產品代碼/版別：	產品名稱/應用：	建立日期：
PL7411 / v0.1	4-CH Meter IC / Sample Code Flow	2015/9/1

內容表

核准	覆核	申請人

申請人→相關部門人員(覆核)→部門主管(核准)

Revision History (only for internal version control)

Revision	Description	Date	Author
0.1	➢ Initial Version	2015/9/1	Yaoming Wu
	➢		

※※※※ 技資中心注意！！！ ※※※※

此頁為內部版本管控用，轉成 PDF 檔對外流通時，

請務必將此頁移除再轉檔！！！

【技資文件】資料夾 權限設定

姓名	部門代號	READ	WRITE	備註

文件編碼： AN-71150901

PL7411

4-CH Meter IC

Sample Code Flow

Document number：AN-71150901

Revision：0.1

Release Date：September 01, 2015

Prolific Technology Inc.
7F, No. 48, Sec. 3, Nan Kang Rd.
Nan Kang, Taipei 115, Taiwan, R.O.C.
Telephone: +886-2-2654-6363
Fax: +886-2-2654-6161
E-mail: sales@prolific.com.tw
Website: http://www.prolific.com.tw

Table of Contents

List of Figures

1. SPI interface

1.1 SPI Init Flow

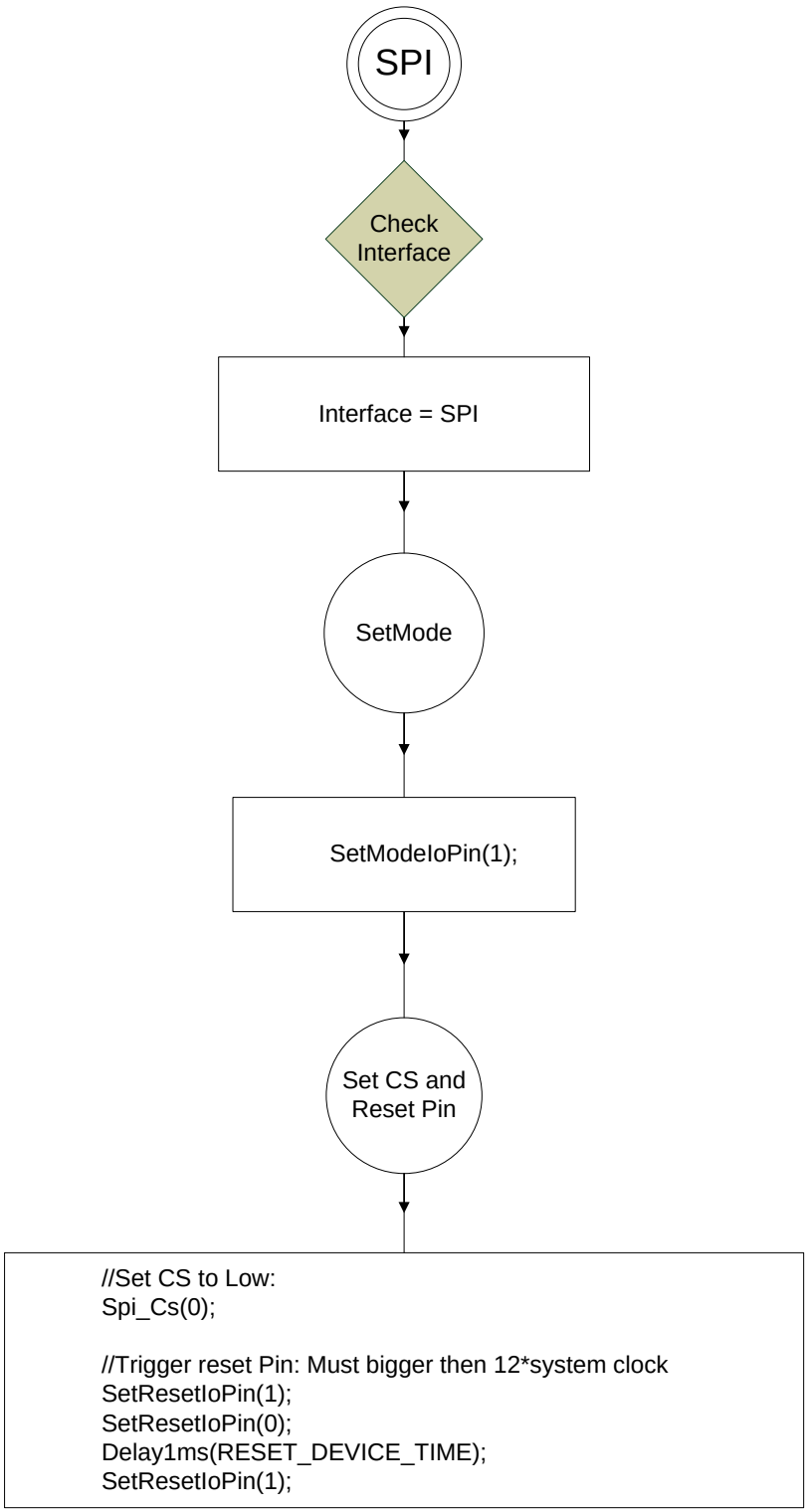
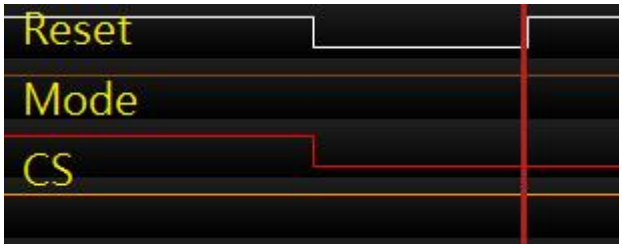


Figure 1.1：SPI Init Flow

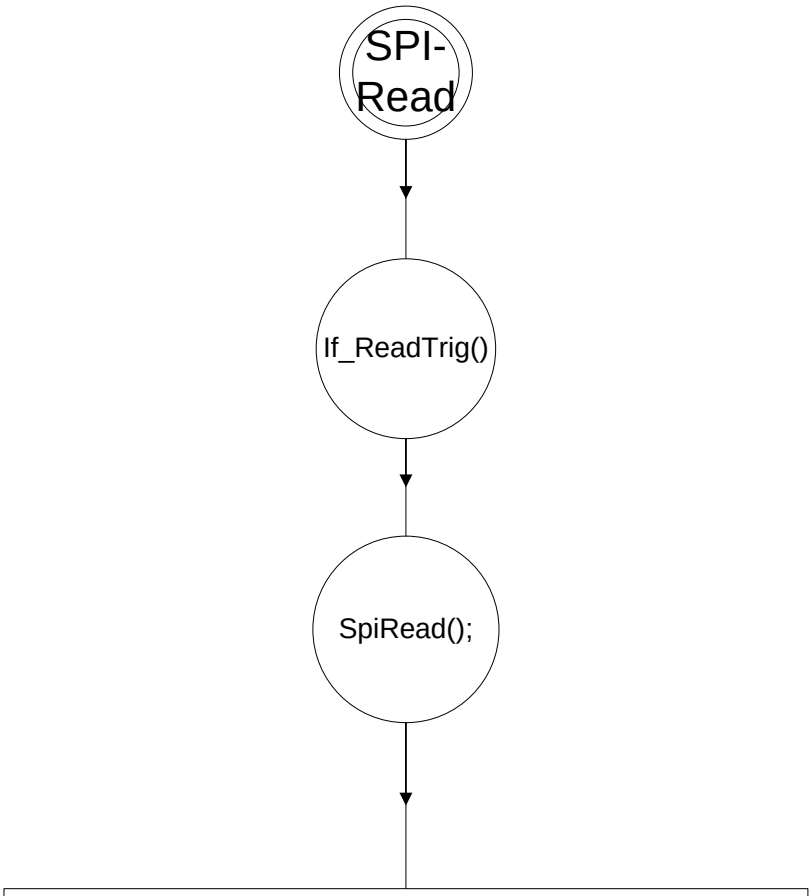
1.2 SPI Reset Waveform

IO Mode is latch when resetn is from low to high, the period Must bigger then 12*system clock , After Reset , need wait 10ms let HW move OTP to RAM, Then can set SPI command

