

Προγραμματίζοντας τον arduino



Προγραμματίζοντας τον arduino

Η γλώσσα:

- μοιάζει πολύ με τη C
- είναι υψηλού επιπέδου
- πολύ πιο εύκολη από την assembly
- πλήρως δομημένη σε υπορουτίνες

Προγραμματίζοντας τον arduino

δύο απαραίτητες υπορουτίνες: setup() , loop()

//παράδειγμα

```
void setup() {  
    pinMode(13, OUTPUT);  
}  
void loop() {  
    digitalWrite(13, HIGH);  
    delay(1000);  
    digitalWrite(13, LOW);  
    delay(1000);  
}
```

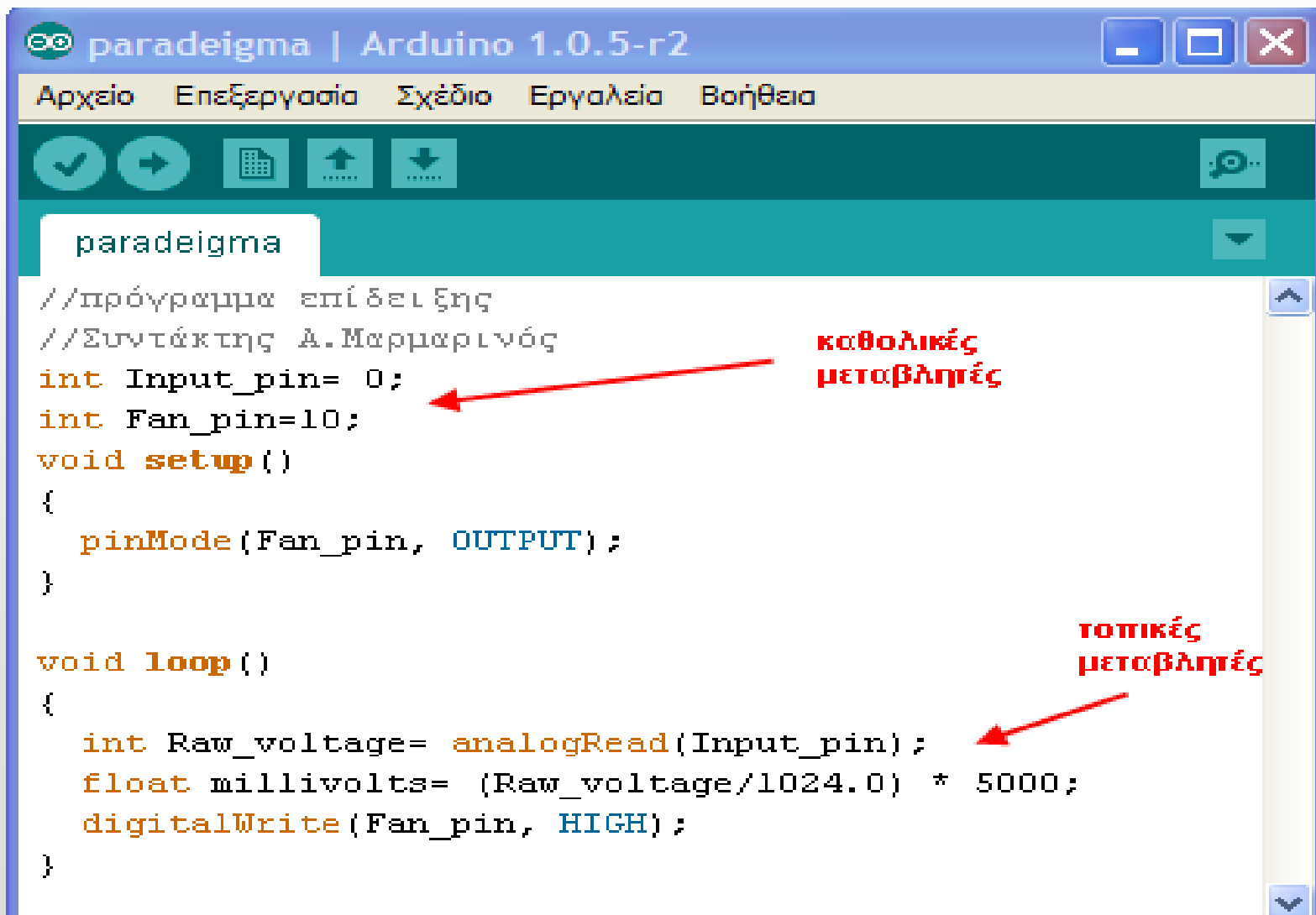
- τι σημαίνουν τα // () {} ;
- τι σημαίνει void;

