

2015 Spring, UCSC extension, Designing with Xilinx FPGAs

Final Project Report

LED control via WEB
Using Zybo board, Xilinx, and node.js

Jae Yang Park

jaeyangp@gmail.com

Objective

This project is to implement Embedded system based on the learning from the class using Xilinx Zynq based board, Zybo, Ubuntu based Xillinux, and node.js. With implemented design LEDs connected on the GPIO can be controlled through Internet Web browser from the remote location. This concept can be applied for the Internet of Things (IoT) technologies.

Internet of Things (IoT)

The **Internet of Things (IoT)**, sometimes **Internet of Everything** is the network of physical objects or "things" embedded with electronics, software, sensors and connectivity to enable it to achieve greater value and service by exchanging data with the manufacturer, operator and/or other connected devices. Each thing is uniquely identifiable through its embedded computing system but is able to interoperate within the existing Internet infrastructure. (Wikipedia.org)

Project Environment

Hardware: Digilent ZYBO board (Xilinx Zynq XC7Z010-1CLG400C)

Operating System: Xillinux-1.3 (Ubuntu based, Kernel 3.12.0)

Node.js: v0.10.28 (pre-compiled for ARM)

In the beginning, Xilinx's Petalinux was considered and tested, but several necessary packages were not included, and rootfs was created into the ramdisk. So, after power off, all working data were erased.

Next, Xillinux was selected. It has almost all necessary packages like gcc toolchains, package manager, and so on. And, the rootfs of Xillinux can be accessed from Linux environment, and it is very convenient to file transfer under no network environment.

The most important thing is the node.js in this project. It should be compiled for ARM, but its compiling was not successful due to ARM toolchain version and other. So, pre-compiled node.js was used. Its version is lower than current one because current version of npm was not working on the project environment.

Xillinux install

1. Partitioning 4GB of Micro SD card: ZYBO_BOOT (1 GB, FAT) and ROOT_FS (EXT4)
2. Download Xillinux from <http://xillybus.com/downloads/xillinux-1.3.img.gz>
3. Extract the archive and double click on xillinux-1.3.img file. It will be mounted automatically
4. Change label 16MB volume to ZYBO_BOOT (it contains uImage)
5. Change label 1.8GB volume to ROOT_FS
6. Download the boot files from <http://xillybus.com/downloads/xillinux-eval-zybo-1.3b.zip>
7. Copy the files (boot.bin, devicetree.dtb) from boot folder into ZYBO_BOOT
8. Download the xillydemo.bit from <http://www.instructables.com/files/orig/F95/7F9Y/I8J9X7OE/F957F9YI8J9X7OE.bit>
9. Copy xillydemo.bit into the ZYBO_BOOT

```
ZYBO_BOOT
boot.bin, devicetree.dtb, uImage, xillydemo.bit
```

