



SMARTEH
LIVING SYSTEMS

USER MANUAL

— Longo programmable controller
LPC Manager

Written by Marko Gomiršek
Copyright © 2004-2005, SMARTEH d.o.o.

User Manual
Software Version: 2.35.F
Document Version: 002
January 31, 2005



Index

Longo programmable controller LPC Manager

1 ABOUT THIS DOCUMENT.....	1
1.1 LPC Family Software Documentation.....	1
1.2 Who Should Read This Document.....	1
1.3 What This Document Contains.....	1
1.4 Terminology.....	2
1.4.1 LPC Family products based terminology.....	2
1.4.2 LPC Manager based terminology.....	2
1.4.3 LON based terminology.....	2
2 FIRST STEPS.....	3
2.1 Why LPC Manager?.....	3
2.2 Installation.....	3
2.2.1 Installation.....	3
2.2.2 Starting the application.....	3
2.3 Quick Tutorial.....	3
2.3.1 Main window.....	3
2.3.2 Creating new project.....	4
2.3.3 Basic settings.....	4
2.3.4 Building Stand-alone application.....	6
2.3.5 Upgrade to Network system.....	6
2.3.6 Finishing.....	7
3 APPENDIX A - COMPATIBILITY.....	8
4 APPENDIX B - ERROR REPORTING.....	9



1 ABOUT THIS DOCUMENT

LPC Manager documentation is part of the LPC software documentation. This introduction provides an overview of the LPC software documentation.

1.1 LPC Family Software Documentation

The following documents are part of the LPC family software documentation:

Document	Description
<i>LPC Software Companion</i>	A guide to the LPC software.
<i>LPC Composer</i>	LPC Composer software manual.
<i>LPC Manager</i>	LPC Manager software manual.

1.2 Who Should Read This Document

This document is intended to be a manual for the *LPC Manager*. It is designed for the user new to *LPC Manager*.

1.3 What This Document Contains

- Chapter 1, “*First Steps*”. Provides new users with the basic knowledge to get them going with the software. A quick tutorial is provided to explain frequently used features. You should be able to use most of the program functions after you complete this chapter.
- Appendix A, “*Compatibility*”. Informs you about system requirements.
- Appendix B, “*Error Reporting*”. Provides you with the list of what should be described in case you found a bug or have an improvement idea.



1.4 Terminology

Throughout this manual, various phrases are used. Here is a description of some of them.

1.4.1 LPC Family products based terminology

LONGO™

... is a family of products (hardware and software) and is a trademark of SMARTEH d.o.o.

LPC-2 Programmable Controller

... is a family of hardware modules (MCU module and I/O modules).

LPC Composer, LPC Manager

... are members of LPC family software.

1.4.2 LPC Manager based terminology

LadderWORK™

... is a trademark of MicroSHADOW Research.

Ladder programming language

One of five CEI/IEC 1131-3 standard programming languages for PLCs and is based on graphical representation.

1.4.3 LON based terminology

LON®, LonWorks®, LonMaker™ for Windows

... are trademarks of Echelon Corporation. LON is an abbreviation for Local Operating Network.

SNVT

Standard Network Variable Type (pronounced “snivet”). Use of these variables types contributes to the interoperability of LonWorks® products from different manufacturers. A complete list of SNVTs (SNVT Master List) is available from the LonMark Association.



2 FIRST STEPS

2.1 Why LPC Manager?

LPC-2 (LONGO Programmable Controller) is a product that can be used in many different fields. Therefore it has to be as versatile as possible. Modularity allows this. After *LPC Composer* has composed the corresponding configuration of the LPC-2 modules and the configuration is physically build then *LPC Manager* is used to develop software application for this configuration. For this purpose the standard CEI/IEC 1131-3 ladder programming language (LD) is used. Refer to *LPC Manager Help* for information about ladder programming language and other CEI/IEC 1131-3 languages.

2.2 Installation

2.2.1 Installation

Standard Windows installer makes this step straightforward. Just an executable file has to be started. For fully functional version of *LPC Manager* the hardware protection key must be installed on the PC. For this purpose run the *sdi.exe* in the *smartkey* folder on the CD-ROM, select the correct port (e.g. *Parallel* or *USB*) and press *Install* button.

2.2.2 Starting the application

Once installed, you can find the application icon in the Windows' start menu.

