MediaTek MT6227 romloader communication protocol

commented by steve@steve-m.de

http://bb.osmocom.org/trac/wiki/MTKRomloader

```
Port opened by process "Flash_tool.exe" (PID: 1620)

Request: 19.04.2010 13:50:30.03664

 A0 A0 A0 A0 A0 A0 A0 A0 A0 A0 A0 A0 A0 A0 A0
 A0 A0 A0 A0 A0 A0 A0 A0 A0 A0 A0 A0 A0 A0 A0

Answer: 19.04.2010 13:50:34.07264 (+0.0000 seconds)

 5F

Request: 19.04.2010 13:50:34.08264 (+0.0100 seconds)

 0A

Answer: 19.04.2010 13:50:34.08264 (+0.0000 seconds)

 F5

Request: 19.04.2010 13:50:34.09264 (+0.0100 seconds)

 50

Answer: 19.04.2010 13:50:34.09264 (+0.0000 seconds)

 AF

Request: 19.04.2010 13:50:34.10264 (+0.0100 seconds)

 05
```

Activation beacon sent by MTK tool,
waiting for romloader to respond
(push and hold power button)

Answer: 19.04.2010 13:50:34.10264 (+0.0000 seconds)	
FA	
Request: 19.04.2010 13:50:34.23264 (+0.1302 seconds)	
A2	read from memory
Answer: 19.04.2010 13:50:34.28264 (+0.0501 seconds)	
A2	read command ACK
Request: 19.04.2010 13:50:34.28264 (+0.0000 seconds)	
80 00 00 00	Configuration Register: Hardware Version Register
Answer: 19.04.2010 13:50:34.40264 (+0.1202 seconds)	
80 00 00 00	
Request: 19.04.2010 13:50:34.40264 (+0.0000 seconds)	
00 00 00 01	
Answer: 19.04.2010 13:50:34.42264 (+0.0200 seconds)	
00 00 00 01 8A 02	"reset default: 0x8A00", so we have minor revision 0x02 here
Request: 19.04.2010 13:50:35.55264 (+0.1302 seconds)	
A2	read from memory

Answer: 19.04.2010 13:50:35.56264 (+0.0100 seconds)	
A2	read command ACK
Request: 19.04.2010 13:50:35.56264 (+0.0000 seconds)	
80 00 00 08	Hardware Code Register
Answer: 19.04.2010 13:50:35.57264 (+0.0100 seconds)	
80 00 00 08	
Request: 19.04.2010 13:50:35.57264 (+0.0000 seconds)	
00 00 00 01	
Answer: 19.04.2010 13:50:35.58264 (+0.0100 seconds)	
00 00 00 01 62 27	0x6227 - we have a MediaTek MT6227
Request: 19.04.2010 13:50:35.58264 (+0.0000 seconds)	
A2	read from memory
Answer: 19.04.2010 13:50:35.59264 (+0.0100 seconds)	
A2	read command ACK
Request: 19.04.2010 13:50:35.59264 (+0.0000 seconds)	
80 00 00 04	Software Version Register

Answer: 19.04.2010 13:50:35.60364 (+0.0100 seconds)	
`80 00 00 04`	
Request: 19.04.2010 13:50:35.60364 (+0.0000 seconds)	
`00 00 00 01`	
Answer: 19.04.2010 13:50:35.62364 (+0.0200 seconds)	
`00 00 00 01 8A 01`	"reset default: 0x8a00", so this is minor revision 0x01
Request: 19.04.2010 13:50:35.85364 (+0.2303 seconds)	
`A1`	write to register
Answer: 19.04.2010 13:50:35.86364 (+0.0100 seconds)	
`A1`	write command ack
Request: 19.04.2010 13:50:35.87364 (+0.0100 seconds)	
`80 04 00 00`	Reset Generation Unit (RGU): Watchdog Timer Control Register
Answer: 19.04.2010 13:50:35.88364 (+0.0100 seconds)	
`80 04 00 00`	
Request: 19.04.2010 13:50:35.89364 (+0.0100 seconds)	
`00 00 00 01`	

Answer: 19.04.2010 13:50:35.90364 (+0.0100 seconds)	
00 00 00 01	
Request: 19.04.2010 13:50:35.91364 (+0.0100 seconds)	
22 00	data to register: 0x2200 MSB of register is "KEY", "write access is allowed if KEY=0x22" 0x00: Disable watchdog
Answer: 19.04.2010 13:50:35.92364 (+0.0100 seconds)	
22 00	
Request: 19.04.2010 13:50:35.93364 (+0.0100 seconds)	
A2	read
Answer: 19.04.2010 13:50:35.94364 (+0.0100 seconds)	
A2	read ack
Request: 19.04.2010 13:50:35.95364 (+0.0100 seconds)	
80 21 00 50	Real Time Clock Register: RTC_POWERKEY1 register
Answer: 19.04.2010 13:50:35.96364 (+0.0100 seconds)	
80 21 00 50	If the RTC has been set to a correct time this register holds 0xA357
Request: 19.04.2010 13:50:35.97364 (+0.0100 seconds)	
00 00 00 01	

