DATA PLOTTER

User manual

Jiří Maier

TABLE OF CONTENTS

CONNECTING TO THE DEVICE

The program searches for available COM ports and displays them in a list (including the name of the device, if available). The list is automatically updated on an ongoing basis.

On Linux, it may happen that an application cannot access a port. This can be solved with commands:

```
sudo usermod -a -G tty $USER
sudo usermod -a -G dialout $USER
```

Any number other than the preset baudrates can be entered in the field. You can also set other parameters, such as parity, by clicking on the button with the setting symbol. If the port is connected, any change to the settings will cause a disconnect. The information button displays all detectable information about the port.

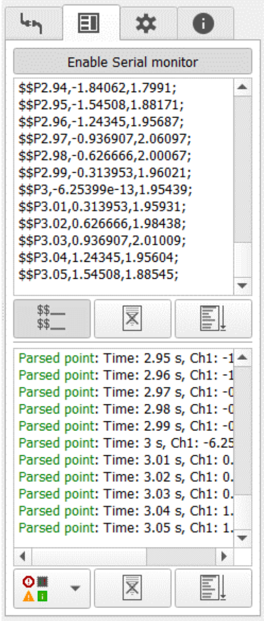
DEVICE DATA INFORMATION

Worthy text field displays the data received from the serial port, the display must be enabled with the button.

The bottom text box displays information about received messages and any error messages, as well as information and warning messages from the device. You can set the listing level:

- Device messages only (**`$$I`** and **`$$W`**)
- Device messages and errors
- Device messages, errors and warnings
- All announced

Messages from the device are also displayed on the first page (connection)



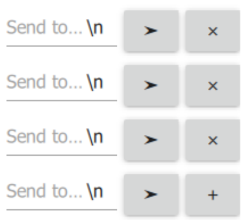
```
$$P2.94,-1.84062,1.7991;
$$P2.95,-1.54508,1.88171;
$$P2.96,-1.24345,1.95687;
$$P2.97,-0.936907,2.06097;
$$P2.98,-0.626666,2.00067;
$$P2.99,-0.313953,1.96021;
$$P3,-6.25399e-13,1.95439;
$$P3.01,0.313953,1.95931;
$$P3.02,0.626666,1.98438;
$$P3.03,0.936907,2.01009;
$$P3.04,1.24345,1.95604;
$$P3.05,1.54508,1.88545;
```

```
Parsed point: Time: 2.95 s, Ch1: -1
Parsed point: Time: 2.96 s, Ch1: -1
Parsed point: Time: 2.97 s, Ch1: -0
Parsed point: Time: 2.98 s, Ch1: -0
Parsed point: Time: 2.99 s, Ch1: -0
Parsed point: Time: 3 s, Ch1: -6.25
Parsed point: Time: 3.01 s, Ch1: 0.
Parsed point: Time: 3.02 s, Ch1: 0.
Parsed point: Time: 3.03 s, Ch1: 0.
Parsed point: Time: 3.04 s, Ch1: 1.
Parsed point: Time: 3.05 s, Ch1: 1.
```

SENDING TO THE DEVICE

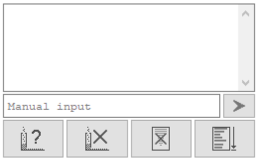
To send it to the device, just type the text in the text box and confirm by pressing enter or clicking the button. A line ending character of your choice is added to the end of the text, if one is selected.

The "plus" button can be used to add more lines. The bottom line is deleted after sending, the other lines keep the text for possible resending.



MANUAL INPUT FOR TESTING

Manual input can be displayed by checking the box on the settings page. Allows you to manually enter data into a text field and process it as if it were data from a device. The information for manual input also displays information about the settings loaded from the file. Characters that are not in basic ASCII may not process correctly (not suitable for testing binary data).



DATA TRANSFER PROTOCOL

On the right, it starts with a pair of **`$$`** followed by a letter indicating the message type:

- **`$$P`** adding a point to the graph
- **`$$C`** adding the whole channel to the chart
- **`$$L`** adding a logical channel to the chart
- **`$$B`** adding a logical point to the graph
- **`$$T`** dump to terminal
- **`$$S`** settings
- **`$$I`** listing of information messages
- **`$$W`** warning message
- **`$$X`** device error (displays a message and disconnects the device)
- **`$$E`** echo (sends the received text back to the device, to test the connection from the device side)
- **`$$A`** initial echo (like echo, but only responds once after connection)
- **`$$R`** file request
- **`$$F`** save to file
- **`$$Q`** QML terminal
- **`$$D`** QML terminal input
- **`$$V`** QML terminal variable
- **`$$U`** unknown (data is written to the terminal, if it does not contain \r, \n is changed to \r\n)

The letter indicating the message type is not case-sensitive.

HANDSHAKE

The recommended procedure is that the connected device has some reset command (e.g. **`reset`**) and periodically sends the **`$$Areset;`** message. If a new connection to the DataPlotter application occurs, a reset command will be sent to the device. It will respond by sending the initial settings (e.g., render the terminal).

LISTING TO THE TERMINAL

Data is written to a terminal supporting ANSI escape sequences. They are written out continuously as they come in. It is only terminated at the beginning of the next message. May not contain a **`$$`** sequence, one **`$`** may occur.

```
$$T\u001b[31;1mAAA\u001b[32;1mBBB\u001b[33;1mCCC\n\r
```

He writes `AAA` `BBB` `CCC` and lines up.

