## GettingStarted

*Getting, compiling, loading, and testing PropellerForth*

Updated Jan 29, 2008 by cbiffle

# Introduction

PropellerForth can be compiled from scratch, or installed from a binary image.

The binary image is recommended. Even if you start with a fully compiled, ready-to-run system, you can still modify everything about the system, including the kernel and the core words.

Compiling PropellerForth from scratch is really only necessary for folks who want to do low-level assembly programming, or modify the kernel. Since PropellerForth currently lacks an online assembler, editing and assembling the kernel (using propasm) is the easiest way to hack.

Once you have PropellerForth running, you'll need a serial terminal emulator to talk to it -- such as `minicom` on Unix/Mac or HyperTerminal on Windows -- and a serial port. (USB-to-serial converters work great.) If you have one, an actual terminal will also work.

# Compiling PropellerForth

**Note for those skimming ahead:** Compiling PropellerForth is not required; in fact, use of a pre-built binary image is encouraged. Scroll down for details.

You will need:

- The sources (described below)
- propasm
- Java 5.0 or later (to run propasm)
- Ruby 1.8 (to run the build system)
- A GNU-compatible `make` to run the outermost Makefile. (The default `make` on Linux and Mac OS X is fine; BSD users might need to separately install `gmake`.)

1. Check out the sources from the Subversion repository, accessible via the Source tab at the top of this page. (TODO: more specific instructions, link.)
2. Change to the directory that the checkout created (probably called `propellerforth`)
3. Run `make`.

Some text will go by; the last line should resemble the following:

```
system.pa -> system.pa.binary, 8156 bytes (437ms)
```

The `system.pa.binary` file mentioned there is your new PropellerForth image.

# Loading PropellerForth

The binary images available on this site or produced by the build system are simply Propeller memory snapshots. They can be loaded like any other Propeller program. Most folks will probably want to load it one of two ways:

1. Using Propeller Tool from Windows
2. Using an SD bootloader like PropDOS.

## Loading PropellerForth from Propeller Tool

1. Move the PropellerForth image to a Windows machine, if it's not already there.
2. Start Propeller Tool.
3. Locate the image using Explorer.
4. Drag it into the Propeller Tool window. A dialog should appear with some colored hexadecimal text and buttons along the bottom.
5. Click 'Load RAM' to temporarily boot the image on the Propeller, or 'Load EEPROM' to program it into EEPROM.

## Loading PropellerForth from PropDOS

*Note:* due to a bug in PropDOS, PropellerForth does not load right away. I (Cliff Biffle) have a fix for PropDOS to enable PropellerForth to load, but it's not ready to be merged into the official PropDOS; contact me for details.

1. Load PropDOS on your Propeller (using your preferred method)
2. Copy the PropellerForth image onto an SD card. You may wish to give it a DOS-style 8.3 name for easier typing.
3. In PropDOS, enter `spin yourimage.bin`

## Saving PropellerForth to EEPROM

If you loaded PropellerForth from PropDOS or used the "Load RAM" option in PropellerTool, PropellerForth is running but will vanish if you reboot. If you decide to save PropellerForth to EEPROM, simply enter `savemem` and press return.

At any later date, you can use `savemem` to save any new words you've defined to EEPROM.

HYDRA users can insert a card, or swap cards, before `savemem` to save their image to a particular card.

# Communicating with PropellerForth

Once you've loaded PropellerForth onto your Propeller, you can communicate with PropellerForth using a serial terminal or terminal emulator. The required settings are

- 19200 bps
- 8 data bits
- No parity
- 1 stop bits
- No flow control

Set your terminal to use only CR (ASCII 13) as end-of-line, both when sending and receiving.

In addition, because the PropellerForth serial code is flaky, it may be helpful to set a line and character delay of roughly 200ms / 50ms, respectively. This will allow you to paste large text files into PropellerForth; the delay is required because the current serial I/O is unbuffered, so PropellerForth may miss characters while it's interpreting or compiling your text. We'll fix this in a later release.

Once your terminal is enabled, hit Enter a few times. PropellerForth should respond with `ok`.

# Testing PropellerForth

A full Forth tutorial is out of the scope of this page, but here are some simple expressions to test PropellerForth's functionality.

In all the Forth examples below, spaces are *very* important! Forth uses spaces in places you might not expect, and if you omit them, it can completely change the meaning of the code.

## Arithmetic

Type

```
2 2 + .
```

and press Enter. PropellerForth should respond

```
4
```

## Hello!

Type

```
.( Hello, world!)
```

(space(s) between ( and H) and press Enter. PropellerForth should greet you.

## Compiling a new word (Propeller Demo Board only)

Type

```
: LEDemit
  255 and  16 lshift  outa L! ;
```

and press Enter. PropellerForth should respond with `ok`. We've just defined a new word, called `LEDemit`, which sets the demo board LEDs. Let's try it:

```
255 ledemit
```

(All LEDs should be on now.)

Note that PropellerForth isn't case-sensitive: we called the word `LEDemit` when we defined it, but called it as `ledemit` without any trouble.

```
0 ledemit
```

(All LEDs should be off.)

---

Comment by covington.jerry, Mar 15, 2009

I can't get any kind of usable results with this because none of the termnals I have tried will work. I remember fighting this once before. Prop Terminal gives the best results but will not generate line feeds so all input and output show up on the same (top) line. This causes obvious problems. HyperTerm? will generate line feeds and will successfully

connect to the prop but Prop Forth prints the sign on message and then never returns any other visible result. I tried playing with various settings (not the hook up stuff, that works) but to no avail. If you have any suggestions I'm at covington(dot)jerry(at)gmail(dot)com.

Comment by covington.jerry, Mar 15, 2009

Hey gents, its me again. Belay my last! I restarted HyperTerm? and for some reason known only to God, HyperTerm? and the prop, it's working now. Go figure! JAC covington(dot)jerry(at)gmail(dot)com.

Comment by cbiffle, Mar 15, 2009

Unfortunately I can't provide any support for terminal emulators on Windows. On Unix systems I use Minicom. It doesn't look like HyperTerminal?'s been updated significantly since Win95; presumably Windows ships with a more modern terminal emulator in recent versions.

Comment by peter.fi...@gbconsite.de, Sep 22, 2009

newer PuTTY versions can use serial lines

Comment by 2008RN2B, Oct 04, 2009

Is there a glossary for Propeller Forth? The only wan I can figure out what words do what is by looking at the eeprom dump and experimenting. Not very efficient! The source code would help.

► Sign in to add a comment