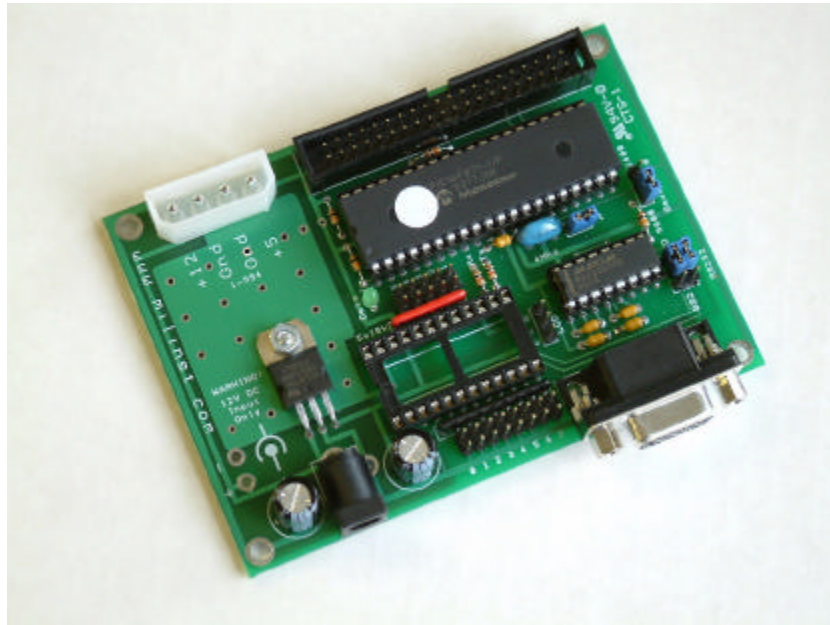


## **RS232 CD ROM DRIVE CONTROLLER, Part # 1-954**



### **INTRODUCTION**

This module allows the CD-audio related functions of any standard atapi/ide CD-ROM drive to be controlled through a standard RS232 serial interface using a simple command syntax. (eg. To play a specific track on a CD you send 2 bytes - a command code followed by the track number, pretty simple!). It is ideal for microcontroller based audio playback applications as the CD ROM drive can be controlled by only 2 I/O pins, but can be used by any device capable of sending and receiving RS232 serial data. In addition, a socket is provided to allow onboard control by an optional BASIC STAMP2 module.

Includes WIN95/98 software for testing and evaluation purposes using a pc serial port.  
Supports 2400 and 9600 baud rates.  
Limited to use with audio only CD's with a maximum of 24 tracks.

Acknowledgement- Module concept, firmware and VBASIC test programme by Ken Bromham

## CIRCUIT BOARD DETAILS

### POWER SUPPLY INPUT CONNECTOR :

The power supply input connector is 2.1 mm **CENTRE POSITIVE**.

Requires 12V DC Regulated supply, **minimum 1.2A** recommended in order to provide enough current for the CD ROM drive. Check polarity before connecting.

The onboard voltage regulator will get fairly warm – this is normal and does not indicate a fault.

### +5V / +12V CD ROM DRIVE POWER OUTPUT CONNECTOR :

Connects to the power supply input on the back of the CD ROM drive. The connectors are keyed so will only fit one way round.

### D9 SOCKET :

Provides the RS232 serial interface connection – configured as DCE, no handshaking (use straight through cable to connect to eg. PC serial port).

When used with onboard stamp module, provides the interface to the stamp module SOUT, SIN, ATN pins. Again, use straight through cable.

### IDE 40 PIN PLUG :

Connects to the ide interface connector on the CD ROM drive. These connectors are keyed but make sure the red end of the ribbon cable connects to pin 1 at both ends. (pin 1 on the circuit board is the end nearest the baud rate jumper).

### Parallax BASIC STAMP 2 SOCKET :

Allows a Parallax BS2-IC stamp device to be added to build a stand alone application. A bank of pull up resistors are placed on stamp pins 0-7 and a connection for a serial LCD or other device on stamp pin 10.

