

Ψηφιακά Συστήματα Μετρήσεων

Arduino # Raspberry Pi
Processing # Python # MATLAB

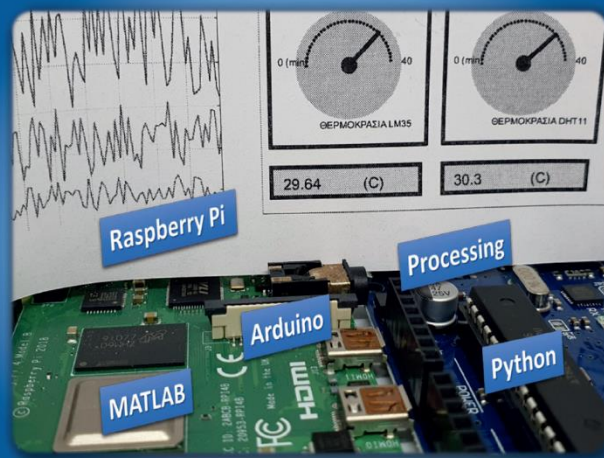
Το βιβλίο αυτό «απαντά» στις πραγματικές ανάγκες ενός μηχανικού, στην κατεύθυνση της ανάπτυξης ενός λειτουργικού και ολοκληρωμένου ψηφιακού συστήματος μετρήσεων. Παρουσιάζει τις επικρατέστερες τεχνολογίες και μεθοδολογίες για την ανάπτυξη των εφαρμογών. Δηλαδή, συγκεντρώνει γνώσεις με μια ενιαία αντίληψη και όχι αποσπασματικά, όπως έχουν συνηθίσει οι μηχανικοί σήμερα, που καλούνται να ανατρέξουν σε τελείως διαφορετικά βιβλία μεταξύ τους, μη γνωρίζοντας ακριβώς ποια εργαλεία θα πρέπει να χρησιμοποιήσουν.

Το βιβλίο είναι εστιασμένο στην ανάπτυξη λογισμικού, εκεί δηλαδή που βρίσκεται ο πυρήνας των σύγχρονων ψηφιακών συστημάτων μέτρησης.

Το βιβλίο συνοδεύεται από πλούσιο υλικό που είναι διαθέσιμο στην ιστοσελίδα panosparazoglou.gr

ΠΑΝΑΓΙΩΤΗΣ ΠΑΠΑΖΟΓΛΟΥ

Ψηφιακά Συστήματα Μετρήσεων



Διαφάνειες

Υλικό βιβλίου

Περισσότερο υλικό στο
panosparazoglou.gr

Κεφάλαιο 7



Απεικόνιση και επεξεργασία μετρήσεων σε υπολογιστή με το Processing

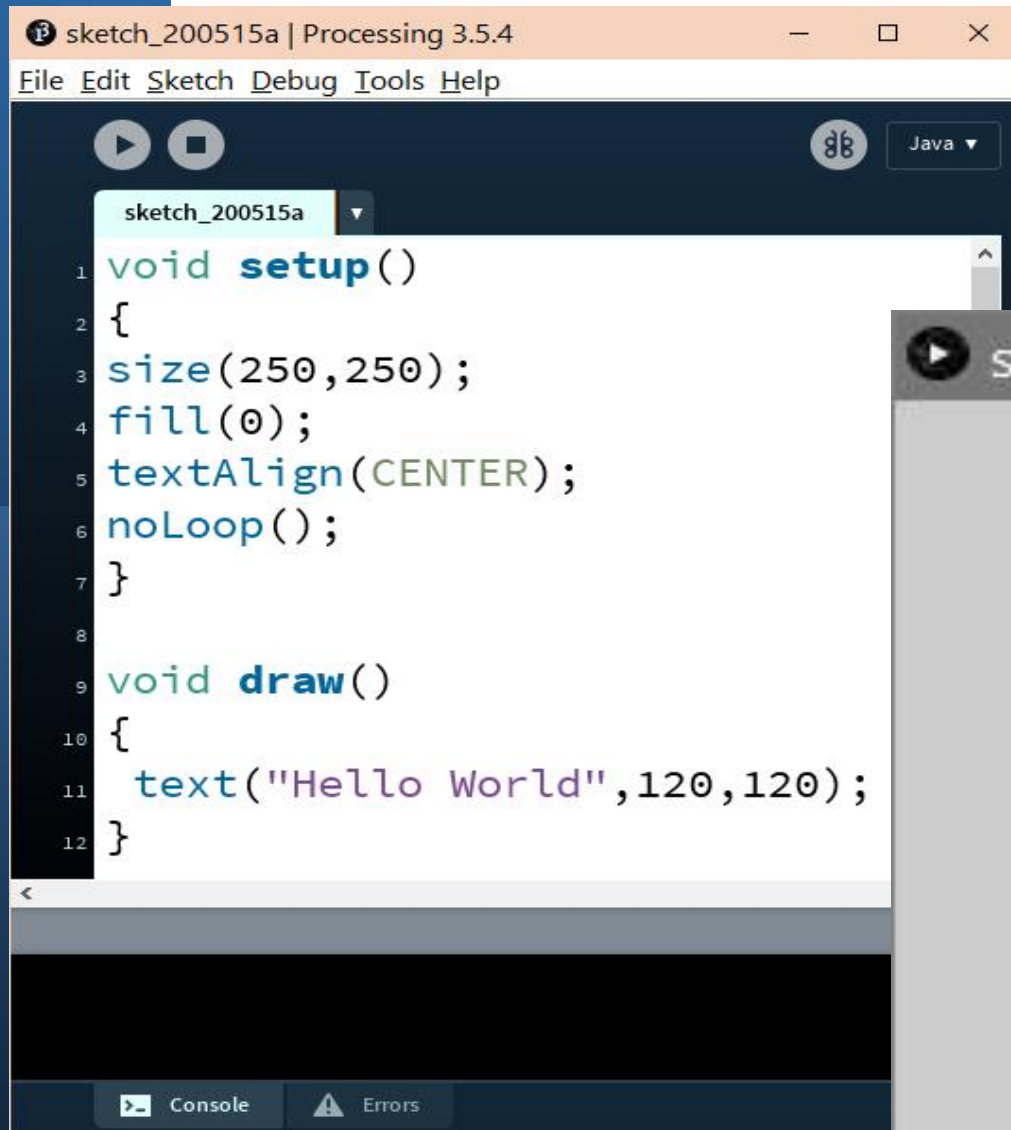


Δυνατότητες που πρέπει να έχει ένα λογισμικό

- Ανάγνωση μετρήσεων από τη μετρητική διάταξη (π.χ. Arduino)
- Απεικόνιση μετρήσεων σε πραγματικό χρόνο
- Αποθήκευση μετρήσεων
- Εφαρμογή φίλτρων και συναρτήσεων εξαγωγής δεδομένων
- Μετάδοση μετρήσεων (π.χ. στο διαδίκτυο)
- Γραφικό περιβάλλον διαχείρισης συστήματος μετρήσεων
- Διαδραστικότητα (έλεγχος μετρήσεων και συστήματος από το χρήστη)



