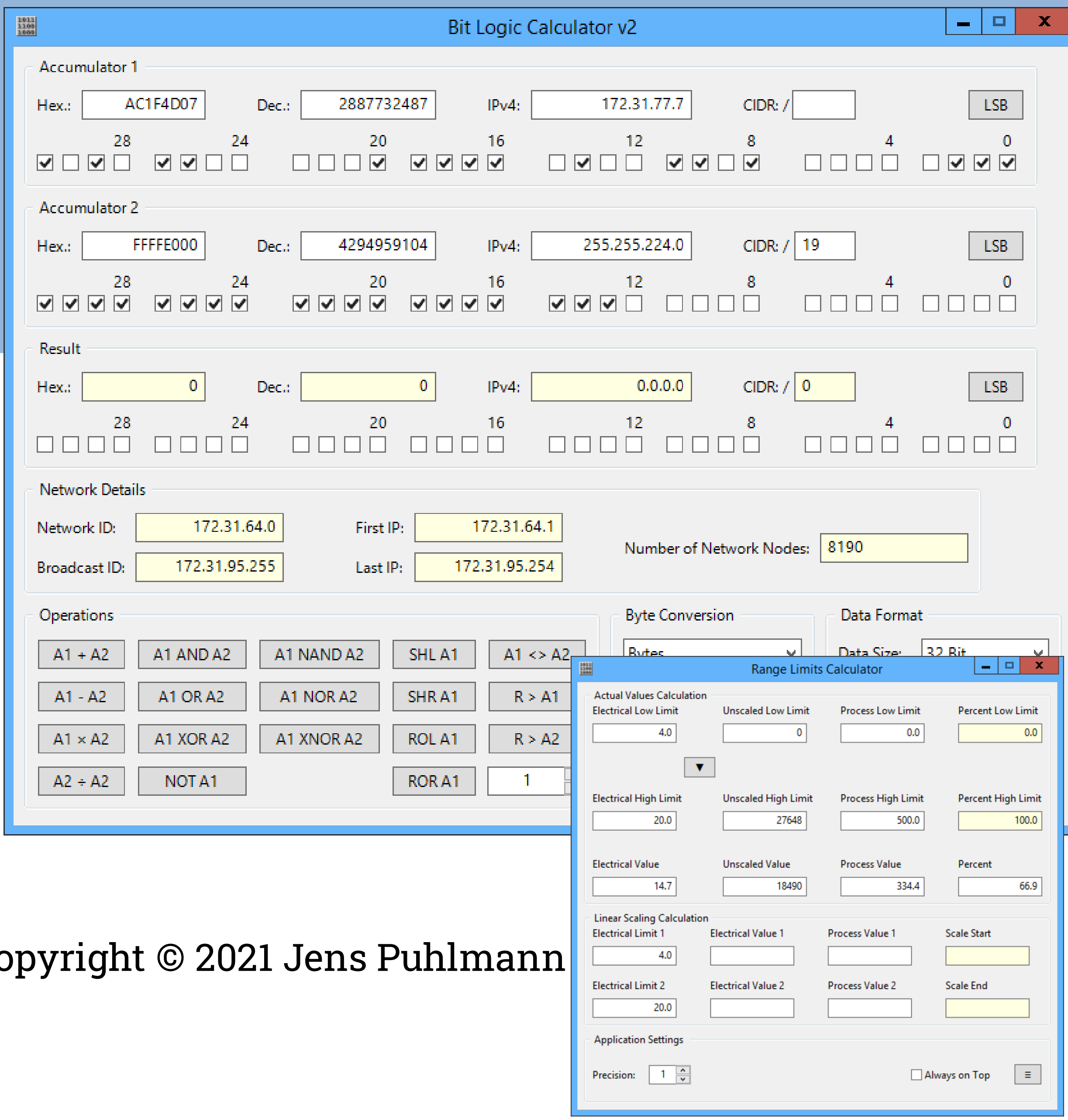
Bit Logic Calculator

USER MANUAL



1 Introduction

Bit Logic Calculator (BitCalc) is a programmer's calculator, specifically optimized for programming tasks necessary in an industrial work environment (plant automation).

- It supports the user with data format conversion, value conversion and network addressing tasks:
- Data conversion between hexadecimal, decimal and binary format (8 bit, 16 bit or 32 bit)
- Display format using switchable between signed or unsigned integer
- Data conversion between IPv4 Network address format, binary (32 bit) format or CIDR (Classless Inter Domain Routing format for subnet masks)
- Calculation on of network ranges based on a subnet mask and IP address
- Two accumulators for basic mathematical or logical operations:
  - Addition, subtraction, multiplication or division
  - Logical AND, OR, XOR, ...
  - Bit Shifting (SHL, SHR, ROL, ROR)
- Value conversion between any of the following data formats:
  - IT Units (Base = 1024): KiB, MiB, GiB or TiB
  - SI Units (Base = 1000): kB, MB, GB, TB

A special feature of BitCalc is the **on-the-fly** conversion of any values. Depending on the user input, all associated conversion values change fully automatic without further user input.

What's New in v2 (compared to v1.6):

- Automatic adjustment of of windows elements to the selected font size of the operating system; the previous selection for "Large Font" is removed
- Windows themes are supported
- Significantly enhanced Network Address Calculation features, including CIDR input and network details calculation
- Value conversion now supports floating point format as input, permitting conversion of fractional values (e.g. "5.75 MB")
- An actual user manual in PDF format
- Windows NT is no longer supported

Useful application scenarios for BitCalc:

- It is very useful as teaching tool for the different data formats
- Evaluation of status words / command words in industrial automation projects
- Evaluation of network addressing schemes in IPv4 format
- Teaching tool to explain the concept of Addressing / Subnetting in IPv4 networks

1.1 System Requirements

Software Requirements:

- Microsoft Windows 2000
- Microsoft Windows XP / Windows Server 2003
- Windows Vista (untested)
- Microsoft Windows 7
- Microsoft Windows 8 / Windows Server 2012
- Microsoft Windows 10

