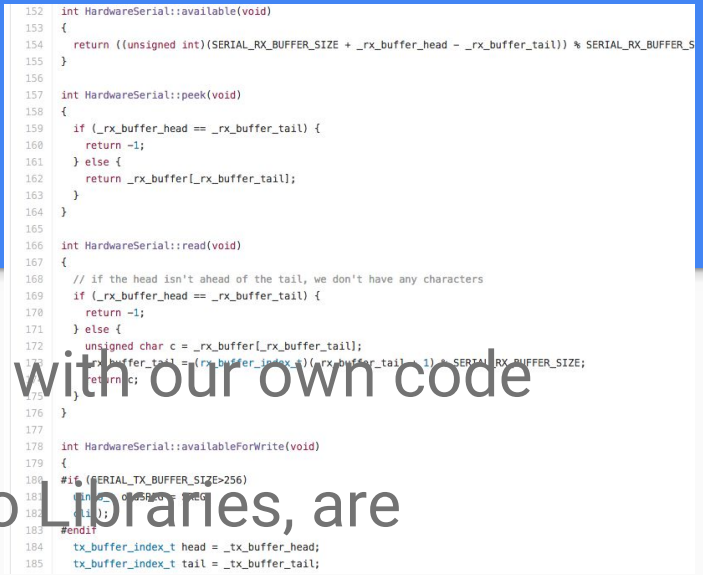
UCLA Rocket Project Electronics Workshop 4

2016/10/28

Next week - Arduino libraries



```
int HardwareSerial::available(void)
{
  return ((unsigned int)(SERIAL_RX_BUFFER_SIZE + _rx_buffer_head - _rx_buffer_tail)) % SERIAL_RX_BUFFER_S
}

int HardwareSerial::peek(void)
{
  if (_rx_buffer_head == _rx_buffer_tail) {
    return -1;
  } else {
    return _rx_buffer[_rx_buffer_tail];
  }
}

int HardwareSerial::read(void)
{
  // if the head isn't ahead of the tail, we don't have any characters
  if (_rx_buffer_head == _rx_buffer_tail) {
    return -1;
  } else {
    unsigned char c = _rx_buffer[_rx_buffer_tail];
    _rx_buffer_tail = (rx_buffer_index_t)(_rx_buffer_tail + 1) % SERIAL_RX_BUFFER_SIZE;
    return c;
  }
}

int HardwareSerial::availableForWrite(void)
{
#if (SERIAL_TX_BUFFER_SIZE>256)
  uint8_t oldSREG = SREG;
  cli();
#endif
  tx_buffer_index_t head = _tx_buffer_head;
  tx_buffer_index_t tail = _tx_buffer_tail;
```

More generally: using external libraries with our own code

1. External libraries, especially Arduino Libraries, are exceptionally well written
2. Unnecessary to write drivers for many hardware devices, it only has to be done once

Node.js

1. Short introduction to bash
2. Short introduction to javascript
3. Node.js - description
4. Quick html document
5. Server code
6. Server setup

Short introduction to bash

1. Bash is a Unix shell - a command language, computer language used to control the operating system
2. Getting to bash:
   a. On Mac: <Cmd> + <Space>; terminal; <Enter>
   b. On Linux: (depends) search for Konsole, LXterminal, Terminal

Short introduction to bash - Windows 10

1. Bash available on most Windows OSs through Cygwin - a rather large program, installation takes a little time
2. Bash available on Windows 10 64-bit:
   a. Settings app -> Updates & Security -> For Developers; activate the developer mode
   b. Control Panel -> Programs -> Turn Windows Features On or Off; check Windows Subsystem for Linux (Beta)
   c. Restart; use Cortana to find Bash

Short introduction to bash - simple commands

1. **pwd** - print current directory; shows where you are
2. **cd** *directory\ name* - change directory; changes directory, relative or absolute
3. **ls** - list; lists all items in the current directory or in the directory provided as an argument

Short introduction javascript

1. Syntax very similar to C++
2. **var** *instead of* **int**, **double**, **char**
3. **for** (var i = 0; i < 10; i++) { };
4. **if** (x == 2) { };
5. **var** i = 0; i++; i *= 2;
6. **console.log**(“Message”); - *instead of* **cout** or **printf**

Javascript Callback explanation

1. Callback is a function that can be passed as an argument

function hello() { console.log(“Hello World!”); } //declaration of a function in Javascript

setInterval(hello, 1000); //calling function hello() every second

2. Equivalently

setInterval(function() { console.log(“Hello World!”); }, 1000); //anonymous function

Node.js - description

1. A javascript library designed to run web servers
2. Single threaded (doesn’t block the operating system on modern [after 2007], multi-threaded CPUs)
3. Extremely optimized and efficient
4. Asynchronous - programming comprises of instructing the server how to handle events
5. Modular design, tasks handled using modules (libraries)

Server code

```
var http = require("http"); //equivalent to #include, import, web module
var fs = require("fs"); //file system access, ability to read and write files from the disk
var web_server = http.createServer(function(request, response) {   //order!
  console.log(request);
  response.writeHead(200, {"Content-Type": "text/plain"});
  response.end("Hello World");
}).listen(8888);
```

Server setup

1. Open bash
2. Navigate to the folder where your file is:
    a. cd ~ //cd to home directory shortcut
    b. cd .. //cd to the parent directory shortcut
3. Execute the server:
    a. node “server file name” //to start
    b. <Ctrl-C> //to stop

Server test

1. Open a web browser
2. Type in 127.0.0.1:8888
3. Alternatively localhost:8888
4. Alternatively <your global ip>:8888
5. ---
6. Format: IP:PORT //port chosen with **listen** argument
7. Default port for http (if port field is omitted as in IP) is 80, requires admin privileges to run node, dangerous

Quick html document - filename: index.html

```
<html>
    <body>
        <p>Hello World!</p>
    </body>
</html>
```

Server code - reading index.html

```
var http = require("http"); //equivalent to #include, import, web module
var fs = require("fs"); //file system access, ability to read and write files from the disk
var web_server = http.createServer(function(request, response) {          //order!
  console.log(request);
  fs.readFile("index.html", function(err, data) {
    if (err) throw err; //usually occurs when file doesn't exist
    res.writeHead(200, {"Content-Type": "text/html"});
    res.end(data); }); }).listen(8888); //first close for readFile, second for createServer
```

Final considerations

1. Request can be parsed and response can be adjusted based on the request
2. Request can be used to send data (images, video) to the server
3. Server can be set up to serve files from a folder, being a web server by serving all elements of an html page: html files, images, gifs
4. Request can be used to send login information to the server
5. The code has to be adjusted when receiving large requests