



ТЕОРИЈА ЕЛЕКТРИЧНИХ КОЛА
ХАРДВЕРСКИ ПРОЈЕКАТ
ДИГИТАЛНИ ЈЕДНОСМЕРНИ ВОЛТМЕТАР
ГРАФИЧКИ КОРИСНИЧКИ ИНТЕРФЕЈС – ДОДАТАК

СТУДЕНТИ:

Глорија Дошло 2019/0065

Кристина Рајковић 2019/0447

МЕНТОР:

доц. др Никола Баста

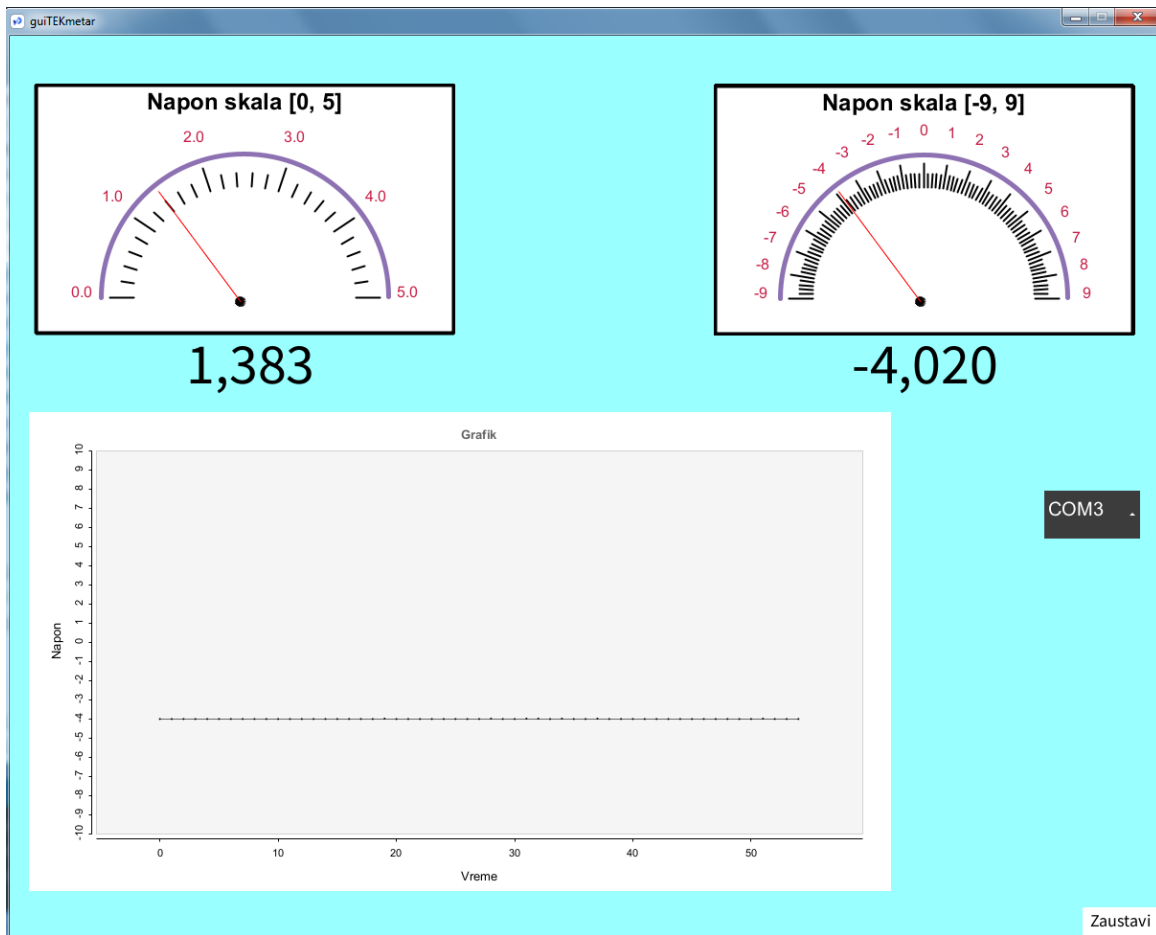
Фебруар 2022. г.

