

ΑΝΩΤΑΤΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΠΕΙΡΑΙΑ
ΤΕΧΝΟΛΟΓΙΚΟΥ ΤΟΜΕΑ

ΑΕΙ Πειραιά Τ.Τ.

Τμήμα Μηχανικών Ηλεκτρονικών

Υπολογιστικών Συστημάτων

Εργαστήριο Μικροελεγκτών

8η Εργαστηριακή άσκηση

Επιμέλεια:

Στυλιανός Βουτσινάς

BSc, MSc, PhD(Cand.)

Υπεύθυνος Εργαστηρίου:

Δρ. Έλληνας Ιωάννης (Καθηγητής)

BSc, MSc, PhD

Αναλυτική Περιγραφή

8η Εργαστηριακή άσκηση: Απεικόνιση θερμοκρασίας σε IOT εφαρμογή με χρήση γλώσσας Processing

Τεκμηρίωση Ορισμών

Τεκμηρίωση Ορισμών

#define SPAN 100

μακροεντολή που ονοματίζει το πλήθος δειγμάτων του φίλτρου
Ορισμός στη γραμμή 5 του αρχείου [lm35Wtimer.ino](#).

#define TEMPIN A2

μακροεντολή που ονοματίζει το pin A2.
Ορισμός στη γραμμή 3 του αρχείου [lm35Wtimer.ino](#).

int analog = 0

μεταβλητή που θα φέρει την τιμή της θερμοκρασίας σε στάθμες
Ορισμός στη γραμμή 9 του αρχείου [lm35Wtimer.ino](#).

volatile boolean reading = false

μεταβλητή που θα ελέγχει την αποστολή δεδομένων προς τον υπολογιστή
Ορισμός στη γραμμή 7 του αρχείου [lm35Wtimer.ino](#).

Timer t

ορισμός αντικειμένου timer.
Ορισμός στη γραμμή 13 του αρχείου [lm35Wtimer.ino](#).

Τεκμηρίωση Συναρτήσεων

void loop ()

Υλοποιεί ένα φίλτρο κυλιόμενου μέσου. (Moving Average Filter) για SPAN μετρήσεις οι οποίες λαμβάνονται ανάμεσα στις διακοπές του timer.

Ορισμός στη γραμμή 30 του αρχείου [lm35Wtimer.ino](#).

```

31 {
32   t.update();
33   i++;
34   if (reading) {
35     for (i = 0; i < SPAN; i++) {
36       analog += analogRead(TEMPIN);
37     }
38     analog /= SPAN;
39     Serial.println(analog);
40     analog = 0;
41     reading = false;
42   }
43 }
```

void setup ()

Η συγκεκριμένη συνάρτηση, εκτελείται μία φορά και χρησιμοποιείται για να αρχικοποιήσει τον μΕ. Ορίζει σειριακή επικοινωνία 96008N1 και εσωτερική τάση αναφοράς τα 1.1V. Ο timer κάθε 1 δευτερόλεπτο θα καλεί την συνάρτησης takeReading.

Ορισμός στη γραμμή 19 του αρχείου [lm35Wtimer.ino](#).

```

20 {
21   Serial.begin(9600);
22   analogReference(INTERNAL);
23   t.every(1000, takeReading);
24 }
```

Το διάγραμμα δείχνει ποιες συναρτήσεις καλούνται από αυτή:



void takeReading ()

κάθε φορά που προκαλείται διακοπή από τον timer θέτει τιμή true στην volatile μεταβλητή reading.

