

# LOG Storm Studio

## User's Guide

## Table of Contents

1Introduction.....	4
2Starting up a LOG Storm Studio.....	4
2.1What do you need to get started?.....	4
2.2Connecting and configuring your LOG Storm device.....	4
3Using LOG Storm for the first time.....	6
3.1LOG Storm Device at a glance.....	6
3.2What is a “Project” in LOG Storm Studio ?.....	6
3.3LOG Storm Studio interface at a glance.....	7
3.4Step-by-Step : define your project.....	8
3.4.1Setting up the 'Logging session'.....	8
3.4.2Setting up the 'trigger'.....	8
3.4.3Saving the project.....	9
3.4.4Grouping data.....	11
3.4.5Using the waveform viewer.....	12
4Using the serial protocols decoders.....	13
4.1I2C Decoder.....	13
4.1.1How to set up I <sup>2</sup> C decoder.....	13
4.1.2Capturing data.....	15
4.1.3Decoding the I <sup>2</sup> C protocol.....	18
4.2Using the SPI Decoder.....	20
4.2.1How to set up SPI decoder.....	20
4.3Decoders controls.....	24

## References

[]

## History

Version	Date	Description
1.00	22 Feb 2012	Initial revision
1.01	12 April 2012	Update for LOG Storm / LOG Storm Studio Release
1.02	20 Nov 2012	Added I2C and SPI decoders
1.03	02 Jan 2013	Update for LOG Storm Studio 1.1.4 release – Turned 'getting started guide' into 'user's guide'.

## 1 Introduction

## 2 Starting up a LOG Storm Studio

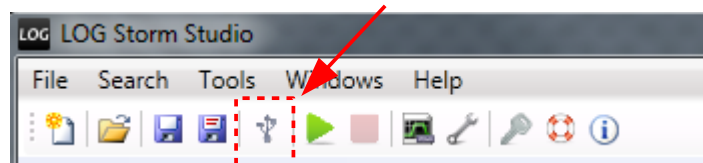
### 2.1 What do you need to get started?

*To start using LOG Storm, you need:*

- A LOG Storm Device;
- LOG Storm Studio Software;
- A PC installed with Microsoft Windows XP or Microsoft Windows 7 (32 bit or 64 bit version);
- .NET 4.0 or .NET 4.0 client profile framework installed (or a more recent version) – download link: <http://www.microsoft.com/downloads/en/details.aspx?FamilyID=e5ad0459-cbcc-4b4f-97b6-fb17111cf544>
- One free USB port.

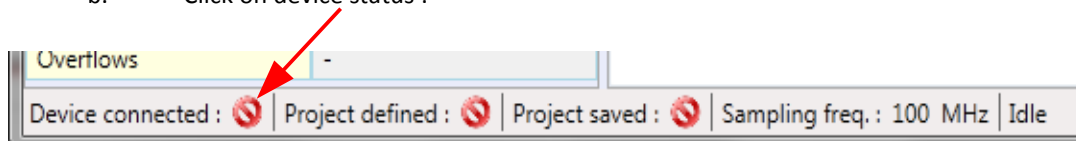
### 2.2 Connecting and configuring your LOG Storm device

- 1) Install LOG Storm Studio software, including USB driver and license file (please refer to LOG Storm Studio Installation guide)
- 2) Start LOG Storm Studio Software
- 3) Open the device selection window:
  - a. Locate and click on the 'Select Device' icon in the main toolbar:



- or:

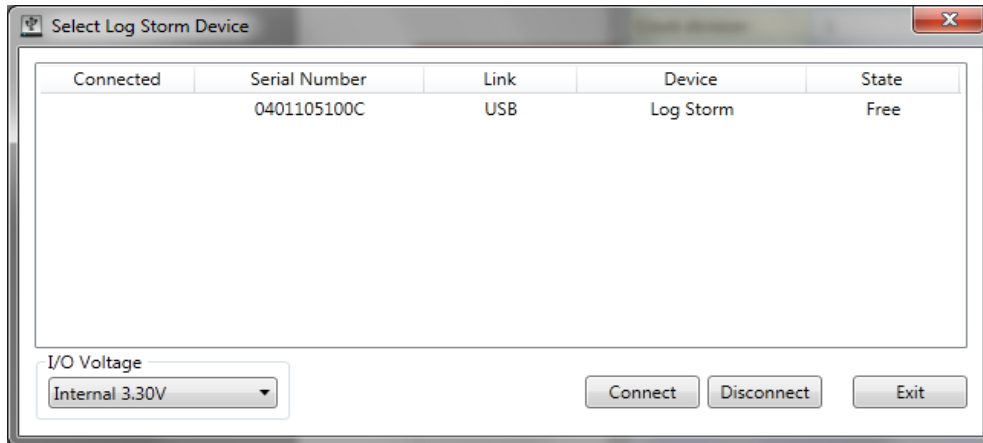
- b. Click on device status :



- or:

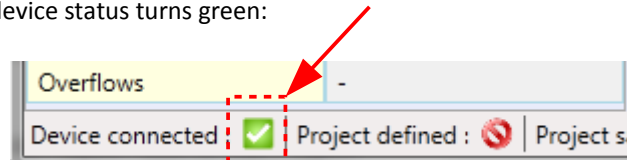
- c. Go to **Tools > Select Device**

- 4) Click on your device in the device selection window and click on button "Connect".



If your device is not physically connected to your PC or if the USB driver is not properly installed, the device won't be recognized and the list in the 'Select Device' Window will be empty.

After a few seconds, the device is properly configured and ready to be used. The 'device selection window disappears and the device status turns green:



**REMARK: LOG Storm device is now properly configured to run with a 3.3V LVCMOS signaling standard.**

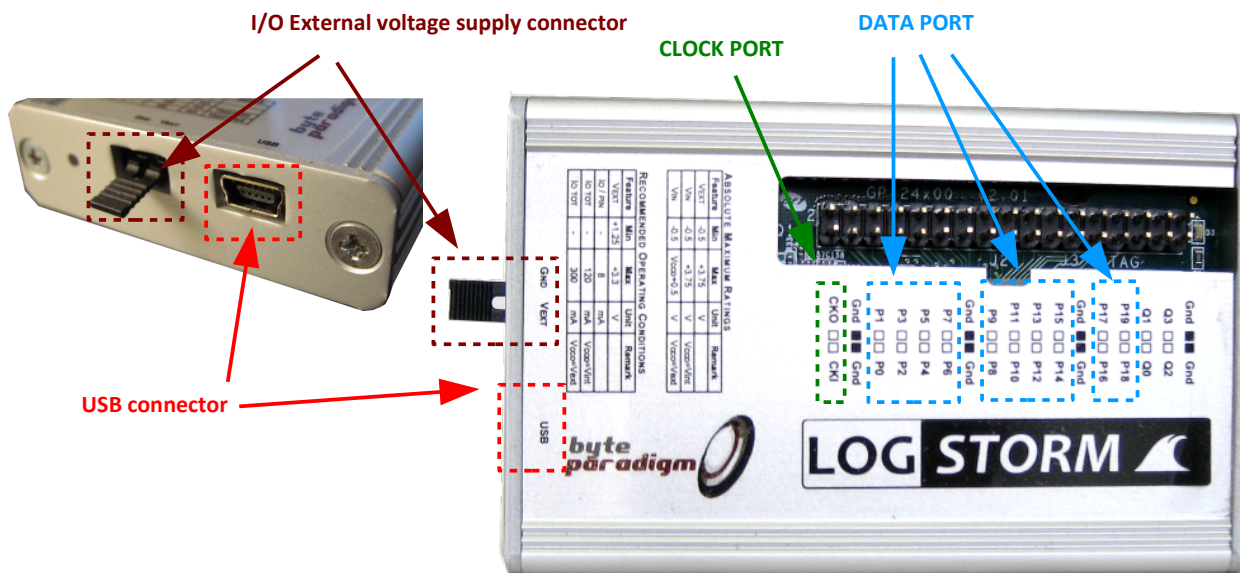
**To use a different voltage for the I/Os, you must:**

- 1) Select the right range in the 'Select LOG Storm Device' window**
- 2) Supply the proper corresponding voltage through the I/O external voltage connector.**

**! If you don't supply the external voltage, the device will automatically switch back to 3.3V voltage for the I/Os!**

## 3 Using LOG Storm for the first time

### 3.1 LOG Storm Device at a glance



There are 3 main parts to the LOG Storm device connector:

