The TpForth project

Reuben Thomas*
Technopoint International Inc.†

23rd August 1999

Abstract

The TpForth project aims to produce a comprehensive Forth development and run-time environment for embedded systems. TpForth already sports a visual IDE, with a real-time VM running on several platforms and remote debugging. CASE and project management support, code analysis and native code generation are planned. The system is available under a novel "community license", which aims to build a community of users while protecting commercial interests.

1 Introduction

The TpForth project is a cooperative project run by Technopoint International, Inc., aiming to produce a freely-available comprehensive Forth development and run-time environment for embedded systems. Not only does TpForth offer a state of the art IDE with an unrivalled set of features, but it is freely available under a special "community license", which aims to encourage open development while protecting users' commercial interests.

This paper is organised as follows: section 2 describes why the project was started; section 3 discusses the novel licensing arrangements; section 4 sketches the current state of the system, and section 6 its future; and section 7 examines its implications for the Forth community at large.

2 The genesis of TpForth

The TpForth project was started because Technopoint realised that it was difficult for a small company to have all the expertise in-house needed to develop technologies and products to satisfy a rapidly changing market. To evolve quickly enough, they needed a new strategy and marketing model. At the same time, the spread of the internet was making it practical for geographically distant partners to collaborate, especially on software projects, as well as providing a new means for marketing to a world-wide audience.

*rrt@sc3d.org
†http://www.technopoint.net/tpforth/

For a group of companies working in embedded systems, an obvious focal point is the tools they use, in particular, their software development tools. Technopoint decided to form a group of companies based around a common development system, which could be developed by a community collaborating over the internet. They aimed to use a software development model very similar to that adopted by what became known as the open source movement: Technopoint would coordinate development of the project, make frequent updates, and the system should be freely available. This would encourage new users to join in, and lower the cost of marketing. However, an open source license was unsuitable, as it would prevent companies owning their modifications and improvements to the system, and discourage new users from joining. A new license was needed, that would foster the development of a community while allowing members to retain control over their intellectual property.

In order to distribute it under their new license, Technopoint needed to own their development system, so it was natural to develop a new system. Also, no existing system met all their needs. TpForth had already been started, so it seemed natural to use it as the basis for their project; as the focus was on embedded systems, Forth was the natural choice of language.

3 The license

The license adopted by the TpForth project divides users of the system into three classes: non-commercial users (which includes private and educational use, as well as evaluation by potential commercial users), commercial users, and partners. Private use is free, whereas commercial users pay an annual fee; no royalties are required. Partners are commercial users who have made a significant contribution to the project; this might consist of adding functionality to the IDE, contributing a new library, porting the run-time system to a new architecture, or maintaining a nationalised version.

The category of "partner" is central to the license, as it encourages contributions, which in turn help other users. This in turn improves the system, thus encouraging even more users to take it up and improve it. Users who do not wish to donate major contributions can of course sell them to Technopoint; the license also imposes the condition that bug-fixes must be donated free of charge, a condition which aims to make the product as reliable as possible. Technopoint is also keen to encourage and sponsor research and development work in academia.

The run-time system and IDE are treated slightly differently by the license: the source code of the run-time system is available to all users, whereas that of the IDE is only available to fee-paying users and partners.

Thus the license aims to strike a balance between the need to build a community and allow rapid, open development, while protecting the interests of the community's members, and encouraging users who are not ready to embrace the Open Source development model.