PRINTING INFORMATIVE WARNING AND ERROR MESSAGES

The information and warning text is displayed in the text field where messages about processed data are printed. Informational is marked in green, warning in red. The error message is displayed in a pop-up window and disconnects the port.

The error message is terminated by a semicolon. Information and warning messages are terminated at the beginning of the next message, and the semicolon is not used (so it can be included in the text). It must not contain the sequence **`$$`**. One **`$`** may occur.

```
$$IThis is information   $$WThis is a warning    $$XThis is an error;
```

SETTINGS

Sets chart and GUI parameters. A list of available settings is at the end of this document.

- The setting is of the form: **`identifier:value`**
- To set analog or math channel: **`ch:channel:numberChannel:identifier:value`**
- For setting logical channels **`log:numberLogics:identifier:value`**

Each setting is terminated by a semicolon. You can have multiple settings in one message. Identifiers are not case-sensitive. Numeric values are always written as a number, binary representation is not possible here.

Channels are numbered from 1, for math channels the numbers are 17, 18, 19. Logic channels are numbered 1 and 2, for the main logic channel (no number in the channel list) use number 3.

```
$$Svrange:100;
```

Sets the vertical range to 100.

LIST OF SETTING COMMANDS

Identifier	The meaning of the parameter being set	Type and range of value
baud	Baudrate	(number)
clearch	Clear channel	1...16
clearall	Clears all channels	(none)
clearlog	Deletes the logical channel (the one for direct addition)	(none)
haxis	Type of timeline	0/1
hlabel	Timeline label	(text)
hrange	Time range in rolling mode	0.001...1000000
hunit	Horizontal axis unit	(text)
noclickclr	List of background colours of characters that cannot be sent	(see chapter Interactive control)
clickclr	List of character background colours that can be sent	(see chapter Interactive control)
rstcmd	Send command after connection	(text)
vaxis	Show values on the vertical axis	0/1
vcenter	Vertical chart position	(value)
vlabel	Vertical axis label	(text)
vrange	Range of values	0.000001...1000000
vunit	Unit of values on the vertical axis	(text)
theme	Colour scheme	light/dark
trigline	Line mode showing trigger level	"on", "off", "auto"
trigch	The channel on which the trigger is	1...16
trigpos	Trigger value	(number)
xyclr	Color XY chart	“0,0,0”…”255,255,255”
ch:?:sty	Channel style (?=1...16 or 17...19 for math)	0~line...5~squareFilled
ch:?:clr	Channel color (?=1...16 or 17...19 for math)	“0,0,0”…”255,255,255”
log:?:sty	Logical channel style (?=1/2 = Logic1/2,?=3=Logic)	0~line...5~squareFilled
log:?:clr	Logic channel color (?=1/2 = Logic1/2, ?=3=Logic)	“0,0,0”…”255,255,255”

QML TERMINAL

The Qt framework allows you to create a GUI using the QtQuick environment and a QML script. This can be used to render a full GUI instead of a pseudo-graphical terminal.

You can use QtCreator (or a regular text editor) to create the script. A template with a description of its use can be obtained directly in the DataPlotter developer options window

SENDING A FILE FOR RENDERING

The content of the QML file is sent by the $$Q message. The data in the message is compressed and uses byte stuffing. To generate the message content, use the button on the New Terminal page.

DIRECT INPUT

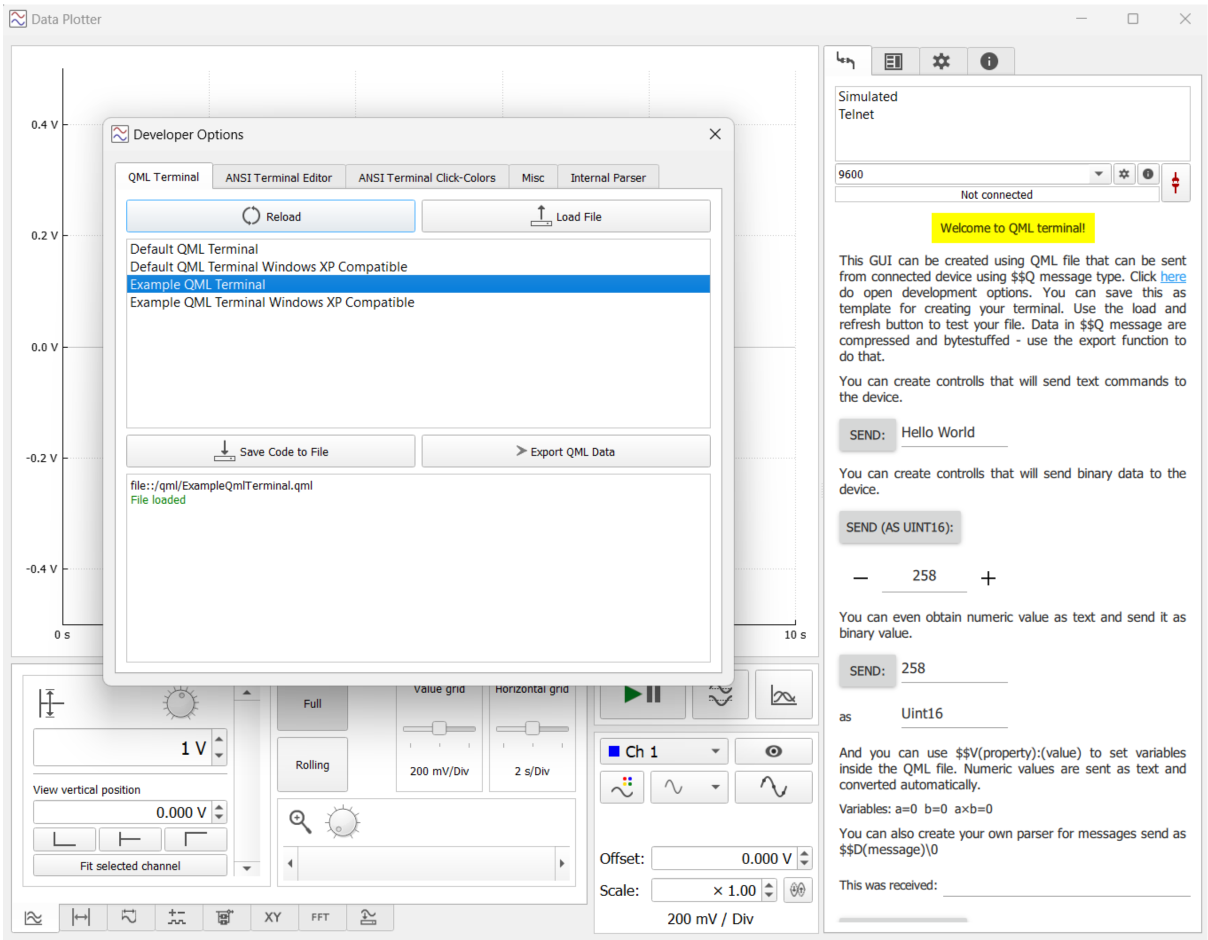
The message of this type is passed to the function for processing in QML code (the user can program the processing himself), see the sample file.