Το περιβάλλον Processing



sketch_200515a | Processing 3.5.4

File Edit Sketch Debug Tools Help

```
1 void setup()  
2 {  
3   size(250,250);  
4   fill(0);  
5   textAlign(CENTER);  
6   noLoop();  
7 }  
8  
9 void draw()  
10 {  
11   text("Hello World",120,120);  
12 }
```

Console Errors



Βασικά συστατικά προγράμματος

Κάθε πρόγραμμα σε Processing, αποτελείται τουλάχιστον από τη συνάρτηση **setup()** και προαιρετικά την **draw()**. Όπως και στο Arduino, η συνάρτηση **setup()** εκτελείται μόνο μια φορά, ενώ η **draw()** εκτελείται επαναληπτικά.

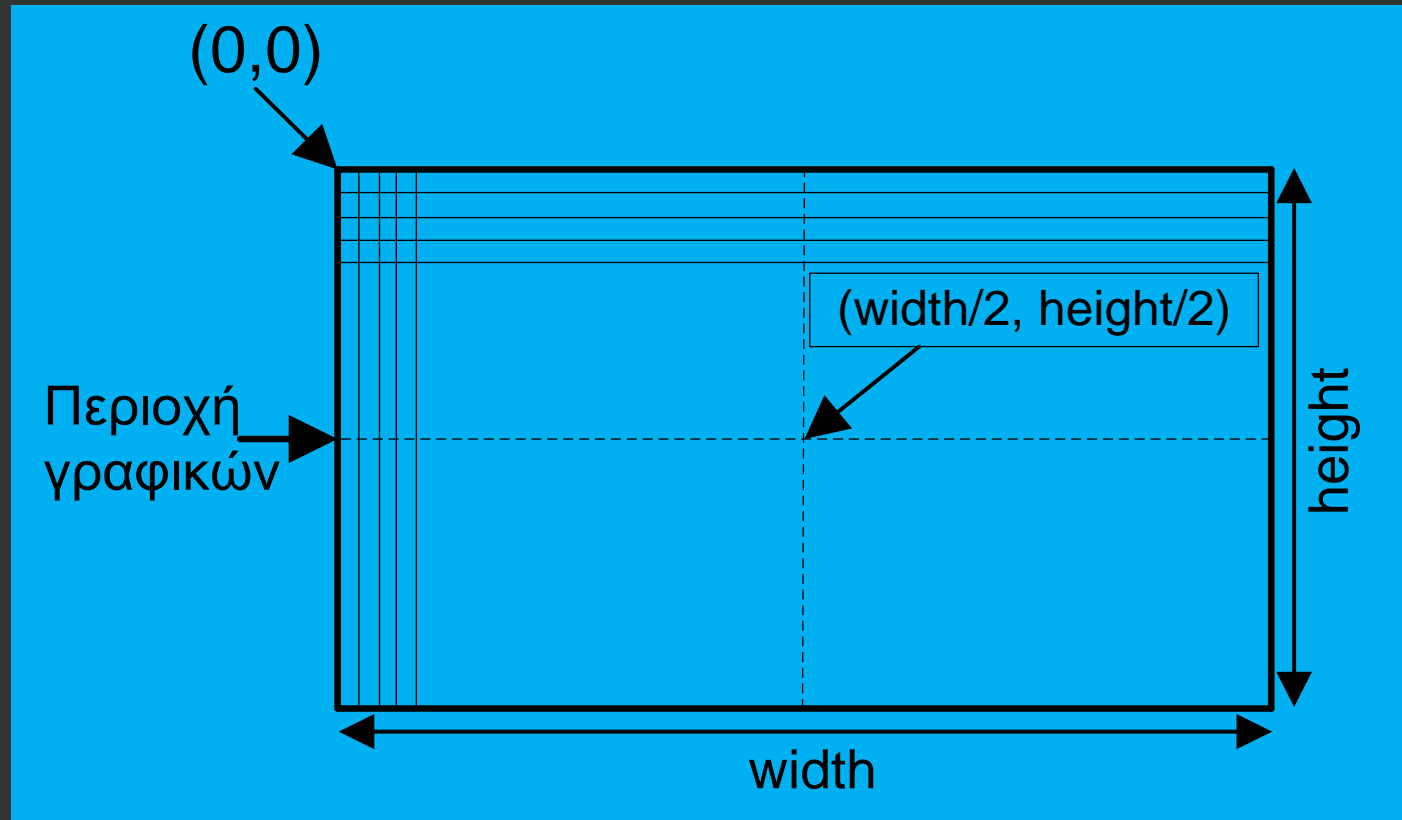
noLoop() = εκτέλεση **draw()** μόνο μια φορά.



```
sketch_200515a | Processing 3.5.4
File Edit Sketch Debug Tools Help

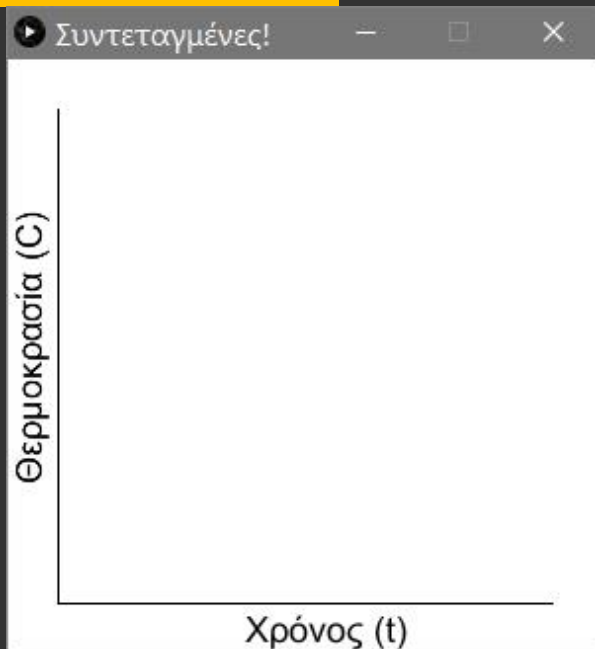
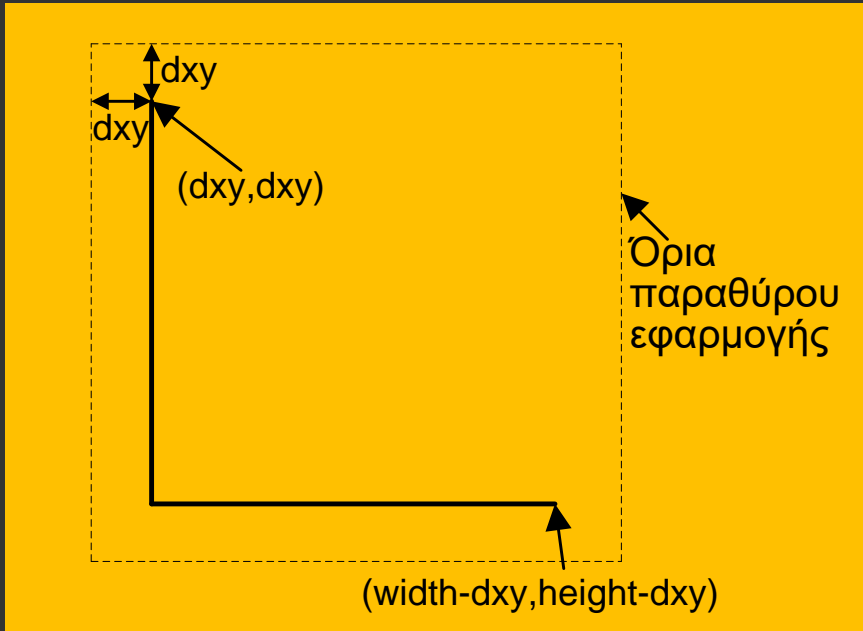
sketch_200515a
1 void setup()
2 {
3   size(250,250);
4   fill(0);
5   textAlign(CENTER);
6   noLoop();
7 }
8
9 void draw()
10 {
11   text("Hello World",120,120);
12 }
```