Προγραμματίζοντας τον arduino

Οι μεταβλητές

- Δηλώνονται οπουδήποτε, ανάλογα με το είδος τους (τοπικές, καθολικές)
- Προηγείται ο τύπος της μεταβλητής και ακολουθεί το όνομα (π.χ. `int x;`)
- Δυνατότητα αρχικοποίησης κατά τη δήλωση (π.χ. `int x=0;`)
- Περιορισμοί στο όνομα

Προγραμματίζοντας τον arduino



```

paradeigma | Arduino 1.0.5-r2
Αρχείο  Επεξεργασία  Σχέδιο  Εργαλεία  Βοήθεια

paradeigma
//πρόγραμμα επίδειξης
//Συντάκτης Α.Μαρμαρινός
int Input_pin= 0;
int Fan_pin=10;
void setup()
{
  pinMode(Fan_pin, OUTPUT);
}

void loop()
{
  int Raw_voltage= analogRead(Input_pin);
  float millivolts= (Raw_voltage/1024.0) * 5000;
  digitalWrite(Fan_pin, HIGH);
}
  
```

καθολικές μεταβλητές

τοπικές μεταβλητές

Προγραμματίζοντας τον arduino

Σημαντικές εντολές

- **pinMode** (pin, INPUT/OUTPUT) ;

Π.χ. **pinMode** (13, OUTPUT) ;

- **digitalWrite** (pin, HIGH/LOW) ;

Π.χ. **digitalWrite** (13, HIGH) ;

- **analogWrite** (pin, value)

αναλογική έξοδος PWM στα ψηφιακά pins (3, 5, 6, 9, 10, 11)

Value: από 0 ως 255

Π.χ. **analogWrite** (10, 255) ;

Αναλογική και
ψηφιακή έξοδος
δεδομένων

Προγραμματίζοντας τον arduino

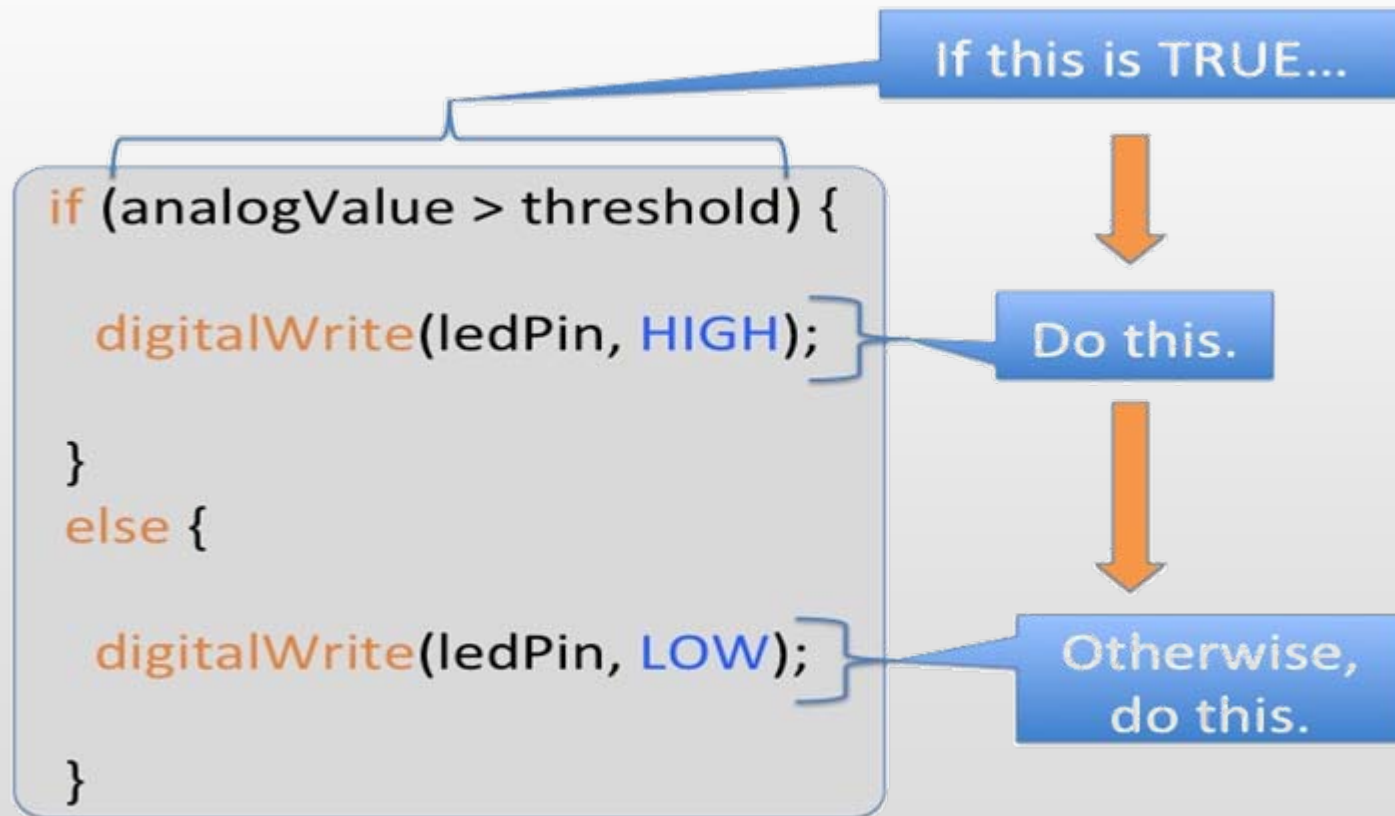
Σημαντικές εντολές (2)