### BAUD RATE JUMPER :

Used to select either 2400 or 9600 baud. The required baud rate should be selected before powering up the circuit board. Use 2400 baud with stamp modules.

### RS232 JUMPERS :

Used to configure the board for either external control through the RS232 interface (CD), or for use with an onboard stamp module (BS2). Both jumpers must be set the same way.

### J1 JUMPER :

J1 should be connected for external RS232 control and removed for onboard stamp control.

### DATA LED :

Flashes to indicate when a command is being executed. The length of the flash gives an indication of the time taken to complete the command.

Also flashes twice following power up.

## CD-ROM Connectors:

### CD ROM DRIVE MASTER / SLAVE JUMPERS:

The CD ROM drive should be configured as **MASTER**.

### CD ROM AUDIO OUTPUT :

Use either the headphone jack on the front of the CD ROM drive or the 4-pin analog audio out on the back of the CD ROM drive.

## SYNTAX AND PROTOCOL

In general the protocol is to send a valid command message to the module, which will then execute the command and return one or more bytes (depending on the command), to indicate success/fail status or to provide the data requested by the command.

A command message consists of a command byte (the command code) followed by 0, 1, 2 or 6 parameter bytes (depending on command). Command bytes and parameter bytes are easily recognised :

COMMAND BYTE has msb set – ie. range 0x80 to 0xff (128 to 255)

PARAMETER BYTE has msb clear – ie. range 0x00 to 0x7f (0 to 127)

Eg. Play track command has command code 134 and requires 1 parameter (track num), so to play track 3 you would send the following 2 bytes : <134> <3>

Stop command has command code 136 and requires no parameters, so to stop playback you would send a single byte : <136>

Some commands simply specify an action and do not need to return any actual data (eg. play, stop), but it is necessary to know whether the command has executed successfully so for commands in this category a single status byte will be returned.

Other commands are used to get actual data values (eg. read toc, read subchannel), in which case if the command executes successfully then the actual data bytes are returned. If the command fails then a single status byte only is returned. This behaviour is summarised below:

### COMMANDS THAT DO NOT NEED TO RETURN DATA BYTES

- if command executes successfully then returns single status byte with value 0x00
- if error then returns single status byte with value 0x8n where n is the error-code (see appendix A).

### COMMANDS THAT DO NEED TO RETURN DATA BYTES

- if command executes successfully then returns actual data values .First byte returned always in range 0x00 to 0x7f (0 to 127). The number of bytes returned depends on the command.
- if error then returns single status byte with value 0x8n where n is the error-code (see appendix A).

So in all cases if the command fails to execute, a single status byte in the range 0x81 (129) to 0x8f (143) will be returned. It is important to check the status of every command sent and to understand the different error-codes.

Invalid command code bytes are ignored.

'Spare' parameter bytes are ignored.

After issuing a command (+ parameters), it is important to wait for the status byte / data bytes to be returned before proceeding.(This is necessary because no handshaking is used).

## COMMAND CODE SUMMARY

128 TEST READY  
129 READ TOC  
130 READ NUM TRACKS  
131 READ END LOCATION  
132 READ TRACK **<TRACK NUM>**  
133 PLAY **<STARTMM>** **<STARTSS>** **<STARTFF>** **<ENDMM>** **<ENDSS>** **<ENDFF>**  
134 PLAY TRACK **<TRACK NUM>**  
135 PLAY ALL  
136 STOP  
137 PAUSE  
138 RESUME  
139 READ SUBCHANNEL  
140 READ AUDIO STATUS  
141 READ AUDIO STATUS + ABS LOCATION  
142 READ AUDIO STATUS + REL LOCATION  
143 SET VOLUME **<VOL L>** **<VOL R>**  
144 SET TIMEOUT **<TIMEOUT>**  
145 READ MEDIA TYPE  
146 LOAD  
147 EJECT  
148 SPIN UP  
149 LOCK  
150 UNLOCK  
151 HARDWARE RESET

