

CyberSense

CyQ 504

OPERATORS MANUAL

Version 12.1

SUPPORT

support@cyq.com

CyberSense Inc.
1330 River Road Nicholasville, KY 40356-9649
www.cyq.com
800-942 9327
Copyrights © 1996 -2007. All rights reserved

TABLE OF CONTENTS

INSTALLATION	-1-
Hardware	
Cymograph Software	
HYPER TERMINAL	-6-
How to Set up Windows Hyperterminal	
Basic Data Acquisition System	
How to Install Hyperterminal	
COMMANDS	-9-
Command Summary	
Acquire Commands	
Configure Commands	
Data Output Format Commands	
Utility Commands	
Why Both Rate and Interval Modes?	
Median Filter	
Burst Mode Filter	
THE GOVERNOR	-23-
How fast can I go?	
ERROR HANDLING	-24-
SERIAL HARDWARE INTERFACE	-25-
INPUTS	-26-
SPECIFICATIONS	-27-
LEGAL STUFF	-28-

INSTALLATION

Quick test

Remove the CyQ hardware product - chances are it is on a DIN rail and already connected. Plug the wall transformer into a 115 vac outlet.

Plug the barrel connector into the back of the 504.

While viewing the front panel LED's turn on the power switch (red handle, up is on). The blue power light should come on followed by a one-half second flash from each of the four green LED's in order from channels 1 to 8. The orange Tx LED should flash briefly.

Firmware Test and Setup

If you intend to write code for the CyQ504, then you may skip to the Hyperterminal section of this manual.

Software Setup

If you want to acquire data, then Cymograph will set up the CyQ504 to provide a real time graphical display of your data.

Cymograph® was developed as a tool for testing our signal conditioning products. We started using it for demonstration purposes. Then we started shipping it at no charge as a stop gap to fill in while someone was making a purchase decision on data acquisition software. People continued to use it, and we began to add to it as requests came in.

Ipsa facto, Cymograph had become their data acquisition software; I believe that also means that it is now our data acquisition software.

There will come a time when you believe that everything is finished.

That will be the beginning. Louis L' Amour - Lonely on the Mountain

Our basic design philosophy is best summarized by Albert's words:

Everything should be made as simple as possible, but not simpler. A. Einstein

How to install Cymograph

The Cymograph CD is in a pocket inside the CyQ manual notebook. Connect the serial cable to a vacant serial port on your computer.



Place the Cymograph CD into a drive.

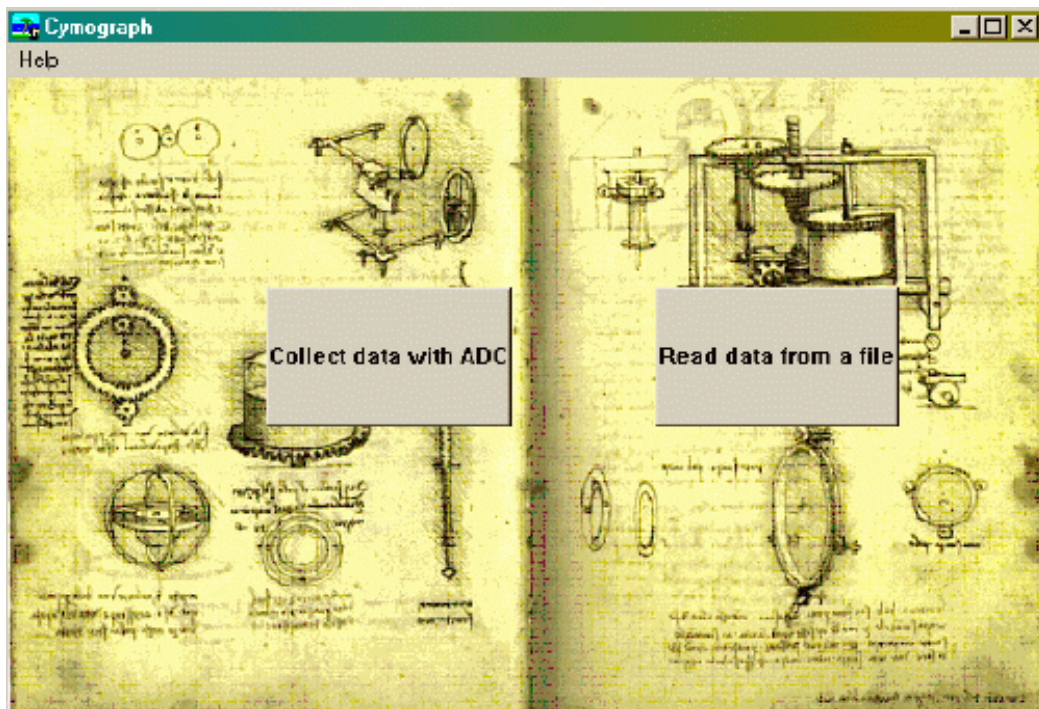
Use Windows Explorer to find the InstallCymograph.exe file.

Click to install Cymograph.

Turn the CyQ504 power on - flip the red handled switch up.

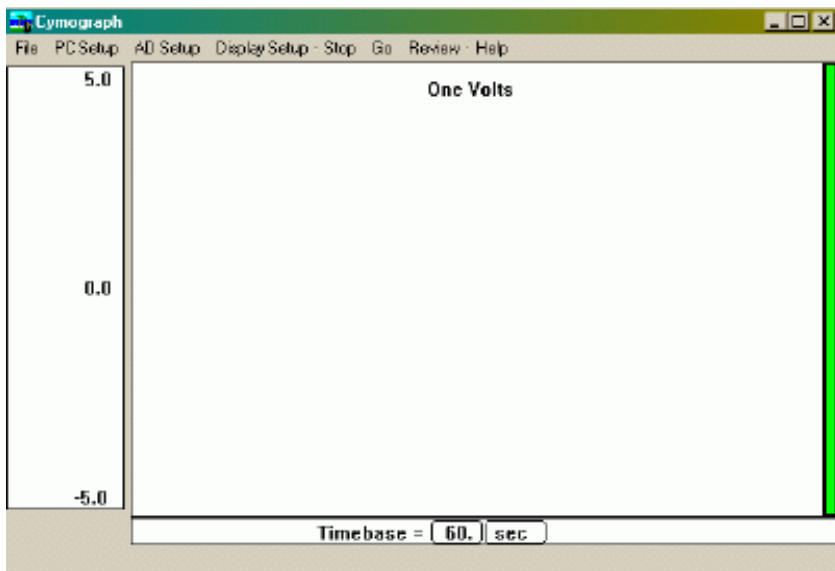
Start Cymograph by clicking the Cymograph Icon on the desktop.



