

**PAMETNO BROJILO U KONCEPTU INDUSTRIJE 4.0****SMART ELECTRICITY METER IN INDUSTRY 4.0**Igor Popadić, *Fakultet tehničkih nauka, Novi Sad***Oblast-ELEKTROTEHNIKA I RAČUNARSTVO**

**Kratak sadržaj** – U radu je dat opis realizacije rješenja povezivanja digitalnog brojila sa web serverom. Dat je opis načina rada digitalnog brojila DIRIS A-10, njegove mogućnosti, a takođe i opis realizacije web servera, prvo povezivanjem tastera koji simulira rad brojila, a na kraju i povezivanjem brojila.

**Ključne riječi:** Digitalno brojilo, IoT, Merjenja

**Abstract** – This paper describes the realization of the solution of connecting a digital meter to web server. A description of how the DIRIS A-10 digital meter works, it's capabilities, and also a description of the realization of a web server, first by connecting a button that simulates the operation of the meter, and finally by connecting the meter.

**Keywords:** Digital electricity meter, IoT, Measurements

**1. UVOD**

Pojava koncepta IoT i sve brži razvoj interneta uslovio je i sve veći razvoj pametnih uređaja koji se mogu povezati na internet i kojima možemo pristupiti pomoću našeg telefona. Sa ovim promjenama došlo je i do pojave pametnog brojila za električnu energiju. Pomoću ovoga brojila korisnik može da vidi koliko je struje potrošio, kad je najviše troši i da shodno tome kontroliše svoju potrošnju i svoje potrebe.

Sa ovim obična digitalna brojila koliko god bila bolja u odnosu na stara, robusna analogna brojila, ipak polako gube na vrijednosti. Elektrodistribucija ide ka tome da ne šalje ljude na teren, već da na web serveru vide potrošnju svakog korisnika.

Tu se javlja problem pravovremenog prikaza podataka na serveru. Vođeci se ovim problemom i željom da korisnicima digitalnog brojila na vizuelno ljepši način prikazemo potrošnju električne energije, kroz ovaj rad je dat primjer pravljenja web servera i povezivanja digitalnog brojila na njega.

**2. OPIS PROBLEMA**

Problem sa kojim se susrećemo u ovome projektu je što digitalno brojilo DIRIS A-10 ima samo analogni izlaz.

Slanjem analognog signala ne može se prikazati broj koji pokazuje potrošnju električne energije u datom vremenskom periodu.

**NAPOMENA:**

**Ovaj rad proistekao je iz master rada čiji mentor je bio dr Marjan Urekar.**

Primarni cilj projekta je naći način kako da se takvo brojilo poveže na web server gdje će u realnom vremenu biti prikazana potrošnja električne energije. I tu dolazi do problema šta uraditi sa tim signalom i kako na osnovu njega na ekranu računara pokazati ono što se nalazi na ekranu digitalnog brojila.

**3. DIRIS A-10**

DIRIS A-10 je izabrano trofazno digitalno brojilo za ovaj rad. To je modularno višenamjensko brojilo za mjerenje električnih vrijednosti u niskonaponskim mrežama. Omogućava prikaz svih električnih parametara, kao i otkrivanje varijacija u temperaturi zahvaljujući svojoj unutrašnjoj funkciji mjerenja temperature.



