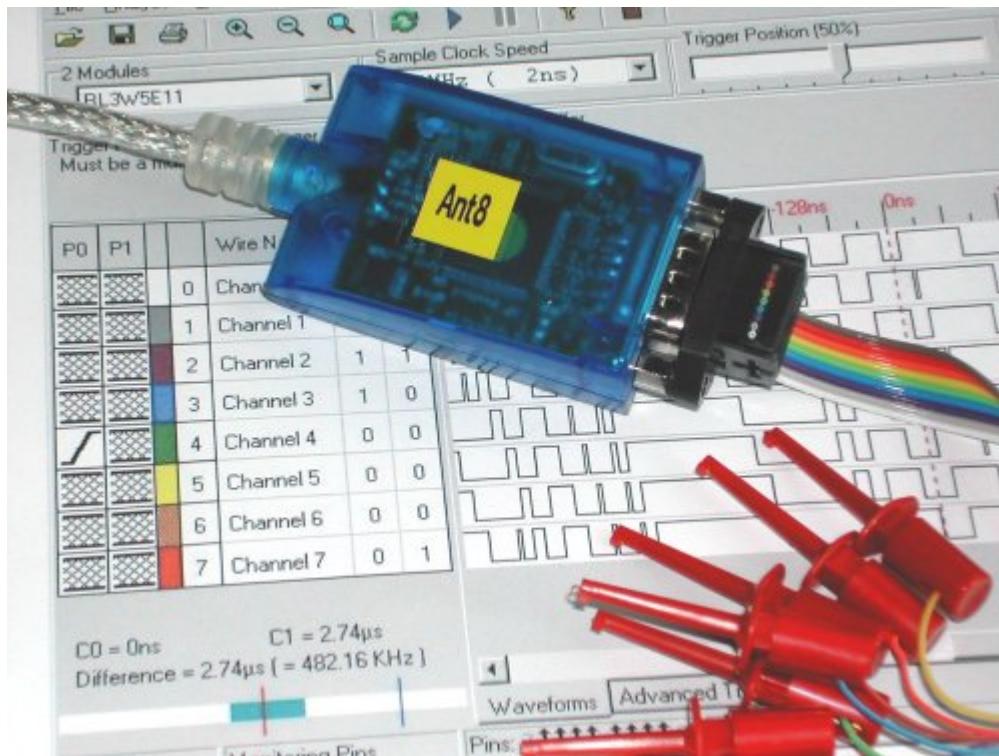


## ANT8 USB Logic Analyser User Manual



ANT8 User Manual created by USB Instruments, 2003

Contents Copyright (c) 2002, RockyLogic Ltd

# Table of Contents

<b>Part 1</b>	<b>Contents</b>	<b>1</b>
1.1	OverView .....	1
1.2	Main Window .....	2
1.3	Advanced Triggering Window .....	3
<b>Part 2</b>	<b>Getting Started</b>	<b>4</b>
2.1	Software Installation .....	5
2.2	Driver Installation .....	10
2.3	Triggering .....	12
2.4	Wire Names and Colors .....	17
2.5	Patterns .....	18
2.6	Signal Levels at the Cursors .....	19
2.7	Waveform Display .....	19
<b>Part 3</b>	<b>File Menu</b>	<b>20</b>
3.1	File Read Configuration .....	20
3.2	File Read Data .....	20
3.3	File Save Configuration .....	20
3.4	File Save Configuration As .....	21
3.5	File Save Data .....	21
3.6	File Save Data As .....	21
3.7	File Print Setup .....	21
3.8	File Print .....	21
3.9	File Exit .....	22
<b>Part 4</b>	<b>Analyzer Menu</b>	<b>23</b>
4.1	Analyzer Continuous .....	23
4.2	Analyzer Run .....	23
4.3	Analyzer Stop .....	23
<b>Part 5</b>	<b>View Menu</b>	<b>24</b>
5.1	View Zoom In .....	24
5.2	View Zoom Out .....	24
5.3	View Zoom All .....	24

<b>Part 6</b>	<b>Help Menu</b>	<b>25</b>
6.1	Help About .....	25
<b>Part 7</b>	<b>File Formats</b>	<b>26</b>
7.1	Configuration File Format .....	26
7.2	Data File Formats .....	27
<b>Part 8</b>	<b>Frequently Asked Questions</b>	<b>29</b>
8.1	FAQs .....	29
<b>Part 9</b>	<b>Hardware Specification</b>	<b>34</b>
9.1	Specification .....	34
<b>Part 10</b>	<b>Control Settings</b>	<b>35</b>
10.1	Modules .....	35
10.2	Module Type .....	35
10.3	Sample Clock Speed .....	35
<b>Part 11</b>	<b>Trigger Settings</b>	<b>36</b>
11.1	Trigger Position .....	36
11.2	Trigger Types .....	36
11.3	Pattern Triggering .....	36
11.4	Edge Triggering .....	37
11.5	Advanced Triggering .....	37
<b>Part 12</b>	<b>Status Information</b>	<b>39</b>
12.1	Pin Status .....	39
12.2	Program Status .....	39
12.3	Status Bar Hint .....	39
<b>Part 13</b>	<b>Cursors</b>	<b>40</b>
13.1	Cursor Measurements .....	40
<b>Part 14</b>	<b>Miscellaneous</b>	<b>41</b>
14.1	Mimic Diagram .....	41
14.2	Long Hint .....	41

# 1 Contents

## 1.1 OverView

### ANT8 USB Logic Analyser

These pages show you how to control the logic analyzer:

- [Main Window](#)
- [Advanced Triggering Window](#)

General information on installing and running the logic analyzer is here:

- [Getting Started](#)
- [Triggering](#)

The menus are described here:

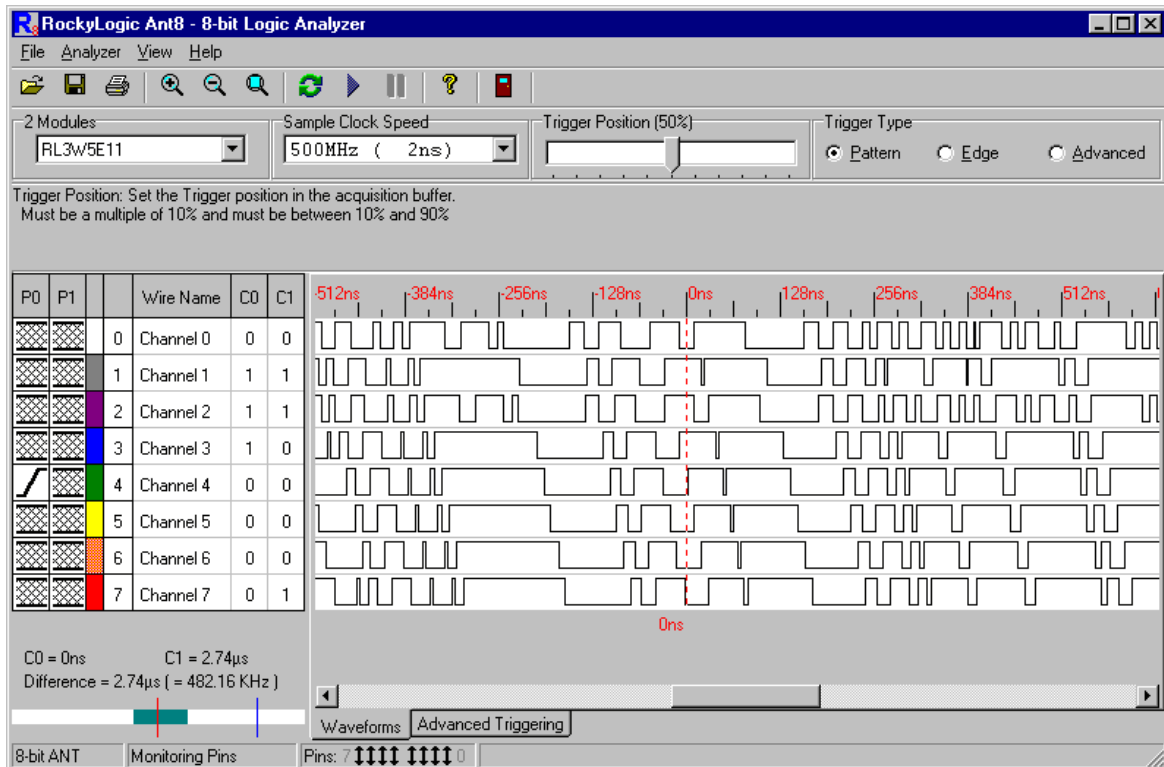
- [File Menu](#)
- [Analyzer Menu](#)
- [View Menu](#)
- [Help Menu](#)

ReferenceMaterial:

- [Specification](#)
- [Configuration File Format](#)
- [Data File Formats](#)
- [FAQs](#)

## 1.2 Main Window

This is the main window of the program. Click on the image below to learn more about each item.



