



Keywords: exponent, floating point, format, I2C, linear, LINEAR11, mantissa, PMBus, serial, SMBus.

TUTORIAL 6943

# EXCEL ADD-IN FUNCTIONS FOR PMBUS NUMBER FORMATS

*Abstract: This Visual Basic Add-In file for Microsoft Excel provides custom spreadsheet functions that convert to and from the PMBus LINEAR11 format.*

## Introduction

This [Visual Basic<sup>®</sup> Add-in file for Microsoft Excel<sup>®</sup>](#) provides custom spreadsheet functions that convert to and from the PMBus<sup>™</sup> LINEAR11 format. This enables programmers and power systems engineers to easily convert PMBus LINEAR11 data received from an IC device into decimal values, or to convert decimal data into LINEAR11 format for transmission to a PMBus device.

In addition, there are special functions that extract the mantissa and exponent separately, which helps new users understand the structure and use of the LINEAR11 number format. This makes the PMBus Add-in a valuable educational tool as well.

These functions are written in Microsoft Visual Basic and can be added to any installation of Microsoft Excel as a temporary or permanent add-in, so that the functions are always available, very much like native Excel spreadsheet functions.

## The PMBus LINEAR11 Number Format

The PMBus standard includes a special format for representing floating-point numerical values. This format, called LINEAR11, is the preferred way to send and receive numerical data for all parameters other than output voltage.

This number format is two bytes long and comprises a 5-bit exponent and an 11-bit mantissa, both of which are signed numbers in two's-complement notation. **Figure 1** shows the structure of the 16-bit LINEAR11 data format.

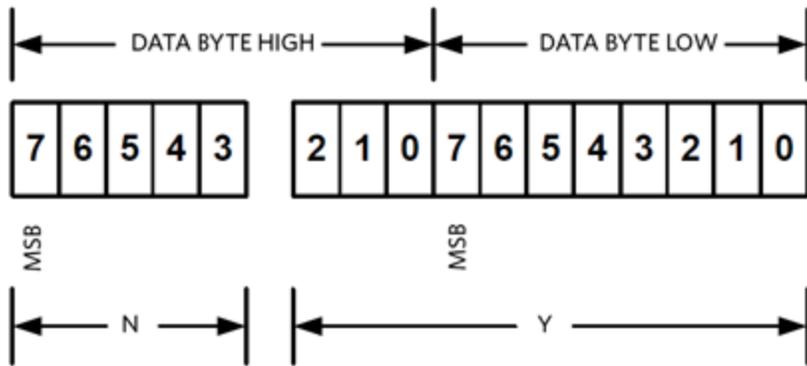


Figure 1. The structure of the LINEAR11 data format.

The exponent is represented by N and the mantissa is represented by Y. Because the mantissa spans the boundary between the two bytes of data, it is difficult to interpret these values "by eye." Furthermore, inherent limitations of the binary and hexadecimal analysis functions in Microsoft Excel make it rather difficult to parse these values in a spreadsheet.

To simplify use and understanding of this PMBus number format, Maxim has created an Excel "custom function" Add-in that can be easily installed on any PC running Microsoft Excel. Once this Add-in is loaded, it is always available for use, just like Excel's built-in functions.

### Installing the Add-In

The PMBus add-in provides optional commands and conversion functions to Microsoft Excel (Figure 2). To add it permanently to an Excel installation, follow this procedure:

1. Place the .XLAM file in the C:\Users\[username]\AppData\Roaming\Microsoft\AddIns\ folder.
2. Open Excel.
3. Click the **File** tab, then click **Options**. The **Excel Options** dialog box will appear.
4. Click the **Add-ins** category. In the **Manage** drop-down list, click **Excel Add-ins**, and then click **Go**. The **Add-Ins** dialog box appears.
5. In the **Add-ins available** box, select the check box next to the add-in that you want to activate, and then click **OK**.
6. To ensure optimum operational compatibility, make any necessary adjustments in the [Microsoft Excel Trust Center](#).
7. If the Add-in functions do not always appear in spreadsheets when entering a formula in a cell beginning with "=", place a copy of the .XLAM file in the C:\Users\[username]\AppData\Roaming\Microsoft\Excel\XLSTART folder.

