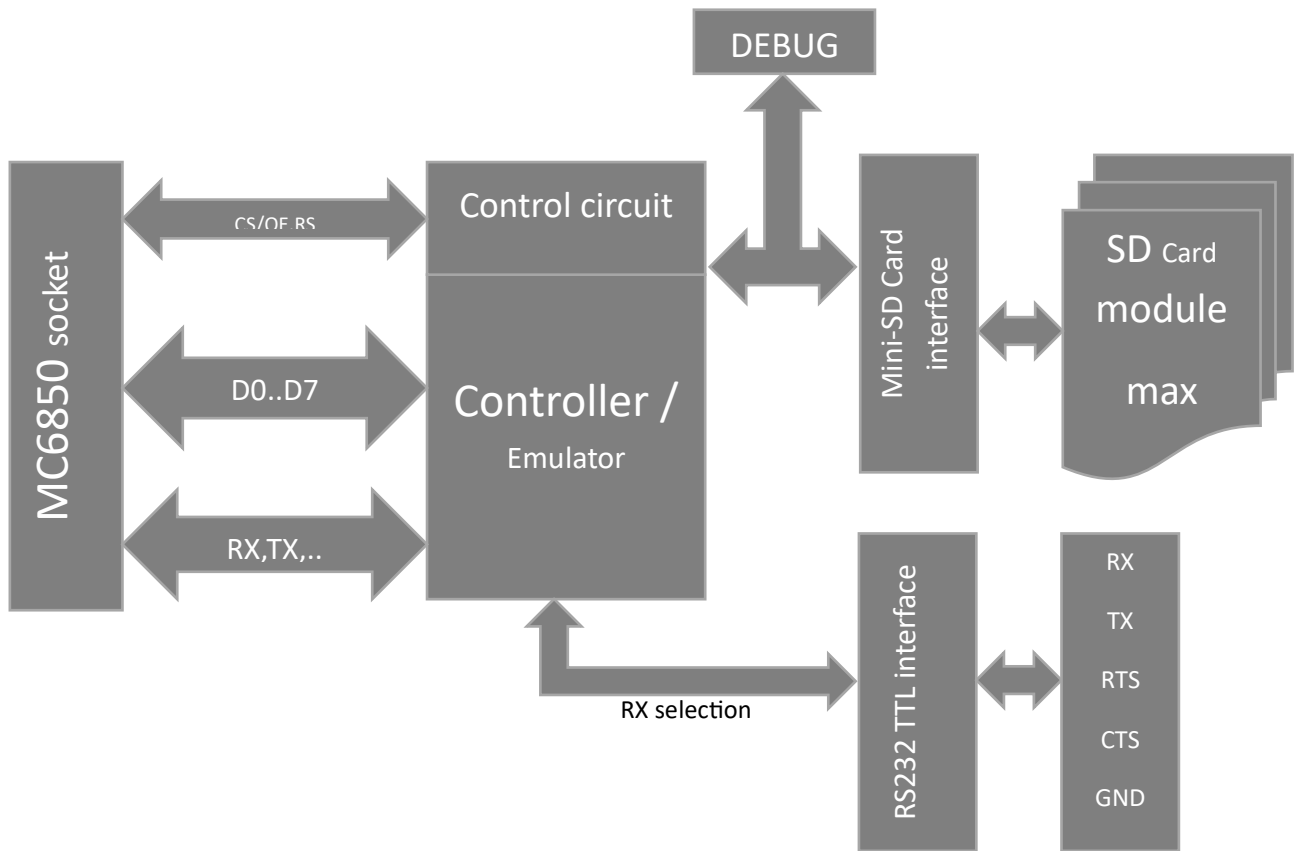


ACIA Emulation - SD CARD REFERENCE (V1.36 12-2025)

Functional diagram:



How to access extended ACIA Control register:

- 1- Master reset with \$03 to ACIA control register (normally not needed (OPTIONAL))
- 2- Set ACIA functions (Baudrate, bits, RTS, CTS...) to ACIA control register
If ACIA special function CR5, CR6 = 11 is set, a byte has to follow to specify the extended function (see tables below)
- 3- The extended function consists of: Extended command byte (low nibble),
Extended function (high nibble)

Example: byte \$07 will call extended function 0 and extended command 7 (= Set FIFO to 8, no return value.

