

ΑΝΩΤΑΤΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΠΕΙΡΑΙΑ
ΤΕΧΝΟΛΟΓΙΚΟΥ ΤΟΜΕΑ

ΑΕΙ Πειραιά Τ.Τ.

Τμήμα Μηχανικών Ηλεκτρονικών

Υπολογιστικών Συστημάτων

Εργαστήριο Μικροελεγκτών

7η Εργαστηριακή άσκηση

Επιμέλεια:

Στυλιανός Βουτσινάς

BSc, MSc, PhD(Cand.)

Υπεύθυνος Εργαστηρίου:

Δρ. Έλληνας Ιωάννης (Καθηγητής)

BSc, MSc, PhD

Αναλυτική Περιγραφή

7η Εργαστηριακή άσκηση: Απεικόνιση θερμοκρασίας σε εφαρμογή με χρήση γλώσσας Processing

Τεκμηρίωση Ορισμών

Τεκμηρίωση Ορισμών

#define SPAN 20

μακροεντολή που ορίζει το πλήθος των επαναλήψεων στο φίλτρο
Ορισμός στη γραμμή [5](#) του αρχείου [lm35.ino](#).

#define TEMPIN A2

μακροεντολή που ονοματίζει το pin A2.
Ορισμός στη γραμμή [3](#) του αρχείου [lm35.ino](#).

Τεκμηρίωση Συναρτήσεων**void loop ()**

Η συγκεκριμένη συνάρτηση, εκτελείται συνέχεια

Ορισμός στη γραμμή [31](#) του αρχείου [lm35.ino](#).

```
31 {
32   int analog = movingAverage();
33   Serial.println(analog);
34 }
```

Το διάγραμμα δείχνει ποιες συναρτήσεις καλούνται από αυτή:

**int movingAverage ()**

Υλοποιεί ένα φίλτρο κυλιόμενου μέσου.(Moving Average Filter)

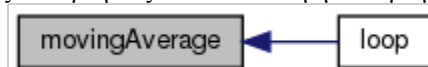
Επιστρέφει:

το μέσο όρο των n δειγμάτων που έχουν οριστεί από την σταθερά SPAN

Ορισμός στη γραμμή [20](#) του αρχείου [lm35.ino](#).

```
20 {
21   int analog = 0;
22   for (int i = 0; i < SPAN; i++) {
23     analog += analogRead(TEMPIN);
24   }
25   analog /= SPAN;
26   return analog;
27 }
```

Το διάγραμμα δείχνει από ποιες συναρτήσεις καλείται αυτή η συνάρτηση:

**void setup ()**

Η συγκεκριμένη συνάρτηση, εκτελείται μία φορά και χρησιμοποιείται για να αρχικοποιήσει τον μE

Ορισμός στη γραμμή [12](#) του αρχείου [lm35.ino](#).

```
12 {
13   Serial.begin(9600); //96008N1
14   analogReference(INTERNAL); // reference to 1.1 V
15 }
```


Κώδικας άσκησης *lm35.ino*

```
3 #define TEMPIN A2
4
5 #define SPAN 20
6
7
8
12 void setup() {
13   Serial.begin(9600); //96008N1
14   analogReference(INTERNAL); // reference to 1.1 V
15 }
16
20 int movingAverage() {
21   int analog = 0;
22   for (int i = 0; i < SPAN; i++) {
23     analog += analogRead(TEMPIN);
24   }
25   analog /= SPAN;
26   return analog;
27 }
28
31 void loop() {
32   int analog = movingAverage();
33   Serial.println(analog);
34 }
```

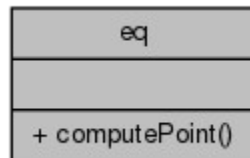
Κώδικας γραφικής εφαρμογής σε Processing 2.2.1

Τεκμηρίωση Κλάσης eq

Κλάση για την αναπαράσταση σημείων σε γράφημα 2 διαστάσεων

Κληρονομεί την ILine2DEquation.