Answer: 19.04.2010 13:50:35.98364 (+0.0100 seconds)	
00 00 00 01 A3 57	0xA357, valid time is set (by the phone fw)
Request: 19.04.2010 13:50:35.99364 (+0.0100 seconds)	
A2	
Answer: 19.04.2010 13:50:35.00364 (+0.0100 seconds)	
A2	
Request: 19.04.2010 13:50:35.01364 (+0.0100 seconds)	
80 21 00 54	Real Time Clock: RTC_POWERKEY2 register same as RTC_POWERKEY1, but has to hold 0x67D2 if time is valid
Answer: 19.04.2010 13:50:35.02364 (+0.0100 seconds)	
80 21 00 54	
Request: 19.04.2010 13:50:35.03364 (+0.0100 seconds)	
00 00 00 01	
Answer: 19.04.2010 13:50:35.04364 (+0.0100 seconds)	
00 00 00 01 67 D2	valid time
Request: 19.04.2010 13:50:35.05364 (+0.0100 seconds)	
A2	read

Answer: 19.04.2010 13:50:35.06364 (+0.0100 seconds)	
A2	read ack
Request: 19.04.2010 13:50:35.07364 (+0.0100 seconds)	
80 21 00 10	Real Time Clock: RTC_ALARM_MASK register
Answer: 19.04.2010 13:50:35.08364 (+0.0100 seconds)	
80 21 00 10	
Request: 19.04.2010 13:50:35.09364 (+0.0100 seconds)	
00 00 00 01	
Answer: 19.04.2010 13:50:35.10364 (+0.0100 seconds)	
00 00 00 01 00 00	0x0000, all RTC interrupts activated
Request: 19.04.2010 13:50:35.11364 (+0.0100 seconds)	
A1	write
Answer: 19.04.2010 13:50:35.12364 (+0.0100 seconds)	
A1	write ack
Request: 19.04.2010 13:50:35.13364 (+0.0100 seconds)	
80 21 00 10	Real Time Clock Register: RTC_ALARM_MASK register

Answer: 19.04.2010 13:50:35.14364 (+0.0100 seconds)	
80 21 00 10	
Request: 19.04.2010 13:50:35.15364 (+0.0100 seconds)	
00 00 00 01	
Answer: 19.04.2010 13:50:35.16364 (+0.0100 seconds)	
00 00 00 01	
Request: 19.04.2010 13:50:35.17364 (+0.0100 seconds)	
00 40	set RTC interrupt mask to 0x0040, this masks RTC_TC_MTH, so no interrupt is generated when the month changes (why?)
Answer: 19.04.2010 13:50:35.18364 (+0.0100 seconds)	
00 40	
Request: 19.04.2010 13:50:35.19364 (+0.0100 seconds)	
A2	read
Answer: 19.04.2010 13:50:35.20364 (+0.0100 seconds)	
A2	read ack
Request: 19.04.2010 13:50:35.21364 (+0.0100 seconds)	
80 21 00 10	once again, read RTC_ALARM_MASK register

Answer: 19.04.2010 13:50:35.22364 (+0.0100 seconds)	
80 21 00 10	
Request: 19.04.2010 13:50:35.23364 (+0.0100 seconds)	
00 00 00 01	
Answer: 19.04.2010 13:50:35.24364 (+0.0100 seconds)	
00 00 00 01 00 40	0x0040, the value it was set to earlier
Request: 19.04.2010 13:50:35.25364 (+0.0100 seconds)	
A1	write
Answer: 19.04.2010 13:50:35.26364 (+0.0100 seconds)	
A1	write ack
Request: 19.04.2010 13:50:35.27364 (+0.0100 seconds)	
80 21 00 10	write to RTC_ALARM_MASK register again
Answer: 19.04.2010 13:50:35.28364 (+0.0100 seconds)	
80 21 00 10	
Request: 19.04.2010 13:50:35.29464 (+0.0100 seconds)	
00 00 00 01	

Answer: 19.04.2010 13:50:35.30464 (+0.0100 seconds)	
00 00 00 01	
Request: 19.04.2010 13:50:35.31464 (+0.0100 seconds)	
00 00	reset mask to 0x0000, this means interrupt on year, month, week and so on is generated
Answer: 19.04.2010 13:50:35.32464 (+0.0100 seconds)	
00 00	
Request: 19.04.2010 13:50:35.33464 (+0.0100 seconds)	
A1	write
Answer: 19.04.2010 13:50:35.34464 (+0.0100 seconds)	
A1	write ack
Request: 19.04.2010 13:50:35.35464 (+0.0100 seconds)	
80 21 00 00	Real Time Clock: Baseband power up register
Answer: 19.04.2010 13:50:35.36464 (+0.0100 seconds)	
80 21 00 00	
Request: 19.04.2010 13:50:35.37464 (+0.0100 seconds)	
00 00 00 01	

