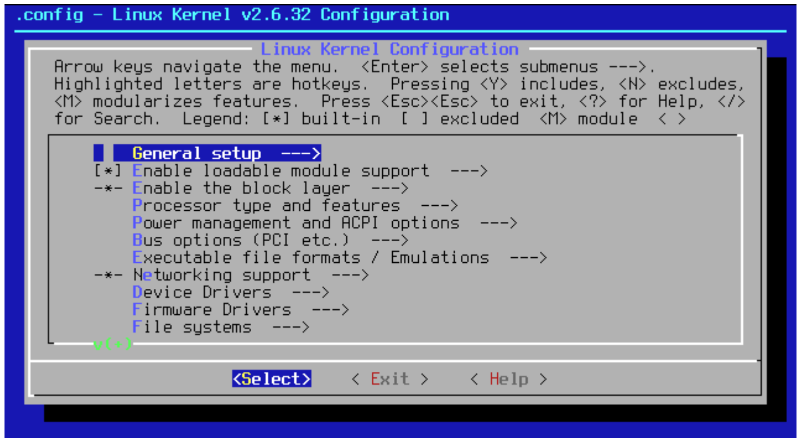
04-4 Graphical Front End via node.js

How to add a pretty face via a web browser

Ways to Add Graphics

- ncurses (http://www.gnu.org/software/ncurses/)
- X Window System (http://www.x.org/wiki/)
- Qt (http://qt.digia.com/) both X-based and embedded
- Web server
  - node.js (http://nodejs.org/)

ncurses

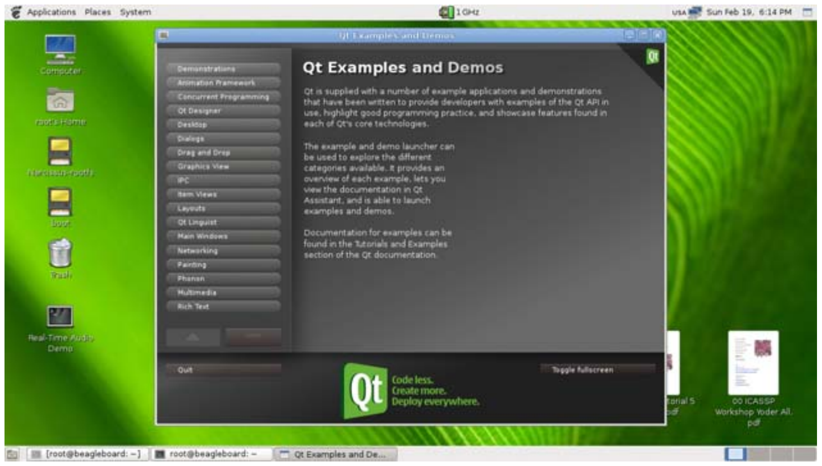


```
.config - Linux Kernel v2.6.32 Configuration
                    Linux Kernel Configuration
  Arrow keys navigate the menu.  <Enter> selects submenus --->.
  Highlighted letters are hotkeys.  Pressing <Y> includes, <N> excludes,
  <M> modularizes features.  Press <Esc><Esc> to exit, <?> for Help, </>
  for Search.  Legend: [*] built-in  [ ] excluded  <M> module  < >

          General setup  --->
      [*] Enable loadable module support  --->
      -*- Enable the block layer  --->
          Processor type and features  --->
          Power management and ACPI options  --->
          Bus options (PCI etc.)  --->
          Executable file formats / Emulations  --->
      -*- Networking support  --->
          Device Drivers  --->
          Firmware Drivers  --->
          File systems  --->
      v(+)

                <Select>    < Exit >    < Help >
```

http://en.wikipedia.org/wiki/File:Linux-menuconfig.png

X Windows

Qt – Both X and embedded



Qt Examples and Demos

Qt is supplied with a number of example applications and demonstrations that have been written to provide developers with examples of the Qt API in use, highlight good programming practice, and showcase features found in each of Qt's core technologies.

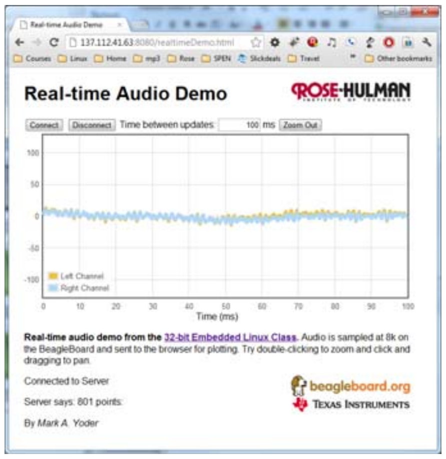
The example and demo launcher can be used to explore the different categories available. It provides an overview of each example, lets you view the documentation in Qt Assistant, and is able to launch examples and demos.