The auxiliary [Advanced Triggering](#) window is used for complex data acquisitions.

[Modules](#)

[Sample Clock Speed](#)

[Trigger Position](#)

[Trigger Types](#)

[Long Hint](#)

[Patterns](#)

[Wire Names and Colors](#)

[Signal Levels at the Cursors](#)

[Cursor Measurements](#)

[Mimic Diagram](#)

[Waveform Display](#)

[Module Type](#)

[Program Status](#)

[Pin Status](#)

[Status Bar Hint](#)

## 1.3 Advanced Triggering Window

Click on the image below to learn more about each item.

The screenshot shows the Advanced Triggering Window interface with the following settings:

- Pattern 0 Combining:**  AND combine,  OR combine
- Pattern 1 Combining:**  AND combine,  OR combine
- X0 definition:**  Pattern 0,  NOT Pattern 0,  Pattern 1,  NOT Pattern 1,  P0 AND P1,  P0 OR P1,  Custom
- Operation:**  X0 and X1,  X0 or X1,  X0 then X1,  X0 Occurs N Times,  X0 Entered,  X0 Exited,  X0 True Longer Than,  X0 True Shorter Than
- X1 definition:**  Pattern 0,  NOT Pattern 0,  Pattern 1,  NOT Pattern 1,  P0 AND P1,  P0 OR P1,  Custom
- Timer/Counter:** 2
- Expression 1:**  $(!P1)$
- Expression 2:**  $(P0 \wedge P1)$

See also the [Triggering](#) page.

## 2 Getting Started

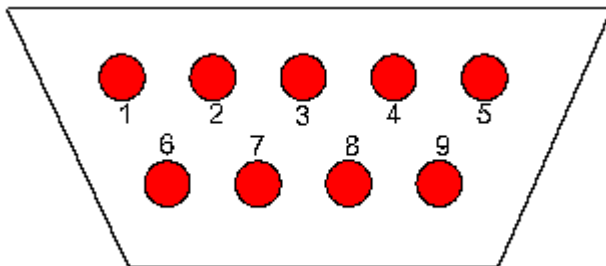
Getting started with the Ant8 software is a 3-stage process

1. Insert the CD provided into the target PC CD ROM drive and install the ANT8 Software and Drivers. See the [Software Installation](#) and [Driver Installation](#) topics.
2. Plug in the Ant8 module. The first time this is done it immediately causes the operating system to prompt you for the location of the necessary USB drivers. See the [Driver Installation](#) topic for details.
3. Run the Ant8 application to verify the installation. The installer will have left shortcuts on your desktop and in your Start menu.

Now you need to connect to the target circuit via logic grabbers, set the acquisition parameters, and acquire some data.

### Connecting to the Target

The pinning for the Ant8 input connector is as shown below. If you peer closely into the connector, you can see these pin numbers.



Signal channels are assigned as follows:

#### Top Row

Pin 1	Ground
Pin 2	Channel 1
Pin 3	Channel 3
Pin 4	Channel 5
Pin 5	Channel 7

#### Bottom Row

Pin 6	Channel 2
Pin 7	Channel 4
Pin 8	Channel 6
Pin 9	Channel 8

See also [Wire Names and Colors](#). Take care to connect Pin 1 to the target ground.

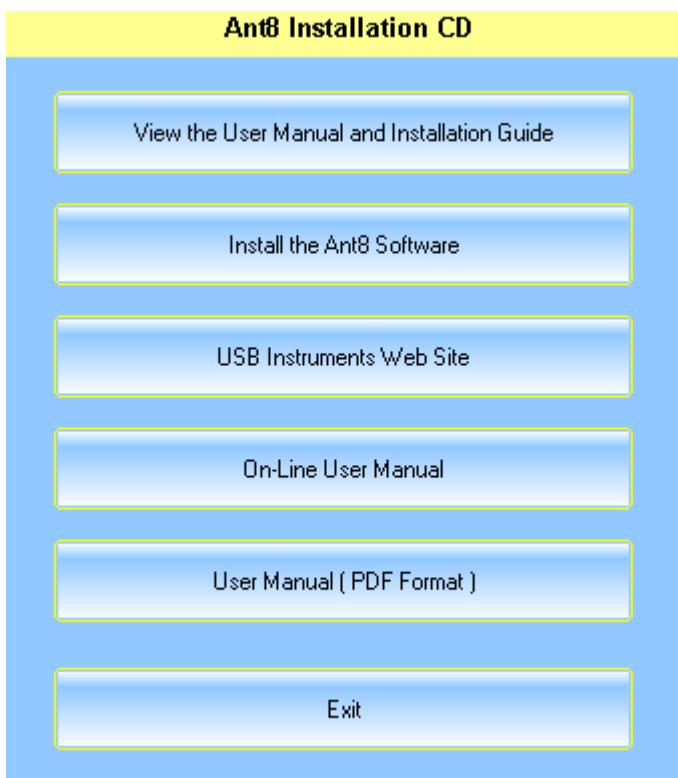
### Setting Acquisition Parameters

The default acquisition parameters, with all pins set to *don't care*, will sample the data on the probes. To make a more controlled acquisition you must set some or all of the controls, such as the patterns P0 and P1, and the Sample Clock speed. See the [Main Window](#) help.

Then press *Run* and off you go.

