

Kurzdokumentation zum USB-LOGI-500 Logikanalysator-Programm

January 10, 2011

Der USB-LOGI-500 dient zum Aufzeichnen digitaler Signale in einer Schaltung. Dazu müssen die aufzuzeichnenden Signale auf externe Pins auf der Baugruppe hinausgeführt werden, an die das Messkabel vom USB-LOGI angeschlossen werden kann. Der USB-LOGI muss vor der Aufzeichnung über eine Konfigurationsdatei konfiguriert werden. Nach Abschluss der Messung werden die Daten in GTKWave angezeigt.

1 Der USB-LOGI-500

Der USB-LOGI-500 tastet die digitalen Signale die aufgezeichnet werden sollen mit einer einstellbaren Abtastrate ab. Sie kann beliebig zwischen 1 Hz und 150 MHz eingestellt werden. Weiterhin wird eine Abtastrate von 600 MHz unterstützt. Bei jeder Messung werden 4096 Abtastpunkte in einem internen RAM gespeichert. Es können maximal 36 Kanäle (Channels) aufgezeichnet werden.

2 Begriffe rund um den USB-LOGI-500

- Kanal (Channel): Messeingang des USB-LOGI
- Signal: Zeile in GTKWave. Signale werden aus einem oder mehreren Kanälen gebildet.
- Abtastrate (Sample Rate): Zeit die zwischen zwei Abtastpunkten liegt, gemessen in Hertz ($1 \text{ Hz} = \frac{1}{s}$)
- Trigger: Bedingungen an bestimmte Signale, bevor die eigentliche Messung startet
- Pre-Trigger: Zeitabschnitt bevor die eigentliche Trigger-Bedingung erfüllt ist

3 Konfigurationsdatei

Die Konfigurationsdatei ist eine XML-Datei, dessen Format weitgehend intuitiv und selbsterklärend ist, wobei auf Groß- und Kleinschreibung geachtet werden muss. Sie beginnt mit den Zeilen

```
<?xml version="1.0"?>
<la>
```

und endet mit der Zeile

```
</la>
```

Dazwischen werden Abtastrate, Pre-Trigger, Signale und Trigger-Bedingungen konfiguriert. Kommentare werden wie folgt geschrieben:

```
<!-- Kommentar (auch über
      mehrere Zeilen möglich) -->
```

3.1 Abtastrate (Sample Rate)

Die Abtastrate wird mit folgender Zeile konfiguriert:

```
<samplerate>...</samplerate>
```

Statt “...” muss ein Wert für die Abtastrate in Hertz (Hz) eingesetzt werden. Gültige Werte sind Zahlen zwischen 1 und 150000000 sowie 600000000.

3.2 Pre-Trigger

```
<pretrigger>...</pretrigger>
```

Gültige Werte sind Zahlen zwischen 1 und 7, was jeweils dem entsprechenden Vielfachen eines Achtel von 4096 Abtastpunkten entspricht. Wird beispielsweise der Wert 4 eingetragen entspricht dies $\frac{4}{8} \cdot 4096 = 2048$ Abtastpunkten.

3.3 Signale

Signale werden in einem Unterabschnitt

```
<signals>
  ...
</signals>
```

definiert. Die Definitionen von Signalen dient der übersichtlicheren Darstellung der Messergebnisse in GTKWave. Einem Signal müssen ein oder mehr Kanäle zugeordnet werden. Jedes Signal braucht einen Namen.

Beispiele für ein 1-Bit Signale:

```

<!-- Dem Signal "Takt" wird Kanal 7 des USB-LOGI zugeordnet. -->
<signal name="Takt">
  <ch>7</ch>
</signal>
<!-- Dem Signal "WE" wird Kanal 0 des USB-LOGI zugeordnet. -->
<signal name="WE">
  <ch>0</ch>
</signal>

```

Beispiel für einen 4-Bit Signal-Bus:

```

<signal name="led">
  <ch>4</ch>
  <ch>3</ch>
  <ch>2</ch>
  <ch>1</ch>
</signal>

```

3.4 Trigger-Bedingungen

Die Trigger-Bedingungen sind optional. Fehlen sie in der Konfigurationsdatei, beginnt die Messung sofort.

