CDCTL01A Datasheet

1 Description

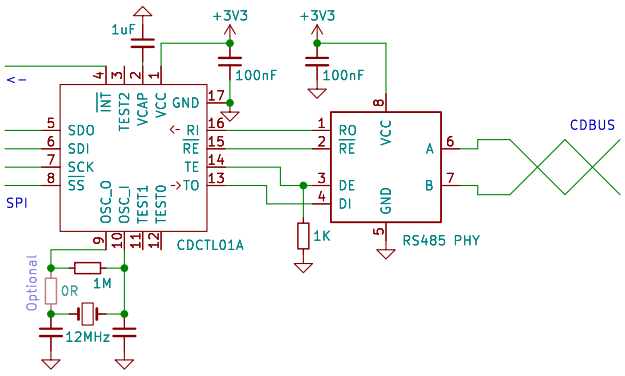
- The CDCTL01A is a SPI-to-UART controller based on the CDBUS IP core, with the UART port fixed to use the CDBUS protocol.
- The CDBUS IP Core is an open source implementation of the CDBUS Protocol.
- The CDBUS Protocol is a protocol for UART communication.

2 Features

- This controller, in addition to being used to enhance RS-485 communication, also supports M-LVDS, single-wire UART communication, UART expansion, and many other applications.
- Supports multi-master communication, peer-to-peer communication, multicast communication and many other capabilities.
- Supports multiple modes such as dual-rate arbitration mode, single-rate high-speed peer-to-peer mode, traditional mode, and full-duplex mode.
- 50Mbps maximum UART baud rate.
- 50MHz maximum SPI clock.
- 8 packet buffers for receive, 2 packet buffers for transmit. (Each buffer is 256 bytes.)
- Small package: QFN16 3x3mm.
- 3.3V (±10%) single power supply.
- Operating teperature range: -40 ~ 125 ℃.
- Crystal oscillator or external clock input.
- 4-wire or 3-wire SPI (SDO and SDI connect together).
- 5V tolerant RI input.
- Ultra-wide system clock range: 32KHz ~ 150MHz.
- Lead(Pb)-free / RoHS-compliant.

3 Applications

- Industrial Automation
- Robotics
- Automotive
- Smart City
- Consumer Electronics
- IoT



Reference Schematic

4 CDBUS Protocol

CDBUS is a protocol for Asynchronous Serial Communication, it has a 3-byte header: `[src_addr, dst_addr, data_len]`, then user data, and finally 2 bytes of CRC. (Same as MODBUS CRC.)

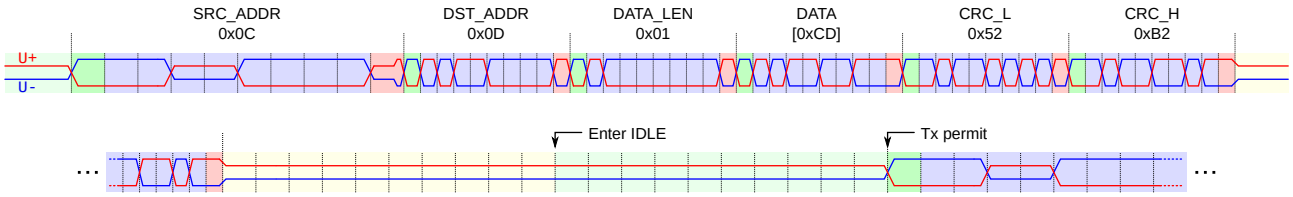
It's suitable for one-to-one communication, e.g. UART or RS-232. In this case, the address for each side are usually carefully selected and fixed, e.g: `[0x55, 0xaa, data_len, ...]`, and the backward is: `[0xaa, 0x55, data_len, ...]`. ($data\_len \leq 255$.)