## 2.1 Software Installation

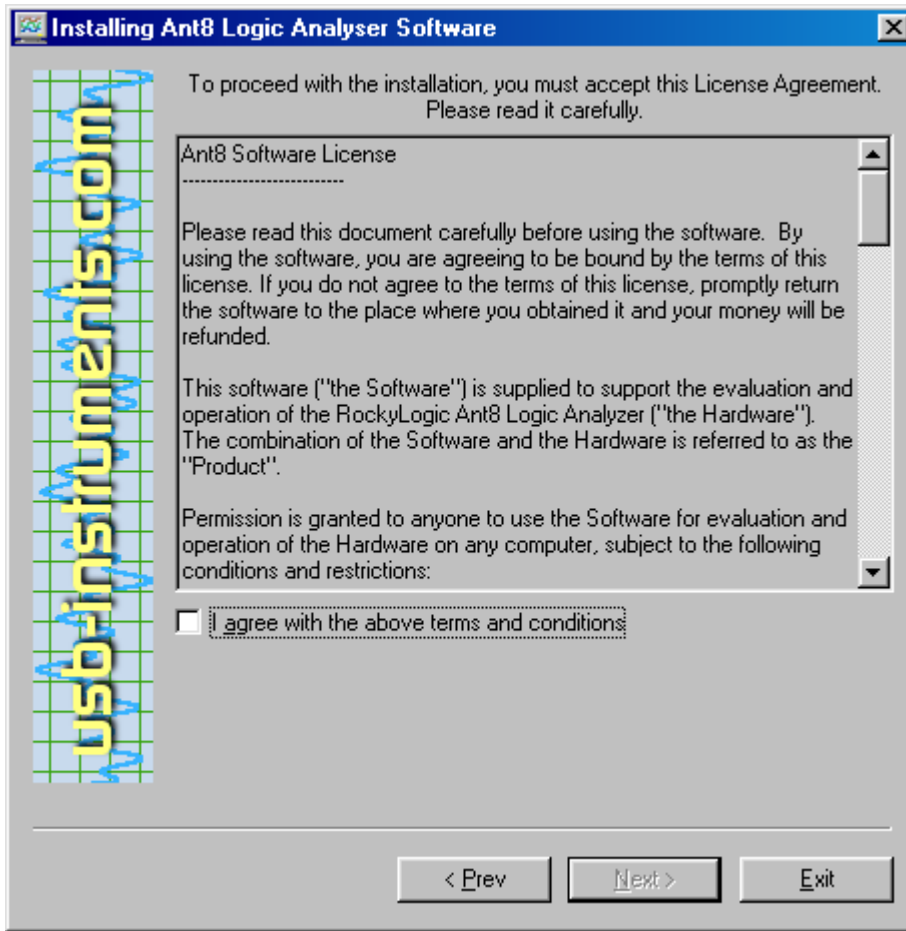
Before connecting the instrument to the PC, insert the supplied installation disk into the CD ROM drive of your PC. The following menu will appear after a few seconds. Click on the "Install the ANT8 Software" button to launch the Ant8 Software installation program.



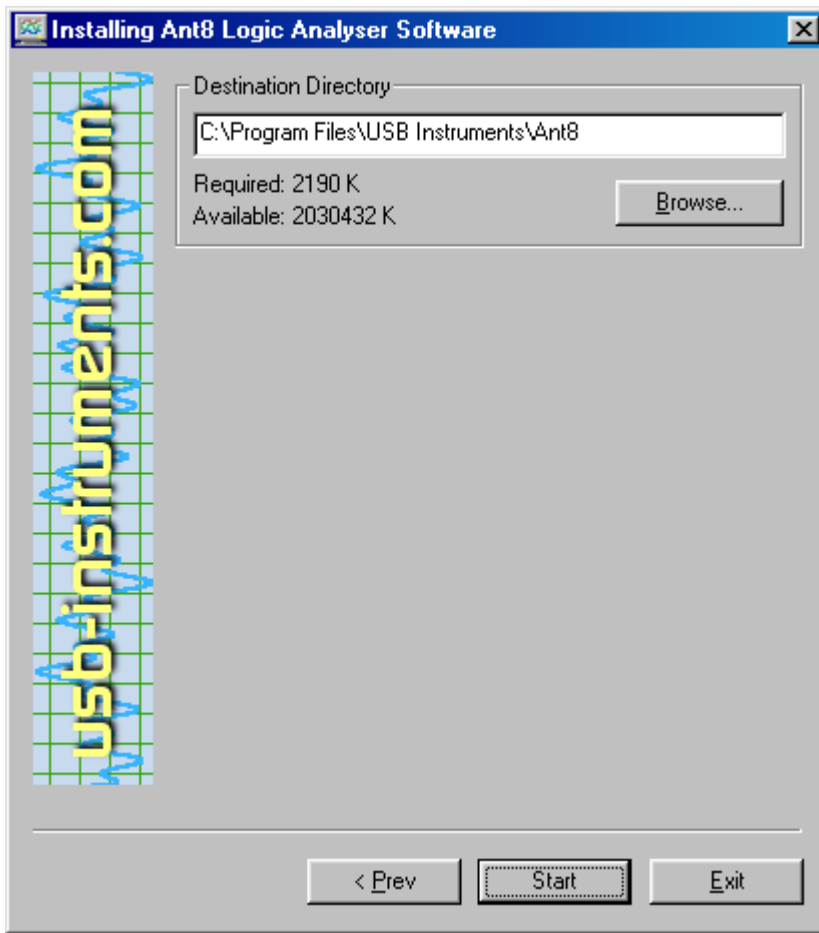
The following screen will appear ..



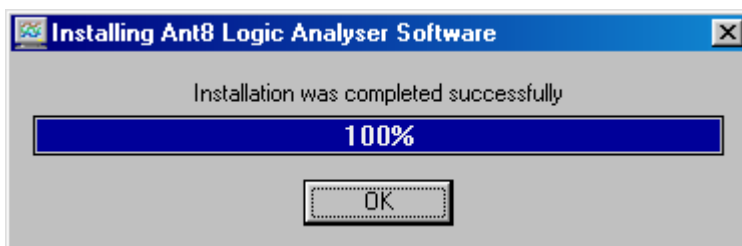
Choose the language you would like the setup program to use. In this example we will select English. Click on the "Next" button to bring up the License Agreement Screen.



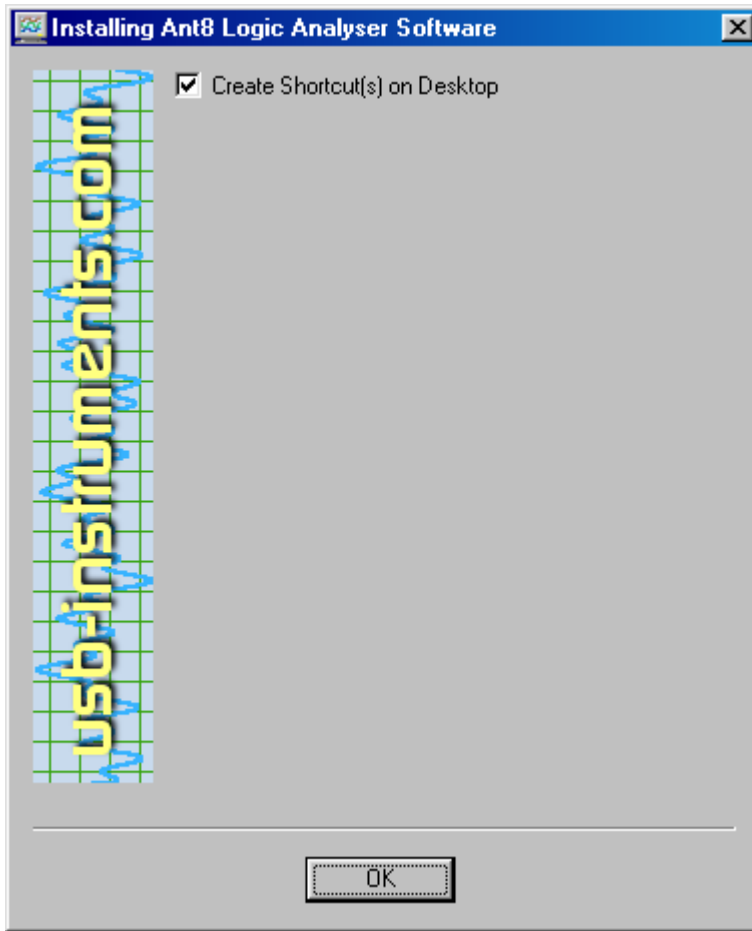
If you agree with the terms and conditions of the License Agreement, check the "I agree" box and click on the "Next" button to continue the installation process else click on the "Exit" button. The "Next" button is disabled if you do not agree to the terms and conditions. Agreeing to the terms and clicking on the "Next" button brings up the following screen.



Select the directory you wish to install the Ant8 Software software into. A default directory is shown on the screen. Unless you have good reason to change it we suggest you use the default suggested by the installation program. Click on the "Start" button to commence copying the files to the Ant8 Software program directory.