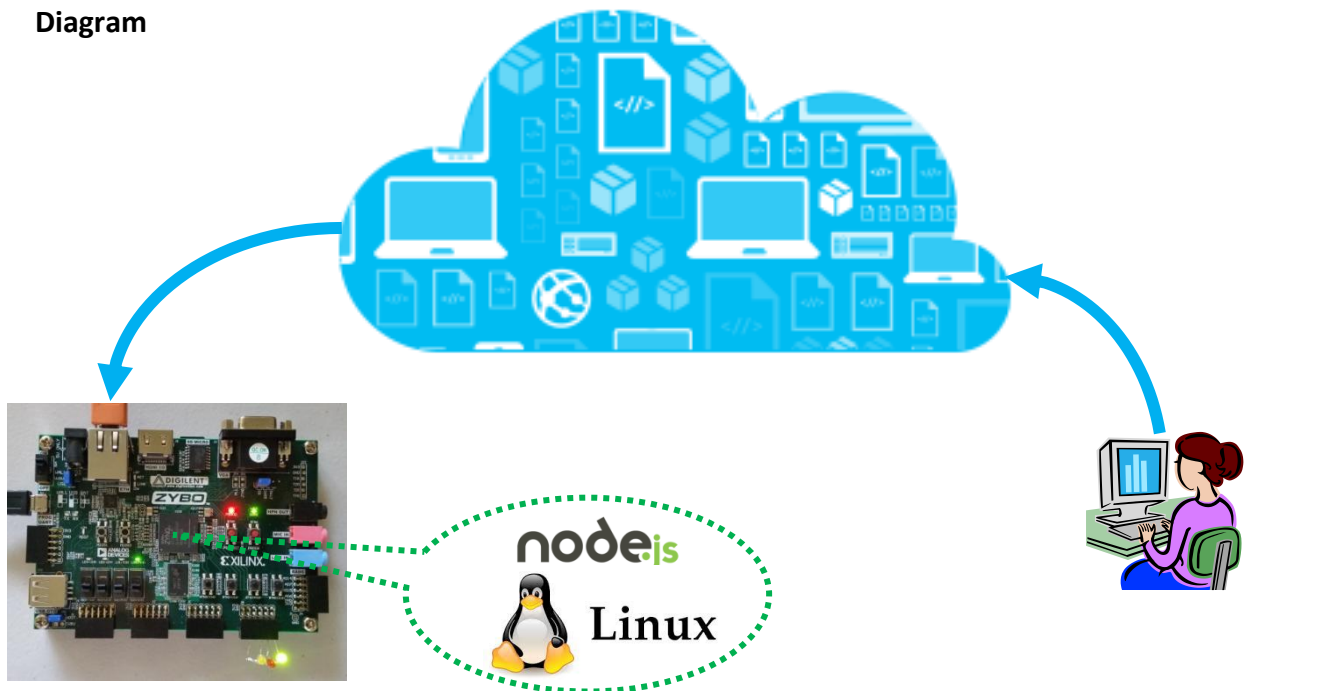
```
ROOT_FS: Linux file system
```

node.js install (pre-compiled for ARM)

```
$ wget http://nodejs.org/dist/v0.10.28/node-v0.10.28-linux-arm-pi.tar.gz
$ tar -xzf node-v0.10.28-linux-arm-pi.tar.gz
$ node-v0.10.28-linux-arm-pi/bin/node --version
v0.10.28
```

In order to use node.js, Internet can be accessible due to module downloading.
Current project environment, this version was verified, and working correctly.

Diagram



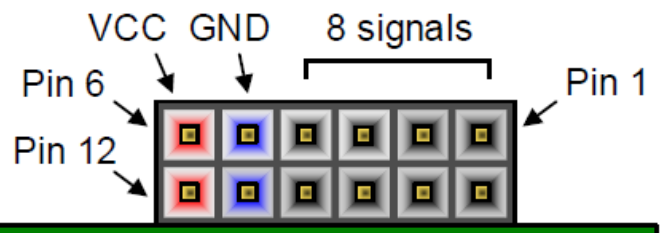
PMOD JB

LEDs are connected to JB;

GREEN: Pin 1 (PS_GPIO[32])

RED: Pin 2 (PS_GPIO[33])

YELLOW: Pin 3 (PS_GPIO[34])



The offset between PS_GPIO and those designated by Linux is 54. So these pins are :

JB-1 PS_GPIO[32] = GPIO 86

JB-2 PS_GPIO[33] = GPIO 87

JB-3 PS_GPIO[34] = GPIO 88

Reference: <http://billauer.co.il/blog/2014/07/bash-gpio-xlinux/>

Project creation

```
$ mkdir led_socket
$ cd led_socket
$ npm init
$ npm install onoff
$ npm install socket.io
$ npm install ...
```

Source Code

package.json

```
{
  "name": "led_socket",
  "version": "0.0.1",
  "description": "LED on/off via Web",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "Jae Yang Park",
  "license": "ISC"
}
```