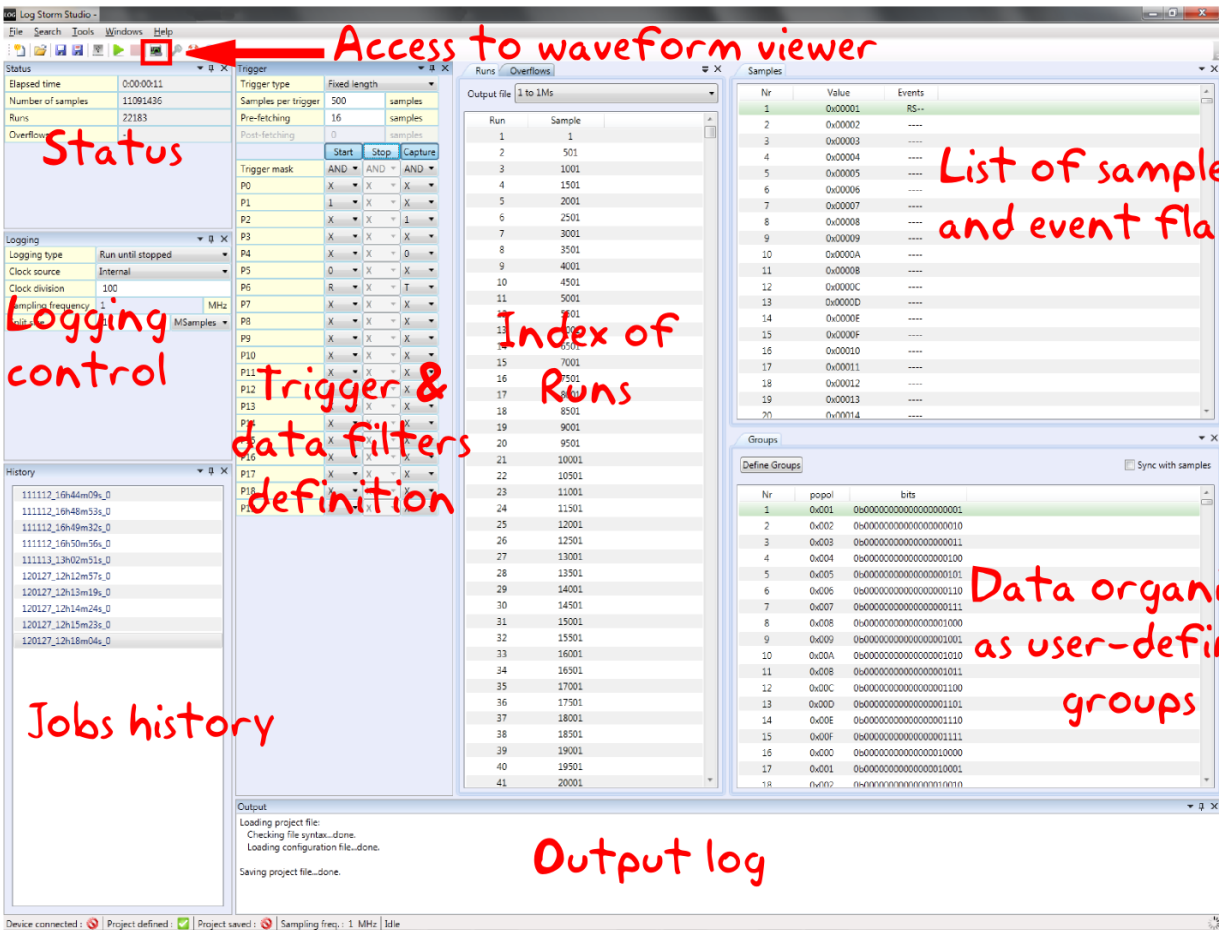
Port name	Number of I/O pins	I/O names	Purpose
<b>DATA PORT</b>	20	P0 ... P19	LOG Storm input data port. Logged data is sampled from this port.
<b>CLOCK PORT</b>	2	CKI CKO	CKO mirrors an internally generated clock signal; CKI can be used as an input pin for providing an external reference clock signal.
<b>GND pins</b>	8	GND	Ground pins – LOG Storm device and slave devices must share the same ground signals.

### 3.2 What is a “Project” in LOG Storm Studio ?

A 'Project' in LOG Storm Studio is a set of settings and definitions that can be saved as a .lsp file and later recalled. Each .lsp file encloses:

- The configuration / setup of a LOG Storm Studio session. It includes: trigger setup, logging setup, data grouping setup, ...
- Pointers to past saved trace data (logging history).

### 3.3 LOG Storm Studio interface at a glance



**Access to waveform viewer** (points to the waveform viewer icon in the toolbar)

**Status** (points to the Status panel)

**Logging control** (points to the Logging panel)

**Jobs history** (points to the History panel)

**Trigger & data filters definition** (points to the Trigger mask section)

**Index of Runs** (points to the Runs panel)

**List of samples and event flags** (points to the Samples panel)

**Data organized as user-defined groups** (points to the Groups panel)

**Output log** (points to the Output panel)

## 3.4 Step-by-Step : define your project

### 3.4.1 Setting up the 'Logging session'

Locate the 'logging' tab in LOG Storm Studio (see picture below). It has got the following parameters:

- **Logging type:** this defines how you will use LOG Storm.
  - Number of triggers: once started, LOG Storm will detect this number of triggers (see further on about 'trigger definition') and stop once this number is reached.
  - Fixed duration: LOG Storm will run during the time specified.
  - Run until stopped: LOG Storm will run until the STOP button is hit.



Logging	
Logging type	Number of triggers
Number of triggers	1 Triggers
Clock source	Internal
Clock division	1
Sampling frequency	100 MHz
Split size	1 MSamples

Logging Trigger History

- **Number of triggers:** this is where the number of triggers to be detected is specified in the corresponding logging type.
- **Clock source:** internal or external. Choose 'Internal' for using LOG Storm's internal reference clock for data sampling. Choose 'External' if you wish to supply a reference clock onto the CKI pin. External clock should be of the same voltage standard as the I/Os (by default: 3.3V) and be a continuously toggling signal applied before using the device.
- **Clock division:** a clock divider uses this value to generate the sampling clock. The sampling clock frequency will be: 100 MHz / Clock division.
- **Split size:** this specifies the maximum size of each stored file.

### 3.4.2 Setting up the 'trigger'

Locate the 'trigger' tab in LOG Storm Studio (see picture below).

2 types of 'trigger' are available:

- **'Fixed length' trigger** defines a START trigger condition and a capture length. LOG Storm starts recording data upon the detection of the START condition and collects the specified number of samples. After this, it waits until the detection of a new START condition. Events occurring before the START condition can be recorded too, by defining a PREFETCH length. In addition, expressions defining which data must be recorded can be defined (green area on the figure).
- **'Variable length' trigger** does not define a capture length, but a STOP condition and optionally a POSTFETCH length. LOG Storm data logger stops after the POSTFETCH length and waits until a new START condition. Similarly to the fixed length, a data capture condition can be defined.

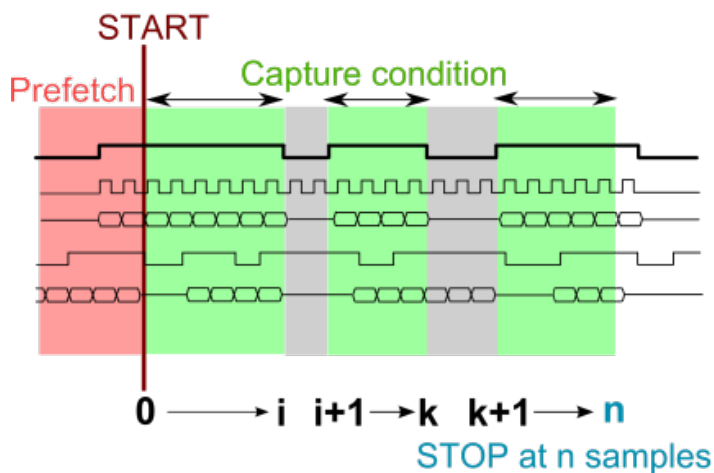
The START, STOP and CAPTURE conditions are defined as logic expressions built on the input digital signals probed on P0 to P19. To enable them, click on 'Start', 'Stop' and 'Capture' buttons in the user interface. Then, select the 'trigger mask', which defines if the values should be combined as a 'AND' or a 'OR' expression.



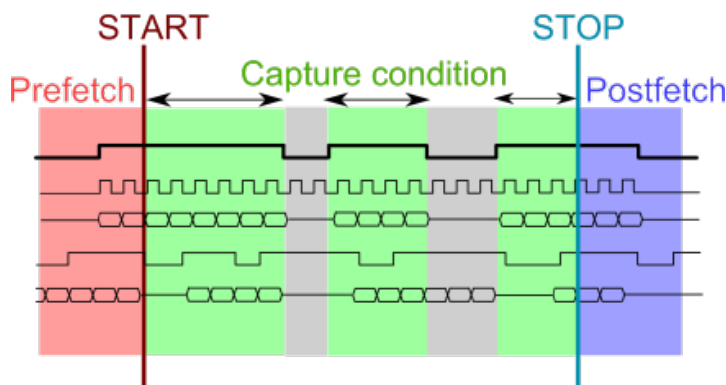
The logic values that can be detected from any of the 20 input bits are:

- **1** or **0** as high or low logic levels
- **R** or **F** as 'rising edge' or 'falling edge' to detect a transition from low to high or high to low.
- **T** for 'toggling' = any type of transition
- **X** for don't care.