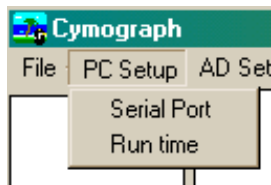


Click *Collect Data with ADC*.

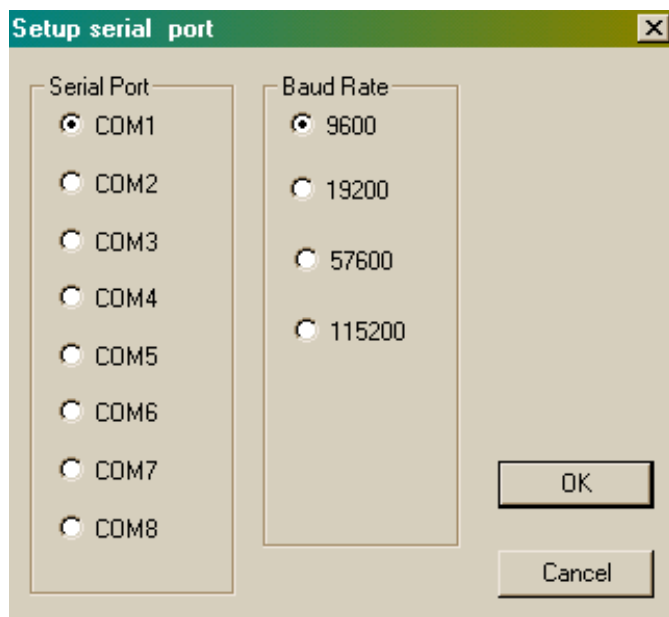
The default window appears:



On the menu bar select *PC setup*



Select *Serial Port* and click OK:



In computerese the nickname for a serial port is COM Port (for communication port). There are usually two ports. They are numbered 1 and 2. (Note: If you are using a USB to serial adapter then COM5 or above is likely to be the one.) One number corresponds to the vacant port that you plugged the cable into.

So, what to click on? Here is the method that experienced professional computer programmer's use: Who knows - I might get lucky, so try COM1; if that don't work then try COM2..

When you succeed you will see



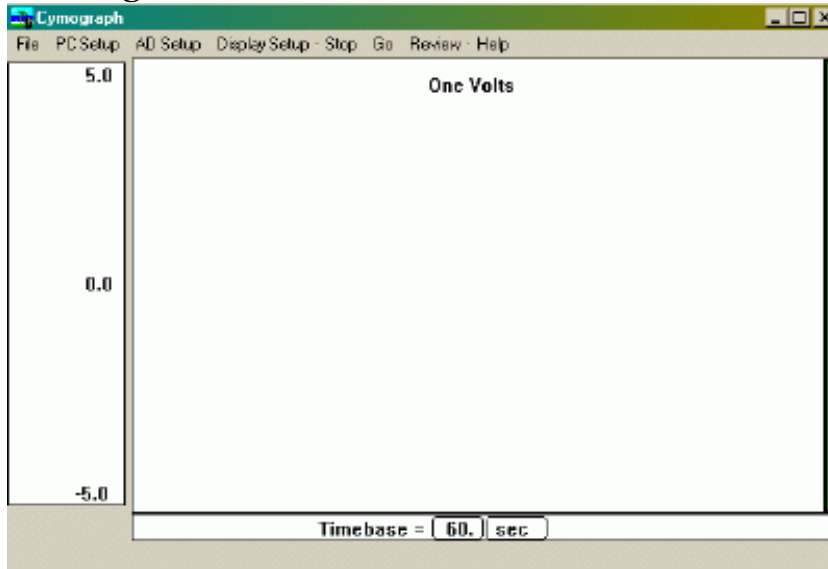
If bad guess, then you will see either



In the first case, try another port.

In the second case, either fix and retry or abort and try another port.

Clicking OK on the Success button will cause the CyQ504 to be initialized:



This is the CyQ504 screen. Click Go, and watch the data begin to appear.

NOTE: If you are using Cymograph you need to go no further.

HYPER TERMINAL

(Included with MS Windows. Updates are on Cnet along with many similar programs such as Tera Term)

From windows

- 1). Click on programs
- 2). Click on Accessories
- 3). Select hyper terminal if available. If it does not appear on the list of accessory programs, then go to: How to install Hyper terminal.

HOW TO SET UP WINDOWS HYPER TERMINAL FOR A COM PORT

- 1) In hyperterminal create a "**new connection**" this can be done by clicking the **Hypertrm.exe** icon or via the menu within HyperTerminal "**File:NewConnection**".
- 2) Type in a name for the connection (for example, cyq504).
- 3) Under "**connect using**" select "**Direct to Com 1**" if your available com port is com 1 otherwise select the appropriate com port. On older computers modems are relentlessly selfish about interrupts. So if a modem is present then avoid the modem port and it's cousin. That is, if the modem is on com 1 then avoid both com 1 and 3. Com 1 and 3 share an interrupt while com 2 and 4 share a different interrupt; in this case avoid 2 and 4.
- 4) Set "**Bits per second**" to 9600.
- 5) Set "**Data Bits**" to 8.
- 6) Set "**Stop Bits**" to 1.
- 7) Set "**Flow control**" to None.
- 8) Click OK - the hyper terminal screen should appear.
- 9). Select **properties** from the menu bar.
- 8) Click on "**settings**", use default: **autodetect** and **back scroll 500 lines**.
- 9) For "**ASCII setup**" use
 - a). Sending - check: "**echo typed characters locally**" box. This lets you see the commands you type in. You may also issue the "ck;" command. This will cause the 504 to echo received characters (keystrokes).

b). Receiving - check: **“wrap long lines”**.