Answer: 19.04.2010 13:50:35.38464 (+0.0100 seconds) 00 00 00 01	
Request: 19.04.2010 13:50:35.39464 (+0.0100 seconds) 43 0E	Page 164: set register to 0x430E, whereas 0x43 is the "Baseband powerup key" or "KEY_BBPU" 0x0E means AUTO powerup enable, enable Baseband powerup of PMIC, and enable RTC write interface, as well as disable RTC alarm
Answer: 19.04.2010 13:50:35.40464 (+0.0100 seconds) 43 0E	
Request: 19.04.2010 13:50:35.41464 (+0.0100 seconds) A1	write
Answer: 19.04.2010 13:50:35.42464 (+0.0100 seconds) A1	write ack
Request: 19.04.2010 13:50:35.43464 (+0.0100 seconds) 80 21 00 08	Real Time Clock Register: RTC IRQ enable register
Answer: 19.04.2010 13:50:35.44464 (+0.0100 seconds) 80 21 00 08	
Request: 19.04.2010 13:50:35.45464 (+0.0100 seconds) 00 00 00 01	

Answer: 19.04.2010 13:50:35.46464 (+0.0100 seconds)	
00 00 00 01	
Request: 19.04.2010 13:50:35.47464 (+0.0100 seconds)	
00 00	0x00, disable all RTC interrupts
Answer: 19.04.2010 13:50:35.48464 (+0.0100 seconds)	
00 00	
Request: 19.04.2010 13:50:35.49464 (+0.0100 seconds)	
A1	write
Answer: 19.04.2010 13:50:36.50464 (+0.0100 seconds)	
A1	write ack
Request: 19.04.2010 13:50:36.51464 (+0.0100 seconds)	**now it masks the RTC setting as valid:**
80 21 00 50	Real Time Clock: RTC_POWERKEY1 register
Answer: 19.04.2010 13:50:36.52464 (+0.0100 seconds)	
80 21 00 50	
Request: 19.04.2010 13:50:36.53464 (+0.0100 seconds)	
00 00 00 01	

Answer: 19.04.2010 13:50:36.54464 (+0.0100 seconds)	
00 00 00 01	
Request: 19.04.2010 13:50:36.55464 (+0.0100 seconds)	
A3 57	set the POWERKEY1
Answer: 19.04.2010 13:50:36.56464 (+0.0100 seconds)	
A3 57	
Request: 19.04.2010 13:50:36.57464 (+0.0100 seconds)	
A1	write
Answer: 19.04.2010 13:50:36.58464 (+0.0100 seconds)	
A1	write ack
Request: 19.04.2010 13:50:36.59464 (+0.0100 seconds)	
80 21 00 54	Real Time Clock Register: RTC_POWERKEY2 register
Answer: 19.04.2010 13:50:36.61464 (+0.0200 seconds)	
80 21 00 54	
Request: 19.04.2010 13:50:36.62464 (+0.0100 seconds)	
00 00 00 01	

Answer: 19.04.2010 13:50:36.63464 (+0.0100 seconds)	
00 00 00 01	
Request: 19.04.2010 13:50:36.64464 (+0.0100 seconds)	
67 D2	set the POWERKEY2
Answer: 19.04.2010 13:50:36.65464 (+0.0100 seconds)	
67 D2	
Request: 19.04.2010 13:50:36.89464 (+0.2403 seconds)	
A1	write
Answer: 19.04.2010 13:50:36.90464 (+0.0100 seconds)	
A1	write ack
Request: 19.04.2010 13:50:36.91464 (+0.0100 seconds)	
80 13 00 38	UART1 Register: UART1_AUTOBAUDSAMPLE
Answer: 19.04.2010 13:50:36.92464 (+0.0100 seconds)	
80 13 00 38	
Request: 19.04.2010 13:50:36.93464 (+0.0100 seconds)	
00 00 00 01	

Answer: 19.04.2010 13:50:36.94464 (+0.0100 seconds) 00 00 00 01	

Request: 19.04.2010 13:50:36.95464 (+0.0100 seconds)
00 0D
	AUTOBAUDSAMPLE = 0x0D = 13 dec.
	"When systemclock = 13MHz,
	autobaudsample= 6;
	when system clock = 26MHz,
	autobaudsample= 13."

