ConTest

User Guide

Program Version 1.3

Table of Contents

1. Overview

ConTest is a PC application designed to help development of embedded systems. ConTest is able to create dedicated **CON**trol and **TEST** configurations to the user's requirements using UART messages from the user's (target) system. This can be typically either a microprocessor or FPGA.

The only requirement of the target system is that it has a serial (UART) port that can be connected to the outside world. If no dedicated serial port is available, two digital IO pins can be used together with a 'soft' serial port. This can be a very simple implementation, as no flow-control is required, just a transmit data pin and a receive data pin.

If two-way communication is not needed, ConTest will work happily as either a receiver (showing data from the target) or transmitter (controlling operation of the target), and so only one data pin will be required. Of course, a common ground connection will still be needed, so the minimum external connections will be just two pins.

In the simplest implementation, ConTest can be used to replace a 'traditional' terminal program, such as HyperTerminal, and display serial messages as they are sent from the target. No special message formatting is required, and the application opens in this mode.

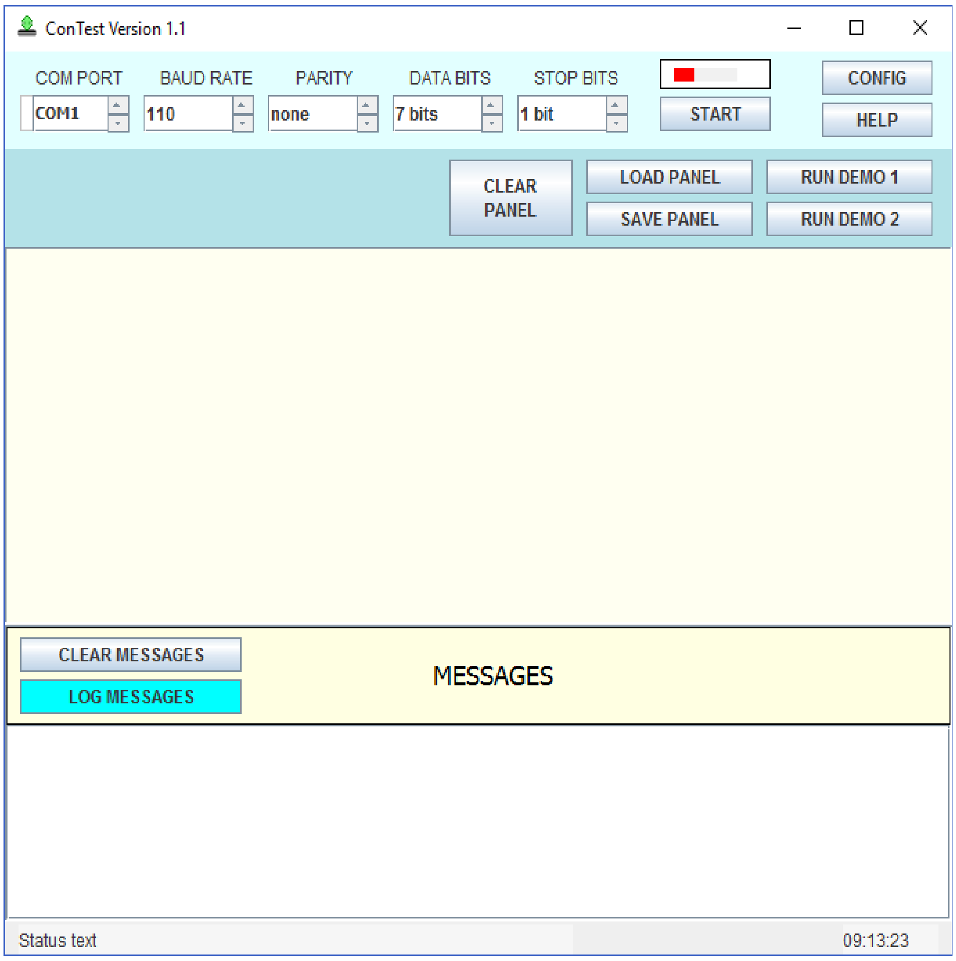
Often, when developing systems, a particular set of variables or values requires monitoring. This can be achieved using a terminal window, but this rapidly becomes difficult to manage when several variables are involved, together with other general status information and error messages. This is where the power of ConTest comes in. Through simple formatted ascii message strings, many types of controls and ways of displaying data are easily implemented on the ConTest display panel, arranged to the user's requirements. These configurations can be used as simple development aids, or as more permanent control panels for long-term control of the target system, with no programming of the PC required.

Examples of the type of controls possible can be seen in the integral demo screens which can be accessed by simple selection from the button on the application. These require no external hardware or set-up to run.

ConTest features a data-logging and analysis tool, where all messages can be collected, time-stamped, and stored to a text file. Data can be collected over extended time periods if required, in order to capture infrequent events.

2. Getting Started

2.1. *Initial Window*



COM1, 110, none, 7 bits, 1 bit

Status text 09:13:23

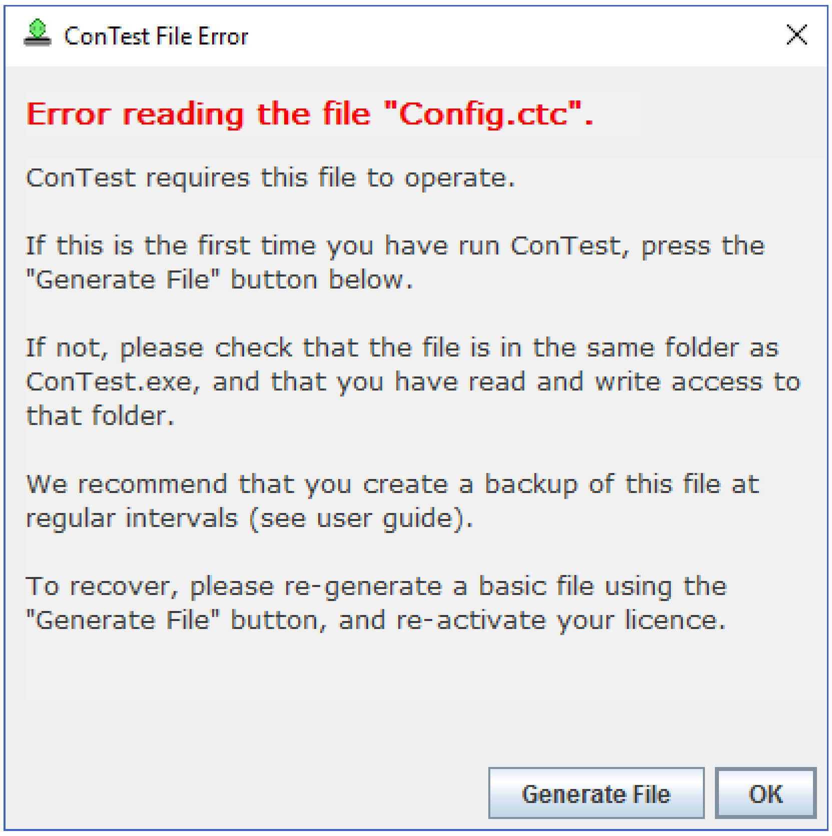
The initial window as shown above is presented when the program first runs, before anything is configured. The window is divided vertically into the following panels:

- Serial Port Settings and controls
- Program controls
- User control panel
- Received messages controls
- Received messages window
- Program status and message bar

2.2. *Configuration File*

ConTest uses an important configuration file, **Config.ctc**, which is included with other downloaded files, and must be present in the same folder as the main program. This is a text file, and so can be viewed in any text editor. THIS FILE MUST NOT BE ALTERED MANUALLY as it contains important information and encrypted license data. Once a full license for this product is obtained, it is strongly recommended that this file is backed-up, to guard against file corruption.

