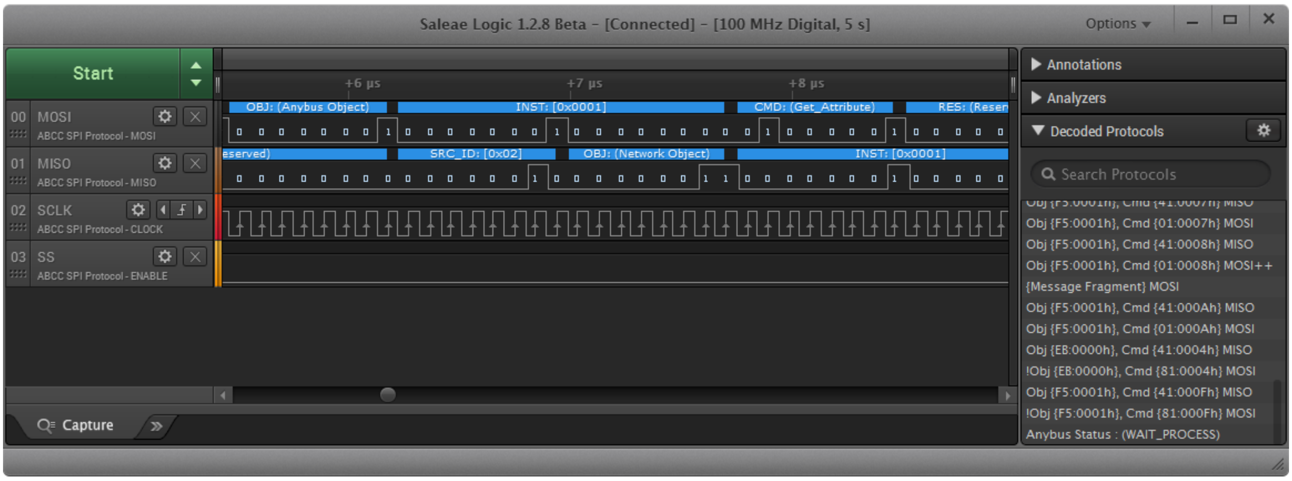
Quick Start Guide

ABCC SPI Messaging Protocol Analyzer:

A Plugin for Saleae Logic Software



```
Obj {F5:0001h}, Cmd {41:0007h} MISO
Obj {F5:0001h}, Cmd {01:0007h} MOSI
Obj {F5:0001h}, Cmd {41:0008h} MISO
Obj {F5:0001h}, Cmd {01:0008h} MOSI++
{Message Fragment} MOSI
Obj {F5:0001h}, Cmd {41:000Ah} MISO
Obj {F5:0001h}, Cmd {01:000Ah} MOSI
Obj {EB:0000h}, Cmd {41:0004h} MISO
!Obj {EB:0000h}, Cmd {81:0004h} MOSI
Obj {F5:0001h}, Cmd {41:000Fh} MISO
!Obj {F5:0001h}, Cmd {81:000Fh} MOSI
Anybus Status : (WAIT_PROCESS)
```

HMS Industrial Networks Inc. Rev. 1.01

35 E. Wacker Dr Suite 1700
Chicago, IL 60601
+ 1 (312) 829 0601

120 Bedford Center Rd
Bedford, NH 03110
+1 (603) 471 0800

Web:
www.anybus.com
www.ixxat.com
www.netbiter.com

History

Revision	Date	Description	Author
1.00	4/27/2016	Initial release	JoCa

Table of content

1 Introduction

This quick start guide documents the installation, features, issues, and general use of the ABCC SPI Analyzer plugin for Saleae Logic Software.

1.1 Description

With the use of the compiled DLL (plugin) obtained from this project, a developer can leverage the use of a Saleae 'Logic' to analyze bi-directional ABCC SPI communication. This plugin parses SPI communication according to the ABCC SPI messaging protocol. Each field within the SPI telegram is added as a multi-layered bubble-text within the Logic software. This means a basic markup is displayed when zoomed-out and very detailed information when zoomed-in on the capture. By coupling this protocol analyzer with other physical signals from the target device (such as CAN, Serial, GPIO, etc.) this tool can help solve difficult-to-diagnose issues that arise during initial development efforts.