Extended command:

Note: "no return" value means the ACIA data port will not return a status or data byte.
In case a byte/word/string is returned, the program "must" read these or the ACIA will perform a timeout reset, which may lock up the user program.

If a status byte is returned, a "1" means no error, a "0" failed command.

In addition, the emulated ACIA uses an internal timeout of about 2 seconds.
Means if a command call is not processed, waits or is interrupted the ACIA will perform a timeout reset (writing a file for example should not be interrupted longer than 2 seconds)

0 Set/Clear special functions

0: do nothing
1: return version info return, return 1 data byte (00..),
2: reset all special functions, no return value
3: select RS232 internal port, no return (uses system RX/TX lines)
4: select RS232 external port, no return (uses onboard pin header RX/TX lines)
5: Set FIFO to 2 , no return (RTS/CTS handshake by user)
6: Set FIFO to 4 , no return (RTS/CTS handshake by user)
7: Set FIFO to 8 , no return (RTS/CTS handshake by user)
8: Set FIFO to 16 , no return (provides automatic RTS/CTS handshake)
9: read 1 byte from EEPROM, send startL, startH, return 1 status bytes (0 to 1023)
10: write 1 bytes to EEPROM, send startL, startH, data, return 1 status byte (control)
11: activate(1)/deactivate(0) setup on Boot from EEPROM parameter table, no return
12: write setup to EEPROM, return 1 status byte

13..15: undefined, no return

1 Execute Loader Code in binary with leading length and start address, no return value

0: PreLoader (0240) to load BOOT.SYS from SD, return lod file for OSI monitor loader
1..15: more code may be placed into ATMEGA in future releases

2 Select baud rate, no return value (SW=software defined, HW=by OSI typical TX clocks

Note: HW are the only baud rates supported by emulation and TX clock analysis on Startup.

TX clock cannot be used directly for the emulated UART due to hardware restriction to only TX clock divider by 1 (Synchron mode of UART)

0: restore from SW to HW serial baud rate from initial reading of TX clock
1: 150 only SW
2: 300 +HW
3: 600 +HW
4: 1200 +HW
5: 2400 +HW
6: 4800 +HW
7: 9600 +HW
8: 14400 only SW
9: 19200 +HW
10: 38400 +HW
11: 57600 only SW -> transmit works, rest not reliable
12: 76800 only SW -> nonstandard, not tested
13: 115200 only SW -> transmit works, rest not reliable
14: 125000 only SW -> not tested
15: 31250 only SW ->> MIDI IN/OUT, +1.58% faster compared to MIDI

3 SD card functions

Note: Only a single file can be opened for R/W. If SD operation holds for more than 2 sec, a soft reset will be initiated to restart the ACIA emulation. So, writing to a file in intervals separated by more than 2 sec will not work!

If operation was successful, a one is returned, elsewhere a zero
(no. 2 added in version V1.35)

0: start and check SD card, returns 1 if available, 0 if not (will cause ACIA reset !)
1: close and eject SD card, no return (card needs to be re-inserted !)
2: get name of entry number (0..255) as string of selected path, return name plus zero
3: set filename, name ends with byte zero (max 24 characters incl. zero), no return
4: check if file exist, returns 1 status byte