2.3 Quick Tutorial

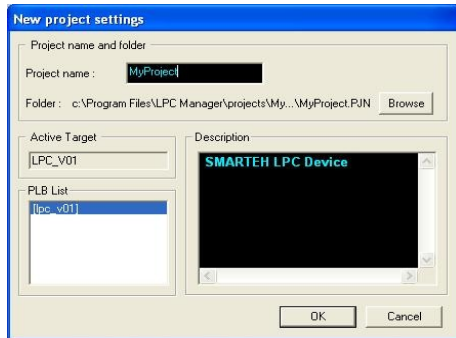
2.3.1 Main window

When the application is started, the main screen appears. As you can see there is the standard Windows menu on the top and status bar on the bottom of the window.

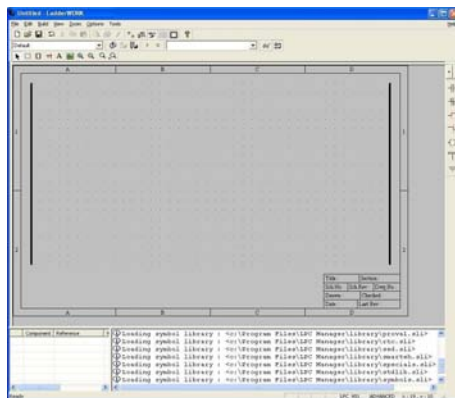
The main part of the screen displays current project. It is divided into two panels. The top part displays view window, while the bottom part is showing the messages window. The bottom part can be resized so that view window can be greater.



2.3.2 Creating new project

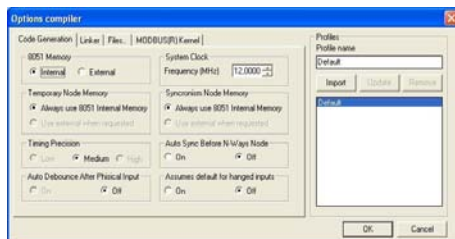


At the moment the main window looks a bit empty. This is because we don't have any project opened nor created. So for the purpose of this tutorial we will create one. Select *File|New* from the main menu. A dialog will appear that will prompt you to enter *Project name*. Write the desired name (e.g. *MyProject*) as shown in the screen-shot and then press *OK* button.

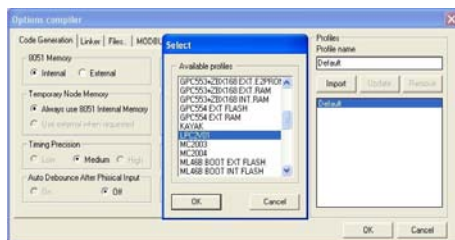


The main view window is opened and in the message window we can see that some libraries are loaded as well. The bottom part can also be rearranged that the watch window (if used) is visible in the bottom left corner. On the right side there are bar menus with standard and custom LD components.

2.3.3 Basic settings



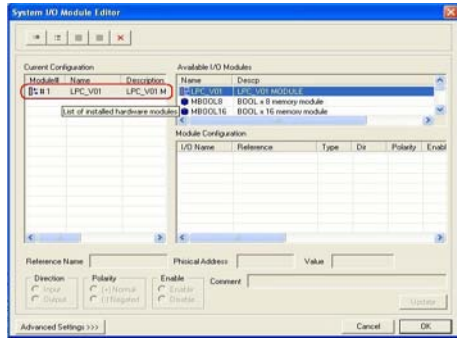
First at least one of the components has to be placed in the view window (e.g. *INPUT*) from the right side bar menu. This must be done otherwise *LPC Manager* will not allow to enter basic settings. Select *Options|Compiler...* from the main menu or press *CTRL+F5*. A dialog window *Options compiler* is opened.



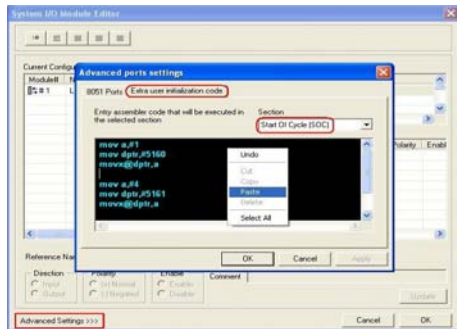
Click the *Import* button and from the *Select* window choose *LPC2V01* device profile and press *OK* button.

The *LPC2V01* device profile is added to the current profile list. Press the *OK* button in the *Options compiler* window too.



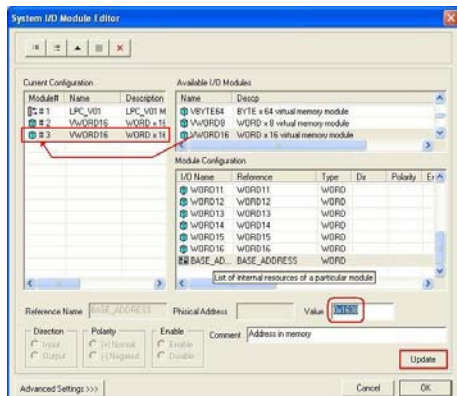


Select *Options|System Edit* from the main menu. A *System I/O Module* dialog window is opened. From the list of available hardware modules double-click the **LPC_V01** device to install it. The device is now present in the list of installed hardware modules. NOTE: **LPC_V01** module must always be present in the first slot with the module number #1.