Σχεδίαση γραφικών



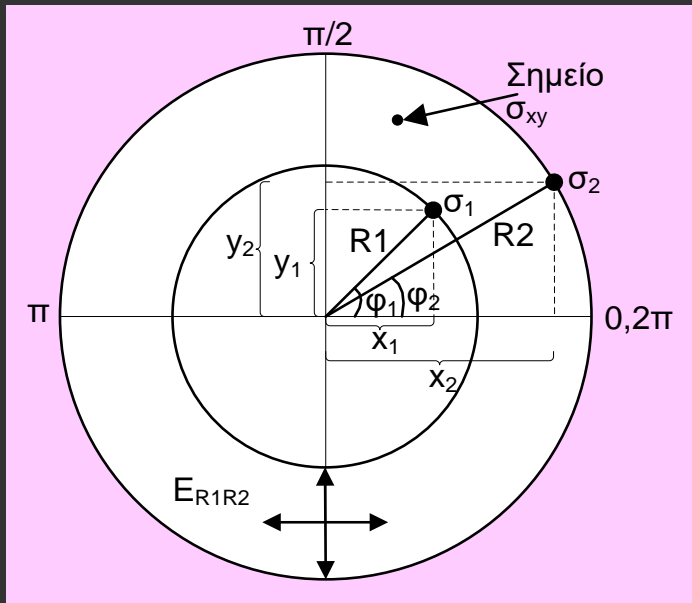
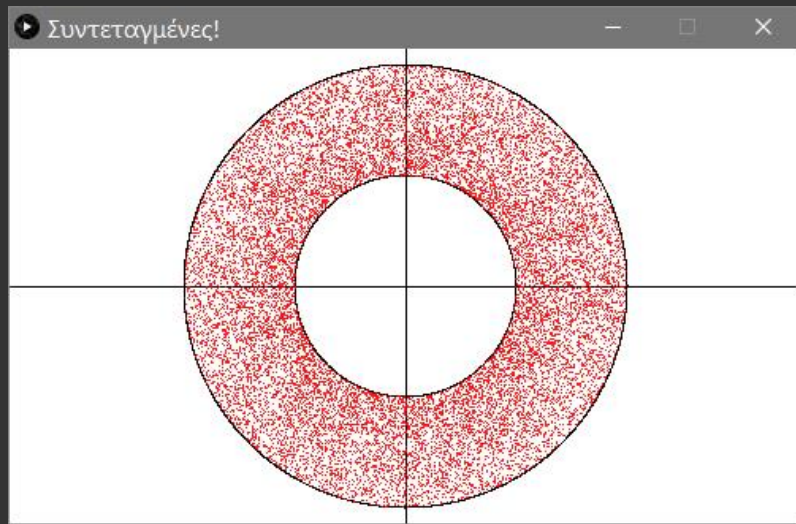
Το σύστημα συντεταγμένων στο Processing

Σχεδίαση αξόνων και εμφάνιση ετικετών



```
void setup()
{
  size(300,300);
  surface.setTitle("Συντεταγμένες!");
  background(255,255,255);
  fill(0);
  textFont(createFont("Arial",18));
  int dxy=25;
  line(dxy,dxy,dxy,height-dxy); //y-άξονας
  line(dxy,height-dxy,width-dxy,height-dxy); //x-άξονας
  text("Χρόνος (t)",(width/2)-30, height-5);
  translate(17,dxy+190);
  rotate(-PI/2);
  text("Θερμοκρασία (C)",0,0);
  rotate(PI/2);
  translate(-17,-dxy-190);
}
```


Σχεδίαση αξόνων και εμφάνιση ετικετών



```
int R1=70, R2=140;
```

```
void setup()
```

```
{
```

```
size(500,300);
```

```
surface.setTitle("Συντεταγμένες!");
```

```
background(255,255,255);
```

```
line(width/2,0,width/2, height); //y-άξονας
```

```
line(0,height/2,width,height/2); //x-άξονας
```

```
stroke(0,0,0);
```

```
for(float R=0; R<=2*PI; R+=0.001)
```

```
{
```

```
point((width/2)+R1*sin(R), (height/2)+R1*cos(R));
```

```
point((width/2)+R2*sin(R), (height/2)+R2*cos(R));
```

```
}
```

```
stroke(255,0,0);
```

```
}
```

```
void draw()
```

```
{
```