10) Under **"File"** menu select **"Save As"** and save the configuration file to a convenient folder.

You can create a shortcut to the configuration file and put it on the desktop, this makes life a bit simpler. The configuration files have a naming convention of *.ht where star is the name you chose when setting up the new connection. For example, “cyq504.ht”.

Connect a serial cable from the DB-9 on the back of the 504 to the serial COM port that you intend to use.

Apply power: the word “CyQ504” will be sent by the 504. That is why the yellow Tx LED blinks on power up.

With Hyperterminal running:

1. Enter “cofv” for configure format volts output.
2. Enter “r” for rate mode (default is channel 1 rate 1 per sec)
3. The default range mode is 5 volts bipolar so you should see:

```
4.995
4.996
.
.
.
4.995
```

The 4.994 volts is due the range divider resistors with the input open circuited. Connecting Pin 1 to Pin 2 (leftmost two pins) on the input plug will cause the readings to go to near zero: 0.002, 0.003,, 0.002.

Typing a “?” and pressing the Enter key will cause a summary of the A/D settings to be printed.

You now have a basic DATA ACQUISITION SYSTEM

The transfer button on the hyper terminal menu bar will allow you to save received data to a file. You can put the CyQ5XX A/D in continuous acquire mode and the data will go to a file name that you select (default is programs/accessories/capture.txt). You can use scalefactor settings to convert the output to your physical units. Our defaults are for the CyQ103, 100 mmHg per volt. You can then import this file into a spreadsheet or whatever. A file

extension of *.csv will cause the file to open in **Microsoft Excel**. You can also direct the output to a line printer for hard copy.

Notes:

- 1. To change the baud rate DURING a session you must (after going through properties to change the rate) click the 'call' button then 'disconnect' and then 'connect'.*
- 2. The ANSI terminal option will display all ASCII characters including symbols for the unprintable ones.*

HOW TO INSTALL HYPER TERMINAL

- 1). Click on **Control panel**
- 2). Click on **My Computer**
- 3). Click on **Install/Remove Programs**
- 4). Click on **Windows settings**
- 5). Click on **Communications**
- 6). Select **Hyper terminal** and click **apply** or OK at the bottom. You will be prompted for your Windows CDROM.
- 7). Done and out'ta there.

CyQ 504 COMMANDS

There are three kinds of commands. *Configure* commands are used to change parameters such as interval between conversions or number of points to average. *Acquire* commands cause or enable conversions to occur. *Utility* commands include memory, baud rate, and other housekeeping chores .

Typing ?; or ?<ENTER> will print out the current status.

All commands **MUST** end with either a semi-colon ';' or the ENTER key. The command is buffered, but not processed by the A/D until a ';' or '\r' is received. The backspace key may be used for corrections on screen.

Where possible the command format allows a value to be assigned with an '=' following the command code or for the feature to be turned on or off with a logical t, f character instead of the '=' sign. This allows setup values to be defined and preserved for future use. It allows the feature to be reactivated with a minimum of hassle. For example: "cfm=3;" will set median points to 3, but it will not affect the usage of the median filter. If the median filter was initially off, then it may be turned on at a later time by "cfmt;" (or the abbreviated "cfm;") To stop using this filter issue the command "cfmf;" Again, note that "cfm=5" neither turns the filter on or off.

The ?; command will give a summary of commands and their current settings. Commands are generally used to write custom code for the CyQ504. Cymograph® handles this for you; should you need to then these commands can be sent using the "Send" dialog in Ad Setup on the menu bar. Clicking on Status on the Ad setup menu will send a command summary to the message window.

In the following N is an integer number; L is a logical true or false, and X is a hexadecimal number. A logical L may be T, t, ';' for true, or F, f for false.

SUMMARY OF COMMANDS

aNNNN; acquire where N is a channel number.
s; stop.
g; go

Configure

Mode

cmr: acquire mode rate
cmt; timed
cms; slow (seconds)
cmp; polled
cmr=N; Sampling rate in rate mode: 1 to 1000
cmt=N; Time in ms between samples in timed mode: 1 to 60,000.
cms=N; Seconds between samples: 1 to 43200 (12 hours)

Span

csb; Span bipolar +/- 5 volts
csu; " unipolar (0 to 10 volts)

Output

cofi; Integer output format -2048 to 2048 or 0 to 4096
cofv; Voltage format
cofs; Scaled format, multiply channel N output voltage by cosN=n;
cofx; Hexadecimal format
cofb; Binary format
cop = N; Set output prefix, 0 is none.
con = N; Set postfix: 0 = none, 1 is cr, 2 is lf, 3 is crlf.
coiL; Send index counter
cocL; Send channel numbers
cooN = n; Channel N offset trim where second n = -125 to +125.
cosN = n; Channel N scale factor in scaled format mode is set to n (<32000);

Filters

cfbL; Burst mode averaging on-off
cfb=N; Cutoff frequency (default = 60)
cfr=N; Number of samples in burst

cfsL; Sample averaging on-off
cfs=N; Number of samples to average

cfmL; Median filter on-off
cfm=N; Median size (1 to 11)

Utility:

e; error light: off, et; resets on next valid command(default), ef; you turn off.

ck; Keystroke (Rx char) echo.
cq; Returns model number.
cq@N; Changes baud rate. N=3 is 9600, N=8 is 57600 ... see text
cg!L; Turns governor on (t, 1, ;) or off (f,0).
?; Transmit command summary and current settings.
mss; Memory save setup.
mls; Memory load saved setup.
msd; Memory save current setup as default.
mpd; Memory Purge default.
\$@R Restart. (case sensitive)

ACQUIRE COMMANDS

- 1. Acquire: "aNNN;"** where NNN... are 1 or more channel numbers. This assigns the channel numbers to be converted in either rate, timed or poll mode. If no channel numbers are present, "a;", then the previously defined channel numbers will be used. In polled mode this will cause a poll to occur. In rate and timed modes this command is used to define the channels. Modes are assigned in configure using "cmr;" or "cmt;" or "cmp;" commands for rate, timed or poll.
- 2. Stop: "s;"** Stops conversions from occurring. If in rate or timed modes and poll cannot occur, conversions are stopped. **CAUTION: Stop MUST be exited with a GO ("g;")** command for the poll (or rate or timed) acquires to work. You will forget this, and frantically pound the keyboard, wondering why nothing is happening.
- 3. Go: "g;"** Allows poll and continuous commands to resume. If stop occurred while a continuous conversion was in effect then the continuous mode is resumed. The **DEFAULT** mode on power up is stop.