152 SLOW READ TOC  
153 SLOW READ END LOCATION  
154 SLOW READ TRACK **<TRACK NUM>**  
155 SLOW READ SUBCHANNEL  
156 SLOW AUDIO STATUS + ABS LOCATION  
157 SLOW AUDIO STATUS + REL LOCATION

## COMMAND CODES IN DETAIL

NOTE : All locations are expressed in minutes, seconds, frames  
MM:SS:FF (1 sec = 75 frames)  
Track 1 always starts at location 00:02:00

**TEST READY 128** - no parameters, returns status byte only

Used to see if drive is in ready state (ie. tray closed, disc loaded and ready for access) without specifying any other action.

**READ TOC 129** - no parameters, returns  $(3n + 4)$  data bytes where  $n$  = number of tracks

Used to return the table of contents data. The first byte returned is the number of tracks on the CD, followed by 3 bytes giving the start location of each track up to and including the start location of the disc lead out area (ie. end of audio).

Eg. If the disc has 2 tracks, then 10 bytes will be returned as shown:

Byte1 = 2 (number of tracks)  
Byte2 = trk1 startmm  
Byte3 = trk1 startss  
Byte4 = trk1 startff  
Byte5 = trk2 startmm  
Byte6 = trk2 startss  
Byte7 = trk2 startff  
Byte8 = leadout startmm  
Byte9 = leadout startss  
Byte10 = leadout startff

The start location of track2 can be used as the end location of track1 etc.  
The start location of the lead out area can be used as the end location of the whole disc.

**READ NUM TRACKS 130** - no parameters, returns 1 data byte

Used to return the number of tracks on the disc. This command is a cut down version of the read toc command.

**READ END LOCATION 131** - no parameters, returns 3 data bytes

Used to return the start location of the disc lead out area (ie. end of audio). This command is a cut down version of the read toc command.

**READ TRACK 132** – 1 parameter **<track num>**, returns 6 data bytes

Used to return the start and end location of a given track. Returns startmm, startss, startff, endmm, endss, endff. This command is a cut down version of the read toc command.

**PLAY 133** - 6 parameters **<startmm> <startss> <startff> <endmm> <endss> <endff>**, returns status byte only

Used as a general play command. The start and end locations must be specified by the 6 parameters in the order shown. Playback stops automatically when the end location is reached. Other commands can be processed while playback is in operation – some commands will have no effect on playback, other commands will, as shown :

COMMANDS THAT INTERRUPT CURRENT PLAYBACK:

play, play track, play all, stop, pause, eject, spin up, hard reset

**PLAY TRACK 134** - 1 parameter <track num>, returns status byte only

Used to play a specific track. Avoids having to read the table of contents.

**PLAY ALL 135** - no parameters, returns status byte only

Used to play the whole CD. Avoids having to read the table of contents

**STOP 136** - no parameters, returns status byte only

Used to stop playback.

**PAUSE 137** - no parameters, returns status byte only

Used to pause playback.

**RESUME 138** - no parameters, returns status byte only

Used to resume playback following pause.

**READ SUBCHANNEL 139** - no parameters, returns 10 data bytes

Used to return the subchannel current positional information in the following order:

byte1 audio status code  
byte2 adr/control fields  
byte3 track number  
byte4 index number  
bytes5-7 current absolute location (mm:ss:ff)  
bytes8-10 current location relative to start of track (mm:ss:ff)

**audio status** code values (decimal):

**17** = play operation in progress

**18** = play operation paused

**19** = play operation successfully completed

**20** = play operation stopped due to error

**21** = no current audio status to return

adr/control fields :

The adr field is the high nibble of this byte. It will have the value 1 in all cases.