1.2.1 spi_en = [mode, spi_cs]= 2'b10



1.3 SPI Read Flow




```
//If length >15byte
if (len > 15){
        SpiSetLen(len);          //Set 0x3805
}else{
        R_Cmd| = len;            //Read Cmd 0x30
}

Read_addr = adr;                 //Set Read  addr

Spi_Cs(0);

/* Read cmd(1B) + address(2B) */
for (cnt=0; cnt<3; cnt++){
        Spi_Trig(TxBuf[cnt]);
}

for (cnt=0; cnt<len; cnt++){
        buf[cnt] = Spi_Trig(0x00);
}

Spi_Cs(1);
```

Figure 1.3：SPI Read Flow

1.4 SPI Command

spi_cmd[7]: inc_adr_dis 1: disable address increment

Ex: checking whether dsp is ready. The address can be fixed.

dsp_status(cfg addr = 0x390a[7]).

If DSPRDY is from 0 to 1, then master can read out dsp out buffer data.

spi_cmd[6:4]: read/write command

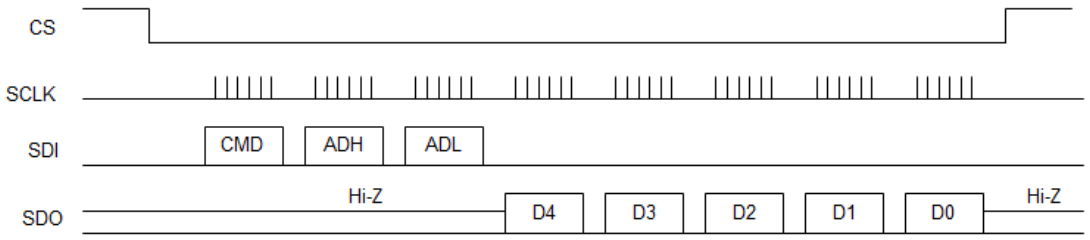
- spi_cmd[6:4] = 3'h1 for read with crc enable
- spi_cmd[6:4] = 3'h2 for write with crc enable
- spi_cmd[6:4] = 3'h3 for read with crc disable
- spi_cmd[6:4] = 3'h4 for write with crc disable

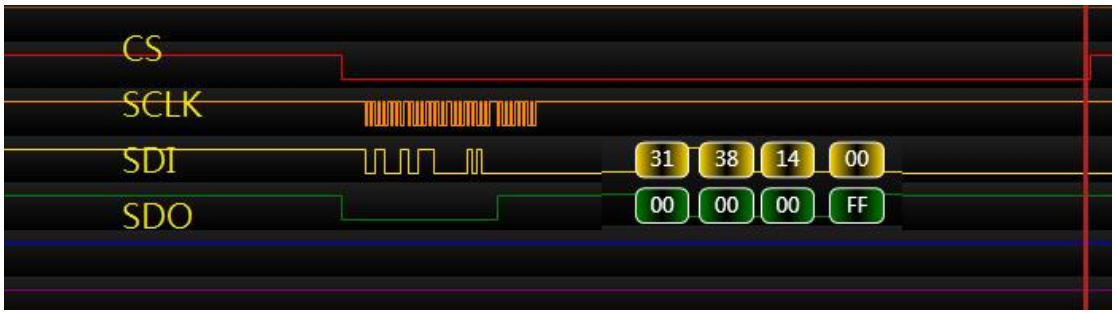
spi_cmd[3:0]: Read/Write package number

- case1: crc enable, set to 4, master must send out 4byte data + 1byte crc. Then slave will send out internal crc value to master.
- case2: crc disable, set to 4, master must send out 4byte data. Then slave will not send out internal crc value to master.

