

```
*****
;* A P P L I C A T I O N   N O T E   F O R   T H E   A V R   F A M I L Y   *
;* Infrapunase universaalpuldi LAUDMANN kuudi vastuvõtmise ja vaatamise *
;* programm. *
;* Programm on firma ATMET materjali AVR410 IR Remout Control Receiver *
;* järgi tehtud. *
;* File Name           : "IR_KOIK.asm" *
;* Ir pult             : Samsungi TV või LAUDMANN *
;* Kuupäev             : 14.11.2003 *
;* Parandused tegi    : Jüri Puhang *
;* Originaal           : AVR410.asm *
;* Target MCU          : AT90S1200 *
;* *
;* SELGITUS: *
;* Programm analüüsib Samsungi puldilt väljastatud koodi. *
;* Kogu kood on 14 bitine *
;* St,St,Ctrj,S4,S3,S2,S1,S0,C5,C4,C3,C2,C1,C0 *
;* Start; Toggle bit; System; Command *
;* *
;* Teistel pultidel avastab ainult start pulsi, *
;* mis indutseerub Port B bit 6 *
;* kui on 3,5 ms pikkune pulss *
;* Vastubõetud koodi bitid c5 kuni co näidatakse programmaatori STK500 *
;* Pordil B LED5 kuni LED0 *
;* The timing is adapted for 4 MHz crystal *
;* *
;*****
.include "1200def.inc" ; Seal on AT90S1200 määrangud

.equ INPUT =2 ;PD2
.equ SYS_ADDR=0 ;The system address

.def S =R0 ; Oleku reg hoidmiseks
; katkestuse korral

.def inttemp =R1
.def ref1 =R2
.def ref2 =R3

.def temp =R16

.def timerL =R17
.def timerH =R18

.def system =R19
.def command =R20

.def bitcnt =R21
.def str =R22 ;Kui start oli siis bit6=1

.cseg
.org 0

rjmp reset
```

## Leht 2 / Lehti 5 Infrapunase TV puldi koodi analüüsiprogramm (eeskuju Atmel AVR410)

```
*****
;* "TIM0_OVF" - Timer/counter overflow interrupt handler *
;* * *
;* The overflow interrupt increments the "timerL" and "timerH" *
;* every 64us and 16,384us. *
;* Prog detect annab timerL-i aja ja kontrollib, selle *
;* möödumisel sisendit *
;* * *
;*****
.org OVFOaddr
TIM0_OVF:   in    S,sreg                ; Oleku reg säilitamine
            inc   timerL                ;Updated every 64us
            inc   inttemp
            brne  TIM0_OVF_exit

            inc   timerH

TIM0_OVF_exit:
            out   sreg,S                ; Oleku reg taastamine
            reti

;*****
;* Example program *
;* * *
;* Initializes timer, ports and interrupts. *
;* * *
;* Calls "detect" in an endless loop and puts the result out on port B. *
;* * *
;*****

RESET:
            ldi   temp,1                ; Taimer ja pordi initsialiseerimine.
            out   TCCR0,temp           ;Timer/Counter 0 clocked at CK

            ldi   temp,1<<TOIE0       ;Enable Timer0 overflow interrupt
            out   TIMSK,temp

            ser   temp                 ;PORTB as output
            out   DDRB,temp

            ser   command              ;Et väljund oleks nullitud
            ser   system
            ldi   str,0b01000000      ;Start algul NULL

            sei                       ;Enable gobal iterrupt

main:      rcall  detect               ;Call RC5 detect routine

            out   PORTB,command       ; Väljastada Toggle+command

            rjmp  main
```

```

;*****
;* "detect" - RC5 decode routine *
;* *
;* This subroutine decodes the RC5 bit stream applied on *
;* PORTD pin "INPUT". *
;* *
;* If successe: The command and system address are *
;* returned in "command" and "system". *
;* Bit 7 of "command" holds the toggle bit. *
;* *
;* *
;* Crystal frequency is 4MHz *
;* *
;*****
DETECT: clr inttemp
        clr timerH

detect1: clr timerL

detect2: cpi timerH,8 ;If line not idle within 131ms
        brlo dll ; End if longer then 131ms
        ret

dll: cpi timerL,55 ;If line low for 3.5ms
     brge start1 ; then wait for start bit
     ; 55x64 us = 3,52 ms
     sbis PIND,INPUT ;If line is
     rjmp detect1 ; low - jump to detect1
     rjmp detect2 ; high - jump to detect2

start1: cpi timerH,8 ;If no start bit detected
        brge exit ;within 130ms then exit

        sbic PIND,INPUT ;Wait for start bit
        rjmp start1

        clr timerL ;Measure length of start bit
        ldi str,0b00000000 ; If lesser then start, then
        ; bit6=1

start2: cpi timerL,17 ;If startbit longer than 1.1ms,
        brge exitstr ; exit

        sbis PIND,INPUT
        rjmp start2

        ;Positive edge of 1st start bit

        mov temp,timerL ;timer is 1/2 bit time
        clr timerL

        mov ref1,temp
        lsr ref1
        mov ref2,ref1
        add ref1,temp ;ref1 = 3/4 bit time
        lsl temp
        add ref2,temp ;ref2 = 5/4 bit time

start3: cp timerL,ref1 ;If high periode St2 > 3/4 bit time
        brge exitstr

        sbic PIND,INPUT ;Wait for falling edge start bit 2

```

## Leht 4 / Lehti 4 Infrapunase TV puldi koodi analüüsiprogramm (eeskuju Atmel AVR410)

```

rjmp start3

clr timerL
ldi bitcnt,12 ;Receive 12 bits
clr command
clr system

sample: cp timerL,ref1 ;Sample INPUT at 1/4 bit time
brlo sample

sbic PIND,INPUT
rjmp bit_is_a_1 ;Jump if line high

bit_is_a_0: clc ;Store a '0'
rol command
rol system

;Synchronize timing
bit_is_a_0a:cp timerL,ref2 ;If no edge within 3/4 bit time
brge exitstr ; exit
sbis PIND,INPUT ;Wait for rising edge
rjmp bit_is_a_0a ;in the middle of the bit

clr timerL
rjmp nextbit

bit_is_a_1: sec ;Store a '1'
rol command
rol system

;Synchronize timing
bit_is_a_1a:cp timerL,ref2 ;If no edge within 3/4 bit time
brge exitstr ; exit
sbic PIND,INPUT ;Wait for falling edge
rjmp bit_is_a_1a ;in the middle of the bit

clr timerL

nextbit: dec bitcnt ;If bitcnt > 0
brne sample ; get next bit

bst system,3 ;Toggle bit läheb
bld command,7 ;command-i 7bit-iks

ser temp ; Inverteerida command
eor command,temp;

; Stardi avastus bit6=1
exitstr: bst str,6
bld command,6

exit: ret

```