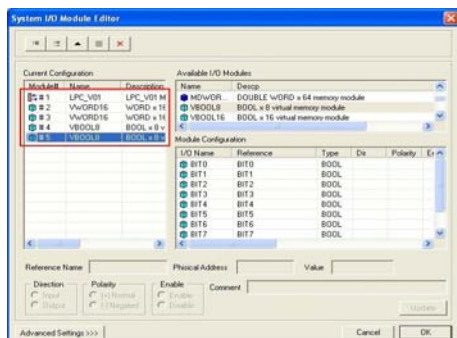


For the purpose of our tutorial, we will continue with the calculated configuration from the *LPC Composer* (refer to *LPC Composer* tutorial). This configuration is MC3 A01 D11 DI1.

Click the *Advanced Settings >>>* button. *Advanced ports settings* window is opened. Click the *Extra user initialization code* tab and from the *Section* drop list select *Start of Cycle (SOC)*. Paste the assembler startup code which has been copied to the clipboard by *LPC Composer* software (refer to *LPC Composer* tutorial).



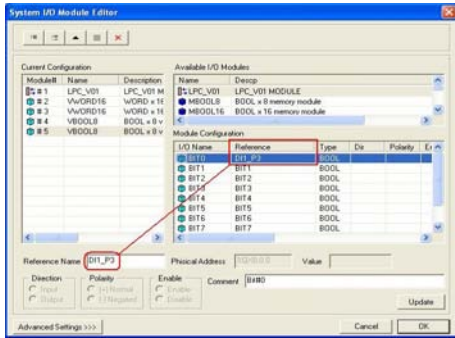
Press the *OK* button. If the hardware configuration would be changed anytime in the future then the *User initialization code* must be updated. To do this the current code must be deleted first before new one is pasted from *LPC Composer*.



For all three I/O modules corresponding memory blocks must be installed. This is done in the *System I/O Module Editor*. First I/O module in our tutorial is analogue I/O module A01 (for technical information refer to LPC-2.A01 User manual). The *LPC Composer* right panel shows the memory blocks which has to be installed. For six inputs the **VWORD16** block with **BASE_ADDRESS** 0x1268 and for two outputs the **VWORD16** block with **BASE_ADDRESS** 0x1638 must be installed. These two have module numbers #2 and #3 in the installed hardware modules list. Each change for the specific field must be confirmed by clicking the *Update* button.

For DI1 digital input modules two **VBOOL8** memory blocks have to be installed. **BASE_ADDRESS** for the first one is 0x1061 (#4) and for the second one 0x1062 (#5).

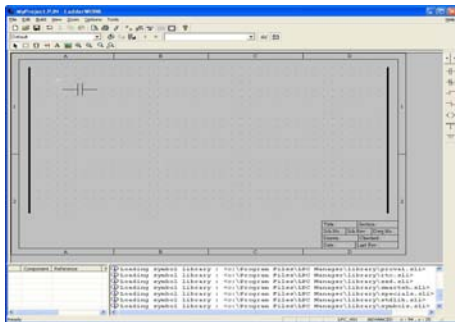




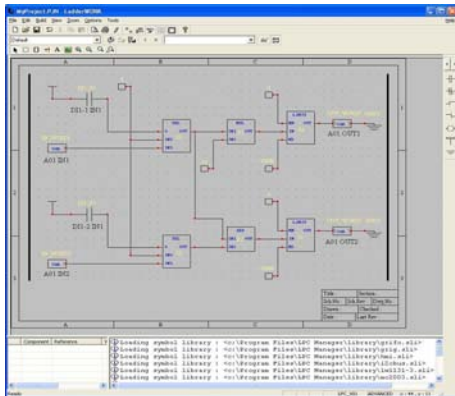
Reference Name for every memory variable can be set to be more representative for the developer.

For LON network usage all needed memory modules must be installed as well (refer to LPC-2.MC3 Help). After all memory modules are present in the installed hardware modules press *OK* button to return to the view window.