Answer: 19.04.2010 13:50:36.96464 (+0.0100 seconds) 00 0D	
Request: 19.04.2010 13:50:36.97464 (+0.0100 seconds) A1	write
Answer: 19.04.2010 13:50:36.98464 (+0.0100 seconds) A1	write ack
Request: 19.04.2010 13:50:36.99564 (+0.0100 seconds) 80 13 00 20	UART1 Register: UART1_AUTOBAUD_EN
Answer: 19.04.2010 13:50:36.00564 (+0.0100 seconds) 80 13 00 20	
Request: 19.04.2010 13:50:36.01564 (+0.0100 seconds) 00 00 00 01	

Answer: 19.04.2010 13:50:36.02564 (+0.0100 seconds)	
00 00 00 01	
Request: 19.04.2010 13:50:36.03564 (+0.0100 seconds)	
00 01	0x01: Auto Baud Function enable
Answer: 19.04.2010 13:50:36.04564 (+0.0100 seconds)	
00 01	
Request: 19.04.2010 13:50:36.15564 (+0.1102 seconds)	
41 54	"AT" seems to be a common auto-baud sample (also seen on Calypso)
Answer: 19.04.2010 13:50:36.16564 (+0.0100 seconds)	
41 54	
Request: 19.04.2010 13:50:36.17564 (+0.0100 seconds)	
FF	
Answer: 19.04.2010 13:50:36.17564 (+0.0000 seconds)	
FF	
Request: 19.04.2010 13:50:36.18564 (+0.0100 seconds)	
FE	

Answer: 19.04.2010 13:50:36.18564 (+0.0000 seconds)	
FE	
Request: 19.04.2010 13:50:36.19564 (+0.0100 seconds)	
A1	write
Answer: 19.04.2010 13:50:36.20564 (+0.0100 seconds)	
A1	write ack
Request: 19.04.2010 13:50:36.21564 (+0.0100 seconds)	
80 04 00 00	Reset Generation Unit (RGU): Watchdog Timer Control Register
Answer: 19.04.2010 13:50:36.22564 (+0.0100 seconds)	
80 04 00 00	
Request: 19.04.2010 13:50:36.23564 (+0.0100 seconds)	
00 00 00 01	
Answer: 19.04.2010 13:50:36.24564 (+0.0100 seconds)	
00 00 00 01	
Request: 19.04.2010 13:50:36.25564 (+0.0100 seconds)	
22 00	data to register: 0x2200 MSB of register is "KEY", "write access is allowed if KEY=0x22" 0x00: disable watchdog

Answer: 19.04.2010 13:50:36.26564 (+0.0100 seconds)	
22 00	
Request: 19.04.2010 13:50:36.27564 (+0.0100 seconds)	
A1	write
Answer: 19.04.2010 13:50:36.28564 (+0.0100 seconds)	
A1	write ack
Request: 19.04.2010 13:50:36.29564 (+0.0100 seconds)	
80 01 00 40	External Memory Interface register: Register 0 for MobileRAM
Answer: 19.04.2010 13:50:36.30564 (+0.0100 seconds)	
80 01 00 40	
Request: 19.04.2010 13:50:36.31564 (+0.0100 seconds)	
00 00 00 01	
Answer: 19.04.2010 13:50:36.32564 (+0.0100 seconds)	
00 00 00 01	
Request: 19.04.2010 13:50:36.33564 (+0.0100 seconds)	this is a 32bit register, so if this is the most significant register word that is being written, it means: chip select 2 is used for MobileRAM
00 02	

Answer: 19.04.2010 13:50:36.34564 (+0.0100 seconds) 00 02	
Request: 19.04.2010 13:50:36.35564 (+0.0100 seconds) AE	32bit register write
Answer: 19.04.2010 13:50:36.36564 (+0.0100 seconds) AE	32bit register write ack
Request: 19.04.2010 13:50:36.37564 (+0.0100 seconds) 80 01 00 00	External Memory Interface register: EMI Control for BANK 0
Answer: 19.04.2010 13:50:36.38564 (+0.0100 seconds) 80 01 00 00	
Request: 19.04.2010 13:50:36.39564 (+0.0100 seconds) 00 00 00 01	
Answer: 19.04.2010 13:50:36.40564 (+0.0100 seconds) 00 00 00 01	
Request: 19.04.2010 13:50:36.42564 (+0.0200 seconds) 44 C0 42 03	Memory and Waitstate configuration: Page 101+

Answer: 19.04.2010 13:50:36.43564 (+0.0100 seconds)	
44 C0 42 03	
Request: 19.04.2010 13:50:36.44564 (+0.0100 seconds)	
AE	32bit register write
Answer: 19.04.2010 13:50:36.45564 (+0.0100 seconds)	
AE	32bit register write ack
Request: 19.04.2010 13:50:36.46564 (+0.0100 seconds)	
80 01 00 08	External Memory Interface register: EMI Control for BANK 1
Answer: 19.04.2010 13:50:36.47564 (+0.0100 seconds)	
80 01 00 08	
Request: 19.04.2010 13:50:36.48564 (+0.0100 seconds)	
00 00 00 01	
Answer: 19.04.2010 13:50:36.49564 (+0.0100 seconds)	
00 00 00 01	
Request: 19.04.2010 13:50:37.50564 (+0.0100 seconds)	
44 C0 42 03	Memory and Waitstate configuration: Page 101+