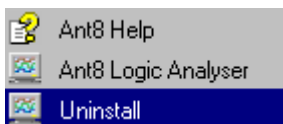
A progress screen (above) will appear as the files are installed. After a few seconds you should see the Installation completed successfully message. Click on the "OK" button to complete the final stage of the installation process.



Finally, select if you wish to place a shortcut icon on the Windows desktop to the Ant8 Software program and click on "OK". The installation process is now complete. If you have selected this option, you can launch the Ant8 Software program from the desktop by double-clicking on the Ant8 Software icon ( pictured below ).



You can also launch the Ant8 Software program from the Start -> Programs -> Ant8 Software menu on the Windows Toolbar.



As shown above, you also have access to the Ant8 Software Help File and Program Uninstaller from there.

Before using Ant8 Software the USB drivers need to be installed. This is done by plugging in the

instrument into a spare USB Port on the PC. The instrument should be plugged into a USB hub port of your PC or alternatively, a self-powered USB hub ( one that has it's own power supply ). If the instrument consumes under 100mA ( check the specification ), you may also plug it into a bus-powered USB Hub if desired.

If this is the first time that the instrument has been plugged-in, Windows will then request the USB drivers for your product. See the [Driver Installation](#) topic for further details.

## 2.2 Driver Installation

Before using Ant8 Software the USB drivers need to be installed. This is done by plugging in the instrument into a spare USB Port on the PC. The instrument should be plugged into a USB hub port of your PC or alternatively, a self-powered USB hub ( one that has it's own power supply ). If the instrument consumes under 100mA ( check the specification ), you may also plug it into a bus-powered USB Hub if desired.

If this is the first time that the instrument has been plugged-in, Windows will then request the USB drivers for your product, Windows will request the USB drivers for your product and will display a Add New Hardware Wizard dialog box. The examples below are for Windows '98 but the procedure is very similar for other Windows versions.



Click on "Next" to bring up the following dialog box ...



Click on "Next" to bring up the following dialog box ...



Check "CD-ROMdrive" and un-check any other options.

NOTE : If the CD is not available and the Ant8 Software software has already been installed, then check the "Specify a location" box and use the Browse button to select the "drivers" sub-directory of the Ant8 Software Program Files directory instead.

Keep Clicking on "Next" until the installation is finished as per the screen below.



Click "Finish" and the driver installation is now complete.

## 2.3 Triggering

The Ant8 module searches for pattern in the signals on its input probes and triggers when a pattern is recognized. The Ant8 supports three trigger types:

- Pattern triggering
- Edge triggering
- Advanced triggering

Pattern and Edge triggering are simple acquisition modes, controlled via a single pattern recognizer. Advanced triggering can make use of both of the pattern recognizers in the Ant8, and a timer/counter unit.

Pattern and Edge triggering are fairly common terminology. Alternatively you can think of them as *AND* triggering and *OR* triggering.

### Pattern Triggering

In Pattern triggering you specify a pattern of lows and highs, rising and falling edges, and don't cares, *all* of which must be recognized across the input channels to trigger the logic analyzer.

For instance, consider the following pattern:

```
Channel 0: rising edge
Channel 1: don't care
Channel 2: don't care
Channel 3: don't care
Channel 4: 1
Channel 5: 0
```

```
Channel 6: 1
Channel 7: 0
```

The analyzer will trigger when channels 4 to 7 see the pattern 1010 and channel 0 sees a rising edge. Note that pattern triggering is activated by the logical AND of the various conditions.

## Edge Triggering

In Edge triggering you specify a pattern of lows and highs, and rising and falling edges, *any* of which must be seen on the on the input channels to trigger the logic analyzer. Don't cares are ignored in edge triggering.

For instance, consider the following pattern:

```
Channel 0: rising edge
Channel 1: rising edge
Channel 2: don't care
Channel 3: don't care
Channel 4: don't care
Channel 5: don't care
Channel 6: 1
Channel 7: 0
```

The analyzer will trigger when either channel 0 sees a rising edge or channels 6 is high or channel 7 is low. Note that edge triggering is activated by the logical OR of the various conditions.

## Advanced Triggering

Triggering in the Ant8 is implemented via two pattern matchers, a timer/counter, and an internal data acquisition state machine. Advanced triggering allows you to more fully control the operation of the matchers and the state machine.

Setting up an advanced triggering data capture involves the following stages:

1. define the data patterns, P0 and P1
2. decide whether the data pattern terms are to be ANDed together or ORed together
3. define the functions (X0 and X1) which combine the patterns
4. set the timer/counter value
5. choose the advanced trigger operation

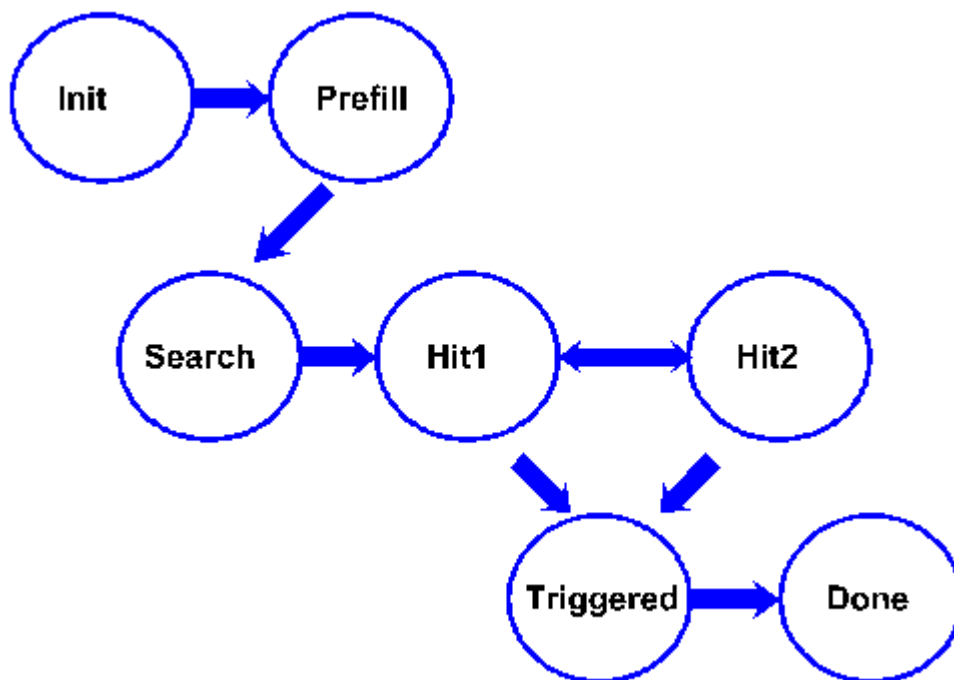
For instance, this could be an advanced setup:

1. set the patterns as for a simple trigger
2. pattern terms are ANDed together for both P0 and P1
3.  $X0 = (P0 \text{ or } P1)$ ,  $X1 = (\text{not } P1)$
4. Timer/Counter = don't care
5. operation is X0 then X1, which is  $(P0 \text{ or } P1)$  then  $(\text{not } P1)$

Each stage is described in more detail below.

## Data Acquisition State Machine

The following diagram shows the operation of the state machine:



Transitions between states are as follows:

**Idle State** Move to Prefill when Run is signaled from the Host PC.

**Prefill State** Move to Search when the defined prefill percentage of the acquisition memory has been filled

**Search State** Move to Triggered if the SearchTriggered condition has been seen. Else move to Hit1 if the SearchHit1 condition has been seen.

**Hit1 State** Move to Triggered if the Hit1Triggered condition has been seen. Else move to Hit2 if the SearchHit2 condition has been seen.

**Hit2 State** Move to Triggered if the Hit2Triggered condition has been seen. Else move to Hit1 if the SearchHit1 condition has been seen.