`$$D(data)\0`

CHANGE THE VALUE OF A VARIABLE

Variables (properties) in QML code can be easily set by the following messages. The values are always in text form and are processed according to the type of the variable (a number written as text will be treated as a "real" number if it is assigned to a variable of numeric type (int, real).

`$$V(name):(value);`



Default QML Terminal
Default QML Terminal Windows XP Compatible
Example QML Terminal
Example QML Terminal Windows XP Compatible

file::/qml/ExampleQmlTerminal.qml
File loaded

Simulated
Telnet

Welcome to QML terminal!

This GUI can be created using QML file that can be sent from connected device using $$Q message type. Click here do open development options. You can save this as template for creating your terminal. Use the load and refresh button to test your file. Data in $$Q message are compressed and bytestuffed - use the export function to do that.

You can create controlls that will send text commands to the device.

SEND: Hello World

You can create controlls that will send binary data to the device.

SEND (AS UINT16):

– 258 +

You can even obtain numeric value as text and send it as binary value.

SEND: 258

as Uint16

And you can use $$V(property):(value) to set variables inside the QML file. Numeric values are sent as text and converted automatically.

Variables: a=0 b=0 a×b=0

You can also create your own parser for messages send as $$D(message)\0

This was received:

SENDING TEXT FROM A FILE

The program can send text from a file to the serial port. The text can be divided into blocks of set length. A terminating character can be added to the end of the text. The file transfer is initiated by the device (microcontroller) sending a file request:

```
$$Rnew,(length),(end);
$$R(length),(end);
$$R;
```

- Keyword "new": prompts the user to select a file. If it is not present, it continues with the previous file
- Length: number of characters in one block. Can be replaced by keyword "all" which sends everything at once
- Ending: a character to be added after the last character of the text (can be omitted - no character):
  - "0": null
  - "EOT": character End of transmission (0x04)
  - "EOF": end of file character (depends on the platform, usually -1)
  - "SEMIC": semicolon
  - "DOLLAR": dollar ($)
  - "LF": new line (\n)
  - "CR": carriage return (\r)

The letter "s" can be added after the termination designation (e.g. "0s" "EOTs"), in which case the last block will be padded to its full length with this character (for "all" it has no effect).

EXAMPLES

```
$$Rnew;
```

Prompts the user to open the file.

```
$$Rnew,all;
```

It prompts the user to open the file and sends the whole file.

```
$$Rnew,64,0s;
```

Prompts the user to open the file, sends the first 64 characters. The last block will be padded with \0 (null) characters.

```
$$R;
```

Sends another block of the already opened file.

```
$$R64,0;
```

Sends the next block of an already opened file and overwrites the previous length and ending settings (the last character will be \0, but the last block can be less than 64 characters.

EXAMPLES OF USAGE

File: "Hello world 123456789" **`$$Rnew,8,0s;`** Sends: "Hello wo" (8 characters) **`$$R;`** Sends: "rld 1234" (8 characters) **`$$R;`** Sends: "56789\0\0\0" (8 characters) **`$$R;`** Sends: "\0\0\0\0\0\0\0\0" (8 characters)	File: "Hello world 123456789" **`$$Rnew,8,0;`** Sends: "Hello wo" (8 characters) **`$$R;`** Sends: "rld 1234" (8 characters) **`$$R;`** Sends: "56789\0" (6 characters) **`$$R;`** Sends: "\0" (1 character)
File: "Hello world 123456789" **`$$Rnew,32,0s;`** Sends: "Hello world 123456789\0\0\0\0\0\0\0\0\"... (32 characters)	File: "Hello world 123456789" **`$$Rnew,32,0s;`** Sends: "Hello world 123456789\0"
File: "Hello world 123456789" **`$$Rnew,all,0s;`** Sends: "Hello world 123456789\0"	File: "Hello world 123456789" **`$$Rnew,all,0;`** Sends: "Hello world 123456789\0"

WRITE TO FILE

Upon receipt of this message, the user will be prompted to select a file to save the contained text. The text in the message is terminated with the termination character \0

```
$$F(data)\0
```

When sending, note that if the function for sending to the serial port has an argument of the string (char*) type, it will not send the \0 character at the end of the string. It is necessary to use some function that has both a pointer and an array length as arguments.