CONFIGURE COMMANDS:

Current configuration: “?” causes modes, rate, and the number of points to be averaged to be printed followed by the interval between conversions for averaging. This is helpful during the development phase to make sure that the command you thought you entered was the command you entered. It also serves as a Help file for the command mnemonics.

Basic acquire mode commands

“cmr;” Configure acquire mode as rate. Sampling rate conversion. Channels following the acquire command (“aNN...;” will be converted repeatedly at the default rate(1 per sec) or previously set value for rate (see setup command “cmr=N;”).

“cmt;” configure acquire more as timed. Channels following the acquire command (“aNN...;” will be converted repeatedly at the default interval (1 sec) or previously set value (see setup command “cmt=N;”).

“cmp;” configure acquire mode as polled. An aNN..; command will cause channels NN... to be converted, and the results transmitted.

Rate in rate mode: “cmr=N;” where N ranges from 1 to 4000 samples per second. DEFAULT=10 samples per second. This command sets overall conversions per second.

Time between conversions in timed mode: “cmt=N;” where N ranges from 1 to 60,000. DEFAULT=1000 ms. This may be an averaged conversion.

Filters

Median

“cfmL;” enable (“cfb;” or “cfbt” or “cfb1;”) or disable (cfbf;” or “cfb0;”) median filter.
“cfm=N;” Set median filter size to N where N is 2 to 12.

Burst mode averaging.

Enable, disable burst mode averaging “cfbL;” where L is logical.
“cfb;” enables burst mode for N samples set by “cfb=N;”.
“cfbf;” disables burst mode averaging.

Number of conversions to average in burst mode averaging: “cfb=N;” where N

ranges from 1 to 255. Conversion results are summed until this number is reached then divided, and the result submitted to the sample averaging filter. DEFAULT= 10.

Burst mode sampling rate: “cfr=N;”. The rate in samples per second that the active AD channels are sequentially sampled and summed. A sampling rate of 600 with a count of 10 will attenuate 60 hz power line noise by >50 db.

Sample averaging

“cfsL;” Enable or disable sample averaging.

“cfs=N;” Set number of samples to average to N. N samples will be acquired and summed at the rate set for rate mode or at intervals set by timed mode. The sum will then be divided by N and the average transmitted on the N'th acquire. For example, if cfs=10 and cmr=5 conversions per second, then an averaged sample will be transmitted every 2 seconds.

SPAN

Bipolar span: “csb;” The input voltage range is -5 to + 5 volts. The corresponding output range is -2048 to +2048 in integer format mode. In binary mode the output data is offset binary with zero volts corresponding to integer 2048. Toggles unipolar off.

Unipolar range: “csu;” The input voltage range is 0 to 10 volts. The corresponding output range is 0 to 4096 in ASCII integer format mode. This toggles bipolar off.

DATA OUTPUT FORMAT

All output begins with the special character all bits set, hexadecimal 0xFF, binary 1111 1111 (all 8 bits set to 1).

‘v’ volts output: “<0xFF>#.###.....” Channel values are printed in volts with 0.001 volt (1 mv) resolution. Values range from -4.999 to 4.999 in bipolar, and 0.000 to 9.999 in unipolar mode (precision +/- 2.5 mv. typical accuracy +/- 10 mv . Channels are comma separated followed by a CR LF at the end, where CR is carriage return and LF is linefeed. In other words: start separator followed by comma delimited variable length numbers (due to the sign), and ending with a carriage return. Makes nuts and bolts sense, especially if you are using the AD to actually measure a voltage instead of, say, a pressure related to voltage only as an intermediate scale factor.

‘i’ causes the output format to be integer characters: <0xFF>number, number,....CR LF . In other words: start separator followed by comma delimited variable length numbers, and ending with a carriage return line feed. This is the DEFAULT. The range is -2048 to 2048 in bipolar and 0 to 4096 in unipolar. (varies slightly with A/D

offset error) Easy to see what is going on with the A/D.

'x' causes the output format to be in hexadecimal notation: <Prefix>number, number,... <PostfixCR LF>. In other words: start separator followed by comma delimited variable length numbers, and ending with a carriage return. Included for the comfort of programmers.

'b' binary output: "<0xFF>####...." each channel value is represented by two ascii characters; many unprintable. The digits are concatenated without delimiters. There is no CR at the end. This is a fixed length format with the length determined by the number of active channels. This allows much higher conversion speeds. Each character transmitted requires 10 bits. In integer mode one channel requires at least 5 digits plus a start character and final CR,LF; that's 80 bits. In binary mode only 2 digits plus a start character are used or 30 bits. Almost three times as fast. However, it is completely unreadable nonsense on a screen. For a computer a fixed length format with a start byte is quite easy to decode to an integer using either basic or c languages (wait for the start byte, shift left 8 next byte, add next byte, repeat for each channel).

"cocL;" causes channel numbers to be included in the transmission
ch:value,ch;value....

"coil;" will cause the first number to be an index number. This number will increment by 1 for each transmission. 000 001 002 003,....255 and roll over to 000. This feature is helpful in reassuring you that there is no missing data due to the communication link. Turning it off reduces the bandwidth requirement which is in the direction of goodness, that is, if it works correctly with the index then it is even more likely to work without it.

"cooN = N;" Sets the offset trim for channel N (1 to 4) to value N (-99 to 99). With input zero (either at the 504 or a preceding signal conditioner) there will be a slight offset error causing the AD to send a value that is either high or low. Entering the value of this error will cause it to be subtracted from the transmitted data sent by the 504. For example with channel one shorted the value transmitted is 2. Set coo1=2; and the value transmitted will become zero. This also serves as a single point calibration. For example, with 2.000 applied to channel 1 the reading is 1.95 volts. In volts mode 1 bit is about 2 mv, therefore, the offset error is about -2 bits; entering coo1 = -2; will correct channel 1 to read 2.000 volts.