- 5: set path, ends with byte zero (max 32 characters), no return
- 6: return directory list, name plus zero. Ends with zero if end of directory.
The directory output may be interrupted by sending a break \$03 to the Controle port. Returns zero after a break command. Break must be done at the beginning of the next directory string output. (note: older ACIA firmware may not work correctly)
- 7: get free disk space in MB, returns 2 data bytes 16bit
- 8: get file size in bytes, returns 2 da bytes 16bit, returns 65535, if file larger than 65535 bytes (in BASIC -1) or 0 if file does not exist.
- 9: read file, read 2 bytes file size, returns zero word on error or no file present, followed by binary data. Terminate by write 03 to ACIA control port if needed, no return.
- 10: create folder in selected path (folder name in filename) returns 1 status byte
- 11: delete folder in selected path (folder name in filename), returns 1 status byte
- 12: delete file (defined before with path), returns 1 status byte
- 13: rename file or folder, send new filename, name ends with byte zero (max 24 characters), returns 1 status byte

Note: before rename, do step 3,5 and 4, to define file to be renamed !

- 14:write file, send 2 bytes file size (max 64k), return 1 status byte, if file was allocated.
Next send binary data, may be terminated by writing \$03 to ACIA control port, returns 1 status byte when done.
- 15:Get last SD Error, returns 1 data byte.

4: IRQ enable/disable timer, does not set IRQ flag in Status register

NOTE: activates even if IRQ enable is not set, IRQ signal resets after 10µsec by itself

Note: CTS can be used as an interrupt input and disables normal CTS behavior

- 0: disable timer IRQ
- 1: IRQ on CTS goes low. IRQ resets after 14µsec by itself
- 2: IRQ 128 µsec
- 3: IRQ 256 µsec
- 4: IRQ 512 µsec
- 5: IRQ 1.024 msec
- 6: IRQ 2.048 msec
- 7: IRQ 4.069msec
- 8: IRQ 10 msec
- 9: IRQ 16.67 msec (60 Hz)
- 10: IRQ 20 msec (50 Hz)
- 11: IRQ 32.78 msec (max)
- 12: read OSCCAL factory calibration value, returns 1 byte
- 13: read actual OSCCAL, returns 1 byte
- 14: write OSCCAL, changing will lead to non-function TX/RX baud rates !!
- 15: reset OSCCAL to calibration value

Note: to use IRQ on OSI's ACIA, you have to connect pin 7 (Socket of ACIA) with the CPU IRQ line Pin 4, in case you are using an OSI 600 board

OSCAL allows variation of system clock in a range of +/- 25%

5.15:Function not yet implemented

Program Example for checking SD card presence and free memory on card:

```
ACIA_S      = $F000      ; SERIAL ACIA Control Port
ACIA_D      = $F001      ; SERIAL ACIA Data Port
SDEXIST     = $xxxx     ; PLACE IN MEMORY

SAMPLE: JSR   CLEAR      ; Do SDCLEAR on startup

          LDA   #0
          LDA   SDEXIST
          BEQ   TEND:     ; no SD present

          LDA   #$71     ; Extended mode
          STA   ACIA_S
          LDA   #$37     ; Check free size on SD
          JSR   ACIA_WRITE ; Get file size

          JSR   ACIA_READ
          TAY
          JSR   ACIA_READ ; Low to Y
                                ; High to A
TEND:     RTS           ; Returns size in A,Y

CLEAR:
          LDA   ACIA_D   ; Clean up any pending byte
          LDA   #$03
          STA   ACIA_S

          LDA   #$71     ; enter Extended mode
          STA   ACIA_S

          LDA   #$30
          JSR   ACIA_WRITE ; function Check if SD present

          JSR   ACIA_READ ; read return value
          STA   SDEXIST
REND:    RTS

ACIA_READ:
          LDA   ACIA_S
          LSR
          BCC   ACIA_READ
          LDA   ACIA_D
          RTS

ACIA_WRITE:
          PHA
WR1:     LDA   ACIA_S
          AND   #$02
          BEQ   WR1
          PLA
          STA   ACIA_D
          RTS
```