International Headquarters
Myllyojankatu 2 A
FIN-24100 SALO
FINLAND
tel +358 (0)2 727 7700
fax +358 (0)2 727 7720
www.nordicid.com

NORDICID

Nordic ID RFID

NUR packet contents and CRC-16 explained

Version 1

NORDICID

1 NUR protocol packet contents

The NUR protocol packet is divided into three parts:

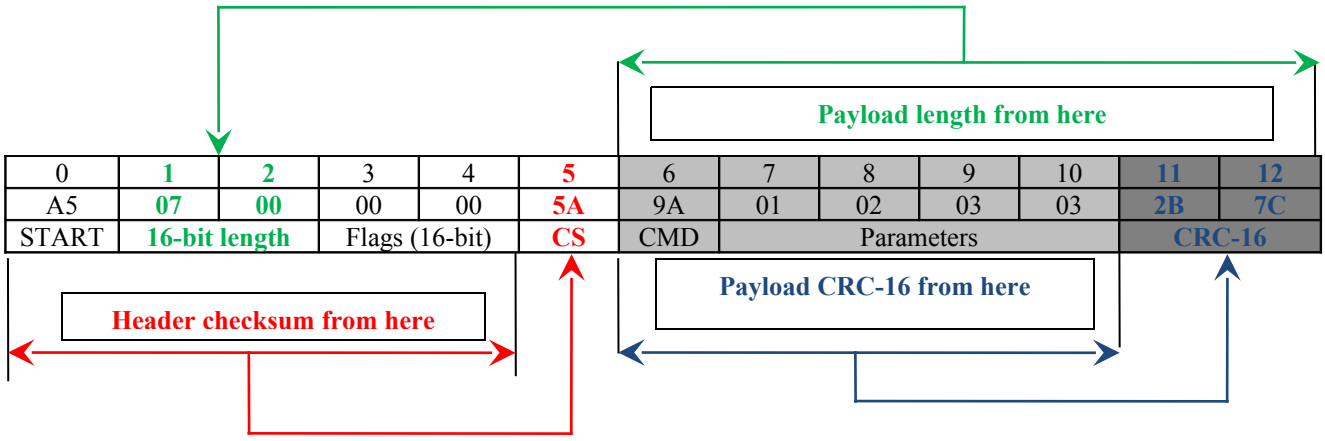
1. Packet header
2. Command payload: command byte + parameters related to the command
3. Packet CRC (CRC-16, CCITT)

The packet header has its own checksum. The CRC is calculated over the command byte and its parameters excluding the packet header. The payload length field in the packet header includes the command byte, its parameters and the CRC-16 (2 bytes).

2 Packet content diagram

The diagram below illustrates a NUR protocol packet including parameters added to a command. The upmost row contains byte indices in a TX byte buffer, the middle row contains hexadecimal values for this example and the row on the bottom explains the contents. The command itself is imaginary; it is generated just for this example. See NUR getting started application notes for actual commands.

NOTE: the multi-byte values are in little-endian format.



3 Packet contents

The example packet contents:

Bytes	Contents	Is
0…5	`A5070000005D (6 bytes)`	Packet header where A5 = start byte 07 00 = 0x0007 payload length (including the CRC-16 bytes) 00 00 = packet flags 5D = header checksum that is CS = `FF(+)A5(+)07(+)00(+)00(+)00 = 5D` (+) = exclusive OR
6…10	`9A01020304`	The command and its parameters. Command value in this example is `0x9A` and the imaginary parameter bytes are `0x01, 0x02, 0x03 and 0x04.`
11…12	`2B7C`	The command and it parameters' CRC-16: **`0x7C2B`**.

4 Calculating the CRC

The below C-code calculates the example. In the accompanying file, NurCRC16.c, there are a few examples on the CRC-16 calculation including the building of this example packet.

```
static unsigned short crc16Table[256];
static int init = 0;

static void InitCRC16()
{
  int i, j;
  /* Make sure integer is 32-bit. */
  unsigned int c;

  for (i = 0; i < 256; i++) {
    c = i << 8;
    for ( j = 0; j < 8; j++ ) {
      c = (c & 0x8000) ? MSG_CCITT_CRC_POLY ^ (c << 1) : (c << 1);
    }
    crc16Table[i] = (unsigned short)c;
  }
}

unsigned short NurCRC16(unsigned short crc,
              unsigned char *buf, unsigned int len)
{
  if (!init) {
    InitCRC16();
    init = 1;
  }

  while (len--) {
    crc = (crc << 8) ^ crc16Table[(crc >> 8) ^ *buf++];
  }

  return crc;
}
```

When the example packet is cast to a byte buffer and the above CRC calculation is done from the command byte onwards, the CRC-16 will be correct.