Διάγραμμα Συνεργασίας για την κλάση eq:



Δημόσιες Μέθοδοι

- double [computePoint](#) (double x, int pos)

Μετατρέπει τις στάθμες που διαβάστηκαν από την σειριακή θύρα σε θερμοκρασία

```

27     {
28     return xPos/9.31;
29     }
  
```

Τεκμηρίωση Συναρτήσεων

void controlEvent (ControlEvent theEvent)

Η συγκεκριμένη συνάρτηση, καλείται κάθε φορά που υπάρχει αλλαγή στα χειριστήρια της εφαρμογής (π.χ. στο κουμπί σύνδεσης της σειριακής)

```

113     {
114     if (theEvent.isController()) {
115         switch(theEvent.getController().getId()) {
116             case(1):
117                 try {
118                     if (b1.getBooleanValue()==true) {
119                         try
120                         {
121                             int index = int(d1.getValue());
122                             System.err.println(index);
123                             COMPort = new Serial(this, Serial.list()[index], 9600);
124                             b1.setCaptionLabel("Disconnect_COM");
125                             COMPort.clear(); //Discart data from previous connections
126                         }
127                         catch(Exception e1)
128                         {
129                             System.err.println("Error while opening Digitiser port: " + e1);
130                             System.err.println("Please check your connection and ensure you
've selected the correct Com port");
131                         }
132                     } else {
133                         COMPort.clear();
134                         COMPort.stop();
135                         System.out.println("Connection Terminated. ");
136                     }
137                 }
138                 catch(Throwable e2)
139                 {
140                     System.err.println("Generic error: "+ e2);
141                 }
142                 break;
143             }
144         }
145     }
  
```

void draw ()

Η συγκεκριμένη συνάρτηση, εκτελείται σε επανάληψη

```

89     {
90     background(128);
91     g.draw();
92     fill(255);
93     textSize(26);
94     text("Arduino Temperature plotter", 300, 40);
95 }

```

void serialEvent (Serial COMPort)

Η συγκεκριμένη συνάρτηση, καλείται κάθε φορά που καταφθάνουν δεδομένα στην Σειριακή Πόρτα. Διαβάζει δεδομένα μέχρι να συναντήσει χαρακτήρα αλλαγής γραμμής

```

100     {
101     String inString = COMPort.readStringUntil('\n');
102     if (inString != null) {
103         inString = trim(inString);
104         int data = int(inString);
105         inByte1 = data;
106         xPos = inByte1;
107     }
108 }

```

void setup ()

Αρχικοποίηση του γραφικού περιβάλλοντος

```

33     {
34     size(800, 600);
35     cp5 = new ControlP5(this);
36
37     b1 = cp5.addButton("Connect COM")
38         .setColorBackground( color( #799195) )
39         .setPosition(160, 10)
40         .setSize(105, 20)
41         .setSwitch(false)
42         .setId(1)
43         .setOn()
44         ;
45
46     d1 = cp5.addDropDownList("Available COMs");
47     d1.clear();
48     int numCOM = Serial.list().length;
49     for (int i=0; i<numCOM; i++)
50     {
51         d1.addItem(Serial.list()[i], i);
52     }
53     d1.setPosition(30, 30);
54     d1.setSize(100, 160);
55     d1.setItemHeight(20);
56     d1.setBarHeight(20);
57     d1.getCaptionLabel().getStyle().marginTop = 3;
58     d1.getCaptionLabel().getStyle().marginLeft = 5;
59     d1.getCaptionLabel().getStyle().marginRight = 5;
60     d1.getValueLabel().getStyle().marginTop = 3;
61     d1.setColorBackground(color(#799195));
62     d1.setColorActive(color(255, 128));
63
64
65     r = new RollingLine2DTrace(new eg(), 1, 0.001f); // original: 0.01f
        changed: 1, 0.001f
66     r.setTraceColour(255, 0, 0);
67     r.setLineWidth(2);
68
69     g = new Graph2D(this, 650, 500, false);
70     g.setAxisColour(0, 0, 0); // Black
71     g.setFontColour(0, 0, 0); // Black
72     g.addTrace(r);
73     gb = new GridBackground(new GWColour(230));

```

```
74  gb.setGridColour(180, 180, 180, 180, 180, 180);
75  g.setBackground(gb);
76  g.position.y = 50;
77  g.position.x = 100;
78  g.setYAxisTickSpacing(100);
79  g.setXAxisTickSpacing(1);
80  g.setXAxisLabel("Time (sec)");
81  g.setYAxisLabel("Temperature");
82  g.setYAxisMax(100);
83  g.setYAxisMin(0);
84  g.setXAxisMax(5f);
85 }
```


