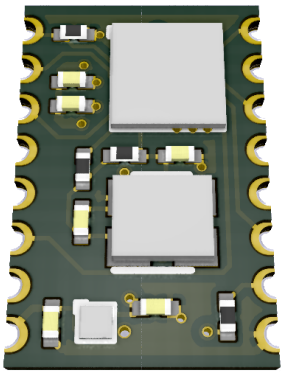
CDCTL-B1 數據手冊

DUKELEC

August 14, 2017



Contents

1 功能描述 **3**
1.1 概述 . . . 3
1.2 特性 . . . 3

2 CDBUS 協議 **3**

3 硬件 **4**
3.1 電路參考 . . . 4
3.2 內部結構 . . . 5
3.3 引腳定義 . . . 5
3.4 尺寸規格 . . . 6
3.5 極限參數 . . . 6
3.6 建議工作參數 . . . 6
3.7 直流參数 . . . 6

4 寄存器列表 **7**

5 流程圖 **10**
5.1 RX . . . 11
5.2 TX . . . 12

6 設備接口 **12**
6.1 SPI 13
6.2 I2C 13

7 操作示例 **13**
7.1 初始化 13
7.1.1 兼容模式和傳統模式 14
7.2 TX 14
7.3 RX 14

8 版權說明 **15**

1 功能描述

1.1 概述

CDBUS 是一款基於 RS485 的通訊協議，它只定義了 ISO/OSI 模型的數據鏈路層。
CDBUS 協議由 DUKELEC 公司於 2009 年設計，以便捷、多主對等、高速通訊爲目標。

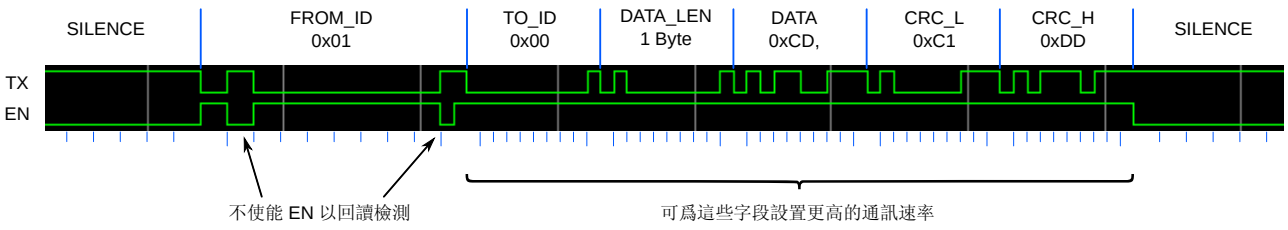
1.2 特性

CDCTL-B1 型號模組支持：

- 支持 CDBUS 多主對等通訊協議，使用發送方地址按位仲裁
- 每個數據幀可裝載 253 字節數據
- 8 個接收緩衝頁，2 個發送緩衝頁，每個頁 256 字節
- 16 位硬件 CRC 校驗
- 波特率範圍 458 bps 至 10 Mbps（如果需要可以支持更高）
- 仲裁字段和後續數據可設定不同波特率
- 可兼容傳統 RS485 總線設備
- 支持 SPI 和 I2C 接口
- 配置和使用簡單