- `int val = analogRead(pin)`
π.χ. `int voltage = analogRead(2);`
- `int val = digitalRead(pin)`
π.χ. `int pir = digitalRead(2);`
- `delay(time_ms);`
π.χ. `delay(2000);`
- `Serial.begin(bps), Serial.println(var)`

Αναλογική και
ψηφιακή είσοδος
δεδομένων
[A0-A5] [D0-D13]

Προγραμματίζοντας τον arduino

Δομές ελέγχου ροής: **If**



Προγραμματίζοντας τον arduino

Δομές ελέγχου ροής: **If**

τελεστής	περιγραφή
() == ()	είναι ίσα?
() != ()	είναι άνισα?
() > ()	μεγαλύτερο
() >= ()	μεγαλύτερο ή ίσο
() < ()	μικρότερο
() <= ()	Μικρότερο ή ίσο

ποια η διαφορά;

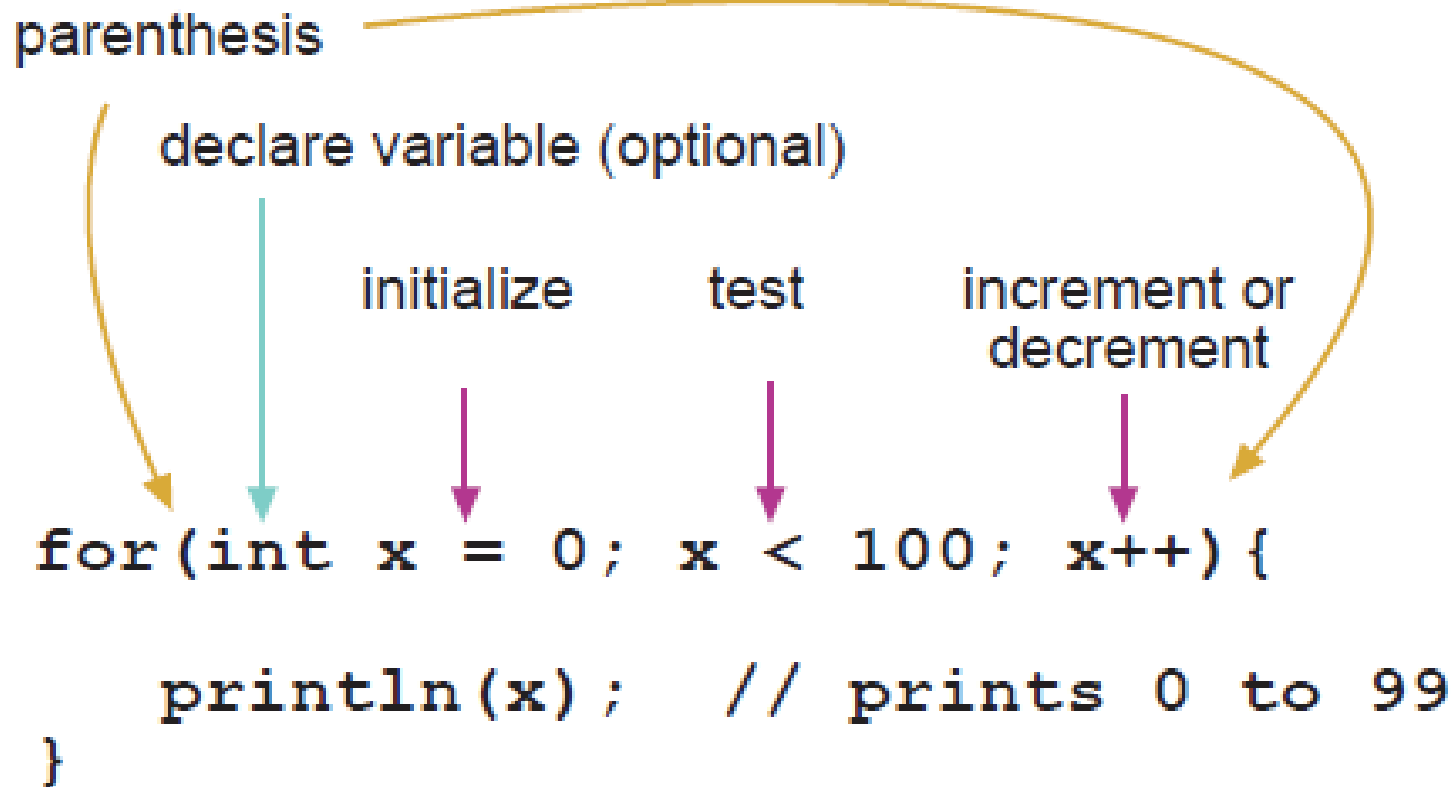
if (x==10)

if (x=10)

```
if (value == 10)
{ // do something
}
else
{ // do something else
}
```