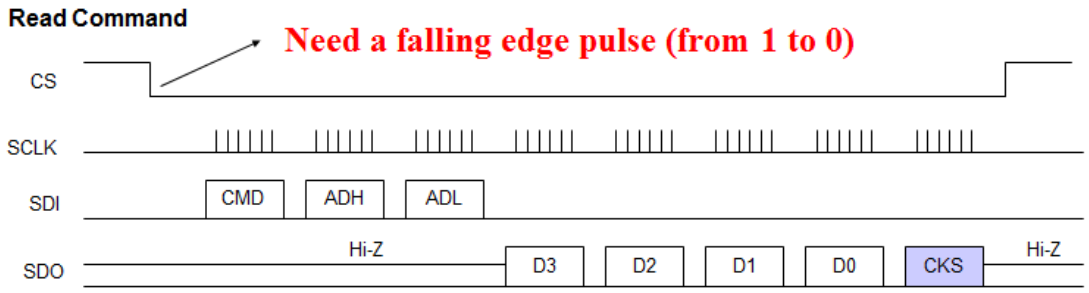
1.5 SPI Read Waveform

Read Command





1.6 SPI Read CRC Waveform



1.7 SPI Write Flow

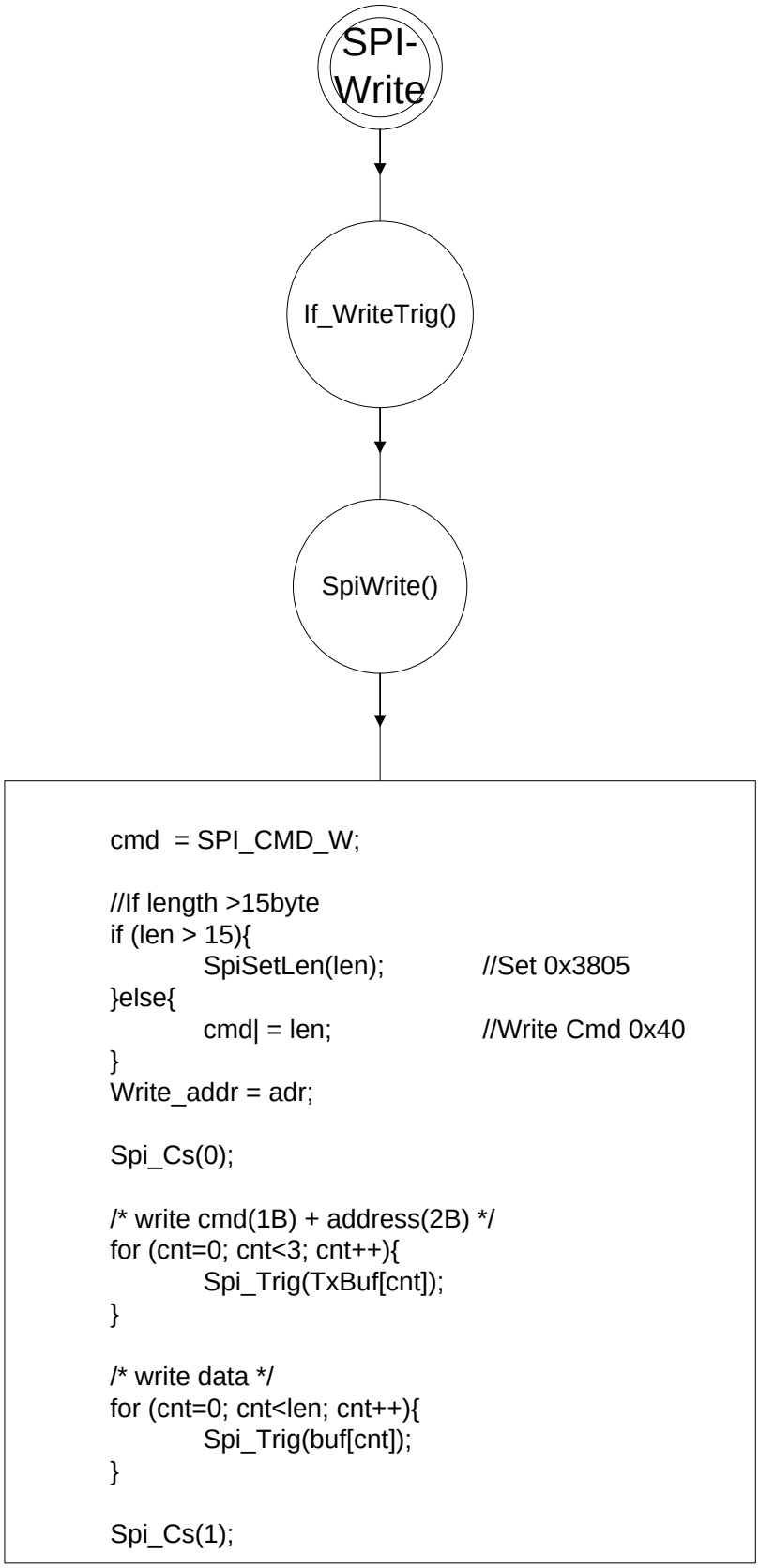
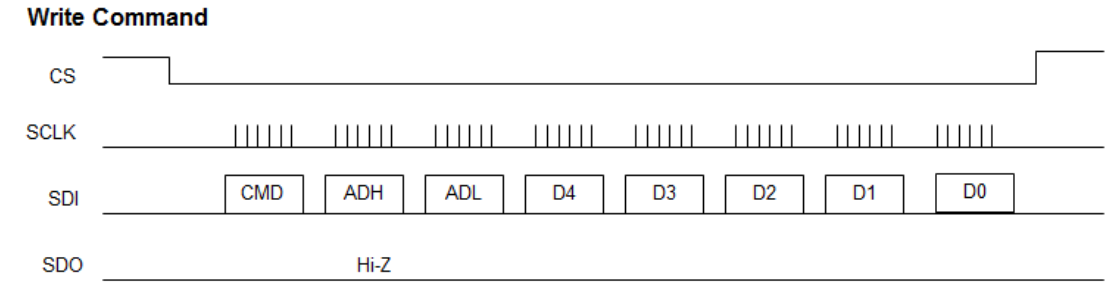
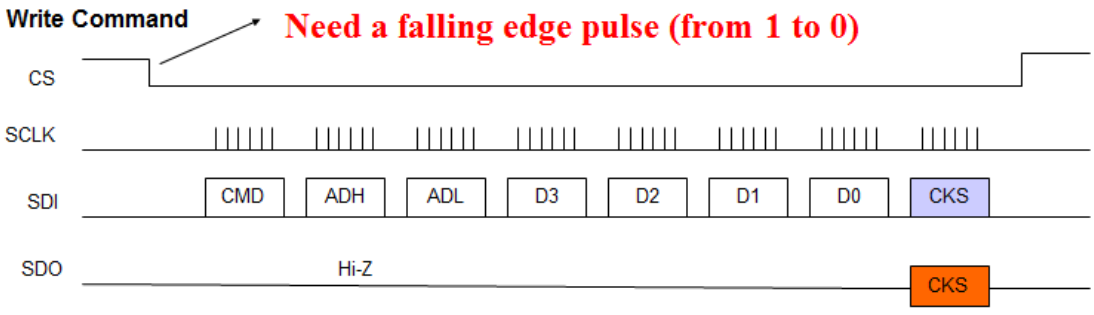


Figure 1.7：SPI Write Flow

1.8 SPI Write Waveform



1.9 SPI Write CRC Waveform



2. I²C Interface

2.1 I²C Init Flow

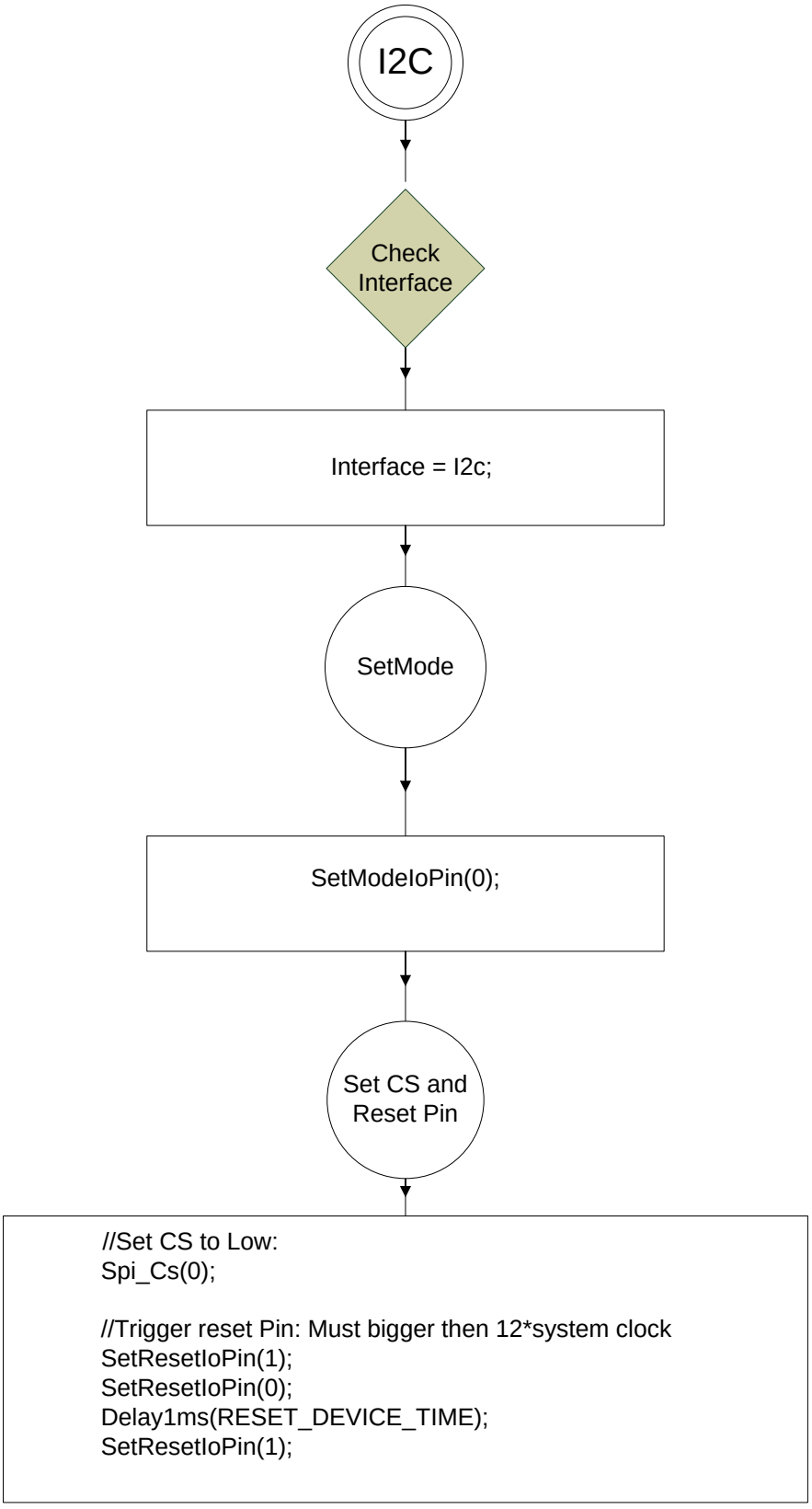
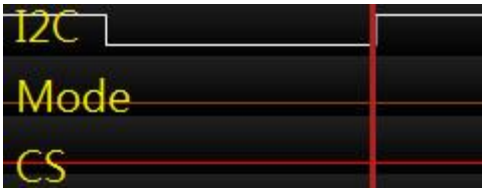