2 Software Overview

2.1 System Requirements

Recommended:

- Saleae Logic (software) version 1.2.8
- Saleae Logic analyzer (hardware)
- PC running Windows 7
- Precompiled AbccSpiAnalyzer64.dll or AbccSpiAnalyzer.dll plugin
    - File version 1.0.0.0
    - Product version 1.1.32.1-beta
        - NOTE: This product version is a combination of the Analyzer SDK version this DLL is based on and represents a level of compatibility with the given Logic software. In this case this DLL was built on the SDK version 1.1.32. The remaining portion of this version is an HMS specified release indicator which will reset with each update of the fundamental SDK framework. In this case since this is the first release of an experimental plugin and the Saleae Logic software is still in beta, the release indicator is “1-beta”.

Other versions of Windows or Logic software may work but have not been tested.

2.2 Installation

To install this plugin, it is assumed that the user has previously installed the Saleae Logic software using the default installation options.

__**For 64-bit versions of Windows**__:

Please locate and copy the “x64” version of the DLL found in:

“.\plugins\Win64\AbccSpiAnalyzer64.dll”

And copy to:

“C:\Program Files\Saleae LLC\Analyzers\”

__**For 32-bit versions of Windows**__:

Please locate and copy the “x86” version of the DLL found in:

“.\plugins\Win32\AbccSpiAnalyzer.dll”

And copy to:

“C:\Program Files\Saleae LLC\Analyzers\”

If properly installed, the analyzer will appear in the “Analyzers” sub-window’s “add” option (+) as shown in the figure below. Please note: it may be necessary to first click “Show more analyzers” option before it is displayed.

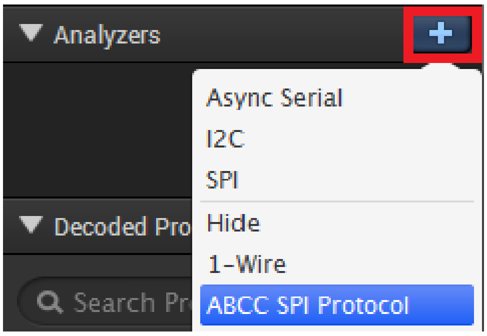
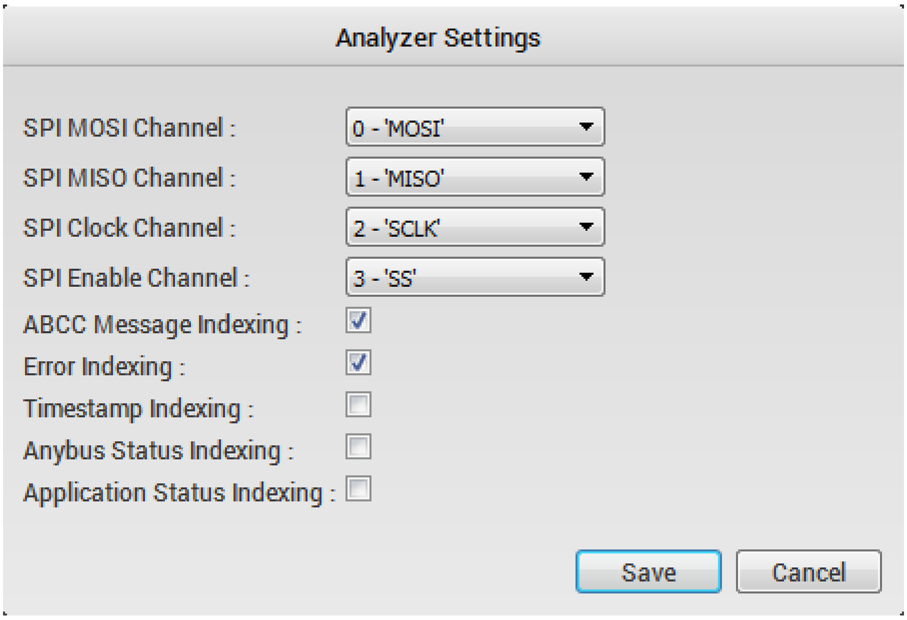


Figure 1: Adding the analyzer

After the analyzer has been selected, the settings window will be displayed as shown in the figure below. The default indexing options provide the most basic and, perhaps, most commonly used options, but can be changed according to your needs.



