聆思科技
LISTENAI

LS2100 BLE SoC Datasheet

CSDS-23001-031_V0.1 February 13, 2023

LISTENAI Proprietary and Confidential

Prepared and Provided Under NDA

DISCLAIMER

The information presented in this document belongs to Anhui LISTENAI Co., Ltd. and/or its affiliates (LISTENAI). Without prior written permission of LISTENAI, no entity or individual shall copy, modify, or distribute part or all of the document in any way.

The information provided in this document is for reference only. LISTENAI does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information. LISTENAI shall have no liability for any errors contained in this document.

LISTENAI reserves the right to make corrections, modifications, enhancements, improvements, and other changes to the content at any time without notice. LISTENAI reserves all the right for the final explanation.

LISTENAI, 聆思科技, and other LISTENAI icons are trademarks of Anhui LISTENAI Co., Ltd. All other trademarks and trade names mentioned in this document are the property of their respective holders.

Update History

Version	Date	Update Descripti
V1.0	November 22, 2022	Initial release.

Table of Contents

No matter where you go, there you are.

—Banzai

1 Footnotes

Autonumbered footnotes are possible, like using[1] and[2].

They may be assigned 'autonumber labels' - for instance,[4] and[3].

Footnote references, like[5]. Note that footnotes may get rearranged, e.g., to the bottom of the "page".

Auto-symbol footnotes are also possible, like this:*[0] and†[0].

[1] This is the first one.
[2] This is the second one.
[4] a.k.a. *fourth*
[3] a.k.a. *third*
[5] A numerical footnote. Note there's no colon after the `]`.
[0] This is the first one.
[0] This is the second one.

2 超链接

2.1 替换引用 (Substitution Reference)

替换引用就是用定义的指令替换对应的文字或图片，和内置指令 (inline directives) 类似。

这是 github 的 Logo，我的 github 用户名是:Who。

2.2 隐式超链接 (Implicit Hyperlink)

小节标题、脚注和引用参考会自动生成超链接地址，使用小节标题、脚注或引用参考名称作为超链接名称就可以生成隐式链接。

隐式超链接详细内容，参见隐式超链接 *(Implicit Hyperlink)* 。

If CLOCK_MODE equals 1, there is no idle time between back-to-back characters if data is ready in the transmit FIFO. In this case, because *sync_delay* equals one pclk as described in Section **??** *Equation Example*, the requirement to avoid idle time between consecutive characters is met for all {DLH,DLL} values.

If CLOCK_MODE equals 1, there is no idle time between back-to-back characters if data is ready in the transmit FIFO. In this case, because *sync_delay* equals one pclk as described in Section **??** *Error*, the requirement to avoid idle time between consecutive characters is met for all {DLH,DLL} values.

3 简单表 (Simple Tables)

标题可合并单元格

Inputs		Output
A	B	A or B
False	False	False
True	False	True
False	True	True
True	True	True

行也可以合并单元格

Inputs		Output
A	B	A or B
False	False	False
False		True
False	True	True
True	True	True

合并方式?

A	B	A or B
False	False	False
False	True	True
False	True	True
	True	True

4 X

乘号 ×

4.1 分隔符

分隔符就是一条水平的横线，是由 4 个 - 或者更多组成，需要添加换行。

上面部分

下面部分

4.2 定义列表 (Definition Lists)

定义列表可以理解为解释列表，即名词解释。

条目占一行，解释文本要有缩进；多层可根据缩进实现。

定义 1 这是定义 1 的内容

定义定义 2 这是定义 2 的内容

4.3 文档测试块 (Doctest Blocks)

文档测试块是交互式的 Python 会话，以 >>> 开始，一个空行结束。

```
>>> print "This is a doctest block."
This is a doctest block.
```

新的一行。

4.4 文字块 (Literal Blocks)

下面是文字块内容：

```
这是一段文字块
同样也是文字块
还是文字块
```

这是新的一段。

A paragraph containing only two colons indicates that the following indented or quoted text is a literal block.

```
Whitespace, newlines, blank lines, and
all kinds of markup (like *this* or
\this) is preserved by literal blocks.

The paragraph containing only '::'
will be omitted from the result.
```

The :: may be tacked onto the very end of any paragraph. The :: will be omitted if it is preceded by whitespace. The :: will be converted to a single colon if preceded by text, like this:

```
It's very convenient to use this form.
```

Literal blocks end when text returns to the preceding paragraph's indentation. This means that something like this is possible:

```
    We start here
  and continue here
and end here.
```

Per-line quoting can also be used on unindented literal blocks:

```
> Useful for quotes from email and
> for Haskell literate programming.
```

4.5 行块 (Line Blocks)

行块对于地址、诗句以及无装饰列表是非常有用的。

行块是以 | 开头，每一个行块可以是多段文本。

前后各有一个空格。

下面是行块内容：

这是一段行块内容
这同样也是行块内容还是行块内容

这是新的一段。

从此鲜花赠自己，纵马踏花向自由。
我与旧事归于尽，来年依旧迎花开。

Line blocks are useful for addresses,
verse, and adornment-free lists. （显示为两行）

Each new line begins with a
vertical bar (“|”).
Line breaks and initial indents
are preserved.
Continuation lines are wrapped portions of long lines; they begin with spaces in place of vertical bars.
（显示为一行）

4.6 块引用 (Block Quotes)

块引用是通过缩进来实现的，引用块要在前面的段落基础上缩进。

通常引用结尾会加上出处 (attribution)，出处的文字块开头是 –、—、—，后面加上出处信息。

块引用可以使用空的注释.. 分隔上下的块引用。

注意在新的块和出处都要添加一个空行。

下面是引用的内容：

> “真的猛士，敢于直面惨淡的人生，敢于正视淋漓的鲜血。”
>
> —鲁迅
>
> “人生的意志和劳动将创造奇迹般的奇迹。”
>
> —涅克拉索

Block quotes are just:

> Indented paragraphs,
>
> > and they may nest.