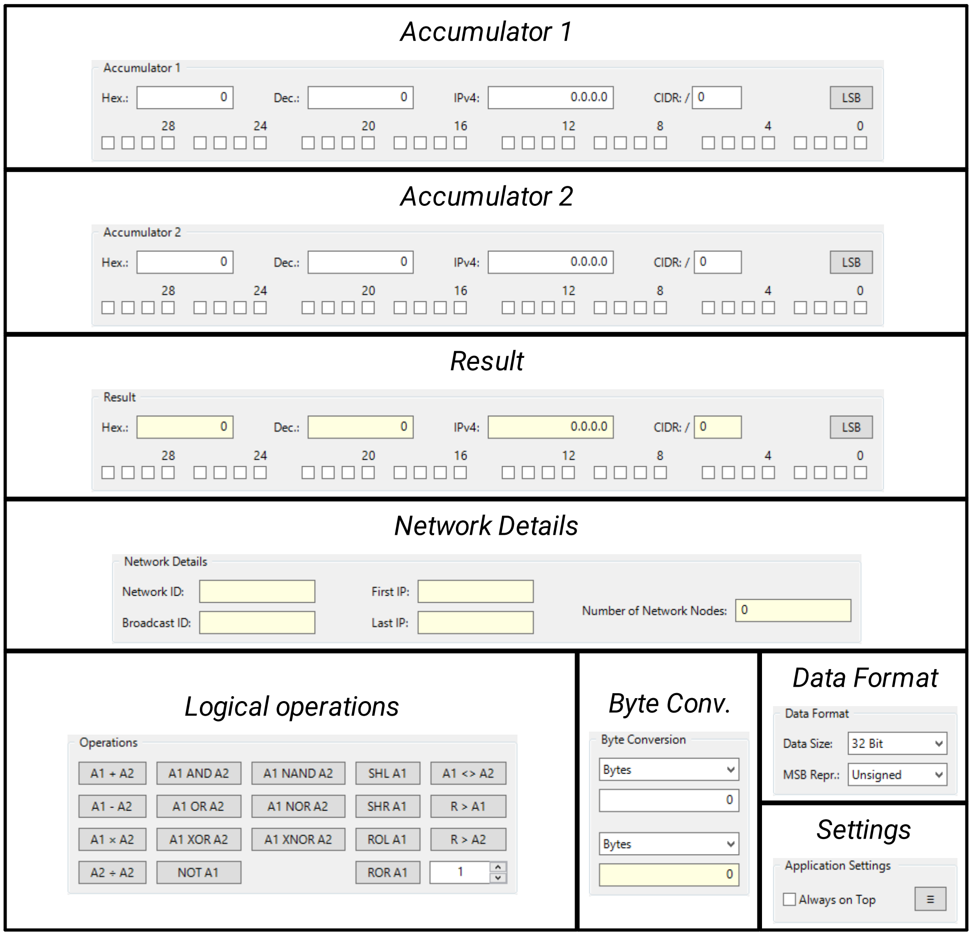
Hardware Requirements:

- A minimum screen resolution of 1024x768 pixels

A 64-bit version of the application is also available, but there is no functional difference between both applications.
Technically, the main advantage of the 64-bit version is the fact that it runs native on Windows, while the 32-bit version runs on an emulation layer (WoW64 [Windows 32-bit on Windows 64-bit]). Since Bit Logic Calculator is not very demanding regarding processor performance or memory, choosing one over the other makes no real differences. From a programmers point of view, the 64-bit version is more a proof-of-concept, rather than anything else.

2 Basic Functions

BitCalc's application window is separated into 8 distinctive user sections:



Accumulator 1 and Accumulator 2:

Here, the values used for logical operations or network calculations are entered. During data input, the associated alternate input fields will automatically update. For example, if a value is entered in the decimal input field for Accumulator 1, then the hexadecimal and binary displays for Accumulator 1 will automatically change. If the binary inputs are changed, then the decimal and hexadecimal displays will change.

Note: The CIDR display will also update automatically, but only if a valid subnet mask is detected, otherwise, the input field will remain empty.

A button with the description **LSB** (Least Significant Bit) is provided to change the label display format for the binary indications by toggling the start position (right side of the binary display) from a "0" to a "1". This makes it easier to perform value comparisons from systems not using the common "0" based binary indication. This not only impacts the labels above the bit positions, but also the tool tip text displayed when hoovering the mouse over a binary check-box.

Aside from the labels and the tool tip texts, nothing else changes and all other operations still work the same way as before.

Result:

This section will populate, if any logical operation is selected, except for the last three functions (swapping accumulators, copying the result value to either Accumulator 1 or Accumulator 2).

The result section can also be toggled to display the binary value pattern starting with "0" or "1", using the **LSB** button provided.

Network Details:

This section indicates the network details, if Accumulator 1 or Accumulator 2 contain a valid subnet mask. In that case, the opposite accumulator value is treated as IP address and the network range parameters for that network are calculated:

- Network ID
- Broadcast Address
- First usable IP Address
- Last usable IP Address

If neither accumulator contains a valid subnet mask, or if both accumulators contain a valid subnet mask, then this section will remain empty.

Logical Operations

Here, logical operations with the accumulators are performed. Most functions work with both accumulators as input values (AND, OR,...).

However, there are some operations that only require a single input value. In this case, Accumulator 1 is always used.

If any of the bit shifting functions (Shift-Left, Shift-Right, Rotate-Left and Rotate-Right) is used, then the total number of shifting positions can be entered in a value input field.

Data Conversion:

Here, data conversion between the most common byte display formats can be performed. Supported are the most common IT formats (Base = 1024) as well as the newer SI units (Base = 1000). The conversion is possible from any listed unit to any listed unit.

The input value can also be entered in floating point format, so a conversion using fractional numbers (e.g. "5.72 GiB") is possible.

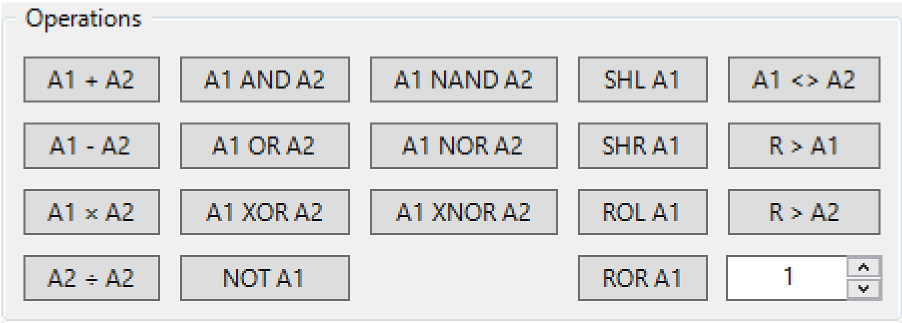
Data Format:

It is possible to switch the data input between the following 3 data formats and two different decimal display variations. **MSB** stands for Most Significant Bit and tells the computer if the highest bit should be treated as "sign" bit to display decimal values signed or unsigned. This also changes the display ranges for the decimal input fields:

Data Size Setting	MSB Representation	
	Unsigned	**Signed**
8 Bit	BYTE (0...255)	SHORT-INT (-128...127)
16 Bit	WORD (0...65535)	SMALL-INT (-32768...32767)

32 Bit	LONG-WORD (0…4294967295)	LONG-INT (-2147483648…2147483647)

2.1 Logical Operations



The most common logical operations have can be performed using the two accumulators. For simplicity reasons, accumulator swapping and result copying functions have been grouped into the Logical Operations section of the program window as well, even though they are technically not logical operations.

