How to use the Oberon System

Astrobe Embedded Project Oberon
CFB Software, Oct 2021

Adapted from the original authored by
Niklaus Wirth, Nov 2015

The Oberon System was designed and implemented in 1990 as an example of a modern, interactive operating system with windows and mouse. In contrast to commercial systems it was to be simple and well-structured in order to be published in its entirety and available for teaching the essentials. It was programmed in the concise language Oberon, and it requires only a small fraction of the resources which commercial systems demand.

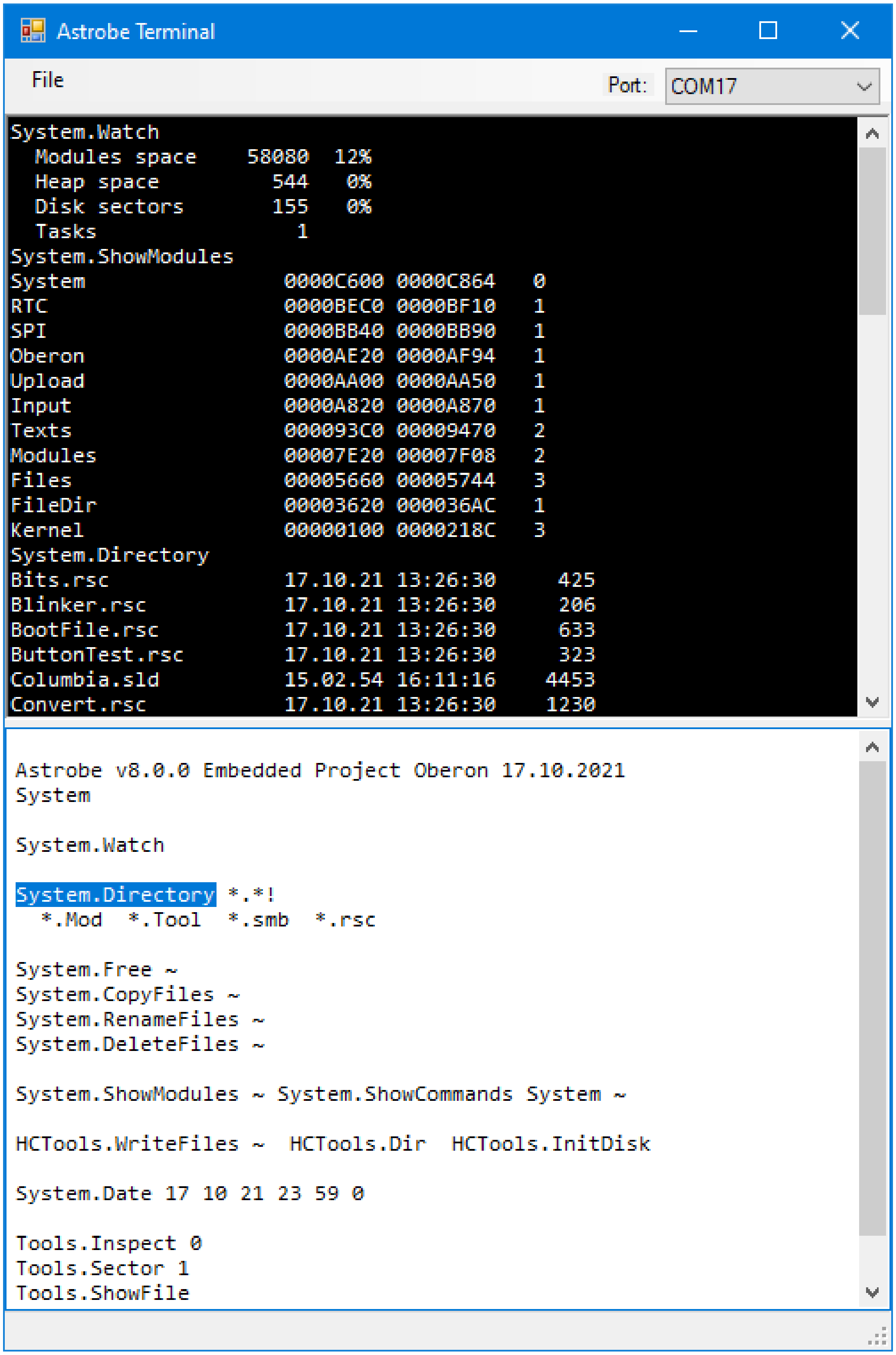
Embedded Project Oberon is a subset of the Oberon System, It includes the Kernel, loader, filesystem, SPI, RS232 and SD card support. It is intended for use on embedded target systems where limited user-interaction is required e.g. dataloggers, autonomous model vehicles, equipment control systems etc.

The higher-level layers of the Project Oberon OS with its GUI, VGA display, mouse, keyboard, compiler, editor etc. are not required on the target FPGA system because all software development can be done on Windows.