2 CDBUS 協議

CDBUS 示例時序：



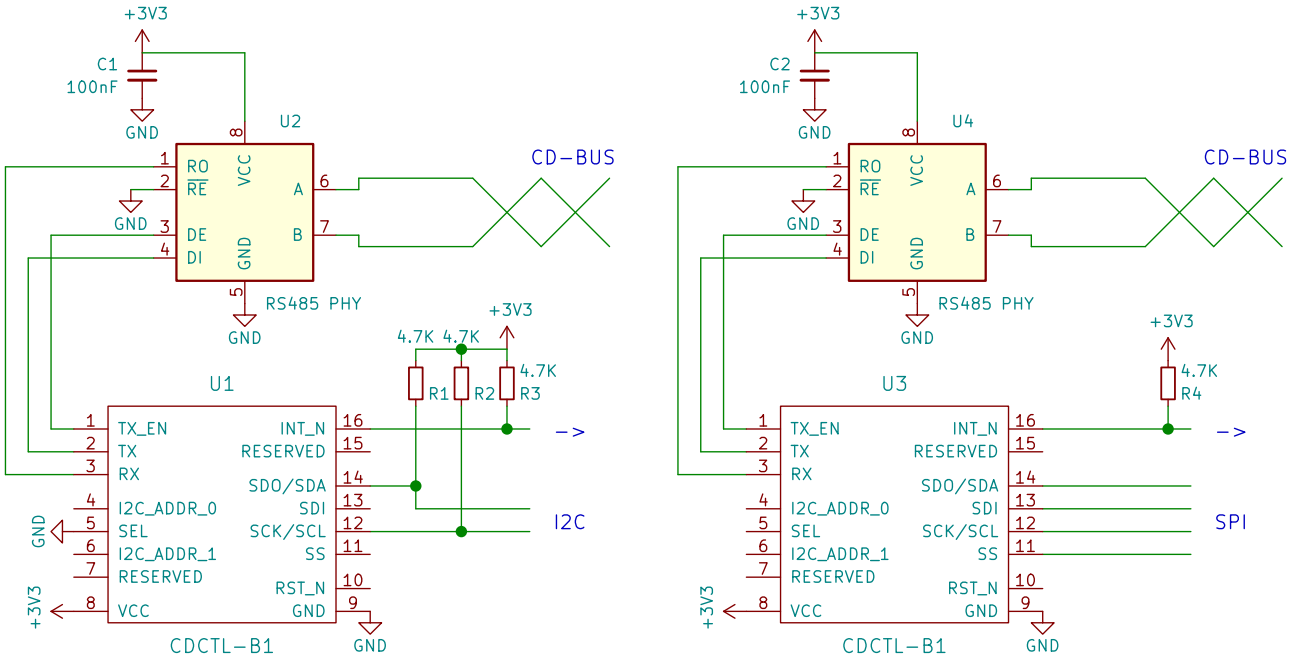
字段	長度（字節）	用途
SILENCE	0~25.5 默認： 2 (20 bits)	分隔數據幀 總線在數據幀結束後繼續保持 SILENCE 位長的時間爲 1, 總線便進入空閒模式。 當總線爲空閒模式才允許接收。 當總線持續空閒一段時間（默認 10 bit）後才開始發送。

FROM_ID	1	發送方 ID 在發送此字段時，所有的 1 不使能 TX_EN 腳，從而回讀總線狀態以判斷是否有更高優先級節點同時在發送。若有，該節點立即停止並推後發送；若無，在最後一次回讀後使能 TX_EN 並保持到數據幀結束。此字段會在所有數據 1 的中間位置進行回讀，因爲發送與接收存在延時，所以通常設置低於 1 Mbps.
TO_ID	1	接收方 ID, 255 爲廣播幀。
DATA_LEN	1	裝載的數據長度，範圍：0~253 字節，每個緩存頁是 256 字節，最前 3 字節被 FROM_ID、TO_ID 和 DATA_LEN 使用。
DATA	0~253	裝載的數據
CRC_L	1	CRC 低 8 位，與 Modbus RTU 使用相同的 CRC 標準。
CRC_H	1	CRC 高 8 位

