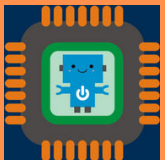


Ενίσχυση των δεξιοτήτων προγραμματισμού με μικροελεγκτές

Προτάθηκε από τη Mirela TIBU

“Grigore Moisil” Theoretical Highschool
of Informatics , Iasi, Romania



Μια Εργαλειοθήκη για την προώθηση της χρήσης
Δεξιοτήτων STEM αξιοποιώντας Εφαρμογές
Μικροελεγκτών



Co-funded by the
Erasmus+ Programme
of the European Union

Project No. 2019-1-RO01-KA202-063965

Το έργο αυτό χρηματοδοτήθηκε με την υποστήριξη της Ευρωπαϊκής Επιτροπής. Το περιεχόμενο του παρόντος εγγράφου αντικατοπτρίζει μόνο τις απόψεις της συντακτικής του ομάδας και η Επιτροπή δε μπορεί να θεωρηθεί υπεύθυνη για οποιαδήποτε χρήση των πληροφοριών που περιέχονται σε αυτό.

Περιεχόμενα

Επιστήμη των υπολογιστών

Σκοπός
Περιγραφή
Μαθησιακοί Στόχοι
Μεθοδολογίες Μάθησης
Ομάδα-στόχος
Μικροεπεξεργαστής VS Μικροελεγκτής
Αρχιτεκτονική Ενσωματωμένων Συστημάτων
Κατανόηση Προγραμματισμού
Έννοιες που χρησιμοποιούν μικροελεγκτές
Επιστημονικοί τομείς που καλύπτονται
Εκτίμηση
Βιβλιογραφία



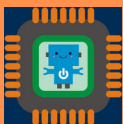
ΣΚΟΠΟΣ

Έκφραση δημιουργικού τρόπου σκέψης, στη δόμηση και επίλυση προβλημάτων

Διαμόρφωση συνηθειών για τη χρήση συγκεκριμένων αλγοριθμικών εννοιών και μεθόδων υπολογιστή για την προσέγγιση ποικίλων προβλημάτων

Εκδήλωση στάσεων απέναντι στην επιστήμη και τη γνώση

Εκδήλωση πρωτοβουλίας και διάθεση για αντιμετώπιση διαφόρων εργασιών



Μια Εργαλειοθήκη για την προώθηση της χρήσης
Δεξιοτήτων STEM αξιοποιώντας Εφαρμογές Μικροελεγκτών

Project No. 2019-1-RO01-KA202-063965

Το έργο αυτό χρηματοδοτήθηκε με την υποστήριξη της Ευρωπαϊκής Επιτροπής. Το περιεχόμενο του παρόντος εγγράφου αντικατοπτρίζει μόνο τις απόψεις της συντακτικής του ομάδας και η Επιτροπή δε μπορεί να θεωρηθεί υπεύθυνη για οποιαδήποτε χρήση των πληροφοριών που περιέχονται σε αυτό.



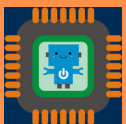
Co-funded by the
Erasmus+ Programme
of the European Union

Περιγραφή

- Προσέγγιση ITC - Μικροεπεξεργαστής VS Μικροελεγκτής
 - ✓ Εντοπισμός περιοχών όπου χρησιμοποιούνται υπολογιστές / ενσωματωμένα συστήματα στην καθημερινή ζωή
 - ✓ Περιγραφή της αρχιτεκτονικής υλικού για έναν υπολογιστή και ένα ενσωματωμένο σύστημα
 - ✓ Σύγκριση χαρακτηριστικών μικροεπεξεργαστών και μικροελεγκτών
- Προγραμματιστική προσέγγιση - Κατανόηση Προγραμματιστικών Εννοιών με χρήση εφαρμογών Microcontrollers
 - ✓ Δομημένες δηλώσεις προγραμματισμού – αποφάσεις, βρόχοι (IF, WHILE, FOR)
 - ✓ Δήλωση & κλήση άκυρων και μη άκυρων συναρτήσεων
 - ✓ Χρησιμοποίηση πινάκων σε εφαρμογές
 - ✓ Ανάλυση της λειτουργικότητας των συσκευών Arduino για την αναγνώριση των βημάτων εκτέλεσης των δηλώσεων προγραμματισμού