The CDBUS protocol is more valuable for bus communication, e.g. RS-485, M-LVDS or Single Line UART. In this case, it supports:

4.1 Arbitration Mode (CDBUS-A)

- It introduces an arbitration mechanism that automatically avoids conflicts like the CAN bus.
- Support dual baud rate, provide high speed communication, the baud rate in the high speed phase is up to: `sysclk ÷ 3`. (e.g. 50Mbps for 150MHz sysclock.)
- Supports unicast, multicast and broadcast.
- The maximum user data size is 253 bytes.
- Hardware packing, unpacking, verification and filtering, save your time and CPU usage.
- Backward compatible with traditional RS-485 hardware (The arbitration function still works).

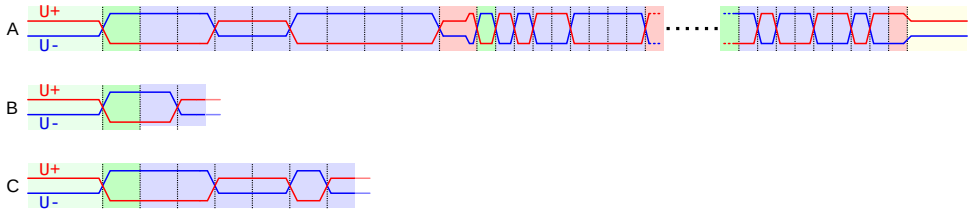
The protocol timing example, include only one user data byte:
(You can set the length of time to enter idle and wait to send.)



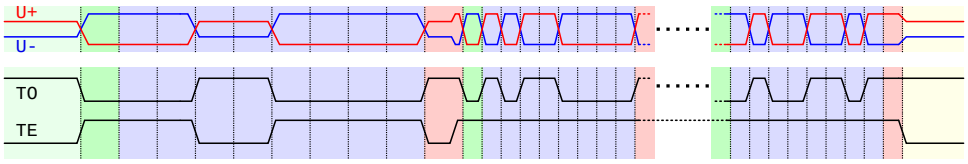
Tips:

- When a high-priority node needs to send unimportant data, you can dynamically increase the time of TX_PERMIT_LEN.

Arbitration example:



Example waveforms for TO and TE pins:

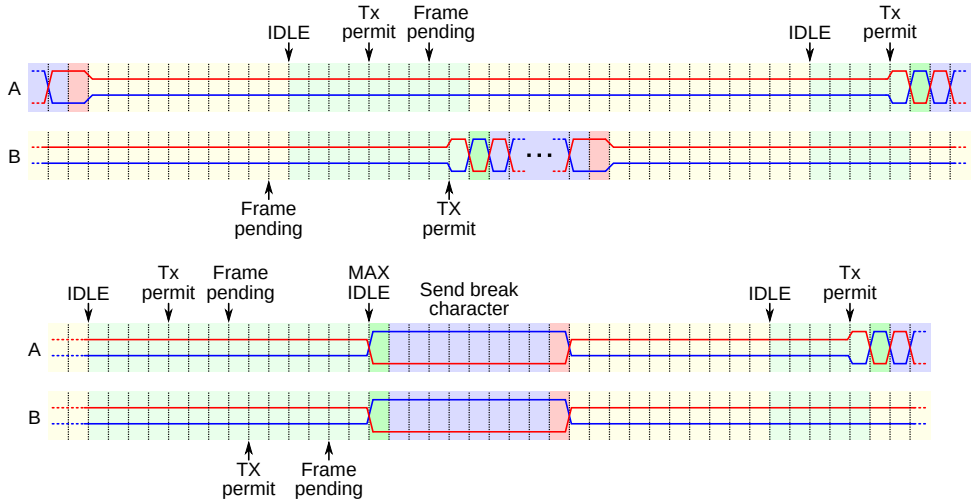


The RX receive data sample point is at 1/2 bit; the TX readback data sample point is at 3/4 bit.

4.2 Break Sync Mode (CDBUS-BS)

In CDBUS-A mode, if the low-speed part takes a lot of time, it will be a communication efficiency bottleneck.
In this case, single-rate peer-to-peer bus communication can be implemented by CDBUS-BS mode:

- The TX_PERMIT_LEN parameter is configured differently for different nodes, and the difference needs to be large enough to avoid conflicts.
- If any node has pending tx frame before TX-permit point, then start send at the TX-permit point.
- Otherwise, wait until the idle time exceeds MAX_IDLE_LEN, and when there has tx frame pending, send a break character first to bring the bus out of the idle state.