## FIXED LENGTH



## VARIABLE LENGTH



Trigger			
Trigger type	Fixed length		
Samples per trigger	8	samples	
Pre-fetching	8	samples	
Post-fetching	8	samples	
	Start	Stop	Capture
Trigger mask	AND	AND	AND
P0	X	X	X
P1	X	X	X
P2	X	X	X
P3	X	X	X
P4	X	X	X
P5	X	X	X
P6	X	X	X
P7	X	X	X
P8	X	X	X
P9	X	X	X
P10	X	X	X
P11	X	X	X
P12	X	X	X
P13	X	X	X
P14	X	X	X
P15	X	X	X
P16	X	X	X
P17	X	X	X
P18	X	X	X
P19	X	X	X

Logging Trigger History

Text boxes allow entering the 'samples per trigger', pre-fetching and post-fetching, according to the chosen trigger type.

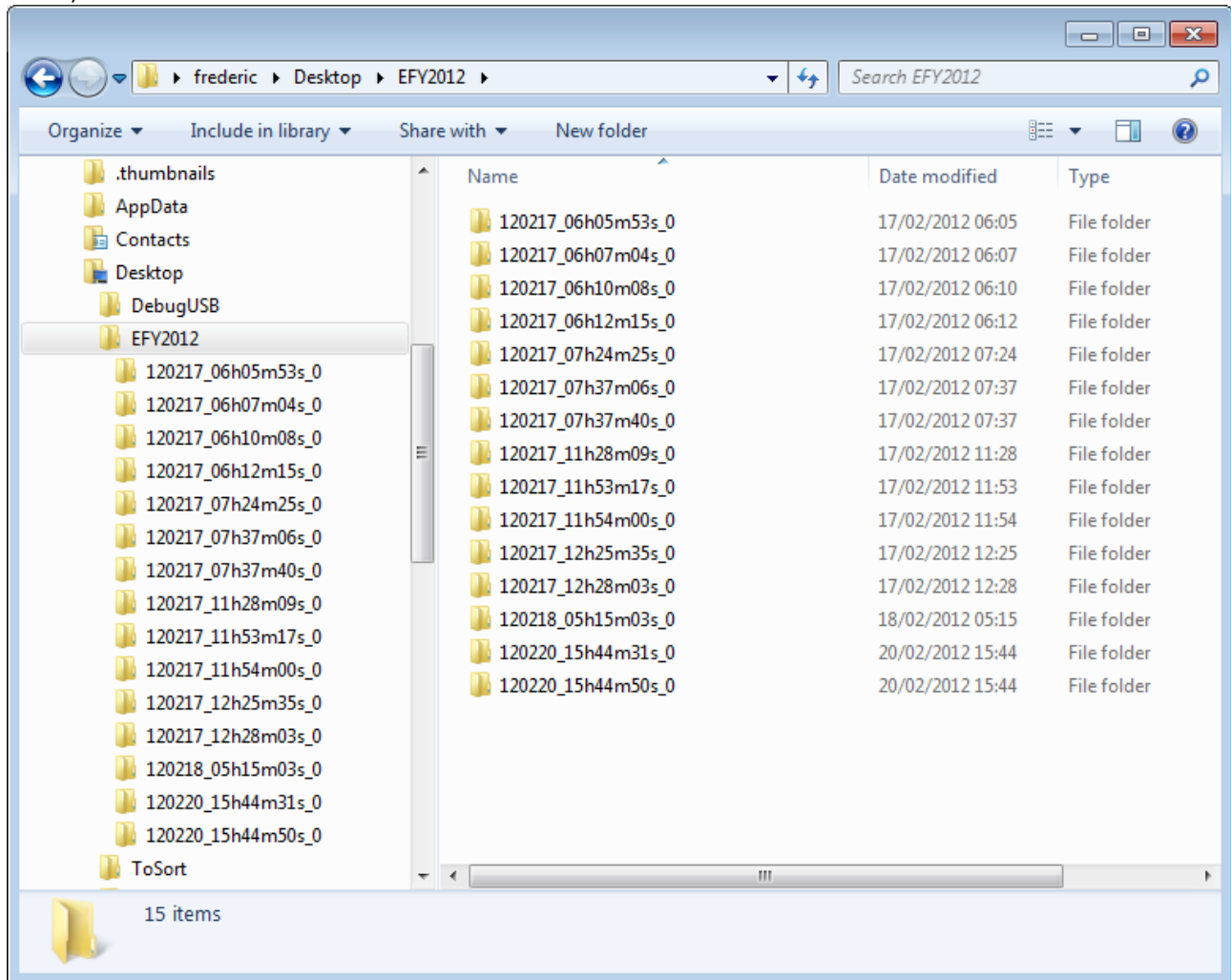
### 3.4.3 Saving the project

Once logging and trigger are defined, you can save the project by clicking onto the 'save icon', or going to the 'File' menu > Save or Save As.

Saving a project file also defines the directories where your logged data will be saved.

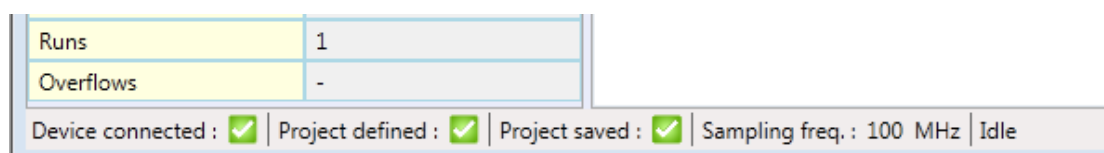
At the location where you save your project file, a directory with the name of the project will be created.

Each LOG Storm RUN will be stored in its own subdirectory. It is named after the time it was run (see screenshot below).



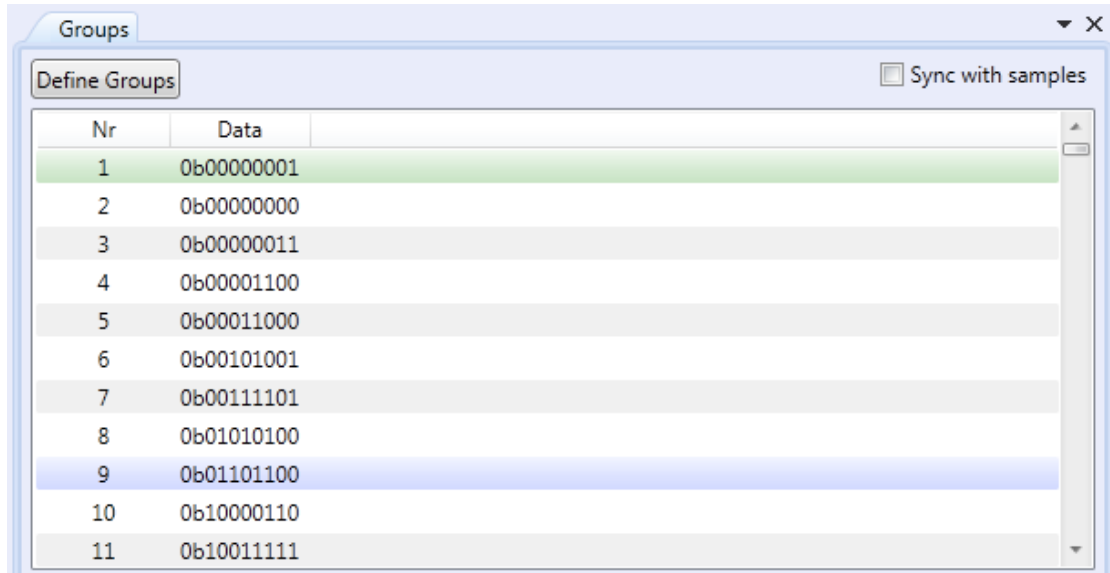
Each run directory will contain a .dat file. This is where the raw data are stored.

Once your project file is saved, you are ready for running your first capture. Status icons at the bottom of the window should turn green.

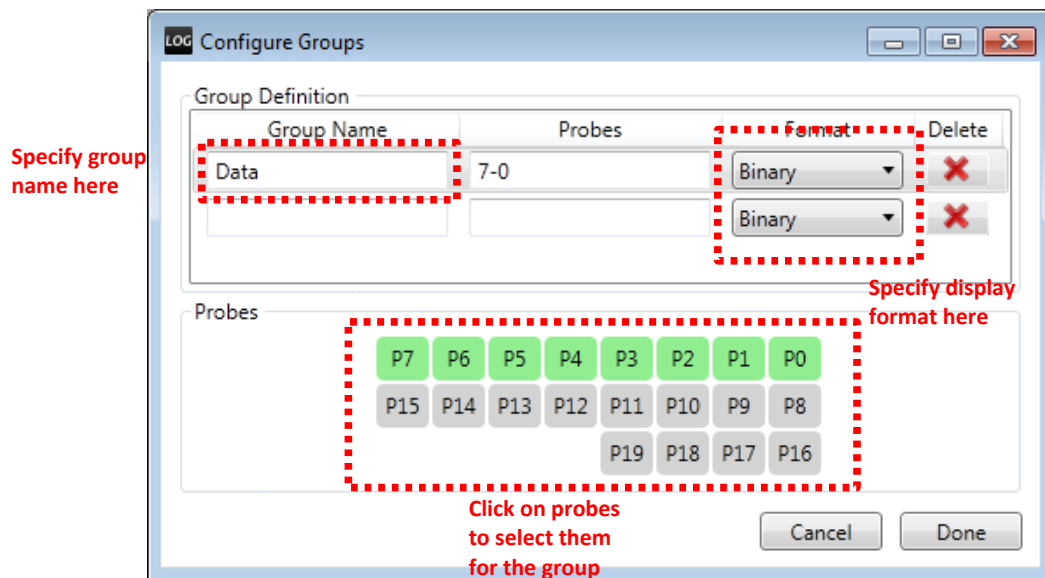


### 3.4.4 Grouping data

You can group data bits to ease readability and data organization.