"cosN=N;" Sets the scale factor for channel N (1 to 4) to value N (-32000 to 32000). The voltage output is multiplied by this factor. For example, the CyQ103 blood

pressure signal conditioner has an output of 10 mv per mmHg or 100 mmHg per volt. Therefore multiplying the voltage by a scale factor of 100 will cause the AD output to be in mmHg.

“cop=N” Sets the output prefix (the sync character that precedes data packet transmission) to ascii value N. For voltage, scaled, and integer the default is 0, no prefix. The prefix in binary format is 0xFF (255, all bits set) and cannot be changed. Examples: 07 will change the prefix to ASCII BEL, the computer will issue a beep; 36 will cause a '\$' preceding data, 42 will result in an '*' asterisk preceding data; 61 will result in an '=' equals sign prefix. Put simply, N = decimal position in the ASCII text table.

“con=N” Sets the output postfix (the carriage return, cr, and line feed, lf, characters that follow transmission). N=1 is cr (carriage return), N=2 is lf (line feed) N=3 is both cr and lf, N=0 is neither cr or lf. For voltage, scaled, and integer the default is cr lf.

Utility commands

1. Baud rate: "cq@X;" where X ranges from 0 to A. Sets the transmit and receive baud rates for the A/D serial output. The DEFAULT rate at power up is 9600. There is always 1 start bit, 8 data bits, and 1 stop bit; there is no parity. The computer baud rate **MUST** be changed to match after this command is executed (in hyper terminal you must click properties, and then configure. Baud rate will not change until you click disconnect and then connect on the toolbar).

The values are: 1=2400, 2=4800, 3=9600, 4=14,400, 5=19,200, 6=28,800, 7=38,400, 8=57,600, 9=115,200, A=230,400.

2. Error reset: "e;" turns off the error LED.

3. Memory:

a. Save setup: "mss;" Memory write setup stores the current setup including modes, channels, and configuration.

b. Load setup: "mls;"; Memory load setup. Loads a saved setup if one is present.

c. "msd;" Memory save default. Stores the current setup as the power up default. Replaces (over writes) a previously stored default. You may wish to first test using the write and load saved setup commands.

Should you accidentally save a setup that renders the AD inaccessible (high data rate, etc.) or a year has passed and you forgot the baud rate ... there is an out. The device *always* powers up at 9600 baud; it checks for and, if present, loads the stored default after the led's stop flashing (about 2 sec). During the time the led's are flashing pressing the escape (ESC) key will cause the module to skip the load stored default. It will come up with the original safe defaults shown in this manual.

d. "mpd;" Purge stored default:

Restores the power up defaults of the module to those shown in this manual.

e. "mld;" loads default settings.

4. "\$@R" will cause a power on restart (case sensitive - capital R).

5. Echo received characters: "ck;" Keystroke echo. Incoming characters are transmitted back to the serial source sending the characters. "ck0;" stops echo.

Why both rate and interval modes?

The two modes allow for the precise control of data collection from over 3000 samples per second to less than 1 sample every 60,000 ms (or sec). At low data rates the resolution of rate mode is poor. For example, you can only choose between 1 or 2 samples per second, corresponding intervals of either 1000 or 2000 ms. In interval mode you can choose 1001, 1002....1999 ms, you have all 1000 choices.

On the other hand, at high data rates in interval mode, you can only have either 1 or 2 ms corresponding to rates of either 1000 or 500 samples per second. Kind of frustrating when you need high data rates and the highest is between 500 and 1000 samples per second. In rate mode you can choose 500, 501,....999 samples per second.

So between the two modes you have the capability to precisely optimize the A/D to your situation.

Averaging modes?

The different methods allow you to choose the most effective combination for the noise content in your environment. They relieve the main computer of performing these common tasks thereby freeing it up for other tasks. The filters reduce the need for high serial bandwidth. Timing accuracy is improved because the 504 microcontroller is a dedicated single task device as opposed to the multitasking time sharing OS in larger computers.

The sample averaging filter is the classic add them up and divide filter for gaussian (normal distribution) noise reduction. The median filter is for popcorn noise, and signals near the limits artifact. Burst mode averaging is over sampling and is helpful in reducing aliasing problems, specifically in removing power line noise.

Median Filter

I have always been amazed at how badly one outlier can trash the calculation of an average. If you pitch it, then the data is tight. Often it was one point in a series of repeat measurements that were averaged at data collection time. If you take the median of the repeats then the single measurement falls out; the mean and the median now coincide. This is as it should be; because the outlier was due to a defect in the measurement. It is perfectly valid to reject the outlier since the median is a more accurate estimate of the central tendency in narrow distributions. On the other hand, I have seen people pitch multiple points out of a broad statistical distribution until the remaining points yield a value in agreement with what they thought (or were told) it should be. I digress, back to the question of how did the crazy number get there in the first place.

Shot (or popcorn) noise is due to very short spikes polluting the signal. In audio systems this is heard as a click or pop. The spike amplitude is generally sufficient to max out the electronics. In processing a signal to remove noise by averaging this is the most feared type of noise.

Let's say we have a perfect signal conditioner and a perfect A to D Converter. The input signal is one volt to the AD. The range of the AD is zero to five. We are acquiring five points and averaging them so: $1+1+1+1+1 = 5$ divided by five equals one volt; perfect. A defective bulb in a fluorescent light fixture two floor down begins to flicker. Now we have $1+5+1+1+1$, $1+1+1+1+1$, $5+1+1+1+1$ or averages of $= 1.8, 1.0, 1.8 \dots 1.0$ volts. Increasing the averaging to say 10 (and assuming this doesn't pick up even more pops) only reduces the average to 1.4. We are definitely in deep doo, doo, and are faced with two choices, either find the source and remove it (politically difficult if some big dude is using the equipment) or filter out the noise somehow.

Enter the median filter. The median of a bunch of numbers is the number that has as many numbers below it as above it. For example, the median of 1,2,3,4,5 is 3; the median of 5,1,2,3,4 is 3; the median of 1,1,5,1,1 is 1 and so forth. We simply sort the numbers just as we would sort cards, and pick the one in the middle. If several readings are the same (you can get shot for this in Nevada), then we simply list them as they are encountered in the sort, for example, 5,2,1,2,3 becomes 1,2,2,3,5 when sorted; the median is 2, the one in the middle. (This is called a 5-point median, five is an odd number so it is an odd median; even medians, for example four, are defined as the average of the two numbers in the middle.)