If a problem is found with this file, the screen below will appear. If this happens, close the program and copy the backup file into the folder containing the main program. As a last resort, a clean version of the file can be re-generated from the error screen. Please note, however, that this will revert the program to a trial license, and previous license information will be lost. (A full license can be recovered by re-entering the original Activation Code, supplied when the license was purchased.)



Error reading the file "Config.ctc".

ConTest requires this file to operate.

If this is the first time you have run ConTest, press the "Generate File" button below.

If not, please check that the file is in the same folder as ConTest.exe, and that you have read and write access to that folder.

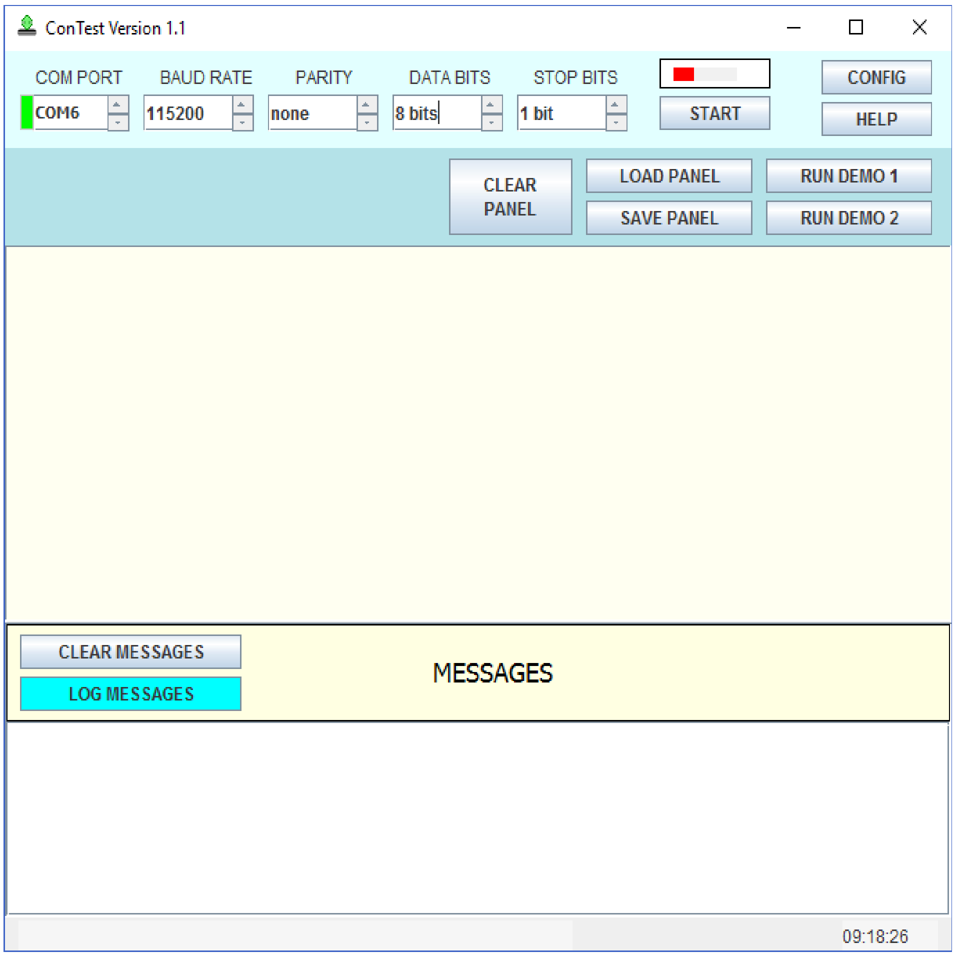
We recommend that you create a backup of this file at regular intervals (see user guide).

To recover, please re-generate a basic file using the "Generate File" button, and re-activate your licence.

2.3. Setting up Serial Port Parameters

In order to receive data from the target system, the serial port parameters first need to be configured to match the target, using the drop-down selection options. Once set, this is permanently visible on the screen, not hidden away in some setup menu. The data is automatically saved and re-loaded the next time the program starts so the user always has the latest configuration.

For more information, please see section 3.1 Serial Port Operation



2.4. *Checking the Serial Link*

Once the serial parameters have been set, click on the STOP/START button to begin receiving serial data. Trigger the target system to generate a serial ASCII message, which should appear in the Received Messages window.

If this does not happen, check the serial parameters match the target, and ensure the bar beside the COM PORT control is green, indicating the COM PORT is active. Also check the physical connection to the target, and ensure your target's signalling voltage matches the serial cable you are using.

Please note that a standard RS232 serial interface cable uses high voltage signal lines (typically 12V or more) and so cannot be connected directly to target boards operating at low voltages. Examples of physical connections are given in section 5.

3. Program Functions

3.1. Serial Port Operation

Communication via the serial port is very straightforward. First set up the standard parameters for serial operation. The following notes will guide you.

- **COM PORT** Usually, when a USB to Serial lead is plugged into a PC, it is automatically detected and allocated a COM port number. Active COM ports are indicated by a green bar beside the selected port. Whenever the corresponding lead is plugged in the bar should remain green while the connection is active. If this is not the case, either the cable is not being recognised by the PC, or the wrong port number has been selected.
- **BAUD RATE** This is a list of the standard baud rates available. Please ensure the baud rate on the target system corresponds to the value selected.
- **PARITY** This can be even, odd, or none. Select none if parity is not being set by the target, or if you are unsure.
- **DATA BITS** Normally, 7 or 8 data bits are used. This value must match the target system.
- **STOP BITS** One stop bit is usually all that is required. If the target needs two bits, this can be selected.

The button to the right of the drop-down options is used to start, stop, and pause reception and transmission of serial messages.

Immediately above this button is the activity gauge. This indicates serial activity on the port and consists of two parts. A red bar moves from left to right one pixel for each message received, and so indicates the rate of message reception. To the right of this, a blue rectangle will light for approximately three seconds if a single message is received, or stay on if subsequent messages are seen.