Most of the logical operations require two values and accumulators 1 and 2 have to be entered. In case where only a single value is necessary for the operation, accumulator 1 will be used as input value.

For bit shifting and bit rotation functions, the number of shifting / rotation positions can be entered into the input field / spin button next to the "ROR A1" button.

- SHL A1 (Shift-Left Accumulator 1) - all shifted bits will be "dropped" and filled with "0" to the right
- SHR A1 (Shift-Right Accumulator 1) - all shifted bits will be dropped and filled with "0" to the left
- ROL A1 (Rotate-Left Accumulator 1) - all bits moved outside on the left will be used to fill the space to the right
- ROR A1 (Rotate-Right Accumulator 1) - all bits moved outside on the right will be used to fill the space to the left

3 Networking

Networking is one of the major improvements of BitCalc v2. Not only are 32-bit values converted to their IP address notation, but CIDR (Classless Inter Domain Routing) notation for subnet masks is now supported as well.

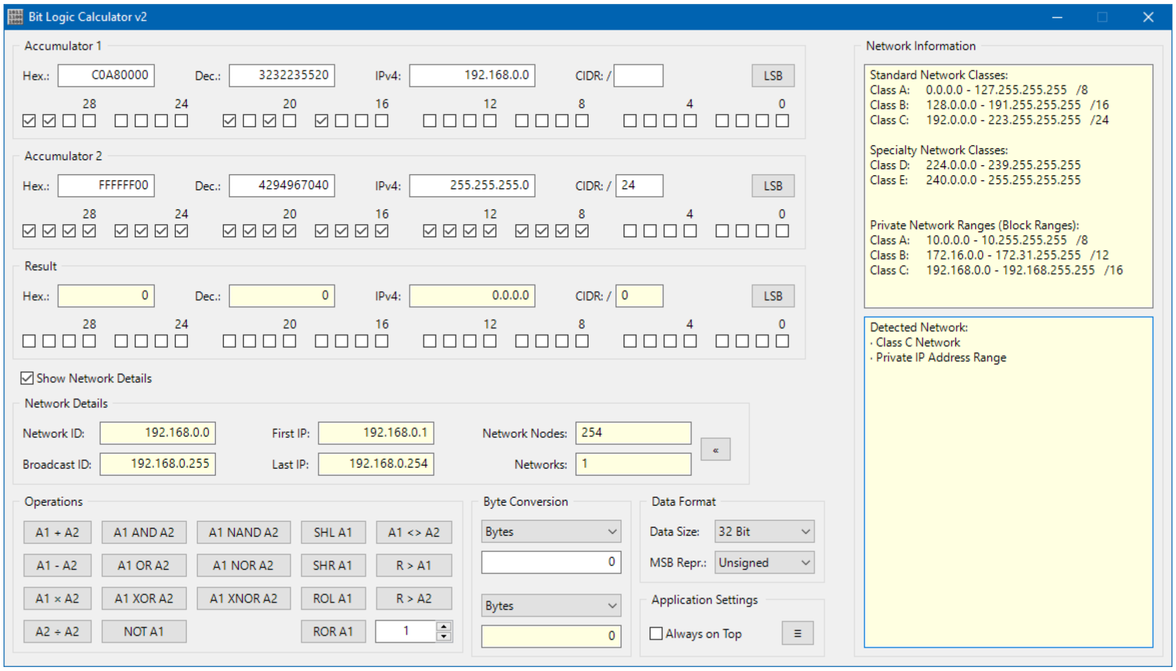
Furthermore, if an IP address and a subnet mask are entered into the accumulators, the associated network parameters (Network ID, Broadcast address, first usable IP address and last usable IP address) are now calculated.

Network calculation functions are only available if the data format is set to 32 bit mode, otherwise, all networking input fields will be unavailable!

During the original design of the manual, several screen shots were taken, showing an approach to use the networking features of BitCalc. Due to the effort involved, the screen captures were not remade after the user interface for BitCalc was changed to include additional networking features.

The following features were added:

- A check box labeled **Show Network Details** - this check box can be used to hide the network details calculation section. This is useful when using BitCalc as training / teaching tool.
- A **Networks** field, where the number of subnetted networks within the network range is calculated. This field depends on the type of network presently entered. For example, a class C network can only be subnetted as CIDR of /24 or higher - attempting to use a CIDR value below /24 will not try to calculate a Networks value, since the range is not valid for this class of network. For a class B network, the minimum CIDR value is /16, while for class A networks, the minimum CIDR value is /8.
- A button to show further network information (to the right of the **Networks** field) - this button shows the standard network classes as well as the reserved private IP address ranges. Please note that the CIDR values displayed for the Private Network Ranges show the reserved Block Sizes, not the valid CIDR values for these ranges! This is also the reason why BitCalc still calculates the overall network range (Network ID, Broadcast ID, First IP, Last IP), even if a CIDR value not applicable for the present network class has been entered.
- In addition, BitCalc attempts to analyze the presently entered network information (IP address and Subnet Mask) and tries to display the detected network details (Network class, Private/Public address range, Invalid range)



3.1 Subnet Masks / CIDR

When using IPv4 for network design, the basic network parameters are determined by IP address and subnet mask. Using these two values, the accessible network range is determined. This calculation does not take into account any class A, B, or C networks; the calculation is performed classless.

Other parameters (e.g. Gateway address or DNS Server address) are not related to IPv4 addressing schemes; so these parameters are not evaluated here.

When any 32-bit values are entered into BitCalc, the software attempts to evaluate if this value matches a valid subnet mask:

- All bits set to "0"
- All bits set to "1"
- All bits to the left are set to "1", while all remaining bits to the right are set to "0"

If a subnet mask is detected, the CIDR value is updated accordingly and the value is treated as a subnet mask. Otherwise, it is assumed that an IP address had been entered.

Both accumulator values can be used as IP address or a subnet mask. However, a network detail calculation can only be performed if exactly one IP address **and** exactly one subnet mask are entered. If both accumulators contain either an IP address or a subnet mask, no network details are calculated.

It does not matter which accumulator contains the IP address or the subnet mask; both accumulators are treated equally.