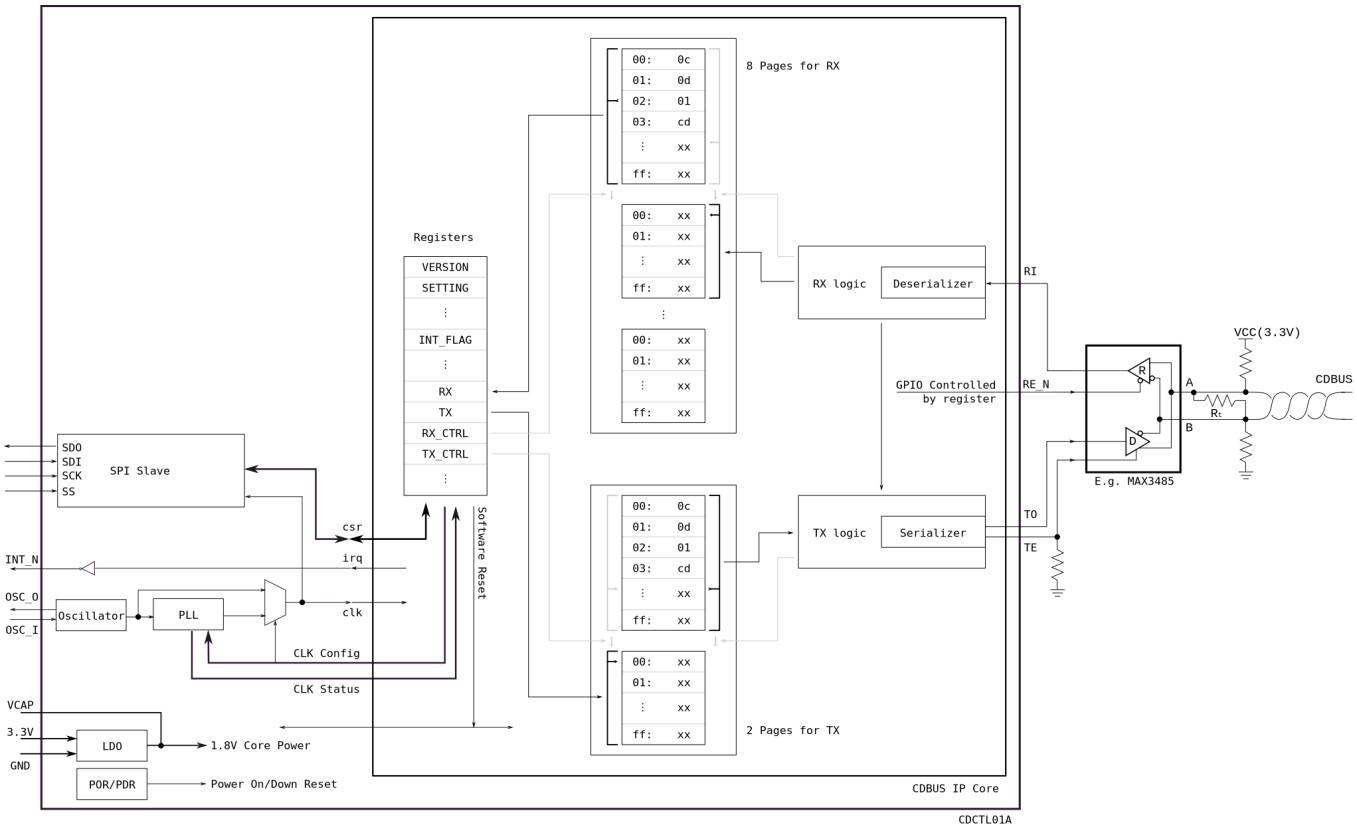


The CDBUS-BS mode is suitable for high-speed applications with few nodes, and it is also suitable for software implementation.