2.3.4 Building Stand-alone application



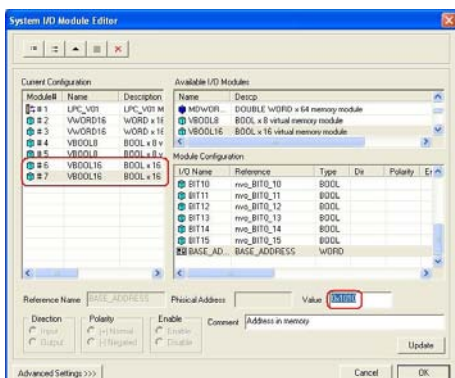
Before we create a logics for selected I/Os we need to save our project; go to *File|Save* menu and save with the name “MyProject” (you may also choose another name if you want). Now the selected name will be displayed in the title bar.



For the purpose of this tutorial we will connect IN1 multiplied by 100 to OUT1 and IN1 + IN2 to OUT2 on LPC-2.A01 module. We will also add a HW digital input in parallel as an enable / disable flag. Both inputs on LPC-2.A01 module can be simulated with a potentiometers and both digital inputs DI1 can be simulated with two switches. Screen-shot shows an example of the application. This example is build for stand-alone system.

The application is now ready to be compiled and uploaded to the LPC-2.MC3. PMC programming cable must be used to connect to the COM connector on the LPC-2.MC3 module.

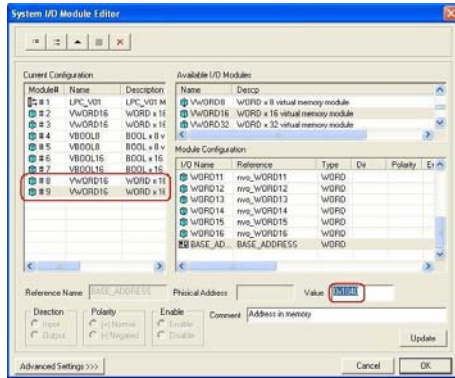
2.3.5 Upgrade to Network system



After we have built the stand-alone application we will upgrade our application to the network application. The scope of this upgrade is that all inputs and both outputs will be connected to the LON network.

First we need to define memory blocks for the network in the *System Edit*. For input and output boolean variables we will add two VBOOL16 memory blocks. Input block (#6) has BASE_ADDRESS 0x1000 and output block (#7) 0x1010. These two blocks define two network variables called nvi_BIT0_xx (hence *nvi* - input from the network) and nvo_BIT0_xx (hence *nvo* - output to the network). Each network variable of this type (SNVT_state) is a structure of 16 bits.

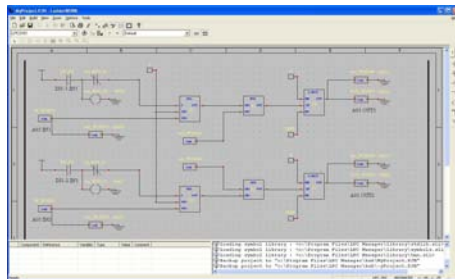




For analogue variables we need to define two memory blocks, one for input and one for output network variables. Both are of type VWORD16. Input block (#8) have BASE_ADDRES 0x1020 and output block (#9) 0x1040. Each network variable of this type (SNVT_count) is unsigned integer (range 0..65535).

All network variables can be browsed by a standard network management tool (e.g. LonMaker™ for Windows). Refer to LonMaker™ for Windows User's Guide for more information.

Now we are ready to upgrade the logics inside the view window. See the screen-shot of the final application upgraded with the LON network interface.



The upgraded application is now ready to be compiled, uploaded and tested. PMC programming cable must be used to connect to the COM connector on the LPC-2.MC3 module.

2.3.6 Finishing

Congratulations, you have just completed the basic tutorial. We have covered most of the tasks you will use in *LPC Manager*. As you can see, the use of the application is simple and intuitive.

The tutorial ends here. Check *LPC Manager* Help for more detailed information about specific application features.



3 APPENDIX A - COMPATIBILITY

LPC Manager has the following system requirements:

- Microsoft Windows 98 or later.
- At least 128Mb RAM.
- 30Mb hard disk space.

IMPORTANT: It is known that some DELL laptops are having problems with RS-232 communication. DELL provides BIOS patch to fix this problem. Please read DELL's information at <http://support.ap.dell.com>.



4 APPENDIX B - ERROR REPORTING

If you think you found a bug in our software or you have an idea of what can be improved or added, you are most welcome to share your thoughts with us. We will consider the possibilities and try to include them in our next release.

You should contact your vendor with the description. The following information should be included:

- Software version.
- Detailed description of the bug or idea.
- If possible, steps that will recreate the problem (if bug is being reported).
- Your contact information (e-mail, phone, fax).

In case we need more information we may need to contact you before we can determine the exact solution.

And remember: the only software without a need for maintenance is the software not being used!