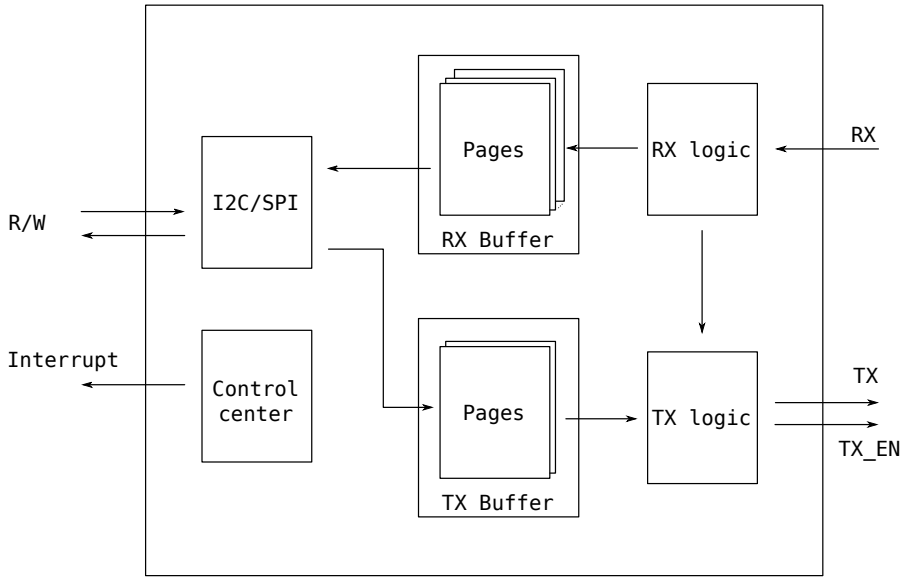
CDBUS 協議只定義數據幀格式，不規定所裝載數據格式；只支持单播和廣播，不支持多播；只提供硬件避讓、避讓後自動重傳，而應答及出錯處理則由上層軟件負責。

3 硬件

3.1 電路參考



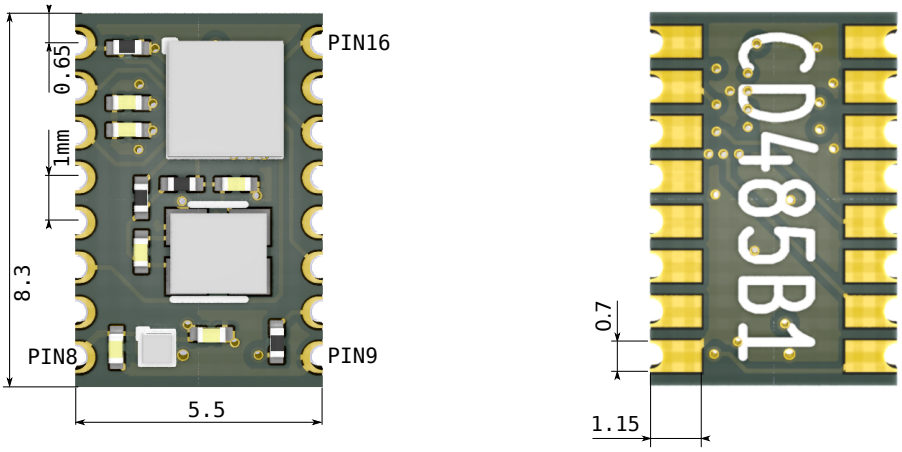
3.2 內部結構



3.3 引腳定義

引腳	定義	I/O	上下拉	描述
1	TX_EN	O	下拉 (10 kOhm)	使能腳，連接 RS485 收發器
2	TX	O	-	發送腳，連接 RS485 收發器
3	RX	I	-	接收腳，連接 RS485 收發器
4, 6	I2C_ADDR	I	上拉 (40 kOhm)	設置 I2C 地址
5	SEL	I	上拉 (40 kOhm)	輸入高選擇 SPI 模式；低爲 I2C 模式
7, 15	RESERVED			留空
8	VCC			電源
9	GND			地
10	RST_N	I	上拉 (10 kOhm)	復位，≥ 200 ns 低脈衝復位（可選）
11	SS	I	上拉 (40 kOhm)	SPI 片選
12	SCK/SCL	I	-	SPI/I2C 時鐘
13	SDI	I	-	SPI MOSI
14	SDO/SDA	I/IO	-	SPI MOSI / I2C SDA
16	INT_N	O	-	中斷，低有效，開漏輸出