From a statistical perspective what we are trying to find is the "central tendency" of a bunch of numbers (a distribution). If the distribution is normal (as would be expected with Gaussian noise) then the distribution will have a tail. To get rid of the noise, you want the value at the peak of the distribution. Clearly this can be done by averaging all the numbers. However, in the real world, you only get to average a sample of all the numbers, say 10 points to estimate the mean. Ever so often you will get a value from the tail of the distribution in your 10 points; this trashes the average for that 10. There is no way to know when or how often this will occur when rolling the dice. This is why a sample median is a better (more reliable) estimate of the central tendency, when the distribution is narrow (low gaussian noise), than a sample average.

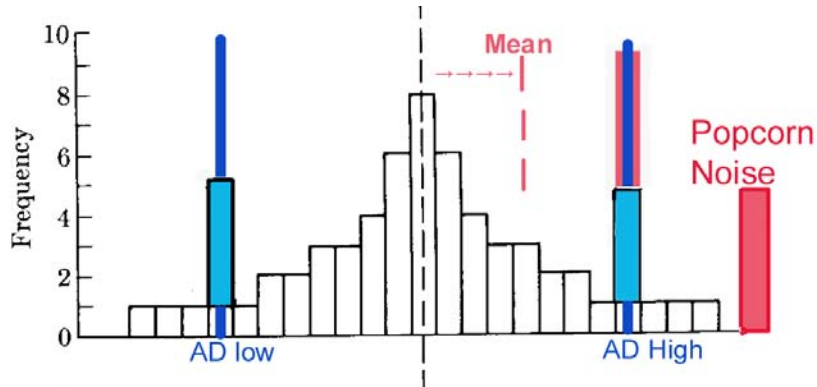


Fig. 1 Popcorn noise spikes appearing as AD converter values at the high range limits. The blue bars show the effect on the normal distribution of under range and over range noise.

Yes, but if there are an equal number of high pops and low pops then the noise cancels out when the signal averaged.

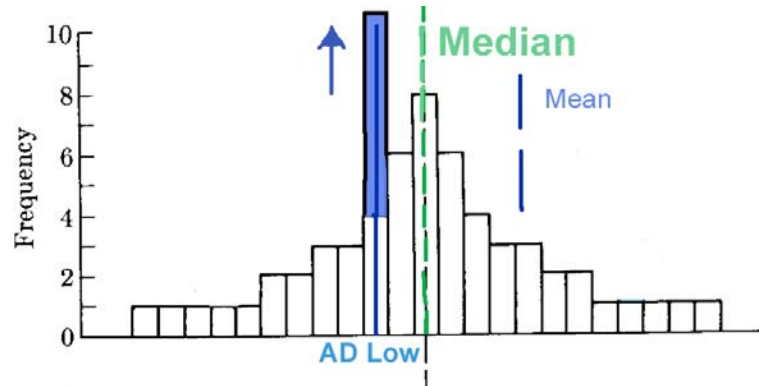
The pops will cause fixed low or high values that average to zero in bipolar mode or half-range in unipolar mode. This tends to pull the signal average to the center of the AD range. If the signal average is near midrange, then the effect is minimized. The further from center the signal is, the more the average is pulled toward midrange.

For examples consider these numbers in bipolar mode, range five volts: -5, -5, -5, 1, 1, 1, 1, 5, 5, 5; this is 10 readings about a true signal of one volt with six spikes. There are equal numbers of high (five volt) and low (-5 volt) noise pops. It seems like they should cancel, but the series adds up to four on 10 readings giving an average of 0.4 volts. This is a lousy estimate of the one volt actual value. Note that a five volt true value would average to two volts in this case. Of course a zero volt true would be dead on.

Median versus averaging at lower and upper AD limits

An analog to digital converter has definite minimum and maximum outputs. For a 12 bit AD these are 0 and 4096. In unipolar mode, with a range of 0 to 10 volts, any signal at or below zero will be a 0, and any signal at or above 10 volts will give 4096. So what happens to our estimates if we input a signal with a normal distribution about a DC level of 1 volt, and gradually lower the DC level toward 0 volts? To find out let us average the output from the AD and compare that with the median. As we approach 0 the tail of the distribution falls below 0 giving us more and more 0's in the AD output, and therefore in our average. When the peak is at zero we still have a positive value from the averaging filter due to the upper half of the distribution.

Fig. 2 AD lower range limit effect on Mean and Median of a low level normally distributed signal.



So the bad news is that to get a zero output from averaging we have to have the whole distribution below zero. The good news is that the median filter tracks the peak and goes to zero as the peak crosses zero; there are as many zero AD readings below the peak as there are positive readings above, and it is the middle reading in the sort that matters. A similar situation occurs at the upper end of the range or in bipolar at the lower end.

Things to watch out for

A median filter should be used with an averaging filter when the peak of the distribution is wide and flat. The median is a mediocre estimate of the central tendency when the distribution is flat topped (look up kurtosis in a statistics text).

Burst Filtering (over sampling)

Aliasing occurs when a spurious signal corrupts the data signal due to the periodic sampling rate of the AD. A noise signal is aliased into the data if it is periodic, and a multiple of the sampling rate. This is the strobe light effect where the motion of a fan blade may be made to appear to be stopped, slowed or reversed. It also makes stagecoach wheels appear to turn backwards in old movies when the frame rate of the camera synchronizes with the rotation of the spokes in the wheel. With an AD converter it often looks like a slow sinusoidal drift in the sampled data.