Notes:

- Bitmask 255.255.255.255 (/32) will not be evaluated for network address details calculation, since this subnet mask does not have room for any number of network nodes. According to Cisco definition, this is not a valid subnet mask and should be avoided when defining network ranges using VLSM. However, there are situations where this subnet mask is used; for example when configuring firewall rules and only a single target address is permissible. So from a network design concept, this subnet mask is indeed possible.
- Bitmask 255.255.255.254 (/31) will only calculate the associated network address and broadcast address, since this subnet mask does not leave any room for network. According to Cisco definition, this is not a valid subnet mask and should be avoided.

3.2 Using the Network Calculations

Due to its unique display style with the binary accumulator displays right underneath each other, BitCalc is a nice application to explain IPv4 network addressing and subnetting.

The following suggestions are based on my personal experience with networking and the approach I take to teach network addressing details at my work environment. By no means am I trying to explain the concepts of IPv4 here in those examples; a good understanding of IPv4 addressing will be necessary to understand these examples.

3.2.1 Example 1 - Understanding CIDR notation

CIDR notation is used to identify the Network ID of any given network address. Simply put, this number represents the total number of bits within a subnet mask set to "1" and is typically easier to use in networking documents, since the notation is much shorter than the notation for a complete subnet mask:

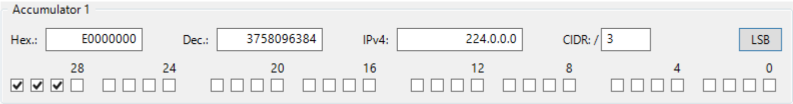
Sample:

```
IP Address / Subnet mask          : 192.168.0.1/255.255.255.0
IP Address / CIDR                 : 192.168.0.1/24
```

Make sure that accumulator 1 is set to "0" in the decimal or hexadecimal display.

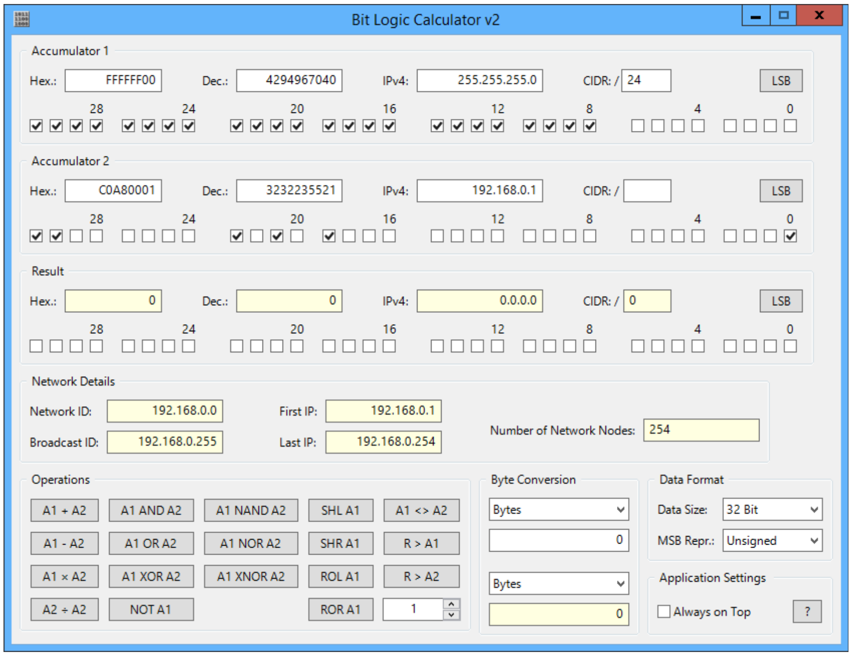
Using the binary display for accumulator 1, set individual bits from the left hand side to "1".

Bit 31 set should result in a CIDR value of "1" (network component = 1 bit) , bits 31 and 30 set should result in a CIDR value of "2", bits 31, 30 and 29 set should result in a CIDR value of "3" and so on.



The subnet mask must be continuous as far as the bit pattern is concerned; there can be only one switch from "1" to "0", no more. All bits set to "1" have to be on the left side of the display. The total count of bits set to "1" determines the CIDR value.

The next example shows the most common CIDR value / associated subnet mask in home networks, together with an typical IP address for that subnet:



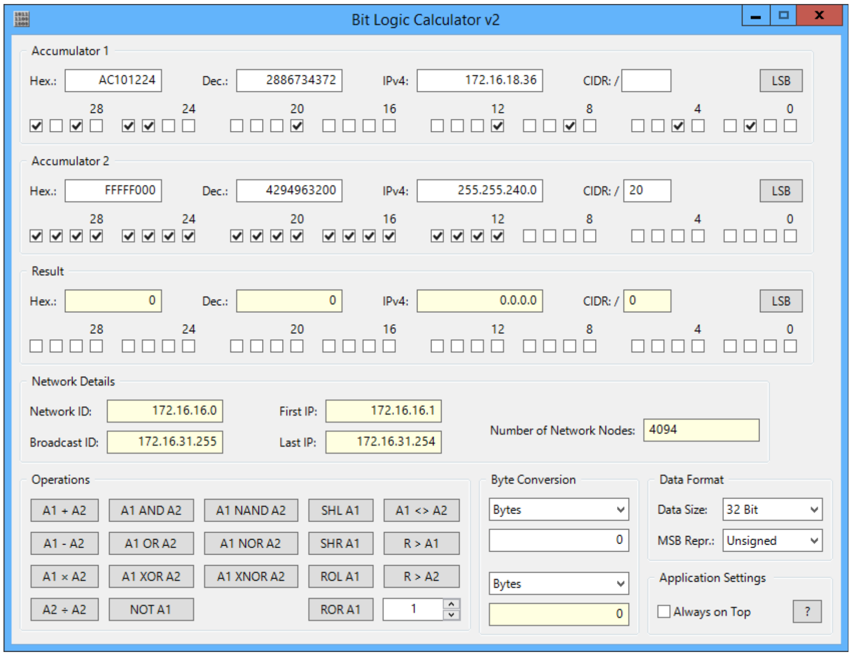
A CIDR value of "24" sets 24 bit positions to "1", starting from the far left, while the remaining bits are set to "0". This results in a subnet mask of 255.255.255.0, which identifies a network with up to 254 network nodes (excluding Network ID and Broadcast ID).

The next two examples attempt to improve the understanding about IPv4 addressing.