3.2. Configuration Options

Show Sent Messages

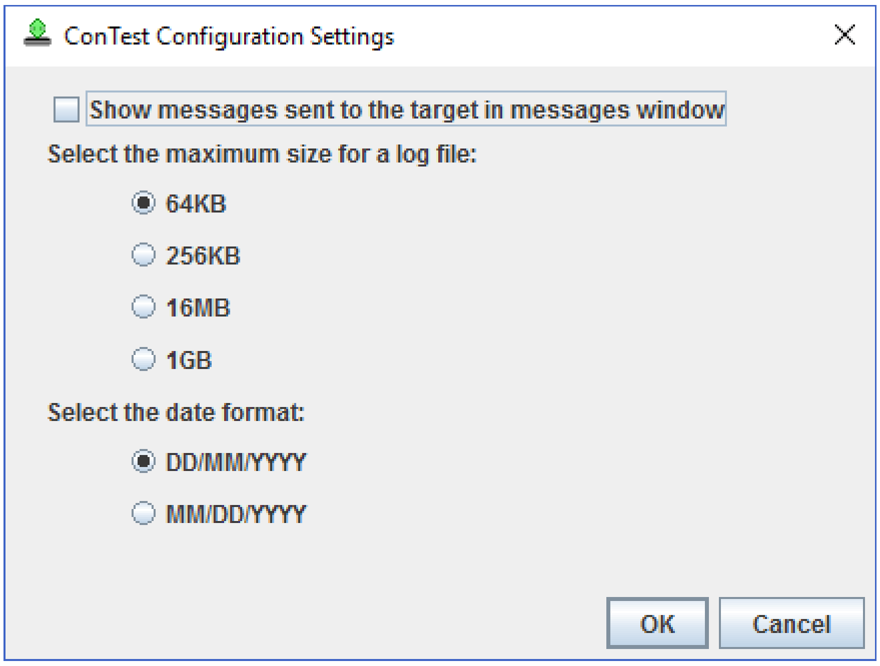
When checked, messages sent to the target will also be displayed on the MESSAGES window, and logged with the receive data.

Select Maximum Log File Size

The maximum log file size can be set with this option. See 3.7. Message Logging.

Select Date Format

The date format used in message logging can be set day-month-year or month-day-year as required.


ConTest Configuration Settings

☐ Show messages sent to the target in messages window

Select the maximum size for a log file:

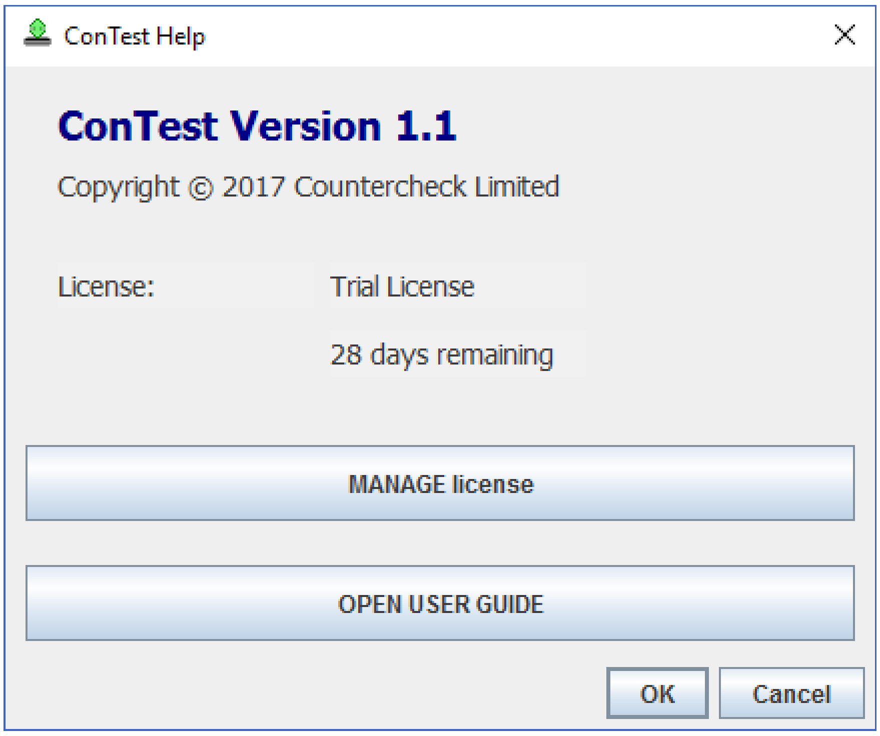
- [✓ 64KB]
- 256KB
- 16MB
- 1GB

Select the date format:

- [✓ DD/MM/YYYY]
- MM/DD/YYYY

3.3. Help Menu

The help menu shows basic program and license information, and includes two buttons, MANAGE LICENSE and OPEN USER GUIDE.


ConTest Help

ConTest Version 1.1

Copyright © 2017 Countercheck Limited

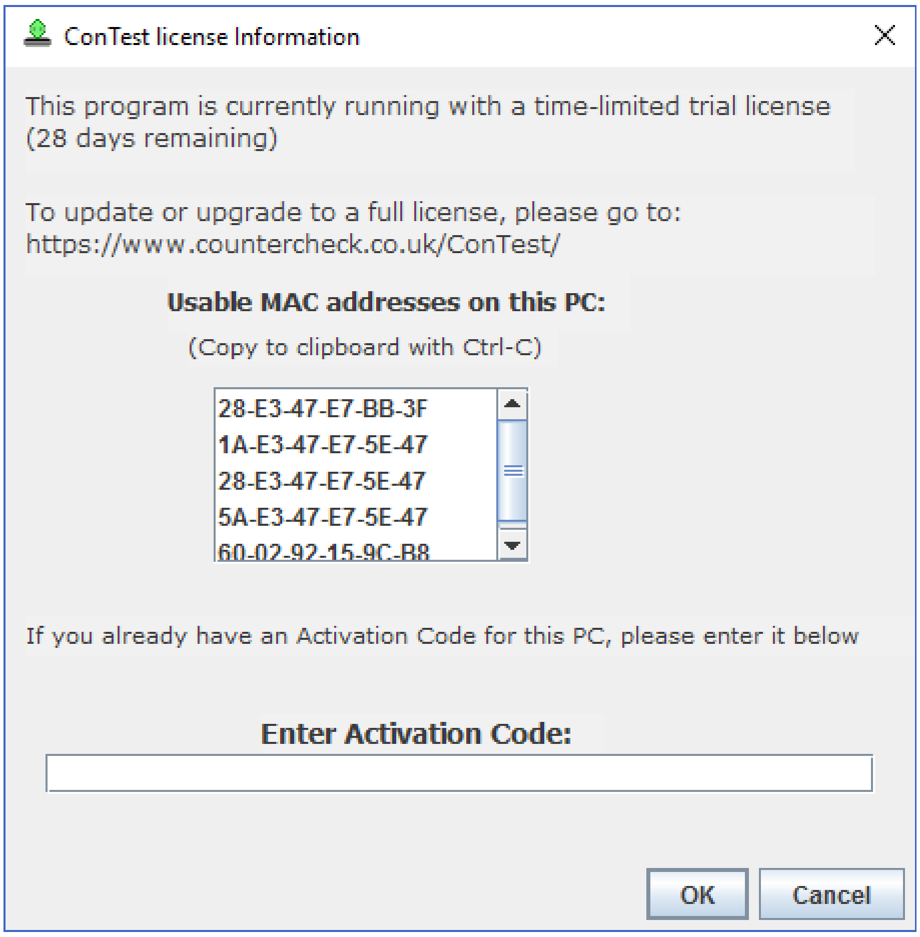
License:	Trial License
	28 days remaining

MANAGE LICENSE

This screen shows the current license type, and the URL to obtain a full license.