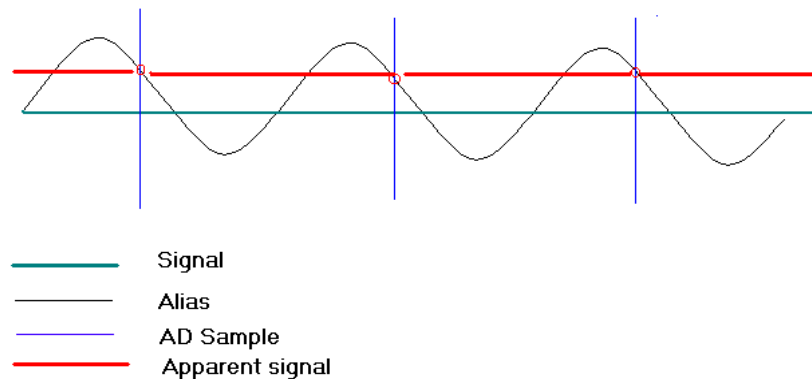


Fig. 3 The stopped case adding to a DC signal.

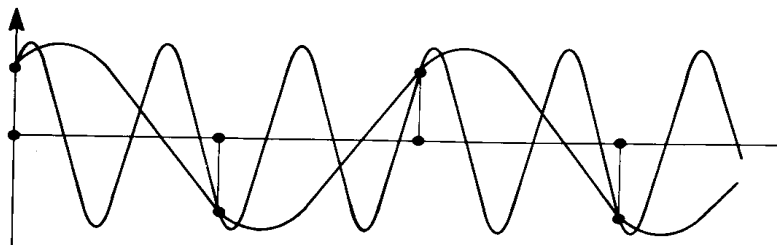


Fig. 4 The slowed case. A high frequency alias appearing as a low frequency signal.

One way to prevent this is to place a low pass filter ahead of the AD. This is usually a simple first order resistor capacitor network. We deliberately restrict the bandpass on our signal conditioners to reduce the aliasing problems for attached AD's. The design rule is to limit the bandwidth to only that needed to amplify the signal to be measured. Using a wider bandwidth amplifier than is needed increases noise contamination and the likelihood of aliasing due to unknown interference.

Another way is to increase the sampling rate to the point where the interfering signal frequency would have to be beyond the bandpass of the AD input signal conditioners. The Burst Averaging option on the 504 provides this over sampling capability. For example, the 103 signal conditioner has a 3 db bandpass or 300 Hz. A burst sampling rate of 600 Hz would push the lowest frequency potential aliasing signal to 600 Hz; this spurious signal would be severely attenuated by the 103 signal conditioner and never reach the AD input.

The requirement for periodicity in the interfering signal reduces the likelihood of aliasing problems. The requirement that it be a multiple of the sampling rate also makes the problem less likely; furthermore the frequency response of signal conditioners ahead of the AD generally decreases at higher frequencies.

The Governor

(Remember when rental trucks had a governor? No matter how hard you pushed the gas pedal - 50 mph was tops)

Default is off. To turn on issue 'cg!;' command.

The maximum attainable conversion rate is affected by a confusing array of settings. Specifically, the data output rate cannot exceed the baud rate. This is dependent on the baud rate setting which can assume 10 values. The time to transmit data is also affected by format. One channel in binary mode requires 3 bytes: the 0xFF start character and 2 for the number. A byte requires 10 bits to be transmitted. So you got to divide the baud rate by 10 or multiply the bytes by 10 to get the bytes per second limit. However, there are several formats with volts being the longest: 0xFF start, followed by sign, followed by a digit, decimal point and four more digits, and finally by a carriage return. More than one channel will cause a comma to be inserted, and if you are printing the channel number, then this also needs to be counted. Internal code time (?) and conversion time (60 us) also need to be factored in.

Okay, so it becomes a guessing game. You try it - if it fails, then you drop the rate. You end up bouncing between trial, error, calculator and scratchpad (and the local pub).

The CyQ 504 *automatically* adjusts the rate downward to the maximum that will allow the data to be reliably transmitted. If you enter a rate that is too high (interval that is too short) then you will see a series of error messages on the screen - the red error light will flicker. Then the error light will stay off and smooth data output will resume. If you enter the ? command you can see what the maximum safe rate (or shortest interval) is for your current baud rate and format settings.

In interval mode the top available rate is 1000 samples per sec at 1 ms interval. The governor may need to increase the interval to accommodate channels or formats. At low data rates the governor functions mainly when averaging is being done. The number of samples to average and the averaging interval are fixed. The governor automatically increases the interval between data outputs to accommodate the averaging. We assume that the averaging is being done to reduce noise just before data transmission, therefore we leave it alone.

Note that we let you run off of the road a few times before we set a speed limit.

ERROR Handling

If an error occurs then a line feed will be followed by one or more r 's followed by an error message. The error light will come on, and can only be turned off with an 'e;' command.

cmd Command buffer overflow. More than 16 characters were received before a command could be processed. Either a ';' was dropped, or there was noise, or a key is stuck on the keyboard.

Rx Receive buffer overflow. Too many commands were sent to fast. Usually happens when commands are being sent by a computer program. Place some wait(1 or more ms) statements in the code.

Tx Transmit buffer overflow. Everything was not transmitted before the next data was ready to be loaded for transmission. The buffer will go into a wait loop for about 1 ms to give the uart buffer time to clear; if this fails then this error will be set. The delay may trigger the governor producing a speeding error. Baud rate needs to be increased or the sampling rate decreased.

Speed Acquire rate is too high or baud rate is too low ... the rate will automatically be decreased.

Other errors will result in the following:

The command will be printed up to the offending character, an underline '_' will be printed, this will be followed by a letter where:

'?' means the last character was invalid

'L' a logical was expected: t,f,0,1.

'=' an equal sign was expected. (In some instances both an L and = error will occur.)

'N' a number was expected, or the number is out of range, e.g., a bad channel number.

'X' a hexadecimal number is out of range or the character is not a hexadecimal number.

For example:

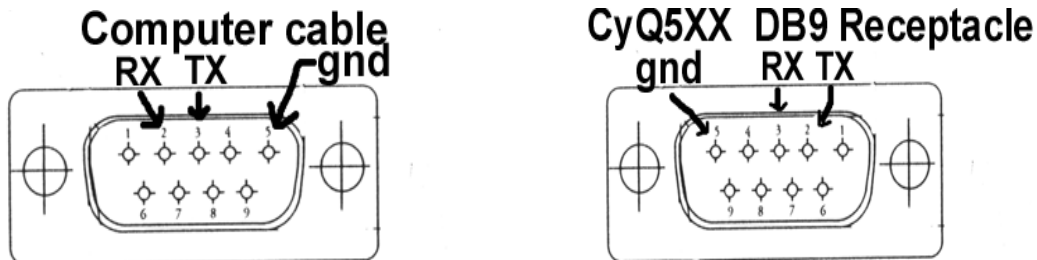
```
type cz;  
***cz_?
```

is the error message because 'z' is not a valid configure command character.