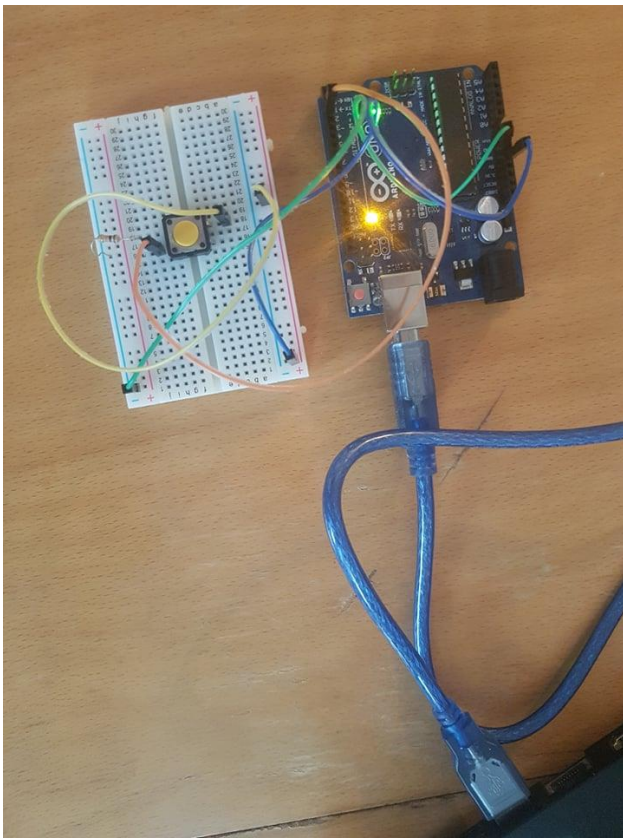
Slika 1. Izgled digitalnog brojila DIRIS A-10

Komunikaciju vrši preko RS485 protokola. Na slici 1. je prikazan izgled prednjeg panela, gdje možemo primjetiti 5 tastera pomoću kojih se mijenja da li će na ekranu biti prikazana struja, napon, temperatura, aktivna i reaktivna snaga.

**3. HARDVER**

Kao što je naglašeno postoji problem kako analogni signal dovesti do računara. Zbog nemogućnosti pristupa laboratoriji i brojilu, odlučeno je brojilo zamijeniti sa tasterom. Analogni signal koji se pošalje prilikom pritiska tastera simulira rad brojila i analogni signal koji bi brojilo slalo. Odlučeno je da se taster postavi na protobord i da se poveže sa arduinom pa onda sa računaram. Za realizaciju ovoga

kola bio je potreban jedan taster, otpornik od 220  $\Omega$ , kratkospojnici, protobord i Arduino Uno. Na slici 2. možemo da vidimo kako izgleda hardverski dio ovoga projekta, tj. kako izgleda kolo koje simulira rad brojila.



Slika 2. Hardverski dio

Arduino ima ulogu da prepozna kad je taster pritisnut i tu dolazimo do prelaska na softversko rješavanje problema.

#### 4. SOFTVER

Prvi korak u rješavanju softverskog dijela problema ovoga projekta je bio napisati kod po kome će arduino da prepoznaje kada je taster pritisnut. Na slici 3. dat je prikaz koda koji to obavlja.

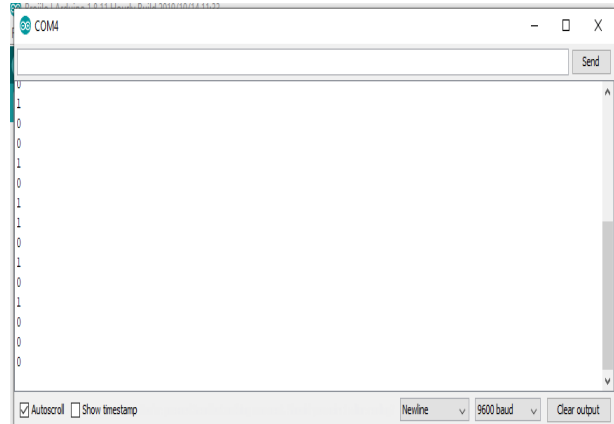
```

Brojilo
int switchButton = 2;
void setup() {
  Serial.begin(9600);
  pinMode(switchButton, INPUT);
}
void loop() {
  int buttonState = digitalRead(switchButton);
  if(buttonState == HIGH){
    Serial.println("1");
  }else{
    Serial.println("0");
  }
  delay(500);
}

```

Slika 3. Arduino kod

Prvi dio koda arduino govori na kom je pinu povezan taster. Onda se u petlji provjerava da li je taster pritisnut, tj. da li je stanje tastera HIGH i ako jeste ispisuje se 1 u prozoru serijskog porta. Ako stanje tastera nije HIGH, tj. taster nije pritisnut ispisuje se 0. Na slici 4. možemo vidjeti kako izgleda ispis nula i jedinica.