Eine Trigger-Bedingung ist entweder wahr oder falsch. Ist die Trigger Bedingung erfüllt (also wahr), startet die eigentliche Messung. Für mehr Flexibilität können zwei verschiedene Trigger A und B definiert werden, die bei Bedarf zu einer gemeinsamen Trigger-Bedingung logisch UND bzw. ODER verknüpft werden können.

```

<trigger when="...">
  <A>
    ...
  </A>
  <B>
    ...
  </B>
</trigger>

```

Gültige Werte für "when" sind:

- A
- B
- A AND B
- A OR B

Den Triggern A bzw. B müssen Kanäle zugeordnet werden. Ein Trigger ist dann erfüllt, wenn alle seine Kanäle bestimmte Werte bzw. Wertänderungen aufzeigen.

```
<A>
  <ch when='...'>7</ch>
</A>
```

Gültige Werte für “when” sind:

- high
- low
- high low
- rising_edge
- falling_edge
- rising_edge falling_edge

“high” bezeichnet dabei die logische 1, “low” die logische “0”. Beispiel für Trigger-Bedingungen:

```
<!-- Trigger A ist erfüllt, wenn auf Kanal 7 eine steigende Flanke erkannt wird -->
<A>
  <ch when='rising_edge'>7</ch>
</A>
<!-- Trigger B ist erfüllt wenn auf dem Signal-Bus 4 downto 1 der Wert “1100” oder ‘
<B>
  <ch when='high'>4</ch>
  <ch when='high'>3</ch>
  <ch when='low'>2</ch>
  <ch when='high low' >1</ch>
</B>
```

Weiterhin kann gefordert werden, dass die Gesamt-Triggerbedingung erst n-mal erfüllt sein muss, damit die Messung startet.

```
<!-- Trigger-Bedingung muss 10 mal erfüllt werden -->
<trigger when='...' counter='10'>
  ...
</trigger>
```

In folgendem Beispiel muss auf dem Signal-Bus 4 downto 1 zehn mal der Wert “1100” oder “1101” bei einer steigenden Flanke auf Kanal 7 anliegen, bevor die Messung startet.

```

<trigger when='A AND B' counter='10'>
  <A>
    <ch when='rising_edge'>7</ch>
  </A>
  <B>
    <ch when='high'>4</ch>
    <ch when='high'>3</ch>
    <ch when='low'>2</ch>
    <ch when='high low' >1</ch>
  </B>
</trigger>

```

4 Bedienung des Programms

Zunächst eine Konfigurationsdatei wie in Abschnitt 3 beschrieben mit einem Texteditor erstellen und mit der Endung .xml abspeichern. Konsole öffnen und mittels dem Kommando cd in das Verzeichnis wechseln, in dem die Konfigurationsdatei abgespeichert wurde. Vergewissern das der USB-LOGI und die Messkabel angeschlossen sind. Start der Messung mit dem Befehl:

```
usb-logi <Dateiname ohne .xml>
```

Es wird zunächst die Konfigurationsdatei eingelesen und auf Fehler überprüft. Zur Kontrolle werden die eingelesenen Daten auf der Konsole ausgegeben. Nun wird der USB-LOGI konfiguriert. Nach füllen des Pre-Triggers wird auf die Trigger-Bedingung gewartet (sofern vorhanden) und die Messung gestartet. Nach Abschluss werden die Messdaten für GTKWave aufbereitet und in GTKWave angezeigt.

Der Zeitpunkt bei dem die Trigger-Bedingung erfüllt wurde, wird mit einem gelben, vertikalen Strich markiert.

5 Komplettbeispiel

```

<?xml version="1.0"?>
<la>
  <samplerate>15000000</samplerate>
  <pretrigger>2</pretrigger>
  <signals>
    <signal name='Takt'>
      <ch>7</ch>
    </signal>
    <signal name='WE'>
      <ch>0</ch>
    </signal>
    <signal name='led'>

```

```
        <ch>4</ch>
        <ch>3</ch>
        <ch>2</ch>
        <ch>1</ch>
    </signal>
</signals>
<trigger when='A AND B' counter='10'>
  <A>
    <ch when='rising_edge'>7</ch>
  </A>
  <B>
    <ch when='high'>4</ch>
    <ch when='high'>3</ch>
    <ch when='low'>2</ch>
    <ch when='high low' >1</ch>
  </B>
</trigger>
</la>
```