LD-LRS3100/LD-OEM 指令手册

编写人 ：李木

日 期 ：20010-02-26

版 本 ：V1.0

本文档详细介绍 LD-0EM/LD-LRS3100 与 PC 间的串口/以太网通讯的接线方法和常用的通讯指令。

一、 LD-OEM/LD-LRS-3100 硬件接线

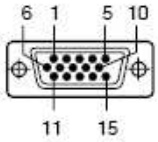
1、 LD-OEM 硬件接线

接线端子以及通讯接口位于 LD-OEM 的底部。

电源以及 IO 端子定义：

Signal	Name on the board	Function
+24 V DC	V-EXT	Operating voltage
GND	GND-EXT	Operating voltage ground
OUT 1	OUT 1	Switching output 1, function depends on application
OUT 2	OUT 2	Switching output 2, function depends on application
OUT 3	OUT 3	Switching output 3, function depends on application
OUT 4	OUT 4	Switching output 4, function depends on application

串口、以太网端子定义：



Pin	Signal	Interface	Function
1	24 V DC		Power supply
2	CAN L	CAN bus (IN/OUT)	CAN bus Low
3	CAN H	CAN bus (IN/OUT)	CAN bus High
4	GND_Data		Data interface ground
5	GND		Power supply ground
6	RD+	RS 422	Receiver+
7	RD-/RxD	RS 422/RS 232	Receiver-
8	TD+	RS 422	Transmitter+
9	TD-/TxD	RS 422/RS 232	Transmitter-
10	OUT 1		Switching output 1, function depends on application
11	TPIP	Ethernet IN	Receiver+
12	TPIN	Ethernet IN	Receiver-
13	TPOP	Ethernet OUT	Transmitter+
14	TPON	Ethernet OUT	Transmitter-
15	OUT 2		Switching output 2, function depends on application
Housing	-	-	Shield

Table 1-6: LD-LRS1000 interface adapter: Pin assignment of 15-pin D Sub HD plug

LD-OEM 以太网接线方法：

Signal LD-LRS1000	Pin (15-pin D Sub HD plug)	Pin (8-pin RJ-45 socket)
TPIP	11	3
TPIN	12	6
TPOP	13	1
TPON	14	2

LD-OEM RS422 接线方法：

LD OEM 上位机

6 (RD+)----------TD+

7 (RD-)----------TD-

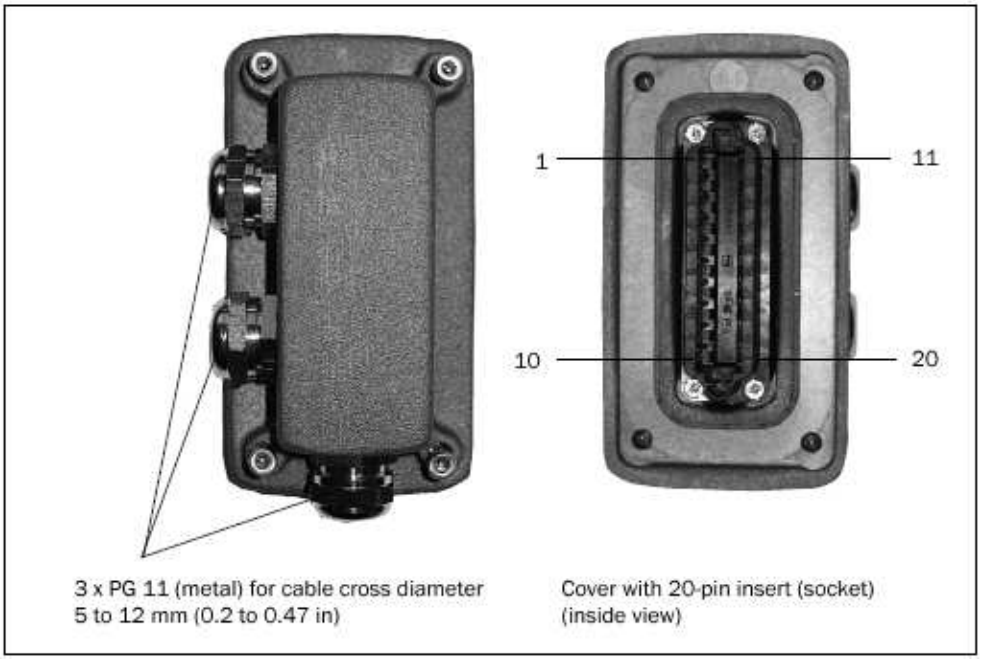
8 (TD+)----------RD+

9 (TD-)----------RD-

Unit 1, 24/F, Talent Center, No.45 Tian He Road, Guangzhou, 510075, China
Tel : 86-20-3830 3155 Fax : 86-20-3830 3248