**Triggered State** Move to Done when the defined postfill percentage of the acquisition memory has been filled.

**Done State** Move to Idle when Reset is signaled from the Host PC.

## P0/P1 Pattern Term Combining

Patterns are established for the two pattern matchers (P0 and P1), as for the simple pattern and edge cases. In addition, you can set the term combining conditions for each matcher. The rules are:

**And Combine:** The complete pattern matcher registers a hit if the term for each signal matches. Don't Care terms always match, so a matcher with all terms set to don't care will always match. *And*

*combining* is the standard setting for Pattern Triggering.

**Or Combine:** The complete pattern matcher registers a hit if any one signal matches. Don'tCare terms nevermatch, so a matcher with all terms set to don't care will nevermatch. *Or combining* is the standard setting for Edge Triggering.

## X0/X1 Stimulus Definition

The pattern recognizer outputs are combined to form two stimuli which drive the state machine. We refer to the pattern recognizer outputs as P0 and P1, and the stimuli are X0 and X1.

For simple triggering:

X0 = P0

X1 = P0, although stimulus X1 is not used in simple triggering

For advanced triggering, six preset combinations of P0 and P1 are defined, and a custom combination can also be defined. The preset combinations are:

X0 = P0

X0 = not P0

X0 = P1

X0 = not P1

X0 = P0 and P1

X0 = P0 or P1

with similar equations for X1.

The custom combination can be defined as an equation using the operators:

!	NOT
&	AND
	OR
^	XOR

The defining equation for a custom combination should be enclosed in parentheses. Spaces are not significant. For instance:

(P0 ^ P1)

i.e. P0 XOR P1

(P0 & !P1)

i.e. P0 AND (NOT P1)

## Advanced Trigger Timer/Counter

The 10-bit Timer/Counter (TC) is automatically loaded in the Idle state and reloaded either on entering the Hit1/Hit2 states or when in the Hit1/Hit2 states. TC counts down to zero either on entering the Hit1/Hit2 states or when in the Hit1/Hit2 states; the count stops at zero.

## Advanced Trigger Operators

The following operators are defined:

**X0 and X1**

Trigger when both stimuli are true in the same clock cycle. The state machine traverses:

Idle->Prefill->Search->Triggered->Done

The programmable transition(s) are

Search->Triggered            when (X0 & X1)

### X0 or X1

Trigger when either stimulus is true. The state machine traverses:

Idle->Prefill->Search->Triggered->Done

The programmable transition(s) are

Search->Triggered            when (X0 | X1)

### X0 then X1

Trigger when either X0 is true, then X1 is true. The state machine traverses:

Idle->Prefill->Search->Hit1->Triggered->Done

The programmable transition(s) are

Search->Hit1                    when (X0)  
Hit1->Triggered                when (X1)

### X0 occurs N times

Trigger when either X0 becomes true for the Nth time. The state machine traverses:

Idle->Prefill->Search->Hit1->Hit2 or Triggered->Hit1 or Triggered->Triggered->Done

The programmable transition(s) are

Search->Hit1                    when (X0)  
Hit1->Hit2                      when (!X0)  
Hit2->Hit1                      when (X0)  
Hit1->Triggered                when (TC=0)  
Hit2->Triggered                when (TC=0)  
TC Countdown Enable        when entered Hit1

### X0 entered

Trigger when X0 becomes true for the first time. The state machine traverses:

Idle->Prefill->Search->Hit1->Triggered->Done

The programmable transition(s) are

Search->Hit1                    when (!X0)  
Hit1->Triggered                when (X0)

### X0 exited

Triggerwhen X0becomes false for the first time. The state machine traverses:

Idle->Prefill->Search->Hit1->Triggered->Done

The programmabletransition(s) are

Search->H it1	when (X0)
H it1->Triggered	when (X0)

### X0 true longer than

Triggerwhen either X0is true for more than N cycles. The state machine traverses:

Idle->Prefill->Search->Hit1->Hit2or Triggered->Hit1->Triggered->Done

The programmabletransition(s) are

Search->H it1	when (X0)
H it1->H it2	when (X0)
H it2->H it1	when (X0)
H it1->Triggered	when (TC=0)
TC Countdown Enable	when in H it1
TC Load	when entered H it2

### X0 true shorter than

Triggerwhen either X0is true for less than N cycles. The state machine traverses:

Idle->Prefill->Search->Hit1->Hit2->Hit1or Triggered->Triggered->Done

The programmabletransition(s) are

Search->H it1	when (X0)
H it1->H it2	when (X0)
H it2->H it1	when (X0)
H it2->Triggered	when (TC != 0)
TC Countdown Enable	when in H it1
TC Load	when entered H it2

## 2.4 Wire Names and Colors

### Wire Names and Colors

The wire names can be changed by clicking once to select a name, then editing in the normal way.

By default, wires are arranged in sequence, with wire 0 at the top. They can be rearranged by dragging near the wire sequence number.

The default wire colors are

wire 0 - red  
wire 1 - orange

wire 2 - yellow  
wire 3 - green  
wire 4 - blue  
wire 5 - purple  
wire 6 - gray  
wire 7 - white

The default color for the ground wire is brown.

These colors match standard ribbon cable coloring. The colors can be changed via the [Configuration File](#) Legal color names are:

BtnFace  
Window  
White  
Gray  
Black  
Red  
Yellow  
Orange  
Green  
Blue  
Aqua  
Purple  
Navy  
Fuchsia  
Olive  
Teal  
Lime

## 2.5 Patterns

### Patterns

There are two pattern matchers (P0 and P1) in the Ant8. The code for setting the patterns is



don't care. In [Pattern Triggering](#) mode, when matches on the pins are AND'd together, this matches on any edge or level. In [Edge Triggering](#) mode, when matches on the pins are OR'd together, this never matches.



pin is low - matches on a low input



pin is high - matches on a high input



rising edge - matches on a rising edge



falling edge - matches on a falling edge



either edge - matches on a rising edge or a falling edge

Use the left mouse button to cycle round the pattern states. The right mouse button returns the

pattern to *don't care*.

## 2.6 Signal Levels at the Cursors

### Signal Levels at the Cursors

Columns C0 and C1 show the signal levels at the cursors.

C0 shows the signal levels at the primary cursor. C1 shows the signal levels at the secondary cursor.

## 2.7 Waveform Display

### Waveform Display

The timescale runs across the top of the waveform window and cursor position feedback runs across the bottom of the waveform window.

The cursors can be repositioned by clicking with the left mouse button to reposition the main cursor (Cursor0) and clicking with the right mouse button to reposition the secondary cursor (Cursor1). The default cursor colors are

Cursor0 - red  
Cursor1 - blue

These colors can be changed via the [Configuration File](#)

The waves are shown with transitions occurring between sample points.

## 3 File Menu

### File Menu

[File|Read Configuration](#)  
[File|Read Data](#)

[File|Save Configuration](#)  
[File|Save Configuration As](#)  
[File|Save Data](#)  
[File|Save Data As](#)

[File|Print Setup](#)  
[File|Print](#)

[File|Exit](#)

### 3.1 File|Read Configuration

#### File|Read Configuration

Prompts for a filename and then loads a new configuration.

See also the [Configuration File Format](#)

### 3.2 File|Read Data

#### File|Read Data

Prompts for a filename and then reads sample data from a native format file.

See also the [Data File Formats](#).

### 3.3 File|Save Configuration

#### File|Save Configuration

Saves the current configuration in the file `ant8.ini`

See also the [Configuration File Format](#)

## 3.4 File|Save Configuration As

### File|Save Configuration As

Prompts for a filename and then saves the current configuration.

See also the [Configuration File Format](#)

## 3.5 File|Save Data

### File|Save Data

Saves the current sample data in native format in the file `ant8.rld`

See also the [Data File Formats](#).