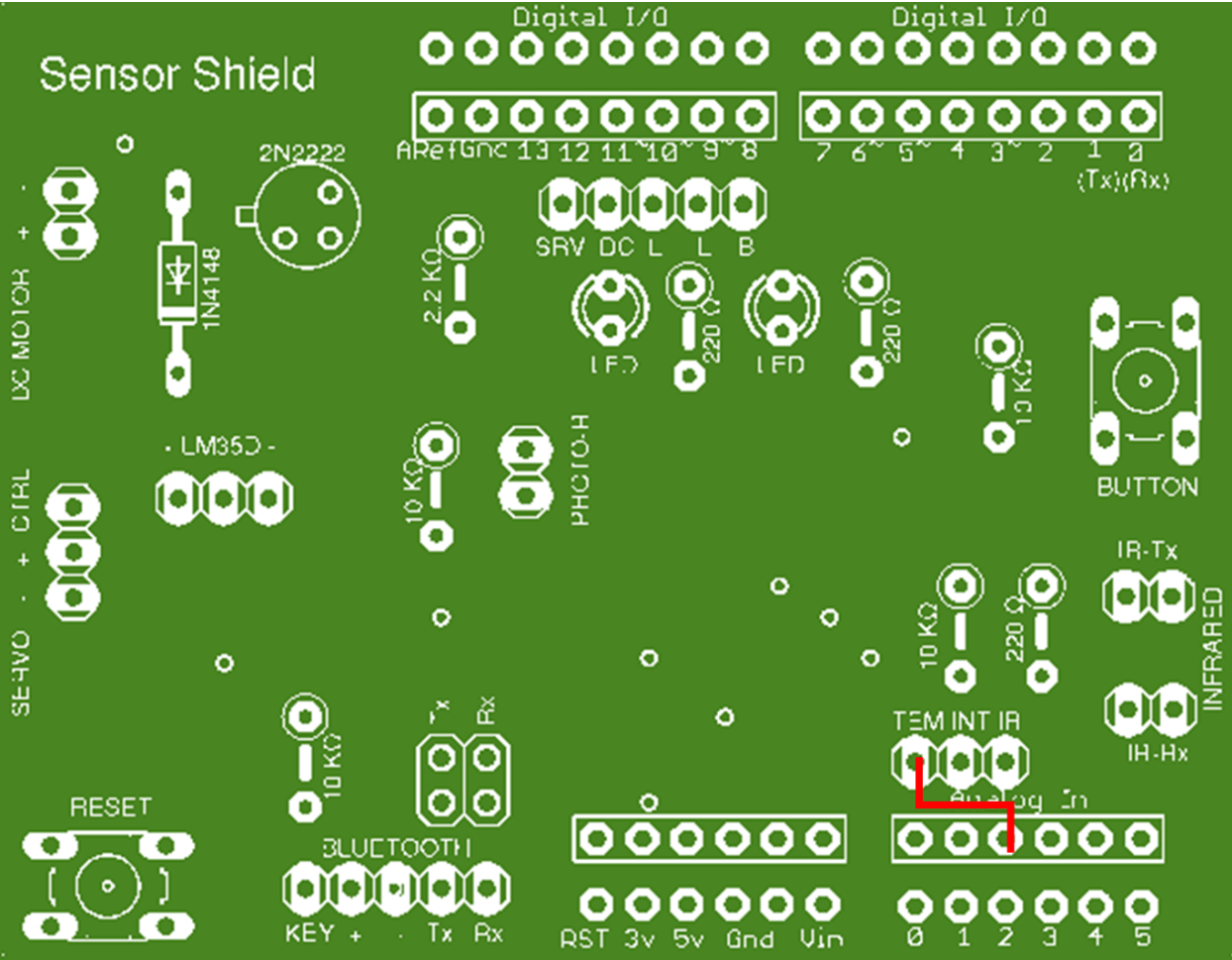
Ορισμός στη γραμμή 47 του αρχείου [lm35Wtimer.ino](#).

```

48 {
49   reading = true;
50 }
```

Το διάγραμμα δείχνει από ποιες συναρτήσεις καλείται αυτή η συνάρτηση:





Κώδικας άσκησης *lm35Wtimer.ino*

```
1 #include "Timer.h"
2 #define TEMPIN A2
3
4 #define SPAN 100
5
6 volatile boolean reading = false;
7 int analog = 0;
8 int i = 0;
9 Timer t;
10
11 void setup()
12 {
13   Serial.begin(9600);
14   analogReference(INTERNAL);
15   t.every(1000, takeReading);
16 }
17
18 void loop()
19 {
20   t.update();
21   i++;
22   if (reading) {
23     for (i = 0; i < SPAN; i++) {
24       analog += analogRead(TEMPIN);
25     }
26     analog /= SPAN;
27     Serial.println(analog);
28     analog = 0;
29     reading = false;
30   }
31 }
32
33 void takeReading()
34 {
35   reading = true;
36 }
```

Κώδικας γραφικής εφαρμογής σε Processing 2.2.1

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)

```

112     {
113     if (theEvent.isController()) {
114         switch(theEvent.getController().getId()) {
115             case(1):
116                 try {
117                     if (b1.getBooleanValue()==true) {
118                         try
119                         {
120                             int index = int(d1.getValue());
121                             System.err.println(index);
122                             COMPort = new Serial(this, Serial.list()[index], 9600);
123                             b1.setCaptionLabel("Disconnect_COM");
124                             dweet("iot-uC-"+ID, "temperature", Float.toString((float)0),
"state", "1");
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                         dweet("iot-uC-"+ID, "temperature", Float.toString((float)0),
"state", "0");
136                         System.out.println("Connection Terminated. ");
137                     }
138                 }
139                 catch(Throwable e2)
140                 {
141                     System.err.println("Generic error: "+ e2);
142                 }
143                 break;
144             }
145         }
146     }