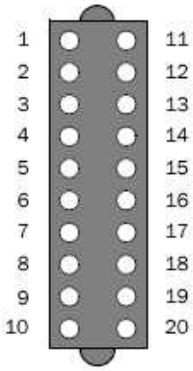
2、 LD-LRS3100 硬件接线

接线端子以及通讯接口位于 LD-LRS3100 的背部。



LD-LRS3100 接线端子定义：

LD-LRS3100: 20-pin Harting socket



Pin	Signal	Interface	Function
1	DC 24 V_HZG		Heater power supply
2	DC 24 V		Electronics power supply
3	OUT 1	Switching output 1	Function depends on application
4	OUT 3	Switching output 3	Function depends on application
5	–	n. c.	–
6	TD–	RS 422	Transmitter–
7	TD+	RS 422	Transmitter+
8	CAN H	CAN bus (IN/OUT)	CAN bus High
9	TPOP	Ethernet OUT	Transmitter+
10	TPON	Ethernet OUT	Transmitter–
11	GND_HZG		Heater ground
12	GND		Electronics ground
13	OUT 2	Switching output 2	Function depends on application
14	OUT 4	Switching output 4	Function depends on application
15	GND_Data		Data interfaces ground
16	RD–	RS 422	Receiver–
17	RD+	RS 422	Receiver+
18	CAN L	CAN bus (IN/OUT)	CAN bus Low
19	TPIP	Ethernet IN	Receiver+
20	TPIN	Ethernet IN	Receiver–
Housing	–	–	Shield

LD-LRS3100 以太网接线方法：

Signal LD-LRS2100/3100	Pin (20-pin Harting socket)	Pin (8-pin RJ-45 socket)
TPIP	19	3
TPIN	20	6
TPOP	9	1
TPON	10	2

LD-LRS3100 RS422 接线方法：

LD-LRS3100　　上位机

6 (TD-)----------RD-

7 (TD+)----------RD+

16 (RD-)----------TD-

17 (RD+)----------TD+

Unit 1, 24/F, Talent Center, No.45 Tian He Road, Guangzhou, 510075, China
Tel : 86-20-3830 3155　Fax : 86-20-3830 3248

3、 LD-OEM/LD-LRS 电缆要求

1） 供电电源电缆

如果供电电源距离扫描头较近5m以内，建议采用0.25mm2电缆。

如果供电电源距离扫描头最大20m，建议采用1.0mm2电缆

Wire cross-section	Cable length
0.25 mm^2 (0.01 in^2 approx. 24 AWG)	5 m (16.4 ft)
0.5 mm^2 (0.02 in^2 approx. 22 AWG)	10 m (32.81 ft)
1.0 mm^2 (0.04 in^2 approx. 18 AWG)	20 m (65.62 ft)

2） 开关量输出电缆

开光量输出电缆最小线径.0.25mm2,电缆长度为50m时，需要0.5mm2线径电缆。

3） RS232/422通讯电缆

通讯电缆的最小线径0.25mm2,建议采用屏蔽双绞线，注意通讯电缆不能与动力电缆并行走线。

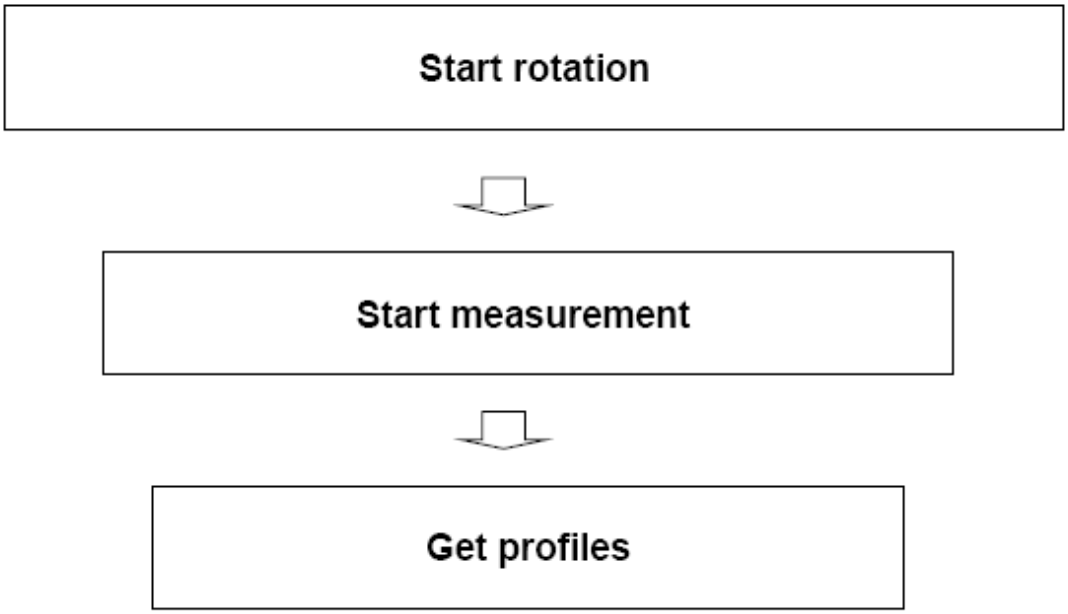
Interface type	Transmission rate	Maximum cable length
RS-232	115200 Bd	10 m (32.81 ft)
RS-422	115200 Bd	100 m (328.08 ft)
CAN bus[1)]	1 MBit/s	30 m (98.43 ft)