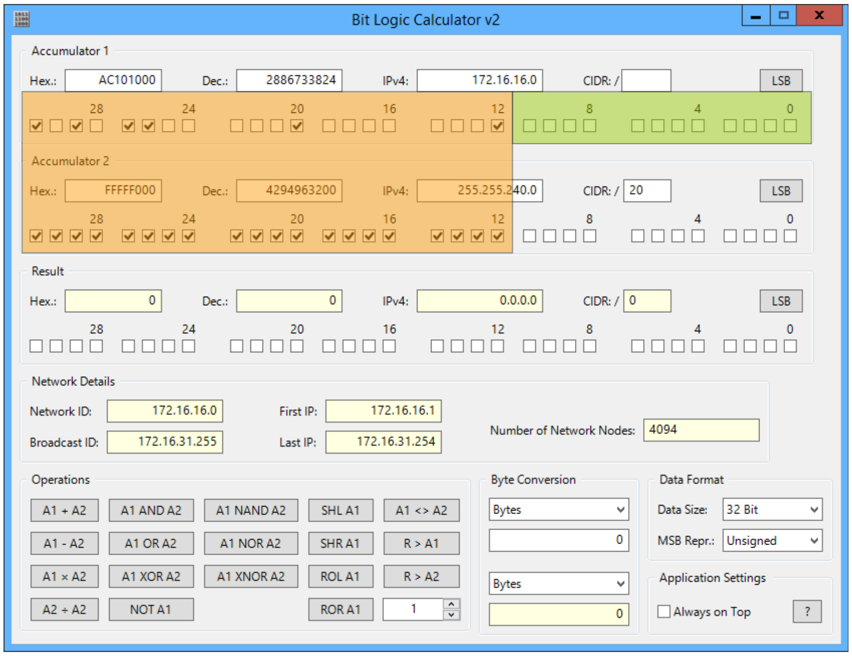
3.2.2 Example 2 - Identify Network Settings manually

For this example, try to ignore the automated network details calculation section in the application window.

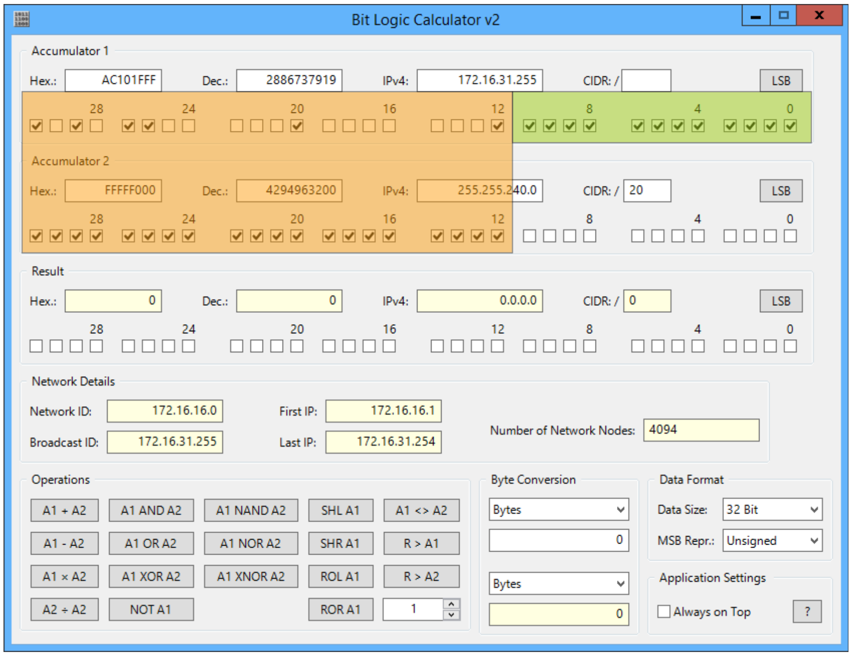
The basic address components should be identified. Enter an IP address of 172.16.18.36 into accumulator 1 and a subnet mask of 255.255.240.0 (CIDR = 20) into accumulator 2:



Turn off all the bits in accumulator 1 where the associated bits in accumulator 2 are set "0" to identify the Network ID in accumulator 1:



Next, find the Broadcast address. Turn on all bits in accumulator 1 where the associated bits in accumulator 2 are set to "0":

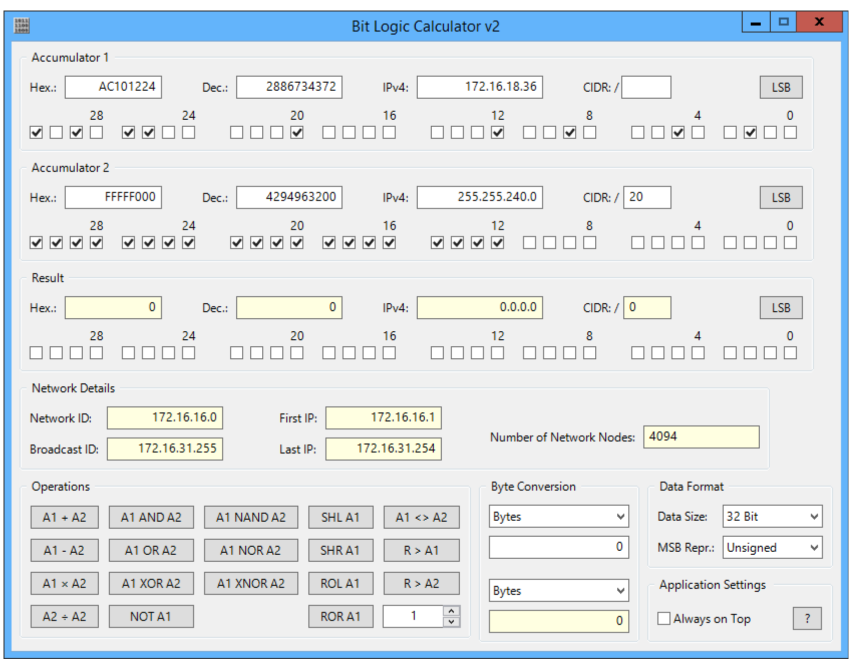


The first usable IP address is the Network ID address + 1. The last usable IP address is the Broadcast address - 1.