Садржај

Графички кориснички интерфејс – унапређена апликација.....	3
Литература	11

ГРАФИЧКИ КОРИСНИЧКИ ИНТЕРФЕЈС – УНАПРЕЂЕНА АПЛИКАЦИЈА

У овом додатку биће представљено једно унапређење постојећег решења које се налази у извештају пројекта. Наиме, проблем који се јављао се огледао у томе што при коришћењу уграђених метара није било могуће на скали урадити прецизно читавање напона јер постоји одређен корак за који се може игла померити, тако да су праве вредности биле приказане са одређеном грешком.



Слика 1: GUI апликација

Као што видимо на слици, све функционалности су остале исте, а измене су вршене из естетских разлога. Да бисмо добили жељену прецизност, метри су реализовани тако што смо из претхонде апликације исекли слике метара и игли и онда вршили ротације слика у складу са прочитаним вредностима како бисмо постигли жељени ефекат.

Код ове апликације је следећи:

```
// biblioteka koja ima dugme, prekidas, padajucu listu itd.
import controlP5.*;
// biblioteka za komunikaciju sa arduinom
import processing.serial.*;
// biblioteka za crtanje grafika
import grafica.*;
import java.util.Arrays;

color backgroundColor = color(153, 255, 255);
color textColor = color(0, 0, 0);

Serial port;
GPlot plot;
ControlP5 cp5;
DropDownList dl;

String arduinoPortName;
String portName;

PImage meter1, meter2, needle;

boolean newData = true;
int xPos = 0;
boolean rectOver = false;
boolean begin = true;
boolean arduinoConnected = false;
boolean portLost = false;

int rectX, rectY, rectWidth, rectHeight;
color rectHighlight = color(204);
color rectColor = color(255);
String[] lastComList;

int mx = 25, my = 50, mw, mh;
int m2x, m2y, m2w, m2h;

void setup() {
    lastComList = Serial.list();
    portName = Serial.list()[0]; //0 as default
    port = new Serial(this, portName, 9600);

    size(1200, 950);
    background(backgroundColor);
    cp5 = new ControlP5(this);

    // pravljenje fonta
    PFont pfont = createFont("Arial", 10, true); //Create a font
    ControlFont font = new ControlFont(pfont, 20); //font, font-size
    // podesavanja za padajucu listu
    dl = cp5.addDropDownList("myList-dl")
        .setPosition(width - 120, height/2)
```

```

        .setSize(100, 200)
        .setHeight(210)
        .setItemHeight(40)
        .setBarHeight(50)
        .setFont(font)
        .setColorBackground(color(60))
        .setColorActive(color(255, 128))
        ;

// stavlja PORT pre nego sto se nesto izabere
dl.getCaptionLabel().set("PORT");

meter1 = loadImage("meter1.PNG");
needle = loadImage("needle-removebg-preview.png");
meter2 = loadImage("meter2.PNG");

mw = meter1.width;
mh = meter1.height;
m2x = width - mx - mw;
m2y = my;
m2w = meter2.width;
m2h = meter2.height;

plot = new GPlot(this);
plot.setPos(20, my + mh + 80);
plot.setDim(800, 400);
plot.setPointColor(color(0, 0, 0, 255));
plot.setPointSize(2);
plot.setTitleText("Grafik");
plot.getXAxis().getAxisLabel().setText("Vreme");
plot.getYAxis().getAxisLabel().setText("Napon");
plot.drawGridLines(GPlot.BOTH);
plot.setGridLineColor(color(0, 0, 0));
// aktivira zumiranje, CENTER je middle mouse button
plot.activateZooming(2, CENTER, CENTER);
// kad se klikne desni klik na grafik, onda se grafik posmatra od (0, y)
plot.activateReset(); // desni klik misem na grafik
// levim klikom na grafik moze da se pomera posmatranje grafika
plot.activatePanning();
// postavlja se donja i gornja granica y ose
plot.setYLim(-10, 10);
// postavlja se koliko y vrednosti ce se pokazati na y osi
// u ovom slucaju se prikazuje od -10, -9, -8, ..., 0, ..., 8, 9, 10
plot.setVerticalAxesNTicks(18);

rectWidth = 80;
rectHeight = 40;
rectX = width - rectWidth;
rectY = height - rectHeight;
// ovaj deo koda sluzi za ispis pocetnog napona, koji je nula
// jer arduino na pocetku nije povezan
background(backgroundColor);
 textSize(60);
 textAlign(CENTER);
 fill(textColor);
 text(0, mx + mw/2, my + mh + 50);
 text(0, m2x + m2w/2, m2y + m2h + 50);