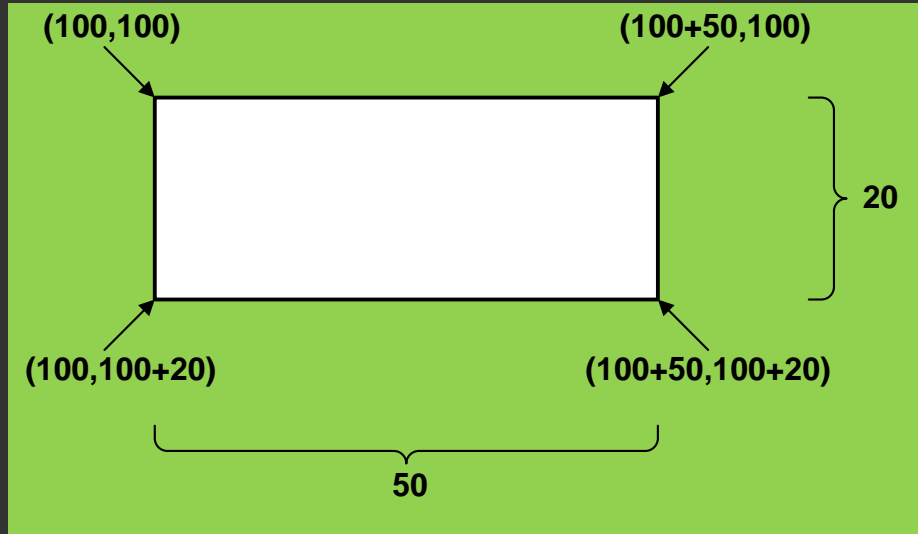
```
float rth=random(0,2*PI);
```

```
int
```

```
rrad=round(random(R1,R2));
```

```
point((width/2)+rrad*sin(rth), (height/2)+rrad*cos(rth));
```

```
}
```



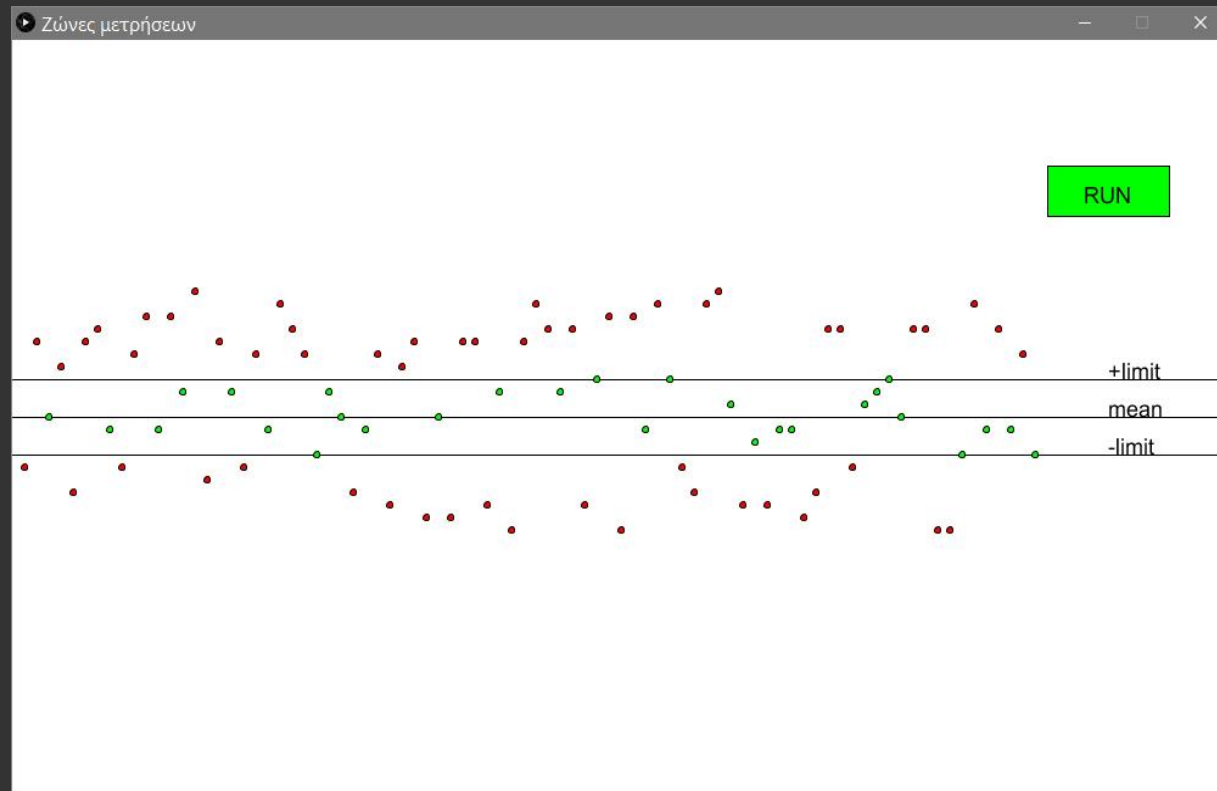
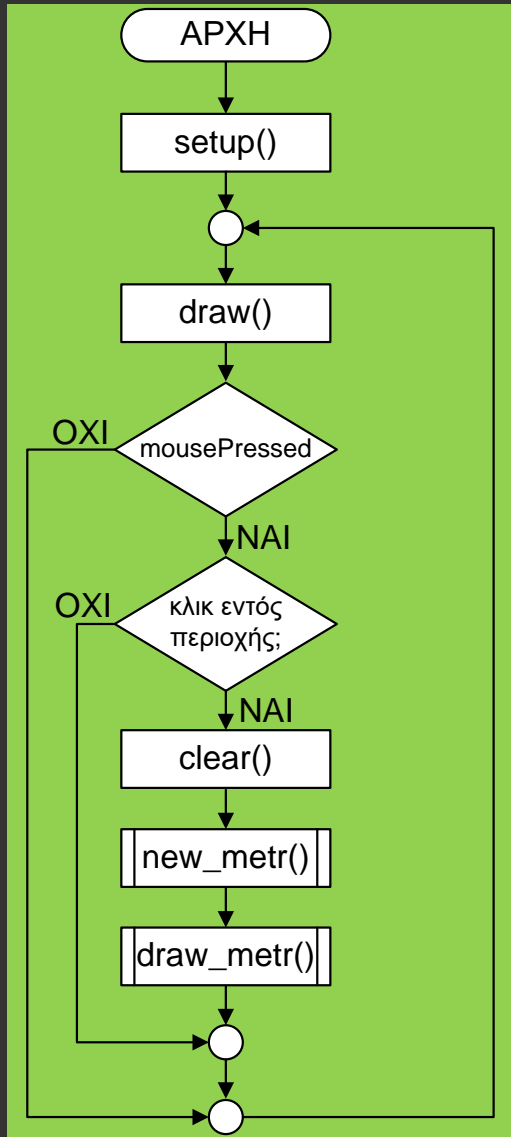
Περιοχή κουμπιού

Δημιουργία κουμπιού (1)