ConTest generates licenses linked to the specific PC the program runs on. This is done using one of the available MAC address found on the PC. To simplify the process of obtaining a license, a list of MAC addresses found on the PC is shown on this screen. All the addresses are unique to hardware installed on each PC, and any one may be used.

Please note that this list may include the MAC address of some external devices which may have been active when the program was run (e.g. WiFi dongles). If a license is created using the address of such an external device, the program will only run when that device is present. This gives a method by which one license may be used on several PCs, by plugging the dongle into the required PC.



This program is currently running with a time-limited trial license (28 days remaining)

To update or upgrade to a full license, please go to:
https://www.countercheck.co.uk/ConTest/

Usable MAC addresses on this PC:

(Copy to clipboard with Ctrl-C)

```
28-E3-47-E7-BB-3F
1A-E3-47-E7-5E-47
28-E3-47-E7-5E-47
5A-E3-47-E7-5E-47
60-02-92-15-9C-B8
```

If you already have an Activation Code for this PC, please enter it below

Enter Activation Code:

OPEN USER GUIDE

Pressing this button automatically opens this document on-screen as a .pdf file.

3.4. User Window Panes

The ConTest window is split into two re-sizeable, application-specific panes. The upper pane shows panel components arranged in accordance with the selected user panel setup (see section 3.10. Generating ConTest Panels) and a lower pane showing all messages not intended for the upper pane.

3.5. ConTest Specific and General Messages

When ConTest receives a message from the target, it interprets the message according to the content. If the message can be resolved into a ConTest message, then it is passed to the upper pane for decoding. All ConTest messages start with the character sequence “$$/” (without quotes), followed by a command sequence. An example of a valid ConTest message is shown below.

$$/ SIZA 700 800

This example sets the size of the application window to 700 pixels by 800 pixels.

Any message received which does not start with $$/, or has some other formatting error, is printed on the lower pane exactly as received. This can be any message the target wishes to display to the user.

ConTest specific messages fall into two categories, panel components and component data. A panel component is an item such as a frame to arrange other components into groups, or something which displays a data value or set of values, such as a text box or a graph. A collection of panel components which form the functions displayed is called a panel.

Panel components are used to create the “look” of the user (upper) panel, while the component data populates the panel with “live” data.

Both categories of messages can be sent from the target board. This makes the target self contained, but requires all the panel setup information to reside in code, on the target. This may be a problem if memory on the target is limited.

Alternatively, the command sequences can be created offline and stored in a simple text file, using any text editor. These can be loaded into ConTest as required, making it simple to switch to a different 'view' of the same data, or quickly change the parameters being monitored.

The text file must consist of a sequence of ConTest commands as described in sections 3.9 and 4.

3.6. Panel Functions

The panel function buttons (on the blue control bar) are used to manage ConTest user panels. Two demonstration panels are available to illustrate the capabilities of ConTest. These run with simulated data and so do not require 'live' serial messages.

LOAD PANEL

A previously created panel stored on the PC can be retrieved using this button. A pop-up menu appears enabling the user to navigate to the required text file containing commands to create the required panel.

SAVE PANEL

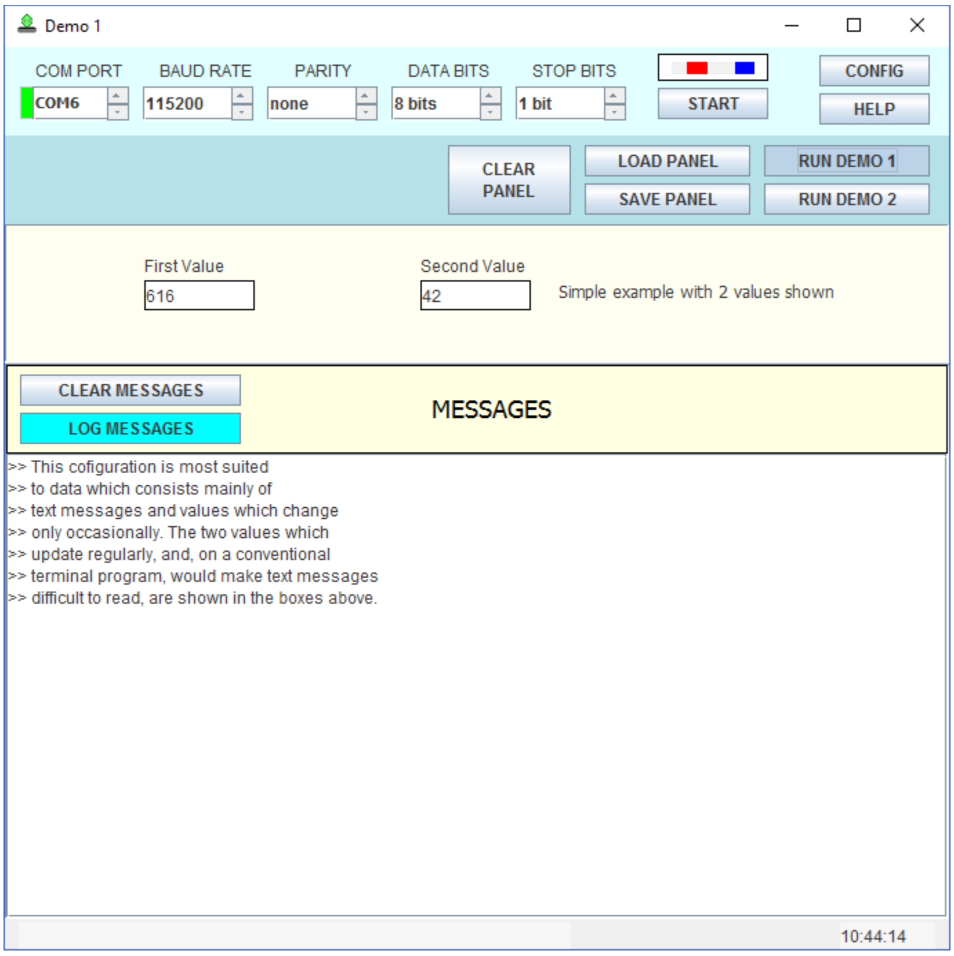
This will save the panel currently on screen to a text file.

CLEAR PANEL

This will remove all panel items currently on the screen.

RUN DEMO 1

When pressed, a simple demonstration program is loaded and begins running immediately. This demonstrates the use of ConTest with minimal set-up effort, and is best suited to situations where asynchronous messages are sent from the target in response to internal or external events.