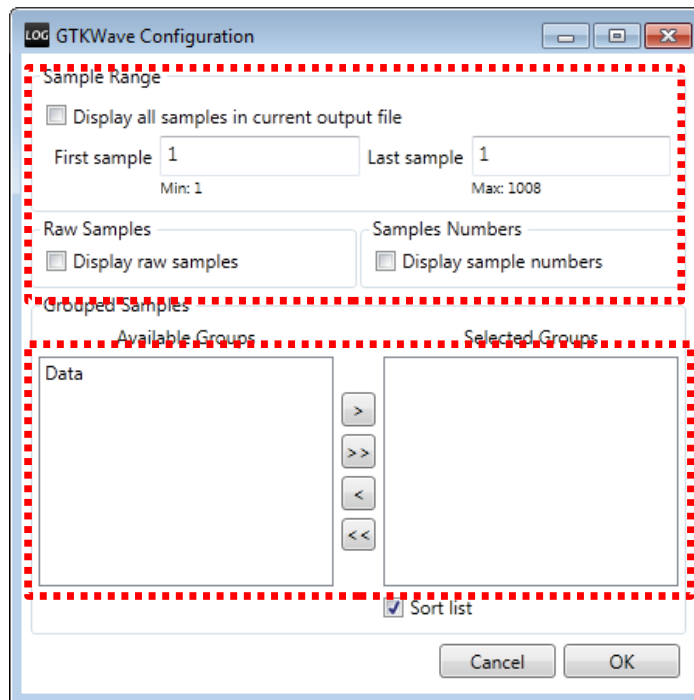


- Locate the 'Groups' tab
- Click on 'Define Groups'



### 3.4.5 Using the waveform viewer

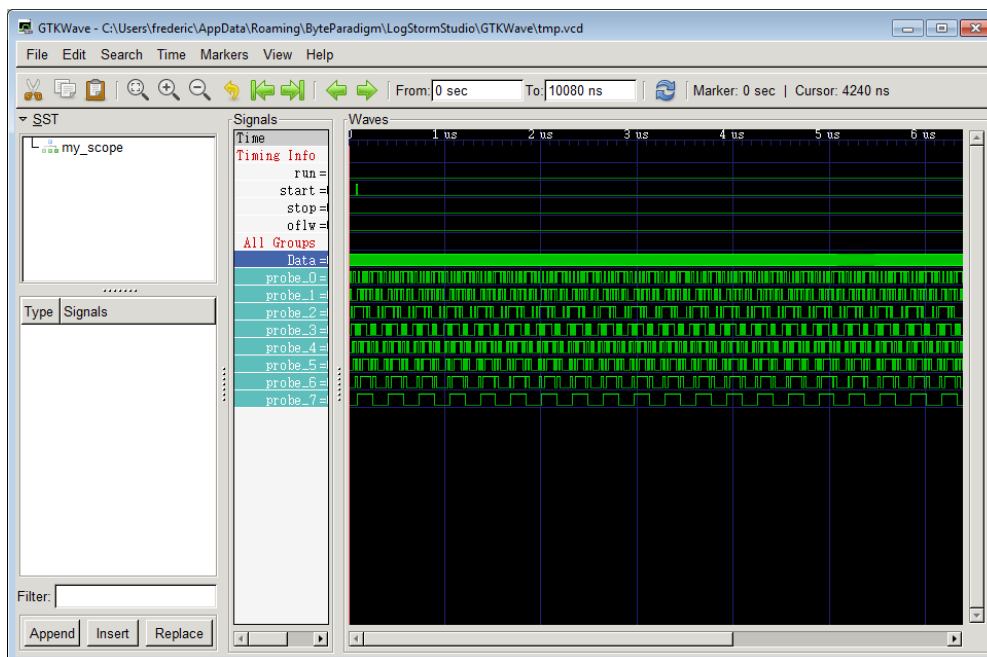
Locate and click on GTK Wave icon.



Select the range to display

Add the groups that you want to display

Once the file to display is generated, you GTK Wave viewer opens:

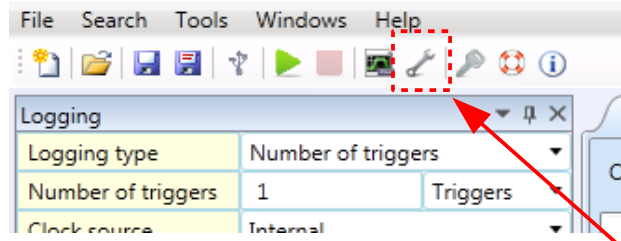


## 4 Using the serial protocols decoders

### 4.1 I2C Decoder

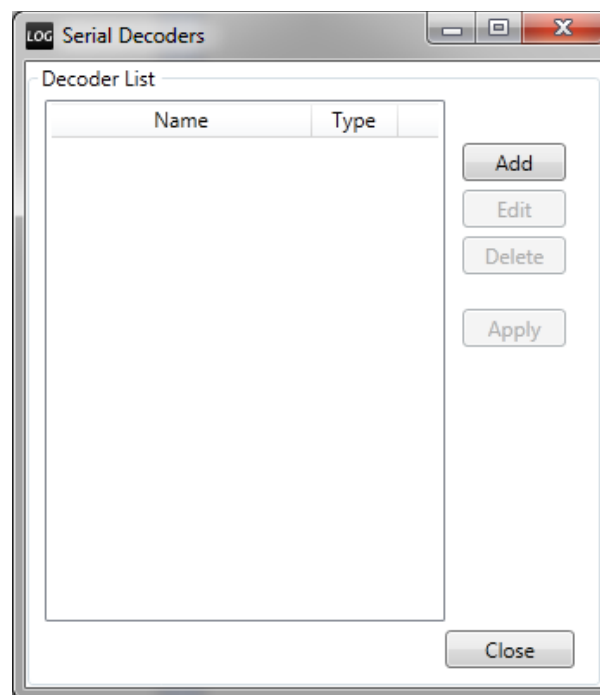
#### 4.1.1 How to set up I<sup>2</sup>C decoder

- Open the 'Decoders window' by clicking on its button in the toolbar.

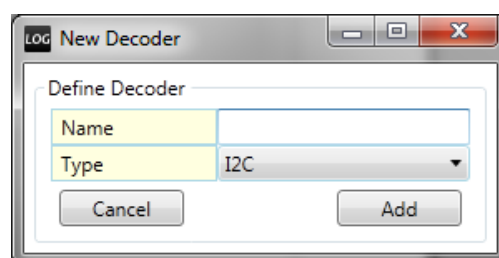


'Decoders window' button

A window opens:

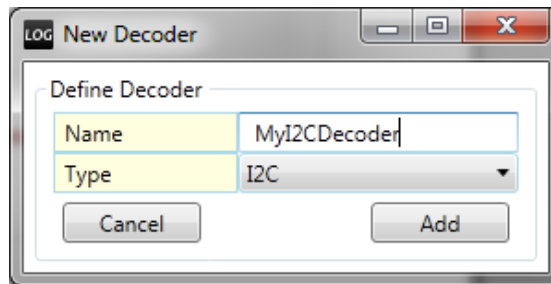


This window gives the list of the existing decoders. A 'decoder' is used to decode a specific protocol from specific LOG Storm Studio inputs. In this case, a **I<sup>2</sup>C protocol decoder** will be defined. Click on 'Add' button.

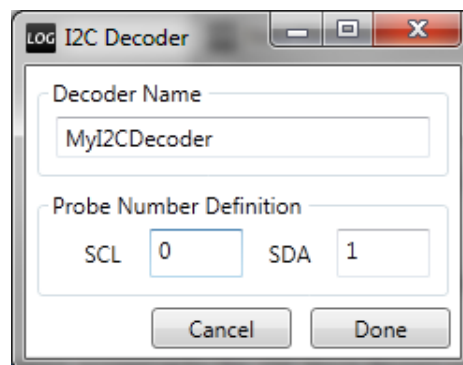


The 'New Decoder' window appears and prompts for the decoder's characteristics. The following items must be specified:

- Decoder name: this is a user-defined name that will be used to specifically recognize the defined decoder during LOG Storm Studio operation.
- Type: protocol type. Choose I2C.
- Click on 'Add' button.