Answer: 19.04.2010 13:50:37.51564 (+0.0100 seconds)	
44 C0 42 03	
Request: 19.04.2010 13:50:37.51564 (+0.0000 seconds)	
A1	write
Answer: 19.04.2010 13:50:37.52564 (+0.0100 seconds)	
A1	write ack
Request: 19.04.2010 13:50:37.52564 (+0.0000 seconds)	
40 00 14 00	
Answer: 19.04.2010 13:50:37.53564 (+0.0100 seconds)	
40 00 14 00	this is a memory location in the internal SRAM (which starts at 0x40000000) (seems to be the address where the binary is uploaded to)
Request: 19.04.2010 13:50:37.53564 (+0.0000 seconds)	
00 00 9E 20	write 0x9E20 there (length of the payload to follow in 16 bit words, 40.480 words)
Answer: 19.04.2010 13:50:37.54564 (+0.0100 seconds)	
00 00 9E 20	It now transmits 79 blocks with 1024bytes + one block with 64byte = 80960 bytes / 2 = 40.480 16 Bit words, this is our payload length

Request: 19.04.2010 13:50:37.62564 (+0.0801 seconds)

```
FF FF EA FF 00 00 E1 0F 10 C0 E3 A0 00 01 E1 80
F0 00 E1 2F 14 40 E3 A0 1D 50 E2 81 10 04 E2 41
D0 01 E1 A0 3C 1F EB 00 30 00 E5 9F FF 13 E1 2F
14 39 40 00 20 01 47 70 B5 80 48 05 4A 06 68 00
68 12 49 04 1A 80 60 08 F0 02 FA DD BD 80 00 00
02 BC 40 01 02 BC 40 01 14 00 40 00 21 00 E0 00
31 01 42 88 D8 FC 47 70 48 E3 47 70 B5 70 6D C2
1C 06 6A D4 6A 51 1D 25 CD 24 36 50 68 23 68 09
7A 34 2C 04 D1 01 20 00 BD 70 2C 02 D1 05 24 13
02 E4 80 0C 24 08 80 14 E0 03 2C 01 D1 0B 02 E4
80 0C 24 87 80 1C 24 07 80 1C 23 01 03 9B 80 0B
21 00 09 5B E0 04 48 D0 BD 70 31 01 04 09 0C 09
42 99 D3 FA 49 CD 80 29 88 11 24 02 43 21 80 11
21 00 E0 02 31 01 04 09 0C 09 42 99 D3 FA 21 04
72 31 6D C0 4C C6 78 01 48 C6 78 00 6A E2 F0 0D
FD DD 6A A0 F0 0D FD D8 E7 C5 42 88 D9 11 18 80
38 01 18 89 39 01 E0 04 78 0B 39 01 70 03 38 01
3A 01 2A 00 D8 F8 E0 06 78 0B 31 01 70 03 30 01
3A 01 2A 00 D8 F8 20 00 47 70 18 82 E0 01 70 01
30 01 42 90 D3 FB 47 70 B5 FF 1C 0F 1C 05 1C 16
22 0F A1 B1 B0 81 F7 FF FF D8 19 AC 19 EF E0 07
22 0F 1C 20 A1 AC F0 0D FD B7 28 00 D0 12 19 A4
42 A7 D8 F5 E0 0E 22 0D 1C 20 A1 AB 38 0D F7 FF