COM PORT COM6, BAUD RATE 115200, PARITY none, DATA BITS 8 bits, STOP BITS 1 bit

First Value 616

Second Value 42

Simple example with 2 values shown

```
>> This cofiguration is most suited
>> to data which consists mainly of
>> text messages and values which change
>> only occasionally. The two values which
>> update regularly, and, on a conventional
>> terminal program, would make text messages
>> difficult to read, are shown in the boxes above.
```

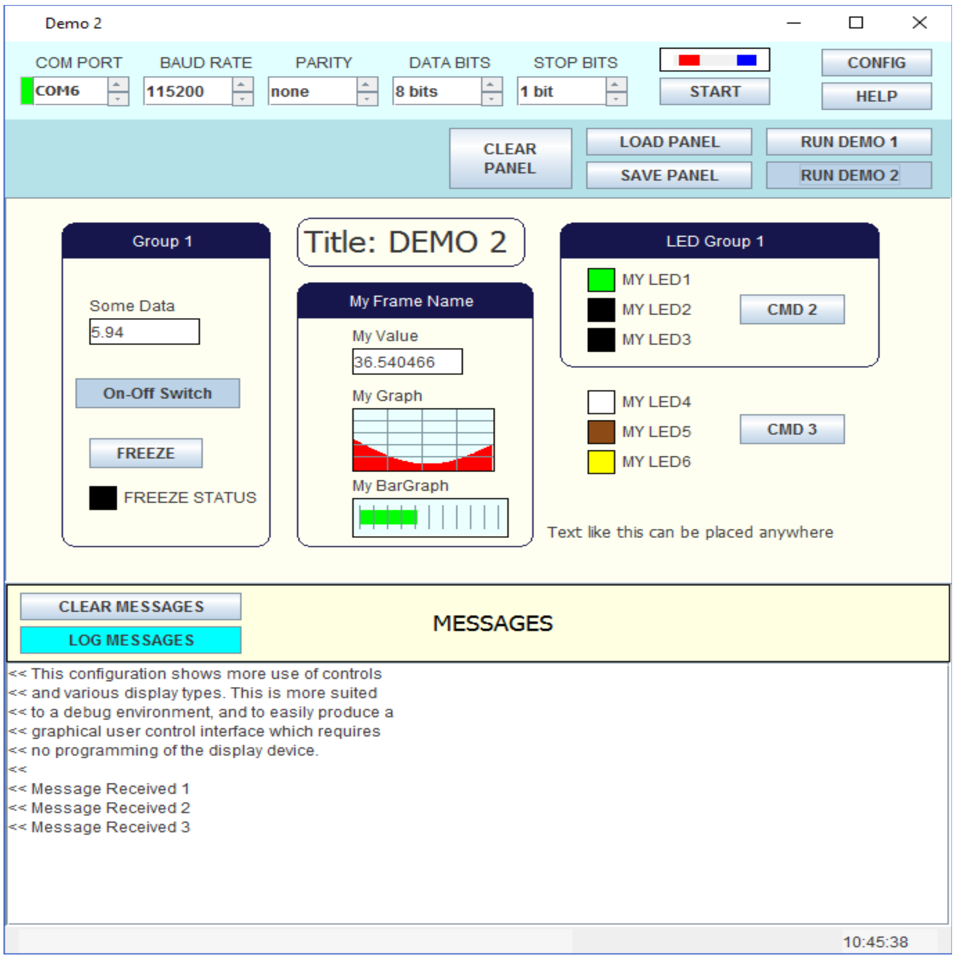
10:44:14

RUN DEMO 2

Demo 2 is a more comprehensive demonstration of the capabilities of ConTest and illustrates how target variables of different types may be handled.

Target variables are shown as current values and bar graphs, and long time trends in the form of a value-time graph.

Switches are shown which can send messages back to the target. On this demonstration, the switches start and stop updates of various values.



```
<< This configuration shows more use of controls
<< and various display types. This is more suited
<< to a debug environment, and to easily produce a
<< graphical user control interface which requires
<< no programming of the display device.
<<
<< Message Received 1
<< Message Received 2
<< Message Received 3
```

3.7. Message Logging

All messages received from, and sent to, the target can be recorded for later analysis in a log file.

To begin logging the data, click on LOG MESSAGES. This will bring up a dialog to select the name and location of the log file. The only restriction is that the user has to have write access to the file created. Any name/extension can be used, but the file will always be saved as a simple text file.

Once the file is created, all subsequent messages will be stored, until the log is closed (using the same button, which is now labelled CLOSE LOG). A warning will be generated if the data exceeds the maximum file size selected.

Each line corresponds to a message to or from the target. The line starts with the date and time stamp of the message, followed by "<<" for messages received from the target, or ">>" for messages sent to the target. Finally, the message content is displayed.

Note that message content can be either ConTest formatted messages (starting with $$/), or free-form text.

3.8. ConTest Command Format

All ConTest commands have the following format:

$$/<separator>XXXX<separator>p1<separator>p2...

where:

$$/	- ConTest format identifier
XXXX	- ConTest Command – see below
p1, p2, etc	- extra parameters for the specific command
<separator>	- either a **space**, or any of **,** **:** **** **_**(underscore)

3.9. ConTest Panel Complete Command List

The following table lists all the available commands interpreted by ConTest.

Commands in yellow are setup commands, and typically are sent on the serial port once at initialisation, or contained in a panel file.

Commands in gray populate the individual components previously defined by the setup commands, and are sent as many times as required to change the information displayed.

A ConTest panel is composed of one or more components, which together form the user's interface for controlling and monitoring the operation of the target system.

ConTest components are created and manipulated using sequences of commands as summarised in the following table. Full details of each function is described in section 4 ConTest Command Reference .

Command	Function	Type	Parameters				
			p1	p2	p3	p4	p5
BARA	Activate Bar Graph	S	identifier	x-offset	y-offset		
BARD	Bar Graph Data	D	identifier	value			
BARL	Bar Graph Label	S	identifier	string			
BTNA	Activate Button	S	identifier	x-offset	y-offset		
BTND	Set Button Data	S	identifier	string			
BTNL	Button Label	S	identifier	string			
FRAA	Activate Frame	S	identifier	x-offset	y-offset	width	height
FRAL	Frame Label	S	identifier	string			
GPHA	Activate Time Graph	S	identifier	x-offset	y-offset		
GPHD	Graph Data	D	identifier	value			
GPHL	Graph Label	S	identifier	string			
INIT	Initialise Panel	S					
LEDA	Activate LED	S	identifier	x-offset	y-offset		
LEDD	LED data	D	identifier	colour			
LEDL	LED label	S	identifier	string			
SIZA	ConTest Window Size	S	width	height			
SIZM	Message Pane Size	S	height				
SVLA	Activate Send Value	S	identifier	x-offset	y-offset		
SVLL	Send Value Label	S	identifier	string			
SVLD	Send Value PrefixData	S	identifier	string			
SWD1	Switch Data 1	S	identifier	string			
SWD2	Switch Data 2	S	identifier	string			
SWTA	Activate Switch	S	identifier	state	x-offset	y-offset	
SWTL	Switch Label	S	identifier	string			
TITL	Panel Title	S	string				
TXTA	Activate Text String	S	identifier	x-offset	y-offset	height	
TXTL	Text Label	S	identifier	string			
VALA	Activate Value Box	S	identifier	x-offset	y-offset		
VALD	Value Data	D	identifier	value			
VALL	Value Label	S	identifier	string			

In the above table, the ***Type*** column is used to differentiate between setup commands (S) and data commands (D).