```

```

    // stvaraju se niti tako sto se prosledjuje ime metode koju nit treba da
    izvrsava
    thread("DropListThread");
    thread("portsListThread");
} // kraj setup-a

void draw() {
    // služi za početna iscrtavanja
    if (begin) {
        plot.defaultDraw();
        update(); // služi za dugme, da bi se iscrtalo na samom početku

        // ako se ne stavi onda se zapamti translate i rotate podesavanja, pa
        posle mogu nastati problemi
        pushMatrix();
        image(meter1, mx, my);
        translate(mx + meter1.width/2 + 5, my + meter1.height/3 - 10 +
        needle.height - 10);
        rotate(HALF_PI);
        image(needle, 0, 0);
        popMatrix();
        pushMatrix();
        image(meter2, m2x, m2y);
        translate(m2x + meter2.width/2 + 5, m2y + meter2.height/3 - 10 +
        needle.height - 10);
        rotate(HALF_PI);
        image(needle, 0, 0);
        popMatrix();
    }
    // proverava da li je doslo do nekog problema i onda ispisuje odgovarajucu
    poruku
    // probleme joj javljaju niti
    if (!arduinoConnected || portLost) {
        fill(backgroundColor);
        stroke(backgroundColor);
        rect(width - 2*rectWidth - 115, height/2 - 80, 4*rectWidth, rectHeight +
30);
        fill(textColor);
        textSize(20);
        textAlign(CENTER);
        if (portLost)
            text("Arduino izgubljen!\nIzaberite port.", width - 2*rectWidth + 20,
height/2 - 60);

        else {
            text("Nema konekcije sa Arduino!\nIzaberite port.", width -
2*rectWidth + 20, height/2 - 60);
        }
    }

    String str = "";
    String[] tokens;
    if (port.available() > 0) {
        str = port.readStringUntil('\n');

        if (str != null) {

```

```

tokens = split(str, ',');
// na pokretanju znalo je da se desi da procita nesto lose
// zato ovaj if proverava da li su procitane dve vrednosti
if (tokens.length == 2) {
    float voltage = float(tokens[1]);
    // ovo se izvrsava sve dok korisnik ne pritisne dugme "zaustavi"
    if (newData) {
        background(backgroundColor);
        textSize(60);
        textAlign(CENTER);

        float val = (float(tokens[0]) / 745.0) * 5;

        fill(textColor);
        text(val, mx + mw/2, my + mh + 50);
        text(voltage, m2x + m2w/2, m2y + m2h + 50);
        float reading;
        // na podeoku 375 arduino pokazuje 0V
        // if je tu samo da malo popravi gresku, pa se deli na dva dela
        // od -9 do 0 i od 0 do 9
        if (float(tokens[0]) >= 375)
            reading = (float(tokens[0]) / 745.0 * PI);
        else
            reading = (float(tokens[0]) / 745.0 * PI) + PI/50;
        //imageMode(CENTER); //draw image using center mode
        image(meter1, mx, my); //drawing meter image
        image(meter2, m2x, m2y);
        pushMatrix(); //saving current transformation matrix onto the
matrix stack
        imageMode(CORNER); //draw image using corner mode
        translate(mx + meter1.width/2 + 5, my + meter1.height/3 - 10 +
needle.height);
        rotate(HALF_PI + reading);
        image(needle, 0, 0); //drawing needle image
        popMatrix();
        pushMatrix();
        translate(m2x + meter2.width/2 + 5, m2y + meter2.height/3 - 10 +
needle.height );
        rotate(HALF_PI + reading);
        image(needle, 0, 0); //drawing needle image
        popMatrix(); //removing the current transformation matrix off the
matrix stack

        // vracanje grafika da opet ide od nula
        if (xPos > 300) {
            while (xPos > 0) {
                plot.removePoint(--xPos);
            }
        }

        plot.addPoint(xPos++, voltage);
        plot.defaultDraw();
    } //kraj if-a newData
}
fill(backgroundColor);
stroke(backgroundColor);