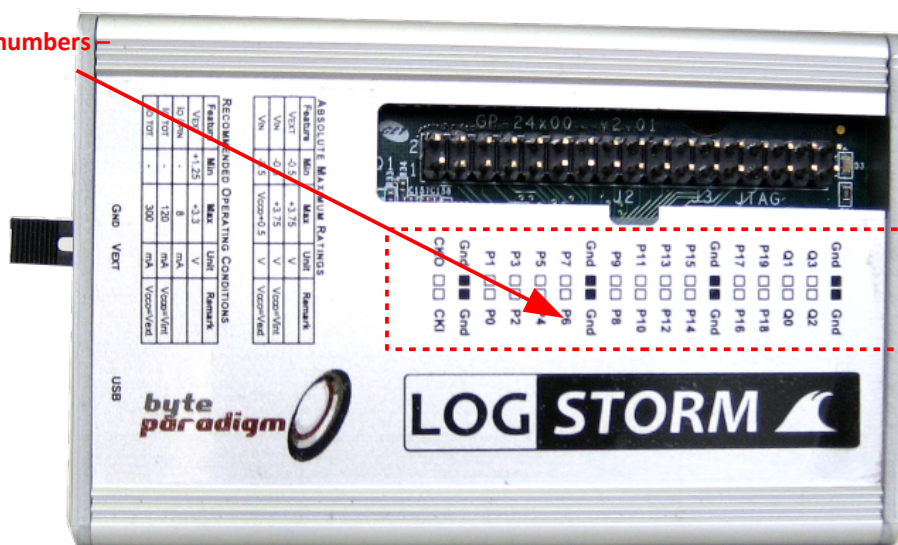


A new window appears, prompting for the LOG Storm device's probes used for the decoding. For I<sup>2</sup>C protocol, only 2 probes will be used: one probe connected to SCL and the other one connected to SDA. We choose P0 and P1 and specify it in this window.



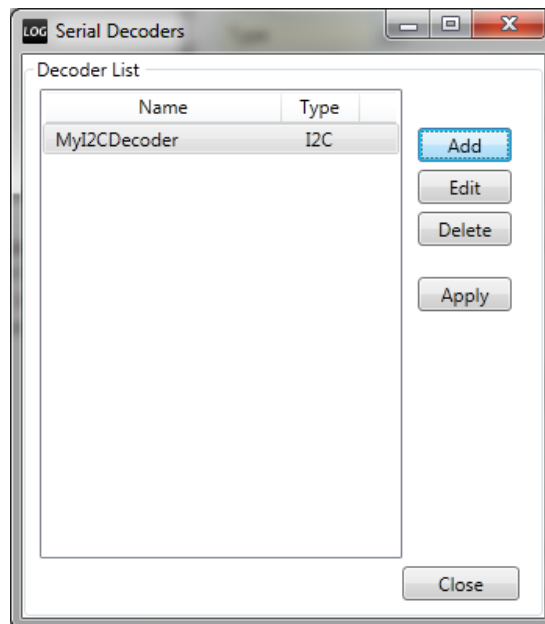
Probe's numbering can be found on the connector label of LOG Storm device:

Label with probe numbers –  
P0 to P19.



Click on '**Done**' button.

Now, the I<sup>2</sup>C decoder setup is over, and it appears in the 'Serial Decoders' window.



Now click on 'Close'.

**Remark:** serial decoders can be defined before or after data capture, as they analyze sets of data captured with LOG Storm data logger / logic analyzer device.

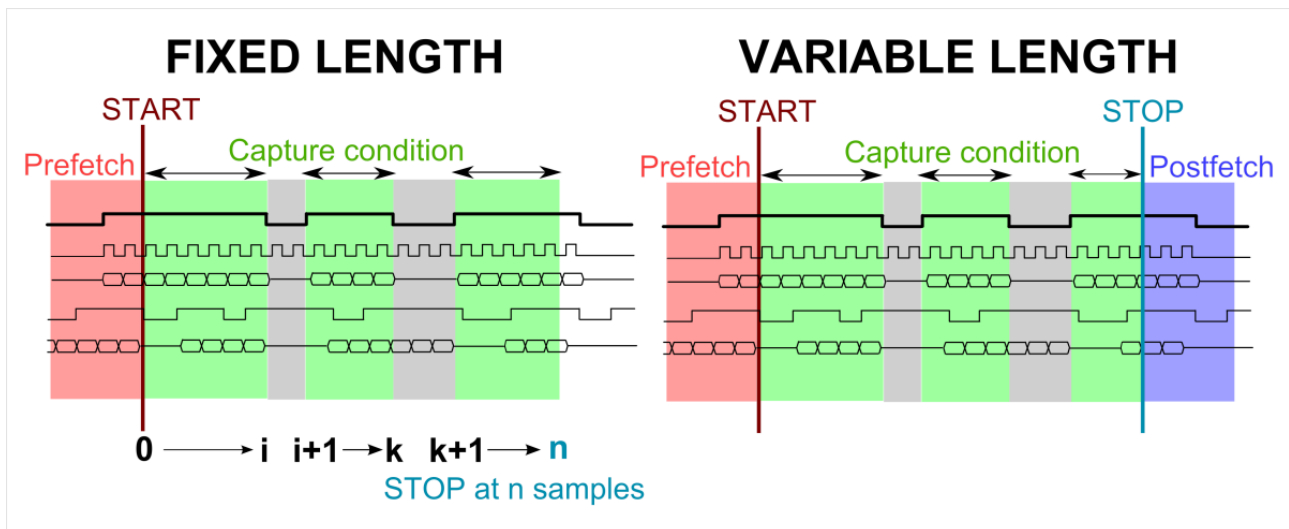
#### 4.1.2 Capturing data

I<sup>2</sup>C decoders defined in LOG Storm Studio decode the I<sup>2</sup>C protocol based on raw data sampled with LOG Storm Studio. To capture the raw data, LOG Storm Studio must be set up and run.

Specifically, the session settings, trigger and capture characteristics must be set up. Please refer to LOG Storm documentation for more details about this. In this guide, we'll propose a simple step-by-step example of such a setup.

We suppose we intend to capture I<sup>2</sup>C bus traffic. We would like to start capturing data when we detect the first 'start condition' on the I<sup>2</sup>C bus. Once this condition is detected, we would like to collect a buffer of 10.000 samples, which we estimate would be sufficient to see the I<sup>2</sup>C traffic and decode what we would like to decode.

**Remark:** LOG Storm's full trigger and capture capabilities can be used here (see below for an overview). LOG Storm Studio's trigger and capture definition can be more complex in order to properly filter the bus traffic that you're interested in.



Picture 1: LOG Storm Studio trigger definition - overview

#### 4.1.2.1 Session setup ('Logging subtab' in the main window):

Logging	
Logging type	Number of triggers
Number of triggers	1
Clock source	Internal
Clock division	10
Sampling frequency	10 MHz
Split size	1 MSamples

Here are the parameters of the capture:

<b>Logging Type</b>	Number of trigger	Data capture is based in trigger condition (and not capture length)
<b>Number of triggers:</b>	1	We would like to detect the trigger condition one time.
<b>Clock source</b>	Internal	We use LOG Storm device internal clock reference to sample data.
<b>Clock division</b>	10	This sets up the sampling clock to 10 MHz (100 MHz / 10). 10 MHz is sufficient to sample the I <sup>2</sup> C bus, which is supposed to toggle at 1 Mbit/s
<b>Split size</b>	1 Msample	We keep the default setting for the file size.



### 4.1.2.2 Trigger setup

**Fixed length trigger:** upon occurrence of the trigger condition, we'll collect a fixed number of samples and then stop.

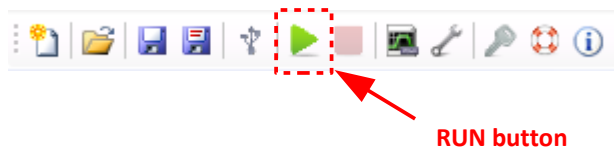
**Samples per trigger:** set to 10000

**Pre-fetching:** we leave this to the default 8 samples.

**Trigger:** trigger is set up to detect a falling edge on P1 while P0 is held at 1 – which corresponds to a 'START condition' on the I<sup>2</sup>C bus.

### 4.1.2.3 Capturing data

Having connected LOG Storm to the I<sup>2</sup>C bus, click on **run button** to capture data:



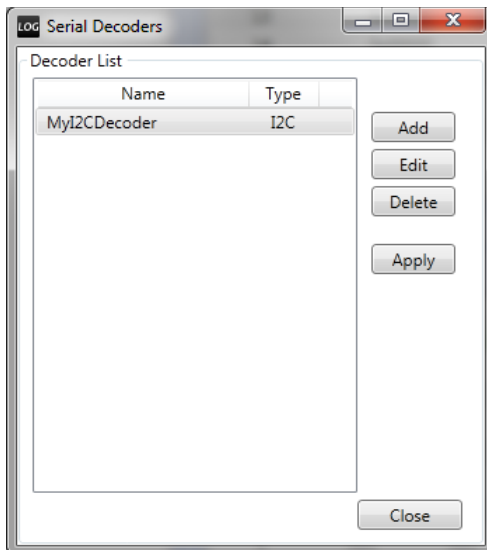
Having defined 'groups' on P0 and P1, we can see that traffic has been captured (refer to 'Getting started guide to know how to define groups').