Analyzer Settings

SPI MOSI Channel : 0 - 'MOSI'
SPI MISO Channel : 1 - 'MISO'
SPI Clock Channel : 2 - 'SCLK'
SPI Enable Channel : 3 - 'SS'
ABCC Message Indexing : ☑
Error Indexing : ☑
Timestamp Indexing : ☐
Anybus Status Indexing : ☐
Application Status Indexing : ☐

Figure 2: Analyzer settings window

Select the corresponding channels that are associated with the physical MOSI, MISO, Clock (SCLK), and Enable (CS/SS) lines to the ABCC. Once satisfied with the remainder of the settings, click “Save”. The analyzer is now ready to be used in a capture. For more detailed information of the analyzer plugin’s settings, capture results, please see the following chapters.

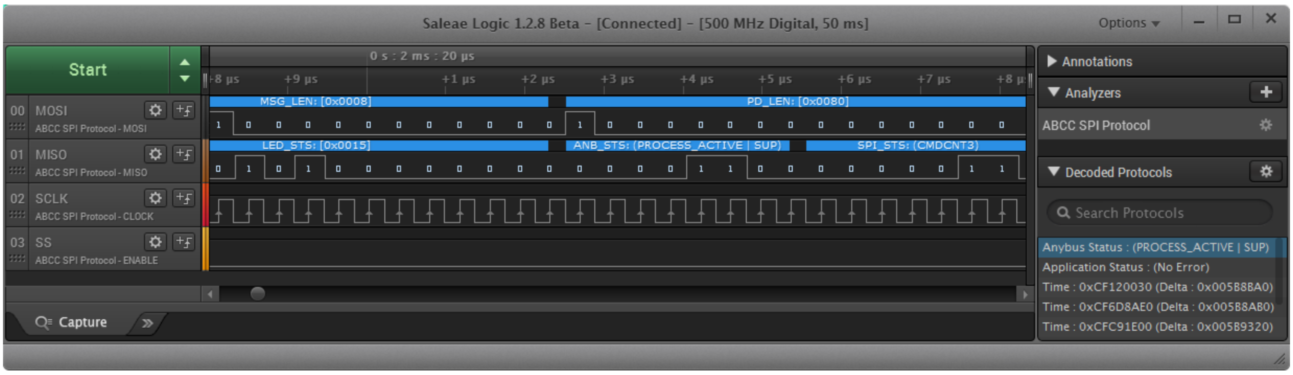
3 Feature List

The ABCC SPI Protocol Analyzer Plugin consists of the following features:

- Multi-layered bubble-text framing of ABCC SPI telegram fields
    - Each field when appropriate are enumerated into their corresponding text-names. For instance, a value of 0xF5 in the object field in the SPI message will be resolved to “OBJ: (EtherCAT Object)” when sufficiently zoomed in. When zoomed out the field may resolve to “OBJ: 0xF5”, “OBJ”, or even just “O” depending on if the text fits in the bubble-text space.
- 3-wire or 4-wire SPI support
    - NOTE: 4-wire is strongly recommended for the current version of this plugin. A future release will improve on the reliability of 3-wire mode, by specifying a minimum inter-packet gap for detecting idle periods between SPI transactions, this will help with state machine alignment when the beginning of a capture starts in the middle of a transaction.
- Indexed and searchable results (with analyzer filter options) for:
    - Network timestamp
    - Anybus Status change events
    - Application Status change events
    - Error events (logical or protocol-wise)
        - Error response messages
        - Application error status events
        - Anybus error status events
        - SPI CPOL/CPHA/SS settings mismatch
        - CRC32 error events
        - Retransmissions (partial support)
        - Buffer full (warning event)
    - New messages (response, command, error response) with support for indication of SPI fragmentation protocol.

4 ABCC SPI Protocol Plugin Overview

Each bubble-text frame (the blue bars seen in the software) correspond to fields within an ABCC SPI frame. When applicable, these fields are processed and converted into human readable enumerations of the flags and values that make up the underlying ABCC communication. For instance, the bubble text frame shown in the figure below shows “ANB_STS: (PROCESS_ACTIVE | SUP)”. This indicates that the ABCC is in the process active state and is being supervised.