3.4 尺寸規格



3.5 極限參數

參數	最小	最大
VCC 電壓	−0.5 V	3.60 V
環境溫度	−65 ℃	150 ℃
節溫 (T_J)	−65 ℃	125 ℃

3.6 建議工作參數

參數	最小	最大
VCC 電壓	3.14 V	3.46 V
節溫	−40 ℃	100 ℃
上電速度	0.6 V/ms	10 V/ms

3.7 直流參數

參數	最小	典型	最大
V_{IL}	-0.3 V	-	0.8 V
V_{IH}	2.0 V	-	VCC + 0.2 V
V_{OL}	0.2 V	-	0.4 V
V_{OH}	VCC - 0.4 V	-	VCC - 0.2 V
I_{OL}	-	-	8 mA
I_{OH}	-	-	-8 mA
Input 或 I/O 漏電流	-	-	+/-10 uA

I/O 寄生電容 (25℃, 1.0 MHz)	-	6 pF	-
器件功耗	-	-	15 mW
V_{PORUP}（上電復位電壓閾值）	0.7 V	-	1.6 V
V_{PORDN}	-	-	1.6 V

4 寄存器列表

地址	名稱	R/W	說明
0x00	VERSION	R	硬件版本，當前爲：0x01
0x01	SETTING	RW	Bits: bit0 TX_PUSH_PULL 如果關閉，TX 爲開漏輸出，且 TX_EN 懸空不使用。 bit1 TX_INVERT 如果設置，TX 將會反向輸出。 bit2 USER_CRC 關閉硬件 CRC 如果關閉：用戶需要自行計算兩字節 CRC 並追寫在數據之後；在讀完數據之後再多讀兩字節 CRC 數據以便自行校驗。用戶數據最大長度將會降至 251 字節。 bit3 NO_DROP 如果設置，當接收出錯置位 RX_ERROR 標誌時會同時保留出錯的數據幀。 通過 RX_PAGE_FLAG 判斷當前 RX 緩存頁中的數據幀是否有錯。 bit[5:4] TX_EN_ADVANCE（僅 NO_ARBITRATE 置位時有效） TX_EN 提前 TX 使能的位長（額外加上 1 個系統時鐘週期）。 bit6 NO_ARBITRATE 關閉仲裁功能，輸出時 TX_EN 一直有效。 默認：x0010000 (x: 不關心，寫 0)
0x02	SILENCE_LEN	RW	總線在數據幀結束後繼續保持 SILENCE 位長的時間爲 1, 總線便進入空閒模式，默認 20 (bits)
0x03	TX_DELAY	RW	當總線進入空閒並保持此段時間，才允許發送，默認 10 (bits) 可以爲越高優先級節點設置越低的值，但至少要保留 1 bit, 以確保所有節點都有足夠時間檢測到總線 IDLE 狀態。