The control field is the low nibble of this byte. The value is determined by the following table :

bit0 = 0 audio without pre-emphasis bit0 = 1 audio with pre-emphasis

bit1 = 0 digital copy prohibited bit1 = 1 digital copy permitted

bit2 = 0 audio track bit2 = 1 data track

bit3 = 0 two channel audio bit3 = 1 four channel audio

**READ AUDIO STATUS 140** - no parameters, returns 1 data byte

Used to return the current audio status. This command is a cut down version of the read subchannel

command.

**READ AUDIO STATUS + ABS LOCATION 141** – no parameters, returns 4 data bytes

Used to return the current audio status plus the current absolute location. This command is a cut down version of the read subchannel command.

**READ AUDIO STATUS + REL LOCATION 142** – no parameters, returns 4 data bytes

Used to return the current audio status plus the current relative location. This command is a cut down version of the read subchannel command.

**SET VOLUME 143** - 2 parameters <volL> <volR> , returns status byte only

Used to set the left and right volume (0 = mute , 127 = max).

**SET TIMEOUT 144** - 1 parameter <timeout>, returns status byte only

Used to set the inactivity timeout period. The parameter <timeout> should be in the range 1 to 15 which will set the timeout period as shown in the following table :

- 1 = 125 ms
- 2 = 250 ms
- 3 = 500 ms
- 4 = 1 s
- 5 = 2 s
- 6 = 4 s
- 7 = 8 s
- 8 = 16 s
- 9 = 32 s
- 10 = 1 min
- 11 = 2 min
- 12 = 4 min
- 13 = 8 min
- 14 = 16 min
- 15 = 32 min

note : not all drives allow this parameter to be changed, in which case the timeout period will be fixed at its default value, usually somewhere in the range of 30 secs to 4 mins.

Explanation :

If the CD ROM drive is hanging around 'doing nothing' (ie. not playing back), then after a period of time equal to the inactivity timeout period has elapsed the disc will spin down to conserve power and minimise wear and tear. This is a handy feature but has some implications relevant to audio playback applications. In particular, if the disc is still spinning when a play command is received then the playback will start almost immediately, the only delay being the time taken to move the optics to the correct start position which takes a fraction of a second. If the disc has spun down, however, it must additionally spin up again first before the playback can start which gives rise to a longer delay, typically 1 - 2 secs.(NB the drive will automatically spin up in response to any play command if it needs to). For applications requiring immediate playback it may therefore be useful to set the inactivity timeout to a high value. The life expectancy (MTBF) values of typical CD ROM drives are pretty respectable, so keeping the disc spinning for perhaps long periods of time while not actually playing back is unlikely to be a problem.

**READ MEDIA TYPE 145** - no parameters , returns 1 data byte

Used to return a code value indicating the type of disc loaded, no disc loaded or tray open. See appendix B for a full list, but the values of most interest are :

2 = 120mm CD- DA audio only  
112 = door closed, no disc present  
113 = door open

**LOAD 146** - no parameters , returns status byte only

Used to close the tray and load a disc if present.

**EJECT 147** - no parameters, returns status byte only

Used to open the tray.

**SPIN UP 148** – no parameters, returns status byte only  
(see also set timeout)

Used to cause the drive to spin up and position the optics at the start of the disc, without specifying any further action.(NB If the disc has spun down it is not necessary to send a spin up command before a play command, the drive will automatically spin the disc up whenever it needs to.) This command can be useful, however, if sent periodically to ensure the drive is kept constantly spinning – particularly relevant for applications requiring instantaneous playback if the CD ROM drive does not support a changeable inactivity timeout period.

**LOCK 149** – no parameters, returns status byte only

Used to prevent the tray from opened (either manually or by the eject command).

**UNLOCK 150** – no parameters, returns status byte only

Used to allow the tray to be opened, following a lock command. Power on and hardware resets will also unlock the drive.

**HARDWARE RESET 151** - no parameters, returns 0x00 always

Used to execute a hardware reset on CD ROM drive. This is similar to a power on reset and will return the CD ROM drive registers to their default values.