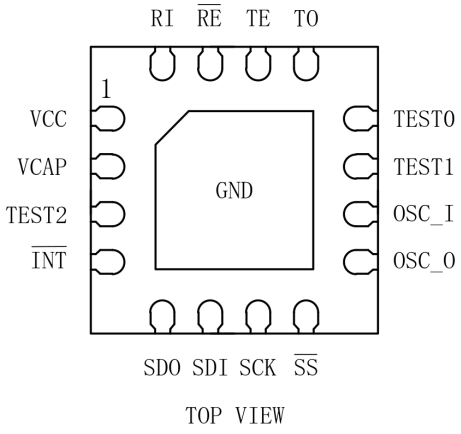
5 CDBUS IP Core

Source code and more details: https://cdbus.org

5.1 Block Diagram



6 Pin Definition

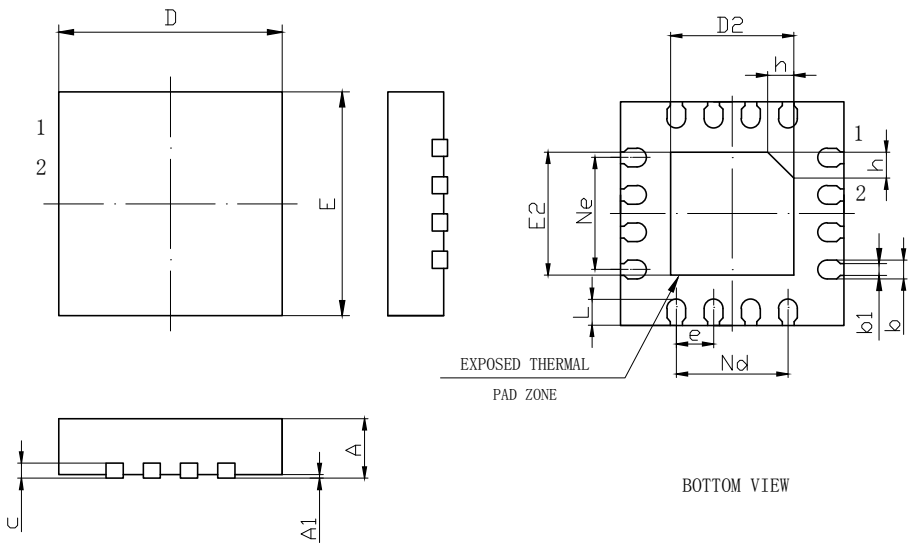


No	Name	I/O	Internal Pull	Description
17	GND			Ground
1	VCC			Supply voltage, 3.3V (±10%), bypass with 100nF ceramic capacitor to ground
2	VCAP			Internal 1.8V (±10%) 100mA LDO output, bypass with 1uF ceramic capacitor to ground
3, 11, 12	TESTx	I	Down	Left empty or connect to ground

4	$\overline{INT}$	O	Up	Interrupt pin, open-drain (default) or push-pull output
5	SDO	O	-	SPI MISO
6	SDI	I	-	SPI MOSI
7	SCK	I	-	SPI clock
8	$\overline{SS}$	I	-	SPI chip select
9	OSC_O	O	-	OSC output (left empty when using external clock input)
10	OSC_I	I	-	OSC input or external clock input
13	TO	O	-	TX output, connect to transmit pin of RS-485 PHY
14	TE	O	-	TX enable, connect to transmit enable pin of RS-485 PHY
15	$\overline{RE}$	O	-	GPIO push-pull output, typically used to control the receive enable pin of the RS-485 PHY
16	RI	I	-	RX input, connect to receive pin of RS-485 PHY (5V tolerant)

7 Specifications

7.1 Mechanical Specifications



SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	0.80	0.85	0.90
A1	—	0.02	0.05
b	0.18	0.25	0.30
b1	0.11	0.16	0.21
c	0.10	0.15	0.20
D	2.90	3.00	3.10
D2	1.55	1.65	1.75
e	0.50BSC		
Ne	1.50BSC		
Nd	1.50BSC		
E	2.90	3.00	3.10
E2	1.55	1.65	1.75
L	0.30	0.35	0.40
h	0.30	0.35	0.40