Figure 2.1：I²C Init Flow

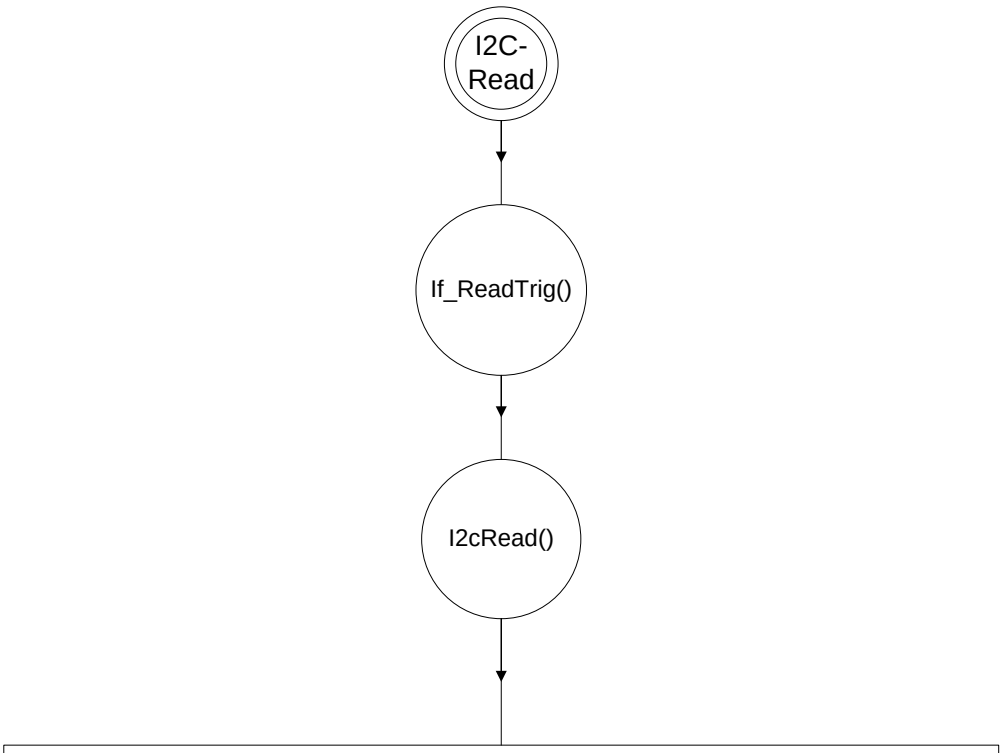
2.2 I²C Reset Waveform

IO Mode is latch when resetn is from low to high, the period Must bigger then 12*system clock , After Reset , need wait 10ms let HW move OTP to RAM, Then can set I2C command.

2.2.1 i2c_en = [mode, spi_cs]= 2'b00



2.3 I²C Read Flow




```
/* write */
Start();                                    //Start bit
WriteByte(I2C_SLAVE_ID);                    //I2C_SLAVE_ID:0xFE
WriteByte(0x00);                            //CRC disable
WriteByte((addr&0xff00)>>8);
WriteByte((addr&0x00ff));

/* read */
Start();                                    //Start bit
WriteByte(I2C_SLAVE_ID | BIT0);             //I2C_SLAVE_ID|BIT0:0xff

for (cnt=0; cnt<len; cnt++){
        if (cnt == (len-1)){
                v = ReadByte(1);  //1 -> NAK
        }else{
                v = ReadByte(0);  //0 -> ACK
        }
        buf[cnt] = v;
}

Stop();
```

Figure 2.3：I²C Read Flow

2.4 I²C Command

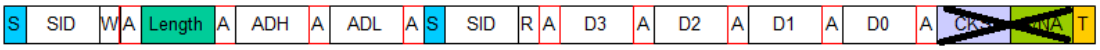
- Slave id: default is 0x7F (2-bit from I/O pin: SID1, SID0)
  - Iocfg: 0x380d default value: 0xFE,
    so first time I2C is set to default function, then master can reset iocfg to disable default function.
  - iocfg[1]: i2c_ref_cs, i2c default is not to reference spi_cs to start i2c.
    set to 1, reference slave id to enable i2c
    spi_cs = 0 is to start so initial master send spi_cs to 0 and mode to 0 to select i2c mode, wait sometime, change spi_cs to 0 to enable i2c r/w.
  - iocfg[0] : i2c_lea_cs, i2c default is not reference spi_cs to leave state.
    set to 1 : use stop to leave state
    spi_cs is from 1'b0 to 1'b1 to leave i2c state machine. if i2c_lea_cs is disable, then i2c only reference stop to leave state.
- In write command:

  byte 1 is slave id + R/W (write is 0, read is 1)
  byte 2 is crc calculation length. When zero, crc is disable;
  when 0xFF, inc_adr_dis is 1 : to disable address increment.
  byte3 is address high : address[15:8]
  byte4 is address low : address[7:0]