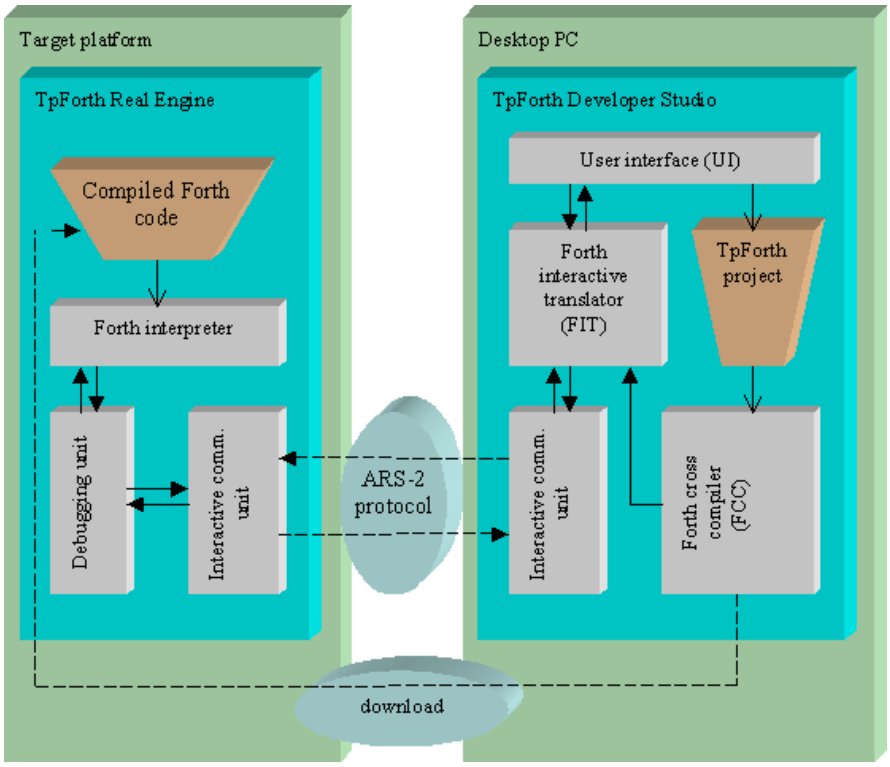

Figure 1: Internal organisation of the TpForth system

4 The TpForth system

The TpForth system is divided into two main programs: the TpForth Developer Studio (TDS), which is the IDE and runs under Microsoft Windows, and the TpForth Real Engine (TRE), which comprises the execution environment and runs on the host. The two systems communicate using the custom ARS-2 (Asynchronous Response System layer 2) protocol. The organisation of these two components is shown in figure 1.

The TRE can be implemented in two ways: as a standalone OS, or hooking into a native OS. Currently there are three implementations, for 8051, 16/32-bit Intel and MIPS. The 8051 implementation uses UniOP[1] as the underlying OS, whereas the 32-bit Intel system works emulates a standalone OS, though the TRE runs in Windows; this system can be used for simulation before running an application on the intended target.

The kernel of the TRE has two layers: the core, which is largely system independent, written in C, and needs only a few macros to be defined for context switching and stack access in each port, and the shell, written in C++, which interfaces to the OS or provides

[1] An OS for control panels; see `www.uniop.com`.

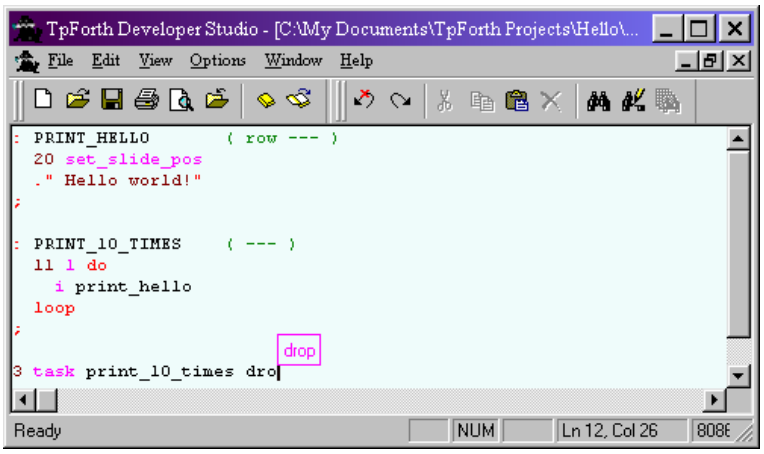

Figure 2: The TDS editor

OS services itself.