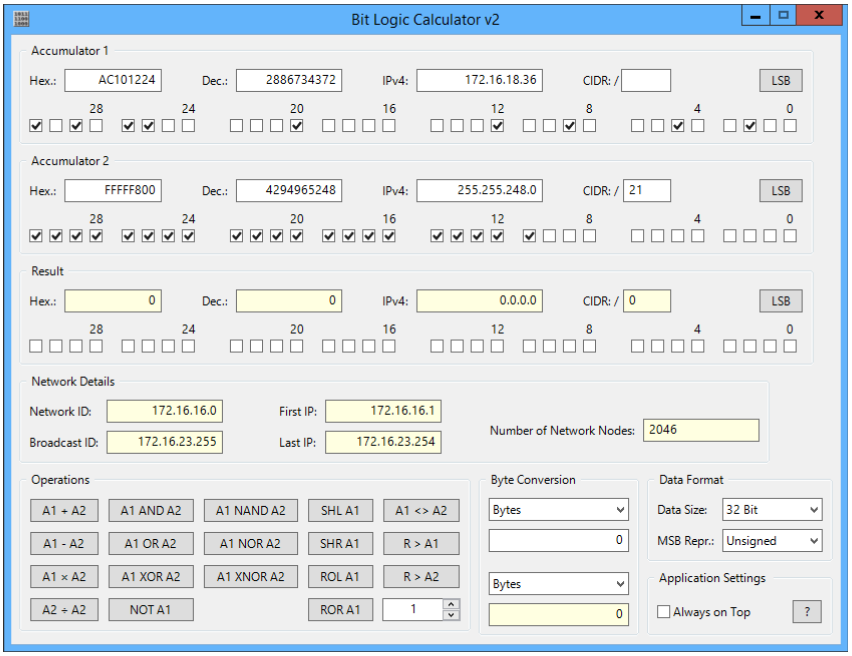
3.2.3 Example 3 - Subnetting

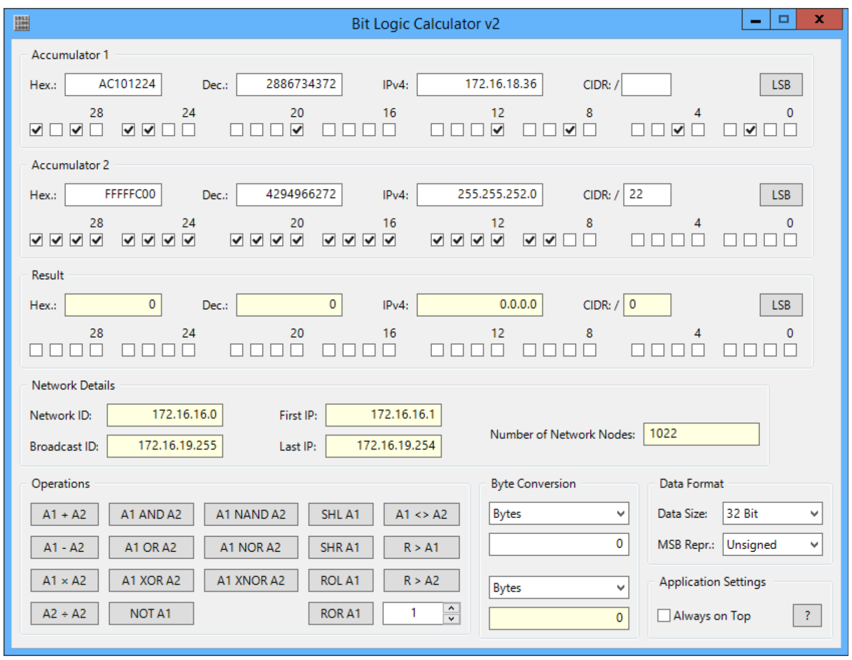
With the knowledge about the relationship between subnet mask and IP address, the concept of subnetting can be easily shown.

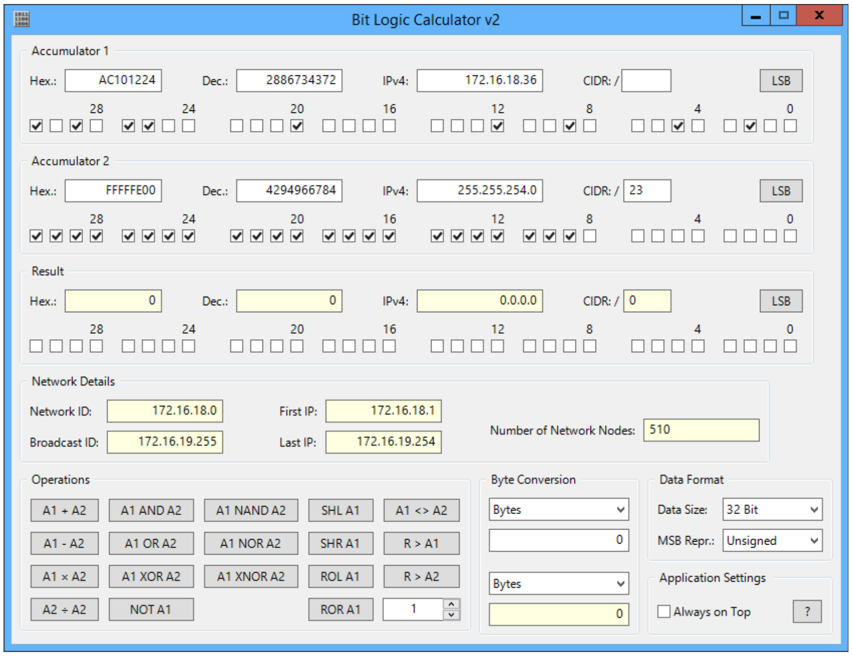
Enter an IP address of 172.16.18.36 into accumulator 1 and a subnet mask of 255.255.240.0 (CIDR = 20) into accumulator 2:

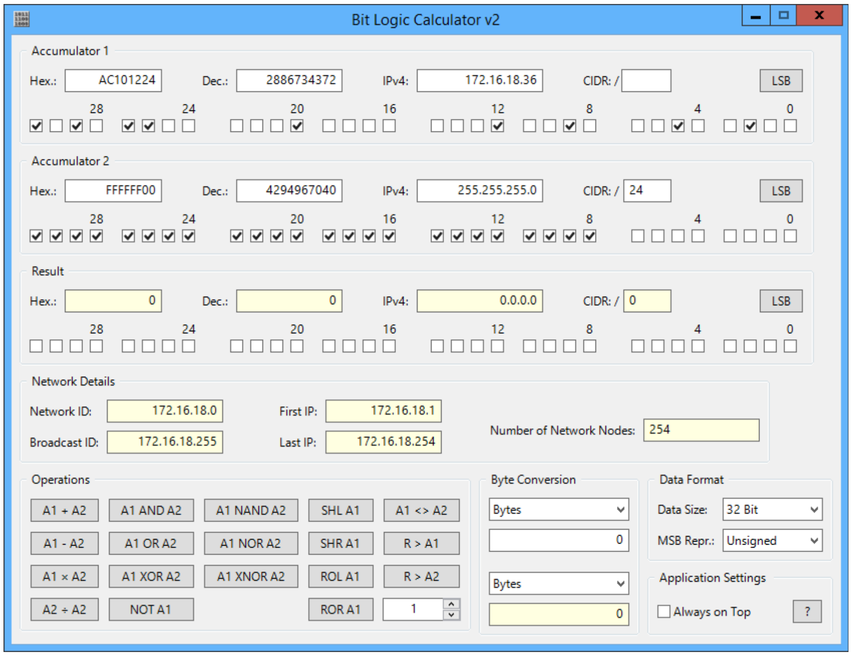


Modify accumulator 2 by either setting or resetting bits at the binary display, thereby expanding or contracting the network range.









