GENIVI GSoC 2017 - Zebra (zebkit)

Overview

The GENIVI Google Summer of Code 2017 project is using this space to document its work. The project aims to bring into the GDP a JavaScript "canvas" library of UI elements. The goal is to show a possible approach to developing JavaScript based UI for the GDP and automotive IVI systems. The tool chosen for this work is work is called Zebra.

The GENIVI Development Platform is an open source platform for building, testing, and deploying InVehicle Infotainment software components on hardware targets. This GSoC project will be an experiment in building and deploying sample user interface components for the GDP, based on HTML5 rather than Qt.

Goals

- Determine the path to integration of an HTML framework into GENIVI's GDP
- Demonstrate use of GENIVI's APIs and software via HTML5

Rationale

- HTML5 is a popular option for developers and HTML is widely supported
- HTML5 is an open standard managed by the W3C and there are multiple open source implementations

Setup zebra on target

The target in this case is qemux86-64 which is an Intel x86 based, emulated system.

These are the steps to include zebkit recipes to gdp and build the target:

1. git clone https://github.com/akilawickey/genivi-dev-platform.git
2. cd genivi-dev-platform
3. source init.sh qemux86-64
4. bitbake genivi-dev-platform

Testing

The first thing to do when using Yocto to build an image is to find amongst all the various build artifacts the actual image you want to run. Yocto keeps that in a directory called "`images`" which should be under the directory you built in, i.e. "`../genivi-dev-platform/gdp-src-build/tmp/deploy/images/`" You'll see the "qemux86-64" directory and your finished image and a bunch of other stuff there. bitbake has environment variables that should find your built image immediately so you can run;

```
$ runqemu qemux86
```

Barring any issues, this should result in a message stating that a Virtual Network Computer (VNC) Server has started on your machine and you can access it with a VNC viewer;

```
VNC server running on 127.0.0.1:5900
```

You should be able now to see screen shots like this;