# 2. neobvezna domača naloga

## Nadzor TV

Vir, 7. 2. 2018

Urban Matko

# Kazalo

**Kazalo** **1**

**Problem** **2**

**Rešitev** **3**

- Inicializacija in pošiljanje preko DBGU 3
- Definicija ukazov 4
- Demonstracija 5

**Koda** **6**

**Prikaz delovanja** **12**

# Problem

Problem naloge je bil nadzor LG televizije preko vmesnika RS232 na FRI-SMS. V problemu je bilo implementirati serijsko povezavo, ter posamezne ukaze, ki jih televizija prepozna, prav tako sem impeleniral delay funkcijo, ki jo uporabim za kratko demonstracijo.

# Rešitev

## Inicializacija in pošiljanje preko DBGU

Kot osnovo sem vzel ze na vajah napisano funkcijo za inicializacijo in pošiljanje preko DBGU-ja.

```
init_dbgu:
    stmfd r13!, {r0-r5, r14}

    ldr r1, =DBGU_BASE

    mov r0, #1 << 11
    str r0, [r1, #DBGU_MR]

    mov r0, #312
    str r0, [r1, #DBGU_BRGR]

    mov r0, #0b1010000
    str r0, [r1, #DBGU_CR]

    ldmfd r13!, {r0-r5, r15}


dbgu_rx:
    stmfd r13!, {r0-r5, r14}
    ldr r1, =DBGU_BASE
    cakaj:
        ldr r2, [r1, #DBGU_SR]
        tst r2, #0b01
        beq cakaj

    ldr r6, [r1, #DBGU_RHR]

    ldmfd r13!, {r0-r5, r15}


dbgu_tx:
    stmfd r13!, {r0-r5, r14}
    ldr r1, =DBGU_BASE
    cakaj1:
        ldr r2, [r1, #DBGU_SR]
        tst r2, #0b010
        beq cakaj1

    str r6, [r1, #DBGU_THR]

    ldmfd r13!, {r0-r5, r15}
```

## Definicija ukazov

Ukaze sem zaradi težav pri oddajanju razdelil po en bajt in jih zapovrstjo poslal iz podprograma za vsako funkcijo. Implementiral sem ukaze za Vklop in Izklop televizije ter nastavitev zvoka za stopnjo 0 in stopnjo 100. Vse ukaze (http://www.proaudioinc.com/Dealer_Area/RS232.pdf) je mozno implementirati na enka način z razdelitvijo kode.

| Vklop televizije | Izklop televizije | Zvok 0% | Zvok 100% |
|---|---|---|---|
| `TV_ON:`<br>`stmfd r13!, {r6, r14}`<br>`ldrb r6,=0x6B`<br>`bl dbgu_tx`<br>`ldrb r6,=0x61`<br>`bl dbgu_tx`<br>`ldrb r6,=0x20`<br>`bl dbgu_tx`<br>`ldrb r6,=0x30`<br>`bl dbgu_tx`<br>`ldrb r6,=0x31`<br>`bl dbgu_tx`<br>`ldrb r6,=0x20`<br>`bl dbgu_tx`<br>`ldrb r6,=0x30`<br>`bl dbgu_tx`<br>`ldrb r6,=0x31`<br>`bl dbgu_tx`<br>`ldrb r6,=0x0D`<br>`bl dbgu_tx`<br>`ldmfd r13!, {r6, r15}` | `TV_OFF:`<br>`stmfd r13!, {r6, r14}`<br>`ldrb r6,=0x6B`<br>`bl dbgu_tx`<br>`ldrb r6,=0x61`<br>`bl dbgu_tx`<br>`ldrb r6,=0x20`<br>`bl dbgu_tx`<br>`ldrb r6,=0x30`<br>`bl dbgu_tx`<br>`ldrb r6,=0x31`<br>`bl dbgu_tx`<br>`ldrb r6,=0x20`<br>`bl dbgu_tx`<br>`ldrb r6,=0x30`<br>`bl dbgu_tx`<br>`ldrb r6,=0x30`<br>`bl dbgu_tx`<br>`ldrb r6,=0x0D`<br>`bl dbgu_tx`<br>`ldmfd r13!, {r6, r15}` | `VOLUME_00:`<br>`stmfd r13!, {r6, r14}`<br>`ldrb r6,=0x6B`<br>`bl dbgu_tx`<br>`ldrb r6,=0x66`<br>`bl dbgu_tx`<br>`ldrb r6,=0x20`<br>`bl dbgu_tx`<br>`ldrb r6,=0x30`<br>`bl dbgu_tx`<br>`ldrb r6,=0x31`<br>`bl dbgu_tx`<br>`ldrb r6,=0x20`<br>`bl dbgu_tx`<br>`ldrb r6,=0x30`<br>`bl dbgu_tx`<br>`ldrb r6,=0x30`<br>`bl dbgu_tx`<br>`ldrb r6,=0x0D`<br>`bl dbgu_tx`<br>`ldmfd r13!, {r6, r15}` | `VOLUME_100:`<br>`stmfd r13!, {r6, r14}`<br>`ldrb r6,=0x6B`<br>`bl dbgu_tx`<br>`ldrb r6,=0x66`<br>`bl dbgu_tx`<br>`ldrb r6,=0x20`<br>`bl dbgu_tx`<br>`ldrb r6,=0x30`<br>`bl dbgu_tx`<br>`ldrb r6,=0x31`<br>`bl dbgu_tx`<br>`ldrb r6,=0x20`<br>`bl dbgu_tx`<br>`ldrb r6,=0x36`<br>`bl dbgu_tx`<br>`ldrb r6,=0x34`<br>`bl dbgu_tx`<br>`ldrb r6,=0x0D`<br>`bl dbgu_tx`<br>`ldmfd r13!, {r6, r15}` |