- Προσδιορισμός εφαρμογών υπολογιστών στην κοινωνική ζωή - συνειδητοποίηση του αντίκτυπου των ενσωματωμένων συστημάτων στην καθημερινή ζωή
 - Προσδιορισμός των ομοιοτήτων και των διαφορών μεταξύ ενός μικροεπεξεργαστή και ενός μικροελεγκτή στην αρχιτεκτονική των υπολογιστών και των ενσωματωμένων συστημάτων.
 - Εξάσκηση στην εφαρμογή των στοιχείων του δομημένου προγραμματισμού - αποφάσεις, βρόχοι, συναρτήσεις. αναπαράσταση και χρήση δεδομένων πίνακα.
 - Οπτικοποίηση του αποτελέσματος της εκτέλεσης διαφόρων ακολουθιών προγραμμάτων μέσω συσκευών που βασίζονται σε μικροελεγκτή.



Μεθοδολογίες Μάθησης

- Εξήγηση
- Επίδειξη
- Συνομιλία
- Αλγόριθμος
- Υλοποιήσεις



Μια Εργαλειοθήκη για την προώθηση της χρήσης
Δεξιοτήτων STEM αξιοποιώντας Εφαρμογές Μικροελεγκτών

Project No. 2019-1-RO01-KA202-063965

Το έργο αυτό χρηματοδοτήθηκε με την υποστήριξη της Ευρωπαϊκής Επιτροπής. Το περιεχόμενο του παρόντος εγγράφου αντικατοπτρίζει μόνο τις απόψεις της συντακτικής του ομάδας και η Επιτροπή δε μπορεί να θεωρηθεί υπεύθυνη για οποιαδήποτε χρήση των πληροφοριών που περιέχονται σε αυτό.



Co-funded by the
Erasmus+ Programme
of the European Union

Ομάδα-στόχος

Μαθητές Λυκείου



Μια Εργαλειοθήκη για την προώθηση της χρήσης
Δεξιοτήτων STEM αξιοποιώντας Εφαρμογές Μικροελεγκτών

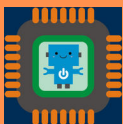
Project No. 2019-1-RO01-KA202-063965

Το έργο αυτό χρηματοδοτήθηκε με την υποστήριξη της Ευρωπαϊκής Επιτροπής. Το περιεχόμενο του παρόντος εγγράφου αντικατοπτρίζει μόνο τις απόψεις της συντακτικής του ομάδας και η Επιτροπή δε μπορεί να θεωρηθεί υπεύθυνη για οποιαδήποτε χρήση των πληροφοριών που περιέχονται σε αυτό.



Co-funded by the
Erasmus+ Programme
of the European Union

Υπολογιστές και Ενσωματωμένα Συστήματα



Μια Εργαλειοθήκη για την προώθηση της χρήσης
Δεξιοτήτων STEM αξιοποιώντας Εφαρμογές Μικροελεγκτών

Project No. 2019-1-RO01-KA202-063965

Το έργο αυτό χρηματοδοτήθηκε με την υποστήριξη της Ευρωπαϊκής Επιτροπής. Το περιεχόμενο του παρόντος εγγράφου αντικατοπτρίζει μόνο τις απόψεις της συντακτικής του ομάδας και η Επιτροπή δε μπορεί να θεωρηθεί υπεύθυνη για οποιαδήποτε χρήση των πληροφοριών που περιέχονται σε αυτό.