Προγραμματίζοντας τον arduino

Δομές επανάληψης: **for**

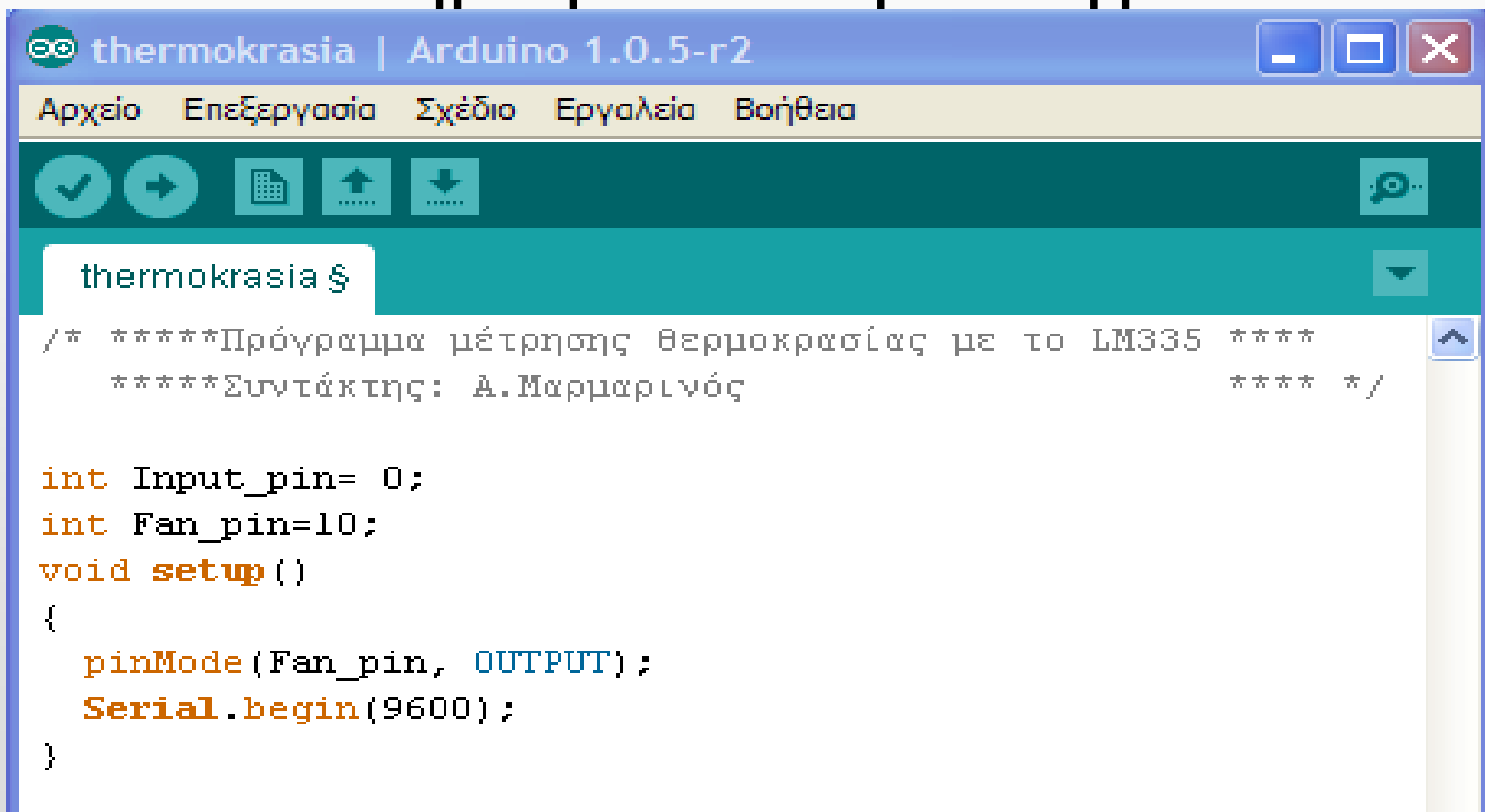


The diagram illustrates the components of a C++ for loop. A code snippet is shown with arrows pointing to its parts: a blue arrow points from 'parenthesis' to the opening parenthesis; a light blue arrow points from 'declare variable (optional)' to 'int x'; a purple arrow points from 'initialize' to '= 0'; another purple arrow points from 'test' to '< 100'; a third purple arrow points from 'increment or decrement' to 'x++'; and a long orange arrow points from the closing parenthesis back to the opening parenthesis, indicating the loop's repetition.

```
for(int x = 0; x < 100; x++) {  
    println(x); // prints 0 to 99  
}
```

Προγραμματίζοντας τον arduino

Ένα ολοκληρωμένο παράδειγμα



The screenshot shows the Arduino IDE interface with the file 'thermokrasia' open. The menu bar includes 'Αρχαίο', 'Επεξεργασία', 'Σχέδιο', 'Εργαλεία', and 'Βοήθεια'. The toolbar contains icons for checking, running, uploading, and downloading. The code editor displays a C++ program for temperature measurement using an LM335 sensor. The code includes comments in Greek, variable declarations for 'Input_pin' and 'Fan_pin', and a 'setup' function that configures the fan pin as an output and initializes the serial port at 9600 baud.

```
thermokrasia $  
  
/* *****Πρόγραμμα μέτρησης θερμοκρασίας με το LM335 *****  
   *****Συντάκτης: Α.Μαρμαρινός ***** */  
  
int Input_pin= 0;  
int Fan_pin=10;  
void setup()  
{  
    pinMode(Fan_pin, OUTPUT);  
    Serial.begin(9600);  
}
```

Προγραμματίζοντας τον arduino

Ένα ολοκληρωμένο παράδειγμα (συνέχεια)