Groups			
Define Groups			
Nr	SCL	SDA	
1	0b1	0b1	
2	0b1	0b1	
3	0b1	0b1	
4	0b1	0b1	
5	0b1	0b1	
6	0b1	0b1	
7	0b1	0b1	
8	0b1	0b1	

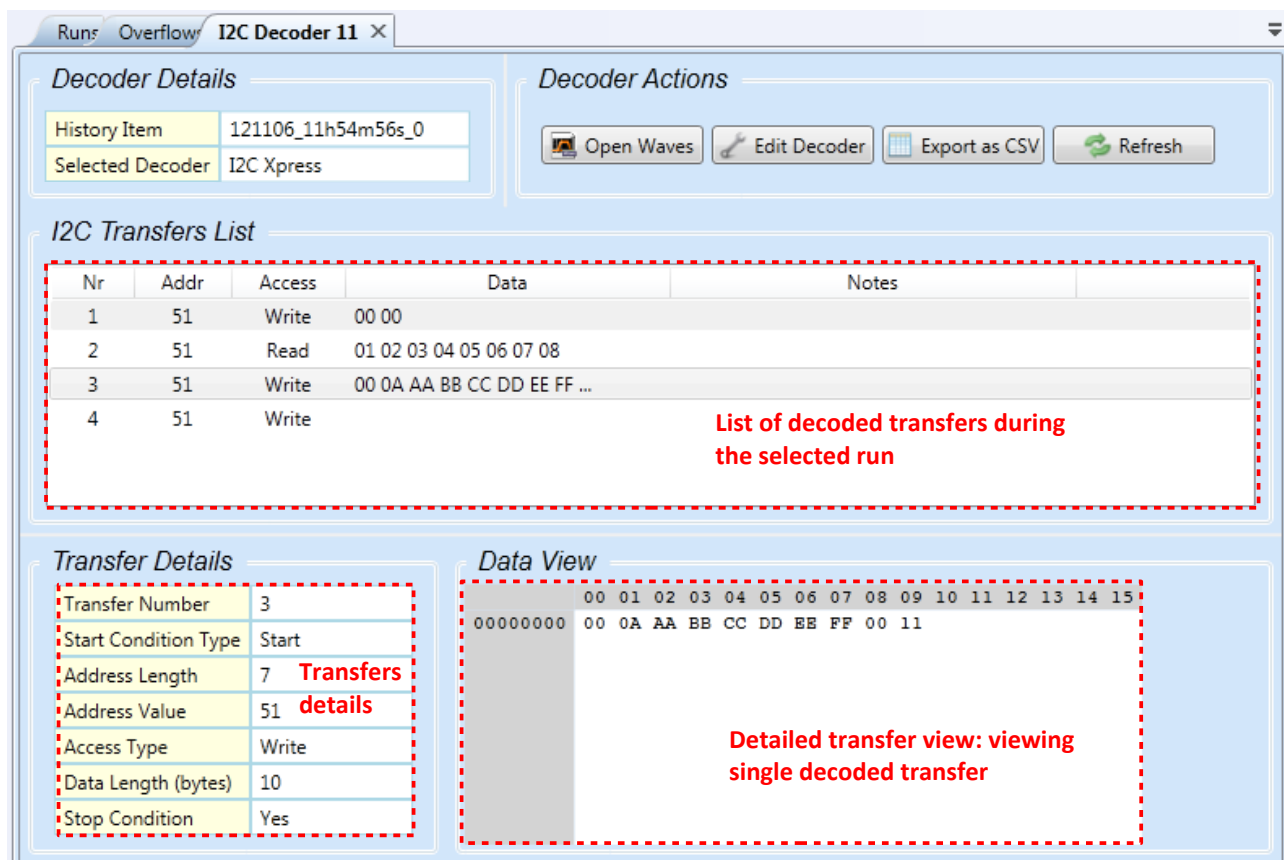
Trigger			
Trigger type	Fixed length		
Samples per trigger	10000	samples	
Pre-fetching	8	samples	
Post-fetching	8	samples	
	Start	Stop	Capture
Trigger mask	AND	AND	AND
P0	1	X	X
P1	F	X	X
P2	X	X	X
P3	X	X	X
P4	X	X	X
P5	X	X	X
P6	X	X	X
P7	X	X	X
P8	X	X	X
P9	X	X	X
P10	X	X	X
P11	X	X	X
P12	X	X	X
P13	X	X	X
P14	X	X	X
P15	X	X	X
P16	X	X	X
P17	X	X	X
P18	X	X	X
P19	X	X	X

### 4.1.3 Decoding the I<sup>2</sup>C protocol

Open the 'Serial decoders window again, select the decoder and click on 'Apply'.



The following tab opens:



**Decoder Details**

History Item	121106_11h54m56s_0
Selected Decoder	I2C Xpress

**Decoder Actions**

Open Waves Edit Decoder Export as CSV Refresh

**I2C Transfers List**

Nr	Addr	Access	Data	Notes
1	51	Write	00 00	
2	51	Read	01 02 03 04 05 06 07 08	
3	51	Write	00 0A AA BB CC DD EE FF ...	
4	51	Write		

List of decoded transfers during the selected run

**Transfer Details**

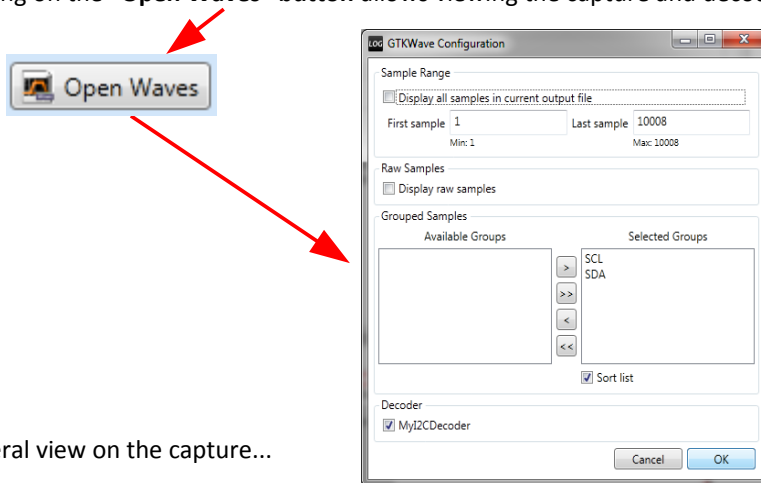
Transfer Number	3
Start Condition Type	Start
Address Length	7
Address Value	51
Access Type	Write
Data Length (bytes)	10
Stop Condition	Yes

**Data View**

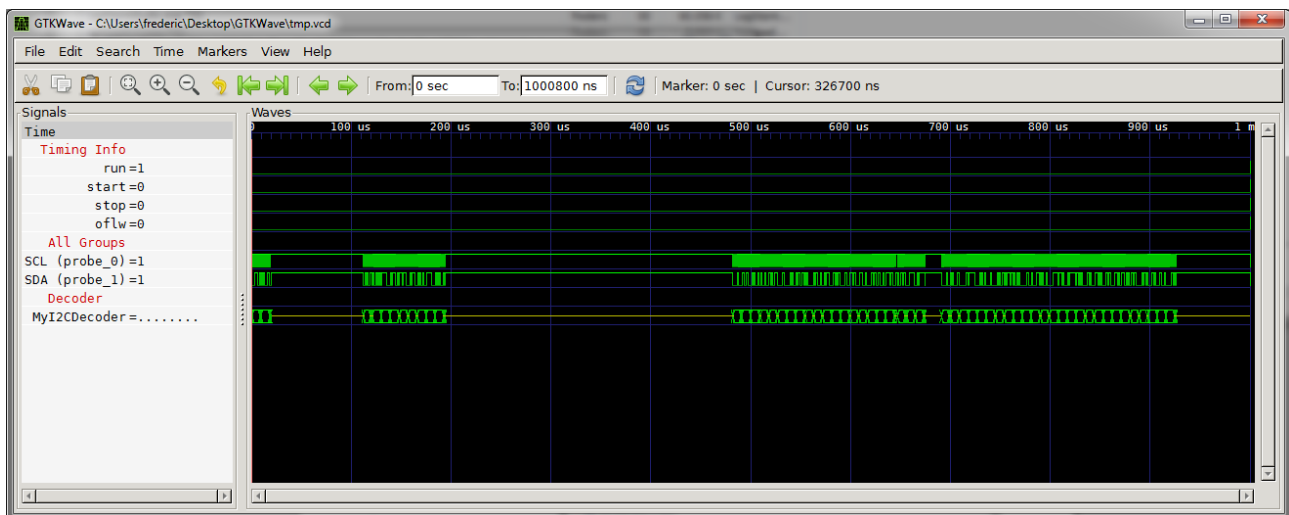
	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
00000000	00	0A	AA	BB	CC	DD	EE	FF	00	11						