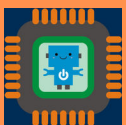


Co-funded by the
Erasmus+ Programme
of the European Union

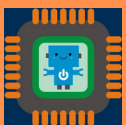
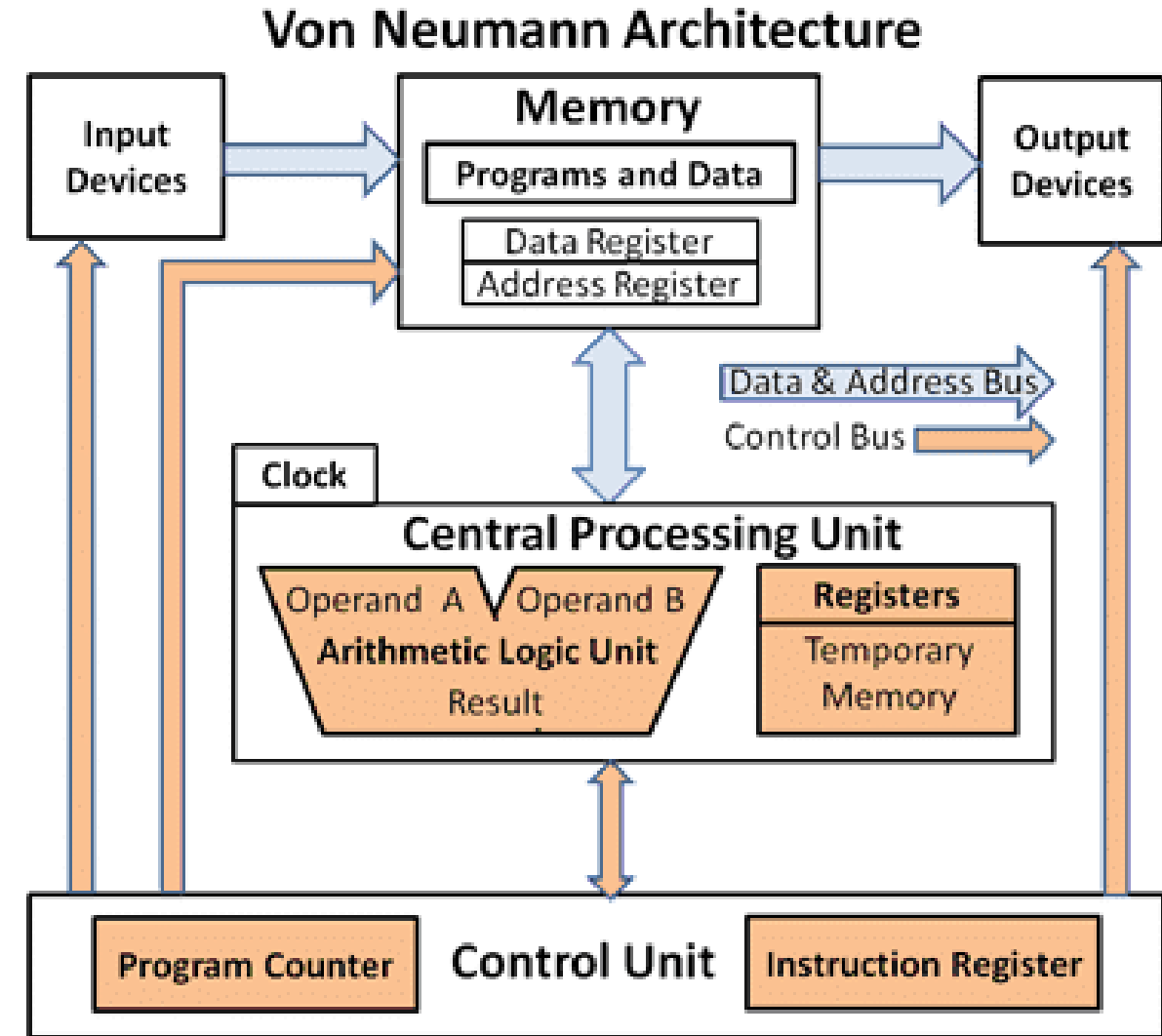
Υπολογιστές και Καθημερινή Ζωή



- Οι υπολογιστές είναι μέρος της καθημερινότητάς μας.
- Computer = Hardware + Software
 - ✓ Υλικό - φυσικά εξαρτήματα
 - ✓ Λογισμικό - προγράμματα, διαδικασίες και ρουτίνες που λέει στον υπολογιστή τι να κάνει.
- Που χρησιμοποιούμε υπολογιστές;