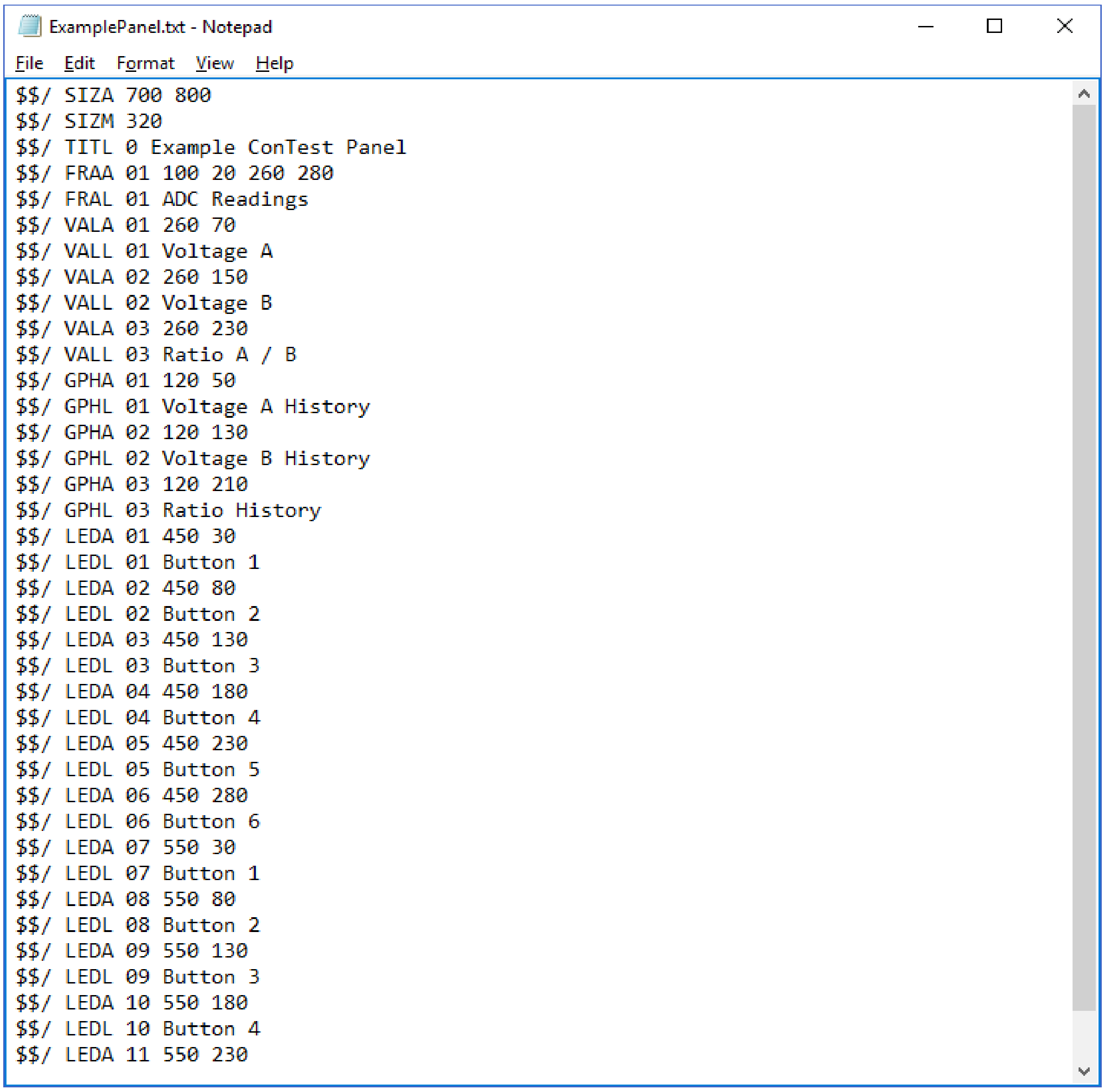
Identifier is a two-digit number, 01 to 99, which is unique to each component, and allocated when the component is activated. Subsequent commands using this identifier refer to the specific instance previously allocated. Setting a label, for instance, to an identifier not previously activated, will result in an error. Re-using a previously activated identifier will also result in an error. Note that the same identifier may be used for different component types, i.e. button 01 and LED 01 are different.

x-offset and ***y-offset*** define the horizontal and vertical positions that a component is drawn on the panel, relative to the top left hand corner of the panel pane. All values are counted in pixels on the PC screen. Note that the reference system is independent of any frames drawn, i.e. components and frames are all drawn on the same 'flat' pane. It is up to the user to ensure components do not overlap, and the overall panel looks acceptable.

3.10. Generating ConTest Panels

ConTest panels consist of a sequence of ConTest commands either received on the serial port, or contained in a text file which is read in. Either method can be used, or a mixture of the two.

Normally, the size information would be first in the sequence, followed by titles and any panels used, and finally the individual controls (buttons) and indicators would be placed on the panel. When the commands are read from a text file, they are executed line by line. Therefore, any command which places a component outside the default pane size will be ignored, even if the size is altered later.
Comment lines may be inserted in the text file, but must begin with the character sequence “//*”



```
$$/ SIZA 700 800
$$/ SIZM 320
$$/ TITL 0 Example ConTest Panel
$$/ FRAA 01 100 20 260 280
$$/ FRAL 01 ADC Readings
$$/ VALA 01 260 70
$$/ VALL 01 Voltage A
$$/ VALA 02 260 150
$$/ VALL 02 Voltage B
$$/ VALA 03 260 230
$$/ VALL 03 Ratio A / B
$$/ GPHA 01 120 50
$$/ GPHL 01 Voltage A History
$$/ GPHA 02 120 130
$$/ GPHL 02 Voltage B History
$$/ GPHA 03 120 210
$$/ GPHL 03 Ratio History
$$/ LEDA 01 450 30
$$/ LEDL 01 Button 1
$$/ LEDA 02 450 80
$$/ LEDL 02 Button 2
$$/ LEDA 03 450 130
$$/ LEDL 03 Button 3
$$/ LEDA 04 450 180
$$/ LEDL 04 Button 4
$$/ LEDA 05 450 230
$$/ LEDL 05 Button 5
$$/ LEDA 06 450 280
$$/ LEDL 06 Button 6
$$/ LEDA 07 550 30
$$/ LEDL 07 Button 1
$$/ LEDA 08 550 80
$$/ LEDL 08 Button 2
$$/ LEDA 09 550 130
$$/ LEDL 09 Button 3
$$/ LEDA 10 550 180
$$/ LEDL 10 Button 4
$$/ LEDA 11 550 230
```

4. ConTest Command Reference

4.1. Initialise

This command takes no parameters, and performs the following tasks:

- Removes all current components
- Removes any messages from the message window
- Resets the ConTest window back to the default size of 700 by 640 pixels.

4.2. Window Size

Two commands are used to re-size ConTest. **SIZA** sets the overall window size (in pixels), while SIZM adjusts the pane separator bar to make the lower pane (message pane) correspond to the given height.

4.3. Title

The ConTest title at the top left of the application window can be replaced by the text following this command. All text up to the end of line is interpreted as the title, so spaces may be used freely.