The following are 'stamp friendly' duplicate commands, where a 30ms delay will be placed between transmission of each returned byte. This allows the stamp plenty of time to do a debug command or write to eeprom before the next byte arrives, useful if you don't have enough ram to store the received bytes. (Read toc command is the main problem). Note that the standard commands are fine for stamp use (at 2400 baud) as long as the returned bytes are stored directly into ram with minimal processing between each byte. The 30 ms delay only applies when in 2400 baud mode, if set to 9600 then these commands function exactly the same as the standard commands.

**SLOW READ TOC 152** – no parameters, returns (3n + 4) data bytes where n = number of tracks

**SLOW READ END LOCATION 153** – no parameters, returns 3 data bytes

**SLOW READ TRACK 154** – 1 parameter **<track num>**, returns 6 data bytes

**SLOW READ SUBCHANNEL 155** – no parameters, returns 10 data bytes

**SLOW AUDIO STATUS + ABS LOCATION 156** – no parameters, returns 4 data bytes

**SLOW AUDIO STATUS + REL LOCATION 157** – no parameters, returns 4 data bytes

## **USING THE MODULE WITH THE PC TEST SOFTWARE**

### **INSTALLING SOFTWARE:**

The executable file CDROMtst.exe can simply be copied to any chosen directory. The associated system file mscomm32.ocx should be copied to your DRIVE:\WINDOWS\SYSTEM directory. This file may already exist on your system in which case there is no need to overwrite the existing file.

Set up the module for use under external RS232 control (see below) and connect to a free serial port on the pc (straight through cable).

Baud rate can be either 2400 or 9600.

Power up the module/CD ROM drive first, then run the application CDROMtst.exe.

Make sure the baud rate settings and com port selection are correct.

Using the application should quickly become self-explanatory.

## **USING MODULE UNDER EXTERNAL RS232 CONTROL**

The module can be controlled by any device capable of sending and receiving RS232 serial data at either 2400 or 9600 baud (8 bit, no parity, 1 stop bit, no handshaking). The module should be set up as follows:

Set RS232 jumpers to CD position.

Set baud rate jumper to match controlling device.

Connect J1 jumper.

Connect power / ide cables to CD ROM drive.

Connect the module D9 socket to the RS232 port of the controlling device.

Module D9-pin2 is transmit.

Module D9-pin3 is receive.

Module D9-pin5 is ground.

Make sure CD ROM drive is configured as master.

## **USING MODULE WITH ONBOARD BASIC STAMP2**

Set RS232 jumpers to BS2 position.

Set baud rate jumper to 2400.

Remove jumper J1.

Connect power / ide cables to CD ROM drive.

Connect module D9 socket to pc (straight through cable).

Make sure the CD ROM drive is configured as master.

Run the stamp development software.

Stamp2 can now be programmed as normal.

PIN8 IS USED FOR SEROUT instructions to send commands to module (TX).

PIN9 IS USED FOR SERIN instructions to receive bytes from module (RX).

Use baudmode value 396.

All other stamp pins are available as normal.

A bank of pull up resistors are provided on PIN0-PIN7 for use with external push buttons etc.

An output for serial LCD or other device is provided on pin 10.

After power up you should wait a minimum of 100ms before sending any commands to the module. In practice, you may as well wait a few seconds as it takes this amount of time for the CD ROM drive to initialise. Please see the supplied diskette for sample Stamp2 programmes.