Detailed transfer view: viewing single decoded transfer

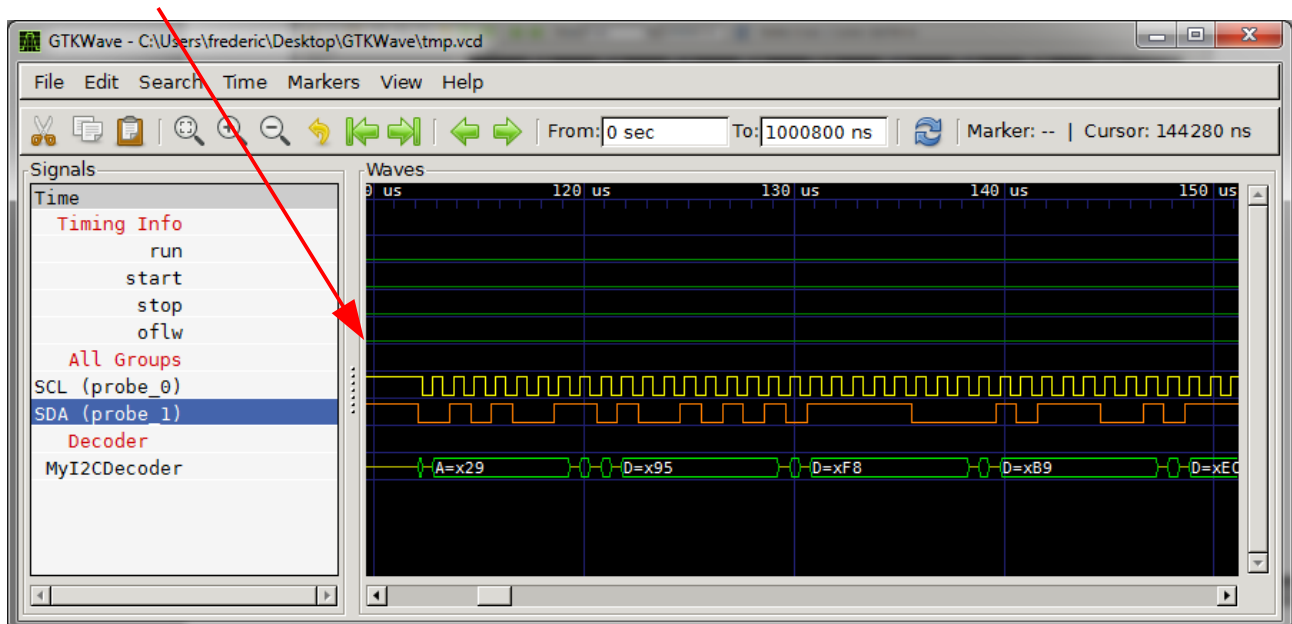
Clicking on the **"Open Waves"** button allows viewing the capture and decoding:



General view on the capture...



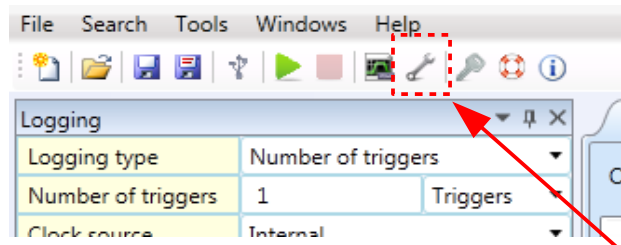
Zoom on a specific transfer.



## 4.2 Using the SPI Decoder

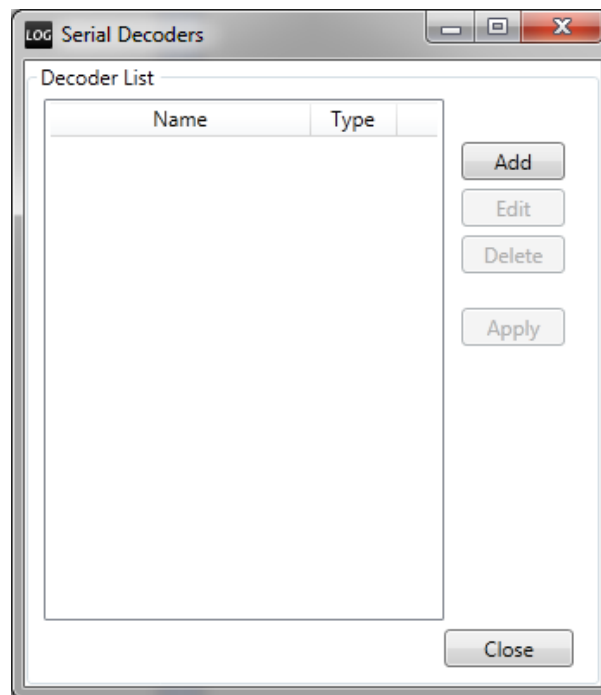
### 4.2.1 How to set up SPI decoder

- Open the 'Decoders window' by clicking on its button in the toolbar.

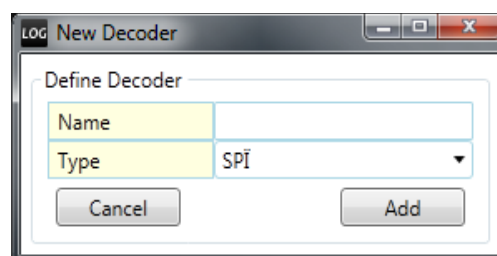


'Decoders window' button

A window opens:

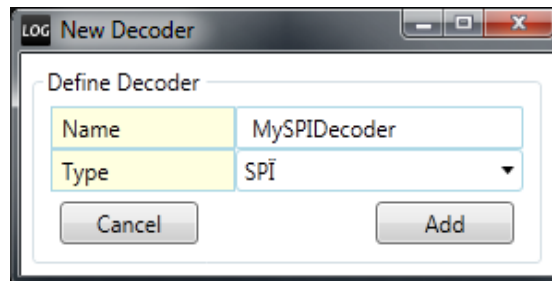


This window gives the list of the existing decoders. A 'decoder' is used to decode a specific protocol from specific LOG Storm Studio inputs. In this case, a **SPI protocol decoder** will be defined. Click on 'Add' button.

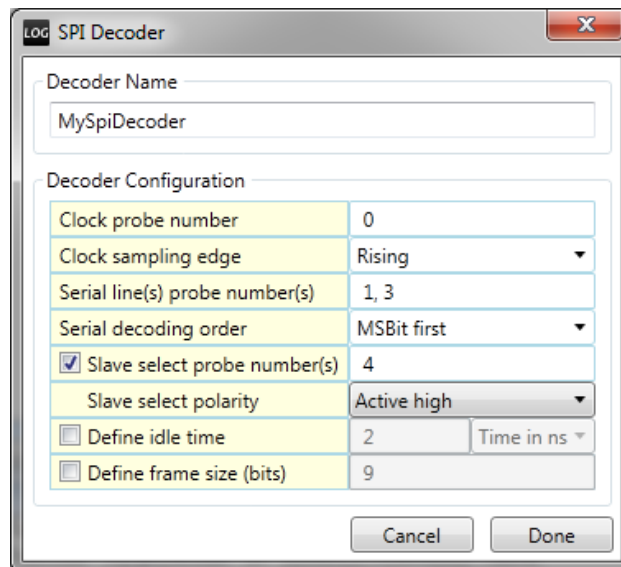


The 'New Decoder' window appears and prompts for the decoder's characteristics. The following items must be specified:

- Decoder name: this is a user-defined name that will be used to specifically recognize the defined decoder during LOG Storm Studio operation.
- Type: protocol type. Choose SPI.
- Click on '**Add**' button.

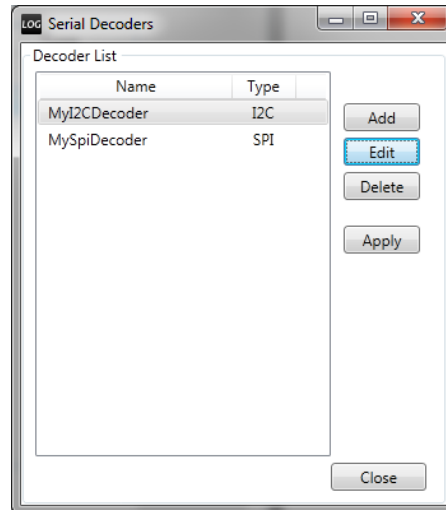


A new window appears, prompting for the LOG Storm device's probes and settings used for the decoding.



Parameter	Description
Clock probe number	Defines the probe number on which the SPI clock (SCLK) signal is connected.
Clock sampling edge	Specifies on which edge of SCLK dat must sampled and decoded.
Serial line(s) probe number(s)	Specifies which probes should be decoded as serial data. Multiple lines can be decoded, in which case they are separated with commas.
Slave select probe number(s)	Defines the slave select line – if applicable. This enables the decoder to find the accesses boundaries. Multiple probes can be used, in which case they are separated with commas.
Slave select polarity	(Active when a slave select probe is specified). Defines the active level of the slave select line.
Define idle time	(Alternative to using slave select probe).
Define frame size	(Alternative to using slave select and/or idle time): defines the size of one frame on the data line. The serial data will be decoded in groups of this size.