Unit 1, 24/F, Talent Center, No.45 Tian He Road, Guangzhou, 510075, China
Tel : 86-20-3830 3155 Fax : 86-20-3830 3248

二、 LD-OEM/LD-LRS 常用指令介绍

1、 串口指令介绍

1) LD-OEM/LD-LRS 获取数据流程



2) 扫描头旋转指令

STX14100004000004030000913A**ETX**

3) 测量模式指令

STX14100003000004047415**ETX**

4) 获取数据指令

STX141000050000030100010 3A2286C**ETX**

获取数据 **0301** 指令介绍：

Profile num:

0001 ——1，表示获取单圈数据，参考以下指令表。

Profile format:

01A2：表示要获取距离值，参考以下指令表。

03A2：表示要获取角度值和距离值，参考以下指令表。

GET_PROFILE

Request command **0301h**:

Description	Requests n profiles of a defined format	
Parameter	*Type*	*Meaning*
PROFILENUM	WORD	Number of profiles, if it is equals 0 the LD-OEM/LD-LRS sends profiles continuously, until the user sends the CANCEL_PROFILE command
PROFILEFORMAT	WORD	16-bit array

Table 5-33: Measurement Services: GET_PROFILE (request command)

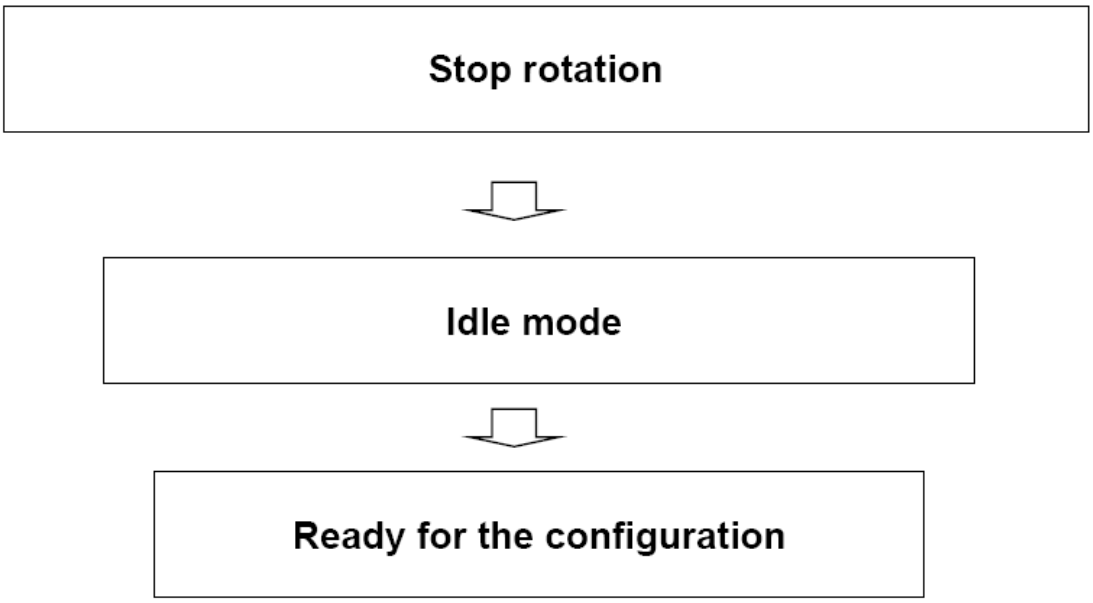
The definition of the 16-bit array PROFILEFORMAT:

Bit	**Meaning**
0	Number of the transmitted profile
1	Profile counter
2	Number of Layer
3	Number of Sector
4	Angle step
5	Number of points of the sector
6	Time stamp when the sector starts
7	Start direction of the sector
8	Measured distances
9	Direction of measured distances
10	Echo amplitudes
11	Time stamp when the sector ends
12	End direction of the sector
13	LD-OEM/LD-LRS mode
14	reserved (always 0)
15	reserved (always 0)

Table 5-34: Definition of the 16-bit array of PROFILEFORMAT

Unit 1, 24/F, Talent Center, No.45 Tian He Road, Guangzhou, 510075, China
Tel : 86-20-3830 3155 Fax : 86-20-3830 3248

5) 扫描头进入停止工作状态步骤



扫描头在测量数据时，如果要进入停止状态步骤如下：

- 停止测量

STX14100004000003027313**ETX**

5.3.3 CANCEL_PROFILE

Request command **0302h**:

Description	Stops the profile output	
Parameter	*Type*	*Meaning*
-	-	-

Table 5-36: Measurement Services: CANCEL_PROFILE (request command)

- 扫描头停止旋转

STX14100004000004027413**ETX**

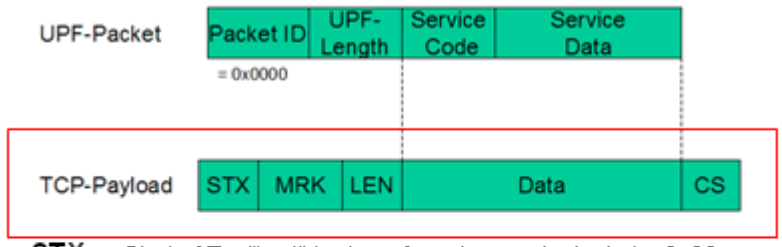
5.4.2 TRANS_IDLE

Request command **0402h:**

Description	Sets the LD-OEM/LD-LRS into the IDLE mode: the motor of the rotating prism stops and the laser is switched off	
Parameter	*Type*	*Meaning*
-	-	-

2. 以太网指令介绍

通过以太网接口获取数据端口号为 Port **49152**，以太网接口数据结构如下：

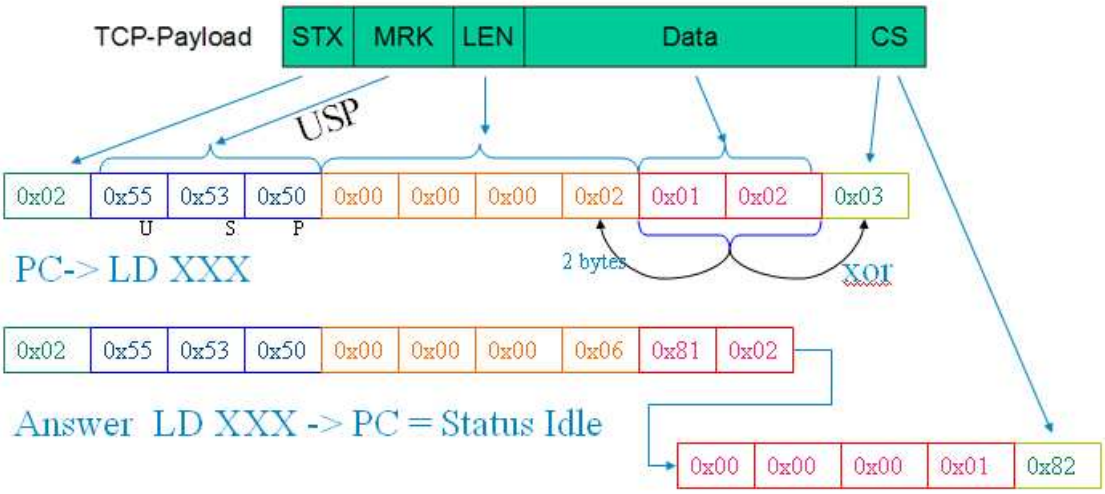


STX „Start of Text", will be transferred as a single byte, 0x02.

MRK Definition of the transmission format „USP"= 0x55, 0x53,0x50 (3 Bytes),

LEN UPF-Length = the number of following bytes in <data>,coded as 32-Bit-Integer (four bytes) without leading sign; the msb (most significant byte) must be transmitted first of all.

CS Checksum, is a single byte, calculated as exclusive-or-relation of all bytes contained in „Data".