```

```

    rect(0, height - rectHeight, 400, rectHeight);

    fill(textColor);
    textSize(20);
    textAlign(LEFT);
    // služi za ispisivanje x i y kordinate koje se očitavaju sa grafika
    // tamo gde pokazuje mis
    if (plot.isOverBox(mouseX, mouseY)) {
        float[] value = plot.getValueAt(mouseX, mouseY);
        text("x = " + value[0] + ", y = " + value[1], 0, height - 20);
    }
    update();
}
} // kraj if-a port.available
} // kraj draw-a

// proverava da li je mis presao preko pravougaonika
boolean overRect(int x, int y, int width, int height) {
    if (mouseX >= x && mouseX <= x+width &&
        mouseY >= y && mouseY <= y+height) {
        return true;
    } else {
        return false;
    }
}

// proverava da li je mis presao preko dugmeta i onda mu menja boju
void update() {
    if (overRect(rectX, rectY, rectWidth, rectHeight)) {
        rectOver = true;
        fill(rectHighlight);
    } else {
        rectOver = false;
        fill(rectColor);
    }
    stroke(255);
    rect(rectX, rectY, rectWidth, rectHeight);
    textAlign(CENTER);
    fill(textColor);
    textSize(18);
    text("Zaustavi", rectX + rectWidth / 2, rectY + rectHeight / 2);
}

boolean pressedOnce = true;
// ova metoda proverava da li je pritisnuto dugme i ako jeste stavlja
// newData = false da draw funkcija ne bi ispisivala ono što dobije od arduin
a
// tako da se "zamrzne" ekran
void mousePressed() { // kad se pritisne mis onda se pozove ova metoda

    if (rectOver && pressedOnce) {
        newData = false;
        pressedOnce = false;
        fill(backgroundColor);
        stroke(backgroundColor);
        rect(rectX - 3*rectWidth, rectY, 3*rectWidth, rectHeight);
        fill(textColor);
    }
}

```



```

        textSize(20);
        text("Zaustavljeno!", width - rectWidth - rectWidth, height - 10);
    } else if (rectOver && !pressedOnce) {
        newData = true;
        pressedOnce = true;
    }
}

// poziva se kad je nesto u listi selektovano, proverava sta je selektovano i
// menja port na taj koji
// je izabran
void controlEvent(ControlEvent theEvent) {
    port.clear(); //delete the port
    port.stop(); //stop the port
    begin = false;
    if (theEvent.isController() && d1.isMouseOver()) {
        portName = Serial.list()[int(theEvent.getController().getValue())];
//port name is set to the selected port in the dropDownMeny
        port = new Serial(this, portName, 9600); //Create a new connection
        delay(1000); // ceka arduino da nesto ispise
        // cita da li je nesto ispisano i ako jeste onda je to arduino
        if (port.readStringUntil('\n') == null) {
            arduinoConnected = false;
        } else {
            arduinoPortName = portName;
            arduinoConnected = true;
        }
        //delay(2000);
    }
}

// nit koja svaki put kad mis predje preko nje azurira padajucu listu
// jer se moze desiti da je neki port nestao i on ne treba da se nalazi u pad
// ajucoj listi
void DropListThread() {
    println("DropListThread started");
    while (true) {
        if (d1.isMouseOver()) {
            d1.clear(); //Delete all the items
            for (int i=0; i<Serial.list().length; i++) {
                d1.addItem(Serial.list()[i], i); //add the items in the list
            }
        }
        delay(800);
    }
}

// nit koja proverava da li je neki port izgubljen i ako je taj port arduinov
// onda javlja da nema konekciju sa arduinom, to moze da se desi kad se iscup
// a usb
void portsListThread() {
    println("portsListThread started");
    while (true) {
        println("proveravam uslov");
        String[] tempList = Serial.list();
        if (!Arrays.equals(lastComList, tempList)
            && !Arrays.asList(tempList).contains(arduinoPortName)) {

```

```

        println("Arduino izgubljen");
        portLost = true;
    }
    if (Arrays.equals(lastComList, tempList) && portLost) {
        portLost = false;
    }
    // nema potrebe da radi non stop pa joj se stavlja neko vreme spavanja
    delay(3000);
}
}

```

Такође, у односу на код приказан у извештају, овде постоје додатни коментари који ближе описују одређене делове ради лакшег разумевања. Сва објашњења која су тамо дата могу се применити и овде, уз наравно измене које се тичу метара за читавање.

ЛИТЕРАТУРА

1. Теорија електричних кола – сајт предмета ([TEORIJA ELEKTRIČNIH KOLA, Univerzitet u Beogradu -- Elektrotehnički fakultet \(etf.rs\)](#))
2. Arduino ([Arduino - Home](#))
3. Processing ([Welcome to Processing! / Processing.org](#))