```

void draw ()

Η συγκεκριμένη συνάρτηση, εκτελείται σε επανάληψη

```

87     {
88         background(128);
89         fill(255);
90         textSize(26);
91         text("Arduino Temperature IOT", 300, 40);
92     }

```

void dweet (String *thingName*, String *temperature*, String *valueTemp*, String *state*, String *valueState*)

Η συγκεκριμένη συνάρτηση, κάθε φορά που καλείται φτιάχνει ένα JSON object της μορφής

```
{
  "this": "succeeded",
  "by": "dweeting",
  "the": "dweet",
  "with": {
    "thing": "my-thing-name",
    "created": "2014-01-15T17:28:42.556Z",
    "content": {
      "hello": "world",
      "foo": "bar"
    }
  }
}
```

Παράμετροι:

in	<i>thingName</i>	το όνομα του Dweet.io καναλιού
in	<i>temperature</i>	το όνομα του πεδίου
in	<i>valueTemp</i>	η τιμή του πεδίου
in	<i>state</i>	η κατάσταση της συσκευής
in	<i>valueState</i>	η τιμή της κατάστασης

```
37
{
  38   try {
  39
  40     JsonObject json= new JsonObject();
  41     json.addProperty(temperature, valueTemp);
  42     json.addProperty(state, valueState);
  43     DweetIO.publish(thingName, json);
  44   }
  45   catch (Exception e) {
  46   }
  47 }
```

void serialEvent (Serial *COMPort*)

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

```
98
99   String inString = COMPort.readStringUntil('\n');
100   if (inString != null) {
101     inString = trim(inString);
102     int data = int(inString);
103     inByte1 = data;
104     xPos = inByte1;
105     dweet("iot-uC-"+ID, "temperature", Float.toString((float)xPos/9.31),
"state", "1");
106   }
107 }
```

void setup ()

Αρχικοποίηση του γραφικού περιβάλλοντος Δημιουργεί ένα παράθυρο 800x70 με μία drop-down list και με ένα κουμπί για την απο/σύνδεση της σειριακής συσκευής

```
53     {
54         size(800, 70);
55         cp5 = new ControlP5(this);
56
57         b1 = cp5.addButton("Connect_COM")
58             .setColorBackground( color( #799195) )
59             .setPosition(160, 10)
60             .setSize(105, 20)
61             .setSwitch(false)
62             .setId(1)
63             .setOn()
64             ;
65
66         d1 = cp5.addDropdownList("Available COMs");
67         d1.clear();
68         int numCOM = Serial.list().length;
69         for (int i=0; i<numCOM; i++)
70         {
71             d1.addItem(Serial.list()[i], i);
72         }
73         d1.setPosition(30, 30);
74         d1.setSize(100, 160);
75         d1.setItemHeight(20);
76         d1.setBarHeight(20);
77         d1.getCaptionLabel().getStyle().marginTop = 3;
78         d1.getCaptionLabel().getStyle().marginLeft = 5;
79         d1.getCaptionLabel().getStyle().marginRight = 5;
80         d1.getValueLabel().getStyle().marginTop = 3;
81         d1.setColorBackground(color(#799195));
82         d1.setColorActive(color(255, 128));
83     }
```


Κώδικας άσκησης serial_plotter.pde

```

import com.google.gson.JsonObject;
import processing.serial.*;
import controlP5.*;
import io.dweet.DweetIO;

DropdownList d1;
Button b1;
ControlP5 cp5;
Serial COMPort;
double xPos = 1; // Initial position of pointer
int inByte1;
String ID = "AM";

/** @brief Η συγκεκριμένη συνάρτηση, κάθε φορά που καλείται φτιάχνει ένα JSON object της μορφής
{
  "this": "succeeded",
  "by": "dweeting",
  "the": "dweet",
  "with": {
    "thing": "my-thing-name",
    "created": "2014-01-15T17:28:42.556Z",
    "content": {
      "hello": "world",
      "foo": "bar"
    }
  }
}
@param[in] thingName το όνομα του Dweet.io καναλιού
@param[in] temperature το όνομα του πεδίου
@param[in] valueTemp η τιμή του πεδίου
*/
void dweet(String thingName, String temperature, String valueTemp, String state, String valueState) {
  try {

    JsonObject json= new JsonObject();
    json.addProperty(temperature, valueTemp);
    json.addProperty(state, valueState);
    DweetIO.publish(thingName, json);
  }
  catch(Exception e) {
  }
}

/** @brief Αρχικοποίηση του γραφικού περιβάλλοντος
Δημιουργεί ένα παράθυρο 800x70 με μία drop-down list
και με ένα κουμπί για την απο/σύνδεση της σειριακής συσκευής
*/
void setup() {
  size(800, 70);
  cp5 = new ControlP5(this);

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