Κώδικας άσκησης serial_plotter.pde

```

import processing.serial.*;
import controlP5.*;
import org.gwoptics.graphics.graph2D.Graph2D;
import org.gwoptics.graphics.graph2D.traces.ILine2DEquation;
import org.gwoptics.graphics.graph2D.traces.RollingLine2DTrace;
import org.gwoptics.graphics.graph2D.backgrounds.*;
import org.gwoptics.graphics.*;
import org.gwoptics.graphics.graph2D.backgrounds.GridBackground;

DropdownList dl;
Button b1;
ControlP5 cp5;
Serial COMPort;
RollingLine2DTrace r;
Graph2D g;
GridBackground gb;
double xPos = 1; // Initial position of pointer
int inByte1;

class eq implements ILine2DEquation {
    public double computePoint(double x, int pos) {
        return xPos/9.31;
    }
}

void setup() {
    size(800, 600);
    cp5 = new ControlP5(this);

    b1 = cp5.addButton("Connect_COM")
        .setColorBackground( color( #799195) )
        .setPosition(160, 10)
        .setSize(105, 20)
        .setSwitch(false)
        .setId(1)
        .setOn()
        ;

    dl = cp5.addDropdownList("Available COMs");
    dl.clear();
    int numCOM = Serial.list().length;
    for (int i=0; i<numCOM; i++)
    {
        dl.addItem(Serial.list()[i], i);
    }
    dl.setPosition(30, 30);
    dl.setSize(100, 160);
    dl.setItemHeight(20);
    dl.setBarHeight(20);
    dl.getCaptionLabel().getStyle().marginTop = 3;
    dl.getCaptionLabel().getStyle().marginLeft = 5;
    dl.getCaptionLabel().getStyle().marginRight = 5;
    dl.getValueLabel().getStyle().marginTop = 3;
    dl.setColorBackground(color(#799195));
    dl.setColorActive(color(255, 128));

    r = new RollingLine2DTrace(new eq(), 1, 0.001f); // original: 0.01f changed: 1,
0.001f
    r.setTraceColour(255, 0, 0);
    r.setLineWidth(2);

    g = new Graph2D(this, 650, 500, false);
    g.setAxisColour(0, 0, 0); // Black
    g.setFontColour(0, 0, 0); // Black
    g.addTrace(r);
    gb = new GridBackground(new GWColour(230));
    gb.setGridColour(180, 180, 180, 180, 180, 180);
    g.setBackground(gb);
    g.position.y = 50;
    g.position.x = 100;
    g.setYAxisTickSpacing(100);
    g.setXAxisTickSpacing(1);
    g.setXAxisLabel("Time (sec)");
    g.setYAxisLabel("Temperature");

```

```

        g.setYAxisMax(100);
        g.setYAxisMin(0);
        g.setXAxisMax(5f);
    }

    void draw() {
        background(128);
        g.draw();
        fill(255);
        textSize(26);
        text("Arduino Temperature plotter", 300, 40);
    }

    void serialEvent (Serial COMPort) {
        String inString = COMPort.readStringUntil('\n');
        if (inString != null) {
            inString = trim(inString);
            int data = int(inString);
            inByte1 = data;
            xPos = inByte1;
        }
    }

    void controlEvent(ControlEvent theEvent) {
        if (theEvent.isController()) {
            switch(theEvent.getController().getId()) {
                case(1):
                    try {
                        if (b1.getBooleanValue()==true) {
                            try
                            {
                                int index = int(d1.getValue());
                                System.err.println(index);
                                COMPort = new Serial(this, Serial.list()[index], 9600);
                                b1.setCaptionLabel("Disconnect_COM");
                                COMPort.clear(); //Discart data from previous connections
                            }
                            catch(Exception e1)
                            {
                                System.err.println("Error while opening Digitiser port: " + e1);
                                System.err.println("Please check your connection and ensure you 've
selected the correct Com port");
                            }
                        } else {
                            COMPort.clear();
                            COMPort.stop();
                            System.out.println("Connection Terminated. ");
                        }
                    }
                    catch(Throwable e2)
                    {
                        System.err.println("Generic error: "+ e2);
                    }
                    break;
            }
        }
    }
}

```