Saleae Logic 1.2.8 Beta – [Connected] – [500 MHz Digital, 50 ms]

MSG_LEN: [0x0008]
PD_LEN: [0x0080]
LED_STS: [0x0015]
ANB_STS: (PROCESS_ACTIVE | SUP)
SPI_STS: (CMDCNT3)

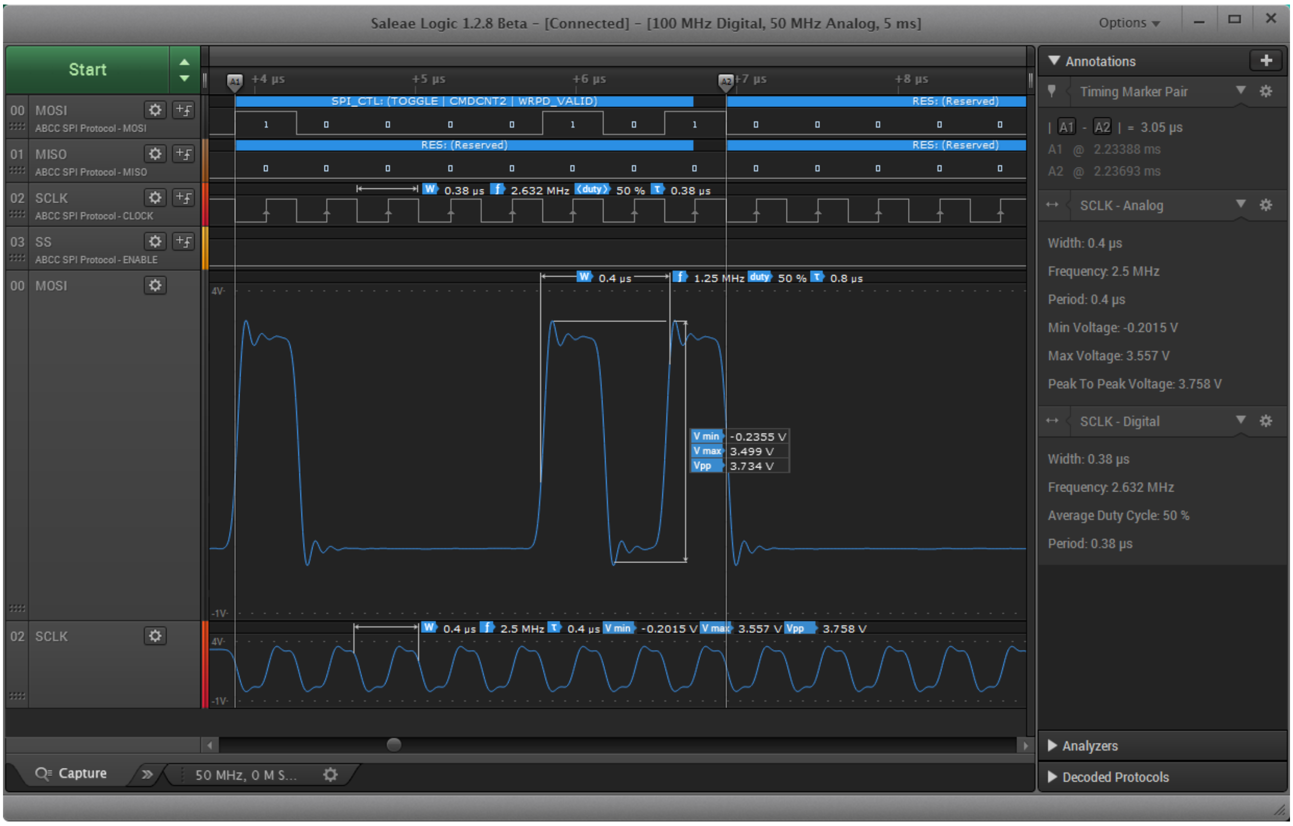
Analyzers: ABCC SPI Protocol

Decoded Protocols:

```
Anybus Status : (PROCESS_ACTIVE | SUP)
Application Status : (No Error)
Time : 0xCF120030 (Delta : 0x005B8BA0)
Time : 0xCF6D8AE0 (Delta : 0x005B8AB0)
Time : 0xCFC91E00 (Delta : 0x005B9320)
```

Figure 3: Example capture using the plugin

With the use of the current generation of Saleae Logic hardware, a user performs basic analog and digital measurements to be reassured that the SPI Master’s and the ABCC’s (SPI Slave) signal quality are within specifications.