```

```

        ;

d1 = cp5.addDropDownList("Available COMs");
d1.clear();
int numCOM = Serial.list().length;
for (int i=0; i<numCOM; i++)
{
    d1.addItem(Serial.list()[i], i);
}
d1.setPosition(30, 30);
d1.setSize(100, 160);
d1.setItemHeight(20);
d1.setBarHeight(20);
d1.getCaptionLabel().getStyle().marginTop = 3;
d1.getCaptionLabel().getStyle().marginLeft = 5;
d1.getCaptionLabel().getStyle().marginRight = 5;
d1.getValueLabel().getStyle().marginTop = 3;
d1.setColorBackground(color(#799195));
d1.setColorActive(color(255, 128));
}

/** @brief Η συγκεκριμένη συνάρτηση, εκτελείται σε επανάληψη
 */
void draw() {
    background(128);
    fill(255);
    textSize(26);
    text("Arduino Temperature IOT", 300, 40);
}

/** @brief Η συγκεκριμένη συνάρτηση, καλείται κάθε φορά που
 * καταφθάνουν δεδομένα στην Σειριακή Πόρτα. Διαβάζει δεδομένα
 μέχρι να συναντήσει χαρακτήρα αλλαγής γραμμής
 */
void serialEvent (Serial COMPort) {
    String inString = COMPort.readStringUntil('\n');
    if (inString != null) {
        inString = trim(inString);
        int data = int(inString);
        inByte1 = data;
        xPos = inByte1;
        dweet("iot-uC-"+ID, "temperature", Float.toString((float)xPos/9.31), "state", "1");
    }
}

/** @brief Η συγκεκριμένη συνάρτηση, καλείται κάθε φορά που
 * υπάρχει αλλαγή στα χειριστήρια της εφαρμογής (π.χ. στο κουμπί σύνδεσης της σειριακής)
 */
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");

```

```
        dweet("iot-uC-"+ID, "temperature", Float.toString((float)0), "state", "1");
        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();
        dweet("iot-uC-"+ID, "temperature", Float.toString((float)0), "state", "0");
        System.out.println("Connection Terminated. ");
    }
}
catch(Throwable e2)
{
    System.err.println("Generic error: "+ e2);
}
break;
}
}
}
```

Κώδικας *dashboard.json*

```

{
  "version":1,
  "allow_edit":true,
  "plugins":[
    "/plugins/all"
  ],
  "panes":[
    {
      "width":1,
      "row":{
        "3":1
      },
      "col":{
        "3":1
      },
      "col_width":"3",
      "widgets":[
        {
          "type":"text_widget",
          "settings":{
            "title":"Time",
            "size":"regular",
            "value":"datasources[\"Time\"][\"full_string_value\"]",
            "animate":true
          }
        }
      ]
    },
    {
      "title":"Temperature(°C)",
      "width":1,
      "row":{
        "3":5,
        "5":5
      },
      "col":{
        "3":2,
        "5":2
      },
      "col_width":"2",
      "widgets":[
        {
          "type":"sparkline_widget",
          "settings":{
            "title":"",
            "value":[
              "datasources[\"iot-uC-AM\"][\"temperature\"]"
            ],
            "include_legend":true,
            "legend":"Temperature(°C)"
          }
        }
      ]
    },
    {
      "title":"Device state",
      "width":1,
      "row":{

```

```

        "3":5,
        "6":5
    },
    "col":{
        "3":1,
        "6":1
    },
    "col_width":1,
    "widgets":[
        {
            "type":"indicator_widget",
            "settings":{
                "title": "",
                "value": "datasources[\"iot-uC-AM\"][\"state\"]",
                "on_text": "Connected",
                "off_text": "Disconnected"
            }
        }
    ]
},
"datasources":[
    {
        "name": "Time",
        "type": "plugin5319f8297df9e78c73000006",
        "settings": {
            "refresh": 1
        }
    },
    {
        "name": "iot-uC-AM",
        "type": "dweet_io",
        "settings": {
            "thing_id": "iot-uC-AM",
            "show_full": false,
            "name": "iot-uC-AM"
        }
    }
],
"columns": 3,
"pane_header_bg_color": null,
"pane_bg_color": null
}

```