server.js

```
// UCSC extension
// Designing with Xilinx FPGA
// Final project
// Zybo and node.js
// LED on/off via Web
//
// node.js modules: socket.io, onoff
//
// Jae Yang Park (jaeyangp@gmail.com)
//

var app = require('http').createServer(handler);
var io = require('socket.io').listen(app);
var fs = require('fs');
var sock;

app.listen(8001, function() {
  console.log('listening on *.8001');
});

// ZYBO PMOD JB 1: 86, 2: 87, 3: 88,..
var GPIO = require('onoff').Gpio;
var led1 = new GPIO(86, 'out');
var led2 = new GPIO(87, 'out');
var led3 = new GPIO(88, 'out');

//Init LED to off
led1.writeSync(0);
led2.writeSync(0);
led3.writeSync(0);

function handler (req, res) {
  fs.readFile('index.html',
    function (err, data) {
      if (err) {
        res.writeHead(500);
        return res.end('Error loading index.html');
      }
      res.writeHead(200);
      res.end(data);
    }
  );
}
```

```

    });
}

io.on('connection', function (socket) {
    sock = socket;
    socket.on('led1', function (data) {
        console.log(data);

        if (data == 'on'){
            led1.writeSync(1);
            socket.emit('ledstatus', 'green');
        }else{
            led1.writeSync(0);
            socket.emit('ledstatus', 'white');
        }
    });

    socket.on('led2', function (data) {
        console.log(data);
        if (data == 'on'){
            led2.writeSync(1);
            socket.emit('ledstatus', 'red');
        }else{
            led2.writeSync(0);
            socket.emit('ledstatus', 'white');
        }
    });

    socket.on('led3', function (data) {
        console.log(data);
        if (data == 'on'){
            led3.writeSync(1);
            socket.emit('ledstatus', 'yellow');
        }else{
            led3.writeSync(0);
            socket.emit('ledstatus', 'white');
        }
    });
});
});

```

Index.html

```

<!DOCTYPE html>
<html>
<head>
<title>Designing with Xilinx FPGAs, Comprehensive</title>
<script src = "/socket.io/socket.io.js" > </script>
<script type=text/javascript src="http://code.jquery.com/jquery-1.7.1.min.js"></script>
<script>
    var socket = io.connect();
    socket.on('ledstatus', function (data) {
        console.log(data);
        $("body").css("background-color", data);
    });
</script>
<script>
function led1On(){
    socket.emit('led1', 'on');
}
function led1Off(){
    socket.emit('led1', 'off');
}
function led2On(){
    socket.emit('led2', 'on');
}
function led2Off(){
    socket.emit('led2', 'off');
}