## APPENDIX A: ERROR-CODES

When there is an error executing a command, the module returns a status byte with value 0x8n where n is the error code. The value of n corresponds,(if you're interested), to the atapi sense key value, as indicated in the following table:

n = 1 (returns 0x81 129) RECOVERED ERROR  
**n = 2 (returns 0x82 130) NOT READY**  
n = 3 (returns 0x83 131) MEDIUM ERROR  
n = 4 (returns 0x84 132) HARDWARE ERROR  
**n = 5 (returns 0x85 133) ILLEGAL REQUEST**  
**n = 6 (returns 0x86 134) UNIT ATTENTION**  
n = 7 (returns 0x87 135) DATA PROTECT  
n = 8 (returns 0x88 136) reserved  
n = 9 (returns 0x89 137) reserved  
n = A (returns 0x8A 138) reserved  
n = B (returns 0x8B 139) ABORTED COMMAND  
n = C (returns 0x8C 140) reserved  
n = D (returns 0x8D 141) reserved  
n = E (returns 0x8E 142) MISCOMPARE  
n = F (returns 0x8F 143) reserved

The values highlighted are of most interest and need to be understood in order to use the module successfully.

**130 NOT READY** – This will be returned in response to any command that needs a disc to be loaded, unless the tray is closed with a disc present and ready for access.

**133 ILLEGAL REQUEST** – This usually indicates an invalid parameter byte (eg. play command with end location < start location, or play track command with an invalid track number). Can also indicate a valid command being used in the wrong place(eg. some drives will return 133 if you try to pause when not currently playing).

**134 UNIT ATTENTION** – This is a warning that the disc may have been changed and arises after a power on or hardware reset, or after a tray open / close cycle. The unit attention error is returned once only, in response to the first command received that would otherwise execute successfully. This can cause some confusion at first – the following sequence should explain more clearly:

power up drive (no disc loaded)  
send play all command  
get 130 not ready error  
send play all again  
get 130 not ready error  
open tray , insert disc, close tray  
send play all again  
get 130 not ready error a few more times(keep resending play all)  
send play all again  
get 134 unit attention once only (because tray has been opened/closed)  
send play all one more time  
now returns 0 and play starts

After this point the unit attention error 134 will not arise again unless the tray is opened / closed or a power on or hardware reset occurs. Note that the unit attention error is returned at the point when the play all command would otherwise execute successfully (ie. get the not ready errors first).  
If the above sequence was repeated, but with a disc already loaded at power up (so you didn't need to open and close the tray), you would still get the 134 error because of the power on reset.

## **APPENDIX B: MEDIA TYPE CODES (decimal)**

- 0 medium type unknown
- 1 120mm CD-ROM data only
- 2 120mm CD-DA audio only
- 3 120mm CD-ROM data and audio combined
- 4 120mm CD-ROM hybrid disc(photo CD)
- 5 80mm CD-ROM data only
- 6 80mm CD-DA audio only
- 7 80mm CD-ROM data and audio combined
- 8 80mm CD-ROM hybrid disc(photo CD)
- 9 – 15 reserved
- 16 type CD-R, size unknown
- 17 120mm CD-ROM (CD-R) data only
- 18 120mm CD-DA (CD-R) audio only
- 19 120mm CD-ROM (CD-R) data and audio combined
- 20 120mm CD-ROM (CD-R) hybrid disc(photo CD)
- 21 80mm CD-ROM (CD-R) data only
- 22 80mm CD-DA (CD-R) audio only
- 23 80mm CD-ROM (CD-R) data and audio combined
- 24 80mm CD-ROM (CD-R) hybrid disc(photo CD)
- 25 – 31 reserved
- 32 type CD-E, size unknown
- 33 120mm CD-ROM (CD-E) data only
- 34 120mm CD-DA (CD-E) audio only
- 35 120mm CD-ROM (CD-E) data and audio combined
- 36 120mm CD-ROM (CD-E) hybrid disc(photo CD)
- 37 80mm CD-ROM (CD-E) data only
- 38 80mm CD-DA (CD-E) audio only
- 39 80mm CD-ROM (CD-E) data and audio combined
- 40 80mm CD-ROM (CD-E) hybrid disc(photo CD)
- 41 – 47 reserved
- 48 medium type unknown
- 49 120mm (HD)
- 50 – 52 reserved
- 53 80mm(HD)
- 54 – 111 reserved
- 112 door closed, no disc present
- 113 door open
- 114 medium format error
- 115 – 127 reserved