

GigaLog

Table of Contents

1 Getting started.....	6
2 Hardware.....	7
2.1 Gigalog S.....	7
Analogue Input Options.....	10
Other inputs / outputs.....	12
xc.....	12
xc<pin>= <value>.....	12
rl<number>=0 1.....	12
2.2 Gigalog M.....	13
mcu=<frequency>.....	14
Analogue Input Options.....	18
Other inputs / outputs.....	20
xc.....	20
xc<pin>= <value>.....	20
mt[1 3]= 0 1.....	21
mot[0 1]= -1 z 0 1 2.....	21
mot <steps> [<speed>].....	21
mot 0.....	21
2.3 Sample rates.....	22
2.4 From raw values to Engineering values, Calibration, Output format.....	22
2.5 Alarm.....	23
3 Command mode, Configuration.....	24
3.1 Enter Configuration, Command mode.....	24
3.2 Syntax.....	25
3.3 Files on the disk.....	25
3.4 Configuration, Commands.....	25
h * all.....	25
h hard conf com adc disk flow lp debug remote rq lcd.....	25
h <command> <pattern>*.....	25
d [-b][-c][-a][-l][-q][-m].....	25
z.....	25
zc.....	25
st.....	25
go.....	25
3.5 Configuration Rs232 data.....	25
rs[0 1 ..7]=[-c d m i s],<baud>,<filename>,<buffer size>,[7 8 e o r 2].....	26
rs[ch[.ch2]]].....	26
fs=<frame>.....	26
fe=<frame>.....	26
3.6 Configuration Analogue Inputs.....	27
an= <path>[,h].....	27
an+.....	27
ad= <storage rate>[,<ADC rate>].....	27
as= <frame>.....	27
am= <frame>.....	27
ae= <frame>.....	27
a<ch>[-<ch2>]=[-d c a k p z][<term>][*<m>][+<p>][,<c>][<<min>][><max>][,n=<name>][,d=<diff>] [,p=<reference>][l=<lcd>][m=[a][m][M]].....	27
a<ch>=v[*<m>][+<p>][,<c>][,c=<expr>][,n=<name>][l=<lcd>].....	27
a<ch>=vc[*<m>][+<p>][,<c>][,c=<expr>][,n=<name>][l=<lcd>].....	28
a<ch>=vp[*<m>][+<p>][,<c>][,n=<name>][l=<lcd>].....	28
a[<ch>[-<ch2>]] [<n>[,<r>]].....	28
ar[<ch>[-<ch2>]] [<filter>,<series>].....	28
av[<ch>]= <n>.....	28
ax=<adc config>,<buffer size>,<m samples>.....	28
ax.....	28
et=<et0>,<et1>.....	28
po=<threshold low>,<threshold high>.....	28
tr=[+ - i + - i]<channel>,<threshold low>,<threshold high>[,<prologue>,<epilogue>[,<filter>]].....	28

iadc.....	29
3.7 Configuration Disk, Files.....	30
format [<drive>][<volumename>].....	30
ls [<path>].....	30
fa <path> <text>.....	30
touch <path> <text>.....	30
cp <path> <path>.....	30
mv <path> <path>.....	30
cmp <path> <path>.....	30
rm [-f] <path>[, <path>].....	30
md <filename>.....	30
cd <path>.....	30
pwd.....	30
up [-b -c] <path> [-<off>] [<start>].....	30
cat [-b -c] <path> [-<off>] [<start>].....	30
uc=<n>.....	30
fd <filename> +<hex data> !<crc>.....	30
dx=<files>,<cache buffers>,<retry> <sd2>.....	31
dk.....	31
3.8 File names, Circular Data Storage.....	31
3.9 Communication.....	32
ec[0 1 ..] <text>.....	32
tm [-a] [-m] [-r] [0 1 ..].....	32
3.10 Macros, Execute commands.....	33
m<n>=<stime>,<rtime>,<text>.....	33
m<n>=<onCondition>,<text>.....	33
mx<n> [<time>].....	33
ex [-s] [-b] [-v] [r] [-e] <file>.....	33
bg <commad line>.....	33
3.11 Low power mode.....	34
lp=<mode>,<uptime>,<options>*.....	34
lp.....	34
3.12 Configuration, Miscellaneous.....	35
bn=<name>.....	35
nm=<name>.....	35
pw=<password>.....	35
of= <dec. point><separator>.....	35
dt=yymmdd.....	35
ti.....	35
ti=hhmm[ss] [dt=yymmdd].....	35
rtc.....	35
t.....	35
wt <n>ms <n>s <n>m <n>h.....	35
wd=0 1.....	35
bo= 0 1.....	35
dl [<file>].....	35
3.13 Conditional Instruction.....	36
if <primary> = = > > = < <= <primary> <true commands> [else <false commands>].....	36
3.14 Configuration, Frames.....	36
3.15 Configuration, Special characters.....	36
3.16 Configuration, Counters.....	36
3.17 Maintenance, Debug.....	38
mcu.....	38
pio [<port>[=<value>]].....	38
dk.....	38
mux.....	38
m <address> [<count>].....	38
mal.....	38
sn.....	38
db=<n>.....	38
pr.....	38
ax.....	38
xbl.....	38
xrl.....	38
xsZ.....	38

xreset.....	38
xexc.....	38
xwd 123.....	38
xz.....	38
4 GigaTerm.....	39
4.1 Connecting to a local board.....	39
4.2 Connecting to a remote board by TCP/IP.....	39
4.3 GigaTerm as a terminal.....	40
4.4 Handling configurations.....	40
4.5 Inputs.....	41
4.6 From raw values to Engineering values, Calibration, Output format.....	42
4.7 Display current values of inputs.....	43
4.8 Macros.....	44
4.9 Upload.....	45
4.10 Firmware Download.....	45
4.11 First Help.....	45
5 GigaData.....	46
6 Remote Communication.....	47
6.1 Mobile network.....	48
Quick guide: How to send your data to an Internet server.....	48
Connecting, Configuration of the modem, modem server.....	49
apn=<apn name>,<apn user>,<apn password>.....	49
mm [-v].....	49
mm=<log>,<timeout cmd>,<timeout connect>,<timeout disconnect>,<timeout ftp command>.....	49
mmstart [-v] [v=<verbose>] [<tmdm> <tnetwork> <tall>].....	49
mmstop.....	49
mmq.....	49
at<cmd>.....	50
Sending, receiving SMS.....	50
sm -v <Number>,<Text>.....	50
6.2 FTP.....	51
ftp=<address>,<userid>,<password>,<dir>[<mode>].....	51
ftppop[en] [-s] [<infoFile>].....	51
ftp [-c -d -t -s]* <filename> [<serverfilename>].....	51
ftpget [-a -s]* <serverfilename> [<filename>].....	51
ftpinfo [-s] <infoFile>.....	51
ftpmx <n>.....	51
ftpclose [-s].....	51
6.3 GNSS / GPS.....	52
gps=<mode><port>.....	52
gps.....	52
gpsgo.....	52
gpsti.....	52
tz=<tz offset>,<daylight saving additional offset>.....	52
6.4 Wifi.....	53
wf=<ssid>,<password>.....	53
ip=<ip>,<ip mask>,<ip gateway>,<dns1>,<dns2>.....	53
wfstart [r][v=n][vx=n].....	53
wfstop.....	53
wf [v][r][v=n][vx=n].....	53
wfdownload <file>.....	53
6.5 Registration.....	53
nm=<hostname>.....	53
pw=<password>.....	53
reg.....	53
6.6 Http Server.....	53
6.7 Bluetooth.....	54
7 Connecting several boards.....	55
7.1 Gigalog Master with two GigaLog Slaves.....	56
7.2 Gigalog Master with a Remote Acquisition Module.....	57
7.3 Configuration.....	58
rq=<on>,<gap time>,<module timeout>.....	58
rq<module>=<rs port>,<slave id>,<protocol>,<input0>,<inputs>[,<options>].....	58
rqz.....	58
rq v=<verbose>.....	58

rqs1=<slave id>,<protocol>.....	58
7.4 Modbus Commands.....	59
mbscan[ch] [<timeout>].....	59
mb[ch] <hexstring>.....	59
mbrd[ch] <id> <register> <format>.....	59
7.5 Low power Mode.....	60
7.6 Gigalog S.....	61
8 Display.....	62
8.1 Connection and power supply.....	62
8.2 Page Bar graph.....	62
8.3 Page Log Data.....	63
8.4 Page Scope.....	64
8.5 Page Installation.....	65
8.6 Page Command terminal.....	66
8.7 Configuration, Commands.....	67
grt=<language>,<channels>,<channels per page>,<seconds>,<minutes>,<hours>,<days>,<bits>,<start> 67	67
grst=<start>.....	67
gr<ch>=<input>,<show>,<min>,<max>.....	67
grb[<n>]=<in0>,<ins>,<min>,<max>.....	67
gre[<n>]=<input>,<style>+,<x>,<y>,<wd>,<ht>,<min>,<max>[,<text>].....	67
gry= [c][b][i][j][t][s].....	67
grc=<ch>.....	67
grw [<color>,<text>].....	67
grpop <time>,<text>.....	67
gri <filename>.....	67
grx.....	67
grp <filename>.....	67
gra.....	67
grlp=<timeout>,<mode>.....	67
groff.....	67
gron.....	67
8.8 Electronic Paper Display (epd).....	68
epd=<display>.....	68
epd=.....	68
epd[0-19]=<x>,<y>,<wd>,<ht>,<style>,<text>.....	68
epd.....	68
epd d.....	68
epd z.....	68
epd on[=<rttime>].....	68
epd off.....	68
epd demo.....	68
9 Application Notes.....	69
9.1 Memory cards, File lengths, Transfer to the PC.....	69
9.2 Tuning the ADC.....	70
9.3 Calculation from the analogue input raw value.....	72
Virtual input, calculation.....	72
Long time counter.....	72
9.4 Temperature Sensor LM60.....	74
9.5 Thermocouples.....	75
9.6 Pt100, Pt1000.....	76
9.7 Low power mode current.....	78
9.8 OEM Version.....	79
10 Gigalog S.....	80
10.1 Technical Summary.....	80
10.2 Component placement.....	81
10.3 Schematics.....	82
11 Gigalog M.....	83
11.1 Technical Summary.....	83
11.2 Component placement.....	84
11.3 Schematics.....	85

Information in violet background is only for Gigalog S.

Information in green background is only for Gigalog M.

The GigaLog board is supplied with a firmware and the GigaTerm PC program to configure the board, to upload recorded data, to download another firmware, and the GigaData PC program, to display the data.

This manual does not show, how to program the GigaLog board in C. Programming your own firmware is only necessary, when the supplied firmware cannot be configured for your special application.

© Andreas Meyer,
Version 2301, January 2023

1 Getting started

You need

- A GigaLog board with or without housing
- A power supply from 6 to 13 V.
- A standard USB cable.
- Or a RS232 null modem cable to connect the board to a COM port of your PC (female 2-3, 3-2, 5-5 female) (Manual chapter 4.1)
- A micro sd memory card
- A PC, running Windows
- The GigaLog package installed on the PC: GigaTerm, GigaData.

Put the sd memory card into the header.

Power supply, 6 to 13 V DC.

Power on the board.

Use the PC utility GigaTerm to communicate with the board.

Null modem cable from the PC COM port to GigaLog RS232 port RS0.

GigaTerm: Select the COM port at 115200 baud, click on Open to open the port.

You may also use the USB to connect to the board

Click on  to read the current configuration from the board. The board will now answer:

```
GigaLog ...
rs0=c,115200,gigalogrs0.txt,300,0 rs1=c,115200,gigalogrs1.txt,300,0
...
board running
```

Stop the running board (ch 3.1).

```
st
ok
```

Display all inputs.

```
a
a00: 8022881 1249.698mV a01: 6688844 1041.900mV a02: 5795473 902.742mV a03: 5126645 798.561mV
a04: 4589396 714.876mV a05: 4143864 645.476mV a06: 3751096 584.296mV a07: 3392776 528.482mV
a08: 2448474 381.391mV a09: 2448496 381.394mV a10: 2448487 381.393mV a11: 2448483 381.392mV
a12: 2448851 381.450mV a13: 2448510 381.397mV a14: 2448850 381.450mV a15: 2448694 381.425mV
```

Set sampling rate to 1 second. (3.3). Inputs a0 and a1 are already declared as analogue inputs. Start the board.

```
ad=1s
ok
go
resume
```

The board now starts sampling data. Wait some seconds. Stop the board again. List the files on the sd memory card.

```
st
ok
ls
in.txt          352
```

Send "up in.txt" to the board, to display the stored data.

```
up in.txt
>2019:07:03 11:51:24 946.684 225.440
>2019:07:03 11:51:25 946.693 225.452
>2019:07:03 11:51:26 946.690 225.445
>2019:07:03 11:51:27 946.690 225.447
EOF
```

Or use GigaTerm, tab Upload, to load the data into a file on the PC.

You may then click on the button "GigaData in.txt", to start GigaData, and display the data.

2 Hardware

2.1 Gigalog S

Power supply

Power supply recommendations.

	Voltage	Current	
Gigalog S	6-15 V	100 mA	
Gigalog S + Am320240	12 V \pm 5%	500 mA	close jumper VLCD
Gigalog S + Gsm65	7- 15 V	1000 mA	
Gigalog S + Am320240 + Gsm65	12 V \pm 5%	1500 mA	close jumper VLCD

SD card disk handling

The board uses a FAT32 file systems.

Memory cards with 32Gb or more use normally exFAT. You have to reformat the card before use.

The configuration format command can be used to format the sd card. (config: format)

When formatting a sd-card with a PC, select a big cluster size, to avoid useless FAT operations.

Avoid lots of data on the memory card. The first write access to the memory card after Reset will need a lot of time, to seek the first free cluster. This information will then be cached.

Avoid long data files. The first write to the file after Reset has to search the end of the file, which may take a lot of time. This information will then be cached.

The firmware does a verification read after each read, and each write operation. To increase speed, you may remove this verify, by changing the 3. parameter in the dx command.

When the LED on the board is on, the board is writing to the memory card.

Do not take off the card, or cut the power supply. This may damage the file system on it.

See also chapter Application Notes → Memory cards.

Display

A liquid crystal display, 2 lines x 16 characters, displays the state of the board.

The display shows the actual values of the inputs, independent of the storage of data. Use configuration command a= to change the name, the time, and the format to display for each channel.

The 2x16 display can be replaced by a 4x16 display, or by a 320x240 graphical colour display with touchscreen.

LED

The LED on the board signals:

- Flashing each second: Board is ready to work
- On: Writing to disk. Do not take off the memory card. Taking off the memory card may damage the file system on it.
- Flashing once all 10 seconds: Low power mode.

Rs232 interfaces RS0 and RS1

SubD 9 pole male communication port

1		
2	RxD	to GigaLog, configuration, data
3	TxD	from GigaLog, configuration
4		
5	GND	Common
6		
7		
8		
9		

The 2 serial ports RS0 and RS1 are configured at 115200 baud from the factory, and will accept commands as input.

Both ports can be configured, to store incoming data on a file on the memory card.

Rs485 interface RS2

RS2 a Rs485 port using the 2 screw terminals Dat+ and Dat-. Logical the port works like RS0 and RS1.

USB

When connecting the board the first time to an USB port of your PC, Windows detects a new peripheral. The board uses a standard Windows driver. Nevertheless Windows may ask for the driver. Direct Windows to the usb driver directory in the installed software, subdirectory "usbdriver". Windows treats the USB port as a COM port and gives it a free number.

STOP

Digital input Stop can be used to enable sampling.

Stop is open collector. Tie Stop to 0V to disable sampling. The run/stop switch on the box panel is tied to Stop.

The Stop input is available on the XRL header, pin 3.

If you connect this pin to XRL pin 4 (GND), then the board will not start sampling, when coming out of a Reset.

The "Stop" and "Go" command also disable, enable sampling.

The graphical LCD has a STOP/GO switch on the head line left.

When stopped, the board will not execute scheduled macros.

Analogue Inputs

Gigalog has 16 analogue inputs in single ended mode with reference to GND.

Pairs of inputs (a0-a1, a2-a3,...,a14-15) can be used in differential mode

The Adc is a Texas Instrument ADS1258 in delta-sigma technology.

The analogue signals pass by an integrated multiplexer with an 80-Ohm resistance.

Inputs have an impedance of > 10 M Ohm.

The ADC has a resolution of signed 24 bits.

Each input can be from -100 to +1300 mV.

The voltage reference AREF is a LM4041-AIM3-1.2 1225 mV \pm 1.2 mV at 25°C and \pm 10 mV at 0..70°C.

Typical resolution of the ADC for a perfectly stable input signal

Storage sample rate	Resolution in bits
< 10 ms	16
> 10ms, < 100 ms	18
> 100 ms	19

When used as digital input: A logical 0 is 0 to 0.8 V. A logical 1 is 1.2V to 3V. An open input is undefined.

ADC -> voltage

$$U = \text{ADC} * 1225 / 0x780000 = \text{ADC} * 0.1557668 \mu\text{V}$$

Voltage -> ADC:

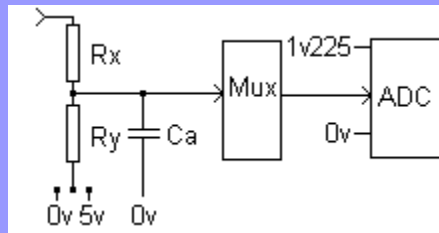
$$\text{ADC} = U(\text{mV}) * 6419.85$$

XA: The analogue inputs are also available on an IDC header for flat cable.


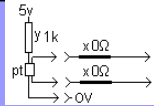



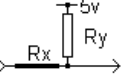

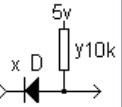



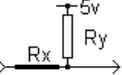

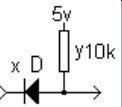
	2	4	6	8	10	12	14	16	18	20
	AVCC	A1	A3	A5	A7	A9	A11	A13	A15	AREF
	AGND	A0	A2	A4	A6	A8	A10	A12	A14	AGND
	1	3	5	7	9	11	13	15	17	19

Analogue Input Options

GigaTerm helps you to configure the inputs. Choose an input, choose the input type.



	Input type		Vi -> Vadc	Rx Ω	Ry Ω	Ca	Select	
a	Direct voltage -0.1 .. 1.2 V			0	-			
a2	Direct voltage -0.1 .. 1.2 V with RC filter			1k		220nF		$1/(6.28 * R * c)$
a3	1 kHz			10k		330nF		
a4	Higher voltages			10k	2k2		0V	$V_0 = V_i * R_y / (R_x + R_y)$
a5	-0.5 .. 6.5 V			10k	1k			$V_i = V_0 * (R_x + R_y) / R_y$
a6	-1 .. 13 V			33k	1k			$V_0 > -100 \text{ mV}$
a7	-2 .. 41 V			100k	1k			$V_0 < 1.25 \text{ V}$
a11	Negativ + Positiv voltage			1k	4k7	1k Ω	5V	
a12	-1 .. +1 v			22k	10k	1k Ω		
a13	Differential Inputs			10k		100n		
	-1.2 .. +1.2 V			(x 2)		(x 2)		
a14	Differential Inputs			10k	2k2	100n		
	-12 .. +12 V			(x 2)		(x 2)		
a10	Current			0	47		0V	$V_0 = I_i * R_y$ $I_i = V_0 / R_y$
a15	Current differential inputs			0	47	100n		$V_0 = I_i * R_y$ $I_i = V_0 / R_y$
	-25.. 25 mA			(x 2)		(x 2)		
k0	Small voltage -100 .. 125 mV	Thermocouple		0	-			$V_0 = V_i$
k1	Typ eK							
k9	Type J							
	Type K w/o compensation							
p	Resistance			0	1k		5v	
p1	Pt100			0	10k			
p3	Pt1000			0	1k			
	Pt100 3-wire			0	1k		5v	

p4	Pt100 4-wire			0	1k	5v	
d	Digital input			0	-		
d1	Digital input with pullup			0	10 k	5v	
d2	Digital input, higher voltage			D	10 k	5v	
c	Counter			0	-		
c1	Counter with pullup			0	10 k	5v	
c2	Counter, higher voltage			D	10 k	5v	

Other inputs / outputs

XC: Additional signals are available on an IDC header for flat cable.