```

```

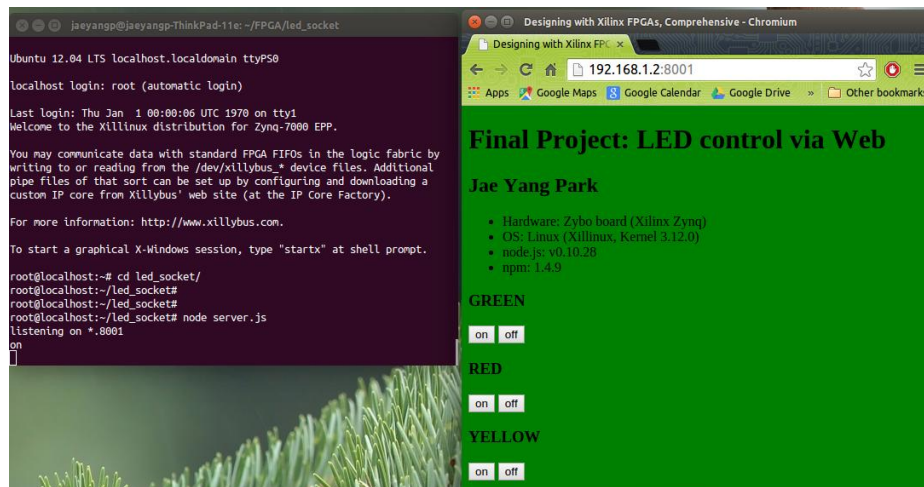
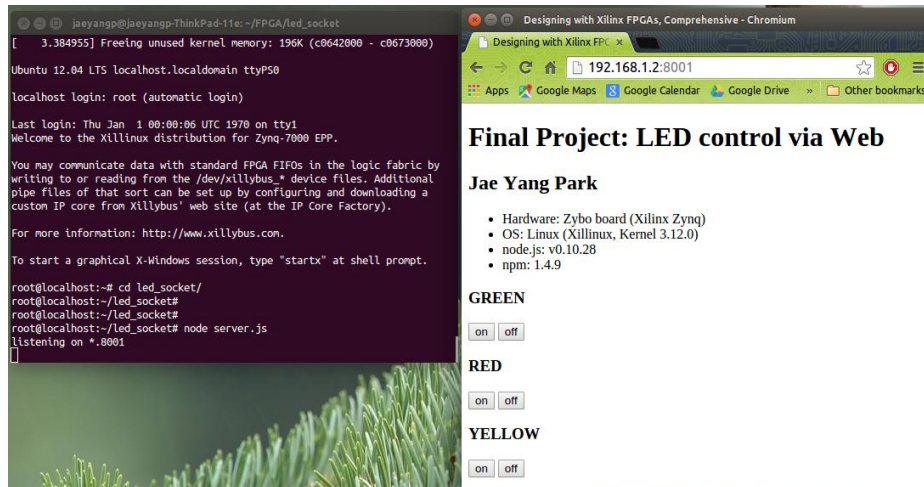
function led3On(){
    socket.emit('led3', 'on');
}
function led3Off(){
    socket.emit('led3', 'off');
}
</script>
</head>

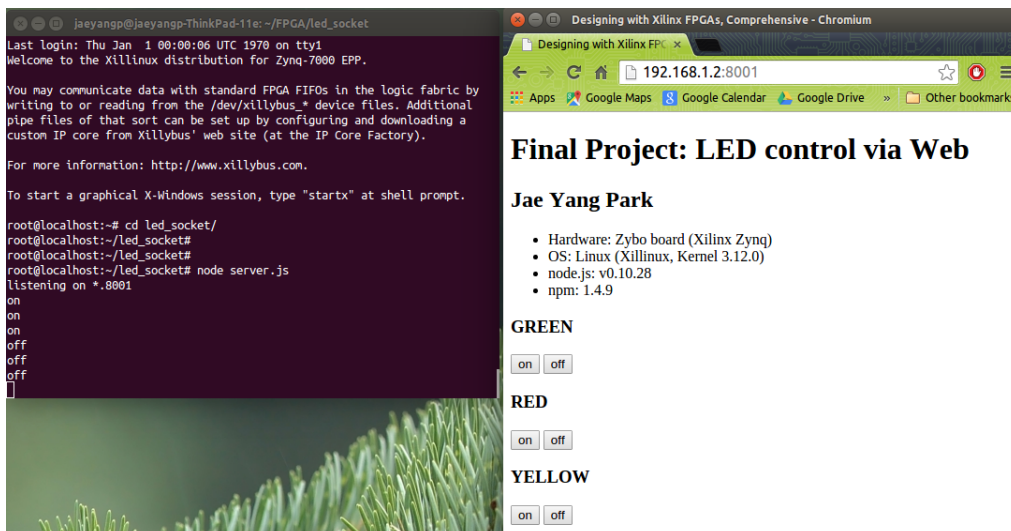
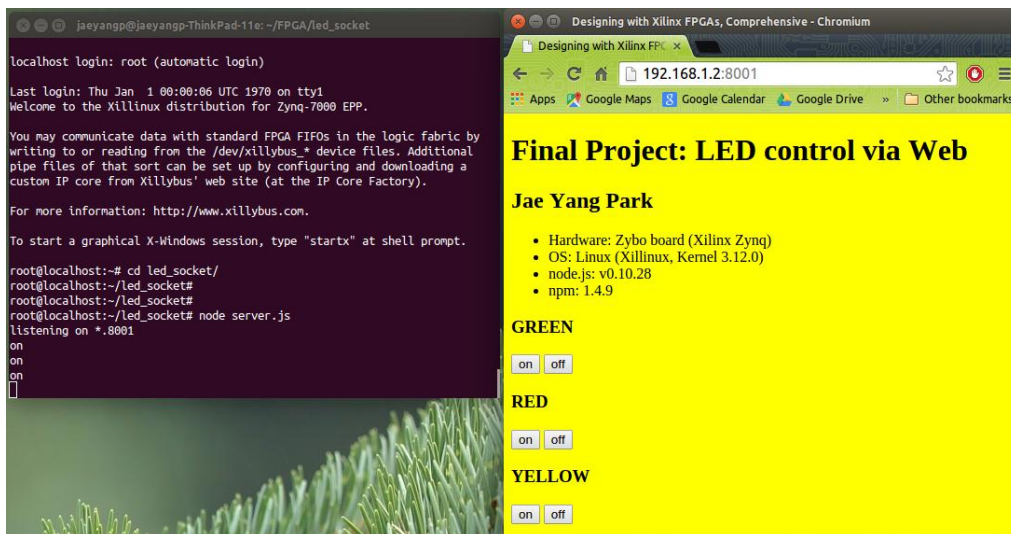
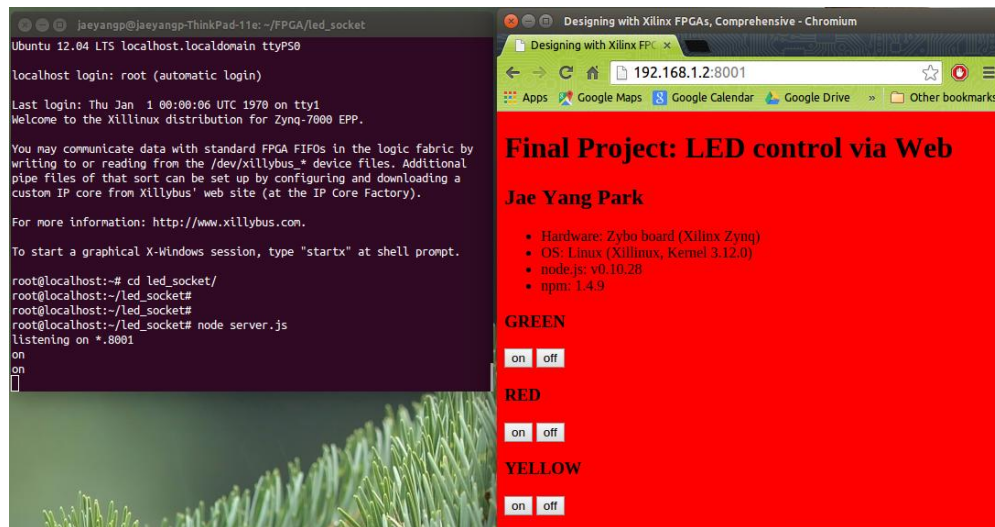
<body>
<h1>Final Project: LED control via Web</h1>
<h2>Jae Yang Park </h2>
<ul>
<li>Hardware: Zybo board (Xilinx Zynq)</li>
<li>OS: Linux (Xillinux, Kernel 3.12.0)</li>
<li>node.js: v0.10.28</li>
<li>npm: 1.4.9</li>
</ul>
<h3>GREEN</h3>
<input type="button" name="on" id="onButton" value="on" onClick="led1On();">
<input type="button" name="off" id="offButton" value="off" onClick="led1Off();">
<br>
<h3>RED</h3>
<input type="button" name="on" id="onButton" value="on" onClick="led2On();">