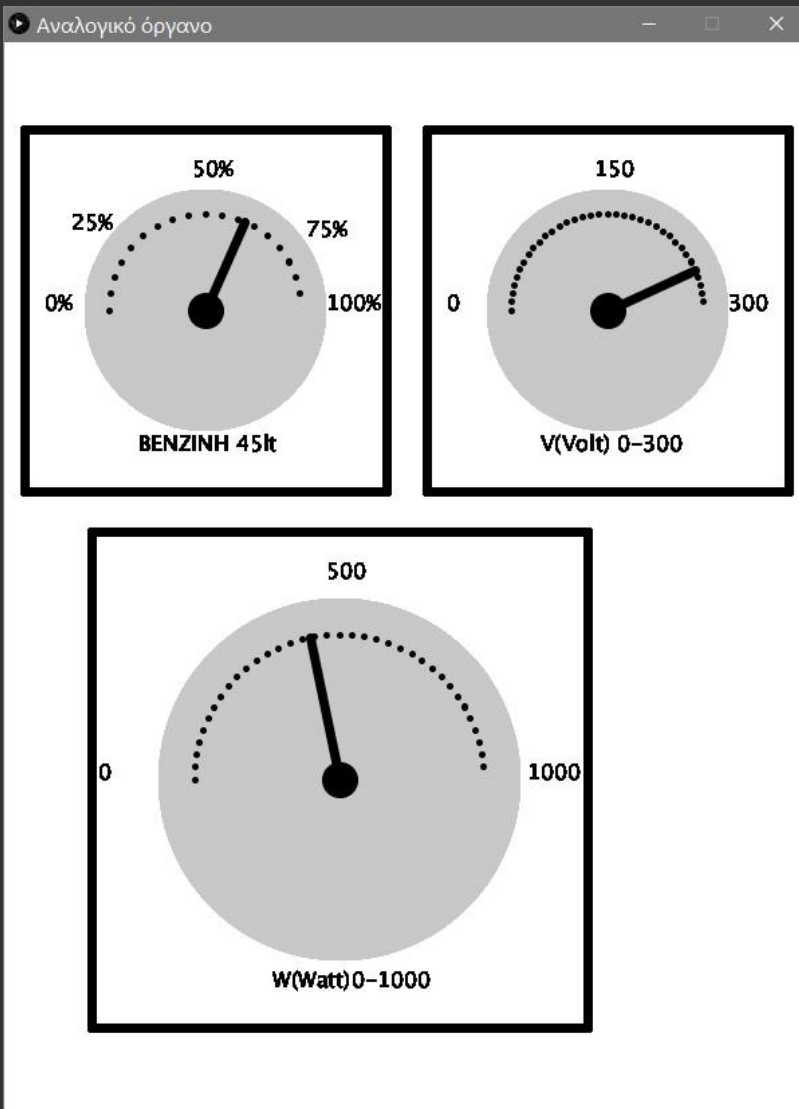
```
if (mousePressed)
  if (mouseX>bx && mouseX<bx+bw && mouseY>by && mouseY<by+bh)
  {
    //Εντολές που θα εκτελεστούν αν έγινε κλικ εντός της παραπάνω περιοχής
  }
```

Δημιουργία κουμπιού (2)