Interaction with applications running on the FPGA target development board is achieved using the Astrobe Terminal on a Windows PC connected via a 115200 baud serial link. The user interface was inspired by the Project Oberon GUI. While not as powerful, it is significantly more flexible than the typical command-line interface of serial terminals and consoles. The following describes the operation and available commands.

How to begin

Assuming you have installed Astrobe for RISC5 and followed the instructions in the Getting Started section of the Astrobe Help file, select *Run* > *Run* from the Astrobe menu. The following Astrobe Terminal display appears:



```
System.Watch
  Modules space    58080  12%
  Heap space         544   0%
  Disk sectors       155   0%
  Tasks                1
System.ShowModules
System                0000C600 0000C864   0
RTC                   0000BEC0 0000BF10   1
SPI                   0000BB40 0000BB90   1
Oberon                0000AE20 0000AF94   1
Upload                0000AA00 0000AA50   1
Input                 0000A820 0000A870   1
Texts                 000093C0 00009470   2
Modules               00007E20 00007F08   2
Files                 00005660 00005744   3
FileDir               00003620 000036AC   1
Kernel                00000100 0000218C   3
System.Directory
Bits.rsc              17.10.21 13:26:30     425
Blinker.rsc           17.10.21 13:26:30     206
BootFile.rsc          17.10.21 13:26:30     633
ButtonTest.rsc        17.10.21 13:26:30     323
Columbia.sld          15.02.54 16:11:16    4453
Convert.rsc           17.10.21 13:26:30    1230
```

```
Astrobe v8.0.0 Embedded Project Oberon 17.10.2021
System

System.Watch

System.Directory *.*!
  *.Mod  *.Tool  *.smb  *.rsc

System.Free ~
System.CopyFiles ~
System.RenameFiles ~
System.DeleteFiles ~

System.ShowModules ~ System.ShowCommands System ~

HCTools.WriteFiles ~  HCTools.Dir  HCTools.InitDisk

System.Date 17 10 21 23 59 0

Tools.Inspect 0
Tools.Sector 1
Tools.ShowFile
```

The upper (output) section displays the output from applications which are running on Embedded Project Oberon. It is a normal Windows text display so the standard Windows shortcut keys (CTRL-A, CTRL-C, CTRL-V etc.) can be used to select, copy and paste the contents to other Windows applications.

The lower (input) section is used to execute and maintain a list of available commands. It is an editable text window so you can add, edit or delete commands or any other text. The contents of this window can be save as a text *.*Tool* file for use in future sessions. You can have several tool files saved for different uses at different times. The system is supplied with two tool files. *System.Tool* is shown above. *Examples.Tool* contains commands to run the Astrobe examples in addition to the commonly used system commands.

The relative sizes of the upper and lower sections can be adjusted by dragging the splitter bar which separates them.

Commands

Commands in Oberon have the form *M.P*, where *M* is the name of the module containing the procedure *P*. Commands are activated by pointing at them and double-clicking the left mouse button (the *command button*).

The following commands appear in the standard tool *System.Tool*. Examples of the parameters which follow the command are shown. The character ~ indicates that a list of names must be inserted before the ~ character, and the command will apply to all named objects.

```
System.Directory            e.g. *.*  *.Mod   *.rsc
System.CopyFiles => ~       e.g. file1 => file2  file3 => file4~
System.RenameFiles => ~     e.g. file1 => file2  file3 => file4~
System.DeleteFiles ~        e.g. file1 file2 file3~
System.ShowModules
System.ShowCommands         e.g. System
System.Free ~               remove specified modules from memory
```

When clicking a command *M.P*, module *M* is searched in the file system and, if found and not already present, loaded into memory. Then its command *P* is searched and executed (if found). A list of loaded modules can be generated by the command *System.ShowModules*, and a list of its commands is obtained by the command *System.ShowCommands*.

Any exported parameter-less procedure in any (compiled) module is accessible as a command. Its parameters are accessed via a *Scanner*. Refer to the source code of the procedure *Parameters* in the supplied example module *Test* to see how this is done. The example command *Test.Parameters* can be executed to see the result.

The core of the Embedded Oberon System

The system's core consists of a loop, *Oberon.Loop*, which continuously checks for an upload request or for character input to appear. The command is identified, control is dispatched, and the command is executed. A command may stem from an explicit double-click (left button) on a text of the form *M.P*.

A further source of input is the keyboard. If CTRL-Enter key is pressed, the command that the cursor is positioned on is executed.

Any other keyboard operations can be used to modify the text in the *Input* area of the display just like a normal Windows text editor. You can add your own commands or modify the existing ones.

The initially loaded system contains, apart from module Oberon, the command module System, the loader and linker (module Modules), a file system (modules Files, FileDir), and the disk space manager and the garbage collector (module Kernel).

References

The original version of this document, *Using Oberon*, which summarises how to use the full Project Oberon system, can be downloaded from:

https://www.inf.ethz.ch/personal/wirth/ProjectOberon/index.html

http://www.projectoberon.com/