<input type="button" name="off" id="offButton" value="off" onClick="led2Off();">
<br>
<h3>YELLOW</h3>
<input type="button" name="on" id="onButton" value="on" onClick="led3On();">
<input type="button" name="off" id="offButton" value="off" onClick="led3Off();">
</body>
</html>

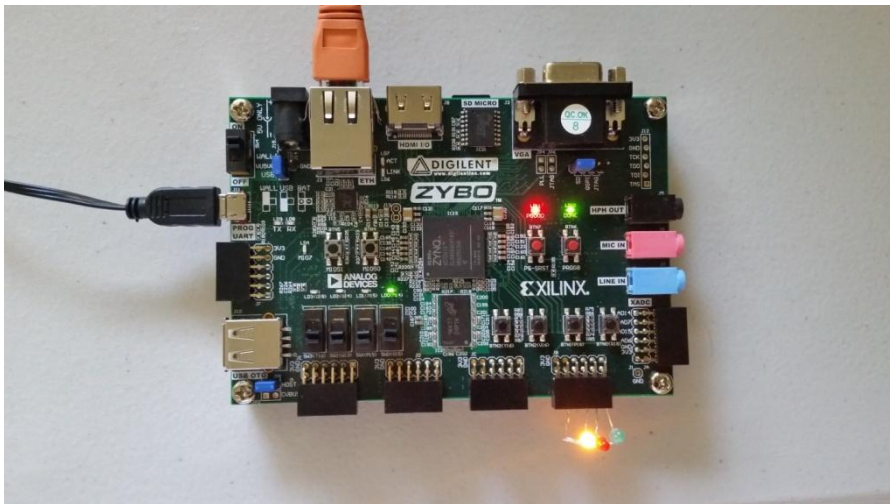
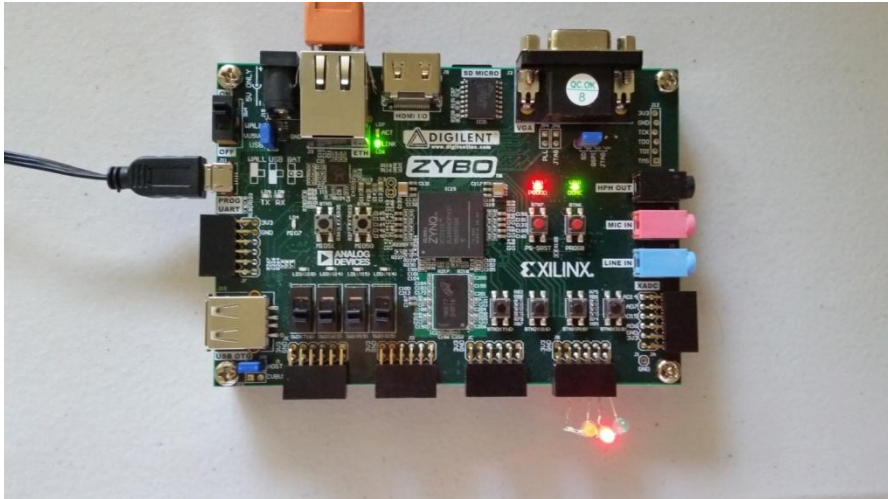
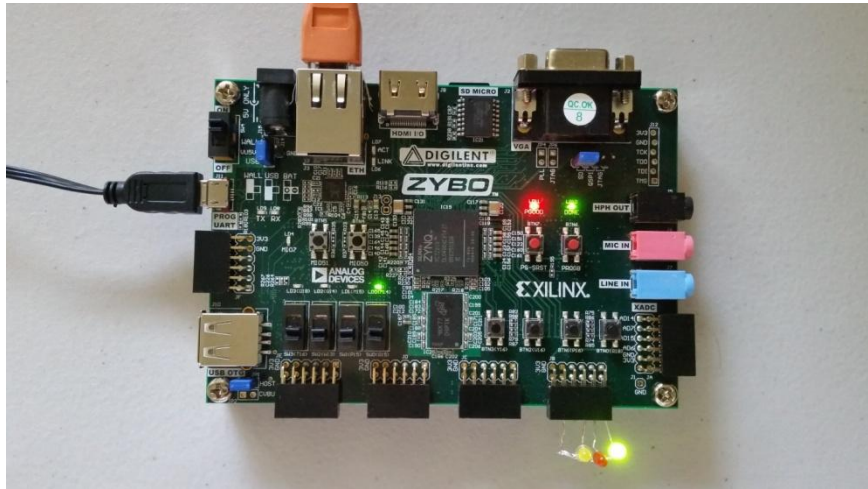
```

Pictures

```
jaeyangp@jaeyangp-ThinkPad-11e: ~/FPGA/led_socket
root@localhost:~/led_socket# node server.js
listening on *.8001
```







Further works

For this project, pre-built hardware, boot image and root file images. However, pre-built environment is not optimized development environment. There were following difficulties during this project:

- Less features of pre-built OS
- New driver installation and compiling
- Cross compiling for packages
- Attaching USB WIFI adapter

In order for customized develop environment, building cross toolchains and kernel is required. Without this work, valuable time, schedule, is spent with no result.

For future work, building toolchain, creation of customized Linux image and boot image, compiling device driver, and attaching wifi dongle which is Linux compatible, and updated version of node.js compiling are required.

Conclusion

Through this project I realized the application of node.js is a lot for Internet of Things, embedded system area. It is easy to learn and good for embedded systems.

I've learned many topics through this project, conversation and lectures in the class.

References

1. <http://www.instructables.com/id/Installing-Xillinux-on-ZYbo/>
2. <https://nodejs.org/>
3. <https://www.npmjs.com/>
4. <http://billauer.co.il/blog/2014/07/bash-gpio-xillinux/>