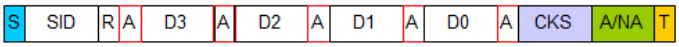
2.5 I²C Read Waveform

Read Command

Sequential Random address read



Sequential current address read



S : Start ; T : Stop

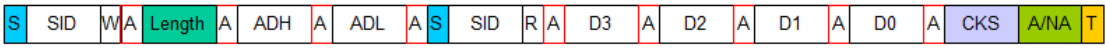
A : ACK

NA : No ACK

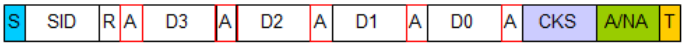
2.6 I²C Read CRC Waveform

Read Command

Sequential Random address read



Sequential current address read

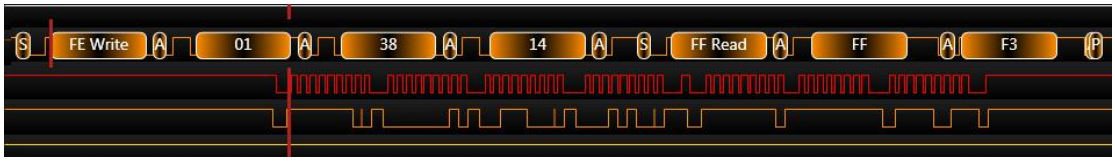


S : Start ; T : Stop

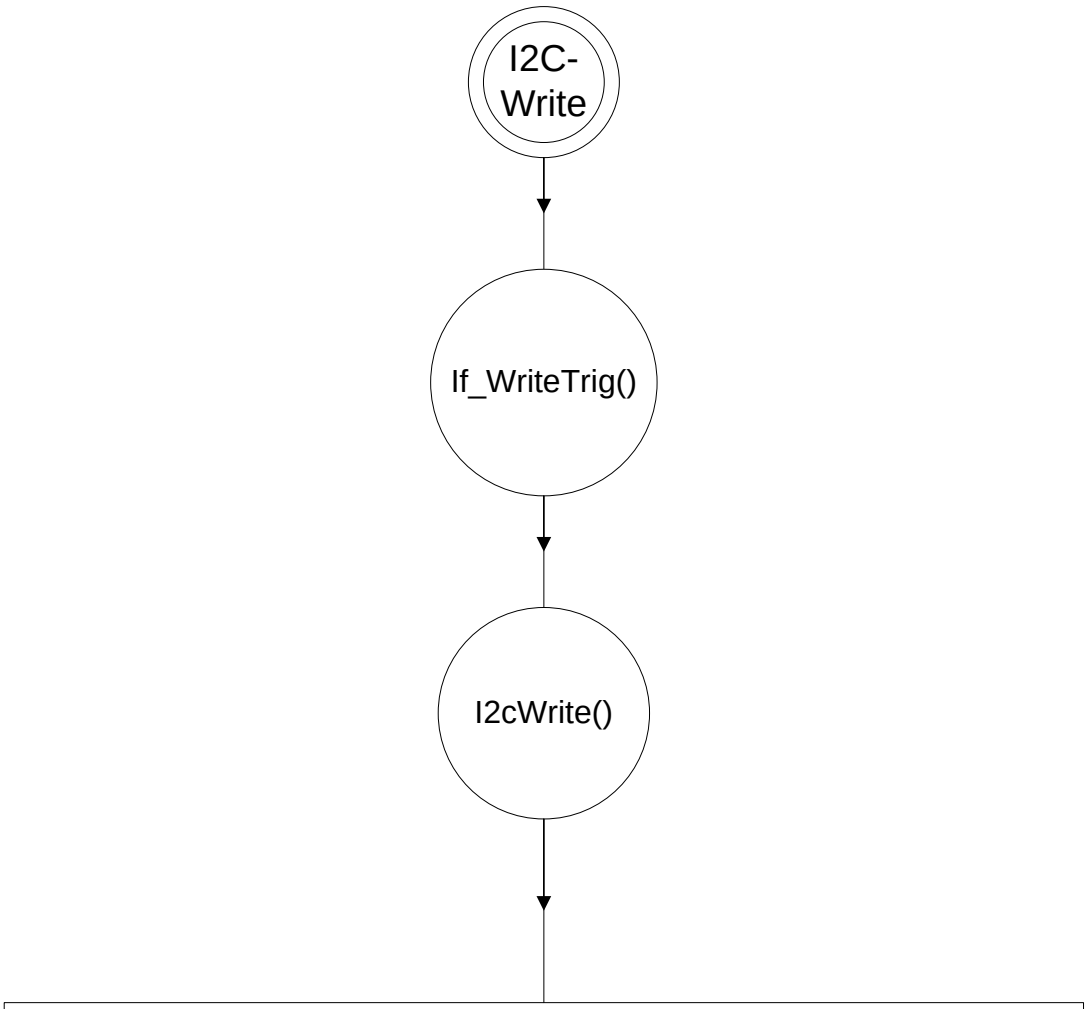
A : ACK

NA : No ACK

Sequential Random address read:



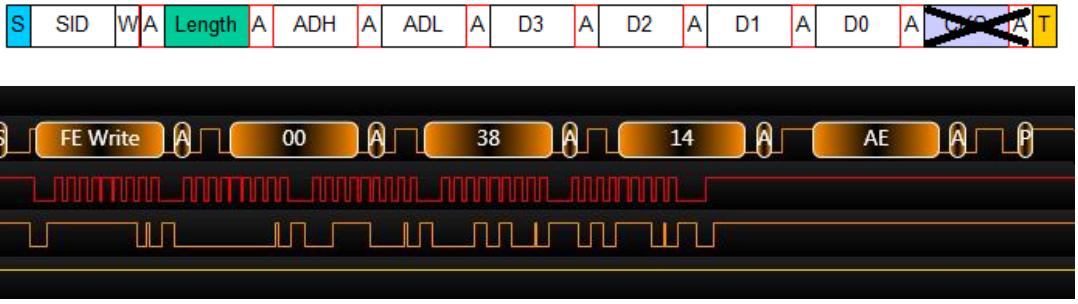
2.7 I²C Write Flow




```
/* write */
Start();                                    //Start bit
WriteByte(I2C_SLAVE_ID);                    //I2C_SLAVE_ID:0xFE

WriteByte(0x00);                            //CRC disable
WriteByte((addr&0xff00)>>8);
WriteByte((addr&0x00ff));

for (cnt=0; cnt<len; cnt++){
        WriteByte(buf[cnt]);
}

Stop();
```