通过以太网获取数据流程：

1、扫描头转动

02 55 53 50 00 00 00 04 04 03 00 00 07

2、进入测量模式

02 55 53 50 00 00 00 02 04 04 00

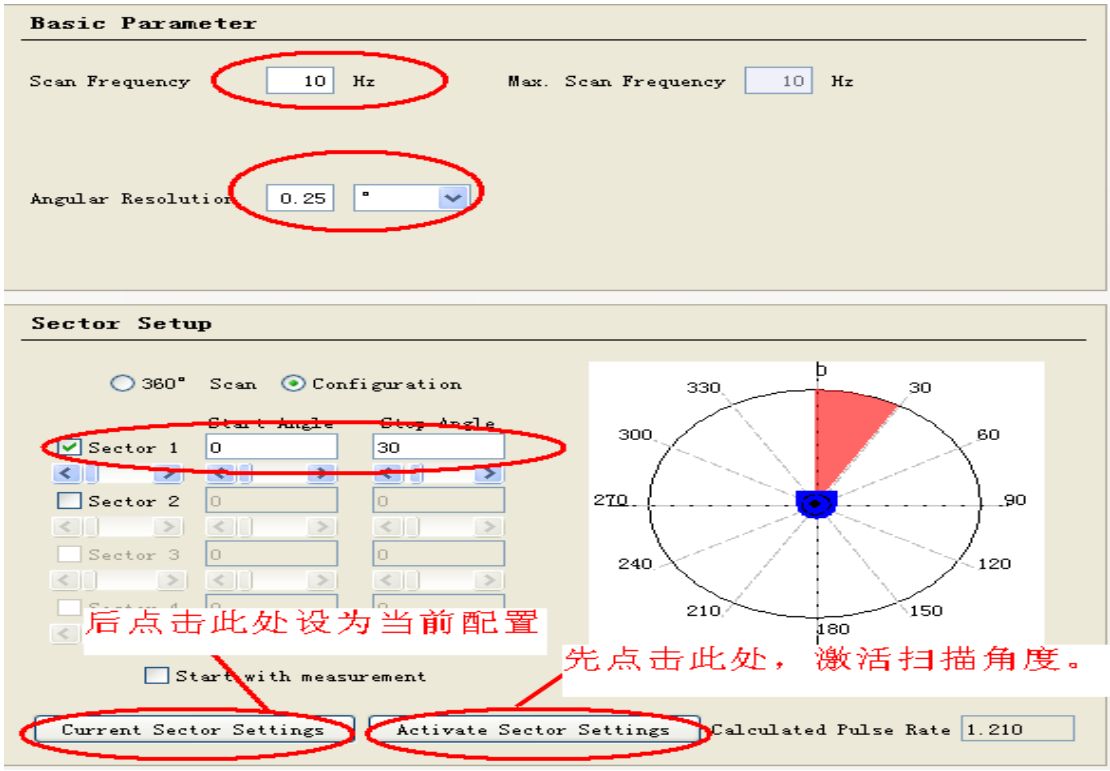
3、获取数据

02 55 53 50 00 00 00 06 03 01 00 01 03 A2 A2

三、 举例分析

扫描角度 **0-30** 度
角度分辨率 **0.25** 度
扫描频率：10HZ

SOPAS 软件参数设置，如下图所示。



数据分析：

┐1014（10：扫描头地址；14：电脑地址）007e（数据长度 126 个字）ffff（起始数据包标志字，固定格式 ffff）0003（第 3 个数据包）8301（扫描头回应命令 0301）03a2（取数据的内容）0102（01 表示取 1 圈数据，02 表示整个扫描角度分为 2 个区域 0-30 以及 31-359）a077（当前测量圈数，该值随着测量圈数持续递增）0000（固定值 0）01e4（步进角度值，30+0.25=30.25 度）0079(测量点数 121 点)0000(起始角度，0 度)02c1(第 1 个点距离值 2.75m）0000（第 1 个点角度，0 度）02b9（第 2 个点距离值 2.72m）0004(第 2 个点角度，0.25 度，依次类推)

02b7000802b5000c02b9001002b4001402b3001802ab001c02a9002002a4002402a3002802a0002c029d0030029b0034029a00380298003c029a0040029900440295004802 9a004c029700500294005402960058029200 5c029900600293006402940068029200 6c02920070028f0074028e0078029000 7c02900080028f0084028d0088028d008c028a0090028b0094028800980288009c028900a0028400a4028500a8028600ac028500b0028700b4028300b8028100bc028000c0027f00c4027d00c8027b00cc027a00d0027d00d4027d00d8027a00dc0274（第 57 个点距离值，2.45m）00e0（第 57 个点角度，14 度）0275(第 58 个点距离值，2.46m)ce0b（CRC 校验位）└