POINT

Adds data to analog channels point by point in decimal or binary representation.

`$$P(time),(ch1),(ch2),(ch3);`

- Time:
  - Value (numeric or binary): time (x-coordinate) of the point
  - Special commands:
    - "-": sample index since connection
    - "-auto": time since connection
    - "-tod": time of day (seconds from midnight) (TOD = time of day)
- Ch1... Ch16 (maximum 16 values)
  - Value (numeric or binary): channel value at this point
  - Special commands:
    - "-": the channel has no value at this time

Consecutive binary values do not need to be separated by a comma.

EXAMPLES

NUMERIC NOTATION

`$$P123.00,1.10,2.20,3.30;`

At 123.00, channel 1 has a value of 1.10, channel 2 has a value of 2.20 and channel 3 has a value of 3.30.

`$$P123.00,1.10,-,3.30;`

At time 123.00, channel 1 has a value of 1.10, channel 2 has no value and channel 3 has a value of 3.30.

`$$P-,1.10,2.20,3.30;`

The time coordinate of this point is equal to the order of the point since connection (starting at time 0 and increasing by one for each subsequent point).

`$$P-auto,1.10,2.20,3.30; or $$P-tod,1.10,2.20,3.30;`

The time coordinate of this point is equal to the time since the connection (or time of day in the latter case) at which this point was received.

POINT WRITTEN IN BINARY

`$$PU2??U2??U2??U2??U2??;`

Time and three values in unsigned integer type. Note that it is not necessary to separate the values with a comma (but commas can be used).

COMBINED NOTATION

`$$PU2??U2??,123.00,U2??;`

Time and channels 1 and 3 have values of unsigned integer type, channel 2 has the value 123.00, note that the numeric value is separated by commas.

`$$PU2??U2??,-,U2??;`

Time and channels 1 and 3 have values of unsigned integer type, channel 2 is empty at this point.

`$$P-,U2??U2??U2???`

The time coordinate of this point is equal to the order of the point since connection (starting at time 0 and incremented by one for the next point).

CHANNEL

Adds the entire data set to one analog channel, the data is binary.

```
$$C(header);(data type)(data..........);
```

Depending on the data type, different header types can be used:

UNSIGNED INT

```
$$C(ch),(time step),(length);U?(data..........);
$$C(ch),(time step),(length),(bits),(max);U?(data..........);
$$C(ch),(time step),(length),(bits),(min),(max);U?(data..........);
$$C(ch),(time step),(length),(bits),(min),(max),(zero index);U?(data....);
```

SIGNED INT OR FLOATING PIONT

```
$$C(ch),(time step),(length);F?(data..........);
$$C(ch),(time step),(length),(zero index);F?(data..........);
```

- ❖ Ch: positive integer (numeric or binary): channel number (1 ... 16), or multiple numbers separated by '+'
- ❖ Time step: value (numeric or binary): time interval between consecutive samples
- ❖ Length: positive integer (number or binary): number of samples (not bytes) in this channel
- ❖ Bits: positive integer (numeric or binary): number of bits used in the value (to calculate min and max)
- ❖ Min: value (numeric or binary): values will be remapped so that 0 corresponds to this value
- ❖ Max: value (numeric or binary): values will be remapped so that 2^bits correspond to this value
- ❖ Zero index: positive integer or zero (numeric or binary): sample index corresponding to time 0. If omitted, the first sample (index 0) is at time 0. Useful for pretrigger.

Consecutive binary values do not need to be separated by a comma.

EXAMPLES

SIMPLE VARIANT WITH UNSIGNED INTEGER VALUES

```
$$C1,0.001,20;U2????????????????????????????????????????;
```

Channel 1 data, sample interval is 0.001 seconds (first sample is at time zero), channel has 20 samples in 16-bit unsigned integer (40 bytes after "U2").

VALUES IN UNSIGNED INTEGER WITH REMAPPING

```
$$C1,0.001,20,12,-1.5,1.5;U2????????????????????????????????????????;
```

Data for channel 1, sample interval is 0.001 seconds (first sample is at time zero), channel has 20 samples in 16-bit unsigned integer. The values are remapped so that a value of 4096 (2^12) corresponds to 1.5V and a value of 0 corresponds to -1.5.

VALUES IN FLOATING POINT

```
$$C1,0.001,10,5;F4????????????????????????????????????????;
```

Data for channel 1, sample interval is 0.001 seconds (first sample is at time zero), channel has 10 samples in 32-bit float. The sample with index 5 (counted from zero) is at time 0, the samples before it are at negative times.

MORE CHANNELS TO SKIP

```
$$C1+2+3+4,0.001,24;U2????????????????????????????????????????????????;
```

The data is for channels 1, 2, 3 and 4. The values are alternated in this order (the first pair of bytes is for channel 1, the second pair for channel 2...). The specified length is the number of samples of all channels together (in this example, each channel has 8 samples).

LOGIC CHANNEL

Adds the whole data set to the logical channels, the values are binary, of unsigned int type.

```
$$L(header);(data type)(data..........);
```

```
$$L(time step),(length);U?(data..........);
$$L(time step),(length),(bits);U?(data..........);
$$L(time step),(length),(bits),(zero index);U?(data..........);
```

- Time step: value (numeric or binary): time interval between consecutive samples
- Length: positive integer (number or binary): number of samples (not bytes) in this channel
- Bits: positive integer (numeric or binary): number of bits to be displayed (starting from LSB)
- Zero index: positive integer or zero (numeric or binary): sample index corresponding to time 0. If omitted, the first sample (index 0) is at time 0. Useful for pretrigger.