Saleae Logic 1.2.8 Beta – [Connected] – [100 MHz Digital, 50 MHz Analog, 5 ms]

SPI_CTL: (TOGGLE | CMDCNT2 | WRPD_VALID)
RES: (Reserved)

Timing Marker Pair
| A1 - A2 | = 3.05 µs
A1 @ 2.23388 ms
A2 @ 2.23693 ms

SCLK - Analog
Width: 0.4 µs
Frequency: 2.5 MHz
Period: 0.4 µs
Min Voltage: -0.2015 V
Max Voltage: 3.557 V
Peak To Peak Voltage: 3.758 V

SCLK - Digital
Width: 0.38 µs
Frequency: 2.632 MHz
Average Duty Cycle: 50 %
Period: 0.38 µs

Figure 4: Example taking physical measurements

The figure below illustrates the multi-layered bubble text that the plugin supports. This example shows how the SPI Control field looks when zoomed out, and how it may look when sufficiently zoomed-in, revealing more verbose information about the state of the field. Additionally, when sufficiently zoomed-in, the logic signal will be drawn with ‘0’ and ‘1’ indicators over the exact sample point in the corresponding channel which reflect the state of the signal during that sample period.

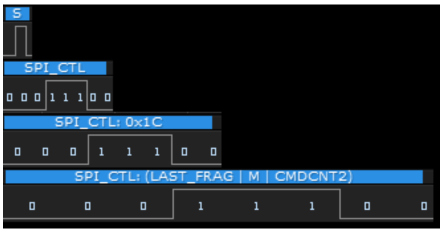
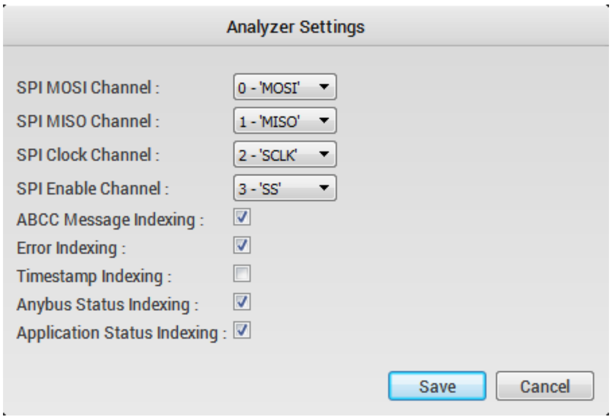


Figure 5: Example showing multi-layered bubble text for the SPI Control Field

To determine if the last, most detailed, bubble-text is currently being displayed the user can simply make the following observations:

- All frames will contain a single-character bubble-text entry
    - This character will either be the first character of the “tag” or an “alert” represented by ‘!’.
- All frames will contain a “tag” bubble-text entry
    - This entry is meant to be short but readable enough for the user to determine its meaning when sufficiently familiar with the ABCC SPI telegram fields.
- All frames will contain a “value”
    - This value will obey the “display radix” options that are part of the Logic software.
- Optionally, frames may contain a descriptive/enumerated bubble-text entry
    - When having to zoom-in and out, a user can reasonably determine if there are deeper levels of bubble-text available by the present of parenthesis or square brackets encapsulating the information following the “tag”. A “tag” and the information that follows is always separated by a colon ‘:’.

4.1 Analyzer Settings Window


Analyzer Settings

SPI MOSI Channel : 0 - 'MOSI'
SPI MISO Channel : 1 - 'MISO'
SPI Clock Channel : 2 - 'SCLK'
SPI Enable Channel : 3 - 'SS'
ABCC Message Indexing : ☑
Error Indexing : ☑
Timestamp Indexing : ☐
Anybus Status Indexing : ☑
Application Status Indexing : ☑

Figure 6: Available options for analyzer

4.1.1 ABCC Message Indexing

When enabled, this option will index each “New message” in the “Decoded Protocols” sub-window. Retransmissions, commands, and (error) responses all fall into this category. The default state of this option is “enabled”.