Documentation for examples can be found in the Tutorials and Examples section of the Qt documentation.

```
$ opkg install qt4-demos
$ qtdemo
```

http://elinux.org/ECE497_Notes_on_Qt

Via the Web via node.js



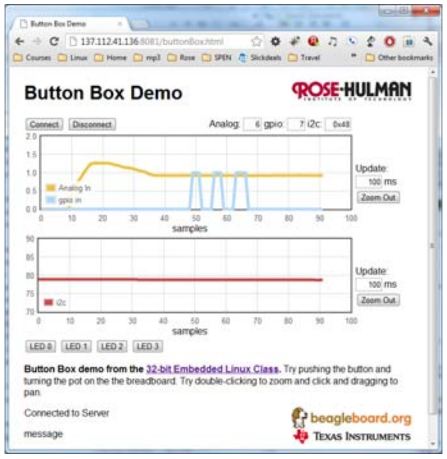
Real-time Audio Demo

Real-time audio demo from the 32-bit Embedded Linux Class. Audio is sampled at 8k on the BeagleBoard and sent to the browser for plotting. Try double-clicking to zoom and click and dragging to pan.

Connected to Server

Server says: 801 points

By *Mark A. Yoder*



Button Box Demo

Button Box demo from the 32-bit Embedded Linux Class. Try pushing the button and turning the pot on the the breadboard. Try double-clicking to zoom and click and dragging to pan.

Connected to Server

message

```
beagle$ cd ~/exercises/realtime
beagle$ node boneServer.js
```

node.js

- Platform built on Chrome's JavaScript runtime for easily building fast, scalable network applications.
- Uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices.
- Programmed in JavaScript on both server and client.

http://nodejs.org/

node.js example: Webserver

- This simple web server written in Node responds with "Hello World" for every request.

```
var http = require('http');
http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/plain'});
  res.end('Hello World\n');
}).listen(1337);
console.log('Server running on port 1337');
```

- To run the server, put the code into a file example.js and execute it with the node program:

```
$ node example.js
Server running on port 1337
```

Things to know

- JavaScript
  - socket.io
  - jQuery
  - DOM
- html
- CSS

- Where to you start?

Javascript – C-like

```
#include <stdio.h>
main() {
    int i;
    for(i=0; i<5; i++) {
       printf("i=%d\n", i);
    }
}
```

```
var i;
for(var i=0; i<5; i++) {
    console.log("i=%d", i);
}
```

JavaScript in 10 minutes

- By Spencer Tipping
- https://github.com/spencertipping/js-in-ten-minutes
- 27 pages
- Here are the highlights…

JS - Types

- **Strings** – e.g. 'foo', "foo" (single vs. double quotation – no difference)
- **Numbers** – e.g. 5, 3e+10 (all numbers behave as floats)
- **Booleans** – true and false.
- **Arrays** – e.g. **[**1, 2, "foo", [3, 4]**]**.
- **Objects** – e.g. {foo**:** 'bar', bif**:** [1, 2]}, which are really just hashtables.
- **Functions** – e.g. var example=function (x) {return x + 1}.

JS - Functions

- Functions are first-class lexical closures

```
var f = function () { // f is toplevel, so global
    var x = 5;        // x is local to f
    y = 6;            // y is global
};
```

- Watch out

```
var f = function () { // f is toplevel, so global
    y = 6;            // y is global
    x = 42;
Do stuff…
    var x = 5;        // x is local to f
};
```

JS - Semicolon

- Javascript doesn't require a semicolon at the end of each line, but you should anyway.

```
var x = f
(y = x) (5)
```

- Is treated as:

```
var x = f(y = x) (5)
```

- You probably meant

```
var x = f;
(y = x) (5);
```

JS - Equality

- Never use `==` or `!=`
- Always use `===` or `!==`
- All these are `true`:

```
null  == undefined
null  == 0
false == ''
''    == 0
true  == 1
true  == '1'
```

JavaScript: The Good Parts

- Intended for programmers who, by happenstance or curiosity, are venturing into JavaScript.
- Also intended for programmers who have been working with JavaScript at a novice level and are now ready for a more sophisticated relationship with the language.
- Most programming languages contain **good parts and bad parts**. I discovered that I could be a better programmer by using only the good parts and avoiding the bad parts.
- JavaScript is a language with more than its share of bad parts.
- 172 pages

Things to know

- JavaScript
  - socket.io
  - jQuery
  - DOM
- html
- CSS

socket.io

- http://socket.io/
- **Socket.IO** aims to make realtime apps possible in every browser and mobile device, blurring the differences between the different transport mechanisms. It's care-free realtime 100% in JavaScript.

socket.io

- Server

```
var io = require('socket.io').listen(80);
io.sockets.on('connection', function (socket) {
  socket.emit('news', { hello: 'world' });
  socket.on('my other event', function (data) {
    console.log(data);
  });
});
```