FF C4 22 0D 1C 20 A1 A8 38 0D F0 0D FD A5 28 00
D0 02 1B A4 42 AC D8 EE 42 AC D9 01 42 A7 D8 1B
20 00 68 29 68 2A 42 91 D0 03 48 97 30 02 B0 05
BD F0 30 01 28 05 D3 F4 68 29 43 CB 20 00 60 2B
68 2A 42 8A D0 02 48 90 30 04 E7 F0 30 01 28 05
D3 F5 48 8D 30 03 E7 EA 98 04 28 00 D0 02 99 04
1B 60 60 08 20 00 E7 E2 B5 F1 4F 93 B0 81 6D FB
68 DE 36 44 CE 07 3E 50 46 94 6D B2 6D F5 6D 74
95 00 6E 35 36 40 78 36 2E 00 D1 02 48 7E 30 01
BD FC 78 1E 4B 88 6A DB 2E 07 60 0B D1 00 E0 02
6B 39 46 63 60 19 6A B9 60 01 20 7D 01 00 F7 FF
FF 1D 6C 78 99 00 60 08 6C B8 60 28 21 03 6B B8
04 09 43 88 99 01 04 0E 43 30 60 20 20 C8 F7 FF
FF 0D 20 01 60 10 20 C8 F7 FF FF 08 21 03 6B F8
04 09 43 88 43 30 60 20 20 04 60 10 20 C8 F7 FF
FE FD 21 03 6B B8 04 09 43 88 43 30 60 20 20 04
60 10 20 C8 F7 FF FE F2 24 00 26 02 60 16 20 C8
F7 FF FE EC 34 01 2C 08 D3 F8 21 01 6C B8 07 C9
43 08 60 28 20 C8 F7 FF FE E1 20 00 E7 B0 29 00
D0 06 4A 61 00 C0 6D D2 68 D2 18 10 68 40 60 01
20 00 47 70 29 13 D0 12 DC 09 29 07 D0 05 29 09
D0 03 29 0C D0 01 29 0D D1 1B 49 58 E0 18 29 14
D0 12 29 16 D0 12 29 17 D0 10 29 19 D1 11 4B 54
6B 59 4A 54 42 91 D1 01 49 53 E0 09 6B 59 4A 53
42 91 D1 06 49 52 E0 03 49 52 E0 01 49 4E 31 CF
E7 CD 47 70 B5 FF 48 50 B0 83 68 01 68 40 91 01
99 03 90 02 20 00 70 08 99 04 4F 43 70 08 99 05
26 00 60 08 9B 06 60 18 6D F8 68 C1 31 40 78 09
29 00 D1 03 48 30 30 01 B0 07 BD F0 6A C0 68 C1
1C 38 F0 0D FC B2 1C 05 D1 4E 24 00 48 3F 23 24
88 00 49 3F 68 09 43 58 5A 08 49 3E 42 88 D0 09
A8 01 5D 00 78 39 42 88 D0 02 78 79 42 88 D1 01
26 01 E0 3C A8 01 5D 00 F7 FF FF 2E 1C 05 D1 33
A8 01 5D 00 25 08 28 08 D0 03 6D F9 00 C0 68 C9
58 0D 22 0F 21 00 1C 28 F7 FF FE C7 22 0F 1C 28
A1 19 F7 FF FE AA 22 0F 1C 28 A1 17 F0 0D FC 8C
28 00 D1 1A 99 03 20 02 70 08 A8 01 5D 00 99 04
22 01 70 08 99 05 05 12 60 0D 01 D1 1C 28 9B 06
F7 FF FE B2 1C 05 2C 01 D1 06 2E 00 D0 04 6D F8
78 01 20 00 F7 FF FF 66 E0 04 25 2F 01 AD 34 01
2C 02 D3 AB 1C 28 E7 9F 0B BB 00 00 07 03 00 00
```

