```html
<html>
<head>
    <script src="/socket.io/socket.io.js"></script>
    <script src="http://code.jquery.com/jquery-1.11.1.js">
    </script>
</head>
<body>
<button id="led">Switch LED </button>
<script>
    var socket, text, toggleVal = 0, button = $('#led');
    var socket = io();

    $('#led').click(function () {
        toggleVal += 1;
        toggleVal %= 2;

        if (toggleVal == 0) buttonState = '0';
        if (toggleVal == 1) buttonState = '1';

        socket.emit('toggleLED', buttonState);
        return false;
    });
</script>
</body>
</html>
```

Clientseitiger Code (HTML)

```javascript
var express = require('express');
var app = express();
var http = require('http').Server(app);

app.get('/', function (req, res) {
    res.sendfile('./index.html');
});
http.listen(3000);

// Receive Data from Client via socket.io
// and send it to Arduino via serialport
var socketio = require('socket.io')(http);
var spPackage = require("serialport");
var SerialPort = spPackage.SerialPort;
var portname = "/dev/tty.usbmodemfa141";
var sp = new SerialPort(portname,
    {baudrate: 9600});
sp.open();

socketio.on('connection',function (socket){
    socket.on('toggleLED',function (data){
        //send to Arduino
        sp.write(data);
    });
});
```

Serverseitiger Code (JavaScript)

1. socket.io

2. serialport

```
int ledPin = 13;
String readString;

void setup() {
    Serial.begin(9600);
    pinMode(ledPin, OUTPUT);
}

void loop() {
    while (Serial.available()) {
        delay(3);
        char
        c = Serial.read();
        readString += c;
    }

    if (readString.length() > 0) {
        Serial.println(readString);

        if (readString == "1") {
            digitalWrite(ledPin, HIGH);
        }
        if (readString == "0") {
            digitalWrite(ledPin, LOW);
        }
        readString = "";
    }
}
```

Arduino Code

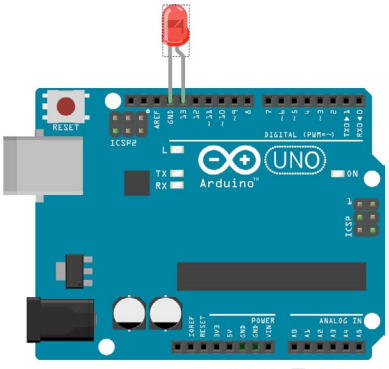

Abb. 11: Weg eines Clients zu einem Arduino Board über eine WebSocket-Verbindung (*1*) und eine serielle Verbindung über USB und TTL-seriell (*2*).