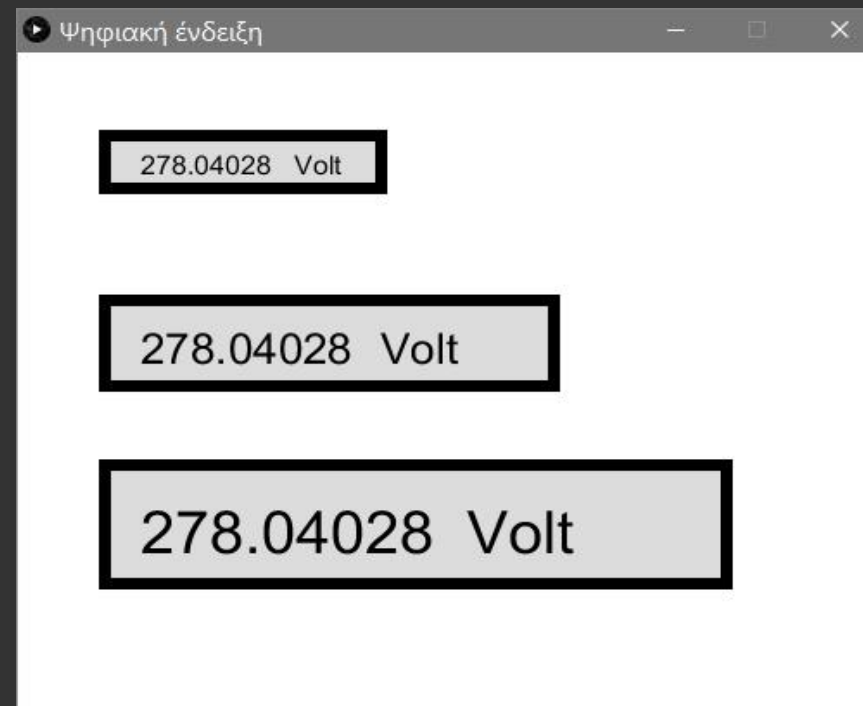
Δείτε τον κώδικα της εφαρμογής 7.5



Εικονικά αναλογικά όργανα



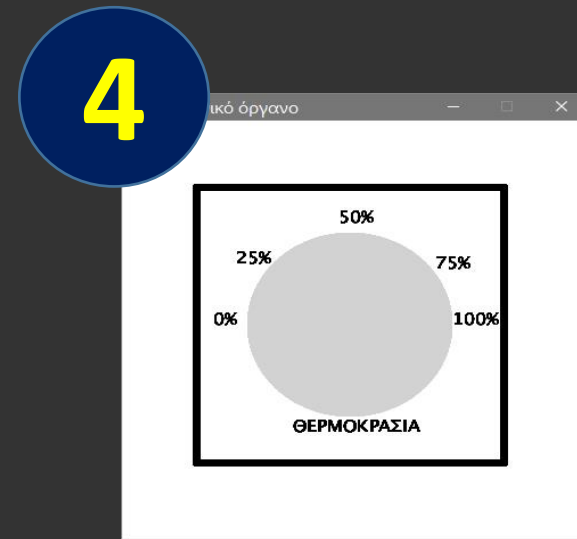
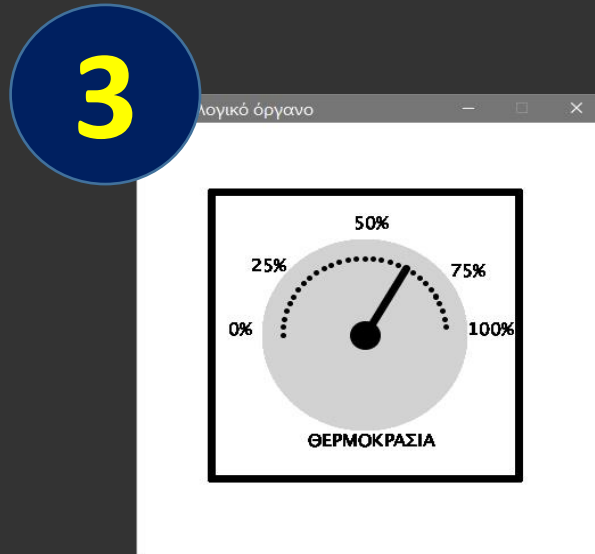
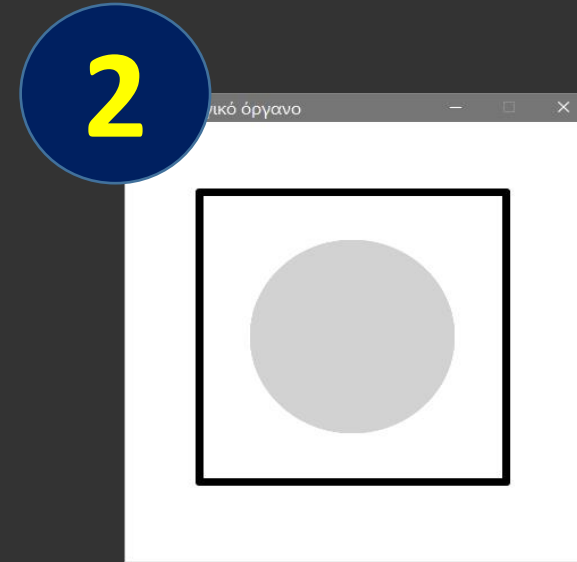
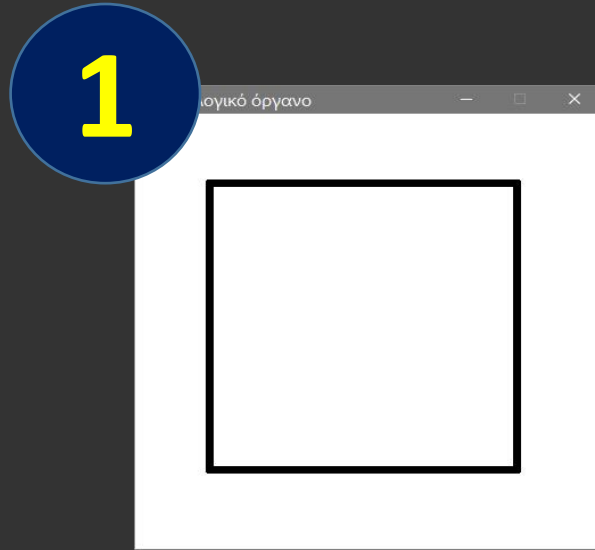
- Αποτύπωση μετρήσεων (διάγραμμα) σε πραγματικό χρόνο
- Ρυθμιστικά ελέγχου (π.χ. κουμπιά, αναλογικές μπάρες)
- Εικονικά όργανα μετρήσεων



