Spry Web Browser

Project Proposal

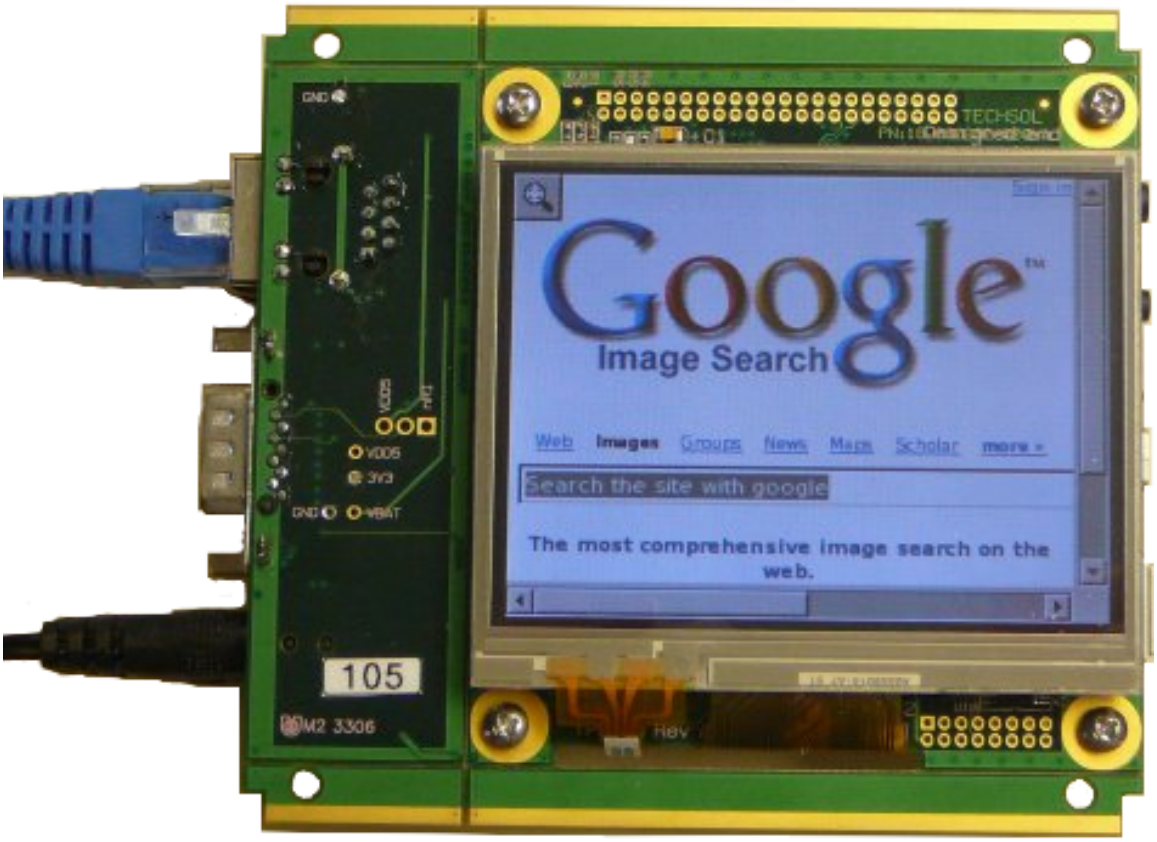
Project Manager:
Doug Penner

Contributors:
Brendan Neva
Steffen L. Norgren
Eddie Zhang

COMP 4908 • BCIT • May 4, 2009

Introduction & Background

TechSol is a local company that focuses on the embedded devices based on ARM technology running Linux. This allows them to develop low-cost solutions for a variety of environments, ranging from industrial to residential applications.



In order to speed up application development as well as facilitate maximum portability, Techsol has been developing simple web-applications to act as control interfaces for anything from thermostat control to industrial control.

The challenge involved with this solution using Linux on ARM-based devices is that all of the web browsers that currently exist for the platform are somewhat ill-suited. All current Linux web browsers are all either too slow, too clunky, or simply do not respond well to a touch-screen environment.

Purpose & Objectives

TechSol contracted this project in order to solve the problem of there currently being no suitable web browsing software that fits the needs of their touch-screen web applications. As such, by developing a custom web browser to serve their web applications on ARM devices running Linux, they are hoping to bypass the technical limitations encountered with current Linux web browsing software.

The major objective of this project is to deliver a fast and stable web browser with a low memory footprint that is based on the WebKit and Gtk+ frameworks. The WebKit framework was chosen because it has a great deal of momentum behind the project and is currently the fastest rendering engine available. Additionally, Gtk+ was chosen as the GUI framework simply because it is far less resource intensive than Qt, is fully open source and has a great deal of support for developing applications in C, TechSol's preferred programming language.

Deliverables

This project is to make use of the WebKit application framework along with Gtk+ to create a minimal web browser written in C. This web browser must be able to properly process XHTML, CSS, and JavaScript.

Additionally, time will be spent performing application benchmarking in order to ascertain where certain inefficiencies and correcting them. If these inefficiencies lie within the WebKit or Gtk+ frameworks themselves, a bug report should be submitted upstream and, if possible, an attempt to correct the inefficiency should be made.

The web browser must have the following features:

- fast
- lean (small memory footprint)
- minimal GUI elements
- optimized for touch screens (no right-click or scroll wheels)
- intuitive
- optimized for small displays (320x240), but usable on large displays
- completely command line configurable.

Skills Involved

This project will require the following skills:

- code optimization
- HCI (Human Computer Interface) Design
- bug tracking
- upstream bug reporting
- code research (tracing existing framework code)
- and benchmarking

Possible Complications

There are several foreseeable complications involved with this project, however, they all stem from existing bugs and code from the WebKit and Gtk+ frameworks. For instance, a major performance bottleneck may be found through the course of benchmarking the application, but it may lie within the WebKit or Gtk+ frameworks themselves and not the code developed for the project.

It is for these types of complications that we are to submit bug reports upstream to the WebKit or Gtk+ projects. If we find a solution to these issues, we are to create a patch to the problem and submit that along with the bug report.

However, in the situation where the problem is beyond our abilities to correct, we may be forced to accept the issue until it has been corrected upstream.

Responsibilities & Co-ordination

Currently, the project will be split up into the following tasks and responsibilities:

- Initial Prototype (Doug, Steffen, Eddie, Brendan)
- Source code structuring (Steffen)
- Command Line Arguments (Doug, Steffen)
- Graphics – icons, colours (Eddie)
- WebKit Research (Steffen)
- GTK Research (Doug)
- Fullscreen Support (Doug)
- Context Menu (Doug)
- Documentation formatting and checking (Eddie)
- Benchmarking (Brendan)
- Disable Highlighting (Doug, Steffen, Eddie Brendan)