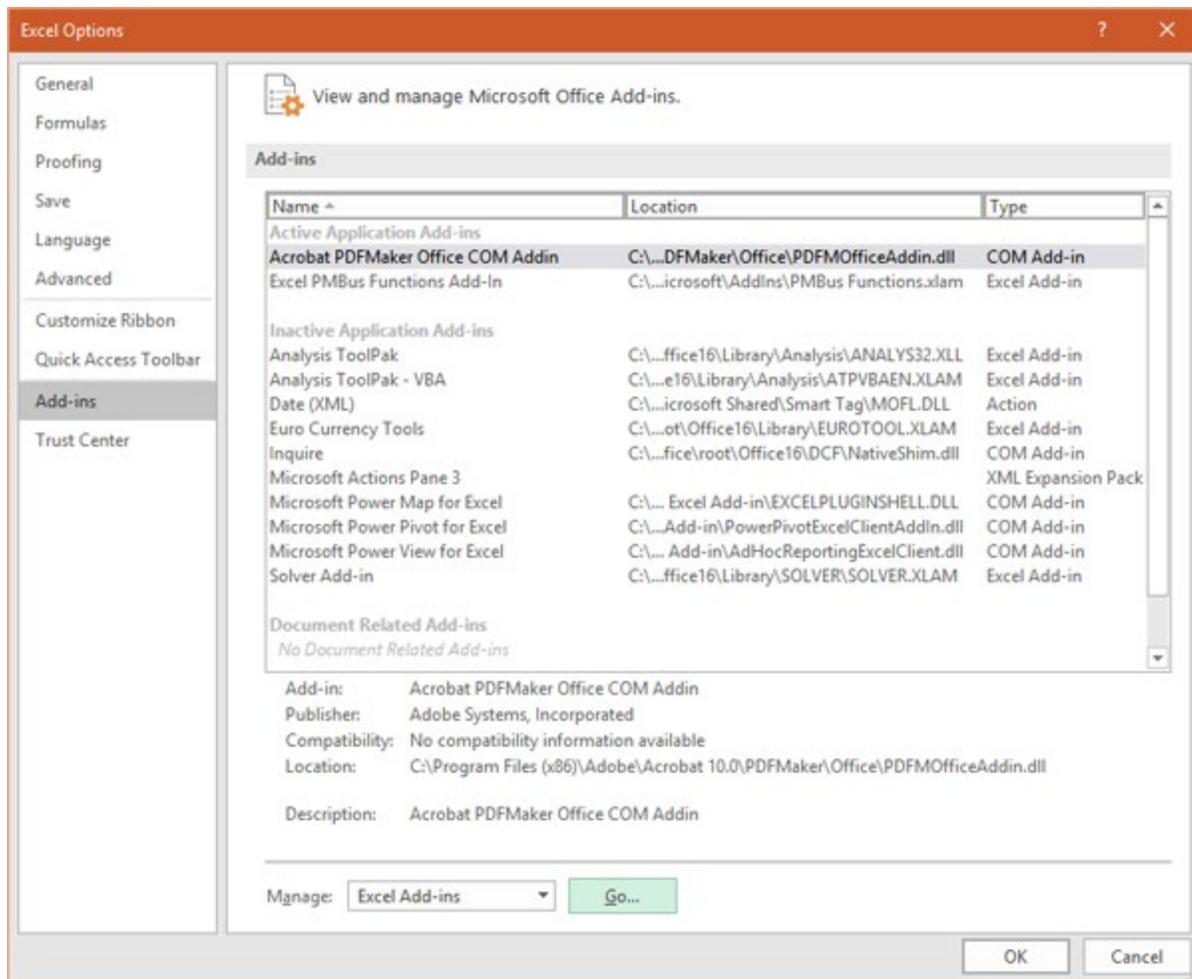


Figure 2. Managing Add-Ins in Microsoft Excel

## Custom Function Naming Convention

The functions were named in a pattern following the existing Microsoft Excel "Analysis ToolPak" add-in functions, for example Excel's DEC2HEX() function, which converts decimal (DEC) values to their hexadecimal (HEX) equivalent.