Slika 4. Ispis nula i jedinica u prozoru serijskog porta

Sledeći korak u radu ovoga projekta je bio pravljenje web stranice na kojoj će se podaci prikazati. Tu se javlja problem i kako pokupiti nule i jedinice što ih arduino šalje i onda kako ih iskoristiti. Stranica je napravljena kao localhost stranica na adresi 3000. Zatim se pomoću funkcije serialport hvataju nule i jedinice koje šalje arduino koji je povezan u ovome slučaju na port 4 i te nule i jedinice se ispisuju u konzoli. Na slici 5. je dat prikaz koda.

```

JS index.js
c: > Users > Korisnik > Desktop > Brojilo > JS index.js > ...
1 var http = require('http');
2 var express = require('express');
3 var app = express();
4 var server = http.createServer(app);
5 var io = require('socket.io')(server);
6
7
8 const SerialPort = require('serialport');
9 const Readline = require('@serialport/parser-readline');
10 const port = new SerialPort("COM4", { baudRate: 9600 });
11 const parser = new Readline();
12 port.pipe(parser);
13
14 parser.on('data', line => console.log('> ${line}'));
15 app.set('view engine', 'ejs');
16 app.use(express.static(__dirname + '/public'));
17 app.get('/', function (req, res) {
18   res.render('index')
19 });
20
21 port.on('open', function(){
22   console.log('serial port opened');
23 });
24 io.on('connection', function(socket){
25   console.log('socket.io connection');
26   parser.on('data', function(data){
27     data = data.trim();
28     console.log(data);
29     socket.emit('data', data);
30   });
31 });
32 server.listen(3000, function(){
33   console.log('Listening on port 3000...');
34 });

```

Slika 5. Kod za stranicu

Nakon što je napravljena web stranica sledeći problem koji se pojavio je kako prikazati potrošnju. Da bi to riješili napravljena su dva prozora na web stranici. U prvom se ispisuje broj impulsa i taj broj se povećava za 1 sa svakim novim pritiskom tastera.

U drugom prozoru se ispisuje potrošnja električne energije u imp/Ws. U kodu možemo vidjeti da se konstanta množi sa impulsima.

Na stranicu je još dodat i grafik koji se iscrtava u real time-u sa svakim povećanjem potrošnje. Na slici 6. i 7. je dat prikaz tog koda.

```

1  indexjs  ●  indexjs  X
c > Users > Korisnik > Desktop > Brojilo > views > indexjs > ...
2  <html>
3  <head>
4  <title>Get data</title>
5  <div id="chart"></div>
6  </head>
7  <body style="background-color: #lightgrey">
8  <h1>Arduino data</h1>
9  <p>The button is <span id="button-state"></span></p>
10 <input type="text" id="in" value="0"/> <p>imp</p>
11 <input type="text" id="inc" value="0"/> <p>imp/WS</p>
12 <script src="/socket.io/socket.io.js"></script>
13 <script src="https://cdn.plot.ly/plotly-latest.min.js"></script>
14 <script>
15   var socket = io();
16   var i = 0 ;
17   var buttonState=document.getElementById('button-state');
18   socket.on("data", function(data){
19     if(data === "1" && i>= 0){
20       buttonState.innerHTML = "pressed";
21       i += 1;
22     }else{
23       buttonState.innerHTML = "not pressed";
24     }
25     document.getElementById('in').value = i;
26   });
27

```

Slika 6. Prvi dio koda

```

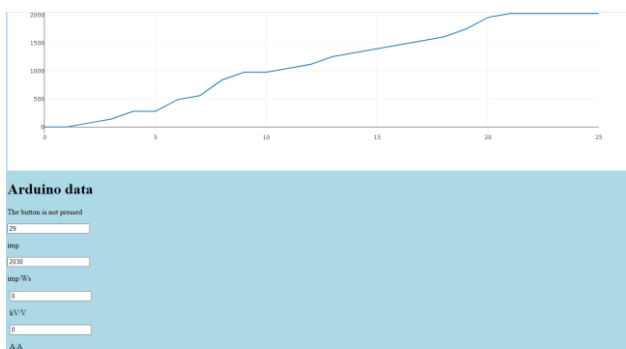
28   var k = 0 ;
29   var buttonState=document.getElementById('button-state');
30   socket.on("data", function(data){
31     if(data === "1" && i>= 0){
32       buttonState.innerHTML = "pressed";
33       k = i*70;
34     }else{
35       buttonState.innerHTML = "not pressed";
36     }
37     document.getElementById('inc').value = k;
38   });
39
40   function getData() {
41     return k;
42   }
43
44   Plotly.plot('chart',[[
45     y :[getData()],
46     type:'line'
47   ]]);
48
49   setInterval(function() {
50     Plotly.extendTraces('chart', { y: [[getData()]] }, [0])
51   }, 2000);
52
53 </script>
54 </body>
55 </html>