7.2 Absolute Maximum Ratings

Parameter	Min.	Max.
Storage Temperature (Ambient)	−55 ℃	150 ℃
Electrostatic Discharge Human Body Model (ESD-HBM) [1]	-	±2000 V
Electrostatic Discharge Charged Device Model (ESD-CDM) [2]	-	±750 V
High Temperature Latch-Up [3]	-	±200 mA / +1.5 VccMax

1. Refer to AEC-Q100-002. Zap 1 pulse with 0.3 sec interval.
2. Refer to AEC-Q100-011.
3. Refer to JESD78F:2022, Ta=+125℃.

7.3 Operating Conditions

Parameter	Min.	Max.
Operation Temperature	−40 ℃	125 ℃

7.4 DC Electrical Characteristics

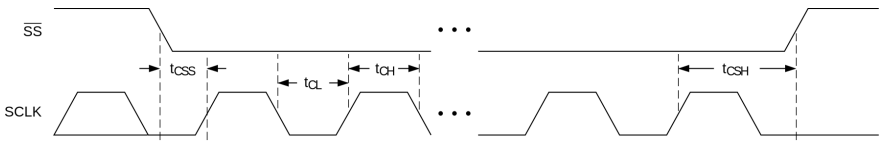
Parameter	Min.	Typ.	Max.
Supply Voltage VCC	-10%	3.3 V	+10%
V_{IL}	-0.3 V	-	0.8 V
V_{IH}	2.0 V	-	3.6 V
V_{IH} (RI pin only)	2.0 V	-	5.5 V
V_{OL}	-	-	0.4 V
V_{OH}	2.4 V	-	-
I_{OL}, I_{OH}	-	12 mA	-
I_{OL}, I_{OH} (OSC_O pin only)	-	2 mA	-
Input or I/O Leakage	-	-	+/-10 uA
I_{VCC} ($F_{SYS} = 32KHz$)	-	140 uA	-
I_{VCC} ($F_{SYS} = 12MHz$, pll off)	-	2.8 mA	-
I_{VCC} ($F_{SYS} = 12MHz$, pll on)	-	8.7 mA	-
I_{VCC} ($F_{SYS} = 60MHz$)	-	10.6 mA	-
I_{VCC} ($F_{SYS} = 150MHz$)	-	14.3 mA	-

7.5 Timing Specifications

Symbol	Parameter	Min.	Max.
$F_{CRYSTAL}$	Crystal frequency	4 MHz	32 MHz
F_{EXT}	External clock input	32 KHz	32 MHz
F_{SYS}	System clock frequency	32 KHz	150 MHz
F_{SCK}	SPI clock frequency	-	50 MHz
F_{UART}	Baud rate	-	50 Mbps

Additional restrictions on RX and TX register reads and writes: $F_{SCK} \leq F_{SYS} \times 80\%$

Writing to any register requires: $t_{CSH} \geq \frac{1}{F_{SYS} \times 80\%}$



8 Register Reference

Register Name	Addr	Access	Default	Description (8-bit width by default if not specified)
VERSION	0x00	RD	0x10	Hardware version
CLK_CTRL	0x01	RD/WR	0x00	Clock Control
SETTING	0x02	RD/WR	0x10	Configs
IDLE_WAIT_LEN	0x04	RD/WR	0x0a	Waiting time to enter idle
TX_PERMIT_LEN_L	0x05	RD/WR	0x14	Waiting time to allows sending (10 bits)
TX_PERMIT_LEN_H	0x06	RD/WR	0x00	
MAX_IDLE_LEN_L	0x07	RD/WR	0xc8	Max idle waiting time in BS mode (10 bits)
MAX_IDLE_LEN_H	0x08	RD/WR	0x00	
TX_PRE_LEN	0x09	RD/WR	0x01	Enable TE how long ahead than TO output (2 bits)