Consecutive binary values do not need to be separated by a comma.

EXAMPLES

```
$$L0.001.20;U2????????????????????????????????????????;
```

The interval between samples is 0.001 seconds (the first sample is at time zero), the channel has 20 samples in 16-bit unsigned integer (40 bytes after "U2"), all 16 bits are displayed.

```
$$L0.001,20,12;U2????????????????????????????????????????;
```

The interval between samples is 0.001 seconds (the first sample is at time zero), the channel has 20 samples in 16-bit unsigned integer, only the last 12 bits are displayed.

```
$$L0.001,20,16,10;U2????????????????????????????????????????;
```

The interval between samples is 0.001 seconds (the first sample is at time zero), the channel has 20 samples in 16-bit unsigned integer, all 16 bits are displayed. The sample with index 10 (counted from zero) is at time 0, the samples before it are at negative times.

LOGIC POINT

Adds data to the logic channels point by point.

`$$B(time),(value),(bits);`

- Time:
  - Value (numeric or binary): time (x-coordinate) of the point
  - Special commands:
    - "-": sample index since connection
    - "-auto": time since connection
    - "-tod": time of day (seconds from midnight)
- Value: value in unsigned integer
- Bits: positive integer (numeric or binary): number of bits to be displayed (starting from LSB)

Consecutive binary values do not need to be separated by a comma.

EXAMPLES

`$$B123.00,U2???;`
The time is 123.00, a 16-bit logical value.

`$$BU2??U2???;`
Time as unsigned integer. 16-bit logical value.

`$$BU2??U2??,12;`
Time as unsigned integer. 12-bit logical value.

`$$B-car,U2??;` or **`$$B-tod,U2??;`**
The time coordinate of this point is equal to the time since the connection (or time of day in the latter case) at which this point was received.

NUMERICAL VALUES

The decimal point is used.

```
123.45
```

The number can also be written in scientific notation

```
1.23e-3
```

It must always begin with a digit:

`1e-3`: correct

`e-3`: Wrong

EXAMPLE OF SENDING A NUMERIC VALUE

MBED:

```
serial.printf("$$P%u.0,%.3f,%.3f;",millis(),value1,value2);
```

BINARY VALUES

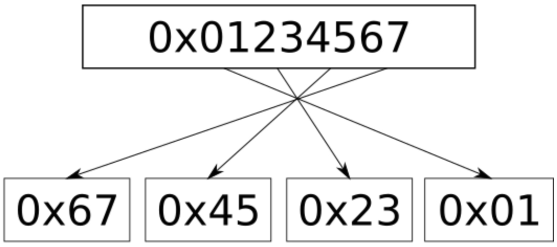
When sending data in binary format, it is necessary to specify the data type before the bytes of the number itself:

Little-endian		Big-endian	
u1	8-bit unsigned integer	**U1**	8-bit unsigned integer
u2	16-bit unsigned integer	**U2**	16-bit unsigned integer
u3	24-bit unsigned integer	**U3**	24-bit unsigned integer
u4	32-bit unsigned integer	**U4**	32-bit unsigned integer
i1	8-bit signed integer	**I1**	8-bit signed integer
i2	16-bit signed integer	**I2**	16-bit signed integer
i4	32-bit signed integer	**I4**	32-bit signed integer
f4	float	**F4**	float
f8	double	**F8**	double

LITTLE-ENDIAN AND BIG-ENDIAN

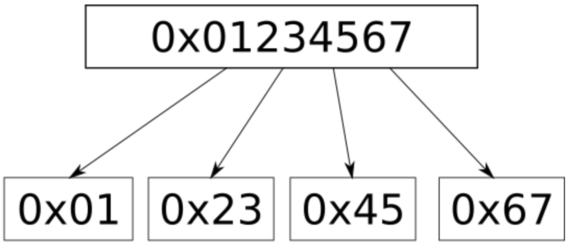
Most platforms are little endian

LITTLE-ENDIAN



At the lowest address in memory is a byte with the LSB of the number

BIG-ENDIAN



At the lowest address in memory is a byte with the MSB of the number

EXAMPLE OF SENDING A VALUE IN BINARY

MBED:

```
float value = 123.45;
char *value_bytes = (char *)&value;
serial.printf("f4");
serial.putc(value_bytes[0]);
serial.putc value_bytes[1]);
serial.putc(value_bytes[2]);
serial.putc(value_bytes[3]);
```

VALUES WITH UNIT

For example, if you want to send an integer value in mV, you can do it by adding the letter **m** before the data type identifier, the value sent with this prefix will be divided by one thousand.

Possible prefixes are T, G, M, k, h, D, d, c, m, u, p, f, a.

EXAMPLE

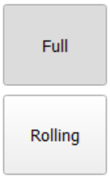
```
$$C1,uU2??,20;mU2????????????????????????????????????????;
```

The sample interval is a 16-bit unsigned integer in µs, the channel values are a 16-bit unsigned integer in mV.

CHART

Up to 16 analog channels, 3 math channels and 3 logic channel groups with a maximum of 32 bits can be displayed in the graph. Two of the logic channel groups are for converting integer analog input to logic channels (ADC bit display), the third is for direct logic input using `$$L` or `$$B` messages.