2	4	6	8	10	12	14	16	18	20	22	24	26
VCC	VCA	PA4	PA11 TWCK(I2C)	PA19 RX(CAN)	PA21 TF(SSC)	PA23 MOSI(SPI) TD(SSC)	PB28 AD1 PWM1 TIOB2	PB30 AD3 PWM3 PCK2	AD6	DAC0*	DAC2*	DAC3*
GND	PA2 16 mA	PA3 16 mA	PA10 TWD (I2C)	PA15 TCLK2	PA20 TX(CAN)	PA22 SCLK(SPI) TK(SSC)	PA24 MISO(SPI)	PB29 AD2 PWM2 PCK1	AD5	AD7	DAC1*	GND
1	3	5	7	9	11	13	15	17	19	21	23	25

Hardware commands:

xc

display the state of the pins

xc<pin>= <value>

<pin>= 1..26

set a pin to a new input type, and or a new value

<value>= 0 | 1 | i | u for a digital pin

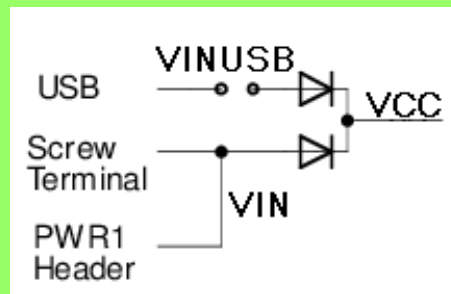
<value>= <n> | <n>V | <n> mV for a DAC. (Option)

rl<number>=0|1

Set relay <number> to state 0 or 1. The relay is open in state 0, closed in state 1.

2.2 Gigalog M

Power supply



Power supply recommendations.

	Voltage	Current @6V	Current @12V
Gigalog M	6 - 13 V	200 mA	200 mA
+ hy35a	6 - 13 V	100 mA	70 mA
+ mobile modem	6 - 13 V	1000 mA	600mA

The board can be supplied by the USB port. As soon, as Vin is higher than the USB voltage, Vin will power the board. USB voltage is typically 5V, and often lower than 5V. This is sufficient, to run the board, but results from the analogue inputs may have not the full resolution. Moreover the USB port of a computer, or a hub may be unable, to deliver the needed current.

Do not overload the computer USB port.

Do not use the USB port as power supply, when using the modem.

Do not use the USB port as power supply, when using a motor.

SD card disk handling

The board uses a FAT32 file systems.

Memory cards with 32Gb or more use normally exFAT. You have to reformat the card before use.

The configuration format command can be used to format the sd card. (config: format)

When formatting a sd-card with a PC, select a big cluster size, to avoid useless FAT operations.

Avoid lots of data on the memory card. The first write access to the memory card after Reset will need a lot of time, to seek the first free cluster. This information will then be cached.

Avoid long data files. The first write to the file after Reset has to search the end of the file, which may take a lot of time. This information will then be cached.

The firmware does a verification read after each read, and each write operation. To increase speed, you may remove this verify, by changing the 3. parameter in the dx command.

When the LED on the board is on, the board is writing to the memory card.

Do not take off the card, or cut the power supply. This may damage the file system on it.

See also chapter Application Notes → Memory cards.

Display

The board may use a 320x240 graphical colour display with touch screen: HY35A.

The display shows the actual values of the inputs, independent of the storage of data. Use configuration command a= to change the name, the time, and the format to display for each channel.

LED red

The LED on the board signals:

- Flashing each second: Board is ready to work
- On: Writing to disk. Do not take off the memory card. Taking off the memory card may damage the file system on it.

LED NET green

Displays the state of the modem:

off	modem off, or not ready
on	searching network
flashing 200ms	Registered 4G
flashing 800ms	Registered 2G/3G

See commands mmstart, mmstop to control the modem.

Set the MCU speed

mcu=<frequency>

Set the processor clock of the microprocessor.
Legal values are 300, 150(default), 75.

Rs ports

Serial communication channels

Rs0	Rs232
Rs1	Rs232
Rs2	Rs485
Rs3	XC
Rs4	XC
Rs5	Rs485
Rs6	XC
Rs7	Modem
Rs8	USB
Rs9	LCD terminal
Rs10	IP http-server shell
Rs11	Bluetooth

Rs232 interfaces RS0 and RS1

SubD 9 pole male communication port

1		
2	RxD	to GigaLog, configuration, data
3	TxD	from GigaLog, configuration
4		
5	GND	Common
6		
7		
8		
9		
10	VIN	not available on the SubD connector

The 2 serial ports RS0 and RS1 are configured at 115200 baud from the factory, and will accept commands as input.

Both ports can be configured, to store incoming data on a file on the memory card.

Rs485 interfaces RS2 and RS5

RS2, and RS5 are two Rs485 ports using 6-pin IDC headers for flat cable.

1	GND
2	WKUP
3	DAT+
4	DAT-
5	GND
6	VRS2 / VRS5

The jumper field JRS2RS5 controls the voltage on the RS485 buses for the slave modules



The Ip= command can switch MT3, and thereby the power of the slave modules on, when the master wakes up from low power mode.

RS2, RS5 are empty places on the board for RS485 line termination resistors.

WKUP is an open collector line, to wake up all boards on the bus from low power mode.

In low power mode, the master uses this signal, to wake up the slave boards.

The signal is also available on the CTRL connector.

RWKUP is an empty place for a pull-up resistor on the open-collector WKUP line.

USB

When connecting the board the first time to an USB port of your PC, Windows detects a new peripheral. The board uses a standard Windows driver. Nevertheless Windows may ask for the driver. Direct Windows to the usb driver directory in the installed software, subdirectory "usbdriver". Windows treats the USB port as a COM port and gives it a free number.

CTRL IDC header STOP WKUP

1	2	3	4	5
STOP	GND	WKUP	LED	VCPU

Digital input Stop can be used to enable sampling.

Stop is open collector. Tie Stop to 0V to disable sampling. The run/stop switch on the box panel is tied to Stop.

Use a SPDT centre off (single pole double throw) on-off-on switch, to control the board.

Switch

down	centre	up
Stop	Run	Wake up

The "Stop" and "Go" command also disable, enable sampling.

The graphical LCD has a STOP/GO switch on the head line left.

When stopped, the board will not execute scheduled macros.

Analogue Inputs

Gigalog has 16 analogue inputs in single ended mode with reference to GND.

Pairs of inputs (a0-a1, a2-a3,...,a14-15) can be used in differential mode

The Adc is a Texas Instrument ADS1258 in delta-sigma technology.

The analogue signals pass by an integrated multiplexer.

Inputs have an impedance of > 10 M Ohm.

The ADC has a resolution of signed 24 bits.

Each input can be from -100 to +4000 mV.

The voltage reference AREF is a LM4040AIM3-4.1 4096 mV \pm 4mV at 25°C and \pm 40 mV at 0..70°C.

Typical resolution of the ADC for a perfectly stable input signal

Storage sample rate	Resolution in bits
< 10 ms	16
> 10ms, < 100 ms	18
> 100 ms	19

When used as digital input: A logical 0 is 0 to 0.8 V. A logical 1 is 1.2V to 4V. An open input is undefined.

ADC -> voltage

$$U = \text{ADC} * 1225\text{mV} / 0x780000 = \text{ADC} * 0.520833 \mu\text{V}$$

Voltage -> ADC:

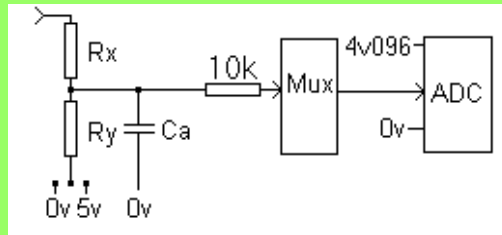
$$\text{ADC} = U(\text{mV}) * 1920$$

XA: The analogue inputs are also available on an IDC header for flat cable.


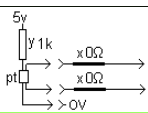



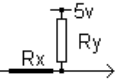
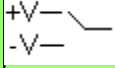
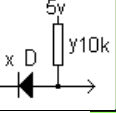



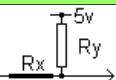
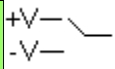
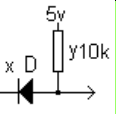
2	4	6	8	10	12	14	16	18	20
AVCC	A1	A3	A5	A7	A9	A11	A13	A15	AREF
AGND	A0	A2	A4	A6	A8	A10	A12	A14	AGND
1	3	5	7	9	11	13	15	17	19

Analogue Input Options

GigaTerm helps you to configure the inputs. Choose an input, choose the input type.



	Input type		Vi -> Vadc	Rx Ω	Ry Ω	Ca	Select	
a	Direct voltage -0.1 .. 4 V			0	-			
a2	Direct voltage -0.1 .. 4 V with RC filter			1k		220nF		$1/(6.28 * R * c)$
a3	50 Hz			10k		330nF		
a4	Higher voltages			10k	2k2		0V	$V_0 = V_i * R_y / (R_x + R_y)$
a5	0 .. 22 V			10k	1k			$V_i = V_0 * (R_x + R_y) / R_y$
a6	-1 .. 44 V			33k	1k			$V_0 > -100 \text{ mV}$
a7	-2 .. 135 V			100k	1k			$V_0 < 1.25 \text{ V}$
a11	Negativ + Positiv voltage			1k	4k7	1k Ω	5V	
a12	-10 .. +90 V			22k	10k	1k Ω		
a13	Differential Inputs			10k		100n		
	-4 .. +4 V			(x 2)		(x 2)		
a14	Differential Inputs			10k	2k2	100n		
	-40 .. +40 V			(x 2)		(x2)		
a10	Current			0	200		0V	$V_0 = I_i * R_y$ $I_i = V_0 / R_y$
a15	Current differential inputs			0	200	100n		$V_0 = I_i * R_y$ $I_i = V_0 / R_y$
	-20.. 20 mA			(x 2)		(x 2)		
k0	Small voltage -100 .. 125 mV Thermocouple			0	-			$V_0 = V_i$
k1	Type eK							
k9	Type J							
k9	Type K w/o compensation							
p	Resistance			0	1k		5v	
p1	Pt100			0	10k			
p3	Pt100 3-wire			0	1k		5v	

p4	Pt100 4-wire			0	1k		5v	
d	Digital input			0	-			
d1	Digital input with pullup			0	10 k		5v	
d2	Digital input, higher voltage			D	10 k		5v	
c	Counter			0	-			
c1	Counter with pullup			0	10 k		5v	
c2	Counter, higher voltage			D	10 k		5v	

Other inputs / outputs

XC: Additional signals are available on an IDC header for flat cable.

2	4	6	8	10	12	14	16	18	20	22	24	26
V33	V50	MT2	MT4	PA13	PA15	PA20 IADC9	PD16 RS6T	PD31 RS3T	PD19 RS4T		PD0 DAC1	GND
GND	VMT	MT1	MT3	PA12	PA14	PA19 IADC8	PD15 RS6R	PD28 RS3R	PD18 RS4R		PB13 DAC0	GND
1	3	5	7	9	11	13	15	17	19	21	23	25

Hardware commands:

xc

display the state of the pins

xc<pin>= <value>

set a pin to a new input type, and or a new value

o0 o1 i u	switch to digital pin, set to output 0, output 1, input, input with pullup
0 1	digital output, or dac
iad ad	switch to iadc: internal analogue to digital converter of the mcu
rs	switch to rs-port
dac	switch to dac: digital to analogue converter
<n> <n> V <n> mV	dac value: raw value (0..4095), Volt (0..3.3V), millivolt (0..3300mV)

Only some of these values are possible for each pin.

Motor interface

The pins MT1 (xc5), MT2 (xc6), MT3 (xc7), MT4 (xc8) may control

- two power switches (12V 1A),
- two DC motors(12V 1A), or
- a 2 coil bipolar stepper motor (12V 1A).

The pin VMT on the XC is the power supply of the motor interface. It is normally connected by the jumper JMT to VCC, the supply voltage of the board. You may cut JMT, to supply another voltage to the motor interface.

Power Switches:

mt[1|3]= 0|1

MT1 (Xc5) as a power switch

mt1=	MT1 (xc5)
-1, or z	open
0	GND
1	Vin

MT3 (Xc7) as a power switch

mt3=	MT3 (xc7)
-1, or z	open
0	GND
1	Vin

DC Motors:

mot[0|1]= -1|z|0|1|2

MT1, MT2 (Xc5, xc6) to control a motor.

mot0=	dc motor on MT1, MT2
-1, or z	idle
0	brake
1	forward
2	backward

MT3, MT4 (xc7, xc8) to control a motor

mot1=	dc motor on MT3, MT4
-1, or z	idle
0	brake
1	forward
2	backward

Stepper motor:

mot <steps> [<speed>]

Connect one coil to MT1, MT2 (xc5, xc6), the other to MT3, MT4 (xc7, xc8).

Execute <steps> half-steps forward (positive value), or backward (negative value)
<speed> in ms for each step.

mot 0

Set idle. No current.

Output summary:

mot0=	mot1=	MT1 xc5	MT2 xc6	MT3 xc7	MT4 xc8
-1, or z		open	open		
0		GND	GND		
1		Vin	GND		
2		GND	Vin		
	-1, or z			open	open
	0			GND	GND
	1			Vin	GND
	2			GND	Vin

2.3 Sample rates

Sample rates can be from 1 millisecond to 24 hours.

The ADC sample rate is higher than the storage sample rate.

GigaLog stores the average sum of the last period in the disk file.

(GigaLog can also store the minimum and maximum values of the last period)

The output of the ADC is signed 24 bits.

The average sum will be calculated in 32 bit float.

Storage sample rate on disk Examples	Proposed ADC sample rate for average sum
1 ms	1 ms
100 ms	1 ms
1 s	5 ms
1 m	300 ms
1 h	18 s
24 h	432 s

Speed considerations

GigaLog is designed to store all analogue inputs at 100 Hz (10 ms).

If using higher rates or storing data from Rs232 at the same time, you must verify that the board and the memory card are able to follow.

2.4 From raw values to Engineering values, Calibration, Output format

The raw value is at the output of the ADC.. The real value is the value stored on the disk. You can specify for each analogue input an expression to calculate a real value from the raw value.

This expression is also used, to calibrate an input.

For example

$$a3=a*7+200,2$$

For a raw value of 0, the real value on the disk is $0*7+200/100 = 2.00$. For a raw value of 35, the value is $35*7+200/100= 4.45$

The multiplier, and the term are 32 bit float.

The expression also gives the output format

Expression	Raw value	Real value
a	24002	24002
a*0.002,0	24002	48
a*2,3	24002	48.002

GigaTerm software helps you to find the expression.

2.5 Alarm

Each input can be configured to trigger an alarm.

You may configure for each input a minimum threshold, and a maximum threshold(configuration a)

You may set for all inputs an alarm filter delay et0, and an alarm redo delay et1(configuration et)

The alarm is calculated on data directly from the inputs, not on the data stored on the memory card.

There may thus be slight differences.

Analogue inputs

Compare the raw value to the thresholds.

When the raw value is less than the min threshold, trigger alarm.

When the max threshold is not equal 0, and the raw value is greater than the max threshold, trigger alarm.

Digital inputs

Compare the input to the thresholds.

When the min threshold is not equal 0, and the input is 0, trigger alarm.

When the max threshold is not equal 0, and the input is 1, trigger alarm.

Counters

Compare the counter to the thresholds.

When the max threshold is not equal 0, and the counter is greater than the max threshold, trigger alarm.

When storing the counter to the disk, the counter is less than the min threshold, trigger alarm.

Thermocouples, Pt100

Compare temperature to the thresholds.

When the min threshold is not equal 0, and the temperature is less than the min threshold, trigger alarm.

When the max threshold is not equal 0, and the temperature is greater than the max threshold, trigger alarm.

Thermocouple temperatures are in 0.01°C (7700= 77°C) , Pt100 in 0.1°C (770= 77°C).

Using alarms:

Define macro 1 as onAlarm, macro 2 as onAlarmend.

When an alarm is active for a period of et0, macro 1 will be executed.

When the alarm does not disappear within et1, macro 1 will be executed again.

When the alarm disappears for et0, macro 2 will be executed.

Et0, and et1 are in multiples of .1 seconds, i.e. 10 represent 1 second.

Examples for macro 1

fa alarm.log %d Alarm; rI0=1	Write into a file on the memory card: date and time, Alarm; switch relay 0
sm 0603154848,Alarm %0 %1	Send SMS, using a mobile modem, send "Alarm", dump inputs a0, and a1

Example for macro 2

rI0=0	Switch relay 0
-------	----------------

3 Command mode, Configuration

3.1 Enter Configuration, Command mode

Use the PC utility GigaTerm to communicate with the board.

You may use the USB port, or any serial port.


The USB port is always in command mode.

You can not use a serial port RS232, or Rs485, that has been configured to store incoming data to the memory card.

If you use a RS232 port, you need a null modem cable (female 2-3, 3-2, 5-5 female) to connect the board to the PC.

Select a COM port at 115200 baud, click on Open to open the port.

To work remote with a modem, see chapter GigaTerm.

Click on  to read the current configuration from the board. The board will now answer:

GigaLog M v2101

...

V2101 Indicates the version of the board software: January 2021.

3.2 Syntax

A command line is limited to 80 characters, it ends with a <cr> carriage return or a <lf> line feed.

Command: <cmd>[<ch>[<ch2>]][=] <par>*

Parameters of a command are separated by ' ', or by ',' (space, or comma).

Several commands in a line are separated by ';' (semicolon)

Commands start with a character, except: '#'

A parameter -<small character>* represents a flag

Exceptions: The following commands use ' ',';' (space, comma, semicolon) as argument

- m<ch>= macro define
- # rq slave command
- bg execute in background
- at send command to modem

Exceptions: The following commands use ' ','' (space, comma) as argument

- ec echo
- fa file add
- fd file upload (intern)
- at send command to modem
- sm send sms
- lcd lcd message

3.3 Files on the disk

diary.log	Log file Use the 'fa' command to append a line to this file.
onreset.txt	if this file exist, the firmware will execute commands in this file after reset, by 'ex -sbe onreset.txt' The file may include a command to delete itself
in.txt	log of analogue input data, see commands 'an', and 'ad'
logrs<n>.txt	log of data from the rs<n> port. See command 'rs'

3.4 Configuration, Commands

h *|all

h hard|conf|com|adc|disk|flow|lp|debug|remote|rq|lcd

h <command>|<pattern>*

Display help all, on chapters, or on commands

d [-b][-c][-a][-l][-q][-m]

Display configuration. Without a flag: all information.

- b basic information
- c communication, counters
- a analogue inputs
- l lcd
- q rq modules
- m macros

z

Zero: get back factory standard, clears all configuration values.

zc

Zero counters bm..

st

Stop log mode. This command is like switching the STOP switch to Stop.

go

Start log mode. This command is like switching the STOP switch to Go.

