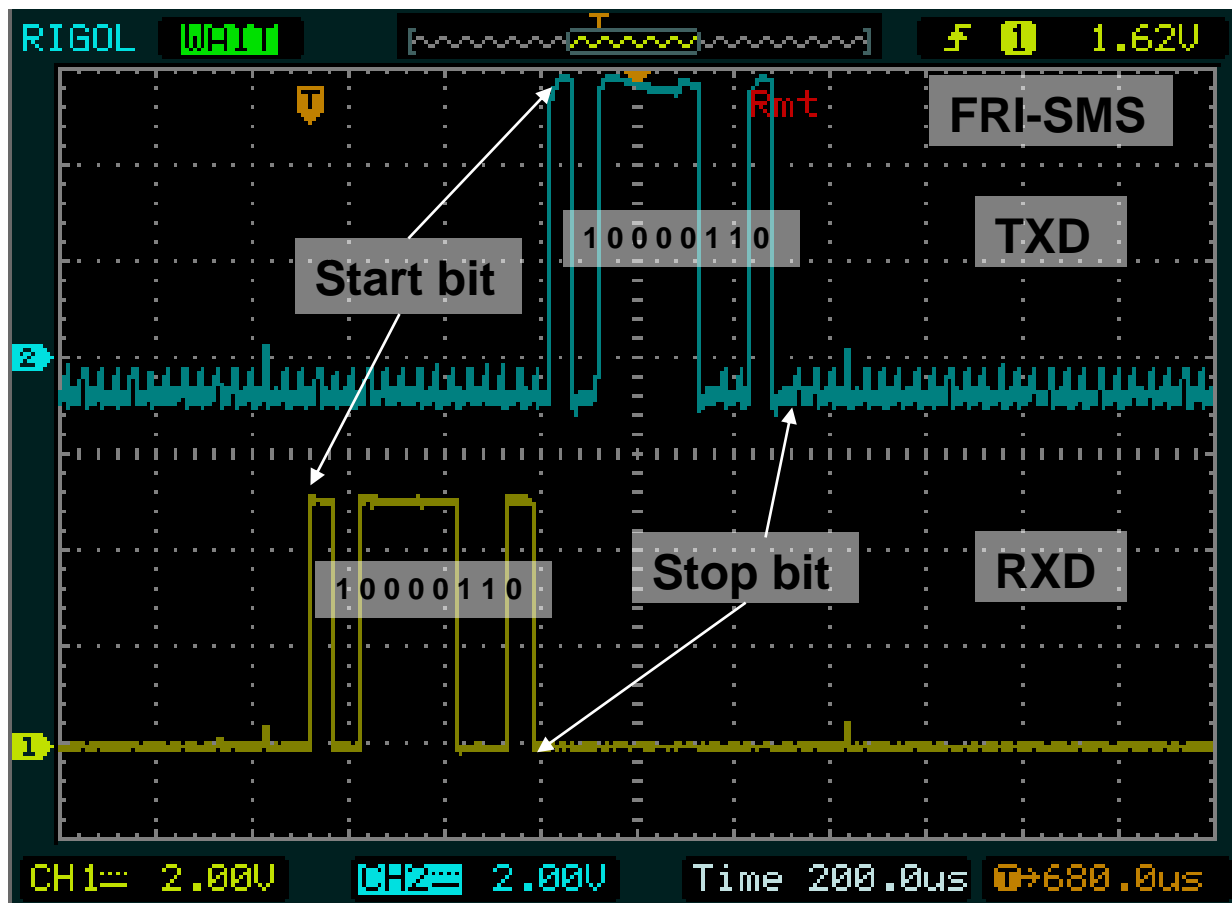
- ascii koda :
97=0b01100001
- [RT]XD :
 - Start bit
 - 10000110
b₀ b₇
 - Stop bit

Meritev periode signala (1 baud):

- 52μs ≈ 19200 baud

Zaslona osciloskopa – RS232 TXD in RXD

„echo“ aplikacija na strani FRI-SMS



‘a’ :

- ascii koda :
97=0b01100001
- [RT]XD :
 - Start bit
 - 10000110
b₀ b₇
 - Stop bit