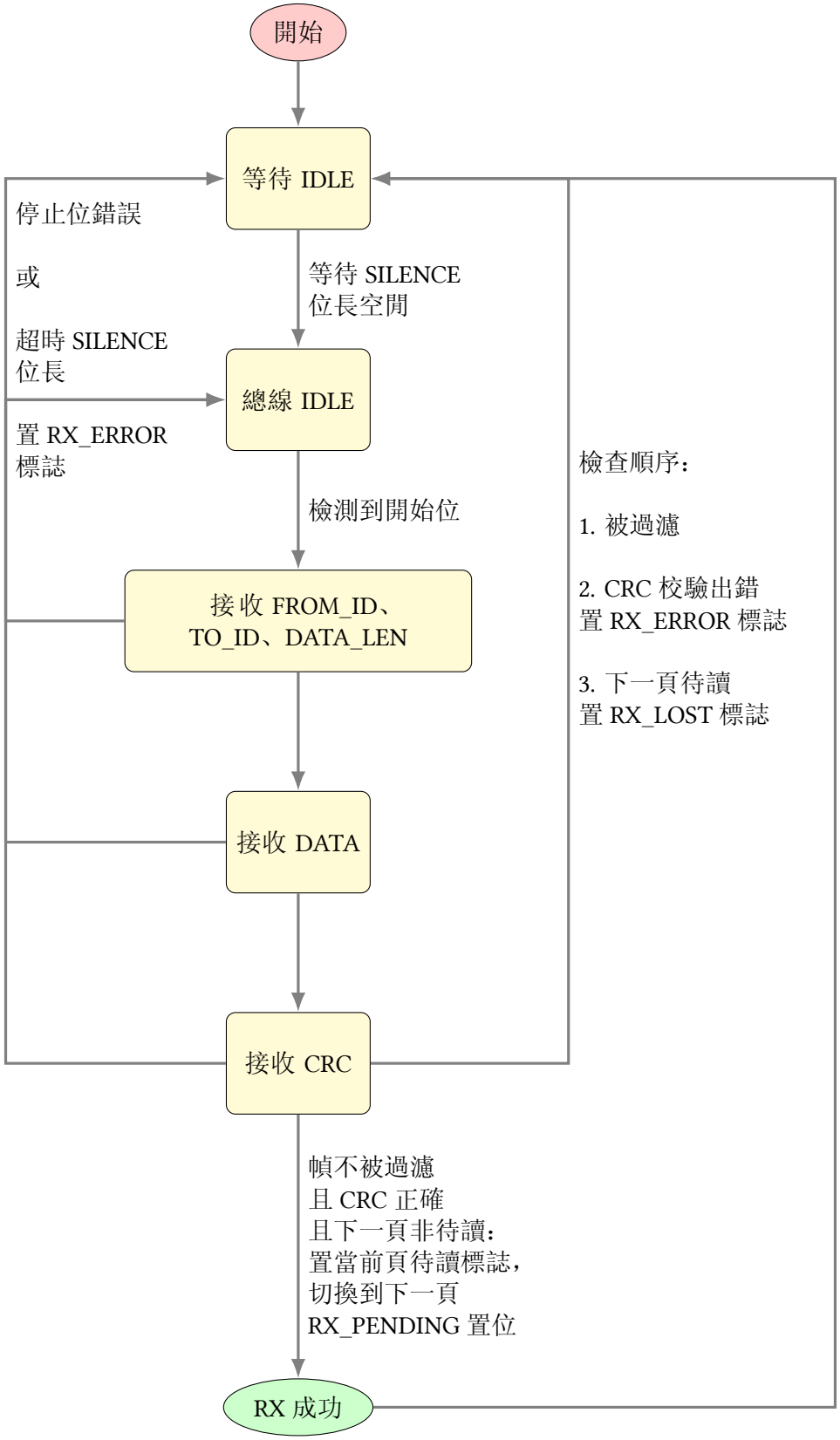
<table>
<tr><td>0x04</td><td>SELF_ID</td><td>RW</td><td>僅用做接收過濾：（由上至下進行匹配）
<table><tr><th>FROM_ID</th><th>TO_ID</th><th>SELF_ID</th><th>接收或丟棄</th></tr><tr><td>not care</td><td>not care</td><td>255</td><td>接收（嗅探模式）</td></tr><tr><td>= SELF_ID</td><td>not care</td><td>!= 255</td><td>丟棄（避免環路）</td></tr><tr><td>!= SELF_ID</td><td>255</td><td>not care</td><td>接收（廣播）</td></tr><tr><td>!= SELF_ID</td><td>!= 255</td><td>= TO_ID</td><td>接收（點對點）</td></tr><tr><td>not care</td><td>!= 255</td><td>!= TO_ID</td><td>丟棄</td></tr></table>默認：255</td></tr>
<tr><td>0x05</td><td>PERIOD_LS_L</td><td>RW</td><td>PERIOD_LS 低 8 位 (LS: Low Speed)
爲 SILENCE、TX_DELAY 和 FROM_ID 字段設置波特率（也包括 EN_ADVANCE）。
計算公式 $factor = sysclock \div bond\_rate - 1$
舉例：當前系統時鐘是 30 MHz，設置 259 最接近 115200 bps（默認 259）</td></tr>
<tr><td>0x06</td><td>PERIOD_LS_H</td><td>RW</td><td>PERIOD_LS 高 8 位，共 16 位</td></tr>
<tr><td>0x07</td><td>PERIOD_HS_L</td><td>RW</td><td>PERIOD_HS 低 8 位 (LS: Low Speed, 默認 259)
爲 TO_ID、DATA_LEN、DATA 和 CRC_L/H 字段設置波特率。</td></tr>
<tr><td>0x08</td><td>PERIOD_HS_H</td><td>RW</td><td>PERIOD_HS 高 8 位，共 16 位</td></tr>
</table>