## 3.6 File|Save Data As

### File|Save Data As

Prompts for a filename and data type, and then saves the current sample data.

See also the [Data File Formats](#).

## 3.7 File|Print Setup

### File|Print Setup

Presents the standard dialog for choosing printers and setting printer options.

## 3.8 File|Print

### File|Print

Printing is not implemented in this software release.

### 3.9 File|Exit

#### File|Exit

Exits the program, with no automatic saving of configuration or sample data.

## 4 Analyzer Menu

### Analyzer Menu

[Analyzer|Continuous](#)

[Analyzer|Run](#)

[Analyzer|Stop](#)

### 4.1 Analyzer|Continuous

#### Analyzer|Continuous

Toggles between single-shot mode and continuous mode.

In single-shot mode mode, the acquisition logic takes a single reading.

In continuous mode, the acquisition logic takes a reading, displays it, then cycles round to take another reading. All acquisition parameters are reloaded before each reading so, for instance, the clock speed can be varied while the logic is in continuous mode.

### 4.2 Analyzer|Run

#### Analyzer|Run

The current settings are sent to the logic analyzer, then it is instructed to start the prefill->trigger->postfill sequence. In continuous mode ([Analyzer|Continuous](#)), the software will repeat this sequence indefinitely.

A run may be stopped by clicking the [Analyzer|Stop](#) button.

### 4.3 Analyzer|Stop

#### Analyzer|Stop

Immediately halts the logic analyzer and displays the contents of the analyzer's acquisition memory. Does not disable continuous mode ([Analyzer|Continuous](#)), but does stop a continuous acquisition sequence.

## 5 View Menu

### View Menu

[View|Zoom In](#)  
[View|Zoom Out](#)  
[View|Zoom All](#)

### 5.1 View|Zoom In

#### View|Zoom In

Doubles the current zoom level, up a maximum of 16 pixels per sample.

### 5.2 View|Zoom Out

#### View|Zoom Out

Halves the current zoom level, to a minimum when all available samples are displayed on the screen.

### 5.3 View|Zoom All

#### View|Zoom All

Displays all available samples on the screen.

## 6 Help Menu

### Help Menu

[Help|Contents](#)

[Help|About](#)

### 6.1 Help|About

#### Help|About

Shows version and copyright information.

The program version applies to the console application running on your PC. The firmware versions apply to the logic which is downloaded to the analyzer module.

## 7 File Formats

### 7.1 Configuration File Format

The configuration file is a standard Windows .INI file. The various sections are shown in the example below. See also [Wire Names and Colors](#), [Patterns](#), and [Advanced Triggering](#)

#### [General]

```
AntProgramVersion=Aug 25 2002
ConfigWriteDate=27/08/02
ConfigWriteTime=09:32:51
Module=RL3W5E11
```

#### [WireNames]

```
0=Channel0
1=Channel1
2=Channel2
3=Channel3
4=Channel4
5=Channel5
6=Channel6
7=Channel7
```

#### [WireColors]

```
0=White
1=Gray
2=Purple
3=Blue
4=Green
5=Yellow
6=Orange
7=Red
```

#### [Waves]

```
BackColor=Window
WaveColor=Black
TickBackColor=BtnFace
TickTickColor=Black
TickNumColor=Red
Cursor0Color=Red
Cursor1Color=Blue
```

#### [Pattern0]

```
0=DONT_CARE
1=DONT_CARE
2=DONT_CARE
3=DONT_CARE
4=DONT_CARE
5=DONT_CARE
```

```
6=DONT_CARE
7=DONT_CARE
```

#### [Pattern1]

```
0=DONT_CARE
1=DONT_CARE
2=DONT_CARE
3=DONT_CARE
4=DONT_CARE
5=DONT_CARE
6=DONT_CARE
7=DONT_CARE
```

#### [Trigger]

```
SpeedIndex=0
TriggerPos=50
TriggerType=0
P0Combine=AND
P1Combine=AND
A_Function=(P0)
B_Function=(P0)
ComplexOp=0
TimerCounter=2
```

## 7.2 Data File Formats

Sample data can be saved in either native (RLD) format or comma-separated (CSV) format.

### Native Format

Here is an example of the first few rows of an RLD file - there is one row per sample. Data values are in decimal. This format is a version of a standard Windows .INI file.

#### [General]

```
AntProgramVersion=Sep 20 2002
DataWriteDate=24/09/02
DataWriteTime=11:39:50
```

#### [Trigger]

```
TriggerPos=555
```

#### [WireNames]

```
0=Ch0
1=Ch1
2=Ch2
3=Ch3
4=Ch4
5=Ch5
6=Ch6
7=Ch7
```

[Data]

0=17  
1=17  
2=16  
3=16  
4=16  
5=36  
6=36  
7=36  
8=17

## CSV Format

Here is an example of the first few rows of a CSV file - there is one row per sample. Data values are in binary. This format can be read by most spreadsheet programs.

```
AntProgramVersion,"Sep 20 2002"  
DataWriteDate,"24/09/02"  
DataWriteTime,"11:40:07"  
TriggerPos,555  
"Ch0", "Ch1", "Ch2", "Ch3", "Ch4", "Ch5", "Ch6", "Ch7"  
0,0,0,0,0,0,0,0  
1,0,0,0,0,0,0,0  
0,1,0,0,0,0,0,0  
1,1,0,0,0,0,0,0  
0,0,1,0,0,0,0,0  
1,0,1,0,0,0,0,0  
0,1,1,0,0,0,0,0  
1,1,1,0,0,0,0,0
```

## 8 Frequently Asked Questions

### 8.1 FAQs

#### Logic Analyzer FAQs

- Q1. What is a Logic Analyzer?
- Q2. Why not use an Oscilloscope?
- Q3. Triggering?
- Q4. What is Complex Triggering?
- Q5. How many channels do I need?
- Q6. How fast should the sampling be?
- Q7. Isolation?
- Q8. Probing?

#### USB FAQs

- Q9. Can my computer use USB devices?
- Q10. What is hot-swapping?
- Q11. What is plug and play?
- Q12. Hubs?
- Q13. Self-powered and powered hubs?
- Q14. How fast is USB?
- Q15. How long can cables be?

#### Miscellaneous FAQs

- Q16. How do you spell Logic Analyzer?

### Logic Analyzer FAQs

#### 1. What is a Logic Analyzer?

A Logic Analyzer is essentially a multi-channel version of a digital oscilloscope. One difference is that the logic analyzer only recognises two logic levels - 0 and 1. Also logic analyzers usually implement much more complex triggering than oscilloscopes.

#### 2. Why not use an Oscilloscope?

An oscilloscope can look at the same digital signals as a logic analyzer, but its essential role for most users is analog measurements such as rise-times, fall-times, and peak amplitudes. You use an oscilloscope when you need to see small voltage changes and when you need very accurate time interval measurements.

You use a logic analyzer when you need to see signals the same way your digital circuit sees them, when you need more complex triggering, and when you need more channels.

#### 3. Triggering?

When a logic analyzer is running it continuously samples the input signals and writes the values to

its memory buffer, stopping this process when it triggers. The trigger is the point at which a desired event occurs. The event could be a rising or falling edge or a pattern of zeroes and ones across the signal wires.

Edge triggering is usually set by the logical OR of various signal edges. For instance, trigger when the write strobe falls or the read strobe rises. In practice logic analyzers spend the majority of their lives responding to edge triggers on a single channel.

Pattern triggering is set by the logical AND of various signals. For instance, trigger when chip select is high, the write strobe is low, and the other signals are in any state.

#### 4. What is Complex Triggering?

This is the next stage beyond pattern triggering. The same events are defined as in the previous answer - ORs of various edges, ANDs of various levels. But these events are used to drive a *state machine* within the logic analyzer. The state machine moves between states and eventually arrives at the trigger state.