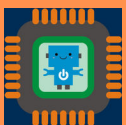
- ✓ CPU = Μικροεπεξεργαστής - «ο εγκέφαλος» του υπολογιστή μας - κάνει όλες τις αριθμητικές και λογικές πράξεις (ALU) και ελέγχει όλες τις δραστηριότητες του συστήματος
- ✓ Μονάδα μνήμης – αποθήκευση δεδομένων και προγραμμάτων
 - RAM – Random Access Memory
 - ROM – Read Only Memory
- ✓ Συσκευές εισόδου/εξόδου



Ενσωματωμένα συστήματα στην καθημερινή ζωή



- Ένα ενσωματωμένο σύστημα είναι ένας υπολογιστής ειδικού σκοπού που χρησιμοποιείται μέσα σε μια συσκευή
- βασίζεται σε μικροελεγκτή που είναι ένα τσιπ βελτιστοποιημένο για τον έλεγχο ηλεκτρονικών συσκευών. Αποθηκεύεται σε ένα ενιαίο ολοκληρωμένο κύκλωμα, αφιερωμένο στην εκτέλεση μιας συγκεκριμένης εργασίας και στην εκτέλεση μιας συγκεκριμένης εφαρμογής
- Που χρησιμοποιούμε ενσωματωμένα συστήματα;



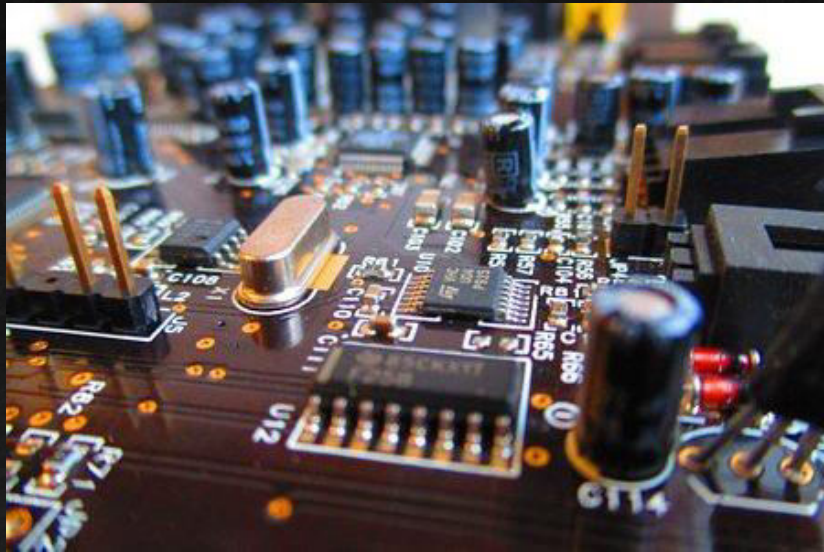
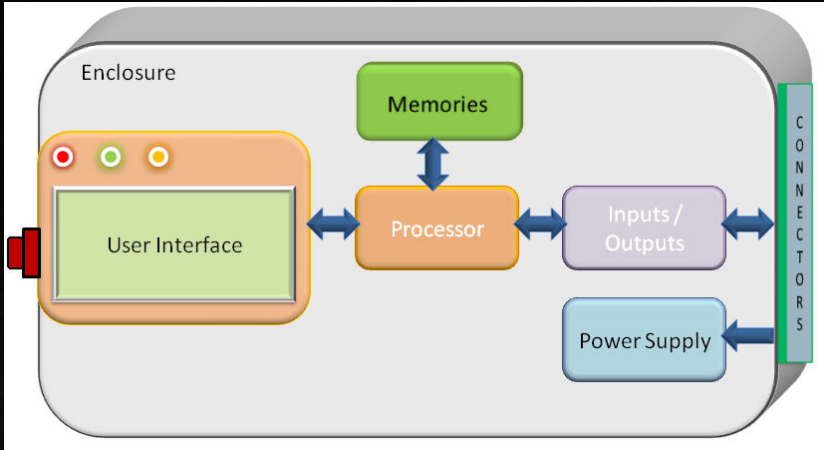
Μια Εργαλειοθήκη για την προώθηση της χρήσης Δεξιοτήτων STEM αξιοποιώντας Εφαρμογές Μικροελεγκτών