- Client - Browser

```
<script>
var socket = io.connect('http://localhost');
socket.on('news', function (data) {
  console.log(data);
  socket.emit('my other event', { my: 'data' });
});
</script>
```

socket.io

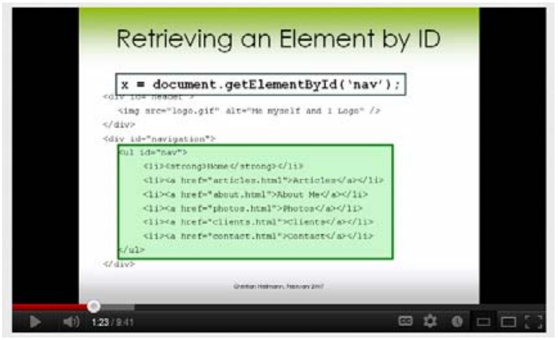
- See: **Getting Started With node.js and socket.io (v0.7+) – Part 2**
- http://codehenge.net/blog/2011/12/getting-started-with-node-js-and-socket-io-v0-7-part-2/
- My code is based on this

Things to know

- JavaScript
  - socket.io
  - DOM
  - jQuery
- html
- CSS

DOM

- **Essentials of the DOM and JavaScript in 10 Minutes**
- http://www.youtube.com/watch?v=URF2sVQWuxU
- **10 minute YouTube video**
- **However we'll use jQuery, it's much more compact**

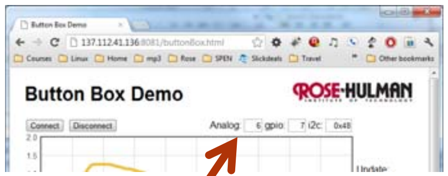

Retrieving an Element by ID

```
x = document.getElementById('nav');
<div id="header">
  <img src="logo.gif" alt="Me myself and i Logo" />
</div>
<div id="navigation">
  <ul id="nav">
     <li><strong>Home</strong></li>
     <li><a href="articles.html">Articles</a></li>
     <li><a href="about.html">About Me</a></li>
     <li><a href="photos.html">Photos</a></li>
     <li><a href="clients.html">Clients</a></li>
     <li><a href="contact.html">Contact</a></li>
  </ul>
</div>
```

Things to know

- JavaScript
  - socket.io
  - DOM
  - jQuery
- html
- CSS

jQuery



- http://jquery.org/
- jQuery is a fast and concise JavaScript Library that simplifies HTML document traversing, event handling, animating, and Ajax interactions for rapid web development. jQuery is designed to change the way that you write JavaScript
- Looks like **`$()`** in html

```
Analog: <input id="ainNum" type="text" value=""
style="text-align: right; width:2em">
```

- **In JavaScript**

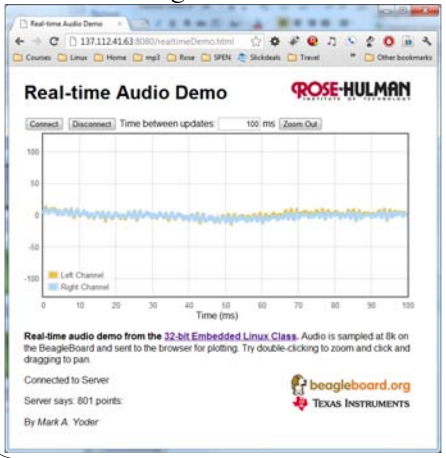
```
$("#ainNum").val(ainNum).change(function () {
        ainNum = $(this).val();
    });
```

Things to know

- JavaScript
  - socket.io
  - DOM
  - jQuery
- html
- CSS
- FLOT
- http://www.flotcharts.org/
- Flot is a pure JavaScript plotting library for jQuery, with a focus on simple usage, attractive looks and interactive features.

To Do

- Look at **`~/exercises/realtime`** and see what you can figure out.



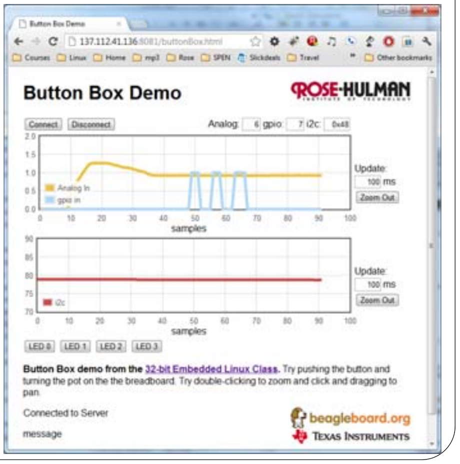
Real-time Audio Demo

Real-time audio demo from the 32-bit Embedded Linux Class. Audio is sampled at 8k on the BeagleBoard and sent to the browser for plotting. Try double-clicking to zoom and click and dragging to pan.

Connected to Server

Server says: 801 points.

By *Mark A. Yoder*



Button Box Demo

Button Box demo from the 32-bit Embedded Linux Class. Try pushing the button and turning the pot on the the breadboard. Try double-clicking to zoom and click and dragging to pan.

Connected to Server

message