An example may clarify this:

- start in state 0
- move from state 0 to state 1 when you see a certain pattern
- move from state 1 to state 2 when the pattern goes away
- go back from state 2 to state 1 when an edge occurs
- trigger when you enter state 1 for the 500th time.

Notice that complex triggering introduces one more feature - a counter. This example illustrates all the essentials of a complex triggering capability - a few states, a few pattern or edge recognisers, and at least one counter.

The Ant8 module has a four-state architecture, with two edge/pattern recognisers, and a time/event counter which can range up to 1023.

Note that complex triggering state machines cannot usually run at the highest acquisition speeds. The Ant8 state machine runs at a maximum of 125MHz. At 250MHz it responds every second input sample, and at 500MHz it responds every fourth input sample.

#### 5. How many channels do I need?

For some applications, you can never have too many channels, and logic analyzers are available with several hundred channels. Typically these systems are used to debug complex microprocessor buses and connecting the myriad probes involved is lots of work.

The Ant8 module is aimed at a different scenario, when you want to see exactly what is happening on a few signals. Typically the Ant8 is hooked up to a few signals to give a trigger source, and to one or two signals which are being debugged.

For instance, when debugging a first-in/first-out (FIFO) memory the Ant8 could look at full and empty indicators, read and write strobes, and a few other signals for reference.

#### 6. How fast should the sampling be?

As you would expect, the maximum sampling rate can never be too fast. The Ant8 module samples

at up to 500MHz, which gives a timing resolution down to 2ns. Of course, the sample buffer fills up quite quickly at that speed, so for slower signals the sampling speed can be stepped down in stages to a minimum of 100Hz, which gives a timing resolution of 10ms.

Sampling beyond 500MHz would need sophisticated probes which would cost more than an Ant8 module. Even 500MHz needs short probe leads to give an accurate result.

### **7. Isolation?**

The Ant8 module is not galvanically isolated. It is connected to the host PC via the power, ground, and signal wires in the USB lead. A degree of isolation could be achieved by running the Ant8 module from an isolated laptop computer, or by using an optical USB extender. But fundamentally the module is not suitable for applications which indicate a need for galvanic isolation.

### **8. Probing?**

There are myriad solutions for connecting logic analyzers to the target circuit. The simplest is to use a ribbon cable terminated in a socket which connects to a plug on the target circuit board. But this needs planning. Even if the plug on the target circuit is designed in, it may be wired up to the wrong signals. The fallback is to use *grabber* clips. In modern designs, you often have to hook the grabbers to tiny wires soldered onto surface mount devices or soldered into vias in the circuit board. The only requirement for high-performance logic analysis is that the probing wires be very short. Minimise the loading on the circuit.

Probes are not included as standard with the Ant8 module, though we do offer a low-cost probe solution which is a set of small logic grabbers on a multicolored cable. Your local distributor may offer other probing solutions.

## **USB FAQs**

### **9. Can my computer use USB devices?**

**Hardware:** You need a USB port in your computer. Most computers built since 1998 have had at least two built in USB connectors. You can also get PCI and CardBus USB adapters for older computers.

**Software:** For Intel computers, you need Windows 98, Windows ME, Windows 2000, Windows XP, or a recent version of Linux. There is limited USB support on Windows 95 OSR 2.1, and none on Windows NT. For Apple Macintosh computers you need MacOS 8.1 or later.

Our products are developed and tested on various versions of Windows and the software we supply runs over Windows.

### **10. What is hot-swapping?**

USB is hot swappable. This means you can walk up to a computer, plug in a new device using USB, and use it right away. Like plugging into a serial port or an Ethernet port.

The Ant8 module is hot-swappable.

### **11. What is plug and play?**

Plug and play (PnP) refers to the ability to use a new peripheral without going through an elaborate

configuration process. Plug and play depends on the operating system used on your computer, Windows 98 and MacOS have a set of basic USB drivers built in that gives them true plug and play for a wide class of USB peripherals.

Even so, newer types of USB devices may need additional drivers installed. The Ant8 module is definitely not a standard peripheral, so it needs special drivers. However, driver installation should be relatively painless. When a Ant8 module is plugged in, the computer reads the vendor ID (VID) and product ID (PID) from the module, then searches in the computer's database for a matching driver. First time round, there will be no matching driver and the user is prompted for the driver disk. Next time the device is plugged in, the driver will be loaded automatically in seconds.

## 12. Hubs?

Most computers have two USB ports. If you need to plug in more than two USB devices you do so by using a USB hub. A hub plugs into your USB port and usually provides either four or seven ports. Many USB devices have built-in hubs - most USB keyboards have another USB port in them for daisy-chaining a mouse, joystick, or other USB device.

## 13. Self-powered and powered hubs?

USB cables distribute power as well as data - up to 500mA at 5V per cable. Low-power devices do not need separate power supplies.

An unpowered, or self-powered, hub uses some of the power coming to it for its own operation, and passes the remainder along to devices plugged into it. This is OK for small hubs with low power devices plugged into it. A good example is this is a keyboard with an powered hub built-in. This has plenty of power for plugging in a mouse or track pad.

Powered hubs have their own power supplies and can supply full power to all the devices that can physically plugged into them. Of course, you do have an extra wire and power block plugged into the wall. Some hubs can operate either way. If they don't have their power supply plugged-in, they operate as an unpowered hub - with a limited ability to power additional USB devices. When you plug-in their power supplied, they function as powered hubs.

The Ant8 module uses several hundred milliamps, particularly when running at the higher speeds. So it needs either a direct connection to a computer or a connection via a powered hub.

## 14. How fast is USB?

USB version 1.1 has a *low speed* mode operating at 1.5 Mbps and a *high speed* mode operating at 12 Mbps. The Ant8 module uses USB 1.1 high speed mode - the load it places on the USB bus is very low.

USB 2.0 is even faster - 480Mbit/s.

## 15. How long can cables be?

Five meters is the maximum cable length allowed by USB. For a longer run you need to insert a hub every five metres, or use one of the specialist extender solutions which reach up to 300 metres! The cable supplied on the Ant8 module is a little over one metre.

## Miscellaneous FAQs

**16. How do you spell Logic Analyzer?**

This is Logic Analyzer in the US. In most of the rest of the world it is Logic Analyser. We use Logic Analyzer because that is what HP/Agilent and Tektronix use.

## 9 Hardware Specification

### 9.1 Specification

This is the specification of the Ant8 logic analyzer module. Additional technical information is presented in the [FAQs](#) section.

#### Acquisition Logic

- Sample Rate: 500MHz maximum, 100Hz minimum. 5-2.5-1 sequence.
- Channels: 8, numbered 0 to 7
- Memory Depth: 3072 samples
- Threshold: 1.4V
- Input Skew: less than 2ns, channel to channel
- Input Impedance: 100Kohms in parallel with 10pF
- Maximum Input Voltages: +40V down to -40V

#### Triggering

- Conditions: 0, 1, Rising Edge, Falling Edge, Either Edge, and DON'T CARE for all channels.
- Pattern Recognisers: 2
- Edges: Trigger on the condition becoming TRUE or on it becoming FALSE
- Pass Count: 0 to 1023
- Trigger Logic: Multi-state trigger logic with Edge, Pattern, and Complex triggering. Complex triggering includes occurrence counting and minimum and maximum duration measurement.
- Trigger Position: 10% to 90% of buffer, in 10% steps

#### Environmental

- Operating Temperature: 5°C to 40°C
- Storage Temperature: -40°C to 75°C
- Size: 5 mm x 35mm x 15mm (2.5" x 1.4" x 0.6")
- Power Consumption: 1.5W maximum
- Power Source: Via USB cable. No external power supply.
- Computer Connection: USB connection to computer. Includes an attached USB cable.