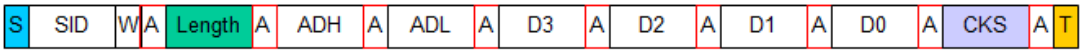
Figure 2.7：I²C Write Flow

2.8 I²C Write Waveform



2.9 I²C Write CRC Waveform

Write Command



3. UART interface

3.1 UART Init Flow

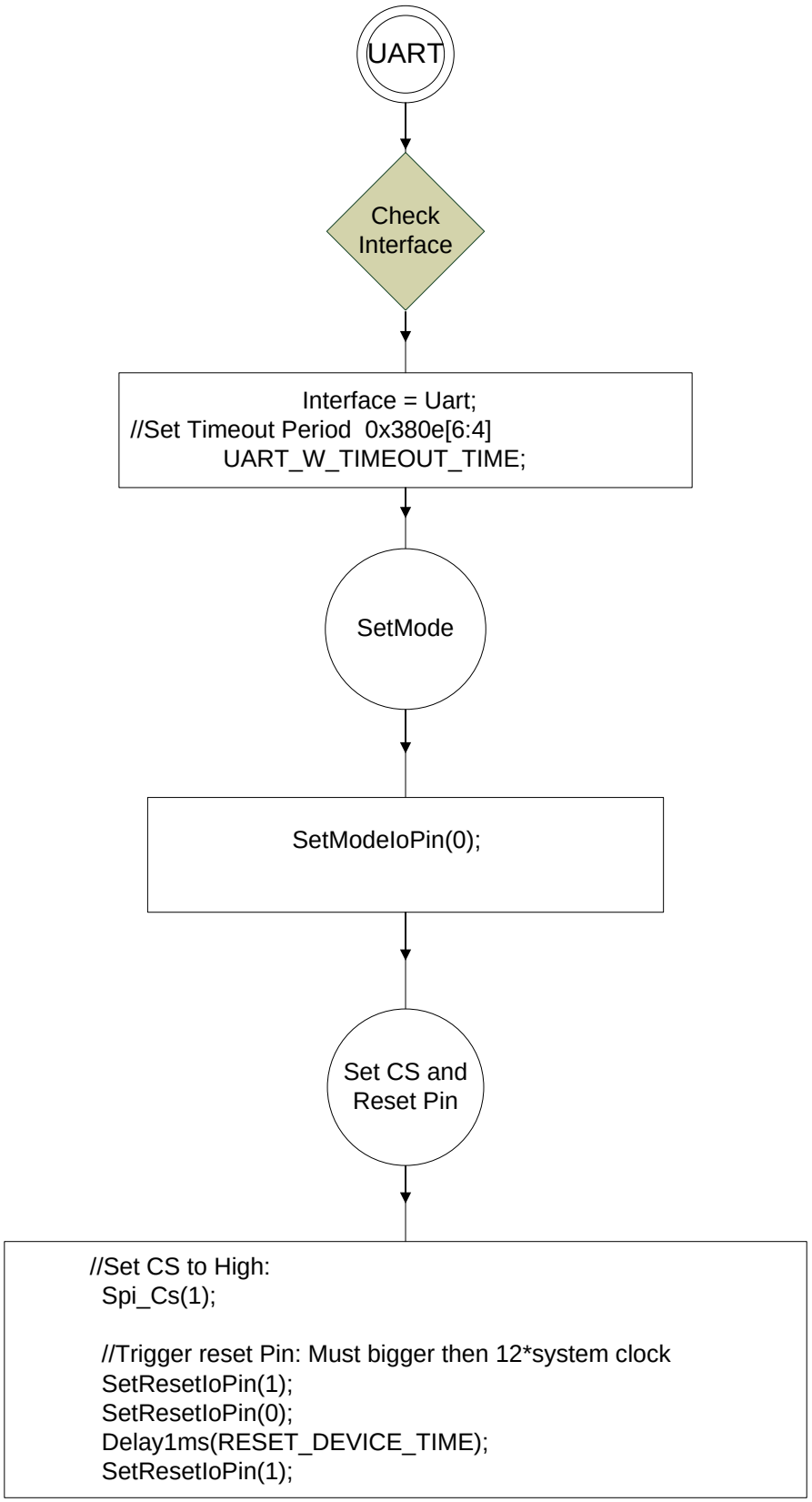


Figure 3.1：UART Init Flow

3.2 UART Reset Waveform

IO Mode is latch when resetn is from low to high, the period Must bigger then 12*system clock , After Reset , need wait 10ms let HW move OTP to RAM, Then can set UART command.

3.2.1 uart_en = [mode, spi_cs]= 2'b01

3.3 UART Config

- iocfg : 0x380d default value: 0xFF,
  so first time UART is set to default function, then master can reset iocfg to disable default function.
- iocfg[7]: uart_ref_cs ,uart default is not to reference spi_cs to start uart.
  set to 1, using slave_id to enable uart function
  (spi_cs = 0 is to start. so initial master send spi_cs to 1 and mode to 0 to select uart mode, wait sometime, change spi_cs to 0 to enable uart r/w)
- iocfg[6]: uart_lea_cs, uart default is not to reference spi_cs to leave state.
  set to 1, using timeout to leave state.
  (spi_cs is from 1'b0 to 1'b1 to leave uart state machine. When read, master can use NACK or spi_cs to leave uart state)
- iocfg[5]: uart_bau_en uart defalut is to enable baud rate detection.

3.4 UART Slave ID

- slaveid[7:0]: 0x380F
  - slaveid[7]: default is high.
    if this bit is set to high, the general call address (00h) is recognized, otherwise it is ignored.
  - uart_slave_id[7:0] = {slaveid[5:0],slaveio};
  - slave io = {**SID1(PAD_P4)** pin, **SID0(PAD_P11)** pin}, 2bit
  -

3.5 UART Set Timeout Waveform

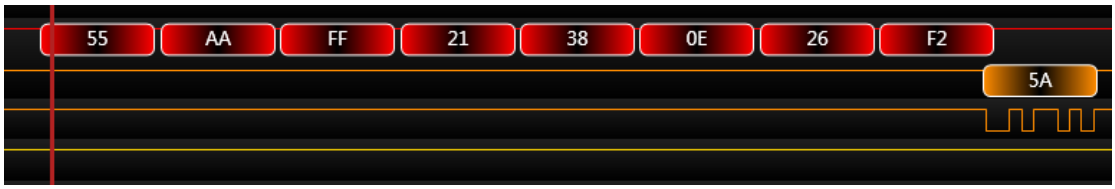
- Uart_to_cfg[6:4] : To select time out period
  - 3'd7 : 2^21 * sys_clk (default setting)
  - 3'd6 : 2^20 * sys_clk
  - 3'd5 : 2^19 * sys_clk
  - 3'd4 : 2^18 * sys_clk
  - 3'd3 : 2^17 * sys_clk
  - 3'd2 : 2^16 * sys_clk
  - 3'd1 : 2^15 * sys_clk
  - 3'd0 : 2^14 * sys_clk

  - If default sysclk is 16MHz, then time out is equal to 62.5ns * 2^21 = 131ms.
  - If default sysclk is 16MHz, then time out is equal to 62.5ns * 2^15 = 2ms.
  - If default sysclk is 4MHz, then time out is equal to 250ns * 2^21 = 524ms.

If read/write with CRC, it's timeout must > the timeout without CRC.