3.5 Configuration Rs232 data

rs[0|1|..|7]=[-|c|d|m|i|s],<baud>,<filename>,<buffer size>,[7|8|e|o|r|2]

Configuration of serial ports.

c : Port in command mode.

m: Ignore command line, that look like messages from a modem

i: Ignore all errors in command lines

d: Log incoming Rs232 data to a file on the memory card.

s: Remote Acquisition Slave: Only accepts # commands.

<baud> Baud rate.

<filename> File name for mode 'd'. The name may include special characters, see chapter file names.

<buffer size> for incoming data

7= 7 bits; 8= 8 bits, o=odd parity, e=even parity, 2=2 stop. Default: 8 bit, no parity 1 stop. r=raw mode

rs[ch[..ch2]]

Show rs states, errors

fs=<frame>

Frame start. See frame.

fe=<frame>

Frame end. See frame.

3.6 Configuration Analogue Inputs

an= <path>[,h]

Name of the file on the disk for analogue data.

The name is limited to 24 characters. It may include directories: [/][<dir>]*<name>

See chapter file names.

,h: Insert header-line at the beginning of the data file: # as field, input names, ae field.

an+

Increment file name. g.adc → g1.adc; g1.adc → g2.adc; g99.adc → g100.adc

Be careful, when using this command in a macro. Each command changes the configuration, and thereby writes into the flash memory. Writing into flash memory is limited to 10 000 cycles.

When the file reaches a size of 2 Gb, the name will be incremented automatically.

ad= <storage rate>[,<ADC rate>]

Analogue data storage, and ADC sample rate. Sample rate will be ignored in low power mode.

When missing, the ADC sample rate will be calculated from the storage rate

- 0 no samples
- 1ms .. in milliseconds
- 1s ... in seconds
- 1m ... in minutes,
- 1h ... in hours

as= <frame>

Frame start each second, minute, or hour in the analogue data file. See Frame.

am= <frame>

Frame start each millisecond, See Frame.

ae= <frame>

Frame end. See Frame.

a<ch>[-<ch2>]=[-<d|c|a|k|p|z|<term>][*<m>][+<p>][,<c>][<<min>][><max>][,n=<name>][,d=<diff>][,p=<reference>][l=<lcd>][m=[a][m][M]]

Configuration of an input, or several inputs (a4-7=..)

- not used, input will be sampled
- d as digital input
- c as counter
- a as analogue input
- k thermocouple; k0 type K, k1 type J, k9 type K without cold junction compensation.
- p Pt100
- z No sampling

Values *m+p,c: see analogue inputs calculation. Determines also the output format.

<term> Number for gigaterm to select input option

<min> and <max> for alarm. See alarm.

<name> for LCD

<diff> 0 single ended, 1 differential, only on inputs 0,2,4,...,14.

<reference> Reference input number. Subtract the reference input from the input (real values)

<lcd> Alpha LCD: l=ntttt. tttt=1..9999: time to display channel on LCD in seconds. 999: do not display.

Graphic LCD: l=nswwc: s=style (0..2), w=width(0..9) cc=color (00..15)

Counters: n= 1 display current count in sample period (see ad=)

n=2: display count of last period

n=0: display max of 1 and 2

m=[a][m][M]: Store average(a), minimum(m), maximum(M) value. Stores up to 3 values per input.

a<ch>=v[*<m>][+<p>][,<c>][,c=<expr>][,n=<name>][l=<lcd>]

Declaration of a virtual input, calculated from other inputs by expression.

Expression is a sequence of up to 5 elements, connected by +, -, *, or /.

Elements are inputs or small constants (0..99).

Example: c=a0-a1*a2

Calculation will be done on the calculated real values of the inputs, strictly from left to right.

The value of a counter is the count of the last full period, as defined by ad.

A virtual input can not trigger an alarm.

See chapter: Calculation from the analogue input raw value

a<ch>=vc[*<m>][+<p>][,<c>][,<c>=<expr>][,<n>=<name>][!=<lcd>]

Declaration of a long time counter, calculated from other inputs.

See chapter: Calculation from the analogue input raw value

a<ch>=vp[*<m>][+<p>][,<c>][,<n>=<name>][!=<lcd>]

Declaration of a personal input. Personal firmware addition.

a[<ch>[-<ch2>]] [<n>[,<r>]]

Dump input <ch>, <ch> to <ch2>, or all analogue inputs n times, speed <r> ms.

You may stop the output by any key from the keyboard.

a0-3 100 1000 dumps channels a0 to a3 each second during 100 seconds.

ar[<ch>[-<ch2>]] [<filter>,<series>]

Calculate input resolution of one, some, or all inputs, digital filter, series of tests.

Digital filter is 10 minimum, except for samples faster than 10 ms. Series needs space in memory, should be 100.

av[<ch>]= <n>

The av command changes long time counters

av= z clears all counters

av<ch>= <n> sets a single counter.

av<ch>+= <n> increments a single counter.

ax=<adc config>,<buffer size>,<m samples>

<adc config> cbdr c=chopping, b=bias, d=delay, r=data rate. See doc ADC ADS1258

<buffer size>: Size of analogue data buffer from acquisition to write to disk.

<m samples>: Number of samples analogue inputs for display data, not for the data on the memory card.

ax

Display additional information about the analogue digital converter

et=<et0>,<et1>

Alarm filter delay et0, and alarm redo delay et1. In multiples of .1 s. See Alarm.

po=<threshold low>,<threshold high>

Power supply monitoring. Values are in mV.

When the power falls under threshold high, stop storing data to disk, synchronize the internal buffers with the disk.

When the power falls under threshold low, stop all disk activities. When power comes back over threshold high, return to normal work, only some data may be lost.

tr=[+|-|i+|i-|]<channel>,<threshold low>,<threshold high>[,<prologue>,<epilogue>][,<filter>]]

tr= Switch of trigger

Sample analogue data to disk only, when trigger is active.

+: Positive, - negative level activates. Channel is an analogue input from 0 to 15.

Case +: When the input rises over the threshold high, start sample data to disk.

When the input falls under the threshold low, stop sample to disk.

Prologue is a number of probes to be held in memory, and to be stored, when the trigger gets valid.

This number is limited by an internal buffer, see Configuration ax <buffer size>, Counters: af, av.

Epilogue gives a number of probes to be stored after the trigger got invalid.

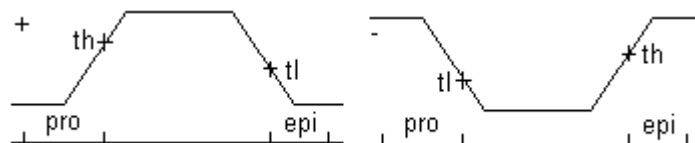
Filter is the minimum time in milliseconds a threshold must be passed.

tr i+ like tr +, but only a one shot sample, when reaching the threshold, including prologue, and epilogue.

tr i- like tr -, tr i on both edges.

tr -t sends a one shot trigger, when the trigger is defined and inactive. Samples prologue and epilogue.

Thresholds are in millivolt.



Examples:

tr=+15,1000,1000 Sample data only, when a15 is over 1V.

tr=+15,1000,1000,20,30,100 Same. Record also the last 20 samples before, and 30 samples after the trigger is activ. Ignore all spikes less than 100 ms.

tr=i+15,1000,1000,20,30,100 When a15 reaches 1V, record the last 20 samples before and the next 30 samples. Ignore all spikes less than 100 ms.
tr=i+15,1000,1000,0,0,5000 When a15 reaches 1V, wait 5s, then record 1 sample. Signals shorter than 5s will be ignored.

iadc

display internal analogue inputs of the mcu.

3.7 Configuration Disk, Files

format [<drive>][<volumename>]

Format disk

ls [<path>]

Display the files in the directory

fa <path> <text>

touch <path> <text>

If the file does not exist, create the file. Append a line containing the text to the file. The text may include spaces. A ';' indicates the end of the command.

For special characters sequences see chapter Special characters.

Other name for the command: touch

cp <path> <path>

Copy the file 1.path to 2.path

mv <path> <path>

Rename the file 1.path to 2.path

cmp <path> <path>

Compare the files

rm [-f] <path>[, <path>]

Remove the file or the files from the disk. -f: Ignore errors.

The command does not remove directories. It accepts wildcards like rm in*.

md <filename>

Make dir.

cd <path>

Change dir:

Examples:

```
cd /
md subdir
cd c:/
cd subdir
cd ..
cd subdir
cd c:/subdir
```

pwd

print current working directory

up [-b|-c] <path> [-<off>] [<start>]

cat [-b|-c] <path> [-<off>] [<start>]

upload file.

The command up gigalog.adc uploads the complete file.

Up gigalog.adc 2004:09:09 12:30:03 uploads only data stored after this date.

Up gigalog.adc -1000 starts at offset filened -1000

Up -b sends binary data in hexadecimal format.

Up -c starts reading data at the file offset uc, sends a start line to switch GigaTerm to receive data, and at the end stores the new file offset in uc. This version can be used from a macro to upload data.

Each uploaded line is preceded by a ">".

Other name for the command: cat

uc=<n>

Set the file offset for the up -c command.

fd <filename>|+<hex data>|!<crc>

File download to disk

dx=<files>,<cache buffers>,<retry>|<sd2>

<files> Number of files the program can open at a time

<cache buffers> Number of sectors in the buffer cache

<retry> 1: no read after write

<sd2> 4: 2. sd-card

A new value will be used after the next Reset.

dk

Show information about disks

3.8 File names, Circular Data Storage

File names in configuration fields, and commands (an, cat, cd, cp, cmp, fa, fd, ftp, md, mv, rm, rs, up) are paths in the file system, and may include directories and subdirectories.

A file name may start with c:/ for the sd-card on the board, and d:/ for an external disk.

Legal constructions are:

abc.adc

abc/cde

c:/abc/cde/xyz.txt

Some commands create non existing directories automatically.

Some commands accept wildcard constructions, like cat di* instead of cat diary.log.

File names may include special codes, that will be replaced, before opening the file on the disk.

sequence in file name	will be replaced by
%M	month yyyyMM
%d	date yyyyMMdd
%D	date of the month dd
%h	hour hh
%m	minute hhmm
%s	second hhmmss
%i	Replace immediately

Examples for the 17. May 2012 at 12:15

an=adc%d.adc	Data will be stored in one file per day adc20120517.adc
an=adc%d%m.adc	Data will be stored in one file per minute adc201205171215.adc
an=d%d/a%d%m.adc	Data will be stored in one file per minute, in one directory per day. directory: d20120517 file: a201205171215.adc

When the time advances, and the file name changes, the firmware closes the file and opens a new file.

The base of the substitution is the current date and time.

If the file name is followed by one of the following sequences, the time will be added to or subtracted from the current date.

+<n>s, -<n>s	add, subtract <n> seconds
+<n>s, -<n>m	add, subtract <n> minutes
+<n>s, -<n>h	add, subtract <n> hours
+<n>s, -<n>d	add, subtract <n> days

For example:

an=adc%d.adc

m3=0,24h, rm adc%d.adc-30d

will create a data file for each day.

When we are the 17. May 2012, today's data will be written to

adc20120517.adc

The macro m3 will remove each day the 30 days old file, today it will remove the file

adc20120417.adc

This allows a circular data storage.

We only keep the data of the last 30 days.

3.9 Communication

ec[0|1|..] <text>

Echo text to a serial port, USB, or graphic terminal. Ec without target channel goes back to the sender. The text may include spaces. A ';' indicates the end of the command. For special characters sequences see chapter Special characters.

lcd <mode>,<contrast>

See chapter Display

tm [-a] [-m] [-r] [0|1|..]

Enter transparent mode to a serial port, or USB.

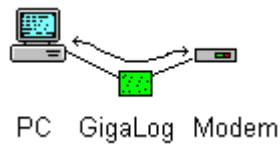
Useful to configure the modem, or a remote acquisition module.

-a Adam module protocol. add CRC to each line for remote acquisition modules.

-m Modbus ASCII protocol. add CRC to each line for remote acquisition modules.

-r raw, or binary mode.

<ESC> to quit the tm mode: wait 3s, then type <ESC>



ip, reg, ftp See chapter Remote Control

apn, mm, sm See chapter mobile phone

gps,tz See chapter GNSS

3.10 Macros, Execute commands

m<n>=<stime>,<rtime>,<text>

m<n>=<onCondition>,<text>

Declare macro <n>

Stime is the time, the macro shall be executed the first time, in seconds from 1.1.2000.

Rtime is the time, the macro when the macro shall be rescheduled after execution, in seconds.

It can also be expressed in <n>ms, <n>s, <n>m, <n>h. Do not use times faster than 100ms.

Text is the command line of the macro. The text goes to the end of the line, including ' ', and ' '.

On macros will be executed on special conditions:

<onCondition>			Example
onReset	a	When coming out of reset	m0=onReset, fa reset.txt reset %d %r Add a message in a log file about reset time and reason
onAlarm	b	Alarm condition start, or redo	m1=onAlarm, rl0=1
onAlarmend	b	Alarm condition ends	m2=onAlarmend, rl0=0
onStop	a	Switching to Stop mode	
OnGo	a	Switching to Go mode	m3=onGo,an=g%i%d%s When switching from Stop to Go mode: change the data file. Note, that some data may be logged to the old file, before opening the new file.
OnDiskerror	a	Disk error	m4=onDiskerror,lc Disk %k Display error message on the lcd

a: macros will be always be executed even in Stop mode.

b: macros will only be executed, when the board is in GO mode.

Time based macros will only be executed in GO mode.

Several macros can be executed in parallel. Macros may wait, using the wt command.

A macro will not be executed again, when it is already running.

Execution of macros does not disturb data logging or other operations.

Be careful with macros, changing the configuration: Changing the configuration will write into the Flash memory.

Writing into flash memory is limited to 10 000 cycles.

mx<n> [<time>]

Schedule macro n in time seconds for execution.

ex [-s] [-b] [-v] [r] [-e] <file>

Execute commands from the file, line by line.

-s silent

-b in background

-v verbose: display commands before execution, echo ok at the end of the file

-r remove file from the disk, after execution

-e stop execution on error, else continue

bg <commad line>

Execute the command line in background.

3.11 Low power mode

lp=<mode>,<uptime>,<options>*

<mode>	
0	None
m	Low power master board
s	Low power slave board

<options>	when coming out of sleep
q=<qrate>	Qrate, see below
mdm	switch on modem
mt1	switch on MT1 (xc5)
mt3	switch on MT3 (xc7)

You may enter several options

Uptime in ms is the wake up time from power up before executing commands.

Probes may need this time to send a good value. The board temperature probe needs 200 ms

The board enters low power mode when

- it is in go mode, and
- there is no macro running, and
- there is no activity on the modem

In power down mode, the board

- does not sample data. Storage rate is equal to sample rate (s. command ad).
- does not accept commands from any serial port

The board does not enter power down mode, or leave power down mode, when

- it has to sample, and store analogue data, see ad command storage rate, or
- there is a macro to be executed, or
- the stop switch is on

The qrate indicates the number of samples gathered in RAM, before writing the data to disk.

The qrate decreases the current in sleep mode.

Qrate= 0 or 1: store always immediately

When the internal buffer overflows, the data will be written to disk.

When switching to Stop mode, the data will be written to disk.

When switching power off without stop, you may loose up to <qrate>-1 samples.

lp

Command lp without parameters: show reasons, why the board can not go to sleep.

Remove the USB cable from the board in low power mode, to reduce consumption.

See also chapter Connecting several boards.

3.12 Configuration, Miscellaneous

bn=<name>

Set the name of the board. This name will be displayed on the LCD and in the configuration.

nm=<name>

The name of the board on the network, to identify the board.

pw=<password>

Password used by remote login.

of= <dec. point><separator>

Output format. 1. character is the decimal point in analogue data.

2. character separates inputs in the analogue data file

Default: of=._ (_ = horizontal tab)

CSV Format: of=, _ as=d, am=m,;

dt=yymmdd

Set date

ti

Reset/Start timer

ti=hhmm[ss] [dt=yymmdd]

Set time, and optional date

rtc

Read date and time from the real time clock.

The software reads date and time once after reset from the real time clock. Since the internal clock on the micro controller may vary from the real time, it may be useful, to reread the time. Executing this command may have an impact on the stored data and on the execution of macros, since the time may jump forward or backward for some seconds.

t

Display board temperature in °C, and supply voltage.

wt <n>ms|<n>s|<n>m|<n>h

Wait <n> milli seconds, seconds, minutes, or hours.

wd=0|1

Disable, enable watchdog. You must reset the board, to activate the watchdog.

The firmware must regularly retrigger the watchdog.

When this retrigger is missing, the watch dog resets the board, to avoid some software problems.

bo= 0|1

Brownout detector.

The brownout detector assures a proper reset, when the supply voltage is insufficient for normal operation and avoids unpredictable corruption.

The brownout detector is needed, when the supply voltage of the microprocessor can drop under 3V, and come back to a normal level, without passing by 0V.

dl [<file>]

Enter software download to download a new firmware over the serial port.

dl <file> will download a firmware from a file on the sd-card. Needs boot software 1402 or later.

Commands sm, ip, ftp, wf: See chapter Remote control

Commands gp, tz: See chapter GNSS

Commands rq, rqz, #nn: See chapter Remote Acquisition Modules

Commands gr*: See chapter Graphic Display

Commands xc, rl, mot see Hardware section

3.13 Conditional Instruction

if <primary> |=|=|>|=|<|<= <primary> <true commands> [else <false commands>]

When the condition is true, execute the true commands, else the false commands, if they exist.

Primary:

a<n> Analogue input, real value,
xc<n> input XC header,
t board temperature in °C
v board voltage in mV,
gs gnss speed in knots, -1 = invalid
constant.

Example: Macro each second: if a7<300 r10=1; r11=0; else r10=0; r11=1

3.14 Configuration, Frames

frame start, frame end, analogue start, analogue millisecond start, analogue end are up to 10 character long fields. Each character represents

d	timestamp yyyy:mm:dd hh:mm:ss[:uuu]
D	timestamp yyyy:mm:dd hh:mm:ss
u	timestamp mm/dd/yyyy hh:mm:ss[:uuu] compatible xel US
e	timestamp dd/mm/yyyy hh:mm:ss[:uuu] compatible xel Europe
f	timestamp dd/mm/yyyy hh:mm:ss compatible xel Europe
t	timestamp hh:mm:ss[:uuu]
m	timestamp milliseconds uuu
-	space
_	horizontal tab
n	carriage return <cr>, new line <nl>
T	board temperature,
v	board supply voltage
g	GNSS position
gs	GNSS speed in knots, -1=invalid
any other	output directly

Ex. "fs=d_" "fe=n" will precede each frame with a timestamp, and a tab, and add a <cr><nl> at its end.

Ex. "fs=dn" "fe=dn" adds a timestamp line before and after the frame.

Ex. "as=d_" "am=m:_" precedes analogue data with a full date, when the second changed, else only with the new microsecond.

This is compatible with the GigaData program.

3.15 Configuration, Special characters

Some commands like ec,fa,sm,lc,grw recognize, and replace special characters in the text.

%d,%D,%u... Date and time format: see chapter Frames, above

%g GNSS position

%gs GNSS speed in knots, -1=invalid

%t board temperature

%v supply voltage.

%n no newline at the end of the line.

%<n>[-<m>] Analogue input n [to m]

%* all declared inputs

%h<n>[-<m>] Head line: input names

%h[*] Head line: input names

%H complete input line: as field, input names, ae field.

%k Details of last disk error

%r Reset reason: power-up, button, watchdog, brownout, download, download disk, software, panic, restart. Needs boot version 1402 or newer. See command d.

%rd Reset time

%rr Reset reason + time

%u Up-time in hh:mm:ss without low-power-mode time

%_ replace space by horizontal tab,

%, replace space by ','.