SERIAL HARDWARE INTERFACE RS232C

This is a standard serial cable for DB9 interconnection. It is NOT a null modem cable. The crossover from transmit to receive is done at the receptacle on the A/D. The required condition is that transmit from the computer connects with receive at the peripheral (A/D) while transmit from the A/D connects to receive at the computer.

CyQ.....	CyQ5XX A/D
TX PI 3.....	Pin 3 RX
RX Pin 2.....	Pin 2 TX
GND Pin 5.....	Pin 5 GND



The interface is implemented with a MAX232A integrated circuit from Maxim Inc. It has been tested to 115.2 kb.

The jackscrew standoffs are threaded into a captive nut; there is no hardware to come off inside the case. If the standoffs come loose, then simply remove and reinsert them. Do not over tighten or you will break them; then you really go a mess.

Note: The DB9 and DB25 connector pinouts are completely different. Specifically - pins 2 and 3 are switched for Tx and Rx.

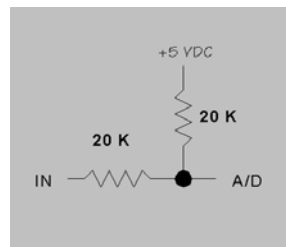
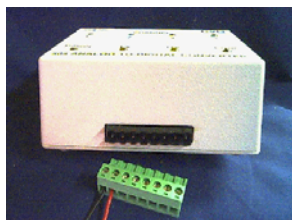
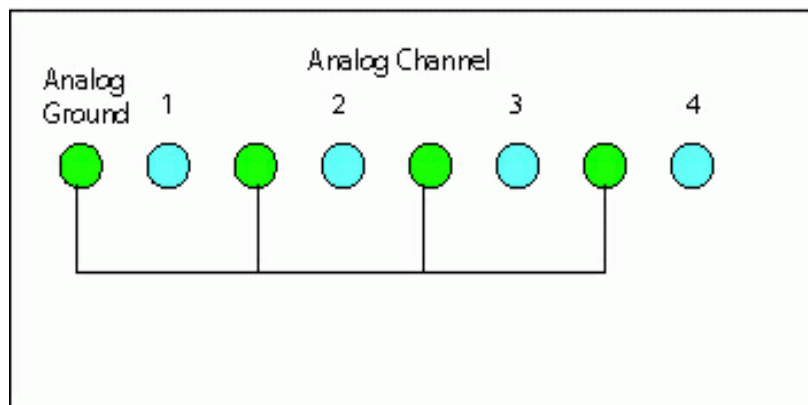
INPUTS

POWER

A 7.5 to 12 vdc unregulated wall transformer with 100 ma or more current capability will work. This is a 2.1 mm barrel connector. Center positive. The red handle power switch is directly above. Up is on and down is off. This input is diode protected against reverse polarity, and surge protected above 18 volts. Further protection is provided by a 500 ma self resetting fuse.

A/D Channels

Wire connect input numbers line up with the LED channel numbers from left to right. On the wire connect plug inputs the signal pins alternate with analog grounds ; strip wire and bend the bare wire over the insulation - insert into opening (insulation and all) and tighten screw.



SPECIFICATIONS

- Full scale input span: ± 4.999 , 0 to 9.999 vdc.
- Input resistance: 20 kohm to 25 pf to either 20 kohm to +5 vdc (bipolar) or ground (unipolar) mode.
- Leakage current: ± 1 ua.
- Resolution : 12 bits
- No missing codes: 12 bits
- Integral linearity error: ± 2 bits
- Differential linearity error: ± 1 bit
- Offset error: ± 5 LSB (12.5 mv)
- Offset error stability: ± 1 LSB
- Offset error match: ± 5 LSB system.
- Gain error: ± 4 LSB
- Noise: 30 uv rms
- Power supply rejection: 70 dB
- Power supply: 7.5 to 12 vdc unregulated wall transformer. Dropout is 6 volts.
- Power supply current : 36 ma
- Size: width 4 in.(10 cm), depth 3 in.(7.6 cm), height 2.0 in. (5 cm).
- Weight: 7.7 oz (218 g).
- Material: ABS plastic case, textured gray, 4.49 X 3.25 inches (11.2X8.3 cm).

LEGAL STUFF

Limited Warranty

CyberSense warrants to the original purchaser or end-user complete satisfaction for 90 days - money back or exchange; we pay shipping within USA. This product is warranted to be free from defects in material or workmanship for a period of five years. During this period the product will be replaced or repaired without charge for either parts or labor.

This warranty is voided if the product is modified, misused, subjected to abnormal environments. This warranty is void if repairs are attempted or the case is opened for any reason without our permission. We will gladly assist with troubleshooting your setup by phone or email.

Fragile sensors such as force, pressure, temperature are excluded from this warranty, and are sold without warranty. You must test immediately upon receipt. We test immediately prior to shipping. If you use a sensor, then you broke it.

Purchase price refund including shipping (exception on shipping: international orders) for 90 days or repair and replacement for a period of 5 years (terms may vary internationally) are the **ONLY REMEDY** of the purchaser. This warranty is in lieu of all other warranties either express or implied; specifically, any warranties of implied merchantability or fitness for a particular use or purpose. CyberSense shall in no way be liable for indirect or consequential damages of ANY kind or nature.

Some jurisdictions do not allow exclusion or limitation of consequential or incidental damages, or how long implied warranties last; therefore, the above limitations may not apply to you. This warranty gives you specific legal rights. In some states you may have other rights.

Use in life support systems: **NO!**

Trademarks

CyQ and Cymograph are registered trademarks of CyberSense, Inc.; CyberSense, Inc. is a trade name. Any other trademarks, trade names, service marks, or service names owned or registered by any other company and used in this manual are the property of their respective companies.