0x09	INT_FLAG	R	中斷標誌： bit0 BUS_IDLE 指示總線是否進入 IDLE 模式。 bit1 RX_PENDING 指示 RX 緩存是否有頁待讀。 寫 1 到 RX_CTRL[CLR_RX_PENDING] 清除當前頁待讀標誌。 bit2 RX_LOST 當一個幀正確抵達且不被過濾，但卻因爲沒有更多頁用做下一次接收而被丟棄，此標誌置位。 寫 1 到 RX_CTRL[CLR_RX_LOST] 清此標誌。 bit3 RX_ERROR 當一個不被過濾的幀停止位錯誤、超時或校驗錯誤，此標誌置位。 寫 1 到 RX_CTRL[CLR_RX_ERROR] 清此標誌。 bit4 TX_BUF_CLEAN 指示是否所有 TX 緩存頁都未標記爲待發送。 bit5 TX_CD 當檢測到有更高優先級節點時推後發送並置此標誌。 寫 1 到 TX_CTRL[CLR_TX_CD] 清此標誌。 此位用作調試使用。 bit6 TX_ERROR 檢測到衝突後，當總線再次空閒超過設定時間硬件會自動重發，但如果連續重發 3 次都發生衝突，則取消發送，並置位此標誌。 寫 1 到 TX_CTRL[CLR_TX_ERROR] 清此標誌。
0x0A	INT_MASK	RW	中斷允許 當 INT_FLAG & INT_MASK != 0 時 INT_N 輸出低，否則輸出高阻（默認 0x00）
0x0B	RX	R	讀 RX 緩存頁數據，地址自動增加 共有 8 個 RX 緩存頁，每一頁 256 字節。 當硬件端成功接收到不被過濾的幀：如果下一頁未標記爲待讀（可用作下一次接收），將當前頁標記爲待讀並切換到下一頁；否則丟棄該幀並置位 RX_LOST. RX_PENDING 位指示用戶端當前頁待讀，寫 1 到 CLR_RX_PENDING 清除當前頁的待讀標誌並切換到下一頁。寫 1 到 RST_RX 清除所有頁的待讀標誌，並復位接收邏輯。

0x0C	TX	W	寫 TX 緩存頁數據，地址自動增加 共有 2 個 TX 緩存頁，每一頁 256 字節。 當用戶寫完數據，需要等待 TX_BUF_CLEAN 置位，然後才可以通過 START_TX 置位當前頁的待發送標誌，並自動切換到下一頁（否則什麼都不會發生）。 當頁的待發送標誌被置上，硬件將會啓動發送，當發送完畢，硬件端清頁的待發送標誌並切換到下一頁。
0x0D	RX_CTRL	W	RX 控制: bit0 RST_RX_POINTER 寫 1 歸零當前 RX 緩存頁的讀指針 bit1 CLR_RX_PENDING (自動包含 bit0) bit2 CLR_RX_LOST bit3 CLR_RX_ERROR bit4 RST_RX (自動包含 bit0, 2, 3)
0x0E	TX_CTRL	W	TX 控制: bit0 RST_TX_POINTER 寫 1 歸零當前 TX 緩存頁的寫指針 bit1 START_TX (自動包含 bit0) bit3 CLR_TX_CD bit4 CLR_TX_ERROR
0x0F	RX_ADDR	RW	讀寫當前 RX 緩存頁的讀指針
0x10	RX_PAGE_FLAG	R	（僅 NO_DROP 置位時使用） 0 代表當前 RX 緩存頁中的數據幀正確； 非 0 表示數據幀錯誤，其值指示最後接收到的字節地址，包含 CRC 字段。