<u>Example Decodes:</u>

Obj {03:0001h}, Cmd {41:0003h} MOSI

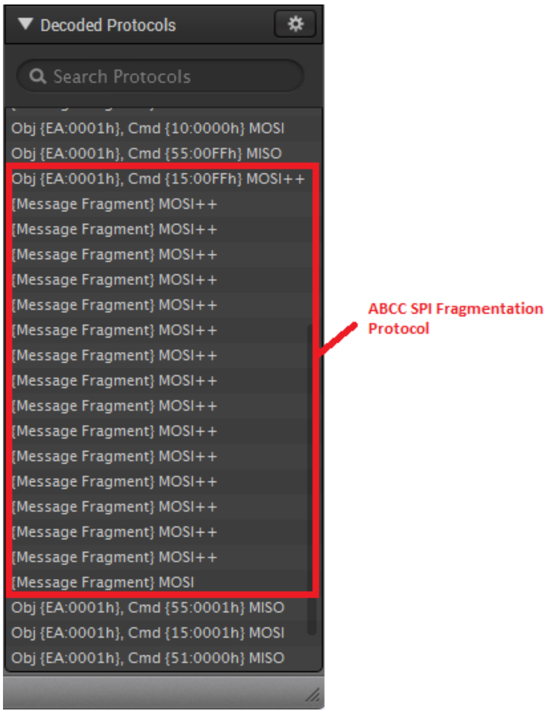
A new message for object 0x03 (Network object) instance 0x0001 with command byte 0x41 (Get_Attribute) for attribute 0x03 (Data format). To keep things compact (since the Saleae Logic software does not have a resizable decodes window) the message fields are represented in the decoded section as raw hexadecimal format. This may change in a future release.

Obj {F8:0001h}, Cmd {01:0006h} MOSI++

Like the previous example this indicates a new message. The presence of "++" at the end of the entry indicates that this is message is using the ABCC's SPI fragmentation protocol.

{Message Fragment} MOSI++

Indicates the corresponding message data is a continuation of the ABCC's SPI fragmentation protocol. The presence of "++" at the end of the entry indicates that this is message is not the "Last Fragment".



```
Obj {EA:0001h}, Cmd {10:0000h} MOSI
Obj {EA:0001h}, Cmd {55:00FFh} MISO
Obj {EA:0001h}, Cmd {15:00FFh} MOSI++
{Message Fragment} MOSI++
{Message Fragment} MOSI++
{Message Fragment} MOSI++
{Message Fragment} MOSI++
{Message Fragment} MOSI++
{Message Fragment} MOSI++
{Message Fragment} MOSI++
{Message Fragment} MOSI++
{Message Fragment} MOSI++
{Message Fragment} MOSI++
{Message Fragment} MOSI++
{Message Fragment} MOSI++
{Message Fragment} MOSI++
{Message Fragment} MOSI++
{Message Fragment} MOSI++
{Message Fragment} MOSI
Obj {EA:0001h}, Cmd {55:0001h} MISO
Obj {EA:0001h}, Cmd {15:0001h} MOSI
Obj {EA:0001h}, Cmd {51:0000h} MISO
```

ABCC SPI Fragmentation Protocol

Figure 7: Example showing how the message fragmentation protocol is displayed in decoded protocols subwindow