Project No. 2019-1-RO01-KA202-063965

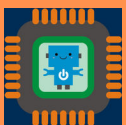
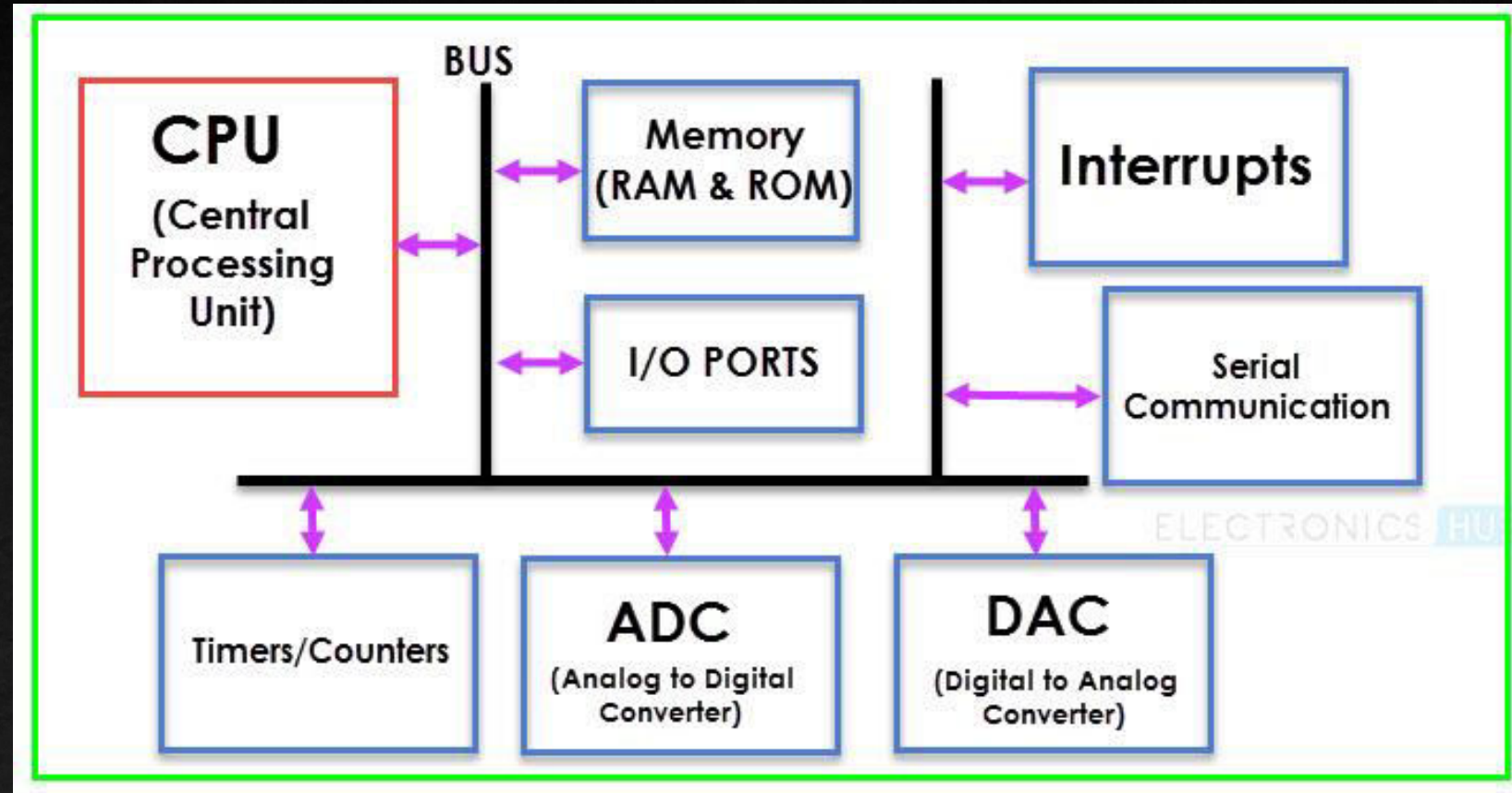
Το έργο αυτό χρηματοδοτήθηκε με την υποστήριξη της Ευρωπαϊκής Επιτροπής. Το περιεχόμενο του παρόντος εγγράφου αντικατοπτρίζει μόνο τις απόψεις της συντακτικής του ομάδας και η Επιτροπή δε μπορεί να θεωρηθεί υπεύθυνη για οποιαδήποτε χρήση των πληροφοριών που περιέχονται σε αυτό.