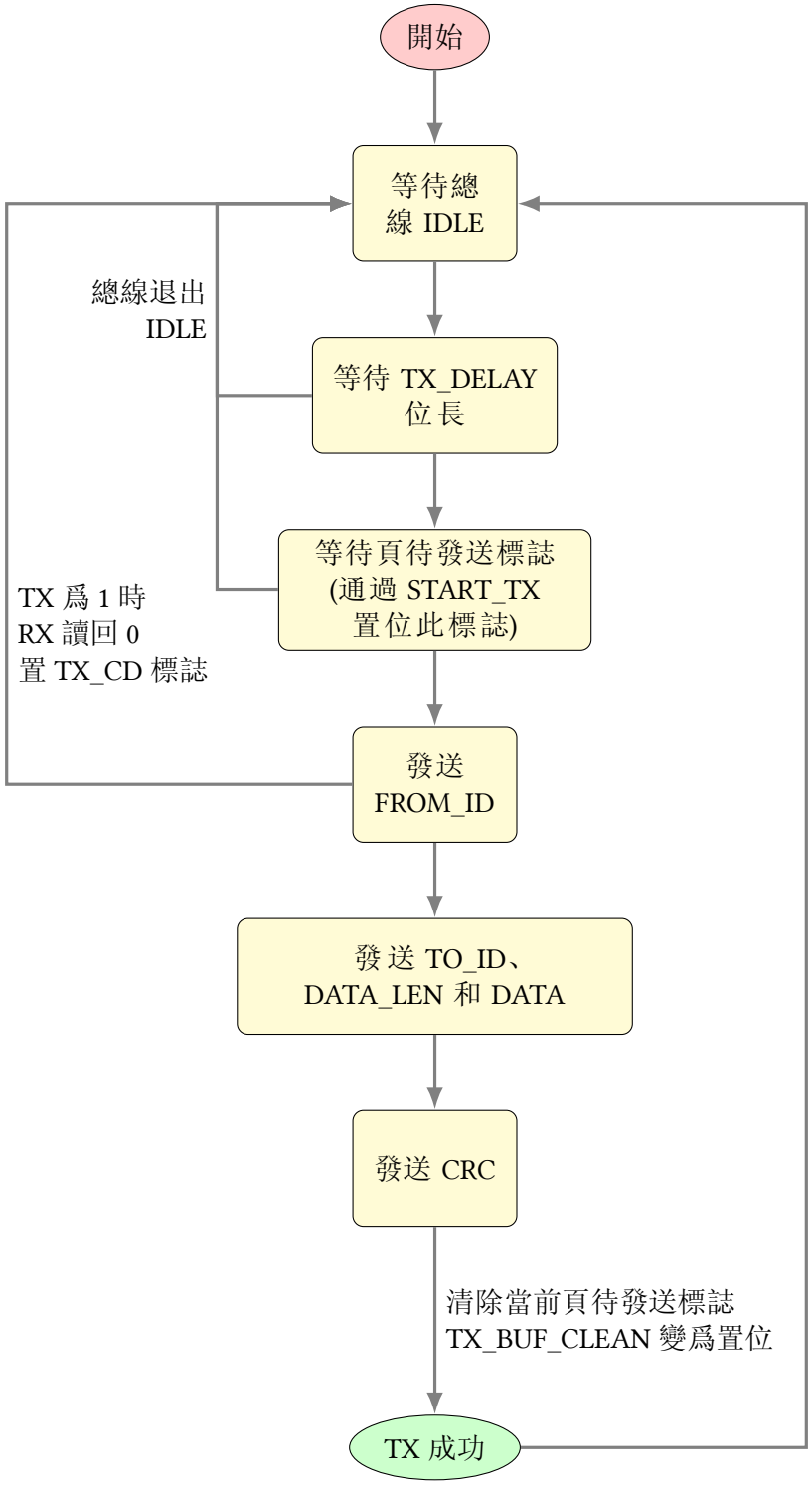
5 流程圖

5.1 RX



如果當前幀接收不夠兩個字節，或者將會被丟棄則不會設置 RX_ERROR 標誌。

5.2 TX



6 設備接口

SPI 和 I2C 頻率低於 $sysclock \div 10$.

除了 RX 和 TX, 其餘寄存器通常只讀寫 1 字節。

6.1 SPI

讀寫：

```
start (NSS = 0)
Write reg address with bit7:  0: read, 1: write
Read or write arbitrary length of data
stop (NSS = 1)
```

6.2 I2C

```
Write address: 0xc0 | (I2C_ADDR << 1)
Read address:  0xc1 | (I2C_ADDR << 1)
```

I2C_ADDR is the value of I2C_ADDR_n pins.

寫：

```
start
write the write address
write 1 byte reg address
write arbitrary length of data
stop
```

讀：

```
start
write the write address
write 1 byte reg address
restart (or stop + start)
write the read address
read arbitrary length of data, ACK all bytes except last byte
stop
```

7 操作示例

7.1 初始化

```
// enable OUTPUT
cd_write(REG_SETTING, TX_PUSH_PULL);

// set SELF_ID
cd_write(REG_SELF_ID, 0xcd);

// set bondrate
cd_write(REG_PERIOD_LS_L, 39); // 750000 bps
cd_write(REG_PERIOD_LS_H, 0);
cd_write(REG_PERIOD_HS_L, 2);  // 10 Mbps
cd_write(REG_PERIOD_HS_H, 0);