FILTER	0x0b	RD/WR	0xff	Local address
DIV_LS_L	0x0c	RD/WR	0x67	Low-speed rate setting (16 bits)
DIV_LS_H	0x0d	RD/WR	0x00	
DIV_HS_L	0x0e	RD/WR	0x67	High-speed rate setting (16 bits)
DIV_HS_H	0x0f	RD/WR	0x00	
INT_FLAG	0x10	RD	n/a	Status
INT_MASK	0x11	RD/WR	0x00	Interrupt mask
RX	0x14	RD	n/a	Read RX page
TX	0x15	WR	n/a	Write TX page
RX_CTRL	0x16	WR	n/a	RX control
TX_CTRL	0x17	WR	n/a	TX control
RX_ADDR	0x18	RD/WR	0x00	RX page read pointer (rarely used)
RX_PAGE_FLAG	0x19	RD	n/a	RX page flag
FILTER_M0	0x1a	RD/WR	0xff	Multicast filter0
FILTER_M1	0x1b	RD/WR	0xff	Multicast filter1
PLL_ML	0x30	RD/WR	0x12	PLL M[7:0] (M: 9 bits)
PLL_OD_MH	0x31	RD/WR	0x20	PLL OD and M[8]
PLL_N	0x32	RD/WR	0x00	PLL N (5 bits)
PLL_CTRL	0x33	RD/WR	0x01	PLL Control
PIN_INT_CTRL	0x34	RD/WR	0x00	$\overline{\text{INT}}$ pin Control
PIN_RE_CTRL	0x35	RD/WR	0x00	$\overline{\text{RE}}$ pin Control
CLK_STATUS	0x36	RD	0x01	Clock status

8.1 CLK_CTRL:

FIELD	DESCRIPTION
[7]	Soft reset: write 1 to reset device
[0]	Clock select: 0: OSC input, 1: PLL output

8.2 SETTING:

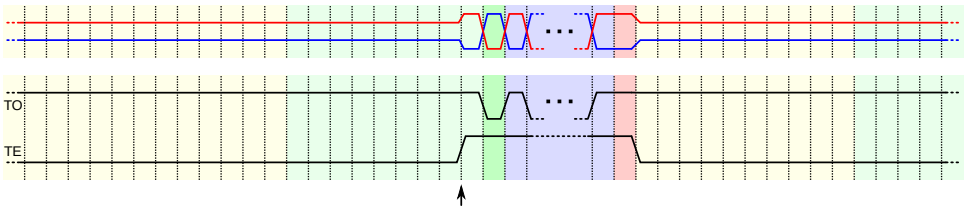
FIELD	DESCRIPTION
[6]	Full duplex mode
[5]	Break Sync mode
[4]	Enable arbitration
[3]	Save broken frame
[2]	CRC maintained by user
[1]	Invert TO pin output
[0]	Enable push-pull output for TO and TE pin

The TO pin defaults to an open-drain output and the TE pin defaults to a high-resistance output.

[6]	[5]	[4]	DESCRIPTION
0	0	1	CDBUS-A mode (default)
0	1	0	CDBUS-BS mode
1	0	0	Full-duplex mode
0	0	0	Traditional half-duplex mode

8.3 TX_PRE_LEN:

Example waveforms for TO and TE pins (TX_PRE_LEN = 1 bit):



Unused for Arbitration mode and the break character automatically generated by BS mode.

8.4 FILTERS:

Match from top to bottom:

SRC_ADDR	DST_ADDR	FILTER	FILTER_Mx	Receive or not	Remarks
not care	not care	255	not care	Receive	Promiscuous mode
= FILTER	not care	!= 255	not care	Drop	Avoid loopback
!= FILTER	255	not care	not care	Receive	Broadcast
!= FILTER	!= 255	not care	any = DST_ADDR	Receive	Multicast
!= FILTER	!= 255	= DST_ADDR	not care	Receive	Unicast
not care	!= 255	!= DST_ADDR	all != DST_ADDR	Drop	

The default value 0xff of FILTER_Mx means not enabled.