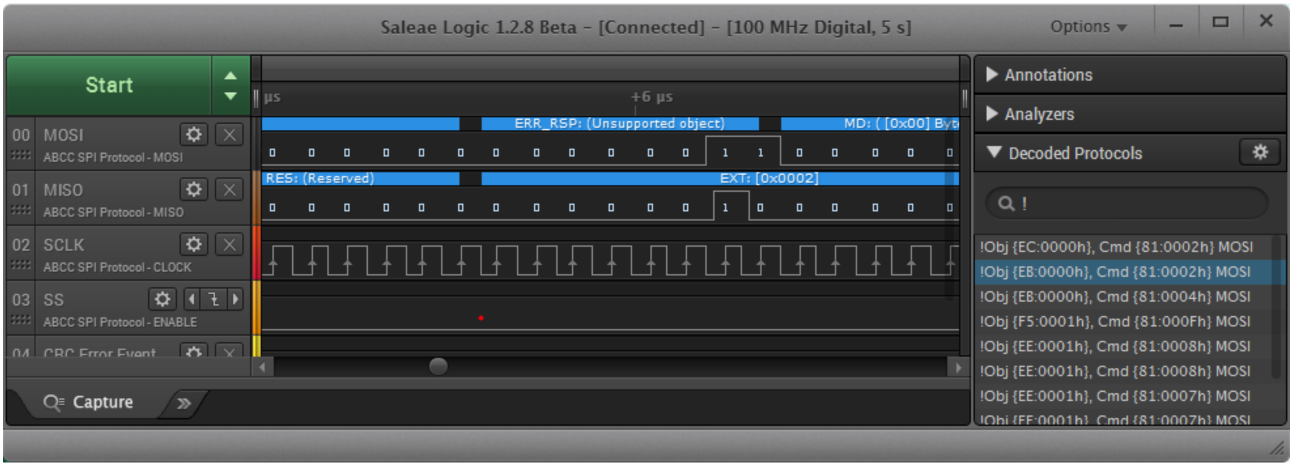
4.1.2 Error Indexing

When enabled, this option will index each "error" event. The default state of this option is "enabled". To make searching for issues as easy as possible, all error events are prefixed with a "!". An error is classified as any of the following:

- SPI settings mismatch (CPOL, CPHA, Slave-Select Active Level, etc.)
- Protocol Fragmentation

  - ○ Not to be confused with ABCC SPI fragmentation protocol. This error is indicated when Slave-Select de-asserts before a complete ABCC transaction completes.
- Error Response Messages
- Application Status is in an “error state”
- Anybus Status is in an “error state”

Note: To help with locating the actual error event, “error markers” (dots, squares, or x’s are placed on the Slave Select signal where the beginning of the bubble-text frame associated with the error event occurred. The current version of Saleae Logic software has some issues with resolving markers that are drawn in close proximity to each other so such markers are relocated to the slave select line in order to help maximize the likelihood that the marker will be drawn on the screen. Ideally, a future update of the Logic software will remove this limitation, and these markers will be relocated to reside on the physical channel where the error event was detected.



!Obj {EC:0000h}, Cmd {81:0002h} MOSI
!Obj {EB:0000h}, Cmd {81:0002h} MOSI
!Obj {EB:0000h}, Cmd {81:0004h} MOSI
!Obj {F5:0001h}, Cmd {81:000Fh} MOSI
!Obj {EE:0001h}, Cmd {81:0008h} MOSI
!Obj {EE:0001h}, Cmd {81:0008h} MOSI
!Obj {EE:0001h}, Cmd {81:0007h} MOSI
!Obj {EE:0001h}, Cmd {81:0007h} MOSI

4.1.3 Timestamp Indexing

When enabled, this option will index each “network timestamp” received from the MISO ABCC message. The default state of this option is “disabled”.

Note: The Saleae Logic software currently does not support adding multiple indexed results to the “Decoded Protocols” sub-window for separate bubble-text entries that exist in the same time-space (even though they reside in separate channels). This means when enabling both “Timestamp Indexing” and “Application Status Indexing” only one entry in the decoded protocols can exist. Currently, priority is given to the “Application Status” events. A future update to the plugin may resolve this limitation.

4.1.4 Anybus Status Indexing

When enabled, this option will index each “Anybus status event” received from the MISO ABCC message. The default state of this option is “disabled”. Even when disabled, if error indexing is enabled, any Anybus state related to an error/exception will be indexed.

4.1.5 Application Status Indexing

When enabled, this option will index each “Application status event” received from the MISO ABCC message. The default state of this option is “disabled”. Even when disabled, if error indexing is enabled, any application state related to an error/exception will be indexed.

Note: The Saleae Logic software currently does not support adding multiple indexed results to the “Decoded Protocols” sub-window for separate bubble-text entries that exist in the same time-space (even though they reside in separate channels). This means when enabling both “Timestamp Indexing” and “Application Status Indexing” only one entry in the decoded protocols can exist. Currently, priority is given to the “Application Status” events. A future update to the plugin may resolve this limitation.