Adding bits to the right side of accumulator 2 will effectively half the available network range with every bit position checked. Conversely, the available network range will double for every bit position reset in accumulator 2.

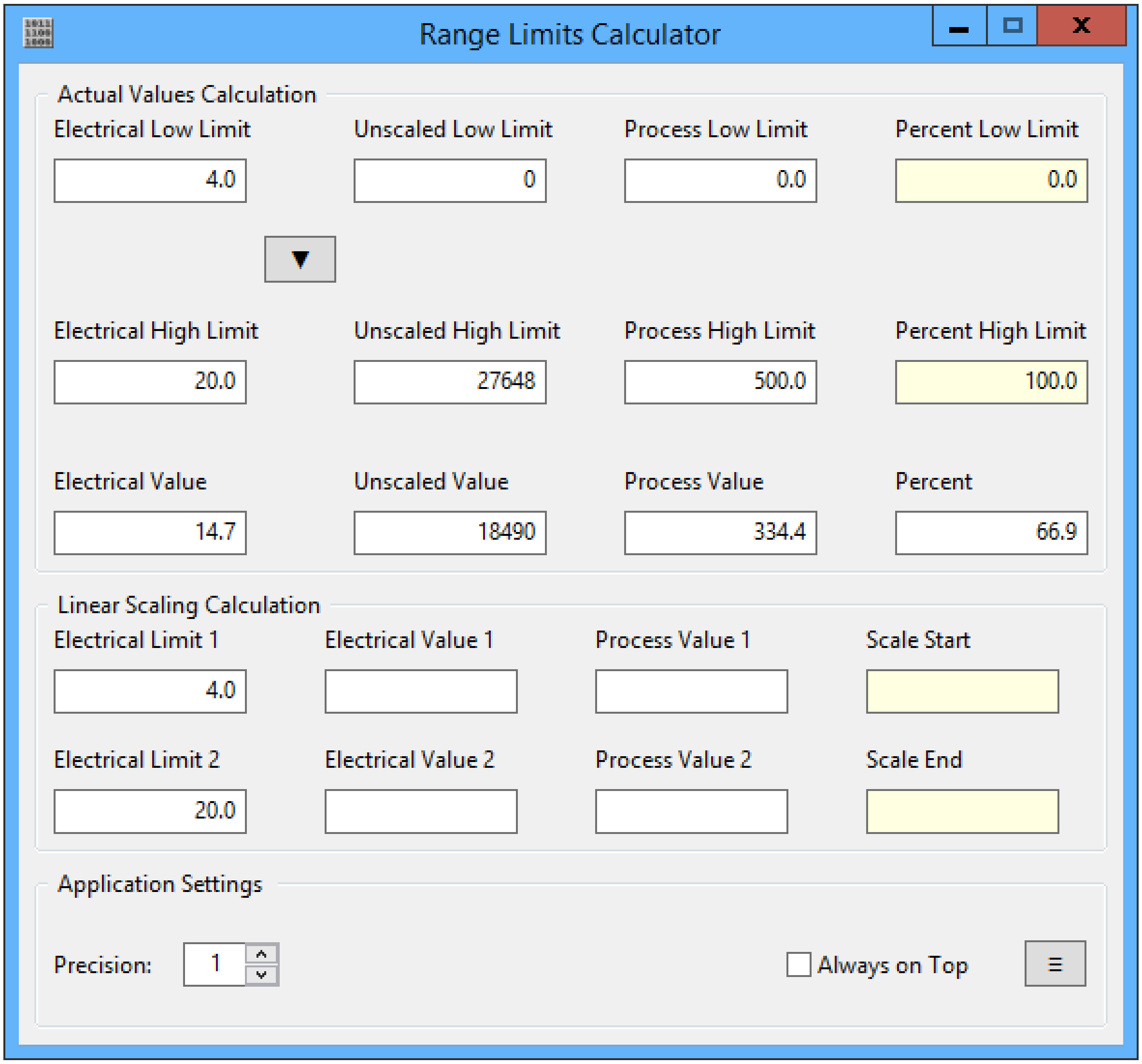
4 Range Calculations

The menu button on the Bit Logic Calculator window can be used to open another application windows: The Range Limits Calculator.

Unlike the Bit Logic Calculator, this application view is less universal and very specific to tasks necessary to Process Automation, specifically the topic of analog-digital conversion used for analog I/O channel calculations.

This window consists of two separate views:

- Measured Value Conversion
- Linear Scaling Calculation



4.1 Measured Value Conversion

This view is used to convert between analog and digital channel data. The calculation is, similar to all BitCalc operations, on-the-fly, so the resulting values are immediately available.

A typical usage scenario is the analysis of analog channel information when using a field bus analyzer, which typically only provides unscaled (raw value) feedback.

Using the range calculations, the expected raw value can be calculated for a specific current feed or voltage feed simulator.

The following conversions are offered:

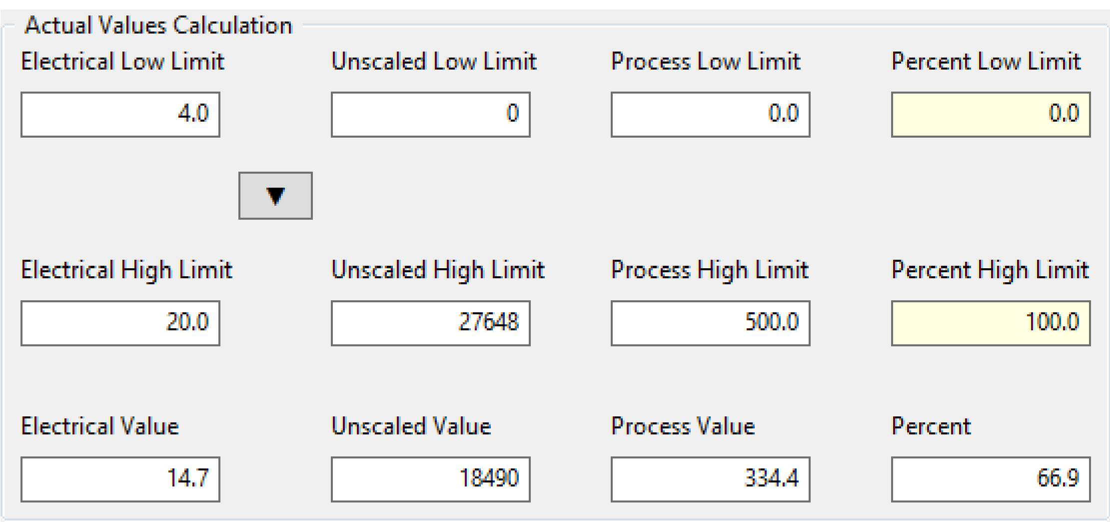
- **Electrical Value** (Current / Voltage) based on channel type
- **Unscaled Value** (Raw Value)
- **Process Value** (Scaled Value)
- **Percentage**

In order for the calculation to be performed, the associated range values for each conversion type have to be filled in first. Each range setting is defined by a minimum and maximum value, which is considered the "nominal range".