4.7 选项列表 (Option Lists)

选项列表是一个类似两列的表格，左边是参数，右边是描述信息。当参数选项过长时，参数选项和描述信息各占一行。

选项与参数之间有一个空格，参数选项与描述信息之间至少有两个空格。

-a	command-line option “a”
-b file	options can have arguments and long descriptions
--long	options can be long also
--input=file	long options can also have arguments
/V	DOS/VMS-style options too

4.8 字段列表 (Field Lists)

标题 reStructuredText 语法说明

作者

- Alice
- Hank
- Wendy

时间 2016 年 06 月 21 日

概述 这是一篇关于 reStructuredText 的语法说明。

Authors Tony J. (Tibs) Ibbs, David Goodger

(and sundry other good-natured folks) （上面空一行，此处另起一行）

Version 1.0 of 2001/08/08

Dedication To my father.

4.9 符号列表 (Bullet Lists)

符号列表可以使用 -、*、+ 来表示。

不同的符号结尾需要加上空行，下级列表需要有空格缩进。

- 符号列表 1
- 符号列表 2
  - 二级符号列表 1
  - 二级符号列表 2
  - 二级符号列表 3
- 符号列表 3
- 符号列表 4

5 Glossary

UART Universal Receiver Transmitter

USB Universal Bus

6 加粗居中 Right-Aligned

6.1 居中

Confidential

6.2 居中加粗

Confidential

6.3 空格

space \

“ confidential “

empty math format Confidential

space

35656 space before

行内代码：“ 行内文本 (inline literal) 通常显示为等宽文本，空格可以保留，但是换行不可以。“

行内代码：`vip_create_buffer()`

行内代码：`space before`

行内代码：行内文本 (`inline literal`) 通常显示为等宽文本，空格可以保留，但是换行不可以。

7 版权所有

```
Copyright |copy| 2023, |MACAMACA (TM)| |---| all rights reserved.

.. |copy| unicode:: 0xA9 .. copyright sign

.. |MACAMACA (TM)| unicode:: MACAMACA U+2122 .. with trademark sign

.. |---|  unicode:: U+02014 .. em dash
 :trim:
```

8 章节自动编号

```
.. numbered::
.. sectnum::
```

9 Topic

标题 reStructuredText 语法说明

作者

- Alice
- Hank
- Wendy

时间 2016 年 06 月 21 日

概述 这是一篇关于 reStructuredText 的语法说明。

If CLOCK_MODE equals 1, there is no idle time between back-to-back characters if data is ready in the transmit FIFO. In this case, because *sync_delay* equals one pclk as described in Section **??** *Equation Example*, the requirement to avoid idle time between consecutive characters is met for all {DLH,DLL} values.

9.1 directive rubric

paragraph heading

9.2 directive container

This paragraph might be rendered in a custom way.

9.3 directive header

If CLOCK_MODE equals 1, there is no idle time between back-to-back characters if data is ready in the transmit FIFO. In this case, because *sync_delay* equals one pclk as described in Section **??** *Equation Example*, the requirement to avoid idle time between consecutive characters is met for all {DLH,DLL} values.

9.4 directive topic

試試行不行

If CLOCK_MODE equals 1, there is no idle time between back-to-back characters if data is ready in the transmit FIFO. In this case, because *sync_delay* equals one pclk as described in Section **??**

Equation Example, the requirement to avoid idle time between consecutive characters is met for all {DLH,DLL} values.

9.5 directive sidebar

出现的位置

出现在哪里呢这个 sidebar

10 分栏（仅对 HTML 有效）

- good
- bad
- excellent
- normal
- qualified
- good
- bad
- excellent
- normal
- qualified
- good
- bad
- excellent
- normal
- qualified
- good
- bad
- excellent
- normal
- qualified
- good
- bad
- excellent
- normal
- qualified

11 Equation Example

11.1 公式中空格的实现方式

```
14\ (roundoff\ value)
\text{14 (roundoff value)}
```

$$14\ (roundoff\ value)$$

$$14 \text{ (roundoff value)}$$

11.2 Error

注意：如果 text{} 内有 “_“，这个命令的使用会导致报错。在 “_“前面加上反斜线就 okay。

```
\frac{\text{Rate of SSI data transmission}}{Rate of DW\_ahb\_dmac response to
destination burst requests}
```

$$\frac{\text{Rate of SSI data transmission}}{Rate of DW\_ahb\_dmac response to destination burst requests}$$

$$\frac{\text{Rate of SSI data transmission}}{\text{Rate of DW\_ahb\_dmac response to destination burst requests}}$$

11.3 公式中短线的实现方式

前面加 slash“““

$$\text{DLF} = BRD_F * 2^{DLF\_SIZE}$$

$$BRD_F * 2^{DLF\_SIZE}$$

$$\text{DLF} = BRD_F * 2^{DLF\_SIZE}$$

This equation ref{equ1} is not okay?

$$\text{DLF} = BRD_F * 2^{DLF\_SIZE} = 0.866132364 * 16 = 13.858117824 = 14 \text{ (roundoff value)}$$

Therefore, the Generated Baud Rate (GBR) is as follows:

$$GBR = \frac{\text{Serial Clock}}{(16 \times GD)} = \frac{133}{16 \times 1.875} = 4433333.333$$

$$Error = \frac{\text{GBR - RBR}}{RBR} = 0.004729$$

11.4 公式中% 前面要加 slash

$$Error\% = 0.004729 \times 100 = 0.473$$