```

Slika 7. Drugi dio koda

## 5. TESTIRANJE SISTEMA

Kada je kod pokrenut i taster pritisnut vidjeli smo da kolo obavlja posao. Na slici 8. dat je prikaz web stranice.



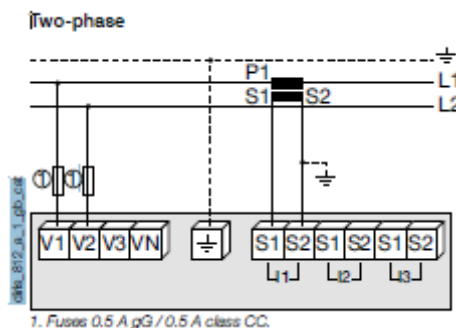
Slika 8. Prikaz izgleda web stranice

Vidimo kako izgleda iscrtan grafik i vidimo kako izgleda prikaz energije i impulsa. Y osa grafika pokazuje potrošnju električne energije koja je izražena u imp/Ws.

Grafik se konstantno ispisuje, a krivulja raste sa svakim novim pritiskom tastera. Sledeći korak je povezivanje brojila umjesto tastera.

## 6. ZAKLJUČAK

Na slici 9. vidimo primjer dvofaznog načina povezivanja brojila.



Slika 9. Primjer dvofaznog brojila

Realizovani sistem u kome se brojilo povezuje na web server nudi mogućnost korisniku da isprati potrošnju električne energije u realnom vremenu. Grafički prikaz podataka, pored prikaza potrošnje i impulsa u brojevima, nudi mogućnost realnog prikaza kako potrošnja raste i da prema tome korisnik preuzme određene akcije povodom smanjenja potrošnje.

Sa realizacijom sajta i hardverskog rješenja, ograničenosti povezivanja digitalnog brojila koje ima izlaz samo za impulse u pogledu povezivanja na internet su otklonjene. Brojilo sada može da se iskoristi u konceptu Industrije 4.0. Sistem odlikuje stabilan rad.

Dodatkom prozora za konstantu naponskih i strujnih transformatora omogućeno je da se provjeri rad brojila pomoću sajta.

Sledeći korak bi bio dodavanje dodatnih opcija na sajt kao što su prikaz struje, napona i temperature, tj. svih opcija koje prikazuje brojilo na svom ekranu. Sve te veličine bi bile prikazane grafički. Sajt bi se mogao naprijediti tako da mu se može pristupiti i sa telefona. Tada sajt ne bi bio korišten samo za brojilo Diris A-10, već svaki korisnik može personalizovati sajt i koristiti ga za više brojila i vidjeti potrošnju za svaki uređaj u domu. Takođe bi se moglo dodati par dodatnih senzora na samo brojilo da se unaprijede njegove performanse.

## 7. LITERATURA

- [1] Indira Knight, "Connecting Arduino to the Web", 2018
- [2] Cuno Pfister, „Getting started with internet of things“, O'Reilly, 2011.
- [3] Peter Waher, „Learning Internet of Things“, Packt publishing, 2015.
- [4] Paul Horowitz, Winfield Hill, „The Art of Electronics“, 2015

### Kratka biografija

**Igor Popadić** rođen je u Trebinju 1996. godine. Diplomirao je na Fakultetu tehničkih nauka u Novom Sadu, na katedri za električna mjerenja 2019. godine. Master rad iz oblasti mjerno-informacionih sistema, odbranio je 2020. godine.