┐1014（10：扫描头地址；14：电脑地址）007e（数据长度 126 个字）0002（第 2 个数据包）00e4（第 58 个点角度值，14.25 度，依次类推）027600e8027600ec026f00f0027400f4027400f8027200fc027101000271010402720108026c010c026d0110026c0114026a0118026d011c026e0120026d0124026d0128026b012c02710130026a013402670138026901 3c026401400268014402670148026801 4c026601500268015402660158026701 5c026801600267016402640168026201 6c026401700266017402670178026101 7c026001800264018

402630188025f018c02600190025f0194025a0198025f019c025c01a0025e01a4025c01a8025f01ac025c01b0025c01b4025d01b8025e01bc025a01c0025901c4025501c8025701cc025901d0025101d4025001d8（第 119 点角度值，29.5 度）024e（第 120 个点距离值，2.3m）ed10（CRC 校验位）└

┐1014（10：扫描头地址；14：电脑地址）0005（数据长度 5 个字）0001（第 1 个数据包）01dc(第 120 个点角度值，29.75 度)0243（第 121 个点距离值，2.26m）01e0(第 121 个点角度值，30 度)75ec（CRC 校验位）└

四、 CRC 校验程序

“C” code example for the CRC calculation (RS>232/RS>422)

Example C code to calculate a CRC sum:

```
/*
*****************************************************************************
Project: generic project
File: crc16c.c
CRC16 calculation
Version: V0.0.1
Date: 20.09.1998
*****************************************************************************
Abstract:
routines for calculating a 16 bits CRC signature using the generator
polynom x^16 + x^12 + x^5 + 1 as recommended by the ITU.T V.42
(former CCITT); all routines use a table driven algorithm
-----------------------------------------------------------------------
Modification History:
0.0.1 20.09.1998
created
*****************************************************************************/
#define CRC16C_C
// includes
#include "cpu-dep.h"
//
=============================================================
// local scope defines
// (global scope in seperate header file: this_file.h)
//
=============================================================
// local scope macros
// (global macros in seperate header file: this_file.h)
```

Baud rate Nominal
bit time
Length of the
time quantum
(Tq)
TSEG1 [Tq] TSEG2 [Tq] SJW [Tq
] Sample Point
1 MBit/s 1 ⌈s 50 ns 14 3 2 15 Tq, 800 ns
500 kBit/s 2 ⌈s 100 ns 15 2 1 17 Tq, 1.70 ⌈s
250 kBit/s 4 ⌈s 250 ns 12 1 1 14 Tq , 3.5 ⌈s
125 kBit/s 8 ⌈s 500 ns 12 1 1 14 Tq, 7 ⌈s
50 kBit/s 20 ⌈s 1.25 ⌈s 12 1 1 14 Tq, 17.5 ⌈s
20 kBit/s 50 ⌈s 2.5 ⌈s 15 2 1 17 Tq, 42.5 ⌈s
10 kBit/s 100 ⌈s 6.25 ⌈s 12 1 1 14 Tq, 87.5 ⌈s
Tab. 46: CAN communication parameter: timing parameter
Operating Instructions
LD(LRS