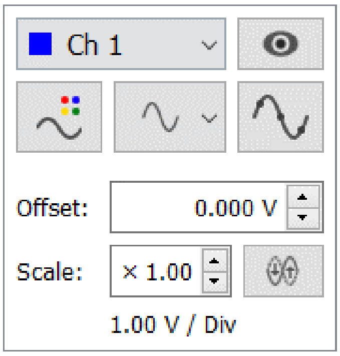
MODES



Fixed: displays the full time range of the received signal. Suitable for waveforms that are redrawn continuously over the same time span.

Rolling (sliding): only the section at the end is displayed, the chart is shifted to the left. Suitable for waveforms that grow further in time and older values remain on the graph.

CHANNEL SETTINGS



Ch 1

Offset: 0.000 V

Scale: × 1.00

1.00 V / Div

The channel to be set is selected at the top, its colour is displayed for easy identification, the colour can be changed. The display style can also be selected (line, dots), the channel can also be hidden.

For analog channels, the interpolation function can be activated, where the waveform is resampled to a higher sampling frequency. For mathematical channel settings, the quality can be adjusted.

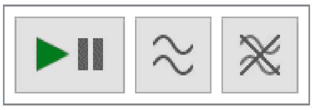
By default you can only select from the channels that are currently in use, this can be changed in the settings.

The offset, i.e. the vertical offset, is set in the middle field. If it is non-zero, a dashed line in the channel color will appear in the graph to show where the channel zero value is currently located. The offset can also be changed by dragging the mouse over the line that indicates the zero level of the channel.

Under offset, the vertical stretch of the channel is set (multiplying all values by the set number). The channel can also be inverted. Neither stretching nor inverting affects the values displayed by the cursors.

The bottom shows the channel scale (the difference of values corresponding to one grid step), this changes according to the grid setting, and the channel stretch.

CHART CONTROL



The pause button pauses (or starts the paused) graph. During the pause, new data continues to be processed and will be added to the chart when the pause ends.

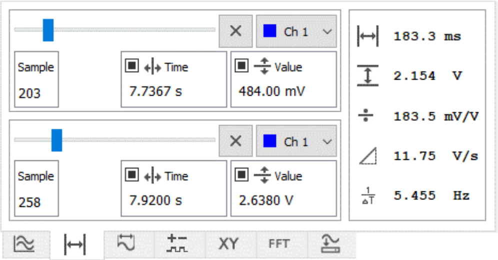
The middle button attempts to automatically set the range and use offsets to spread multiple channels over each other, can also be used for logical channels. In the settings tab you can choose to auto-set automatically when connected.

The button on the right resets the offset and sets the magnification to 1 for all channels.

CURSORS

There are two pairs of vertical and horizontal cursors. Each cursor has three modes

- Perched on a sample
- Free
- Hidden


Ch 1
Sample 203 ☑ Time 7.7367 s ☑ Value 484.00 mV
Ch 1
Sample 258 ☑ Time 7.9200 s ☑ Value 2.6380 V
183.3 ms
2.154 V
183.5 mV/V
11.75 V/s
5.455 Hz

In free mode, the time or value (voltage) at which the time or voltage cursor should be set. In sample bound mode, the slider sets the sample on which the time cursor is positioned, if the voltage cursor is also in this mode, it is positioned on the value of that sample.

The time cursor can only be in free mode for analog channels (not logic and FFT). Only the time cursor is available for the logic channel.

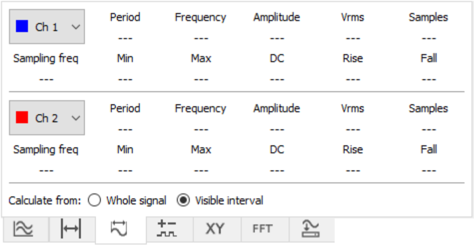
The cursors can also be controlled with the mouse. Clicking on the graph moves the pair of cursors (time and voltage) to the location (nearest sample) of the channel clicked, left button for the first, right for the second). The cursors can then be dragged with the mouse. If the voltage cursor is dragged, it switches to free mode. The time cursor remains in its original mode when dragged.

These cursors are common to the main chart and FFT, XY mode has separate cursors on its page.

The value of the voltage cursor is relative to the offset of the selected channel, but independent of the magnification setting (magnification only stretches the graph, but does not affect the values) and the channel inversion (if the channel is inverted, the values are positive downwards). The cursors can also be set to absolute mode, where they display values relative to the axes of the graph.

MEASUREMENTS

This page displays data about the measured signal.



Ch 1	Period	Frequency	Amplitude	Vrms	Samples
	---	---	---	---	---
Sampling freq	Min	Max	DC	Rise	Fall
---	---	---	---	---	---
Ch 2	Period	Frequency	Amplitude	Vrms	Samples
	---	---	---	---	---
Sampling freq	Min	Max	DC	Rise	Fall
---	---	---	---	---	---

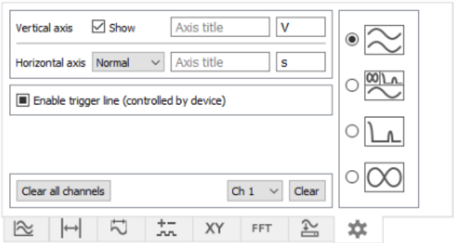
Calculate from: ○ Whole signal ● Visible interval

Two channels can be selected (In addition to the basic channels, the calculation channel can also be selected) for which the data will be calculated.

The values can be calculated from the whole channel (Whole signal) or from the range that is displayed in the graph (Visible interval). DC and RMS values are (if possible) calculated over the whole number of periods.