For these custom PMBus functions, the following mnemonic abbreviations are employed:

- HEX = hexadecimal
- DEC = decimal
- L11 = LINEAR11
- 2 = "to"
- EXP = exponent
- MAN = mantissa
- 4 = "for"

## Using the Custom PMBus Functions

Here's how to use each of the custom functions included in this add-in.

### Decimal to LINEAR11: DEC2L11()

This function converts decimal values into the LINEAR11 format. The conversion will optimize the exponent

to maintain highest possible resolution, unless a specific exponent is "locked" by the optional [exponent] argument.

Format: DEC2L11(number,[exponent],[hexadecimal]) Arguments:

- number: A decimal value you want to convert to LINEAR11, from -33554432 to 33521664.
- exponent (optional argument): Force a specific exponent (N) value, from -16 to +15.
- hexadecimal (optional argument): Boolean value, 0 = return unsigned integer, 1 = return 2-byte hexadecimal.

### **LINEAR11 to Decimal: L112DEC()**

This function converts LINEAR11 values into their decimal equivalent.

Format: L112DEC(number)

Arguments:

- number: Either a hexadecimal text value or an unsigned integer in LINEAR11 format that you want to convert back to decimal. Because this function recognizes arguments in two formats, hexadecimal LINEAR11 arguments must be entered as a text string (use quotes for literals), and integer arguments must be entered as numbers.

### **Exponent for LINEAR11 value: EXP4L11()**

This function returns the exponent value for any given LINEAR11 value.

Format: EXP4L11(number)

Arguments:

- number: A literal 16-bit integer representation of a LINEAR11 value.
- Values outside the unsigned 16-bit range of 0-65535 are rejected, as are non-numeric values.

### **Mantissa for LINEAR11 value: MAN4L11()**

This function returns the mantissa value for the LINEAR11 argument.

Format: MAN4L11(number)

Arguments:

- number: A literal 16-bit integer representation of a LINEAR11 value.
- Values outside the unsigned 16-bit range of 0-65535 are rejected, as are non-numeric values.

### **References/Other Resources**

In addition to the add-in file itself, there is an Excel spreadsheet that shows these functions in use. Please refer to the examples in this file to see how these functions can be used to aid design and development of a PMBus system.

Excel is a registered trademark of Microsoft Corporation.

PMBus is a trademark of SMIF, Inc.

Visual Basic is a registered trademark of the Microsoft Corporation.

## Related Parts

<a href="#">MAX15301</a>	InTune Automatically Compensated Digital PoL Controller with Driver and PMBus Telemetry	
<a href="#">MAX15303</a>	6A Digital PoL DC-DC Converter with InTune Automatic Compensation	<a href="#">Samples</a>

---

### More Information

For Technical Support: <https://www.maximintegrated.com/en/support>

For Samples: <https://www.maximintegrated.com/en/samples>

Other Questions and Comments: <https://www.maximintegrated.com/en/contact>

---

Application Note 6943: <https://www.maximintegrated.com/en/an6943>

TUTORIAL 6943, AN6943, AN 6943, APP6943, Appnote6943, Appnote 6943

© 2014 Maxim Integrated Products, Inc.

The content on this webpage is protected by copyright laws of the United States and of foreign countries.

For requests to copy this content, [contact us](#).

Additional Legal Notices: <https://www.maximintegrated.com/en/legal>