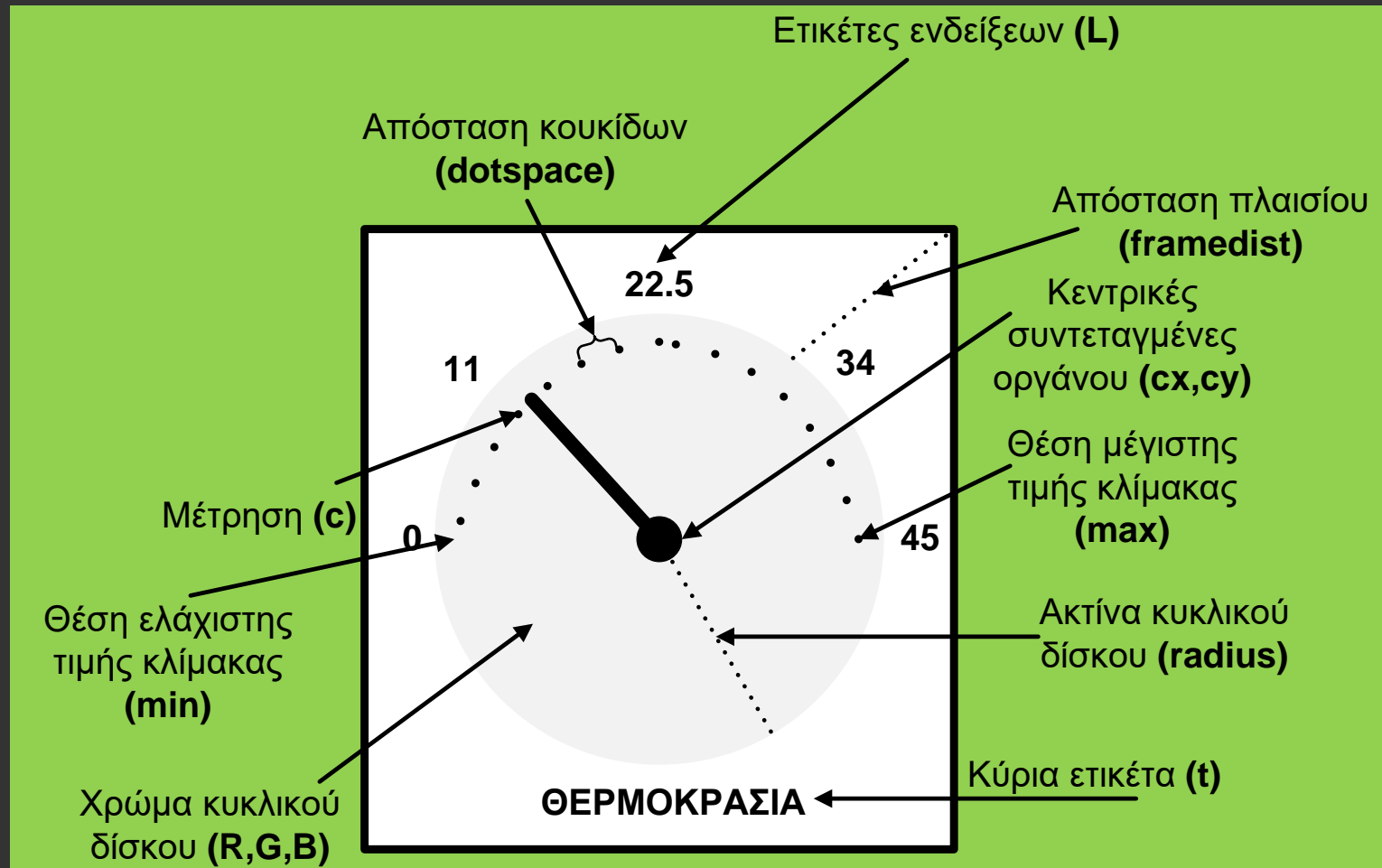
Σχεδίαση εικονικού αναλογικού οργάνου



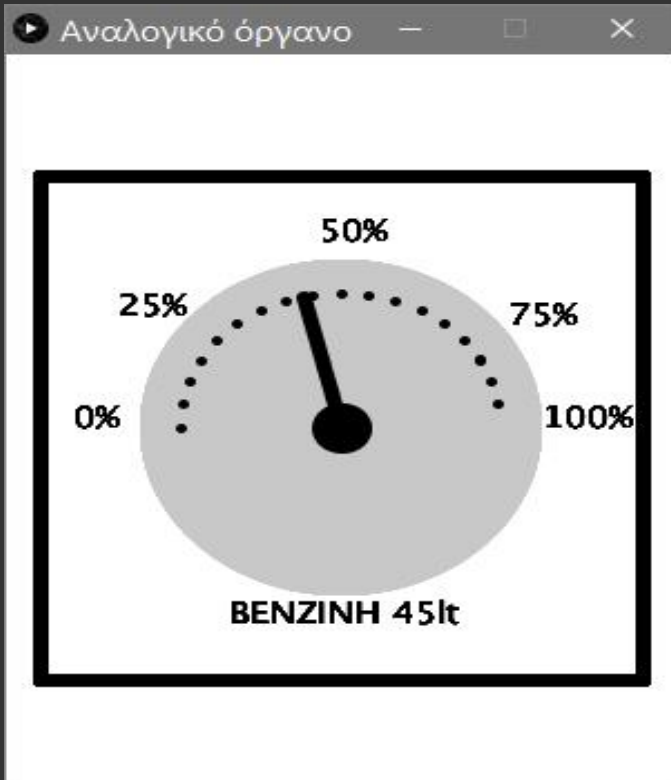
Δείτε την εφαρμογή 7.6



Συνάρτηση σχεδίασης εικονικού αναλογικού οργάνου (1)



Συνάρτηση σχεδίασης εικονικού αναλογικού οργάνου (2)



Παράδειγμα κλήσης

```
String[] label1={"0%","25%","50%","75%","100%"};
```

Καθορίζουμε τις βοηθητικές ετικέτες για να τις «στείλουμε» όλες μαζί στη συνάρτηση

```
draw_analog(150,200,100,m1, "BENZINH 45lt",200,200,200,0,45,10,label1,35);
```

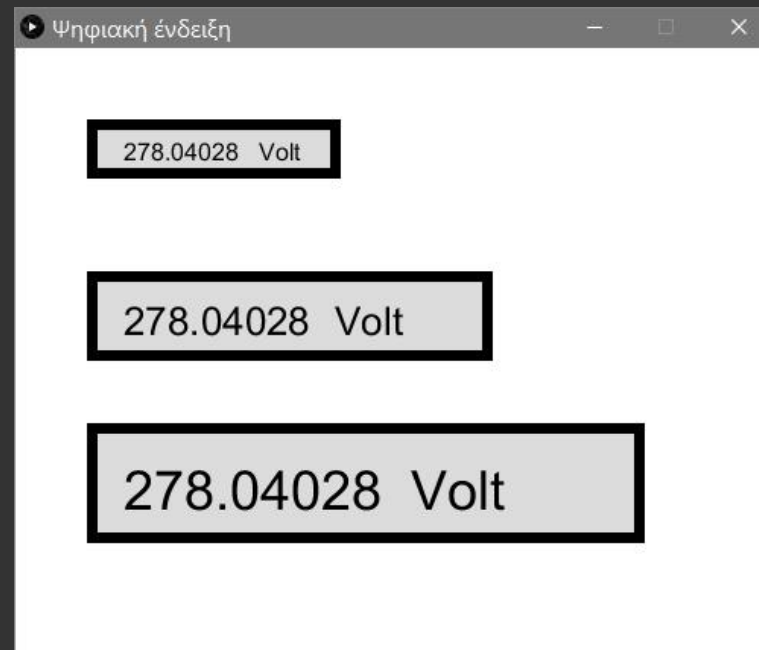
Συνάρτηση σχεδίασης εικονικού ψηφιακού οργάνου (1)

```
void draw_digital(int x, int y, int frameline, int digits, String font, int  
fontsize, float m, String type, int Rf, int Gf, int Bf, int Rm, int Gm, int Bm)
```