The TDS provides a Forth-oriented editor with syntax-colouring, automatic indentation, word-completion (for application words as well as built-ins), and pretty-printing; see figure 2 for an example. Project management facilities allow files to be grouped into projects, and a variety of information is recorded on a per-project basis, even down to breakpoints set for debugging. The source and low-level debugger allows interactive communication with running programs, and provides a variety of displays for the stacks and variables. I/O redirection allows the keyboard and screen of the development system to be used to communicate with the target, and low-level traces of compilation, execution and communication with the host show the data not captured by the higher-level displays.

All versions of the TRE are multithreaded, and the debugger is able to display the state of each task in a multithreaded program. In addition, some versions of the TRE allow multiple processes, though at the moment only one may be debugged at a time.

5 The community

Table 1 shows current members of the TpForth community. It is already widely distributed, and it is still expanding. Intesis have led development work on the TpForth system, while Xylon are developing a VHDL implementation of TRE's VM. All the companies mentioned use TpForth for develoment; NaviOP have written several specialised libraries, and Exor are using TpForth to expand the range of user interfaces that their equipment supports.

Company	Location	Area of expertise
Intesis	Croatia	Embedded systems software and firmware
Xylon	Croatia	Embedded systems hardware and firmware
Sitek	Italy	Industrial automation
NaviOP	Italy	Nautical navigation
Exor Group	Europe, USA, India	Industrial equipment interfaces

Table 1: Current members of the TpForth community

6 Future

TpForth is already a production-quality system, but there are wide-ranging plans for its improvement on a number of fronts:

TDS

"Visual Forth++" Usual OO IDE features to support object orientation (see "Language extensions" below), and GUI builder

CASE Visual representation and design of program flow

Version control Revision control and team development support

Profiling

TRE

Distribution Multiprocessor implementation using both shared and distributed memory models.

Hardware implementation

Native code generation

Language extensions There is nothing Forth users like better than to discuss extensions to the language. TpForth aims to move the discussion along a stage by implementing several "best practise" language extensions so they can be widely used, discussed and improved. Some areas in which this will be done are:

OOP The OOP model will be decided after a survey of current OO systems for Forth. It will support the same range of OO facilities as C++.

Emulation of other languages One of the biggest hurdles for Technopoint is to attract users of other languages. There are two solutions currently being considered: the first is to build a TpJava system, but this will require considerable resources and duplication of effort; the second is to provide a degree of language emulation for Java and C++ within TpForth. The aim

would be not so much to allow projects in those languages to be ported as to make it easy for programmers familiar with them to start using TpForth.

Component libraries

Graphics Support for many graphics and multimedia formats

Networking To allow embedded devices to be controlled and transmit using IP over the internet and intranets

SCADA

VHDL interpreter

TpForth already has a large user base, but Technopoint are working to expand it; recent initiatives include joining the Forth web ring, and this paper, and translations of the TpForth web site are planned.

7 Conclusion

Forth users form a community, but it is an "out of hours" community. We are good at discussing language and methods; indeed, since the ANSI standard this sort of discussion has become much more concrete and useful, through having a universal standard on which to base it.

However, everyday tools, the things we use to work, are not widely discussed or shared, neither across the chasm between free and commercial software, nor between the cells of the various proprietary systems, whether sold commercially or simply used internally.

TpForth is an opportunity for the Forth community to mature another stage, by reaching, if not agreement, then at least consensus on their tools as well as their language. Rather than endless "argument about it and about" TpForth will give them a common, freely available platform in which ideas can actually be put to the test. In addition, it aims to provide a framework for greater cooperation between the multitude of companies using Forth, allowing them better to advance the state of the art not just in their common language, but in the tools they share.