4.4. Frame

A frame can be used to group other components, to improve the 'look and feel' of the panel. The frame has a rounded edge appearance, and contains a user defined tile bar at the top, centred in the frame. Frames can be any size (as long as it fits within the pane) but it is up to the user to ensure correct spacing relative to other components.

4.5. Text String

Text strings can be placed anywhere on the panel, and can be any height up to the maximum point size of 72.

4.6. Value Container

The value container is a rectangular box, containing the last value written to it. Note that 'value' in this case is simply the text string contained in the command. Therefore, the value can represent a number (binary, decimal, hex etc.) or a character string (status, mode, error condition etc.), as required. Value containers are used to display important parameters to the user.

4.7. Bar Graph

Numerical data can be represented as a bar graph. This is a horizontal bar scaled 0-99, and will be set to the last value received. Numbers outside the valid range are ignored, so it is up to the target to scale values appropriately.

4.8. Time Graph

Data trends can be represented by a time graph. This is scaled for the range 0-99. Each time a valid value is sent to a graph component, the value is added to the right hand end of the graph. All previous values are shifted down one place, and the oldest value is removed. The graph displays the last 100 values received. This is useful for showing data trends over time, updated at a rate determined by the rate of receiving new data from the target.

4.9. LED

Single bits of data can be represented as an LED on the panel. The LEDs are rectangular in shape and can be set to any of the colours listed below. This can be used to show boolean states, or multiple colours can be used on a single LED to indicate more possible states. 8 colours are available, corresponding to colour values of 0-7

0	black
1	green
2	orange
3	red
4	blue
5	brown
6	yellow
7	white

4.10. Send Value Container

This is similar to the Value Container in that it contains an ascii string value. However, in this case the data in the container is entered by the user and sent to the target when the adjacent SEND button is clicked. The label on the button can be changed using the **SVLL** command.
The command **SVLD** is, in fact, a setup command, and not normally sent regularly from the target. It is used to prefix the container strings with the string contained in this command. The purpose of this string is to enable the target to determine which control sent the data.
For example, two controls of this type could be set up, the first with an **SVLD** string of "SVA" and the second with an **SVLD** string of "SVB". If the value "55" was entered in the first control by the user, when the SEND button is clicked, the string sent to the target would be

SVA55

If the same data was entered into the second control the string sent would be

SVB55

4.11. Button

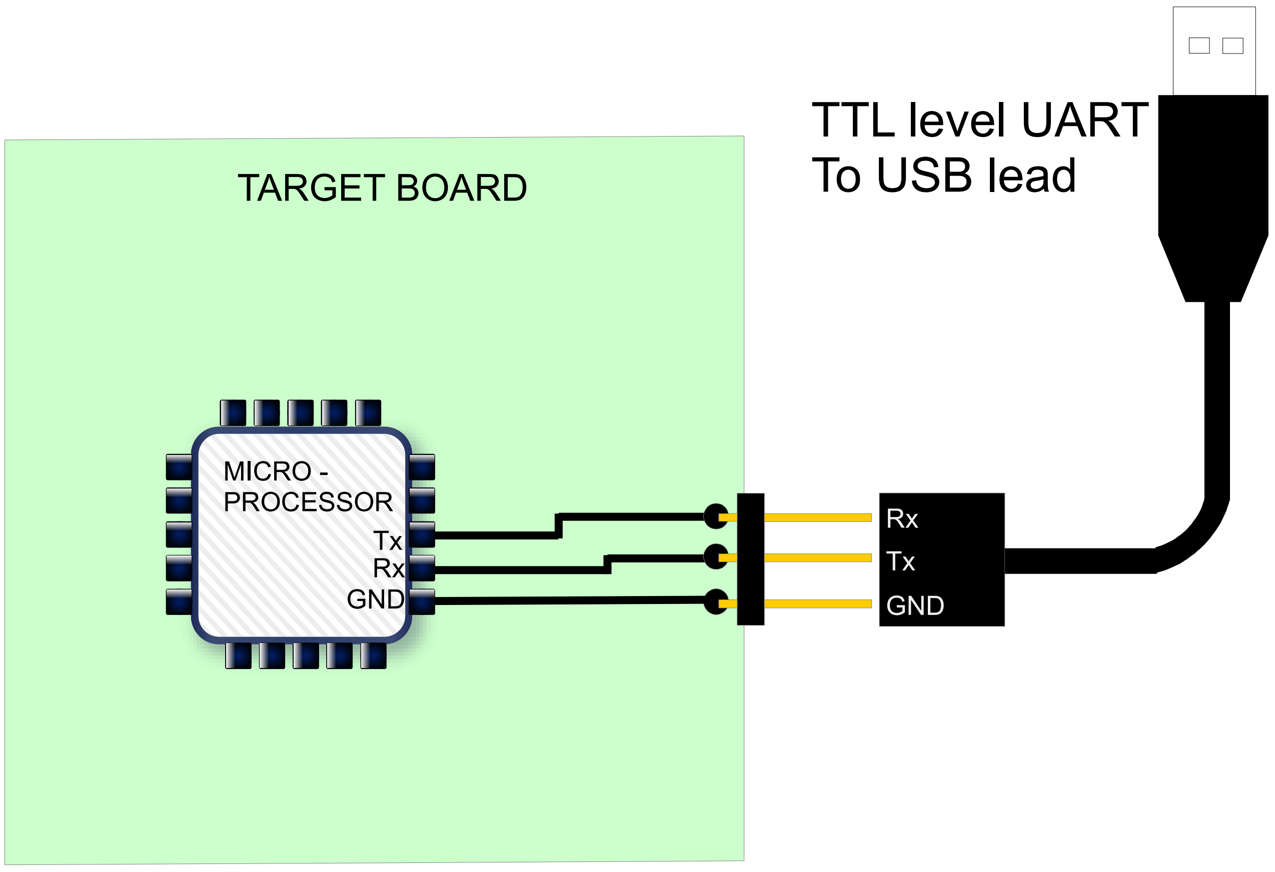
The button component is used to send a message to the target. In response to the user pressing the button (or clicking on the button) a text message is sent to the target via the serial link. The content of the message can be anything from a single character, to a complex string. The message content is set by the target (or in the panel file) by the **BTND** command. One message is sent for each "press" of the button.