This is the first block sent to memory, 1024 Bytes each

Answer: 19.04.2010 13:50:37.64564 (+0.0200 seconds)

```
FF FF EA FF 00 00 E1 0F 10 C0 E3 A0 00 01 E1 80
F0 00 E1 2F 14 40 E3 A0 1D 50 E2 81 10 04 E2 41
D0 01 E1 A0 3C 1F EB 00 30 00 E5 9F FF 13 E1 2F
14 39 40 00 20 01 47 70 B5 80 48 05 4A 06 68 00
68 12 49 04 1A 80 60 08 F0 02 FA DD BD 80 00 00
02 BC 40 01 02 BC 40 01 14 00 40 00 21 00 E0 00
31 01 42 88 D8 FC 47 70 48 E3 47 70 B5 70 6D C2
1C 06 6A D4 6A 51 1D 25 CD 24 36 50 68 23 68 09
7A 34 2C 04 D1 01 20 00 BD 70 2C 02 D1 05 24 13
02 E4 80 0C 24 08 80 14 E0 03 2C 01 D1 0B 02 E4
80 0C 24 87 80 1C 24 07 80 1C 23 01 03 9B 80 0B
21 00 09 5B E0 04 48 D0 BD 70 31 01 04 09 0C 09
42 99 D3 FA 49 CD 80 29 88 11 24 02 43 21 80 11
21 00 E0 02 31 01 04 09 0C 09 42 99 D3 FA 21 04
72 31 6D C0 4C C6 78 01 48 C6 78 00 6A E2 F0 0D
FD DD 6A A0 F0 0D FD D8 E7 C5 42 88 D9 11 18 80
38 01 18 89 39 01 E0 04 78 0B 39 01 70 03 38 01
3A 01 2A 00 D8 F8 E0 06 78 0B 31 01 70 03 30 01
3A 01 2A 00 D8 F8 20 00 47 70 18 82 E0 01 70 01
30 01 42 90 D3 FB 47 70 B5 FF 1C 0F 1C 05 1C 16
22 0F A1 B1 B0 81 F7 FF FF D8 19 AC 19 EF E0 07
22 0F 1C 20 A1 AC F0 0D FD B7 28 00 D0 12 19 A4
42 A7 D8 F5 E0 0E 22 0D 1C 20 A1 AB 38 0D F7 FF
FF C4 22 0D 1C 20 A1 A8 38 0D F0 0D FD A5 28 00
D0 02 1B A4 42 AC D8 EE 42 AC D9 01 42 A7 D8 1B
20 00 68 29 68 2A 42 91 D0 03 48 97 30 02 B0 05
BD F0 30 01 28 05 D3 F4 68 29 43 CB 20 00 60 2B
68 2A 42 8A D0 02 48 90 30 04 E7 F0 30 01 28 05
D3 F5 48 8D 30 03 E7 EA 98 04 28 00 D0 02 99 04
1B 60 60 08 20 00 E7 E2 B5 F1 4F 93 B0 81 6D FB
68 DE 36 44 CE 07 3E 50 46 94 6D B2 6D F5 6D 74
95 00 6E 35 36 40 78 36 2E 00 D1 02 48 7E 30 01
BD FC 78 1E 4B 88 6A DB 2E 07 60 0B D1 00 E0 02
6B 39 46 63 60 19 6A B9 60 01 20 7D 01 00 F7 FF
FF 1D 6C 78 99 00 60 08 6C B8 60 28 21 03 6B B8
04 09 43 88 99 01 04 0E 43 30 60 20 20 C8 F7 FF
FF 0D 20 01 60 10 20 C8 F7 FF FF 08 21 03 6B F8
04 09 43 88 43 30 60 20 20 04 60 10 20 C8 F7 FF
FE FD 21 03 6B B8 04 09 43 88 43 30 60 20 20 04
60 10 20 C8 F7 FF FE F2 24 00 26 02 60 16 20 C8
F7 FF FE EC 34 01 2C 08 D3 F8 21 01 6C B8 07 C9
43 08 60 28 20 C8 F7 FF FE E1 20 00 E7 B0 29 00
D0 06 4A 61 00 C0 6D D2 68 D2 18 10 68 40 60 01
20 00 47 70 29 13 D0 12 DC 09 29 07 D0 05 29 09
D0 03 29 0C D0 01 29 0D D1 1B 49 58 E0 18 29 14
D0 12 29 16 D0 12 29 17 D0 10 29 19 D1 11 4B 54
6B 59 4A 54 42 91 D1 01 49 53 E0 09 6B 59 4A 53
42 91 D1 06 49 52 E0 03 49 52 E0 01 49 4E 31 CF
E7 CD 47 70 B5 FF 48 50 B0 83 68 01 68 40 91 01
99 03 90 02 20 00 70 08 99 04 4F 43 70 08 99 05
26 00 60 08 9B 06 60 18 6D F8 68 C1 31 40 78 09
29 00 D1 03 48 30 30 01 B0 07 BD F0 6A C0 68 C1
1C 38 F0 0D FC B2 1C 05 D1 4E 24 00 48 3F 23 24
88 00 49 3F 68 09 43 58 5A 08 49 3E 42 88 D0 09
A8 01 5D 00 78 39 42 88 D0 02 78 79 42 88 D1 01
26 01 E0 3C A8 01 5D 00 F7 FF FF 2E 1C 05 D1 33
A8 01 5D 00 25 08 28 08 D0 03 6D F9 00 C0 68 C9
58 0D 22 0F 21 00 1C 28 F7 FF FE C7 22 0F 1C 28
A1 19 F7 FF FE AA 22 0F 1C 28 A1 17 F0 0D FC 8C
28 00 D1 1A 99 03 20 02 70 08 A8 01 5D 00 99 04
22 01 70 08 99 05 05 12 60 0D 01 D1 1C 28 9B 06
F7 FF FE B2 1C 05 2C 01 D1 06 2E 00 D0 04 6D F8
78 01 20 00 F7 FF FF 66 E0 04 25 2F 01 AD 34 01
2C 02 D3 AB 1C 28 E7 9F 0B BB 00 00 07 03 00 00
```

The loader repeats every single received block, and sends it back to the host

[Blocks in between skipped]

Request: 19.04.2010 13:50:45.81764 (+0.0200 seconds)

```
00 00 00 00 00 00 00 00 00 00 00 00 00 04 00 00
82 41 40 01 89 15 40 01 00 00 00 00 82 E7 40 01
83 37 40 01 89 19 40 01 89 1D 40 01 89 21 40 01
89 B1 40 01 89 B5 40 01 89 B9 40 01 89 BF 40 01
```

last block: 64 Bytes

Answer: 19.04.2010 13:50:45.82764 (+0.0100 seconds)

```
00 00 00 00 00 00 00 00 00 00 00 00 00 04 00 00
82 41 40 01 89 15 40 01 00 00 00 00 82 E7 40 01
83 37 40 01 89 19 40 01 89 1D 40 01 89 21 40 01
89 B1 40 01 89 B5 40 01 89 B9 40 01 89 BF 40 01
```