The rising and falling edge time is always calculated on the last period of the signal/segment displayed. If the value is displayed with a "less than" sign, the edge is shorter than the sample interval.

CHART SETTINGS


Vertical axis ☑ Show Axis title V
Horizontal axis Normal Axis title s
☑ Enable trigger line (controlled by device)
Clear all channels Ch 1 Clear
● ○ ○ ○

You can set axis labels and the unit of values on the axis (enter the unit as a base without prefix, prefixes are added automatically). You can show or hide the vertical axis (if there are multiple runs above each other, the numbers on the axis are irrelevant). The horizontal axis can also be hidden. If the unit is "time" the time is automatically displayed in seconds or as MM:SS or HH:MM:SS.

The trigger line is used to display the trigger level using commands from the device (see the settings table) and has three modes: unchecked - not displayed, checked - always displayed, square - displayed temporarily when changed

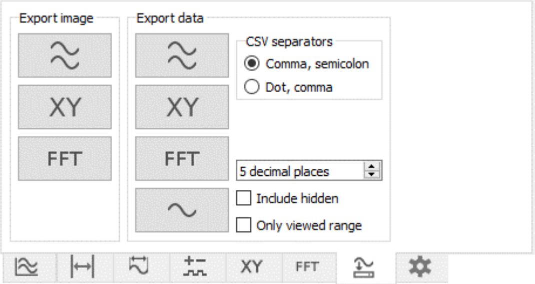
You can set the chart display mode (main, all, XY, FFT). It switches between main and all automatically.

EXPORT

You can export one selected channel, or all, or XY channel to a CSV file.

Depending on your system settings, Excel uses either a decimal point or a comma, so you must select the correct separator type to load the file correctly. The options are:

- decimal point, separated by a comma
- decimal point, separated by semicolon


Export image

Export data

CSV separators

● Comma, semicolon

○ Dot, comma

5 decimal places

□ Include hidden

□ Only viewed range

If "Include hidden" is checked, hidden channels will be exported as well. If "Only viewed range" is checked, only the currently viewed range will be exported (according to the range on the horizontal axis, vertically is not limited).

It is also possible to save the chart as a PNG image.

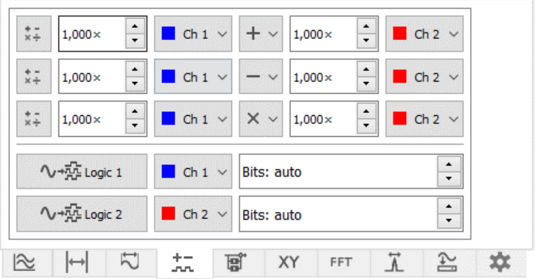
After clicking the button, the user will be asked if the data should be saved to a file or copied to the clipboard (for Ctrl+V). In the case of copying CSV to clipboard, a tab is used as a separator (works for pasting into Excel and similar programs).

CALCULATIONS AND LOGIC CHANNELS

Channels can be added, subtracted, multiplied, divided. It is also possible to select a constant value instead of a channel.

The calculation is activated by clicking on the button of the respective channel.

Channels sent in full channel mode with conversion (or as integer values) can also be displayed as a logical channel. If the bit count is set to automatic, the bit count specified in the message header is used.


1,000× 🟦 Ch 1 + 1,000× 🟥 Ch 2

1,000× 🟦 Ch 1 − 1,000× 🟥 Ch 2

1,000× 🟦 Ch 1 × 1,000× 🟥 Ch 2

Logic 1 🟦 Ch 1 Bits: auto

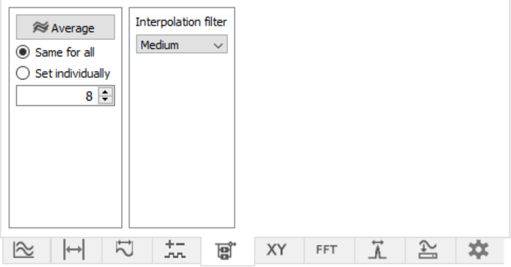
Logic 2 🟥 Ch 2 Bits: auto

After activation, the mathematical channel is calculated backwards for all previous points (if data is added point by point), logical channels are processed only for newly arrived data (not displayed if the chart is paused).

AVERAGING

It is possible to average the channel values. Averaging is activated by a button, the number of channels/points to average can be set for all the same or for each individually. For data added in whole channels (**$$C**), samples of the most recent waveforms are averaged. For adding by points (**$$P**), this acts as a moving average.

This page also sets the filter used for channel interpolation.


Average

● Same for all

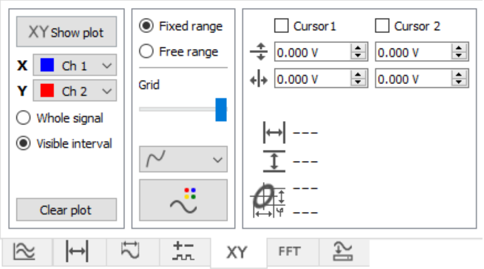
○ Set individually

8

Interpolation filter

Medium

X-Y MODE



The X-Y mode chart is displayed in a separate chart (it is automatically displayed when this mode is enabled and hidden when it is disabled).

In addition to the basic channels, the calculation channel (Math) can also be selected.

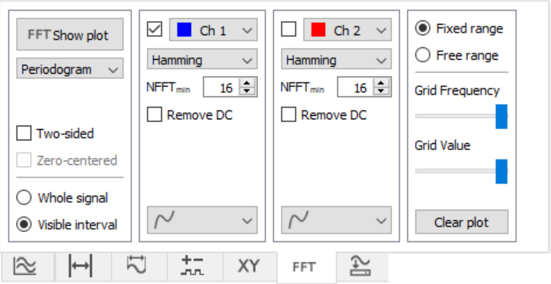
Fixed or free range can be selected, in free mode you can pan and zoom with the mouse.