8.5 DIV_xx_x:

Baud rate divider value:

$$div\_xx[15:0] = \frac{sysclk}{baudrate} - 1$$

The minimum value is 2.
For single rate, DIV_HS needs to be set to the same value as DIV_LS.

8.6 INT_FLAG:

FIELD	DESCRIPTION
[7]	1: TX error: TX is 0, but RX is sampled as 1
[6]	1: TX collision detected
[5]	1: TX page released by hardware
[4]	1: RX error: frame broken
[3]	1: RX lost: no empty page for RX
[2]	1: Break character received
[1]	1: RX page ready for read
[0]	1: Bus in IDLE mode

Reading this register will automatically clear bit7, bit6, bit4, bit3 and bit2.

8.7 INT_MASK:

```
irq = ((INT_FLAG & INT_MASK) != 0)
int_n = !irq
```

8.8 RX_CTRL:

FIELD	DESCRIPTION
[4]	Reset RX block
[1]	Switch RX page
[0]	Reset RX page read pointer

8.9 TX_CTRL:

FIELD	DESCRIPTION
[5]	Send break character
[4]	Abort TX
[1]	Switch TX page
[0]	Reset TX page write pointer

8.10 RX_PAGE_FLAG:

Value zero indicate the frame in current RX page is correct;
Non-zero indicate the pointer of last received byte of the disturbed frame, include CRC.
Always zero if "save broken frame" is not enabled.

8.11 PLL_OD_MH:

FIELD	DESCRIPTION
[5:4]	PLL OD
[0]	PLL M[8]

8.12 PLL_CTRL:

FIELD	DESCRIPTION
[4]	PLL enable: 0: disable pll, 1: turn on pll
[0]	PLL sleep: 1: sleep, 0: run

Reserved bits in this register must remain zero.

PLL output clock:

$$pll\_output = \frac{osc\_input}{pll\_n + 2} \times (pll\_m + 2) \times \frac{1}{2^{(pll\_od[1]+pll\_od[0])}}$$

When the OSC input is equal to 12MHz, the default PLL parameters correspond to an output of 60MHz.

The PLL parameters needs to meet the following conditions:

$$1MHz \leq \frac{osc\_input}{pll\_n + 2} \leq 15MHz$$

$$100MHz \leq \frac{osc\_input}{pll\_n + 2} \times (pll\_m + 2) \leq 500MHz$$

Switch to using the PLL clock:

- Change PLL N, M and OD values if need
- Write 0x10 to PLL_CTRL
- Write 0x01 to CLK_CTRL

8.13 PIN_INT_CTRL:

FIELD	DESCRIPTION
[4]	Output mode: 0: open-drain, 1: push-pull

8.14 PIN_RE_CTRL:

FIELD	DESCRIPTION
[4]	Output mode: 0: disabled (high impedance), 1: push-pull
[0]	Output value: 0: low, 1: high

8.15 CLK_STATUS:

FIELD	DESCRIPTION
[2]	Clock switching status: 1: completed, 0: in progress
[1]	System clock is PLL output
[0]	System clock is OSC input

9 SPI Interface

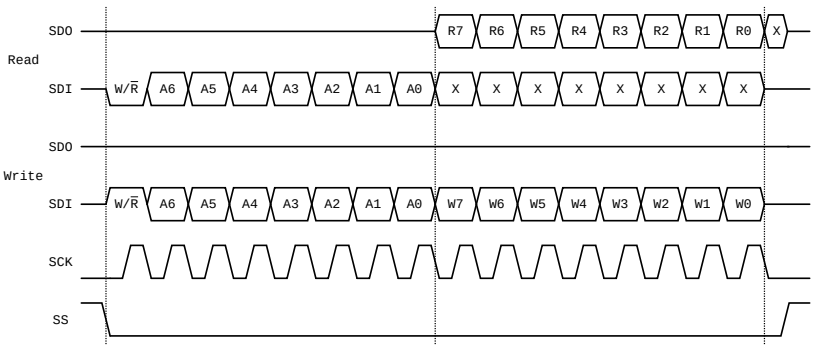
Burst read and write are useful for accessing REG_RX and REG_TX.

