

This routine accesses the FlashROM chip of Retro Replay in a special way, to find out the chip type and manufacturer. The chip used by the RR reports these using a specific command via its programming interface. Included are useful subroutines to deal with the chip on low level.

See the data sheets to learn how to interpret the information retrieved by this code. Manufacturer: ST = 1, AMD = 20 (decimal). These routines are from my Retro Replay flasher utility FMEEPROMPP. They work in that context, but have not been tested in the extracted form.

```
; Retro Replay manufacturer and device code retrieval by FMan/Tropyx
```

```
; Advanced! Issues the FlashROM an Auto Select command that will  
; make it report its type and manufacturer - for an explanation  
; of the command sequence, refer to the M29F010B data sheet.
```

```
manuf = $fc      ; Manufacturer code  
chip = $fd       ; Chip code
```

```
lda #66  
sta $de01  
ldy #$f0        ; 3rd: F0 - Read/Reset (as a precaution)  
jsr RRCmd  
ldy #$90        ; 3rd: 90 - Auto Select command  
jsr RRCmd  
lda $8000  
sta manuf  
lda $8001  
sta chip
```

```
; flow through to reset the chip again - this is important, because  
; otherwise the FlashROM would remain in a special mode and not  
; working normally - also, the interrupts must be re-enabled
```

```
; this routine sends a Read/Reset command to the FlashROM
```

```
RRReset ldy #$f0      ; 3rd: F0 - and this is enough  
        jsr RRCmd  
        lda #2        ; set the cart to off state  
        sta $de00  
        cli  
        rts
```

```
; subroutine that sends the beginning of a Command sequence
```

```
; this is very handy for communicating with the FlashROM's  
; interface, as all command sequences you'd need will start  
; with this same procedure (see the RR programmer's manual)
```

```
; the first two bytes that go out are $aa at chip address  
; $555 and $55 at $2aa - pass the third one (command byte)  
; that will be written to $555 again, in the Y register
```

```
RRCmd sei
lda #$13
sta $de00
lda #$aa
sta $9555 ; 1st: 555 - AA
ldx #$b
stx $de00
lsr
sta $8aaa ; 2nd: 2AA - 55
lda #$13
sta $de00
sty $9555 ; 3rd: 555 - input reg Y
lda #3
sta $de00
rts
```

From:

<https://codebase64.org/> - **Codebase 64 wiki**

Permanent link:

https://codebase64.org/doku.php?id=base:rr_chip_data

Last update: **2015-04-17 04:33**