#### Software Features

- Host Computer: Logic Analyzer front panel software included for Windows 98/ME/2000/XP
- Print Waveforms: Yes
- Data Output: Native text file and CSV file
- Cursors: 1 main cursor plus 1 auxiliary cursor

## 10 Control Settings

### 10.1 Modules

#### Modules

A drop-downlist of the Ant8 modules installed on this PC. The module being used is displayed. The list is refreshed every 10 seconds.

### 10.2 Module Type

#### Module Type

Should be either *8-bit ANT* or *No Module*. The [Modules](#) selector allows you to choose a module based on its USB serial number. The software then probes the module to check that it is indeed an Ant8 module, and the result of that probe is displayed here.

### 10.3 Sample Clock Speed

#### Sample Clock Speed

The speed of the internal clock which samples the incoming data. Select a value between 100Hz and 500MHz.

## 11 Trigger Settings

### 11.1 Trigger Position

#### Trigger Position

The trigger position slider divided the acquisition memory into the *pre-trigger* part and the *post-trigger* part.

The pre-trigger part of the memory is the minimum amount which must be filled with data samples before the logic analyzer starts searching for the trigger pattern.

The post-trigger part of memory is filled after the trigger pattern is detected, then the analyzer stops.

The TriggerPosition can vary from 10% to 90%, in increments of 10%.

### 11.2 Trigger Types

#### Trigger Types

The Ant8 module searches for pattern in the signals on its input probes and triggers when a pattern is recognized. The Ant8 supports three trigger types:

- [Pattern Triggering](#)
- [Edge Triggering](#)
- [Advanced Triggering](#)

Pattern and Edge triggering are simple acquisition modes, controlled via a single pattern recognizer. Advanced triggering can make use of both of the pattern recognizers in the Ant8, and a timer/counter unit.

Pattern and Edge triggering are fairly common terminology. Alternatively you can think of them as *AND* triggering and *OR* triggering.

See also the main [Triggering](#) page.

### 11.3 Pattern Triggering

#### Pattern Triggering

In Pattern triggering you specify a pattern of lows and highs, rising and falling edges, and don't cares, *all* of which must be recognized across the input channels to trigger the logic analyzer.

For instance, consider the following pattern:

```
Channel 0: rising edge
Channel 1: don't care
Channel 2: don't care
Channel 3: don't care
Channel 4: 1
Channel 5: 0
Channel 6: 1
Channel 7: 0
```

The analyzer will trigger when channels 4 to 7 see the pattern 1010 and channel 0 sees a rising edge. Note that pattern triggering is activated by the logical AND of the various conditions.

See also the main [Triggering](#) page, and [Edge Triggering](#) and [Advanced Triggering](#)

## 11.4 Edge Triggering

### Edge Triggering

In Edge triggering you specify a pattern of lows and highs, and rising and falling edges, *any* of which must be seen on the on the input channels to trigger the logic analyzer. Don't cares are ignored in edge triggering.

For instance, consider the following pattern:

```
Channel 0: rising edge
Channel 1: rising edge
Channel 2: don't care
Channel 3: don't care
Channel 4: don't care
Channel 5: don't care
Channel 6: 1
Channel 7: 0
```

The analyzer will trigger when either channel 0 sees a rising edge or channels 6 is high or channel 7 is low. Note that edge triggering is activated by the logical OR of the various conditions.

See also the main [Triggering](#) page, and [Pattern Triggering](#) and [Advanced Triggering](#)

## 11.5 Advanced Triggering

### Advanced Triggering

Triggering in the Ant8 is implemented via two pattern matchers, a timer/counter, and an internal data acquisition state machine. Advanced triggering allows you to more fully control the operation of the matchers and the state machine.

Setting up an advanced triggering data capture involves the following stages:

1. define the data patterns
2. decide whether the data pattern terms are to be ANDed together or ORed together
3. define the functions which combine the pattern
4. set the timer/counter value
5. choose the advanced trigger operation

For instance, this could be an advanced setup:

1. set the patterns as for a simple trigger
2. pattern terms are ANDed together for both P0 and P1
3.  $X0 = (P0 \text{ or } P1)$ ,  $X1 = (\text{not } P1)$
4. Timer/Counter = don't care
5. operation is X0 then X1, which is  $(P0 \text{ or } P1)$  then  $(\text{not } P1)$

See also the main [Triggering](#) page, and [Pattern Triggering](#) and [Edge Triggering](#)

## 12 Status Information

### 12.1 Pin Status

#### Pin Status

This display is updated continuously (every 100ms) when the analyzer is idle, and freezes when the analyzer is taking a reading.

The code is

▬ pin is low (reads as a 0)

▬ pin is high (reads as a 1)

↑ pin has seen a rising transition since last sampled

↓ pin has seen a falling transition since last sampled

↕ pin has seen both rising and falling transitions since last sampled

### 12.2 Program Status

#### Program Status

Shows the current operation being carried out by the Ant8 program - resetting the Module, configuring the module, searching for a pattern, downloading data, and so on.

### 12.3 Status Bar Hint

#### Status Bar Hint

Short hint, echoing the popup hints.

## 13 Cursors

### 13.1 Cursor Measurements

#### Cursor Measurements

Shows the positions of Cursor0 and Cursor1, relative to the trigger point.

Initially a dummy waveform is shown, so these readings are sample indices. When a waveform has been captured, the positions are shown in time units.

## 14 Miscellaneous

### 14.1 Mimic Diagram

#### Mimic Diagram

Shows the cursor positions and the visible element of the captured waveform.

### 14.2 Long Hint

#### Long Hint

Up to four lines of information on setting the more complex options.

# Index

## - A -

- Advance Triggering Window 3
- Analyser Menu 23
- Analyser Menu - Continuous 23
- Analyser Menu - Run 23
- Analyser Menu - Stop 23

## - C -

- Control Settings - Module Type 35
- Control Settings - Modules 35
- Control Settings - Sample Clock Rate 35
- Cursor Measurements 40
- Cursors - displaying signal levels 19

## - D -

- Driver Installation 10

## - F -

- FAQ - Logic Analyser 29
- FAQ - Miscellaneous 29
- FAQ - USB 29
- File Formats - Configuration Files 26
- File Formats - Data Files 27
- File Menu - Exit 22
- File Menu - Print 21
- File Menu - Printer Setup 21

- File Menu - Read Configuration 20
- File Menu - Read Data 20
- File Menu - Save Configuration 20
- File Menu - Save Configuration As 21
- File Menu - Save Data 21
- File Menu - Save Data As 21
- File Menu Settings 20

## - G -

- Getting Started 4

## - H -

- Hardware Specification 34
- Help Menu 25
- Help Menu - About 25

## - I -

- Installation - Software 5
- Installation - USB Drivers 10

## - M -

- Main Window 2
- Miscellaneous - Long Hint 41
- Miscellaneous - Mimic Diagram 41

## - P -

- Patterns - Defining 18
- Printer Setup 21
- Printing 21

**- Q -**

Quitting the Application 22

**- S -**

Settings - Module Type 35

Settings - Modules 35

Settings - Sample Clock Rate 35

Software Installation 5

Specification - Hardware 34

Status Bar Hint 39

Status Information - Pin Status 39

Status Information - Program Status 39

**- T -**

TriggerSettings - Advance Triggering 37

TriggerSettings - Edge Triggering 37

TriggerSettings - Pattern Triggering 36

TriggerSettings - Position 36

TriggerSettings - Types 36

Triggering 12

Triggering- Continuous 23

**- V -**

View Menu 24

View Menu - Zoom All 24

View Menu - Zoom In 24

View Menu - Zoom Out 24

**- W -**

WaveformDisplay 19

Wire Colours 17

Wire Names 17

**- Z -**

Zoom All 24

Zoom In 24

Zoom Out 24