4.12. Switch

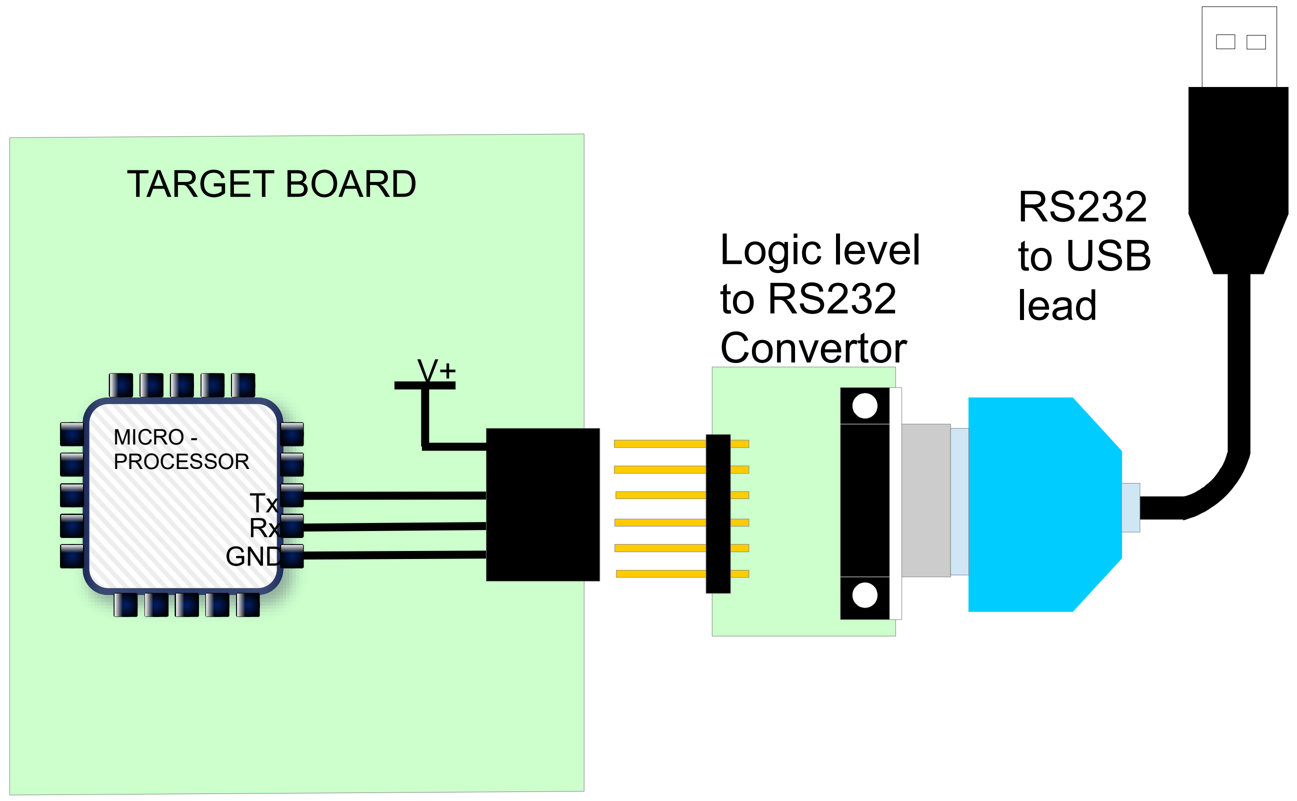
A switch component is an extension of the button. The switch has two latched states, "up" and "down". The switch starts in the position ***state*** from the **SWTA** command. Each subsequent "press" by the user puts the switch into the other state, and sends a message to the target in the same way as for the button. However, in the case of a switch, one of two possible messages are sent. These are set by the **SWD1** and **SWD2** commands, and correspond to the switch being set "down" and the switch being set "up" respectively.

5. Example Hardware Setups

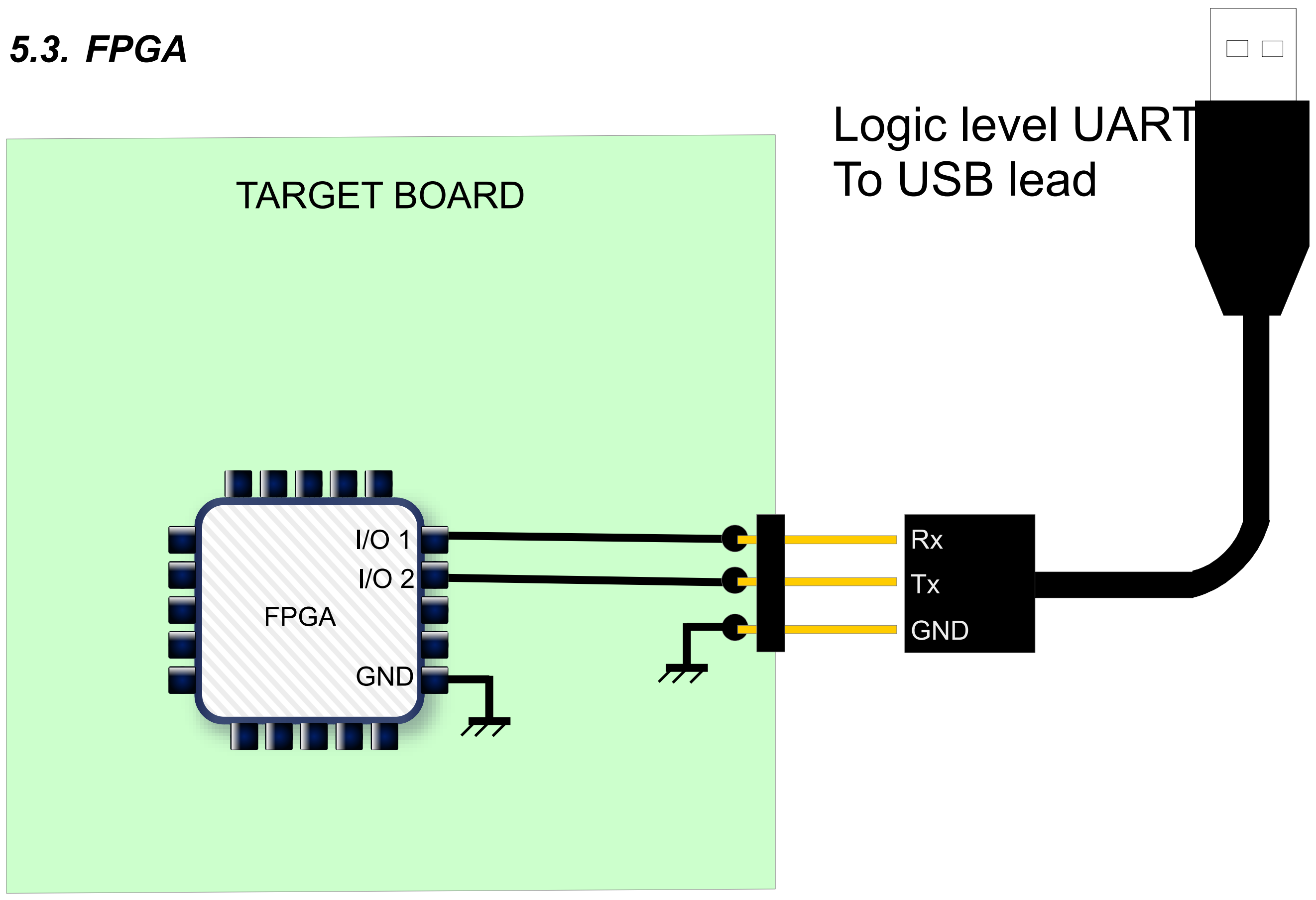
5.1. Microprocessor



5.2. *Alternative Microprocessor*



5.3. *FPGA*


Logic level UART To USB lead
TARGET BOARD
FPGA
I/O 1
I/O 2
GND
Rx
Tx
GND