

### 10.13. RC5 dedicated decoding routine.

```

|*****
!* Copyright(c)   1998 Philips Electronics N.V.,
!*               Philips Consumer Electronics B.V.,
!*               Advanced Systems & Applications Laboratories,
!*               Eindhoven Remote Control
!*
!* Program       : DC5DEC0.S
!* Language:     : Assembler 8051
!* Creation date  : 20-03-1996
!* Author        : Hans Köhler
!* Function      : RC5 decoding with a 8051 processor running at 12MHz.
!*               Once the decoding routine is entered, it should not be interrupted.
|*****

```

```

                .sect    initseg
                .base    0x00
                sjmp     init

init:           mov     psw,#0x00           !select bank 0
                mov     r0,#0xff
clram:          mov     @r0,#0x00
                djnz    r0,clram
                mov     ie,#0x04           !disable all interrupts
wait:           jb     p3(3),wait          !wait for IR signal

irsignal:      mov     r5,#35              !spike and signal drop-out check
                mov     r6,#63              !drop-out's < 60 us are ignored
                clr     c
filter1:       jb     p3(3),drop_out       !signals < 249 are no RC5/RC6 UAW0422
filter:        dec     r5
                nop
                mov     a,r5               !UAW filter 249 µs check
                jnz    filter1             !(249-5)/7 = 34.86 -> 35
rc5check:      mov     r2,#0x02           !now filter check passed
                mov     r3,#0x04
                mov     r4,#0x01
count:         jb     p3(3),startbit
                inc     r5
                mov     r6,#6
                djnz   r6,.
                mov     a,r5               !2t startbit RC5 enlarged max check
                add    a,#-118             !(2613-255)/20 = 117,9
                jnc    count              ! +118-118 c=1 and +117-118 c=0
                sjmp   out                 ! c=1 > UAW max 2613 µs

drop_out:      mov     a,r6
drop_out1:     jnb    p3(3),filter
                dec     r5
                nop
                add    a,#-9               !UAW spike 60 µs check
                !

```

disturb:	jc sjmp	drop_out1 wait	!if a < 0 then carry = 0 !disturbed signal
startbit:	mov mov add jc mov add jnc setb mov	r0,#0x02 a,r5 a,#-53 rc5enl a,r5 a,#-22 out 0x00 r5,#127	!check longest 1t RC5 startbit !(1306-255)/20 = 52.55 ! +53-53 c=1 and +52-53 c=0 !check shortest 1t RC5 startbit !(676-255)/20 = 21.05 ! +21-22 c=0 +22-22 c=1 !first 1tlowlevelflag RC5startpulse
sample:	mov mov mov djnz jb jnb	a,r4 r6,#4 r6, acc(1),tlow p3(3),trans	!Max count(2053-11)/16=127,64 -> 127 !Sample routine to measure high !and low 1t and 2t timing !Sample loop 16 μs
notrans: out:	djnz ajmp	r5,sample norc5	
first1tlow:	add sjmp	a,#-96 check1tl	!first 1tlow timing check value. !(506-11)/16=30,91 → 31 127-31=96
tlow:	jnb	p3(3),notrans	
trans:	xrl mov mov jb add jc mov add mov jnc xrl mov mov jb	a,#0x02 r4,a a,r5 0x00,first1tlow a,#-85 out a,r5 a,#-58 a,r4 twot a,#0x01 r4,a r5,#126 acc(0),sample	!check shortest 1t !(676-11)/16 = 41.56 → 42 127-42=85 !+84-85 c=0 and +85-85 c=1 !pulse too narrow. !check longest 1t !(1120-11)/16 = 69,31 → 69 127-69 = 58. !+57-58 c=0 and +58-58 c=1 pulse ok. !max(2+2+8+2+2+6+5+5)= 32 μs !(2053-35)/16=126,13 !adaptation due to check-time delay
storedata:	clr cpl jb cpl	c c acc(1),data1 c	!bit = 1 !bit = 0
data1:	mov rlc mov mov jnc inc mov cpl jb mov	a,@r0 a @r0,a r5,#125 sample r0 a,r0 a acc(2),sample r6,#0x00	!adaptation due to check-time delay !max(35+4+6+5)= 50 μs !(2053-50)/16=125,19
checklow:	nop nop		

	djnz	r6,checklow	!1t max UAW-0422 = 1120 $\mu$ s
	mov	r6,#23	
	djnz	r6,.	
	jnb	p3(3),out	!50+1+(4x256)+1+(2x23)+2 = 1124 $\mu$ s
	mov	r6,#139	!check on 3tmax no signal
	djnz	r6,.	!1+(139x2)=279 $\mu$ s
	mov	r7,#3	
waitlow:	mov	r6,#0x00	
finish:	jnb	p3(3),out	!1+{(1+(2+2)x256+2)}x3 = 3082
	djnz	r6,finish	!3082+279=3361 $\mu$ s = 3tmax
	djnz	r7,waitlow	
	ajmp	outdat	!there is an RC5 code detected
twot:	jb	acc(0),out	!check shortest 2tmin
	mov	a,r5	!UAW-0422 = 1520 $\mu$ s
	add	a,#-32	!(1520-11)/16 = 94,31 -> 95
	jc	out	!127-95 = 32
	mov	a,r4	
	sjmp	storedata	
rc5enl:	clr	c	
	mov	a,r5	!check shortest 2t enlarged start-bit
	add	a,#-55	!(1352-255)/20 = 54,85
	jnc	out	!+54-55 c=0 and +55-55 c=1
	mov	r5,#72	!nominal value (58+85)/2 = 71.5 ->72
	mov	a,#0x03	!input values to continue rc5check.
	sjmp	trans	
outdat:	mov	a,r2	!check if we received RC5 enlarged
	jb	acc(6),noenl	
	anl	a,#0x3f	!blank start bit and indication bit
	jb	acc(5),togl1	
bro5:	mov	r5,a	!r5 holds toggle and destination address bits
	mov	a,r3	
	orl	a,#0x40	!set bit 6 for RC5 enlarged field
out2	mov	r6,a	!r6 contains command bits
	ajmp	out1	
togl1:	anl	a,#0x1f	!move toggle to position bit 7
	orl	a,#0x80	
	sjmp	bro5	
noenl:	anl	a,#0x3f	!blank start bit and indication bit
	jb	acc(5),toggl1	
embros:	mov	r5,a	
	mov	a,r3	
	sjmp	out2	
toggl1:	anl	a,#0xdf	!mov toggle to bit 7 position
	orl	a,#0x80	
	sjmp	embros	
norc5:	setb	p1(1)	

mov r6,#0  
djnz r6,.  
clr 0x00  
clr p1(1)  
ajmp wait

!Here RC5 "not correct" indication

out1: setb p1(0)  
mov r6,#0  
djnz r6,.  
clr p1(0)  
ajmp wait

!Here RC5 message correct indication