You can also set the grid step and change the chart display (line or points) and chart color.

The XY graph can be calculated from the whole channels (Whole signal), or from the range that is displayed in the main graph (Visible interval).

Two pairs of cursors can be used in the XY chart. They can be controlled with the mouse in the same way as the cursors in the main chart.

FFT



The spectrum is displayed in a separate chart (it is automatically displayed when this mode is enabled and hidden when it is disabled). It is possible to display the spectrum of two waveforms simultaneously.

In addition to the basic channels, the calculation channel (Math) can also be selected.

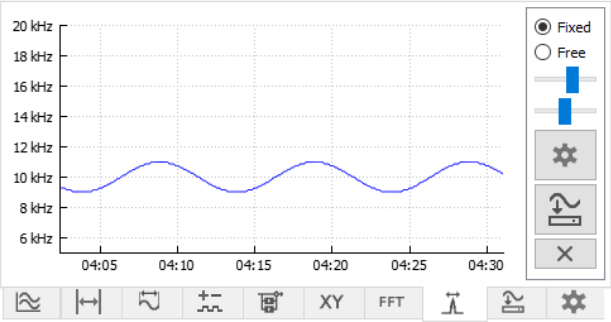
Fixed or free range can be selected, in free mode you can pan and zoom with the mouse.

It is also possible to set the grid step and change the graph display (line or points), the color is selected automatically according to the graph from which the spectrum is calculated.

The FFT graph can be calculated from the whole channel (Whole signal) or from the range that is displayed in the main graph (Visible interval).

Three types of calculation can be selected: spectrum (linear), periodogram in dB and periodogram calculated using the Welch method.

TIME DEPENDENCE OF FREQUENCY



This graph shows the position of the highest value in the FFT plot as a function of time.

It is recommended to remove the DC component in the FFT settings and enable zero padding. These settings are applied automatically when the gear icon button is pressed.

TERMINAL

The terminal allows text output and supports ANSI escape sequences, which allows you to create a pseudographic user interface in the terminal for clear display of measured values, etc.

The minimum width of the terminal is 14 characters. If the width is larger, it is automatically enlarged. The font size is adjusted to the width to make the content completely visible. The terminal can be scrolled vertically with the mouse wheel.

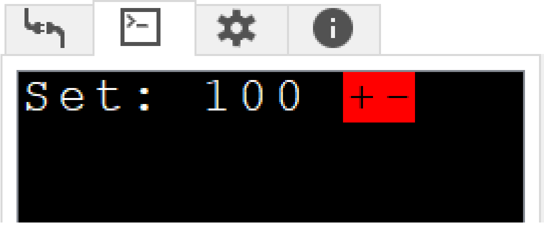
For the terminal to be functional, the "Default QML Terminal" must be loaded, for combination with a custom QML terminal this terminal can be inserted as an object in the QML code.

INTERACTIVE CONTROLS

By clicking on a letter (character) in the terminal, this character is sent to the device, thus a menu can be created for controlling the device by clicking the mouse.

By default, you cannot send a character with a black background (assumed to be a label). Set the list of disabled background colors using the **noclickclr** command (or the negated **clickclr** list), followed by a list of ANSI color sequences. The sequence need not contain **\e[** and **m**. Because the command is terminated by a semicolon, semicolons cannot be used in the list, so they are replaced by periods:

$$Snoclickclr:40,41.1,48.5.34;

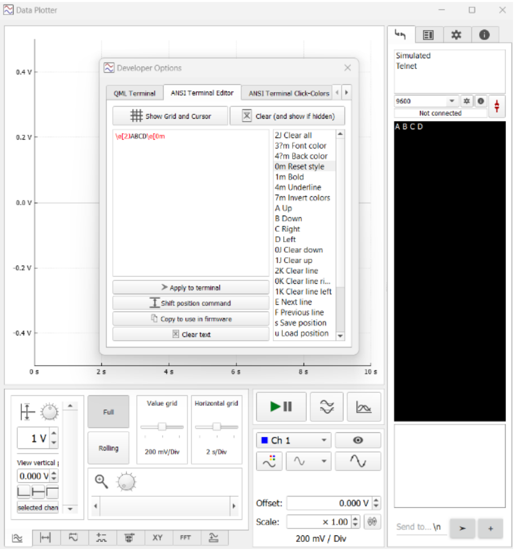


```
Set: 100 +-
```

EXAMPLE OF CREATING AN INTERACTIVE MENU

By clicking on **+** or **-**, the **+** or **-** character will be sent to the device, which can process it as a command to increase or decrease the value setting.

DESIGN AND DEBUGGING



```
\e[2JABCD\e[0m
```

This mode allows you to manually enter text into the terminal, including entering escape sequences and control characters selected from a list. The text entered into the text box can be output to the terminal by clicking on the button. It can also be copied to the clipboard for subsequent use in the device firmware.

The terminal displays the cursor position in this mode. Clicking in the terminal moves the cursor to the appropriate position (and the corresponding cursor move command is added to the text box).

UTF-8 characters can also be written to the terminal if you enter a non-ASCII character in the text field, after copying it to the clipboard with the copy button it is replaced by a sequence for writing this character as a text string. For example, the letter **á** will be copied as **\xc3""\xa1""**.