For electrical values, some common default settings are available through a drop-down menu. Also, some common raw configurations are provided - however, those are limited to the ranges I frequently have to work with and am therefore very familiar with (presently, only Siemens S7-300 analog I/O card channel data and Siemens TI 505 channel data; those pop-up menus will be extended when I come across more PLC manufacturers and the details on their analog digital conversion methods).

After the ranges are defined, the range conversion is performed, whenever an actual value is entered. A linear interpolation is performed, which calculates the other expected actual values based on the point curve for the value what had been entered by the user.

For example, if all analog ranges are entered and the user enters a value a Process Value, then the application calculates the expected electrical value based on the electrical range, the unscaled value based on the unscaled range as well as the percentage value:



4.2 Linear Scaling Calculation

The Linear Scaling calculation can be used to calculate the necessary analog limits for a transducer as used in Process Automation tasks.

This application makes tuning of a transducer very simple:

Step 1: Enter the analog input card channel range (nominal range, e.g. 4-20 mA) into the associated electrical limits input fields

Step 2: Measure the electrical current/voltage presently present at the transducer level and enter it into the input field for **Electrical Value 1**

Step 3: Enter the presently measured physical value into the input field for **Process Value 1**

Step 4: Move / actuate the device connected to the transducer

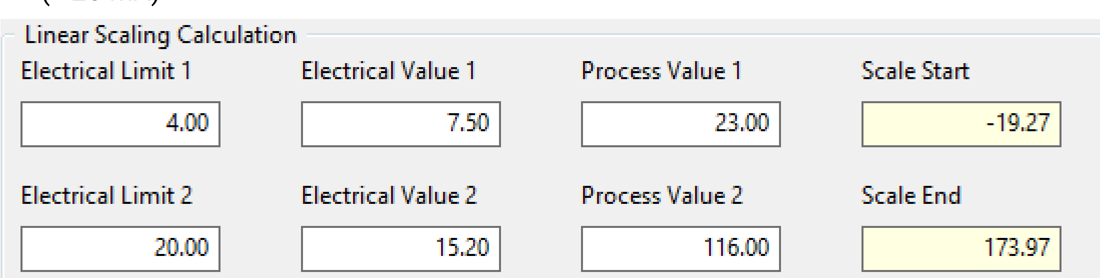
Step 5: Measure the electrical current/voltage presently present at the transducer level and enter it into the input field for **Electrical Value 2**

Step 6: Enter the presently measured physical value into the input field for **Process Value 2**

Step 7: The application now calculates the minimum / maximum scaling ranges for the analog channel based on a linear interpolation, which can now be used in the Process Automation System / PLC

Example: Calculate the scaling ranges for a pressure transmitter connected to a 4-20 mA analog input channel:

1. Enter **4.0** into input field **Electrical Value 1** to define the channel start range
2. Enter **20.0** into the input field **Electrical Value 2** to define the channel end range
3. Make sure that the pressure transmitter is connected to a pressure sensor
4. Measure the current present at the analog input card or at the transducer output, e.g. **7.5** mA, and enter the value into input field **Electrical Value 1**
5. Measure the pressure presently feeding the pressure transmitter, e.g. **23** PSI, and enter that value into input field **Process Value 1**
6. Change the input pressure connected to the transducer
7. Measure the current present at the analog input card or at the transducer output, e.g. **15.2** mA, and enter the value into input field **Electrical Value 2**
8. Measure the pressure presently feeding the pressure transmitter, e.g. **116** PSI, and enter that value into input field **Process Value 2**
9. These values are now used to calculate the scaling range of **-19.27** PSI (= 4 mA) and **173.97** PSI (= 20 mA)



5 Version History

5.1 Version 2.x

[2016-01-11 V2.1]

- Range Limits Calculator functions has been added.
- Bitmask /32 is no longer treated as valid for the network details calculation, since this network mask does not allow a valid IP address range nor network address or broadcast address.
- Bitmask /31 does not attempt to calculate the valid IP address range, since this subnet mask does not allow for any valid node addresses.
- Range Limits calculator has been added. This is a feature very specific to process automation tasks (analog I/O channel data conversion).

[2015-10-11 V2.0]

Initial release of V2.x, with better support for newer operating systems, high resolution screens and the associated automatic interface scaling. Also, networking features are extended.

5.2 Version 1.x

[2007-10-26 V1.6.0.0]

- Network adresses in IPv4 (32 bit) format are supported. Using both accumulators, the relationship between an IP adress and an subnet mask can easily be evaluated using the binary display.
- Most of the input field data evaluation has been rewritten in order to work around a Delphi Bounds check flaw with integer values and in order to better react to data overflows.

[2007-07-06 V1.5.0.0 U3]

- A U3 installer has been created. No further changes.

[2006-09-14 V1.5.0.0]

- A Byte-kByte-MByte-GByte-TByte conversion tool has been added.

[2006-02-03 V1.4.0.0]

- Both accumulators and the result can now use either '0' or '1' as Least-Significant-Bit. This will change the label above the bit positions as well as their tooltip texts, for use with systems that start with '1' instead of the usual '0'.
- Changed transparency limiter to 25% minimum.

[2005-06-10 V1.3.1.0]

- Changed AlphaBlend behavior for transparency handling.

- Limited transparency to 50% so the form cannot completely disappear.
- Used the integrated forms modifier for Always-On-Top in Delphi instead of a Win32 API call.

[2004-01-04 V1.3.0.0]

- Added tool tip text display for (almost) all control elements.

[2002-08-21 V1.2.0.0]

- Added transparency option for Windows 2000 or higher.
- Added Windows XP theme support.

[2002-??-?? V1.1.1.0]

- Fixed compiler error with missing 'Highlight' control (which was used in a previous version).
- Fixed problem with interpretation of signed values in case of an overflow.

[2001-??-?? V1.0.0.0]

- Initial release.