3.16 Configuration, Counters

bm0,bm1,bm2	Max size, Rs232/Rs485 buffer was filled with incoming data, should not approach the total buffer size.
bv0,bv1,bv2	Rs232/Rs485 receive buffer overflow counter
af	Max size, analogue data buffer was filled with incoming data, should not approach the total buffer size.
av	Analogue data buffer overflow counter
dr	Disk compare after read or write error. Retry
de	Disk error.

3.17 Maintenance, Debug

mcu

Display information of the mcu

pio [<port>[=<value>]]

pio

pio a22 b7

Display some, or all peripheral input output pins of the mcu

Values are: o= output, i= input, 0= current value 0, 1= current value 1, up= with pull-up, dn= with pull-down,
a= b= c= d=, selected to peripheral a,b,c,d

pio a1=0

Set a pin, Values are: 0= output 0, 1= output 1, i= input, u=input with pullup

dk

Disk information

mux

Tasks of the multi-tasking kernel

m <address> [<count>]

Display memory of the mcu address space: Ram, Flash, IO. Access is byte-wise

mal

Memory allocation table

sn

Serial number

qc= quality control date, sn= serial number

db=<n>

Set debug mode. Part of the configuration in Flash memory

pr

private commands, Add-ons to the firmware

ax

Adc information

xbl

Disk block cache

xrl

Ram log

xsZ

Size of the configuration structure,

xreset

Reset the board

xexc

Provoke an exception

xwd 123

enters a loop. When the watchdog is enabled, this shall provoke the watchdog reset within 20 s.

xZ

Dump the dynamic configuration memory table

4 GigaTerm

GigaTerm is a Hyperterminal like program to communicate with the board. Use GigaTerm to communicate with the board, to keep configurations in files, to configure the board, to upload stored data. You may ask GigaTerm to write the dialogue with the board into a file GigaTerm.log (Tools, Log).

4.1 Connecting to a local board

Use the PC utility GigaTerm to communicate with the board.

You may use the USB port, or a serial port in command mode.

The USB port is always in command mode.

You can not use a RS232 port, that has been configured to store incoming data to the memory card.

If you use a RS232 port, you need a null modem cable (female 2-3, 3-2, 5-5 female) to connect the board to the PC.

Select a COM port at 115200 baud, click on Open to open the port.

Click on  to read the current configuration from the board.

USB

When connecting the board the first time to an USB port of your PC, Windows detects a new peripheral. The board uses a standard Windows driver. Nevertheless Windows may ask for the driver. Direct Windows to the usb driver subdirectory in the installed software, like "usbdriver". Windows treats the USB port as a COM port and gives it a free number.

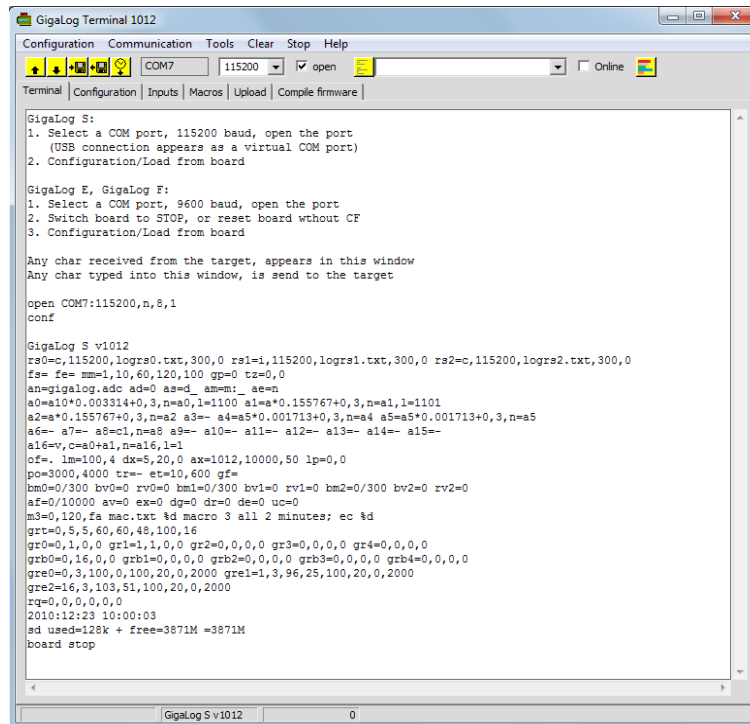
4.2 Connecting to a remote board by TCP/IP

Select TCP/IP as Port. Enter the IP number and the port as address: "123.45.67.89 1024"

Open the port. It may take several seconds to establish a connection.

To disconnect, click on Online, to disconnect the line, and then on Open, to close the port.

4.3 GigaTerm as a terminal



GigaTerm works as a terminal program.

Any character typed into this window, will be send to the board.


Any character received from the board, will be displayed in this window.

You may ask GigaTerm to write the dialogue with the board into a file GigaTerm.log (Tools, Log).



WARNING:

Do not edit existing commands in this window ! You have to type in a command to send it to the board.

4.4 Handling configurations



After having changed the configuration, always type the 'd' command or click on  to verify the configuration in the board.


Copy the configuration of the board to a file on the PC disk:


Click  to get the current configuration of the board. Then click immediately on  to save the configuration into a file.


Copy the configuration from a file to the board:

Erase the existing configuration on the board: Select Configuration → Erase to Factory Standard

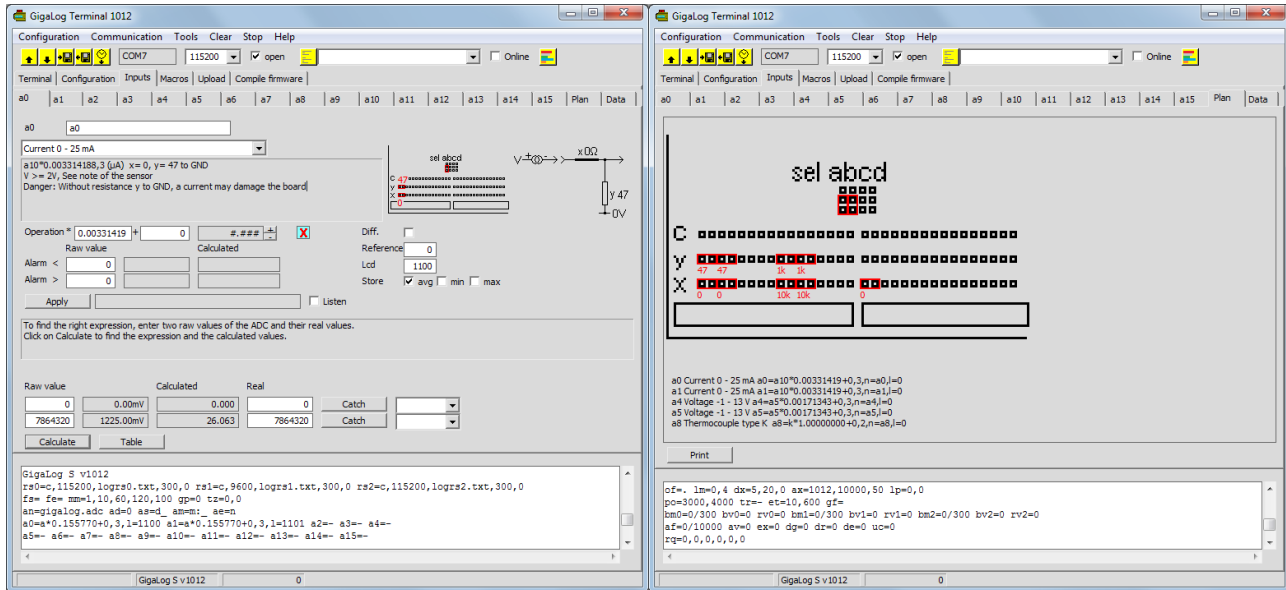
Click  to read the configuration from a file. Then click immediately on  to send the configuration to the board.

Click on  to read the current configuration from the board for verification.

When you are in a doubt, always click on  !

The configuration in GigaTerm and the configuration in GigaLog are not automatically synchronised!
When you are in a doubt, always click on  to get the current configuration of the board.

4.5 Inputs



Click on the tab Inputs, and then on the tab of an input

Fields in this tab will be shown according to the target board, firmware version, and the selected input type.

Select the input type. You see the jumpers and components to put on the board at the right.

Click on Plan to see the components for all inputs.

Calculation: Calculates the real value. See below.

Alarm <: if non zero, if the value is lower than this limit, an alarm occurs.

Alarm >: if not zero, if the value is higher than this limit, an alarm occurs.

Diff.: Single ended input or differential. Default: single ended.

Reference: reference input. Default: none.

LCD: display mode for LCD or graphic LCD. See configuration, analogue inputs, a= option I=.

Store: average, minimal, maximal value. Default: only average value.

Apply: Send the configuration of this input to the board.

Listen: Check this box, to request each second the input state from the board.

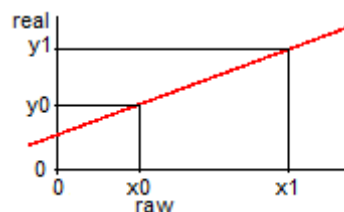
The real value, that will be logged into the disk file, can be calculated from the raw value at the output of the ADC by a linear equation.

To find this linear equation, you have to enter two raw values of the ADC (X0, and X1) and their corresponding real values (Y0, and Y1).

Catch: Copy a listened line or directly from the input into this line.

Calculate: Calculate the operands from the given raw and real values.

Table: Writes a table of raw, and real values into a file, open this file, to verify the expression.

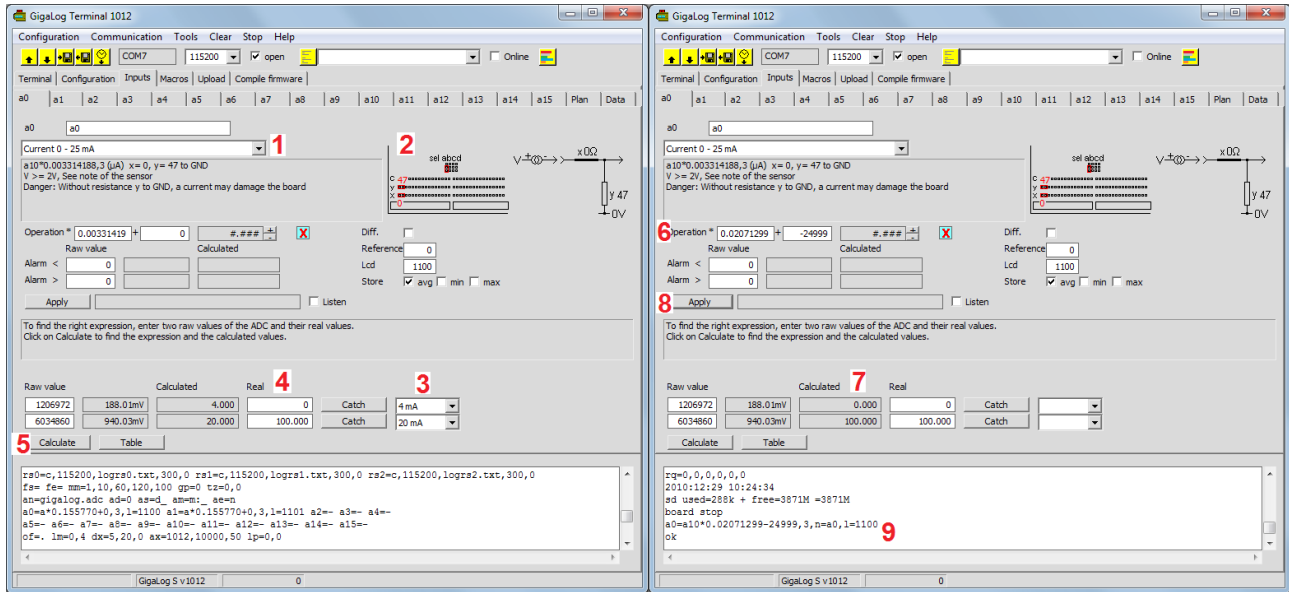


A linear equation

The configuration in Gigaterm and the configuration in GigaLog are not automatically synchronised! Click on "Apply" to send the new configuration to the board.

When you are in a doubt, always click on  to get the current configuration of the board.

4.6 From raw values to Engineering values, Calibration, Output format



Step by step, how to configure the raw to real expression

Example a pressure gauge: Output current 4 to 20 mA; 4mA = 0 bar, 20 mA = 100 bar.

1. Select the input type, in the example: Current 0 to 25 mA
 2. Set resistors and jumpers on the board, BEFORE connecting the external signal to the board
- You need two raw values and their correspondent real values to find the linear equation.

You might either

- Enter raw values by hand in the fields Raw values
- Select a value from some proposed tensions or currents, to get the corresponding raw value.
- Apply some pressure, and thereby apply an input voltage, catch these values. This method is directly based on the inputs, the most exact, and also used, to calibrate an input. It is important to catch two raw values, that are not close to each other; in this example two pressures of 0 and 80 bar are better than 0 and 2 bar.

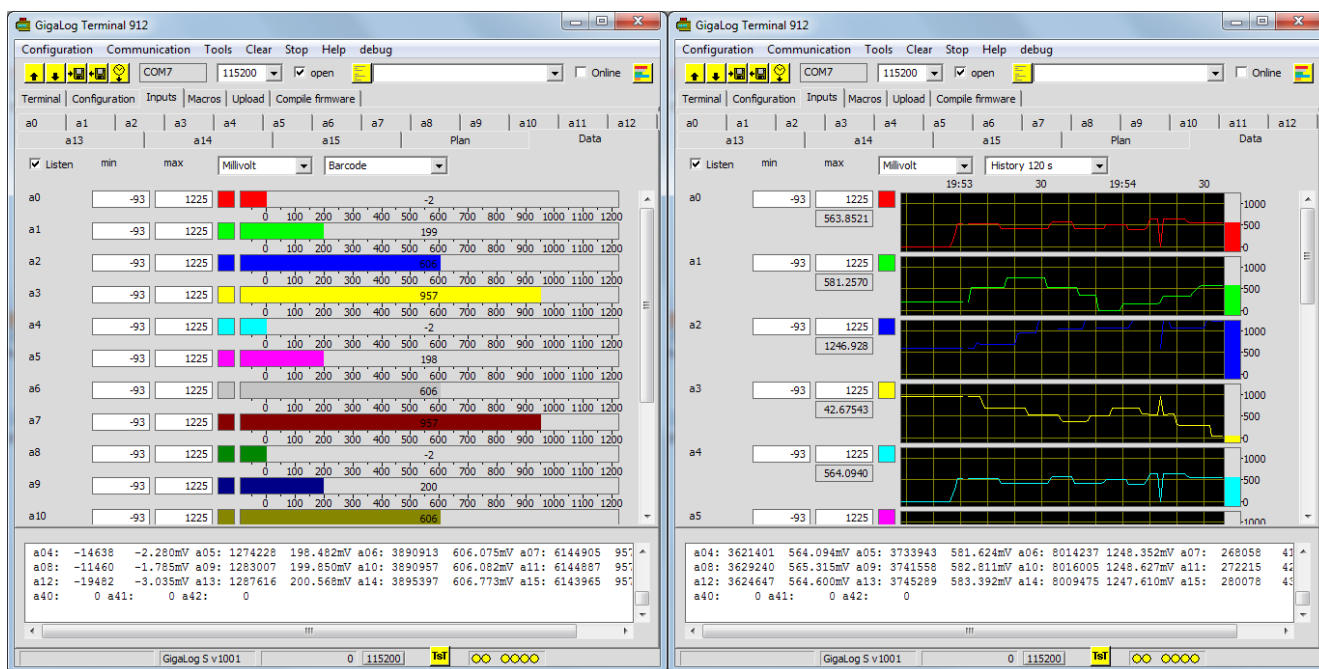
Here we will take the values from the pressure gauge specification.

3. Select 4mA in the first line, to have a raw value.
4. Enter 0 (0 bar) as real value.
3. Select 20mA in the second line
4. Enter 100.000 (100 bar) as real value. The value also describes the output format. A value of 100 will also work.
5. Click on Calculate, to calculate the linear equation.
6. The new operation
7. Verify here, that the new calculated values correspond to the wanted real values.
8. Click on Apply to send the new configuration to the board.
9. The board receives the configuration and responds with OK.

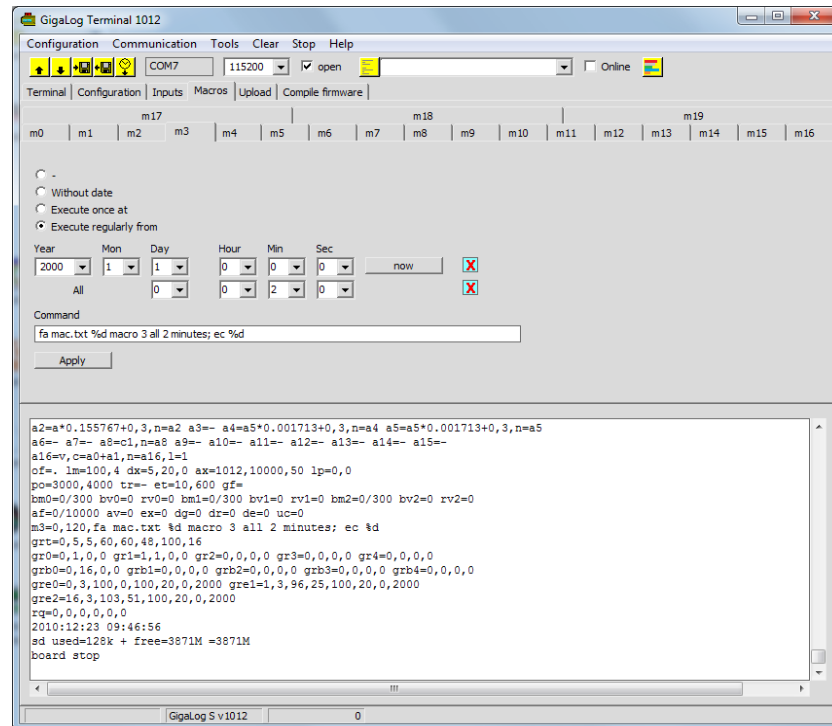
Calibrating a thermocouple: See application notes thermocouple.

Calibrating a Pt100: See application notes Pt100.

4.7 Display current values of inputs



4.8 Macros



A macro is a sequence of commands, separated by ';' like a command line, entered from the terminal.

A macro can be executed regularly, or under certain conditions. You may load up to 20 macros into the EEPROM of the board.

A macro can

- Write a message to the LCD display
- Write a message into a file on the disk
- Write a message to the rs232 port
- Call a center, and send a message, using a modem.
- Send an SMS, using a mobile modem.
- Toggle an output on the XC header, relay, power switch..
- Stop sampling, change the configuration, and more

You may define a macro

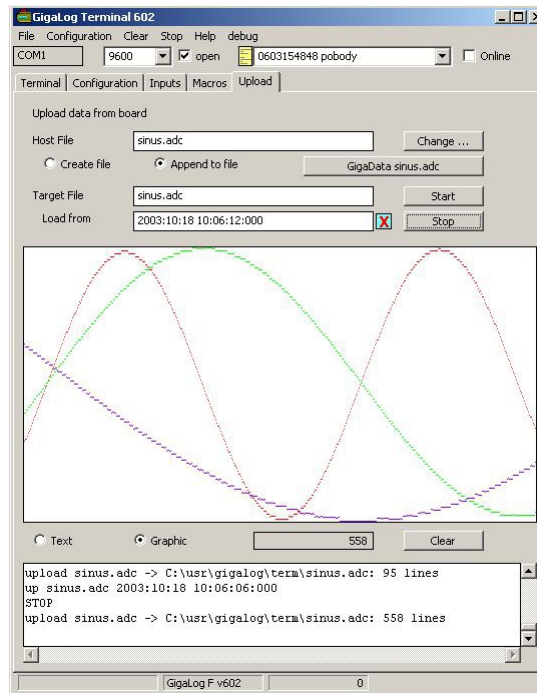
- Without any date
- To be executed once at a certain time.
- To be executed regularly.
- Starting at a certain time, to be executed regularly.
- Starting on a certain condition (on)

For more explanations and examples, see chapter [Configuration → Macros](#).

