Appendice - Codici Utilizzati

Vengono qui presentati i codici che hanno reso possibile il lavoro di tesi.

Codice in Node.js lato Server

```
var pompa;
var statoPompa;
var r = 0;
var s = 0;
var Input = [];
var NpinAI = [];
var NpinDI = [];
var j = 0;
process.argv.forEach(function (val) {
  if(j>=2){
    Input[j-2]=val;
  }
  j++;
  });
for(var b = 0;b<Input.length;b++){
  if(((Input[b])[0]) == "A"){
    NpinAI[s] = parseInt((Input[b])[1]);
    s++;
  }
  else if(((Input[b])[0]) == "D"){
    NpinDI[r]=parseInt((Input[b])[1]);
  r++;
  }
}
```

```
25  var mraa = require('mraa');
26  console.log('MRAA Version: ' + mraa.getVersion());
27  //variabili sensore di pressione
28  var calibrationFactor = 45;
29  var pulseCount = 0;
30  var flowRate = 0.0;
31  var flowMilliLitres = 0;
32  var totalMilliLitres = 0;
33  var sampleTime = 1000.0;
34  var oldTime = 0;
35  var modbus  = require('jsmodbus'),
36  util = require('util');
37  modbus.setLogger(function (msg) { util.log(msg); });
38  var readCoils = function (start, quant) {
39    var respc = [];
40    for (var i = 0; i < start+quant; i += 1) {
41      respc.push(statoPompa);
42    }
43    return [respc];
44  };
45  var readInputRegHandler = function (start, quant) {
46    var o = 0;
47    var resp = [];
48    for(var a = 0;a<NpinAI.length;a++){
49      var analogPin0 = new mraa.Aio(parseInt(NpinAI[a]));
50      var analogValue = analogPin0.read();
51      resp[o] = analogValue;
52      o++;
53    }
54  //sensore di flusso
55    for (var d = 0;d<NpinDI.length;d++){
56      var myDigitalPin = new mraa.Gpio(parseInt(NpinDI[d]));
57      myDigitalPin.dir(mraa.DIR_IN); //set the gpio direction to input
58      myDigitalPin.isr(mraa.EDGE_RISING,pulseCounter);
59      var hrTime = process.hrtime();
60      var millis = (hrTime[0] * 1000000 + hrTime[1] / 1000);
61      if((millis-oldTime)>sampleTime){
62        myDigitalPin.isrExit(); //l'equvalente di detachinterrupt
```

```
63        flowRate = ((1000.0 / (millis - oldTime)) * pulseCount) /
              calibrationFactor;
64        flowMilliLitres = (flowRate / 60) *1000* (millis-oldTime)
              /1000.0;
65        oldTime = millis;
66        totalMilliLitres += flowMilliLitres;
67        pulseCount = 0;
68        myDigitalPin.isr(mraa.EDGE_RISING,pulseCounter);}
69      resp[o] = flowMilliLitres;
70      o++;
71    }
72  //fine sensore di flusso
73    for (var i = o; i < start+quant; i += 1) {
74      resp.push(0);
75    }
76  return [resp];
77  };
78  var writeSingleRegister = function (adrw, valuew) {
79    pompa = (valuew/10);
80    console.log("pwm pompa = " + pompa);
81    return [];
82  };
83  var writeSingleCoil = function (adr, value) {
84    console.log('write single coil (' + adr + ', ' + value + ')');
85    if(value == true){
86      statoPompa = 1;
87      var pwm3 = new mraa.Pwm(3);
88      pwm3.enable(true);
89      pwm3.period_us(2000);
90      pwm3.write(pompa); // valore da 0 a 1
91    }
92    else{
93      statoPompa = 0;
94      var pwm3 = new mraa.Pwm(3);
95      pwm3.enable(true);
96      pwm3.period_us(2000);
97      pwm3.write(0);}
98      return [];
```

```
99    };
100   modbus.createTCPServer(502, '192.168.2.4', function (err, server) {
101   if (err) {
102     console.log(err);
103     return;
104   }
105   server.addHandler(1, readCoils);
106   server.addHandler(4, readInputRegHandler);
107   server.addHandler(5, writeSingleCoil);
108   server.addHandler(6, writeSingleRegister);
109   });
110   //funzione ausiliare
111   function pulseCounter(){
112     pulseCount++;
113   }
```

Codice in Node.js lato Client

```
1   var modbus  = require('jsmodbus'),
2   util    = require('util');
3   var Input = [];
4   var j =0;
5   process.argv.forEach(function (val) {
6     if(j>=2){
7       Input[j-2]=val;
8     }
9     j++;
10  });
11  modbus.setLogger(function (msg) { util.log(msg); } );
12  var client;
13  var on = 0;
14  loop();
15  function loop(){
16    var client      = modbus.createTCPClient(502, '192.168.2.4');
17    if(on == 0){
18      client.writeSingleRegister(0,parseInt(Input[1]));
19      client.writeSingleCoil(0, parseInt(Input[0]));
20    on++;
21    }
```

```
22      client.readInputRegister (0, 2, function (resp) {
23        console.log('inside the first user cb');
24        console.log(resp);
25      })
26      client.readCoils (0, 1, function (respc, err) {
27        console.log('inside the third user cb');
28        console.log(respc);
29        client.close();
30      })
31      setTimeout(loop,1000)
32    };
```