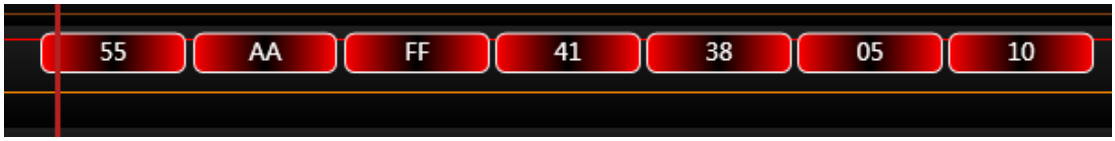
Without CRC: 0x380E=0x13

With CRC: 0x380E=0x26



3.6 UART Read >= 16 bytes Waveform

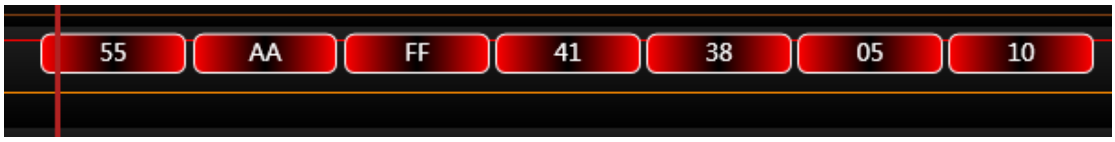
(1)Set 0x3805=0x10 for reading 16 bytes



(2)Read 16bytes

3.7 UART Write >= 16 bytes Waveform

(1) Set 0x3805=0x10 for reading 16 bytes



(2) Write 16bytes+wait timeout

3.8 UART Read Flow

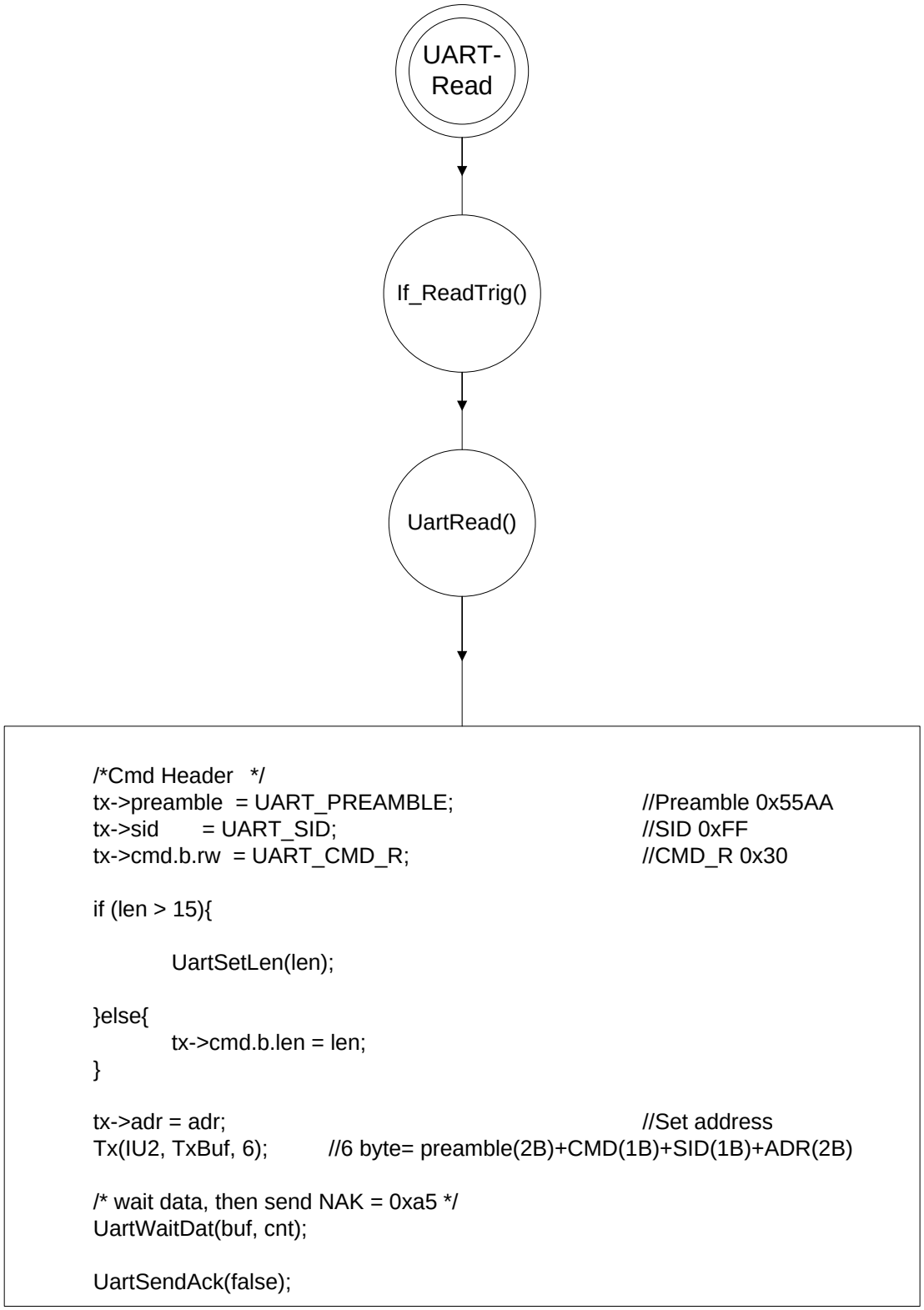


Figure 3.3：UART Read Flow

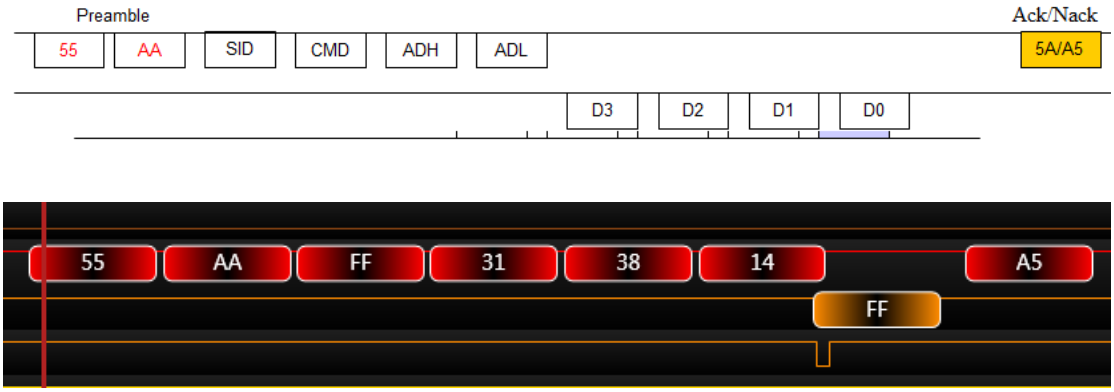
3.9 UART Command

- uart_cmd[7]: inc_adr_dis,
  1: to disable address increment.
  ex: reading dsp_status [DSPRDY, STOP, TRIGCNT] (cfg addr = 16'h390a[7]).
  If DSPRDY is from 0 to 1, then master can read out dsp out buffer data.
- uart_cmd[6:4]: Read/write command
  uart_cmd[6:4] = 3'h1 for read with crc enable
  uart_cmd[6:4] = 3'h2 for write with crc enable
  uart_cmd[6:4] = 3'h3 for read with crc disable
  uart_cmd[6:4] = 3'h4 for write with crc disable
- uart_cmd[3:0]: Read/Write package number
  case1: crc enable, set to 4, master must send out 4byte data + 1byte crc. Then slave will send out 0x5A(ack)/0xA5(nack).
  case2: crc disable, set to 4, master must send out 4byte data. Then slave will not send out 0x5A(ack)/0xA5(nack).
- When length = 0, then hardware (IC) will set the internal length equal to d'96 so FW can read the data of dsp outbuf by 4 times (96*4 = 384 bytes) with crc.
- When crc is disable, the length setting is needed, the state machine is left when time_out.
- uart protocol : start bit + 8bit data + stop bit, no parity bit (10bit)
- 0x55 1byte for baud rate detection
- 0xAA, slave check if baud rate is the same as master.
  - If the same, slave send out 0x5A (Ack), no send out.
  - If not the same, slave send out 0xA5 (NAck), no send out.
- When write command, if slave crc is the same as master crc, then slave send out 0x5A(Ack), not the same, slave send out 0xA5(Nack).
- In write command :
  - If in the time out period, the master sends out another data series, the write state machine will go on writing action. (Address auto increment)
- If time out is got, then the write state machine will go to idle state. Then another write/read command can work.
- otp write time :
  - parameter Tpw_min = 90000 ; //Min Program Pulse Width Time
  - parameter Tpw_max= 110000; //Max Program Pulse Width Time
- In read command:
  - When the read package is end, the master can send out 0x5A for continuing to read another data package (address auto increment); or 0xA5 for stopping the read action, then the read state machine will go to idle state; or wait time out to idle state.

- If spi_cs is to be referenced, then spi_cs from 0 to 1, then the read state machine will go to idle state.