The configuration in Gigaterm and the configuration in GigaLog are not automatically synchronised!
Click on "Apply" to send the new configuration to the board.

When you are in a doubt, always click on  to get the current configuration of the board.

4.9 Upload



GigaTerm can be used to upload files from the board into a file on the PC: Click on tab Upload.

To save a complete file, click on "Create File", and clear the line "Load From".

This creates a new file on the PC, or overwrites an existing file, and uploads the complete file from the target board to the PC.

When opening an existing file on the PC, GigaTerm will search the last date in the file.

To upload only new data, use "Append to File", and "Load from" as last date from the file.

This will only look for new data, not yet uploaded, and append them to the existing file on the PC.

When the file is long, the seek of new data and the data transfer can be long.

Click on STOP to stop the transfer.

The board signals EOF at the end of the file.

When GigaTerm receives a line "up <filename>" from a board, it sets the host file, the target file, selects append to file, and starts uploading the file.

4.10 Firmware Download

GigaTerm can be used to download new firmware into the GigaLog board.

You may load the newest firmware from our Internet server.

The board must be in configuration mode. Select Tools → Firmware Download.

During this operation: Remove modems, Gnss, or any devices, that can send messages to the board !

Firmware download can also be down over the air:

The board has to get the new firmware, using a ftpget.

Then the board has to execute the 'dl <file>' command to load, and execute the firmware.

4.11 First Help

When the firmware does no longer respond, or your configuration does no longer allow you to work.

Place a wire from GND to the boot pad on the board beside the battery.

Reset the board. The board writes "Download S7" on the port RS0. When you are on USB, enter "dl".

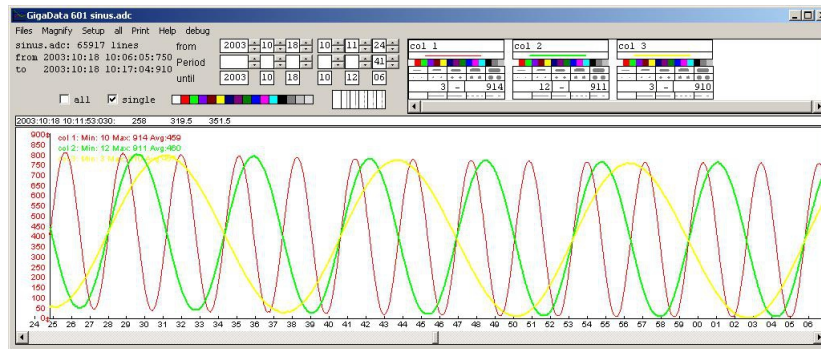
Enter "z" to clear the configuration.

Download a new firmware.

Enter "go" to start the application.

During this operation: Remove modems, Gnss, or any devices, that can send messages to the board !

5 GigaData



GigaData displays graphically the recorded data. The program has a print setup to fit the data on a page, a print preview, and the print of a data page.

Format of the recorded data

One ASCII line for each record: <date>[t<data>]*

For example **2002:11:09 11:56:30:310 1024 378 567**

date may be [<year>:<month>:<day>]<hour>:<minute>[:<second>[:<millisecond>]]

or just <millisecond>: following a line with full date

The day <year>:<month>:<day> may be replaced by <day>/<month>/<year> or <month>/<day>/<year>

Menus

File Menu:

Open Opens a data file.

See ASCII data Open the data file using a text editor to see the original recorded data in ASCII.

Erase data Erases all data in the file.

Erase <filename>.ini. Erases all stored display information about this file: column-names, display styles, y-axis.

Magnify

The program works on a buffer of samples covering the whole period. When the file is long, and the chosen period small, this button loads data from the file for a better resolution. The program does this operation automatically, when there is few data on the screen. Printing is always done directly from the data of the file.

Setup: Display or not the data setup.

Print Print the page of data.

Select a period to display

The easiest way: Click and drag the mouse on the x axis in the preview window.

Or select the start of a period. The less important time slots are set to zero. For instance, selecting a day sets hours and minutes to zero,

Then select the length of a period.

You can move a period using the horizontal scrollbar of the preview window.

The "All" checkbox indicates that the screen covers the total period. Click on it, to get back the initial period.

You may also use the mouse wheel to zoom data.

Display of data

Data is displayed per column of data in the original file.


Give a name to each data column.


Select a color for the display. A column with a white colour is not displayed.

Select a style of output: line, or points, pen size.

Select the minimum and the maximum value to be displayed on the y axis.

You may also select a scale on displacing the \updownarrow arrows on the y axis.

The symbol  indicates: Reading data from file.

The symbol  indicates: Printing active.

6 Remote Communication

You may use a mobile network or WiFi, to access the Internet.

You can use FTP to send data to, and receive data from an internet server.

You may access the board from outside over WiFi, when a HTTP server is running on the board.

ip [-v]

display ip information

mm [-v]

Display mobile phone information

wf [-v]

Display WiFi information

Applications	FTP	Registration	HTTP server
Internet	IP		
Link	Mobile network		WiFi

6.1 Mobile network

Quick guide: How to send your data to an Internet server

Direct your Internet browser to the URL.

controlord.fr/db24/registration.html

Enter your mail address, login, specify a home directory name, and register.

Create a data log file in.txt and log some data lines. Verify the file	an=in.txt; ad=1s; go; wt 10s ; stop ls in.txt 373 ...
Get a SIM card with. Remove the PIN code, if it exists, with a telephone. Put the card into the SIM card holder. Plug the modem board on the GigaLog board Connect the mobile, and the GNSS antennas.	
Configure the serial port. If you want Gnss: start the Gnss server	rs=1,i,115200 gps= 51 rs=7,i,115200 gps= 57
Get the APN information from your provider.	apn= <apn name>,<apn user>,<apn passwd>
Start the modem server	mmstart -v
Wait some seconds, and verify, that the modem works correctly. The third line must not indicate an error.	mm sim: ready, antenna: 12,0, network: 0,1, apn: oran.fr IP: 0.0.0.0, port: 1028, mdm: Telit, GL865-DUAL
If there is any error in the third line, verify the SIM card, antenna, the APN information.	
Enter the information for the controlord.fr ftp server, login, and password. Replace MYDIR by the home directory from your registration above.	ftp=ftp.controlord.fr,controlo-db2,db24,MYDIR
If you need Gnss: Stick the antenna to a window. Verify, that you have a Gnss information If not, wait two minutes.	gps gps=4310.3038,N,00602.1816,E 2013:04:20
Send the data file in.txt. Wait until the command is completed	ftp in.txt cmd queued: ftp in.txt ftp finished ok: ftp in.txt. ...
If the command does not finish correctly, set mm=2 , to have a detailed log. Redo the command. Display diary.log (cat diary.log).	

Now direct your Internet browser to the URL

controlord.fr/db24

Enter your login and your password from the registration, and login

Select in.txt, to display your data.

What next ?

Put the ftp into a macro and send new data regularly, e.g.

macro each hour: **ftp -cdt in.txt**

Even more ?

Get the source code of the server, and create your own server.

Connecting, Configuration of the modem, modem server

Gigalog M: The modem is on the board, connected to RS7.
Green led NET:

off	modem off, or not ready
on	searching network
flashing 200ms	Registered 4G
flashing 800ms	Registered 2G/3G

Gigalog S: Connect the modem to the port RS1.

Put a SIM card into the modem.

If the card is protected by a PIN code, use a telephone, to remove this code.

apn=<apn name>,<apn user>,<apn password>

You need a license of your mobile provider for the gateway to the Internet. This will also give you: APN Name, APN Userid, APN Password. You may also find these parameters in the Internet.

mm [-v]

Show the state of the modem

mm

sim: READY, antenna: 13,0, network: 0,1, context: 1,"IP","internet-entreprise","",0,0,
IP: 10.100.29.0, port: 1028, man: Telit, mod: GE865-DUAL, fw: 10.00.155

Sim, antenna, and network give the state of the network.

There will only be a valid IP number after a command has established a communication.

In case of a problem, the last line indicates a diagnostic.

mm=<log>,<timeout cmd>,<timeout connect>,<timeout disconnect>,<timeout ftp command>

Modem modus. <log>=<logStdout>+<logFile>

<logFile> = 1: log to file diary.log on the memory card.

<logFile> = 2: log to file + more details

<logFile> = 3: log to file + more details: data

<logFile> = 4: log to file + modem dialogue

<logFile> = 7: log to file only modem dialogue

<logStdout> = 10: log to Stdout

<logStdout> = 20: log to Stdout + more details

<logStdout> = 30: log to Stdout + more details: data

<logStdout> = 40: log to Stdout + modem dialogue

<logStdout> = 70: log to Stdout only modem dialogue

mmstart [-v] [v=<verbose>] [<tmdm> <tnetwork> <tall>]

Start the modem server. Establish power supply to the modem.

<tmdm> timeout for modem ok

<tnetwork> timeout for network ok

<tall> timeout all

if modem is not ok after <tmdm> seconds, or

if the modem did not find a network after <tnetwork> seconds, or

if the modem is still on after <tall> seconds,

switch off the modem, and kill the command queue

if mm>=1: enter error message in logfile diary.log

typical macro in low power mode:

mmstart 20,40,60; ftp -cdt in.txt

mmstop

Stop the modem server. Switch off the modem

If the modem server is active, the command will be scheduled, to be executed when all commands are finished.

mmq

Display modem server command queue. Debug only

at<cmd>

Send command directly to the modem on rs1, and show result until "OK", "ERROR", or for 10 seconds.
eg

```
ati
000702046
OK
```

Sending, receiving SMS

sm -v <Number>,<Text>

Send an SMS

For special characters sequences see chapter Configuration Special characters.

The board may receive a command by SMS. For example:

```
fa sms.txt Hello
```

6.2 FTP

You need Wifi or a mobile network running.

ftp=<address>,<userid>,<password>,<dir>[<mode>]

FTP parameters.

address can be in the form 123.45.67.89, or ftp.myserver.com

dir can be in the form dir or dir/dir/dir ...

mode= 0 (active, default) =1 passive

To send data to the controlord.fr server: ftp=ftp.controlord.fr,controlo-db2,db24,<mydir>

ftpop[en] [-s] [<infoFile>]

Open a connection as FTP client to the server.

Store the IP number and GNSS position in <infoFile> on the server.

This command is not mandatory for the following commands.

ftp [-c|-d|-t|-s]* <filename> [<serverfilename>]

Open a connection as FTP client to a server. Change the directory on the server.

Upload a file to the server.

<serverfilename> is the name of the file on the server. If this name is omitted, the command uses the local name.

Ftp cannot append data to a file. An existing file will be overridden.

-c starts reading data at the file offset uc, stores at the end the new file offset in uc.

-d: Append date to the Serverfilename.

-t: Append time to the Serverfilename.

-s silent, else verbose

e.g. ftp -cdt in.txt sends only the newest data since the last transfer, and creates a unique file name on the server.

The internet server will later on append this file to in.txt .

ftpget [-a|-s]* <serverfilename> [<filename>]

Open a connection as FTP client to the server.

Change the directory on the server.

Download the file to the local memory card.

If the name of the local file <filename> is omitted, use the server file name.

-a append to local file, else erase existing file

-s silent, else verbose

There may be an additional empty line at the beginning and at the end of the file.

Restrictions: The file must only contain ASCII data.

A line must not start with "no carrier"

Lines shall not be longer than 100 characters.

In case of receiver overflow, there will be an error msg at the end of the file, starting with #ERROR

ftpinfo [-s] <infoFile>

Open a connection as FTP client to the server.

Store the IP number and GNSS position in <infoFile> on the server.

ftpmx <n>

Schedule macro <n> for execution.

ftpclose [-s]

Close the FTP connection.

This command is not mandatory

The firmware queues these commands for later execution

The firmware sends a message, when it has completed the command.

6.3 GNSS / GPS

gps=<mode><port>

<mode>= 10 Gnss active
 +20 Set real time clock on the board from GNSS time
 +40 Gnss in mobile modem

<port>= 0=rs0, 1=rs1, ...

Gigalog S with modem: gps=51

Gigalog M with modem: gps=57

gps

Show position and time

gpsgo

If GNSS information is not yet available, wait for it. This may take minutes.

Read Gnss from module, and show it.

gpsti

Set real time clock on the board from GNSS time;

tz=<tz offset>,<daylight saving additional offset>

The variable tz is only used, when setting the real time clock from the GNSS time.

The first value is the time difference in hours to GMT. If this value is > 12, it is in minutes.

The second value is the additional offset for summer time.

Changing to summertime is on the morning of the last Sunday in March, to wintertime on the morning of the last Sunday in October.

Example for Central European Time (Paris) tz=1,1. Or tz=60,60.

Setting the real time clock from GNSS

The GNSS sets the real time clock only

- After a reset of the board, or
- At 1:20 am, or
- After gpsti

The setting will only be done, after having received three consecutive frames consistent time information.

The GNSS protocol NMEA displays a position as latitude,N/S,longitude,E/W. Latitude and longitude have the form ddmm.mmmm with dd=degrees, mm.mmmm minutes. Google uses the form dd.dddd

Show GNSS position: Command gps or ec %g

Write GNSS position into a file: fa <file> %g

Add GNSS position to the analogue data file: ae=_gn

6.4 Wifi

You need the optional WBX module for WIFI, and Bluetooth.
Set wf, and ip, before starting the wifi server

wf=<ssid>, <password>

wifi identification of the local wifi network

ip=<ip>,<ip mask>,<ip gateway>,<dns1>,<dns2>

When you leave these addresses at 0, the router will attribute them.

example: ip=192.168.1.123, 255,255,255,0, 192.168.1.1

wfstart [r][v=n][vx=n]

Start the wifi and Bluetooth server

r when the wifi is connected, start registration

v=<n> verbose

vx=<n> verbose of the wifi module

wfstop

Stop the wifi server

wf [v][r][v=n][vx=n]

display the state of the wifi server, and the local IP number.

v: verbose

r, v=, vx=: see wfstart

wfdownload <file>

Load new firmware into the wbx module

6.5 Registration

You need Wifi or a mobile network running.

nm=<hostname>

The hostname is the name of the board. It is used in registration, to access the board from outside HTTP server

pw=<password>

The password is used by the login of the HTTP server, and by Bluetooth

reg

Register the board to the controlord data base.

This command also reads the current time from the server, and sets the real time clock on the board.

You can see on <http://www.controlord.fr/tgt.php?xall> a table of registered targets.

6.6 Http Server

You need Wifi running on the board.

Mobile network is only possible, when the SIM card has a public IP address. This is quite unusual.

Set the hostname, and the password before.

nm=<hostname>

pw=<password>

The Http server looks for the pages on the disk in the directory /www

There are some builtin pages, that are used, when the server can not find them in the directory:

index.html	home page
favicon.ico	icon
404.html	not found page
data_server.php	page sends current state of the board

Access to the http server on the local network:

Use your browser, and the local IP number. The wf command shows the local IP number.

In the browser, enter as url: <http://192.168.1.123> (example)

Access to the http server from the Internet:

Your local network is connected to the Internet by a router.

The router has only one public IP number on the internet, that has to be shared by all local devices.

Set the NAT/PAT of the router, so that a request to port 80 (standard http port) will be passed to the board.

Reduce the firewall of the router, so that a request can pass.

If you know your IP number on the network, you can now access the board from the Internet.

There are many Internet sites, that show your IP number on the network.

Use a browser outside your local network, and enter the url

`http://<your IP>`

This may not work from the local network.

When using a mobile phone, assure that it does not use the local network over wifi.

The IP number of your router normally is not fix, and may change.

Use the register command, to register the board to the Controlord name server.

Use a browser outside your local network, and enter the url

`www.controlord.fr/tgt.php/<hostname>`

This may not work from the local network.

When using a mobile phone, assure that it does not use the local network over wifi.

6.7 Bluetooth

You need the optional WBX module for WIFI, and Bluetooth.

Set the hostname, and the password before.

`nm=<hostname>`

`pw=<password>`

The Android app gidgeterm allows, to connect to the board

It includes a command line interface.

And it includes an emulation of the graphical display.

7 Connecting several boards

You may add more inputs to the GigaLog data logger over serial lines.

Use the RS2 or the RS5 port of the GigaLog board, to create one or several RS485 networks.

The GigaLog is the master of these networks.

You may connect up to 200 slaves to one GigaLog.

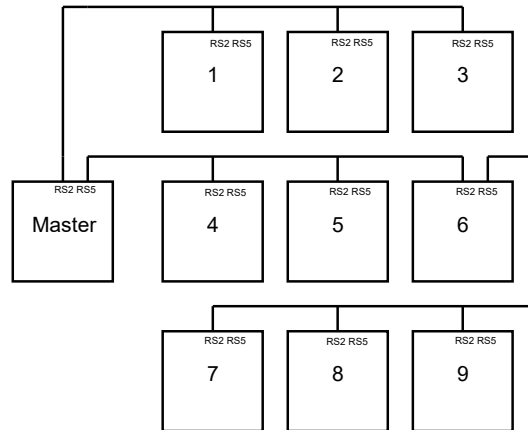
GigaLog handles up to 300 inputs, a0 to a15 are on the board, a16 to a299 may be remote.

Slaves may be:

- Other GigaLog boards
- Remote Acquisition modules. These Modules are frequently used in industries, and they come from several suppliers. Advantech® ADAM-4000, Adlink® NμDAM-6000 ICP® I-7000, eDAM-8000, etc.
- Other modules, using Modbus in Ascii mode.
- Other protocols can be added

You may configure for all inputs the expression to calculate the engineering value from the raw value before storing it to the memory card.

All inputs can be displayed on the display.



A slave board may listen to the master on port RS2, or RS5, or both ports.

The example shows one master board with 9 slave boards.

Slaves 1, 2, and 3 are connected by one RS485 bus to the RS2 port of the master.

Slaves 4, 5, and 6 are connected by another RS485 bus to the RS5 port of the master.

Slave 6 also works as a repeater.

Slaves 7,8,9 are connected to the port RS5 of slave 6.

From the master point of view, slaves 4,5,6, and 7,8,9 are on the same bus.

7.1 Gigalog Master with two GigaLog Slaves

Gigalog M: You may connect one or several slave boards to a master board by using a 6-pin flat cable. You may use the flat cable also, to supply power to the slave boards:
On the master board connect VRS2(5) to VIN, or to MT3 on JRS2RS5
On the slave boards connect VRS2(5) to VIN on JRS2RS5

Slave 1: rql=1,7 rs2=s,115200 a0-3=a*0.5208,3 Slave 2: rql=2,7 rs2=s,115200 a0-3=a*0.5208,3	Start with the Slaves. Set the slave id on the network and the protocol. The board has to accept commands from the network. Declare all used inputs. Slave boards do not need a memory card. They do not have to store data. Storage rate ad can be 0.
Master: rs2=-,115200 rq0=2,1,0,16,4 rq1=2,2,0,20,4 a16-23=a d -aq	Master. For each remote unit: rs-port, slave id, protocol, inputs Declare the inputs, to log on the memory card Verify the configuration
tm -a 2 #01 Answer of Slave 1 #02 Answer of Slave 2 <ESC>	Connect now all three boards by the RS2 ports Rs485 network. You may use a 6pin flat cable. Before starting the server, use the tm command to verify the connection to the slaves.
rq=1 rq a	Start the server. Display the state. Verify, that the slaves answer correctly. Display local and remote inputs

Log data

st	Stop log
rm in.txt	Erase file
ad=1s	1 sample per second.
go	Start log.
	Wait some seconds
st	Stop log
cat in.txt	Display stored data.