Once defined, the decoder is applied by clicking the 'Apply' button.



Runs Overflow **SPI Decoder 9** X

**Decoder Details**

History Item: 120925\_17h54m42s\_0  
Selected Decoder: MySpiDecoder

Open Waves Edit Decoder Export as CSV Refresh

**SPI Transfers List**

Nr	Length (bits)	Selected Slaves	Data
1	9	SS0	MOSI = 93 80 WE = 00 00
2	9	SS0	MOSI = 3C 80 WE = 00 00
3	9	SS0	MOSI = 93 80 WE = 00 00
4	9	SS0	MOSI = 3C 80 WE = 00 00
5	9	SS0	MOSI = 93 80 WE = 00 00
6	9	SS0	MOSI = 3C 80 WE = 00 00
7	1	SS0	MOSI = 00 WE = 00

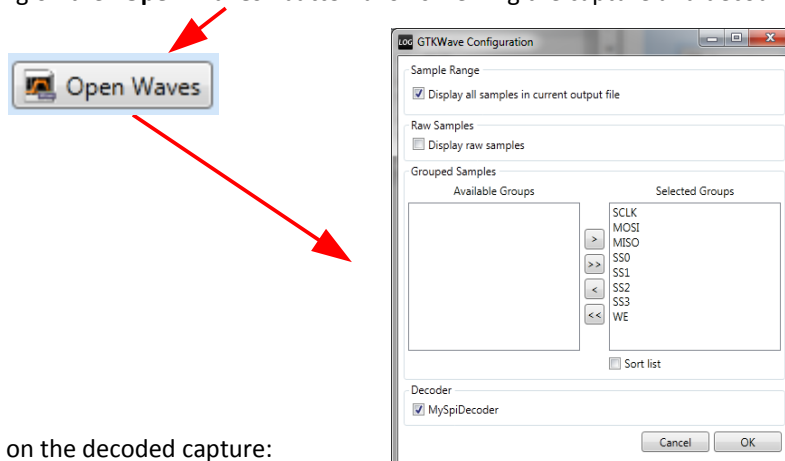
**List of decoded transfers during the selected run**

**Data View**

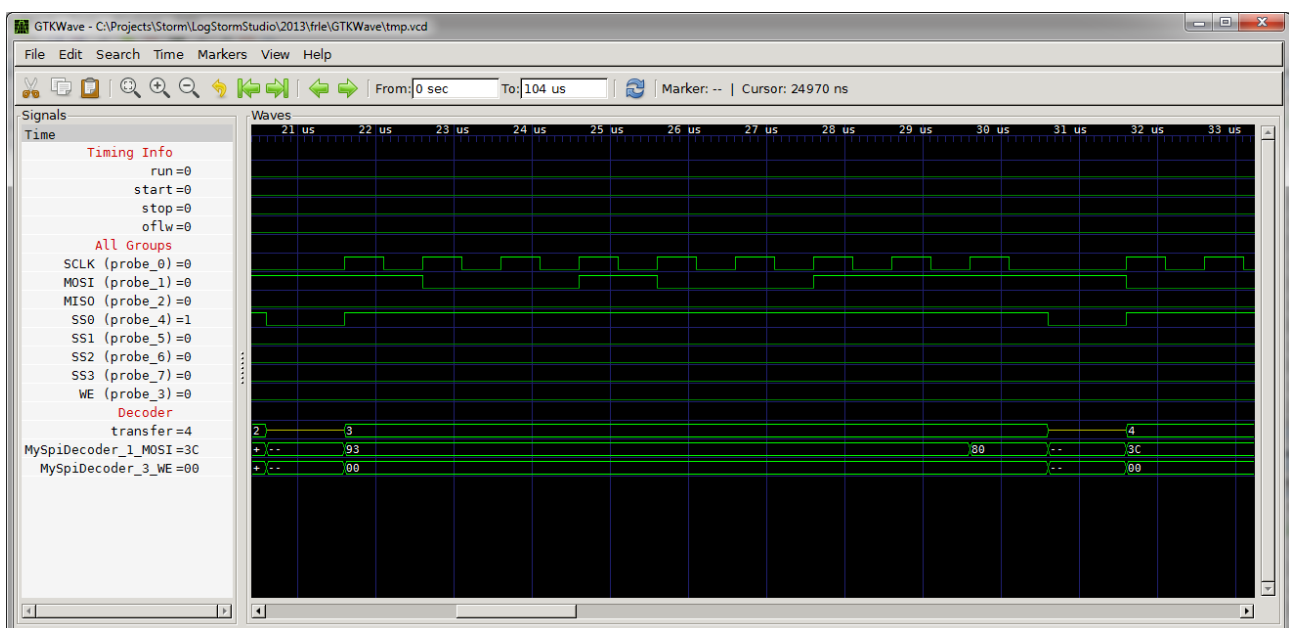
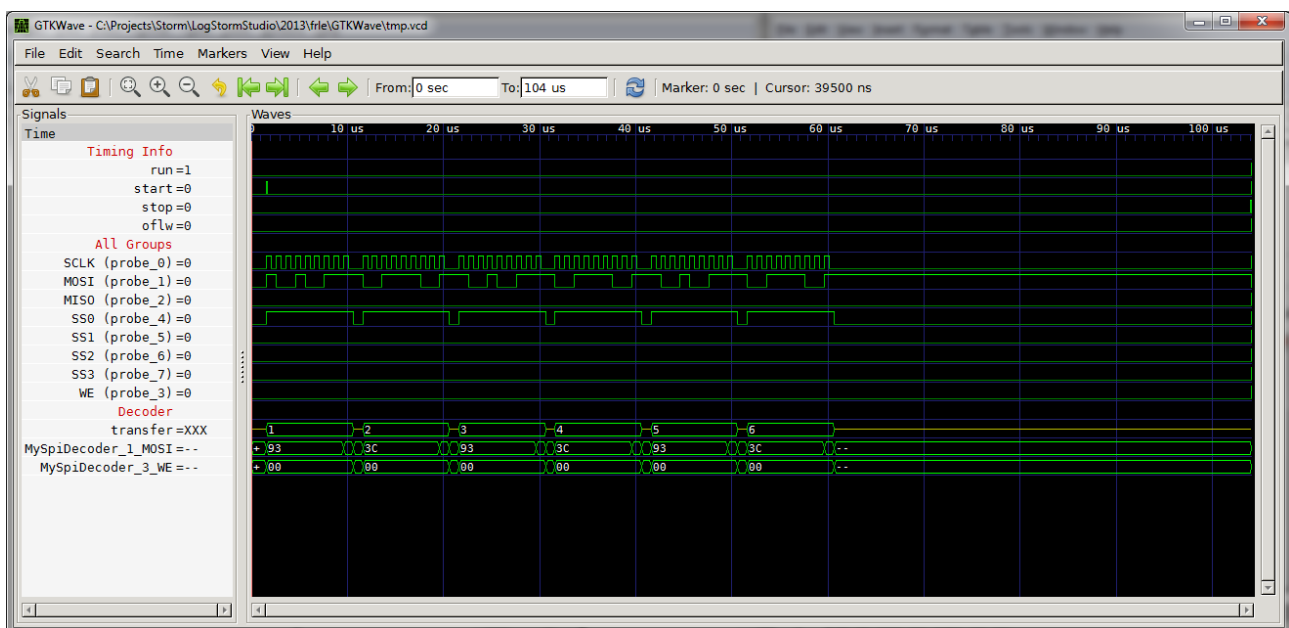
00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15  
00000000 93 80

**Detailed transfer view: viewing single decoded transfer**

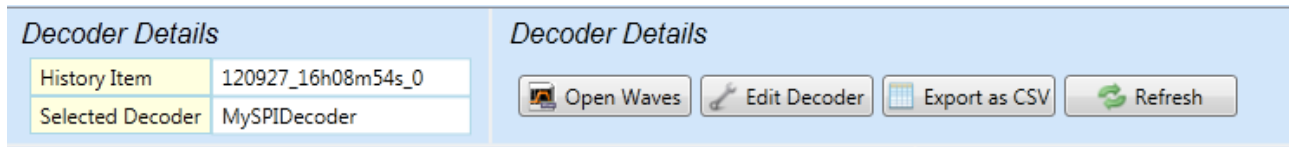
Clicking on the **"Open Waves"** button allows viewing the capture and decoding:



View on the decoded capture:



## 4.3 Decoders controls



**Decoder Details:** provides details on the applied decoder and the data set used for decoding.

**Open Waves :** opens the waveform viewer.

**Edit Decoder :** allows changing the characteristics of the decoder.

**Export as CSV :** allows exporting the decoded data to a comma-separated value file.

**Refresh :** Forces refreshing the decoded data (e.g.: when the characteristics of a decoder have been updated).