```

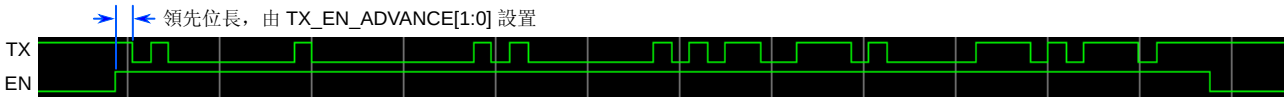
```
// clean RX buffer
cd_write(REG_RX_CTRL, RST_RX);

// enable interrupt
// cd_write(REG_INT_MASK, TX_ERROR | RX_ERROR | RX_LOST | RX_PENDING);
```

7.1.1 兼容模式和傳統模式

PERIOD_LS 和 PERIOD_HS 設置相同爲兼容模式。

進一步置位 NO_ARBITRATE 進入傳統模式：



7.2 TX

```
header_buf[0] = 0xcd; // FROM_ID
header_buf[1] = 0x02; // TO_ID
header_buf[2] = 12;   // DATA_LEN

cd_write_chunk(REG_TX, header_buf, 3);             // write HEADER
cd_write_chunk(REG_TX, data_buf, header_buf[2]);   // write DATA

while (cd_read(REG_INT_FLAG) & TX_BUF_CLEAN == 0); // make sure TX_BUF_CLEAN is set
cd_write(REG_TX_CTRL, START_TX); // sent frame
```

7.3 RX

```
while (cd_read(REG_INT_FLAG) & RX_PENDING == 0);

cd_read_chunk(REG_RX, header_buf, 3);           // read HEADER
cd_read_chunk(REG_RX, data_buf, header_buf[2]); // read DATA

cd_write(REG_RX_CTRL, CLR_RX_PENDING);          // release page
```

8 版權說明

CDBUS 是一個相當開放的協議，硬件實現也相對簡單，除了芯片生產商需要支付少量版權費，其餘任何人都可以免費使用此協議及其變種，只需要在產品說明中保留原始的版權信息。

聯絡：info@dukelec.com