Read or write depend by bit W/$\overline{R}$:

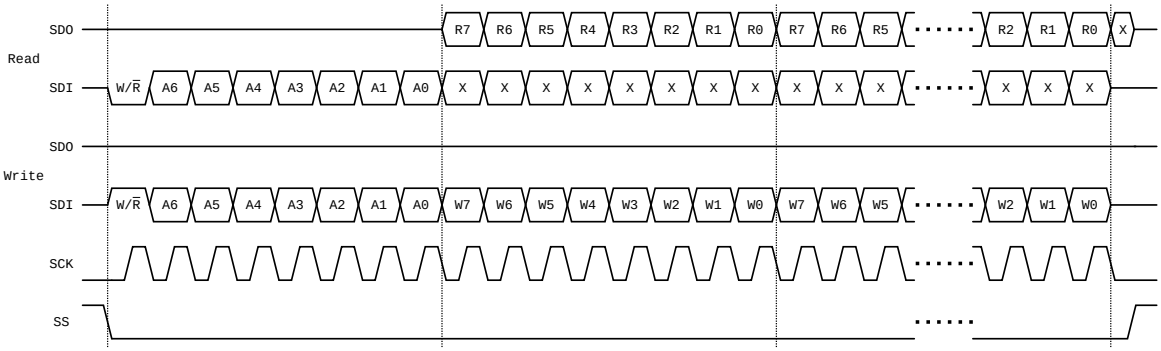
0: Read
1: Write

FIELD	DESCRIPTION
Ax	Register address
Wx	Write data
Rx	Read data
X	Don't care

Read or write single byte:



Burst read or write:



10 Operate Demonstration

10.1 Init

```
cd_write(REG_CLK_CTRL, 0x80);                   // Soft reset
cd_write(REG_PIN_RE_CTRL, 0x10);                // Set RE_N pin to low
cd_write(REG_SETTING, 0x11);                    // Enable push-pull output
cd_write(REG_FILTER, 0x0c);                     // Set FILTER

// Set baudrates
cd_write(REG_DIV_LS_L, 11);                     // 1 Mbps @ 12MHz sysclk
cd_write(REG_DIV_LS_H, 0);
cd_write(REG_DIV_HS_L, 2);                      // 4 Mbps @ 12MHz sysclk
cd_write(REG_DIV_HS_H, 0);

// cd_write(REG_RX_CTRL, 0x11);                 // Reset RX buffers and flags (optional)

// Enable interrupts (optional)
// cd_write(REG_INT_MASK, BIT_FLAG_TX_ERROR | BIT_FLAG_RX_ERROR \
            | BIT_FLAG_RX_LOST | BIT_FLAG_RX_PENDING);
```

10.2 TX

```
uint8_t tx_buf[] = {
    0x0c, 0x0d, 0x02,                           // src_addr, dst_addr, data_len
    0x01, 0x00                                  // data[0], data[1]
};

cd_write_chunk(REG_TX, tx_buf, tx_buf[2] + 3);  // Write frame without CRC
while (!(cd_read(REG_INT_FLAG) & 0x20));        // Make sure we can successfully switch to the next page
cd_write(REG_TX_CTRL, 0x03);                    // Trigger send by switching TX page
```

10.3 RX

```
while (!(cd_read(REG_INT_FLAG) & 0x02));        // Wait for RX page ready
cd_read_chunk(REG_RX, rx_buf, 3);               // Read frame header
cd_read_chunk(REG_RX, rx_buf + 3, rx_buf[2]);   // Read frame data
cd_write(REG_RX_CTRL, 0x03);                    // Finish read by switching RX page
```

11 Copyright Statement

12 Contact Information

- Sales and customer support: sales@dukelec.com
- Technical support: support@dukelec.com
- Business corporation: info@dukelec.com
- Website: `https://dukelec.com`

13 Change History

- 20231125 (v1.1): Add ESD and Latch-Up related data.
- 20230802 (v1.0): Init.