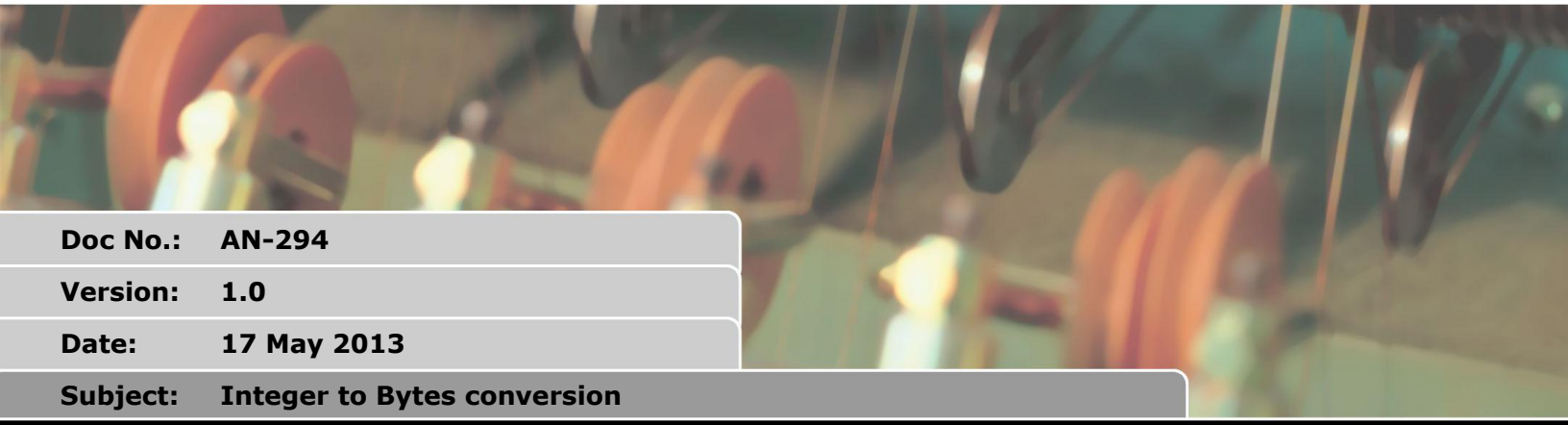
Shannon Way,
Tewkesbury,
Gloucestershire. GL20 8ND
United Kingdom
Tel: +44 (0)1684 292 333
Fax: +44 (0)1684 297 929

187 Northpointe Blvd,
Suite 105
Freeport, PA 16229
United States of America
Tel: +1 724-540-5018
Fax: +1 724-540-5098

Tomson Centre
118 Zhang Yang Rd., B1701
Pudong New Area, Shanghai,
Postal code: 200122
CHINA
Tel/Fax: +86 21 587 97659

SCMC House
16/6 Vishal Nagar
Pimpale Nilakh, Wakad, Pune
PIN 411027
INDIA
Tel: +91 206 811 4902



Doc No.: AN-294

Version: 1.0

Date: 17 May 2013

Subject: Integer to Bytes conversion

APPLICATION NOTE

1. Overview

When handling communication channels like RS232, CAN or Telnet on Ethernet, it is sometimes necessary to send or receive a long integer as 4 bytes.

This short application note describes a software routine that converts a 32 bit integer held in a VR to 4 bytes for transmission. The second part shows the reverse operation, converting 4 bytes back to a long integer.

2. Limitations and differences

The MC2 and MC3 range can handle only 24 bit integers.

The MC3 range has a SHIFTR command, which the MC2 range does not have.

The MC4 range can handle 32 bit integers.

The MC4 range has a shift right >> and a shift left << function.

Older versions of the MC2 range do not support the Hexadecimal notation for entering or displaying variable values.

3. Long to 4 bytes

This example is for the MC302X or MC302-k. It converts the VR to 4 bytes, assuming that the value is between -8388608 and 8388608. Ie a 24 bit signed integer. The routine used the SHIFTR command that is found in the MC302 range firmware.

```
VR(20)=-8388607

byte0 = VR(20) AND $ff
byte1 = SHIFTR(VR(20), 8) AND $ff
byte2 = SHIFTR(VR(20), 16) AND $ff
byte3 = SHIFTR(VR(20), 24) AND $ff

PRINT HEX(VR(20))
PRINT HEX(byte3),HEX(byte2),HEX(byte1),HEX(byte0)
```

The bytes could be transmitted to a serial port like this:

```
PRINT #1,CHR(byte3);CHR(byte2);CHR(byte1);CHR(byte0);
```

They could be included in a CAN telegram:

```
' send a 4 byte telegram using identifier $020
CAN(-1,5,10,$020,4,1)
CAN(-1,7,10,byte3,byte2,byte1,byte0)
```

Here is the same routine tailored for the MC2 range, which does not have the SHIFTR command. The divide operator is used instead. In fact this runs as fast as the SHIFTR in the MC302 because the DSP used in the MC2 range has a fast divide function in the processor.

```
VR(20)= -8388607

byte0 = VR(20) AND $ff
byte1 = (VR(20) AND $00ff00)/256
byte2 = (VR(20) AND $ff0000) /65536
IF SGN(VR(20))=-1 THEN
  byte3 = $FF
ELSE
  byte3 = 0
ENDIF
```

4. 4 bytes to long

Converting the 4 bytes back to a long, the routine needs to consider the sign of the value. If the most significant bit of byte3 is set to 1, then it is a negative number. If the original was a 24 bit integer, then the msb of byte 2 will be the indicator of the sign.

This example is for the MC302X / MC302-K.

```
VR(21)=(byte3*16777216) OR (byte2*65536) OR (byte1*256) OR byte0

IF READ_BIT(23,21)=1 THEN
  VR(21)=(VR(21) AND $FFFFFF) -16777216
ENDIF

PRINT VR(21)[0],HEX(VR(21))
```

This is the function changed a little to enable it to work on the MC2 range.

```
VR(21)=(byte2*65536) OR (byte1*256) OR byte0

IF READ_BIT(23,21)=1 THEN
  VR(21)=(VR(21) AND $FFFFFF) -16777216
ENDIF

PRINT VR(21)[0],HEX(VR(21))
```

Notice that byte 3 is not included in the conversion. This is because when the DSP multiplies by 16777216, there is an error in the resulting floating point value and this distorts the final value in VR(21). As only 24 bits are useful to the MC2xx, this is an acceptable compromise.

5. Conversion routines for the MC4 range

The MC4 range has shift right and shift left operators. It can also handle the full 32 bit signed number. Therefore the routines for conversion are very straight-forward.

Convert 32 bit long to 4 bytes:

```
VR(20)= -8388607

byte0 = VR(20) AND $ff
byte1 = VR(20)>>8 AND $ff
byte2 = VR(20)>>16 AND $ff
byte3 = VR(20)>>24 AND $FF

PRINT ""
PRINT HEX(VR(20))
PRINT HEX(byte3),HEX(byte2),HEX(byte1),HEX(byte0)
```

Convert 4 bytes to a signed 32 bit long integer:

```
VR(21)=(byte3<<24) OR (byte2<<16) OR (byte1<<8) OR byte0

IF VR(21).31=1 THEN
  VR(21)=(VR(21) AND $FFFFFFFF) -2^32
ENDIF

PRINT VR(21)[0],HEX(VR(21))
```