You may replace "a16=a" by "a16=a*1000,3" to select another output format.

7.2 Gigalog Master with a Remote Acquisition Module

Connect the module to RS2.

Put module to INIT mode Switch power on	In INIT mode, module address is 00, 9600 baud.
rq=0	Stop Remote Acquisition server
rs2=-,9600	RS2 9600 baud
tm 2	Start transparent Mode to RS2
\$002	Send to module, read configuration
!AATTCCFF	Answer of the module
%00NNTTCCFF %0001TT0640	Send new configuration with NN= 01 new ID TT= As read from configuration above CC= 06 for 9600 baud, 0A for 115200 baud. FF= 40 Engineering Unit, with crc
Switch off INIT. Reset module by switching power off and on.	
<ESC> rs2=-,9600 tm -a 2	Stop transparent mode. Rs2 9600 Baud Start transparent Mode to RS2 with Adam module CRC
\$012 !AATTCCFF<crc> #01 >+00.639+.....	Send to module 01, read configuration Answer of the module Ask for the data Answer of the module
<ESC>	Stop transparent mode
rq0=2,1,0,16,8 a16-23=a	Declare module, rs2, Id=01, protocol, uses inputs a16 to a23 Declare the inputs
rq=1	Start server
d -aq	Show configuration The new inputs will now appear: a16=- a17=- ... a23=-
rq	Display of statistics with the modules: Successful, error, time out.
a	A16 to a23 are the inputs on the remote module.

7.3 Configuration

You have to configure all used serial ports.

rq=<on>,<gap time>,<module timeout>

Starts or stops the Remote Acquisition Server, who polls the modules for new data.

<gap time> time in milliseconds between two requests to the modules. This time can be 0.

<module timeout> time in milliseconds, when the server treats a request as not responded.

When timeout time is 0, it will be set to 1000 (1s).

rq<module>=<rs port>, <slave id>, <protocol>,<input0>,<inputs>[,<options>]

Configuration of a module. modules can be from 0 to 199.

Inputs of the modules will be mapped to the inputs a<input0> to a<input0+inputs>.

Protocol		options
0	Adam remote modules, engineering values, with crc Also used by Gigalog boards as slaves	none
1000	Modbus Ascii. Data: 32 bit float.	number of the first port of the slave board

Calculating from raw to real value can be done by the slave and/or by the master.

Other protocols can be added

rqz

Zero statistic counters, clear last values from remote modules.

rq v=<verbose>

verbose=1: display errors.

verbose=2: display communication on the bus.

GigaLog as Remote Acquisition Slave

rqsl=<slave id>,<protocol>

Declare the board as a slave on the bus.

Gigalog recognizes the command "#<nn>" on all serial ports.

When the id <nn> is not the own slave id, Gigalog will not respond.

When the id <nn> is the own slave id, Gigalog answers with the data of all configured channels.

Gigalog as a repeater

A board, with RS2, and RS5 declared in slave mode, automatically works as a repeater.

When the board receives a message with its own slave address, it will respond to it.

When it receives any other messages from RS2, or RS5, it will forward it to the other RS485 port.

When it receives a broadcast messages #FF, it will respond, and forward the message.

7.4 Modbus Commands

Modbus ASCII protocol.

Before: Configure serial port, switch out rq.

ch=serial port. Default: last port used. Never specified: 2

mbscan[ch] [<timeout>]

Search for modbus modules with modbus id from 1 to 247.

Default timeout 10ms.

mb[ch] <hexstring>

Send a modbus command to a module. Wait and display response.

mbrd[ch] <id> <register> <format>

Send a modbus Read Holding Register command (0x03) to a module to read one or more registers. Wait and display response.

Format specifies the data, to transfer. It may include numbers for several registers.

format= d= decimal 16bit, x= hexadecimal 16bit, D= decimal 32bit, X= hexadecimal 32bit, f= float 32bit, s= string.

mbscan

board detected on rs2: id=4

board detected on rs2: id=9

board detected on rs2: id=247

ok

```
mbrd 4 100 6f4d
```

> :04030064001085

< 040320000000000003F800000000000000000000000000000000000005000080001C1

0.000000 1.000000 0.000000 0.000000 0.000000 0.000000 0 80 8 1

ok

7.5 Low power Mode

Several boards may be connected and use the low power mode to reduce power consumption.

Hardware

All boards must use a common GND.

The master may

- use the WKUP line, to wake up slaves from low power mode
- switch the power of the slave boards on, and off using the VRS2, or VRS5 line.

Slave boards, using WKUP line

Especially Gigalog boards may use the WKUP line, to wake up from sleep.

Slaves:

The boards may get the power supply over the 6 pin flat cable from the master.

Gigalog slaves: Configuration as described above.

lp=s

Master:

Set the VRS2RS5 jumper, to connect the power supply VIN to VRS2/5. See Gigalog M

Configuration as described above

The master pulls WKUP down, when coming out of sleep.

Use a sufficient <uptime> in the lp command, to allow the slave boards to be ready, to deliver values.

lp=m,3000

Set the timeout in the rq command to a small value, like 100 ms. Example: rq=1,0,100

You may also use a switch, to pull the open-collector WKUP line down, all boards will wake up.

Slave boards, switching power off in low power mode

Slaves:

The boards get the power supply over the 6 pin flat cable from the master.

Gigalog slaves: Configuration as described above.

Master:

Set the VRS2RS5 jumper, to connect MT3 to VRS2/5.

Configuration as described above

The master switches MT3 to VIN, when coming out of sleep.

Use a sufficient <uptime> in the lp command, to allow the slave boards to be ready, to deliver values.

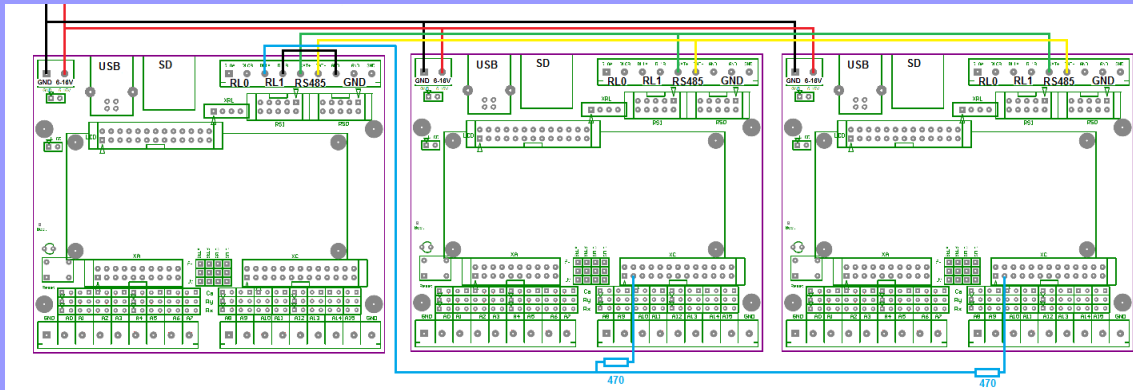
lp=m,3000,0,mt3

Set the timeout in the rq command to a small value, like 100 ms. Example: rq=1,0,100

You may also use a switch, to pull the open-collector WKUP line down, all boards will wake up.

7.6 Gigalog S

Gigalog S has only one RS485 port (RS2). The RS485 bus signals are available on screw-terminals. The board has no VRS2, nor WKUP signal.



Black, red lines: power supply.
Green, yellow lines: Rs485 network.

To wake up the slaves from low power mode,
an additional connection is needed, the blue line in the picture.

To protect the inputs on the slave boards, a 470 Ohm resistance must be used before the inputs.

When the boards do not have a common ground, this connection may damage a slave board!

Master: Connect RL1 one side to GND, the other side is the wakeup line.

Each slave: Connect XC pin 3 by a 470 Ohm resistance to the wakeup line.

8 Display

You may connect a graphical colour display with backlight and touch screen to the board;

8.1 Connection and power supply

Power supply: See chapter Hardware.

Connection directly on the board, or by flat cable to the IDC header on the LCD.

Configuration:

lcd=	LCD
0, 10, 90 (a)	alphanumeric, 2 lines x 16 characters
4, 14, 94 (a)	alphanumeric, 4 lines x 16 characters
5, 15, 95 (a)	alphanumeric, 4 lines x 20 characters
100,101 (s)	AMP320240
200,201 (s)	HY35a
250,251 (s)	HY35a, reverse

(a) 0*= show all, 1*= do not display 1. page, 9*= show only 1. page: board name, date, and time

(s) **0= real data, **1= simulation data

lcd

The command lcd without parameters shows all installed display drivers

lcd <text>

Display text on the alphanumeric LCD display, 2. line.

The text may include spaces. A ';' indicates the end of the command.

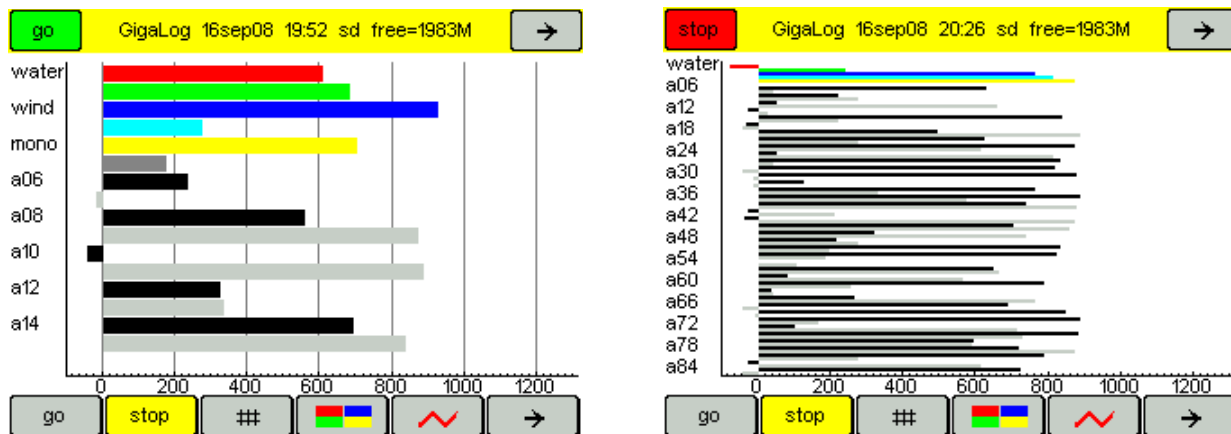
For special characters sequences see chapter Special characters.

For graphical LCD, see command grw.

All configuration values, including those entered by the touch screen, like the parameters to display a channel, can also be entered by a serial input or USB, and are part of the normal configuration.

Following information is only valid for the graphic displays with touch screen:

8.2 Page Bar graph



Show internal and external up to 100 inputs as bar graphs in real time.

You may declare up to 5 bar graph pages.

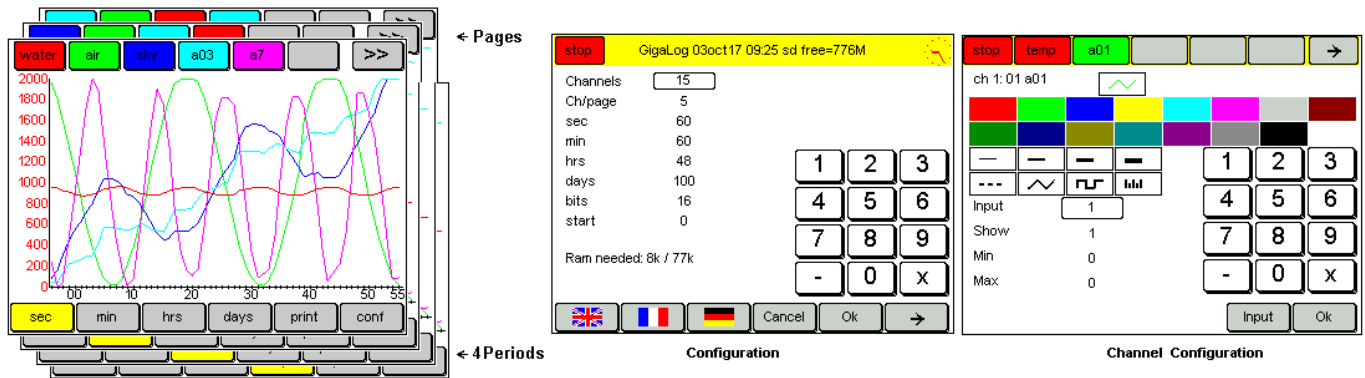
Choose the inputs to display in the configuration.

Min and Max determine the mapping of the data on the x axe.

When both are zero, or Min >= Max, raw values will be shown in millivolt.

Otherwise the LCD shows engineer values.

8.3 Page Log Data



The display shows 15 channels on 3 pages, 5 channels per page.
It shows

- The last 60 seconds, or
- The last 60 minutes, or
- The last 48 hours, or
- The last 100 days.

You may change the number of channels, channels per page, and the length of each period.

The data for the display is stored independent of the data logged on the memory card.

The data for the display will be lost when power is switched off.

The data for a minute is calculated from the last 60 seconds. If there are less than 60 seconds, the minute is calculated from the existing data. If there are more than 60 seconds, they will be ignored.

Data for an hour are calculated from the last 60 minutes, and for a day from the last 24 hours in the same way.

Upper menu

The upper menu displays a state line or the channels of the page.

The menu toggles automatically after some seconds to the state line.

Channels are not equal to inputs. You may assign any input to a channel.

The button of a channel is in its colour, and displays its name.

Clicking on the button of a channel brings the channel into the foreground and its y-axis is displayed.

Clicking on -> changes the page and displays other channels.

Bottom menu

You may select on the bottom menu the period to display

-> Advance the menu

Toggles the grid on or off

Print Store the picture on the screen into a Bitmap file on the memory card.

Configuration of a Channel

Double click on its button in the top menu line.

Select its colour, width, and style. Select an input.

Min and Max determine the appearance of the function on the screen.

Do not mistake with the min and max of an input to trigger an alarm.

When these values are zero, the appearance will be calculated from the data.

Show: display the channel, on or off.

Select a field, and enter the value using the keyboard.

Name, colour, and style belong to the configuration of the input, and can be changed using GigaTerm.

Configuration of the graphic

Click on "conf" in the bottom menu line.

You may change the number of channels, channels per page, and the length of each period.

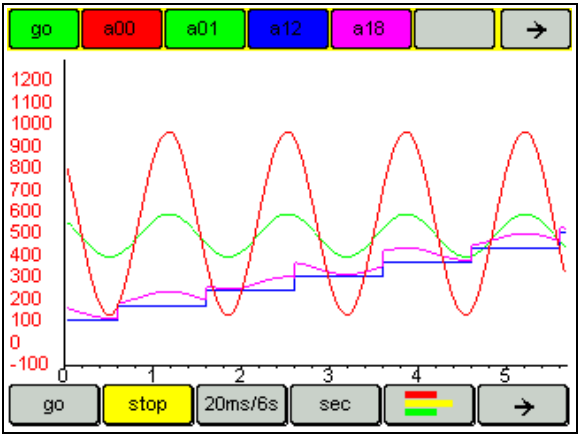
The number of bits per value: 16, 24, or 32 bits.

Be careful with the RAM needed. The displayed needed RAM must not turn to red.

Change the language.

Adjust the touch panel.

8.4 Page Scope

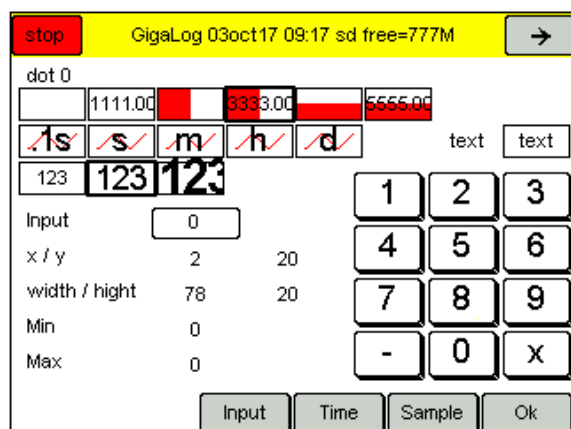
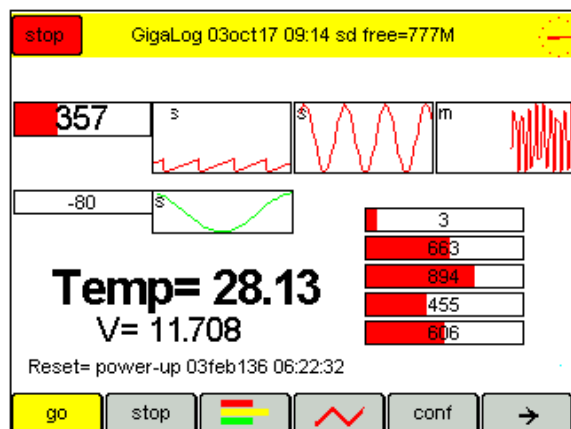


Similar to logged data page. You reach this page by the log data page, click on the "ms" button.

Mode	Sampling	On screen
Scope Inputs are displayed as sampled. Stop stops sampling at the end of the display. Click on Stop again for a one shot on the display.	1 ms	.3 s
	2 ms	.6 s
	5 ms	1.5 s
	10 ms	3 s
	20 ms	6 s
	50 ms	15 s
	100 ms	30 s
	200 ms	60 s
	500 ms	150 s
Log Data stored in volatile RAM. Average data over sample period. Log length configurable.	1 s	90 s
	1 m	60 m
	1 h	48 h
	1 d	100 d

Display of 5 channels per page (configurable)
5 pages of data (configurable)
Display of data on the LCD is completely independent of the data logging to the memory card.

8.5 Page Installation



Show an installation with inputs.

Installation background image

When there is a file "machina.bmp" on the memory card, the display takes this file as background image. Display from a file is slow and only useful for demonstration and test. If the file does not exist, it takes "machina.bmp" from Flash memory. To change this file, you have to rebuild the firmware. Machina.bmp is a 16 colours Bitmap file. Maximal size: width: 320 pixels, height: 180 pixels. When the Flash image is not bigger than 160 x 90 pixels, it will be doubled in size. A 320 x 180 pixel image in Flash memory costs 29 k, a 160 x 90 pixel image only 7 k. The resolution is lower, but normally sufficient.

You can also use the gri command (see below) to replace the image in the Flash memory.

Installation dots:

- Bar graphs

A bar graphs displays the actual state of an analogue inputs.

- Log

Shows the last seconds, minutes, hours of logged data of an input.

- Text

Free text output, May include escape characters to show values. See 3.11 Configuration, special characters.

Click and drag on a dot to displace it.

Click on a dot to change its configuration.

Configuration:

Choose bar graph, log, or text.

Choose an input and a style. Style 0 will not be displayed.

Position on the LCD, width and height.

Min and Max determine the mapping of the data..

When both are zero, or Min >= Max, raw values will be shown in millivolt.