Request: 19.04.2010 13:50:45.05864 (+0.2303 seconds)

```
A4
```

maybe a command to get a checksum

Answer: 19.04.2010 13:50:45.06864 (+0.0100 seconds)

```
A4
```

Request: 19.04.2010 13:50:45.06864 (+0.0000 seconds)

```
40 00 14 00
```

address of the uploaded code

Answer: 19.04.2010 13:50:45.07864 (+0.0100 seconds)

```
40 00 14 00
```

Request: 19.04.2010 13:50:45.08864 (+0.0100 seconds)

```
00 00 9E 20
```

Answer: 19.04.2010 13:50:45.09864 (+0.0100 seconds)	
00 00 9E 20 10 64	size + checksum from phone?
Request: 19.04.2010 13:50:45.21864 (+0.0200 seconds)	
A8	this has to be the branch command
Answer: 19.04.2010 13:50:45.22864 (+0.0100 seconds)	
A8	
Request: 19.04.2010 13:50:46.26964 (+1.0415 seconds)	
40 00 14 00	branch address
Answer: 19.04.2010 13:50:46.27964 (+0.0100 seconds)	
40 00 14 00 C0 03 02 08	branch command successfull?
Request: 19.04.2010 13:50:46.27964 (+0.0000 seconds)	
FF FE 00 08 00 00 00 07 FF 02	now the uploaded flash-loader seems to take over the serial communication
Answer: 19.04.2010 13:50:46.32964 (+0.0100 seconds)	
00 00 00 00 00 08 00 99 02 00 00 00 00 EC 22 7E 22 63 22 60 00 00 0B FD 00 00 00 00 00 00 0B C4 00 FF FF 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 04 70 00 00 00 00 00 01 01 00 80 00 00 5A 00 00 0C 05 00 00 04 07 C1	The command that was used to create this dump was: Read back, address 0x01F00000, length 0x000F000

```
Request: 19.04.2010 13:50:46.36964 (+0.0100 seconds)

 D2 04 01

Answer: 19.04.2010 13:50:46.36964 (+0.0000 seconds)

 5A

Request: 19.04.2010 13:50:47.51064 (+0.1402 seconds)

 C0

Answer: 19.04.2010 13:50:47.51064 (+0.0000 seconds)

 C0

Request: 19.04.2010 13:50:47.56064 (+0.0501 seconds)

 5A

Answer: 19.04.2010 13:50:47.60064 (+0.0401 seconds)

 5A

Request: 19.04.2010 13:50:47.60064 (+0.0000 seconds)

 00

Answer: 19.04.2010 13:50:47.60064 (+0.0000 seconds)

 00
```

```
Request: 19.04.2010 13:50:47.61064 (+0.0100 seconds)

 01

Answer: 19.04.2010 13:50:47.61064 (+0.0000 seconds)

 01

Request: 19.04.2010 13:50:47.62064 (+0.0100 seconds)

 02

Answer: 19.04.2010 13:50:47.62064 (+0.0000 seconds)

 02
```

[cut, this pattern continues up to 0xff]

```
Request: 19.04.2010 13:50:49.78364 (+0.0100 seconds)

 FE

Answer: 19.04.2010 13:50:49.78364 (+0.0000 seconds)

 FE

Request: 19.04.2010 13:50:49.79364 (+0.0100 seconds)

 FF

Answer: 19.04.2010 13:50:49.79364 (+0.0000 seconds)

 FF
```

Request: 19.04.2010 13:50:49.79364 (+0.0000 seconds)

```
F0
```

Answer: 19.04.2010 13:50:49.80364 (+0.0100 seconds)

```
00 00 00 00 00 01 00 00 00 01 00 00 00
```

Request: 19.04.2010 13:50:49.82364 (+0.0200 seconds)

```
D6 01 F0 00 00 00 00 F0 00
```

Answer: 19.04.2010 13:50:49.83364 (+0.0000 seconds)

```
5A
```

Request: 19.04.2010 13:50:49.83364 (+0.0000 seconds)

```
00 00 04 00
```

Answer: 19.04.2010 13:50:49.92364 (+0.0901 seconds)

```
01 00 00 AF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
```

this is the content of the flash that is dumped

```
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
F8 B4
```

Request: 19.04.2010 13:50:49.93364 (+0.0100 seconds)

```
5A
```

Answer: 19.04.2010 13:50:49.02364 (+0.0901 seconds)

```
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FC 00
```

Request: 19.04.2010 13:50:55.07264 (+0.0100 seconds)

```
5A D9
```

Answer: 19.04.2010 13:50:55.07264 (+0.0000 seconds)

```
5A
```

```
Port closed
```