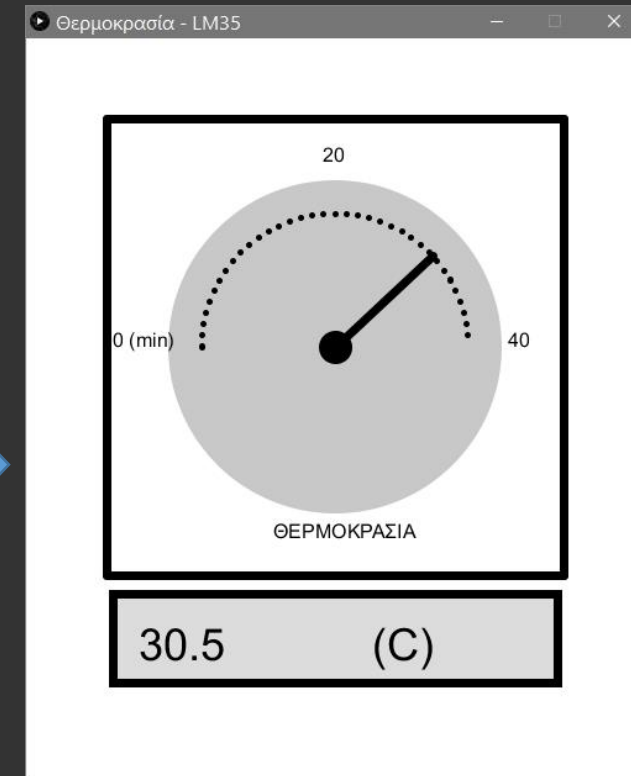
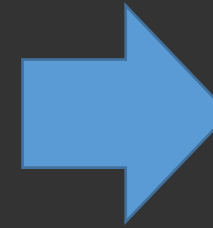
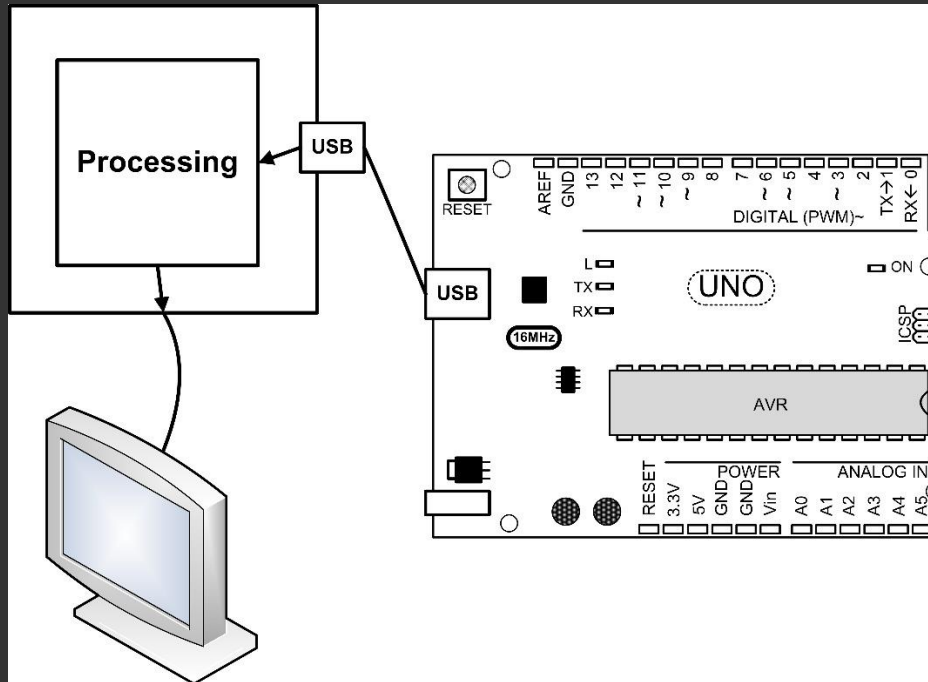
x, y	Συντεταγμένες που θα ξεκινήσει η εμφάνιση (θυμηθείτε ότι εμφανίζουμε στην πραγματικότητα κείμενο και όχι απλά ένα σημείο στο χώρο)
frameline	Πάχος γραμμής πλαισίου
digits	Πλήθος ψηφίων προς εμφάνιση. Αυτό είναι απλά ενδεικτικό για να γίνει υπολογισμός του μήκους του πλαισίου
font	Ονομασία γραμματοσειράς που θα χρησιμοποιηθεί
fontsize	Μέγεθος γραμματοσειράς
m	Η μέτρηση που πρόκειται να εμφανιστεί στο πλαίσιο
type	Μονάδες που θα συνοδεύουν τη μέτρηση (στα δεξιά)
Rf, Gf, Bf	Εσωτερικό φόντο πλαισίου
Rm, Gm Bm	Χρώμα εμφάνισης μέτρησης

Συνάρτηση σχεδίασης εικονικού ψηφιακού οργάνου (2)

```
float m3=random(0,1000);  
draw_digital((width/2)-200,50, 7, 5, "Arial", 16, m3, "Volt", 220,220,220,0,0,0);  
draw_digital((width/2)-200,150,7, 5, "Arial", 26, m3, "Volt", 220,220,220,0,0,0);  
draw_digital((width/2)-200,250,7, 5, "Arial", 36, m3, "Volt", 220,220,220,0,0,0);
```

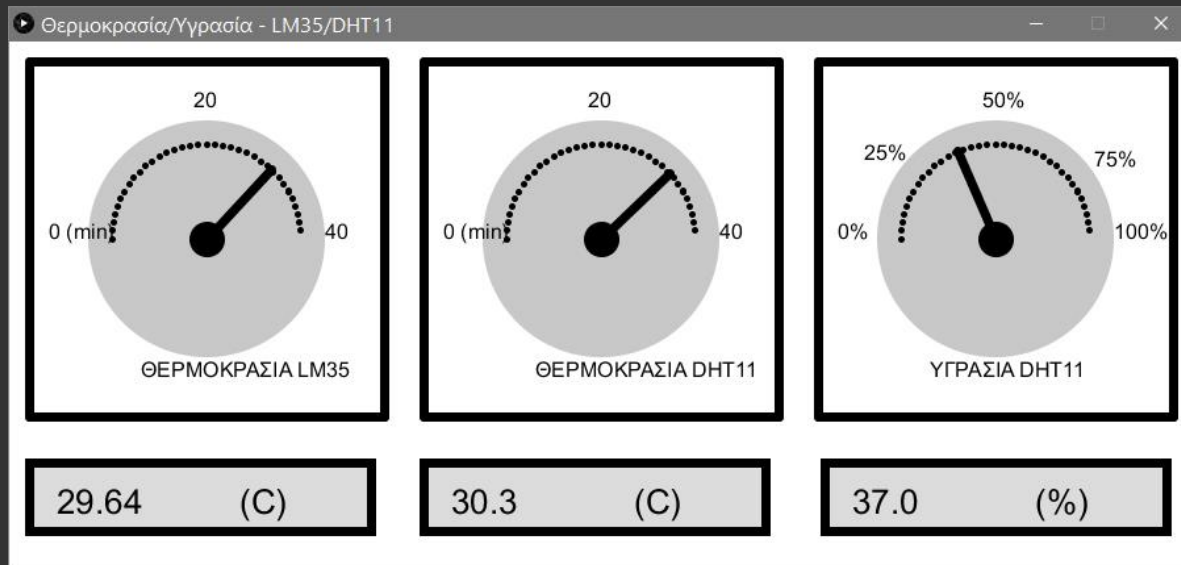


Απεικόνιση μετρήσεων από το Arduino



**Δείτε την
εφαρμογή 7.9**

Απεικόνιση μετρήσεων σε πάνελ οργάνων από το Arduino



**Δείτε την
εφαρμογή 7.10**

```
#include "DHT.h"  
#define DHTPIN 2  
#define DHTTYPE DHT11
```

```
DHT dht(DHTPIN, DHTTYPE);
```

```
void setup()
```

```
{  
  analogReference(INTERNAL);  
  Serial.begin(9600);  
  dht.begin();  
}
```

```
void loop()
```

```
{  
  int a=analogRead(0);  
  float temp=(a*1.07)/10;  
  float h = dht.readHumidity();  
  float t = dht.readTemperature();  
  Serial.print(temp); Serial.print(",");  
  Serial.print(t); Serial.print(","); Serial.println(h);  
  delay(2000);  
}
```

**Κώδικας
Arduino**