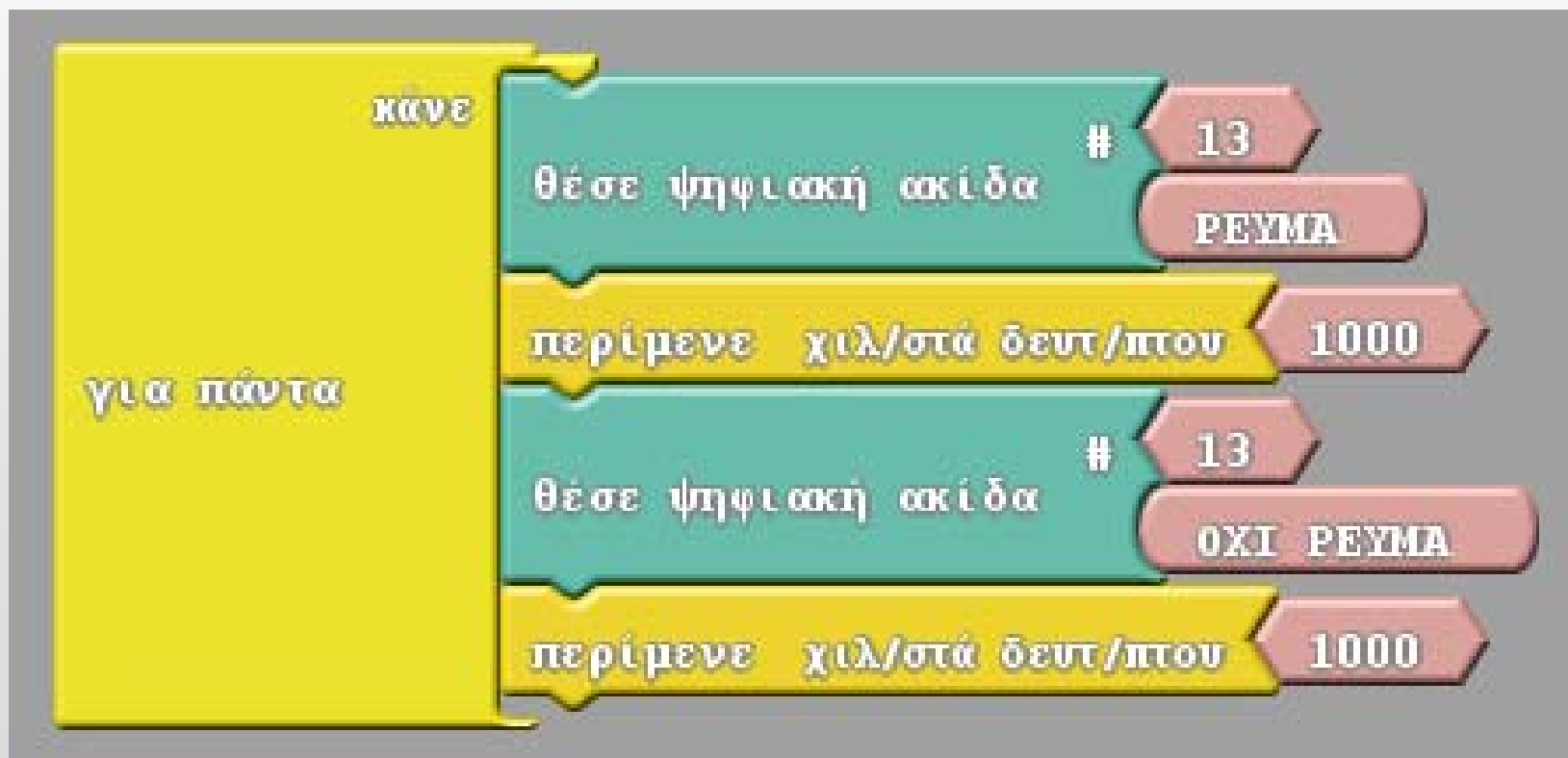
```
void loop()
{
  int Raw_voltage= analogRead(Input_pin);
  float millivolts= (Raw_voltage/1024.0) * 5000;
  float kelvin= (millivolts/10);
  float celsius= kelvin - 273.15;
  Serial.print(celsius);
  Serial.println(" degrees Celsius");

  if (celsius>22)
    digitalWrite(Fan_pin, HIGH);
  else
    digitalWrite(Fan_pin, LOW);
  delay(1000);
}
```

Προγραμματίζοντας τον arduino

Πολύ απλή γλώσσα γραφικού προγραμματισμού:

ArduBlock



Προγραμματίζοντας τον arduino

Ερωτήσεις