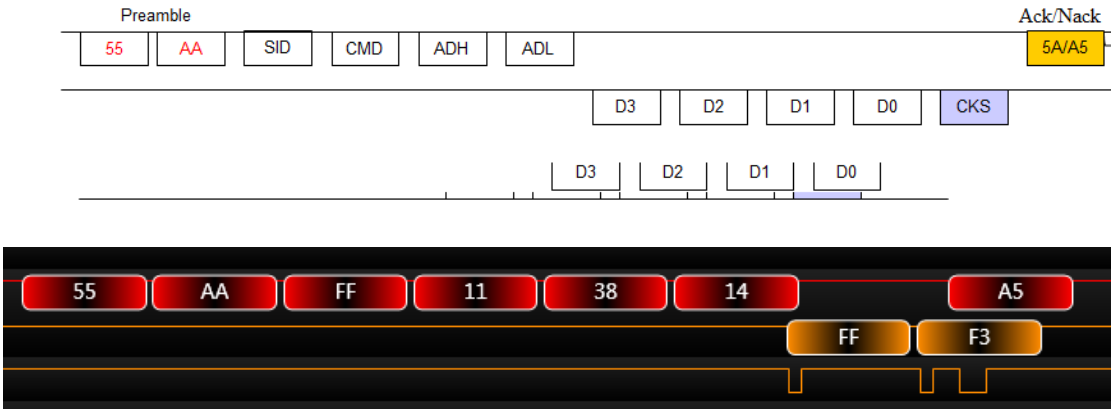
3.10 UART Read Waveform

Read Command



3.11 UART Read CRC Waveform

Read Command



3.12 UART Write Flow

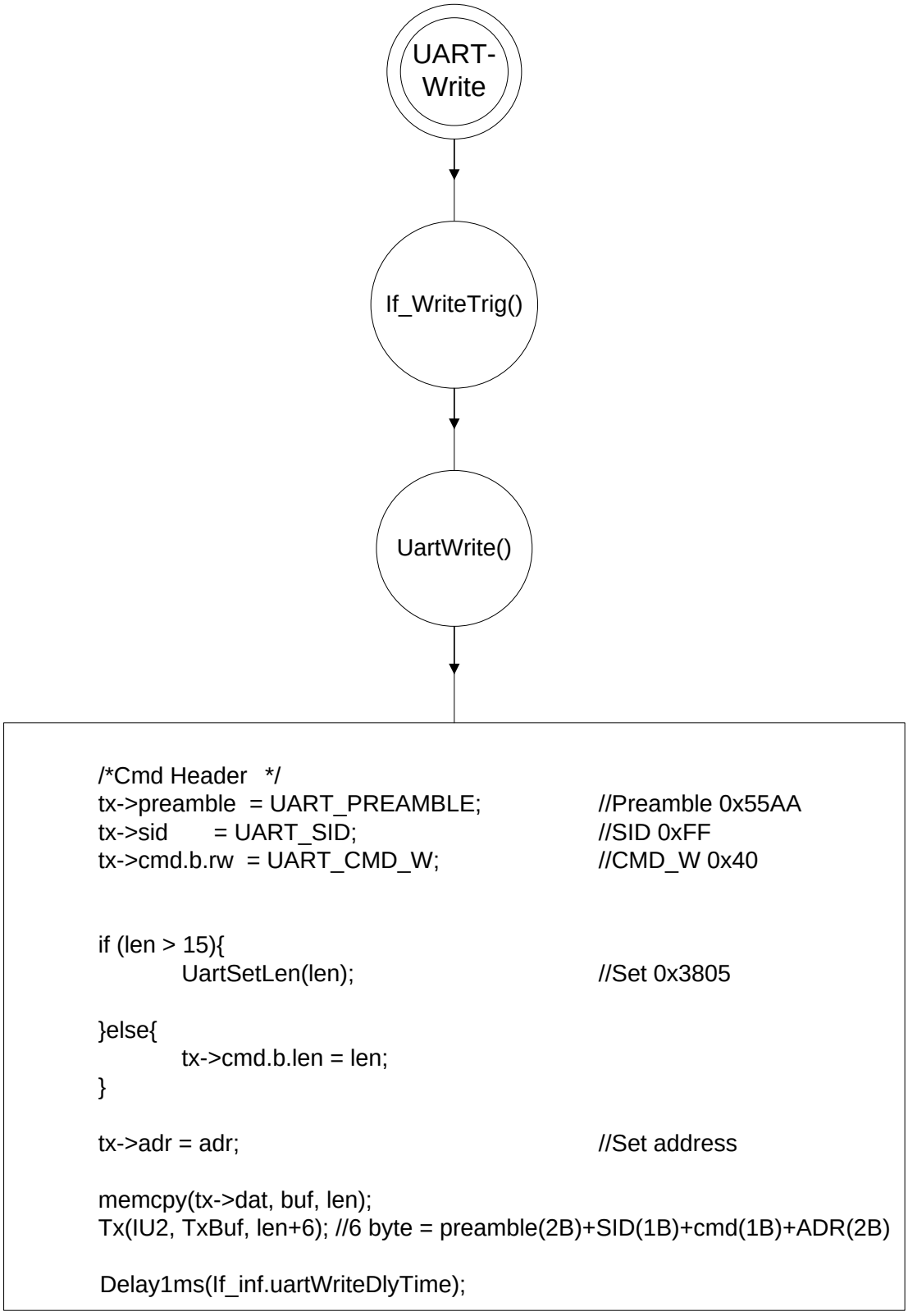
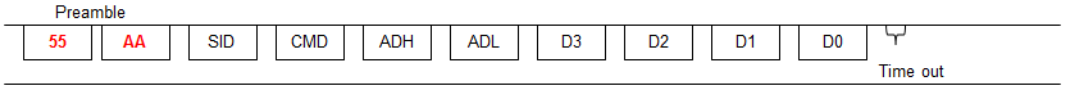


Figure 3.10：UART Write Flow

3.13 UART Write Waveform

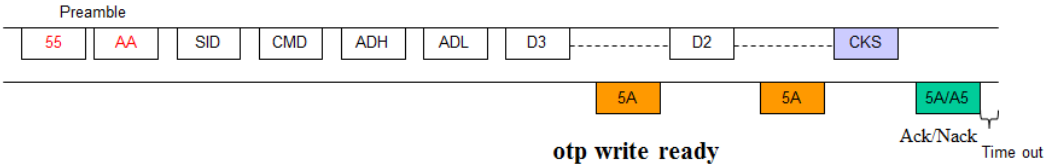
Write Command



Write + wait timeout:

3.14 UART Write CRC Waveform

Write Command



Write+CRC+ wait timeout(not write OTP):

Disclaimer

All the information in this document is subject to change without prior notice. Prolific Technology Inc. does not make any representations or any warranties (implied or otherwise) regarding the accuracy and completeness of this document and shall in no event be liable for any loss of profit or any other commercial damage, including but not limited to special, incidental, consequential, or other damages.

Trademarks

The Prolific logo is a registered trademark of Prolific Technology Inc. All brand names and product names used in this document are trademarks or registered trademarks of their respective holders.

Copyrights

Prolific Technology Inc.

7F, No. 48, Sec. 3, Nan Kang Rd.

Nan Kang, Taipei 115, Taiwan, R.O.C.

Telephone: +886-2-2654-6363

Fax: +886-2-2654-6161

E-mail: sales@prolific.com.tw

Website: http://www.prolific.com.tw