Annex

Chapter **10**

```
//
=====================================================================
// local scope type definitions
// (global scope in seperate header file: this_file.h)
//
=====================================================================
// local scope prototype declarations (type modifier: PRIVATE)
// (global scope in seperate header file: this_file.h)
//
=====================================================================
// global scope global variable definitions (type modifier: PUBLIC)
//
=====================================================================
// local scope global variable definitions (type modifier: PRIVATE)
// XOR table for CRC algorithm, CRC-16, ITU.T X.25
// polynomial: h1021
PRIVATE const WORD crctab[256] =
{
0x0000, 0x1021, 0x2042, 0x3063, 0x4084, 0x50a5, 0x60c6, 0x70e7,
0x8108, 0x9129, 0xa14a, 0xb16b, 0xc18c, 0xd1ad, 0xe1ce, 0xf1ef,
0x1231, 0x0210, 0x3273, 0x2252, 0x52b5, 0x4294, 0x72f7, 0x62d6,
0x9339, 0x8318, 0xb37b, 0xa35a, 0xd3bd, 0xc39c, 0xf3ff, 0xe3de,
```

```
0x2462, 0x3443, 0x0420, 0x1401, 0x64e6, 0x74c7, 0x44a4, 0x5485,
0xa56a, 0xb54b, 0x8528, 0x9509, 0xe5ee, 0xf5cf, 0xc5ac, 0xd58d,
0x3653, 0x2672, 0x1611, 0x0630, 0x76d7, 0x66f6, 0x5695, 0x46b4,
0xb75b, 0xa77a, 0x9719, 0x8738, 0xf7df, 0xe7fe, 0xd79d, 0xc7bc,
0x48c4, 0x58e5, 0x6886, 0x78a7, 0x0840, 0x1861, 0x2802, 0x3823,
0xc9cc, 0xd9ed, 0xe98e, 0xf9af, 0x8948, 0x9969, 0xa90a, 0xb92b,
0x5af5, 0x4ad4, 0x7ab7, 0x6a96, 0x1a71, 0x0a50, 0x3a33, 0x2a12,
0xdbfd, 0xcbdc, 0xfbbf, 0xeb9e, 0x9b79, 0x8b58, 0xbb3b, 0xab1a,
0x6ca6, 0x7c87, 0x4ce4, 0x5cc5, 0x2c22, 0x3c03, 0x0c60, 0x1c41,
0xedae, 0xfd8f, 0xcdec, 0xddcd, 0xad2a, 0xbd0b, 0x8d68, 0x9d49,
0x7e97, 0x6eb6, 0x5ed5, 0x4ef4, 0x3e13, 0x2e32, 0x1e51, 0x0e70,
0xff9f, 0xefbe, 0xdfdd, 0xcffc, 0xbf1b, 0xaf3a, 0x9f59, 0x8f78,
0x9188, 0x81a9, 0xb1ca, 0xa1eb, 0xd10c, 0xc12d, 0xf14e, 0xe16f,
0x1080, 0x00a1, 0x30c2, 0x20e3, 0x5004, 0x4025, 0x7046, 0x6067,
0x83b9, 0x9398, 0xa3fb, 0xb3da, 0xc33d, 0xd31c, 0xe37f, 0xf35e,
0x02b1, 0x1290, 0x22f3, 0x32d2, 0x4235, 0x5214, 0x6277, 0x7256,
0xb5ea, 0xa5cb, 0x95a8, 0x8589, 0xf56e, 0xe54f, 0xd52c, 0xc50d,
0x34e2, 0x24c3, 0x14a0, 0x0481, 0x7466, 0x6447, 0x5424, 0x4405,
0xa7db, 0xb7fa, 0x8799, 0x97b8, 0xe75f, 0xf77e, 0xc71d, 0xd73c,
0x26d3, 0x36f2, 0x0691, 0x16b0, 0x6657, 0x7676, 0x4615, 0x5634,
0xd94c, 0xc96d, 0xf90e, 0xe92f, 0x99c8, 0x89e9, 0xb98a, 0xa9ab,
0x5844, 0x4865, 0x7806, 0x6827, 0x18c0, 0x08e1, 0x3882, 0x28a3,
0xcb7d, 0xdb5c, 0xeb3f, 0xfb1e, 0x8bf9, 0x9bd8, 0xabbb, 0xbb9a,
0x4a75, 0x5a54, 0x6a37, 0x7a16, 0x0af1, 0x1ad0, 0x2ab3, 0x3a92,
0xfd2e, 0xed0f, 0xdd6c, 0xcd4d, 0xbdaa, 0xad8b, 0x9de8, 0x8dc9,
0x7c26, 0x6c07, 0x5c64, 0x4c45, 0x3ca2, 0x2c83, 0x1ce0, 0x0cc1,
0xef1f, 0xff3e, 0xcf5d, 0xdf7c, 0xaf9b, 0xbfba, 0x8fd9, 0x9ff8,
0x6e17, 0x7e36, 0x4e55, 0x5e74, 0x2e93, 0x3eb2, 0x0ed1, 0x1ef0
};
//
//============================================================================
// global scope function definitions (type modifier: PUBLIC)
/*
------------------------------------------------------------------------------
Function: block_crc16_byte
```