# Demonstracija

Za demonstracijo sem implementiral se funkcijo delay, ker ob vklopu televizija potrebuje približno 16 sekund, da jo lahko nadzorujemo naprej. Same ukaze pa izvajamo s posameznimi klici podprogramov.

```
_main:
bl init_dbgu
bl INIT_TC0

bl TV_ON
ldr r0,=16000
bl DELAY_TC0


bl VOLUME_00
ldr r0,=5000
bl DELAY_TC0

bl VOLUME_100

ldr r0,=2000
bl DELAY_TC0

bl TV_OFF
```

# Koda

```
.equ PMC_BASE,  0xFFFFFC00  /* (PMC) Base Address */
.equ CKGR_MOR, 0x20        /* (CKGR) Main Oscillator Register */
.equ CKGR_PLLAR,0x28        /* (CKGR) PLL A Register */
.equ PMC_MCKR,  0x30        /* (PMC) Master Clock Register */
.equ PMC_SR,     0x68        /* (PMC) Status Register */

.text
.code 32

.global _error
_error:
  b _error

.global _start
_start:

/* select system mode
  CPSR[4:0]    Mode
  --------------
   10000        User
   10001        FIQ
   10010        IRQ
   10011        SVC
   10111        Abort
   11011        Undef
   11111        System
*/

  mrs r0, cpsr
  bic r0, r0, #0x1F   /* clear mode flags */
  orr r0, r0, #0xDF   /* set supervisor mode + DISABLE IRQ, FIQ*/
  msr cpsr, r0

  /* init stack */
  ldr sp,_Lstack_end

  /* setup system clocks */
  ldr r1, =PMC_BASE

  ldr r0, = 0x0F01
  str r0, [r1,#CKGR_MOR]

osc_lp:
  ldr r0, [r1,#PMC_SR]
  tst r0, #0x01
  beq osc_lp

  mov r0, #0x01
  str r0, [r1,#PMC_MCKR]

  ldr r0, =0x2000bf00 | ( 124 << 16) | 12  /* 18,432 MHz * 125 / 12 */
  str r0, [r1,#CKGR_PLLAR]

pll_lp:
  ldr r0, [r1,#PMC_SR]
  tst r0, #0x02
  beq pll_lp
  /* MCK = PCK/4 */
  ldr r0, =0x0202
  str r0, [r1,#PMC_MCKR]

mck_lp:
  ldr r0, [r1,#PMC_SR]
  tst r0, #0x08
  beq mck_lp

  /* Enable caches */
  mrc p15, 0, r0, c1, c0, 0
  orr r0, r0, #(0x1 <<12)
  orr r0, r0, #(0x1 <<2)
  mcr p15, 0, r0, c1, c0, 0

.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 */


  .equ PMC_BASE, 0xFFFFFC00      /* Power Manag. Controller Base Addr.*/
  .equ PMC_PCER, 0x10                  /* Peripheral Clock Enable Register */
  .equ PIOC_BASE, 0xFFFFF800
  .equ PIO_PER, 0x00
  .equ PIO_OER, 0x10
  .equ PIO_SODR, 0x30
  .equ PIO_CODR, 0x34

  .equ TC0_BASE, 0xFFFA0000 /* TC0 Channel Registers */
  .equ TC_IMR, 0x02C                /* TC0 Interrupt Mask Register */
  .equ TC_IER, 0x24                    /* TC0 Interrupt Enable Register*/
  .equ TC_RC, 0x1C                    /* TC0 Register C */
  .equ TC_RA, 0x14                    /* TC0 Register A */
  .equ TC_CMR, 0x04                /* TC0 Channel Mode Register (Capture Mode / Waveform 
Mode */
  .equ TC_IDR, 0x28                  /* TC0 Interrupt Disable Register */
  .equ TC_SR, 0x20                    /* TC0 Status Register */
  .equ TC_RB, 0x18                    /* TC0 Register B */
  .equ TC_CV, 0x10                    /* TC0 Counter Value */
  .equ TC_CCR, 0x00                  /* TC0 Channel Control Register */

.global _main


/* main program */
_main:
bl init_dbgu
bl INIT_TC0