8.6 Page Command terminal

On the terminal page, You can enter commands as on a Rs232 or USB terminal.

```
gre2=2,5,118,51,84,90,0,1000 gre3=3,3,46,158,50,12,0,2000
gre4=4,3,200,151,50,12,0,2000
rq=0,0,0,0,0
2008:12:16 09:55:49
sd used=4256k + free=3867M =3871M
board stop
ls
gigalog.adc          4248996 13dec08 10:42:04
ok
grp image.bmp
ok
```

1	2	3	4	5	6	7	8	9	0	=	bs	
a	b	c	d	e	f	g	h	i	j	k	l	m
n	o	p	q	r	s	t	u	v	w	x	y	z
%	.	,	;	+	-	*				cmd	end	

8.7 Configuration, Commands

Other pages allow you to change many important configuration values, like inputs, sampling, and time.

The first screenshot shows the 'Input' configuration page for 'a0'. It includes fields for Name, Type, Calculation, Alarm <, and Alarm >. The second screenshot shows the 'an' configuration page for 'gigalog.adc', with fields for ad, lp, grlp, gry, and gri. The third screenshot shows a numeric keypad with a date '08/01/2014' and an 'OK' button.

You may also use the serial interface (Rs232, or USB) to configure the LCD.

**grt=<language>,<channels>,<channels per page>,<seconds>,<minutes>,<hours>,<days>,<bits>,<start>
grst=<start>**

<start>= 11=Log data sec, 12=min...; 21=Scope 1ms, 22=2ms,... 30=Bar graphs, 40=Installation
100=personal page (See programming manual)

gr<ch>=<input>,<show>,<min>,<max>

grb[<n>]=<in0>,<ins>,<min>,<max>

Bar graph page

gre[<n>]=<input>,<style>+,<x>,<y>,<wd>,<ht>,<min>,<max>[,<text>]

<style>= 1..5 bar graphs, 6-10 log, 14..15 text

= 0 small, 100 normal, 200 big

Installation dot

gry= [c][b][l][i][t][s]

Graphic display user no access to c= configuration, b= bargraphs, l=log data, i=installation, t= command terminal, s= change state Go Stop.

grc=<ch>

Change page, put channel into foreground.

grw [<color>,<text>

Set headline message. Colors are 0=red, 1=green, 2=blue, 3=yellow, 4=cyan...

<color>= 100 + color: wakeup sleeping LCD.

For special characters sequences see chapter Configuration Special characters.

grpop <time>,<text>

Display pop-up message for <time> seconds. A | in the text indicates new line.

gri <filename>

Load installation background image from file to internal Flash memory. Image must have same size, width and 16 colors.

grx

Refresh screen.

grp <filename>

Print to file.

gra

Calibrate touchscreen.

grlp=<timeout>,<mode>

groff

Cut power supply graphic display.

gron

Establish power supply graphic display

Ram usage: (<seconds>+<minutes>+<hours>+<days>) * <channels> * 2/3/4 bytes. 2/3/4 bytes for 16 bit, 24 bit, or 32 bit mode resp.

8.8 Electronic Paper Display (epd)

You may connect an electronic paper display to the board

Connection
Gigalog M

| XC connector | Display |
|--------------|--------------|
| 9 | Busy, purple |
| 10 | Reset, white |
| 11 | DC, green |
| 12 | CS, orange |
| 13 | SCLK, yellow |
| 14 | SDA, blue |
| 1 GND | GND, brown |
| 2 V33 | VCC, grey |

epd=<display>

Select display.

Supported displays are:

| | |
|----------|---|
| epd29bw | Waveshare 2.9" 296x128 black,white 12956 |
| epd29red | Waveshare 2.9" 296x128 black,white,red13339 |

epd=

Show supported displays

epd[0-19]=<x>,<y>,<wd>,<ht>,<style>,<text>

<x>,<y> top left position on screen

<wd>,<ht> dimensions of the rectangle, can be 0

<style>=<fg-color><bg-color>

=10|18|25|41 <color>=b|w|r|y

The text may include spaces. For special character sequences see chapter Special characters.

epd

Send image to the display

epd d

Display configuration

epd z

zero: clear configuration

epd on[=<rtime>]

Start continuous display update. Rtime time between 2 refreshes in seconds.

epd off

Stop continuous display update

epd demo

set demo configuration

9 Application Notes

9.1 Memory cards, File lengths, Transfer to the PC

Analogue data are sampled into a text file, each sample in a line.
A typical line looks like this:

```
2016:11:30 14:07:50 5120,45 33333 1289,00 123456
```

This line includes date, time, and the data of four channels.
It has a length of 48 characters and occupies together with the line end 50 bytes in the file.

You will find a calculator to get the amount of disk space needed at: controlord.fr/lpcalc/lpcalc.htm

Note, that the real sample rate is higher than the selected sample rate. GigaLog stores the average sum of the last period in the disk file. See analogue data, Sample rates.
Very often a slower sample rate is sufficient.

Milliseconds

When the sample rate is faster than 1 second, normally only the millisecond will be stored instead of the whole date time.

This takes less place in the file.

Example for a 50 ms rate:

```
2016:11:30 14:07:50:000 5120,45 33333 1289,00 123456
050: 5120,45 33333 1289,00 123456
100: 5120,45 33333 1289,00 123456
150: 5120,45 33333 1289,00 123456
```

Transfer of data to the PC.

Data can be uploaded with GigaTerm to the PC using the serial interface. Since this is slow, huge data shall be transferred using a memory card reader on the PC. The data is stored in a file on the memory card.

Reading the data on the PC

Data can be read by

- GigaData, shows the data graphically.
- Text editor, like Blocknotes, Word, Write, etc
- Spreadsheet, like Ex*el

Other data formats

Configuration fields as, am, ae determine the format of the stored data

See Configuration, Analogue inputs, Frames. Default values are

| | | | |
|----|--------------------------|---|--|
| as | Frame start | d | Date: yyyy:mm:dd hh:mm[:ss[:uuu]]<tab> |
| am | Frame start, millisecond | m | millisecond <tab> |
| ae | Frame end | n | carriage return new line<cr><nl> |

9.2 Tuning the ADC

The ADC is an independent unit, that scans automatically and cyclic all inputs by a multiplexer, converts the input to a digital value and informs the microprocessor, that a new result is available. The firmware reads out the digital value of the input. During this time, the ADC already converts the next channel.

If you declare an analogue channel as z or as v, for example a15=z, it will not participate in sampling, and the sample rate will increase for the other channels. You may also declare the partner of a differential channel as z, for example a8=a,d=1; a9=z.

Reasons to change the configuration of the ADC

- Increase the resolution
- Increase the speed for fast analogue data, faster than 100 Hz
- Increase the speed for fast counter inputs

| 16 inputs | | | | | | | | |
|----------------------|------|-------|-------|-----------------|--------|--|----------------------------|-------------|
| Configuration values | | | | Measured values | | | | |
| ax | Chop | Delay | Drate | Sps | Sps/16 | | Analogue Inputs resolution | Counter max |
| 1021 | 1 | 2 | 1 | 2760 | 172 | | +1 bit | 82 Hz |
| * 1012 | 1 | 1 | 2 | 6630 | 414 | | 0 | 200 Hz |
| 1013 | 1 | 1 | 3 | 9800 | 613 | | -0.5 bit | 290 Hz |
| 0002 | 0 | 0 | 2 | 14800 | 930 | | -1.5 bit | 450 Hz |
| 0003 | 0 | 0 | 3 | | | | | |

| 2 inputs, a2=z; a3=z; a4=z a15=z | | | | | | | | |
|---------------------------------------|------|-------|-------|-----------------|-------|--|----------------------------|-------------|
| Configuration values | | | | Measured values | | | | |
| ax | Chop | Delay | Drate | Sps | Sps/2 | | Analogue Inputs resolution | Counter max |
| 1021 | 1 | 2 | 1 | 2760 | 1380 | | +1 bit | 480 Hz |
| * 1012 | 1 | 1 | 2 | 6630 | 3314 | | 0 | 1180 Hz |
| 1013 | 1 | 1 | 3 | 9800 | 4900 | | -0.5 bi | 1700 Hz |
| 0002 | 0 | 0 | 2 | 14800 | 7400 | | -1.5 bit | 2600 Hz |

Configuration values:

Ax

Value of the configuration for the analogue to digital conversion: <chop><bias><delay><drate>.

The value 1012 is the default value.

Chop (0..1)

When chop is 1, the adc converts the + and - inputs as wanted, and does then a second conversion with the opposite polarity.

Chop=1 reduces the offset error.

Delay (0..7)

Time after switching the multiplexer to another input, before starting the conversion.

Some time is needed to have a stable input.

Drate (0..3)

The ADC executes several consecutive conversions on an input and calculates the average sum.

A small value indicates many conversions.

Drate corresponds to a digital low pass filter.

Increasing drate reduces the filter but gives more results to the firmware. The firmware also calculates the average sum by another digital filter.

Measured values:

Sps, Sps/16



Samples per seconds, is the number of probes in a second.
Sps/16 is Sps divided by 16, and the number of probes in a second for each input.

Analogue Inputs resolution
Increasing or decreasing of the resolution relative to the default value.

Counter max
When using the input as a counter, the maximal allowed frequency is Sps/16 divided by 2.
For example: Ax=1012, the default value, Sps/16 is 414. The maximal allowed frequency is 200 Hz;
the signal must be stable for 2.5 ms for each level, 0 and 1.

The command ax gives information about the ADC. The above table was made by this command.

Reference: Texas Instruments: Ads1258 datasheet Rev. G Mars 2011.

9.3 Calculation from the analogue input raw value

Calculation from analogue input raw values of the ADC to real values, or engineer values by a linear equation are handled in

- Hardware, From raw values to Engineering values, calibration
- Configuration, Analogue inputs a<ch>=
- GigaTerm, Inputs. From raw values to Engineering values

Reference input

When an input is in relation to another input, you can subtract the reference input with p=<ch>. The subtraction will be done on the real values of both inputs.

Example: Heating: A1= outgoing water temperature, a2, a3,... back coming water temperature of several cycles. The board samples the outgoing temperature, and the temperature difference for each cycle.
A1=t; a2=t,p=1; a3=t,p=1 ...

See Configuration, Analogue inputs a<ch>=[,p=<reference>]
Do not confuse with differential mode.

Virtual input, calculation

Declaration of a virtual input, calculated from other inputs by an expression.

Expression is a sequence of up to 5 elements, connected by +, -, *, or /.

Elements are inputs or small constants (0..99).

Example: c=a0-a1*a2

Calculation will be done on the calculated real values of the inputs, strictly from left to right.

The value of a counter is the count of the last full period, as defined by ad.

You may select any analogue input a0..a15 for a virtual input, the analogues input will then be lost.

You may also select any input from a16 or higher as virtual input.

The result of the calculation can then be recalculated by *m+p,c into another unit or output format.

A16=v*100,2 does not change the result, but the output format in the file and on the display.

Example heat pump:

A1= temperature incoming water, a2= temperature outgoing water, a3= water flow.

Calculation of the power: (a2-a1)*a3: A16=v,c=a2-a1*a3

Use *m+p,c to convert the result into watt.

See Configuration, Analogue inputs a<ch>=v

A virtual input can not trigger an alarm.

A virtual input may use other virtual inputs as parameters. Calculation is done from a0 to a99 without recursion.

Long time counter

A long time counter adds values of a long period.

When switching power off, no data, or only data of the last seconds will be lost.

Long time counters use the same syntax as virtual inputs, with a<ch>=vc instead of a<ch>=v

Example: Tachometer, one pulse per revolution; ad=1m

A7=c stores the current speed in revolutions per minute.

A16=vc,c=a7 counts all revolutions.

Example: Heat pump as above.

Calculation of the energy (a2-a1)*a3: A17=vc,c=a2-a1*a3; Or A17=vc,c=a16

Use *m+p,c to convert the result into watt hours.

A16 stores the current power in watt.

A17 stores the complete energy in watt hours since the begin of sampling.

Long time counters are saved each minute on the memory card in the file counts.txt and counts2.txt. The firmware also stores the counters when switching from Go to Stop mode. There is no counting in Stop mode.

After Reset the firmware reads the counters from the files.

The av command changes the counters

av= z clears all counters

av<ch>= <n> sets a single counter.
av<ch>+= <n> increments a single counter.

See Configuration, Analogue inputs a<ch>=vc

Time counter

a20=vc
m4=0,60,av20+=1

The long time counter A20 is not connected to any input.
The macro m4 increments the counter each minute.
A20 counts the operating hours based on minutes.

9.4 Temperature Sensor LM60

The LM60 is an integrated circuit in TO92 housing.

It senses temperatures from -40°C to $+125^{\circ}\text{C}$.

The LM60 output voltage is linear to the measured temperature, thus easy to connect to a GigaLog board.

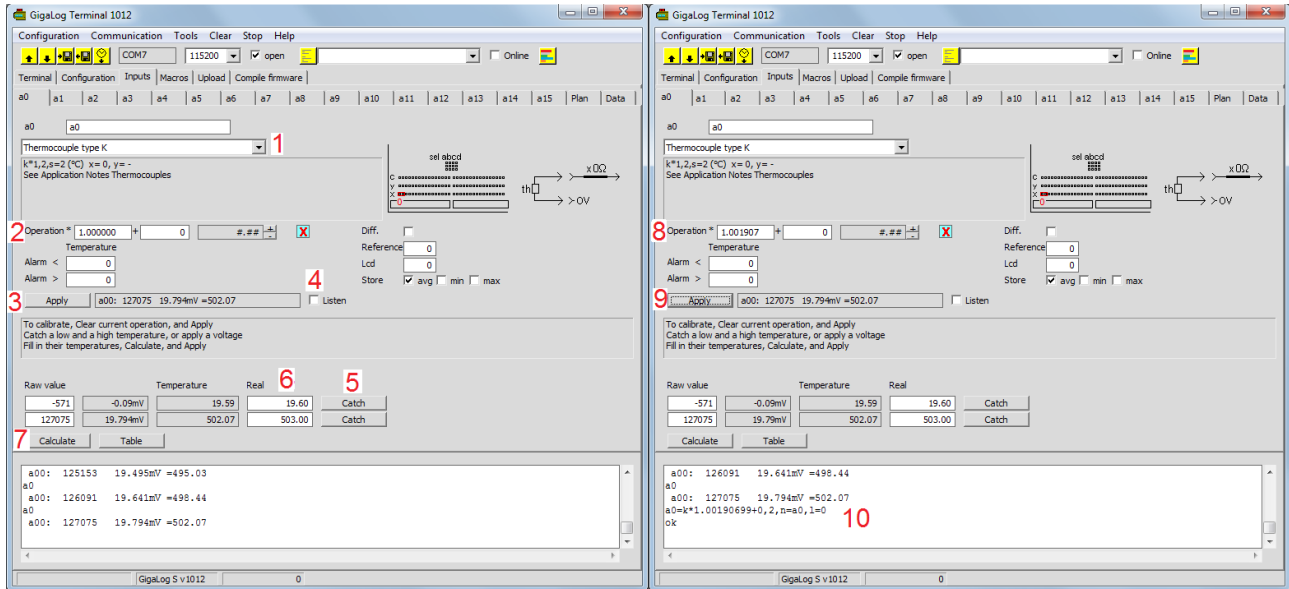
The expression, to calculate temperature from the input voltage, is

$a0 = a \cdot 0.00249219 - 6784,2$ Resolution $0,01^{\circ}\text{C}$

9.5 Thermocouples

You can directly connect a thermocouple type K to an analogue input of the board.
A thermocouple type K measures temperatures up to 1300 °C.
A thermocouple outputs a very small voltage, the signal is very sensitive to noise.
The thermocouple voltage is relative to the board temperature.
Some cold junction compensation is needed, to find the absolute temperature.
This is done by the firmware on the board.

You may calibrate a thermocouple input.
Calibration applies a linear equation to the result, thereby slightly moving it.



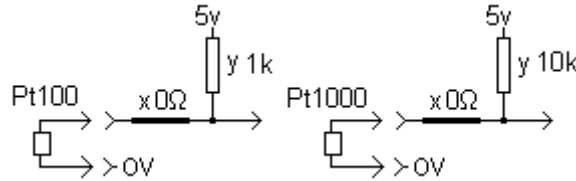
Open GigaTerm, tab Inputs, open the input.

1. Select or reselect thermocouple
2. Verify that you have the original operation: $*1+0 \#.$
3. Click on Apply to send this configuration to the board
4. Click on Listen to visualize the board temperature
5. Catch a low temperature in the 1. line, for instance the surrounding temperature
6. Enter the wanted temperature with two digits: 19.60
5. Catch a high temperature in the 2. line. Take a temperature as high as possible.
6. Enter the wanted temperature with two digits: 503.00
7. Click on Calculate to calculate the linear equation
8. The new operation. The multiplier of the operation shall be close to 1, (between 0.9 and 1.1). Else there is an error.
9. Click on Apply to send the new configuration to the board
10. The board receives the configuration and responds with OK.

You may also simulate a temperature, by applying a voltage to the input, and fetching the temperature from a table.
You have to add the board temperature, and insert the sum as the wanted temperature.

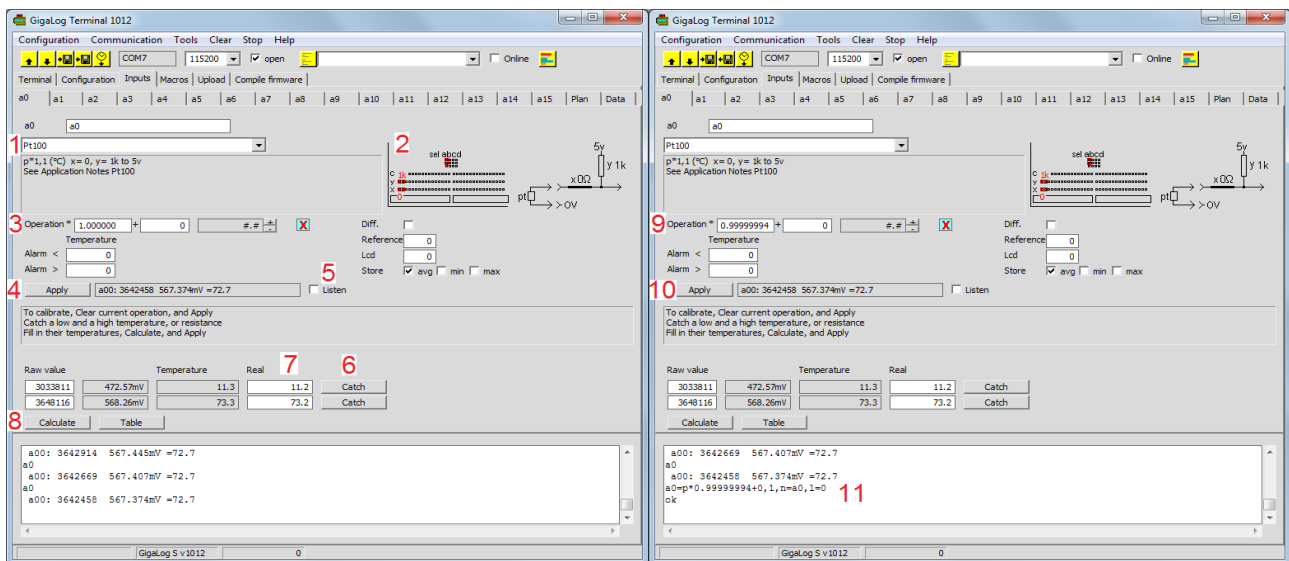
9.6 Pt100, Pt1000

Pt100 measure temperatures from -200°C to $+850^{\circ}\text{C}$. The Pt100 is a resistance that changes with the temperature. To transform this resistance into a voltage at the input of a GigaLog board, you need a 1k Ohm pull up resistor to 5V. The result is a non linear function, that the firmware converts into temperature (0.1°C).



