

Figure 7. Firmware update process for pulse counter

**I2C Communications Spec between pulse counter and Colibri iMX6ULL board**

**Model 1**

The Model 1 will use a custom Packet Format to exchange data between the STM and the Colibri board.

All slaves has hardware address **0x51**.

Request from master:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Start0 | Start 1 | instruction | size | Data0 | Data0 | …. |  | Summ0 | Summ1 |

Where:

**Start0** – 0xAA

**Start1** – 0x55

**Instructions:**

A single byte defines up to 256 instructions between the recipients. Each instruction will have a specific set of data that appropriately sent with the Packet.

1. Get version Application (GET\_VER\_APL)
2. Send part of application (command will send after start update) (SEND\_PART\_APP)
3. Start update Application – send size of application in **BLOCKS (128 Bytes) (**START\_UPDATE**)**
4. Run Application (RUN\_APL)
5. Set leds (SET\_LEDS)
   1. **Payload:** LED\_INDICATOR(8bits) MODE(8bits) PARAM1(16bits) PARAM2(16bits) PARAM3(16bits) (total 8 bytes)
6. Do reset microprocessor (DO\_RESET)
7. Get counter1 (GET\_COUNTER1)
8. Get counter2 (GET\_COUNTER2)
9. Set counter1 (SET\_COUNTER1)
10. Set counter2 (SET\_COUNTER2)

**Size** – A single byte used to define how large the Data portion of a packet could get, in bytes. This value can range from zero to 251.

**Data** - This is the only variable sized portion of the packet. The size is of the Data portion of a packet is determined by the Packet Size value. At the minimum, there can be no Data bytes in the packet when the Packet Size is zero. At the maximum, there can be 251 Data bytes in the packet

When the Packet Size is 251. In case request, ALL data set to FF.

**Summ0 Summ1** - Packet Checksum

This portion of the Packet consists of two bytes. The checksum calculated by adding all of the Packet bytes together. It is not possible to roll over the value of 254 bytes of data in a 2-byte number.

**Response from slave:**

**A** – ACK

**0** – NACK

**For each request should be send response.**

Example of request version:

Master: AA(w) 55(w) 01(w) 00(w) 01(w) 00(w)

Slave: 00(r) 00(r) 00(r) 02(r)

Example of set leds:

Master: AA550508090201F401F400000301 /\* flash SYSTEM\_HEALTH\_LED\_RED, 500 - duration ON, 500 - duration OFF, 00 - total duration \*/

Slave: A(r) (ack)

Example of start update:

Master: AA(w) 55(w) 02(w) 02(w) 01(w) 00(w) s0(w) s1(w) /\* send start update with size of application in blocks (100h Blocks) \*/

Where: s0, s1 – check summ

01, 02…F8 – part of data

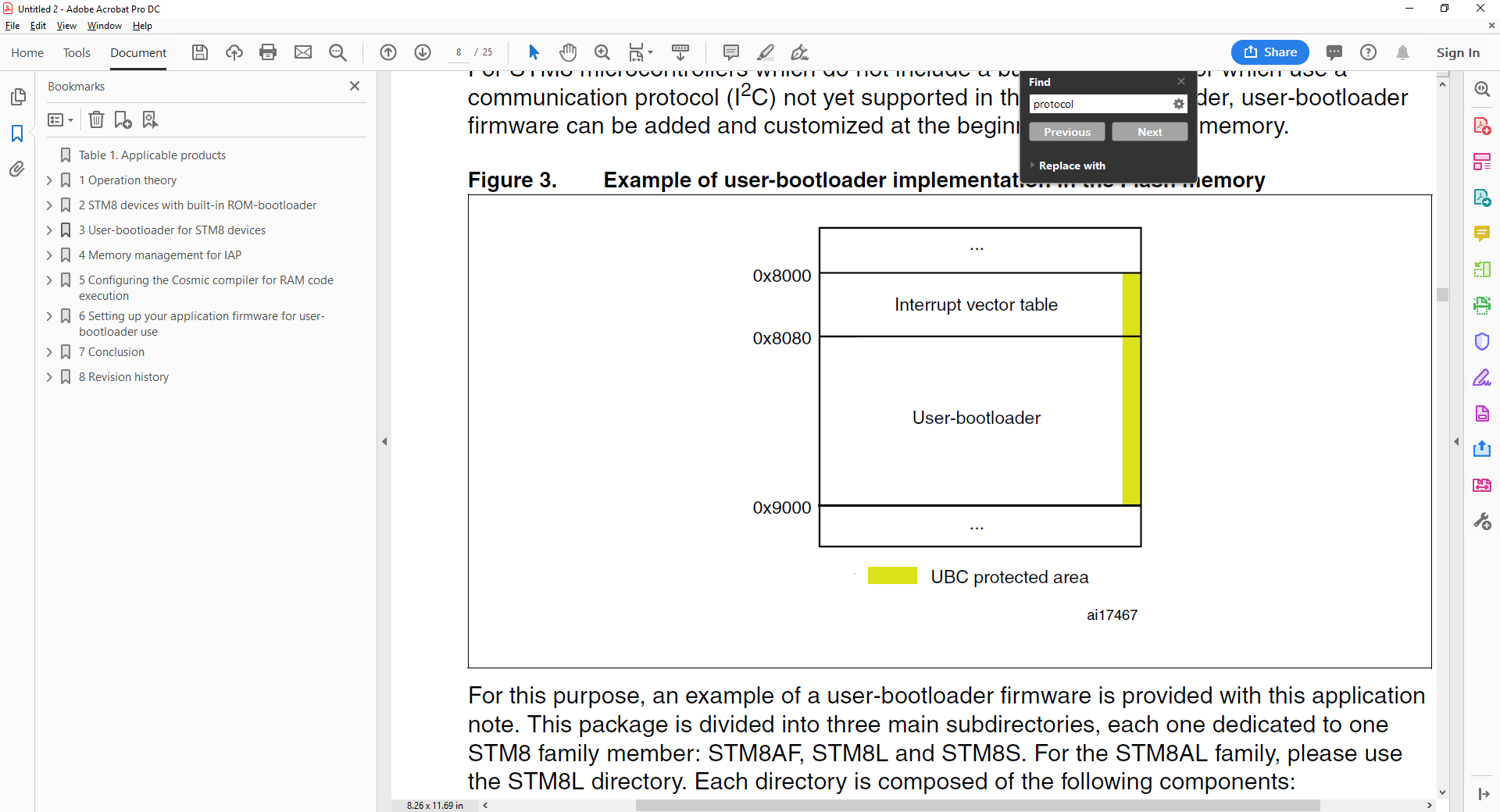
Reading until the slave answers: A(r) (ack) or NACK

Example of get counter1:

Master: AA5507000106

Slave: 0x00 – count of overload

0x0000 - counter



Application