Annex

Abstract: calculates CRC16 signature of a block of bytes

Version: 1

```
---------------------------------------------------------------------------
Return value:
type functional description
WORD CRC signature
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Importlist:
type identifier functional description
BYTE* data pointer to data block
WORD numofbytes number of bytes in data block
WORD initial_crc initial CRC value
WORD[] crctab CRC XOR table (as global variable)
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Exportlist:
type identifier functional description
none
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
*/
PUBLIC WORD block_crc16_byte
(
BYTE* data,
WORD numofbytes,
WORD initial_crc
)
{
WORD crc = initial_crc;
while( numofbytes-- )
crc = ( (crc << 8) | *data++ ) ^ crctab[crc>>8];
return crc;
}
/*
---------------------------------------------------------------------------
Function: block_crc16_word
Abstract: calculates CRC16 signature of a block of data words (16bit)
Version: 1
---------------------------------------------------------------------------
Return value:
type functional description
WORD CRC signature
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Importlist:
type identifier functional description
WORD* data pointer to data block
```

WORD numofbytes number of bytes (not words!) in data block
WORD initial_crc initial CRC value
WORD[] crctab CRC XOR table (as global variable)
- -
Exportlist:
type identifier functional description
none
- -
*/
PUBLIC WORD block_crc16_word
(
WORD* data,
WORD numofbytes,
Operating Instructions
LD(LRS

Annex

WORD initial_crc
)
{
register WORD d;
register WORD crc = initial_crc;
numofbytes >>= 1;
while(numofbytes--)
{
d = *data++;
crc = ((crc << 8) | ((BYTE)(d >> 8))) ^ crctab[crc>>8];
crc = ((crc << 8) | ((BYTE) d)) ^ crctab[crc>>8];
}
return crc;
}
/*

Function: crc16_byte
Abstract: calculates CRC16 signature of a single data byte
Version: 1

Return value:
type functional description
WORD CRC signature
- -

Importlist:
type identifier functional description
BYTE data data byte
WORD initial_crc initial CRC value
WORD[] crctab CRC XOR table (as global variable)
- -
Exportlist:
type identifier functional description
none
- -
*/
PUBLIC WORD crc16_byte
(
BYTE data,
WORD initial_crc
)
{
register WORD crc = initial_crc;
crc = ((crc << 8) | data) ^ crctab[crc>>8];
return crc;
}
/*

Function: crc16_word
Abstract: calculates CRC16 signature of a single data word (16bit)
Version: 1

Return value:
type functional description
WORD CRC signature
- -

Annex

Importlist:
type identifier functional description
WORD data data word
WORD initial_crc initial CRC value
WORD[] crctab CRC XOR table (as global variable)
- -

```
Exportlist:
type identifier functional description
none
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
*/
PUBLIC WORD crc16_word
(
WORD data,
WORD initial_crc
)
{
register WORD crc = initial_crc;
crc = ( (crc << 8) | ((BYTE)( data >> 8 ) ) ) ^ crctab[crc>>8];
crc = ( (crc << 8) | ((BYTE) data )) ^ crctab[crc>>8];
return crc;
}
//
=====================================================================
// local scope function definitions (type modifier: PRIVATE)
//
=====================================================================
// EOF crc16c.c
```

Example C code to generate the CRC table used in the example above:

```
#include <stdio.h>
#define CRC_POLY 0x1021
typedef unsigned short WORD;
WORD get_crctab_val
(
int idx
)
{
WORD value;
WORD old_val;
int k;
value = ( (WORD) idx ) << 8;
for( k=0; k<8; k++ )
{
old_val = value;
value <<= 1;
if( old_val & 0x8000 ) value ^= CRC_POLY;
}
return value;
```

```
}
void main( void )
{
FILE *out;
WORD value;
int k, i;
out = fopen( "crctab.c", "wt" );
```

Operating Instructions

LD(LRS

Annex

Chapter **10**

```
if( out == NULL )
{
puts( "\ncannot generate crctab.c !!\n\n" );
return;
}
fprintf( out, "// put header here\n\n" );
fprintf( out, "#include \"cpu-dep.h\"\n\n" );
fprintf( out, "// XOR table for CRC algorithm, CRC-16, ITU.T X.25\n" );
fprintf( out, "// polynomial: h%4x\n\n", CRC_POLY );
fprintf( out, "const WORD crctab[256] = \n" );
fprintf( out, " {" );
i = 0;
for( k=0; k<256; k++ )
{
value = get_crctab_val( k );
if( i == 0 )
fprintf( out, "\n 0x%04x,", value );
else if( k >= 248 && i >= 7 )
fprintf( out, " 0x%04x", value );
else
fprintf( out, " 0x%04x,", value );
if( ++i >= 8 ) i = 0;
}
fprintf( out, "\n };\n\n" );
fclose( out );
}
```

Unit 1, 24/F, Talent Center, No.45 Tian He Road, Guangzhou, 510075, China
Tel : 86-20-3830 3155 Fax : 86-20-3830 3248