The wires from the sensor to the board should be as short as possible, since they also have a resistance. Use a 0.1 % resistance as pull up resistor y. For 3-wire, and 4-wire sensors, see below.

You may calibrate the Pt100 input. Calibration applies a linear equation to the result, thereby slightly moving it.

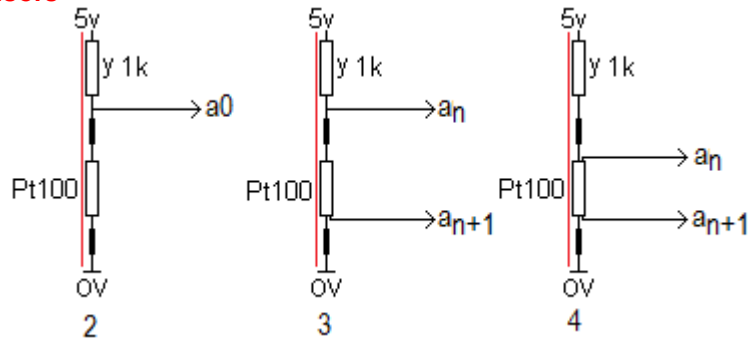


Open GigaTerm, tab Inputs, open the input.

1. Select or reselect Pt100.
2. Verify the resistors and jumpers.
3. Verify that you have the original operation: $*1+0 \#.$
4. Click on Apply to send this configuration to the board
5. Click on Listen to visualize the board temperature
6. Catch in the 1. line a low temperature, as low as possible.
7. Enter the wanted temperature with 1 digit: 11.2
6. Catch in the 2.line a high temperature. Take a temperature as high as possible.
7. Enter the wanted temperature with 1 digit: 73.2
8. Click on Calculate to calculate the linear equation
9. The new operation. The multiplier of the operation shall be close to 1 (between 0.9 and 1.1). Else there is an error.
10. Click on Apply to send the new configuration to the board
11. The board receives the configuration and responds with OK.

You may also simulate a temperature, by applying a resistance to the input, and fetching the temperature from a table.

Pt100 3-wire 4-wire Sensors



The current (red line) goes through the ground wire, the Pt100, the 5V wire, and the 1k resistor. When cables are long, the resistance of the wires will impact the result. 3-wire and 4-wire resistances reduce this error.

The wire resistance does not impact the input, as long as there is no current; The small current, needed by the high impedance analogue input, does not count.

Pt100 3-wire

Refer to picture 3

Gigaterm: Select PT100 3-wire.

The firmware calculates the voltage on the Pt100 as $a_n - 2 \cdot a_{n+1}$, thus subtracting the wire error.

Pt100 4-wire

Refer to picture 4

Gigaterm: Select PT100 4-wire.

The differential input only measures the voltage on the PT100.

You must provide the needed 1k resistor, and the +5 V either from another unused input, or external.

9.7 Low power mode current

You will find a calculator to evaluate the current and the needed battery here: controlord.fr/lpcalc/lpcalc.htm

The average current used in low power mode is calculated by

$I = I_{\text{sleep}} + Q_{\text{sample}} / t_{\text{sample}} + Q_{\text{store}} / t_{\text{sample}} / q_{\text{rate}}$

The ad command specifies the tsample time

The lp command specifies the qrate.

Qsample

The load, to take one sample

Qsample&store

The load, to take one sample and to write one, or several samples to the disk.

Qstore

is Qsample - Qstore

Isleep

The basic current of the board in low power mode (lp=m, board running)

The brownout detector (bo=1) needs an additional current in Isleep.

Removing the brownout detector (bo=0) removes this current.

Drawback: the board will no longer reset, when voltage drops to critical values.

Board will still reset, when voltage goes close to 0 V.

See chapter Configuration Miscellaneous.

Low power mode FTP transfer

Qftp

The load of a complete data transfer by ftp to an Internet server, using the mobile modem.

This load depends on many conditions: Signal reception (antenna), network, provider, ftp server. It's a long way.

Measurement under the following conditions:

an=in.txt

ad=1m

mm=1

lp=1,0,10

macro, all x minutes: mmstart 20,40,60; ftp -cdt in.txt; mmstop

The macro sends only new data, that has not yet been send, to the server.

If the transfer fails, this data will be send with the next transfer.

No data will be lost, all data will be send to the server.

Nevertheless, if many transfers fail, it may be useful, to insert a wait of at least 25 s (wt 25s) or more between the mmstart, and the ftp commands. In this case you have to increase the timeout values of the mmstart command.

Sending small packages of data by FTP to the server each hour costs much more current than sending a long package once a day.

9.8 OEM Version

This chapter explains how to create an OEM version, that you can handle to your customers.

GigaLog:

Change configuration boardname. The name appears in the LCD and elsewhere
bn=<xyz>

GigaTerm:

Rename GigaTerm.exe to <xyz>.exe

All names Gigalog will be replaced by <xyz>

The page to recompile the firmware disappears

<xyz>.exe Will call <xyz>Data.exe instead of GigaData.exe to display data files.

Inputs

You may also edit the list of known inputs for the Input tab

This is especially useful, when using special probes.

Edit file ainputs.txt

- Delete elements
- Add new elements for your specific probes:
- Name, jumpers, resistors,
- Function to calculate engineering values
- Comments

Zip the following files, and handle them to your customer:

<xyz>.exe

<xyz>Data.exe

ainputs.txt Your version

Readme.txt Your version

ledit32.dll

usbdriver

Serial number

Ask us, if you want to give your individual serial number to each board.

10 Gigalog S

10.1 Technical Summary

| | |
|--|--|
| Power supply
Board with alphanumerical LCD 2x16 | typ. 75 ma @ 6-15 V |
| Low power mode
I sleep (qc>= 1705)
I sleep (qc<1705)
I brown out detector
Q sample
Q sample & store
Q store
Ftp first 1 k of data
Ftp each following k of data | 0.025 mA
0.160 mA @7V, 0.200 mA @12V
0.020 mA
13 mAs
23 mAs (Sandisk Ultra)
10 mAs (Sandisk Ultra)
~ 5000 mAs in 40s @6V
~ 80 mAs in 0.5s @6V |
| Analogue Inputs | |
| Impedance
all inputs used
only one input used, others set to a<n>=z | > 10 M Ohm
> 2 M Ohm |
| Input range
total max
measured | -0.1 to 5V
-100 to 1300 mV |
| ADC Resolution
Measured average resolution at storage rate
< 10ms
> 10ms, < 100 ms
> 100 ms | 24 bits signed

16 bits
18 bits
19 bits |
| Reference voltage ADC | 1225 mV ± 1.2 mV @ 25°C; ± 10 mV @ 0..70°C |
| Conversion
ADC -> voltage
voltage -> ADC | $U = \text{ADC} * 1225 / 0x780000 = \text{ADC} * 0.1557668 \mu\text{V}$
$\text{ADC} = U(\text{mV}) * 6419.85$ |
| Digital inputs
A0 to A15
Logical 0
Logical 1
XC
Logical 0
Logical 1 | 0 to 0.8 V
1.2V to 5V

0 to 0.8 V
2V to 5V |
| Real time clock error
typ
max | 5 ppm (3 min/y) @ 25°C + 0.05 ppm/°C
20 ppm (10 min/y) @ 25°C + 0.05 ppm/°C |
| Relays output | 100mA 250 V |
| Environmental temperature
Operating
Storage | -10 .. +50°C
-20 .. +70°C |
| Mechanical length x width x height, weight
Board only
Board with mounted alphanumeric LCD 2x16 | 101 86 15 mm, 75 g
101 86 25 mm, 105 g |

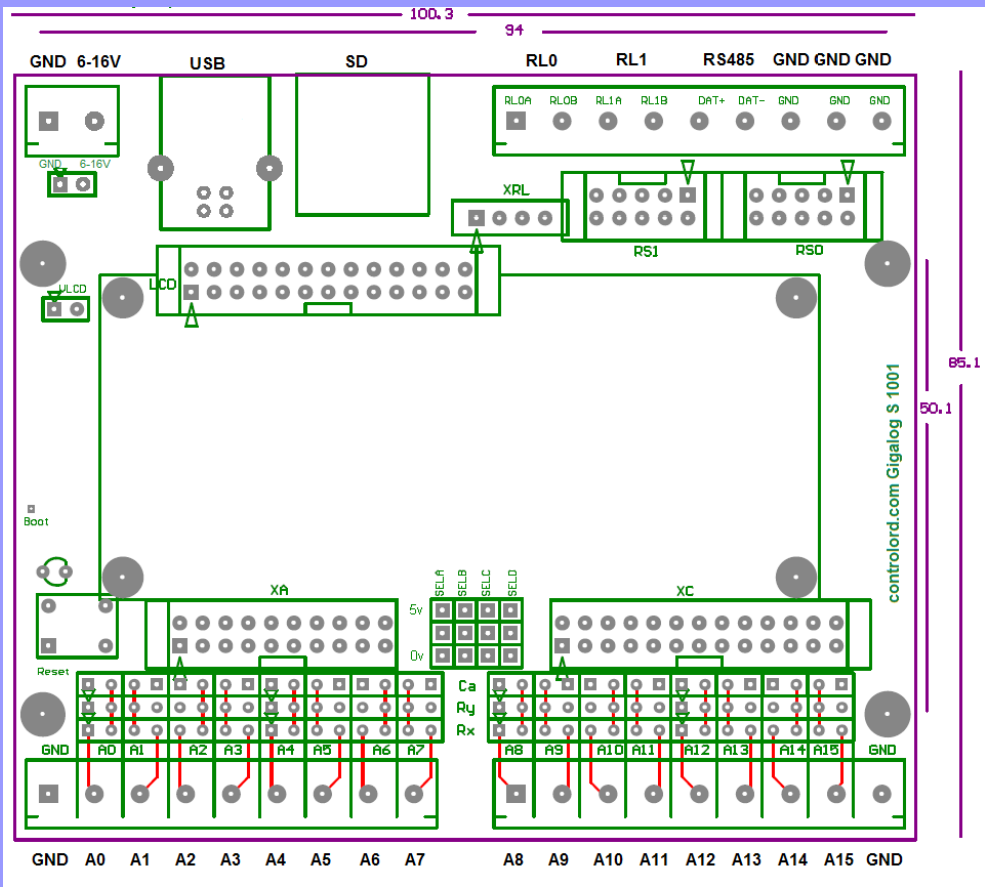
ADC: Texas Instrument ADS1258 in delta-sigma technology.

Voltage reference: LM4041-AIM3-1.2.

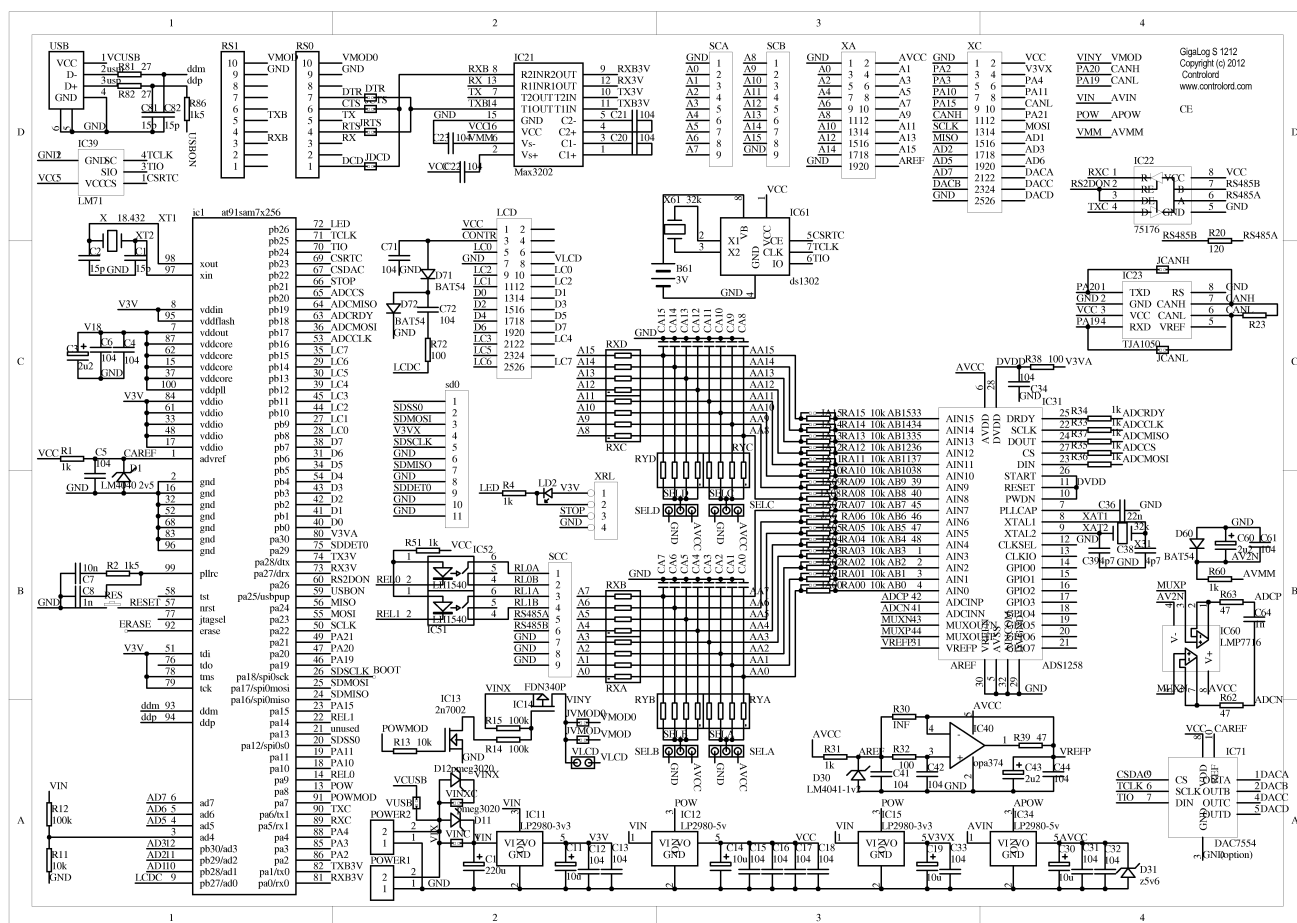
Battery cr1220 3V Lithium, only used for real time clock and counters (bm...).

The configuration is stored in the internal Flash memory of the microcontroller.

10.2 Component placement



10.3 Schematics



11 Gigalog M

11.1 Technical Summary

| | |
|--|---|
| Power supply
Board without display, without modem | typ. 120 ma @ 6-13 V |
| Low power mode
I sleep
I brown out detector
Q sample & store
Ftp first 1 k of data
Ftp each following k of data | 0.020 mA
0.010 mA
54 mAs (Sandisk Ultra)
~ 5000 mAs in 25s @6V
~ 30 mAs in 0.2s @6V |
| Analogue Inputs | |
| Impedance
all inputs used
only one input used, others set to a<n>=z | > 10 M Ohm
> 2 M Ohm |
| Input range
total max
measured | -0.1 to 5V
-100 to 4200 mV |
| ADC Resolution
Measured average resolution at storage rate
< 10ms
> 10ms, < 100 ms
> 100 ms | 24 bits signed

16 bits
18 bits
19 bits |
| Reference voltage ADC | 4096mV ± 4 mV @ 25°C; ± 40 mV @ 0..70°C |
| Conversion
ADC -> voltage
voltage -> ADC | $U = \text{ADC} * 1225 \text{ mV} / 0x780000 = \text{ADC} * 0.520833 \mu\text{V}$
$\text{ADC} = U(\text{mV}) * 1920$ |
| Digital inputs
A0 to A15
Logical 0
Logical 1
XC
Logical 0
Logical 1 | 0 to 0.8 V
1.2V to 5V

0 to 0.8 V
2V to 3.3V |
| Real time clock error
typ
max | 5 ppm (3 min/y) @ 25°C + 0.05 ppm/°C
20 ppm (10 min/y) @ 25°C + 0.05 ppm/°C |
| Motor output | 1000mA 13 V |
| Environmental temperature
Operating
Storage | -10 .. +50°C
-20 .. +70°C |
| Mechanical length x width x height, weight
Board only
Board with mounted alphanumeric LCD | 100 80 15 mm, 75 g
100 80 35 mm, 150 g |

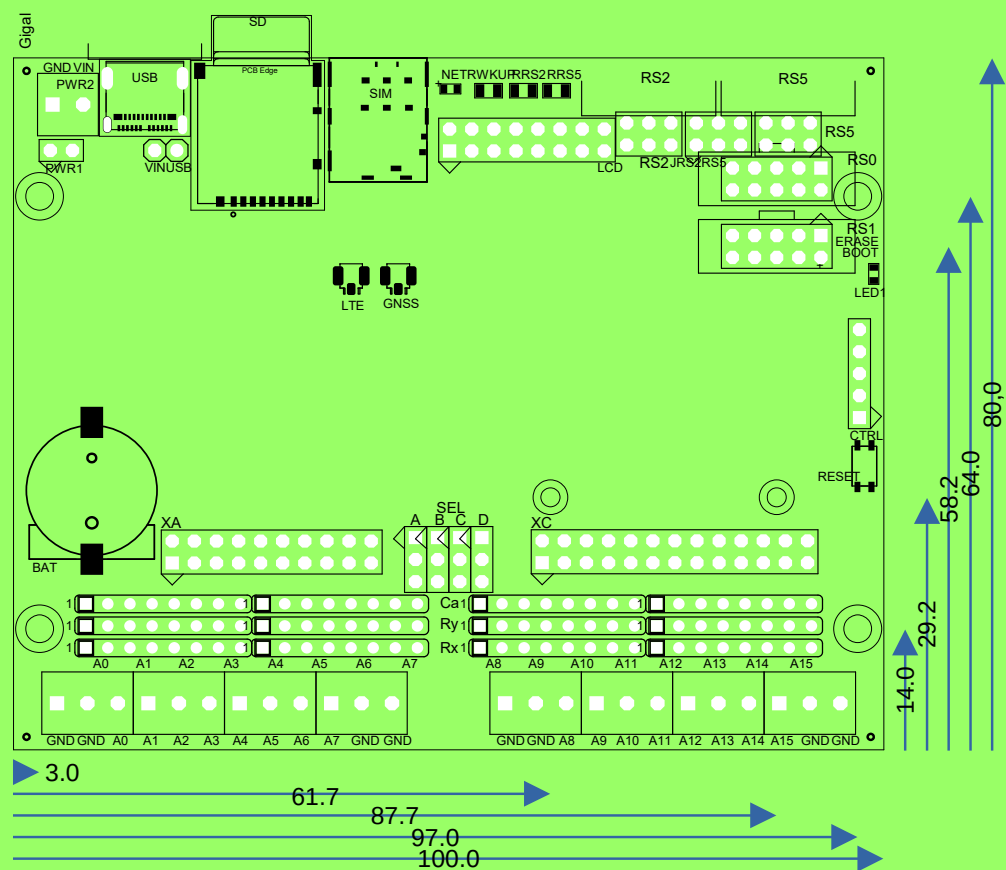
ADC: Texas Instrument ADS1258 in delta-sigma technology.

Voltage reference: LM4040-AIM3-4.1.

Battery cr1220 3V Lithium, only used for real time clock and counters (bm...).

The configuration is stored in the internal Flash memory of the microcontroller.

11.2 Component placement



| | | |
|------------------------|------|------|
| | x | y |
| size | 100 | 80 |
| hole M3 | 3 | 14 |
| | 97 | 14 |
| | 3 | 64 |
| | 97 | 64 |
| hole daughter board M2 | 61.7 | 29.2 |
| | 87.7 | 29.2 |
| RS1 centre | 87.7 | 58.2 |

11.3 Schematics