bl TV_ON
ldr r0,=16000
bl DELAY_TC0
bl VOLUME_00
ldr r0,=5000
bl DELAY_TC0

bl VOLUME_100

ldr r0,=2000
bl DELAY_TC0

bl TV_OFF
@15 sekund potem lahko off


/* user code here */

/* end user code */


_wait_for_ever:
    b _wait_for_ever

init_dbgu:
    stmfd r13!, {r0-r5, r14}

    ldr r1, =DBGU_BASE

    mov r0, #1 << 11
    str r0, [r1, #DBGU_MR]

    mov r0, #312
    str r0, [r1, #DBGU_BRGR]

    mov r0, #0b1010000
    str r0, [r1, #DBGU_CR]

    ldmfd r13!, {r0-r5, r15}


dbgu_rx:
    stmfd r13!, {r0-r5, r14}
    ldr r1, =DBGU_BASE
    cakaj:
        ldr r2, [r1, #DBGU_SR]
        tst r2, #0b01
        beq cakaj

    ldr r6, [r1, #DBGU_RHR]

    ldmfd r13!, {r0-r5, r15}


dbgu_tx:
    stmfd r13!, {r0-r5, r14}
    ldr r1, =DBGU_BASE
    cakaj1:
        ldr r2, [r1, #DBGU_SR]
        tst r2, #0b010
        beq cakaj1

    str r6, [r1, #DBGU_THR]

    ldmfd r13!, {r0-r5, r15}
```

```
INIT_TC0:
  stmfd r13!, {r0, r2, r14}
  ldr r2, =PMC_BASE    /*Enable PMC for TC0 */
  mov r0, #(1 << 17)
  str r0, [r2,#PMC_PCER]

  /*Initialize TC0 MCK/128, RC=375 (1ms) */
  ldr r2, =TC0_BASE
  mov r0, #0b110 << 13 /*WAVE=1, WAVSEL= 10*/
  add r0, r0, #0b011            /* MCK/128 */
  str r0, [r2, #TC_CMR]
  ldr r0, =375                      /* 1 ms at 48 Mhz */
  str r0, [r2, #TC_RC]
  mov r0, #0b0101      /*TC_CLKEN,TC_SWTRG*/
  str r0, [r2, #TC_CCR]
  ldmfd r13!, {r0, r2, r15}

DELAY_TC0:
  stmfd r13!, {r1, r2, r14}
  ldr r2, =TC0_BASE

DLP_TC0:  ldr r1, [r2, #TC_SR]
  tst r1, #1 << 4                              /* CPCS Flag ?*/
  beq DLP_TC0

  subs r0, r0, #1
  bne DLP_TC0
  ldmfd r13!, {r1, r2, r15}


TV_ON:
stmfd r13!, {r6, r14}
ldrb r6,=0x6B
bl dbgu_tx

ldrb r6,=0x61
bl dbgu_tx

ldrb r6,=0x20
bl dbgu_tx

ldrb r6,=0x30
bl dbgu_tx

ldrb r6,=0x31
bl dbgu_tx

ldrb r6,=0x20
bl dbgu_tx

ldrb r6,=0x30
bl dbgu_tx

ldrb r6,=0x31
bl dbgu_tx

ldrb r6,=0x0D
bl dbgu_tx

ldmfd r13!, {r6, r15}
```

```
TV_OFF:
stmfd r13!, {r6, r14}
ldrb r6,=0x6B
bl dbgu_tx

ldrb r6,=0x61
bl dbgu_tx

ldrb r6,=0x20
bl dbgu_tx

ldrb r6,=0x30
bl dbgu_tx

ldrb r6,=0x31
bl dbgu_tx

ldrb r6,=0x20
bl dbgu_tx

ldrb r6,=0x30
bl dbgu_tx

ldrb r6,=0x30
bl dbgu_tx

ldrb r6,=0x0D
bl dbgu_tx

ldmfd r13!, {r6, r15}

VOLUME_00:
stmfd r13!, {r6, r14}
ldrb r6,=0x6B
bl dbgu_tx

ldrb r6,=0x66
bl dbgu_tx

ldrb r6,=0x20
bl dbgu_tx

ldrb r6,=0x30
bl dbgu_tx

ldrb r6,=0x31
bl dbgu_tx

ldrb r6,=0x20
bl dbgu_tx

ldrb r6,=0x30
bl dbgu_tx

ldrb r6,=0x30
bl dbgu_tx

ldrb r6,=0x0D
bl dbgu_tx

ldmfd r13!, {r6, r15}
```

```
VOLUME_100:
stmfd r13!, {r6, r14}
ldrb r6,=0x6B
bl dbgu_tx

ldrb r6,=0x66
bl dbgu_tx

ldrb r6,=0x20
bl dbgu_tx

ldrb r6,=0x30
bl dbgu_tx

ldrb r6,=0x31
bl dbgu_tx

ldrb r6,=0x20
bl dbgu_tx

ldrb r6,=0x36
bl dbgu_tx

ldrb r6,=0x34
bl dbgu_tx

ldrb r6,=0x0D
bl dbgu_tx

ldmfd r13!, {r6, r15}

/* constants */

_Lstack_end:
  .long __STACK_END__

.end
```

# Prikaz delovanja

https://photos.app.goo.gl/iiMHXWLTDJivLHpd8