Co-funded by the
Erasmus+ Programme
of the European Union



- CPU – είναι ένας μικροελεγκτής ή ένας μικροεπεξεργαστής





Ποιό είναι καλύτερο?

Μικροεπεξεργαστής

είναι η καρδιά του συστήματος υπολογιστών.

είναι μόνο ένας επεξεργαστής, επομένως η μνήμη και τα στοιχεία I/O πρέπει να συνδεθούν εξωτερικά

Η μνήμη και το I/O πρέπει να συνδεθούν εξωτερικά, έτσι το κύκλωμα γίνεται μεγάλο

μπορεί να χρησιμοποιηθεί σε συμπαγή συστήματα

το κόστος ολόκληρου του συστήματος είναι υψηλό

λόγω των εξωτερικών εξαρτημάτων, η συνολική κατανάλωση ενέργειας είναι υψηλή. δεν είναι ιδανικό για συσκευές που λειτουργούν με αποθηκευμένη ισχύ όπως οι μπαταρίες.

τα περισσότερα από αυτά δεν διαθέτουν δυνατότητες εξοικονόμησης ενέργειας

χρησιμοποιείται κυρίως σε προσωπικούς υπολογιστές

βασίζονται στο μοντέλο Von Neumann

Μικροελεγκτής

η καρδιά ενός ενσωματωμένου συστήματος

διαθέτει επεξεργαστή μαζί με εσωτερική μνήμη και στοιχεία εισόδου/εξόδου

Η μνήμη και το I/O είναι ήδη παρόντα και το εσωτερικό κύκλωμα είναι μικρό

χρησιμοποιείται σε συμπαγή συστήματα

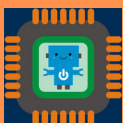
το κόστος ολόκληρου του συστήματος είναι χαμηλό

Καθώς τα εξωτερικά εξαρτήματα είναι χαμηλά, η συνολική κατανάλωση ενέργειας είναι μικρότερη. Έτσι, μπορεί να χρησιμοποιηθεί με συσκευές που λειτουργούν με αποθηκευμένη ισχύ όπως οι μπαταρίες

Τα περισσότερα από αυτά προσφέρουν λειτουργία εξοικονόμησης ενέργειας

χρησιμοποιείται κυρίως σε ενσωματωμένα συστήματα.

βασίζονται στην αρχιτεκτονική του Χάρβαρντ



Μια Εργαλειοθήκη για την προώθηση της χρήσης
Δεξιοτήτων STEM αξιοποιώντας Εφαρμογές Μικροελεγκτών

Project No. 2019-1-RO01-KA202-063965

Το έργο αυτό χρηματοδοτήθηκε με την υποστήριξη της Ευρωπαϊκής Επιτροπής. Το περιεχόμενο του παρόντος εγγράφου αντικατοπτρίζει μόνο τις απόψεις της συντακτικής του ομάδας και η Επιτροπή δε μπορεί να θεωρηθεί υπεύθυνη για οποιαδήποτε χρήση των πληροφοριών που περιέχονται σε αυτό.



Co-funded by the
Erasmus+ Programme
of the European Union



Ποιό είναι καλύτερο?

έχουν μικρότερο αριθμό καταχωρητών, επομένως περισσότερες λειτουργίες βασίζονται στη μνήμη.

είναι μια κεντρική μονάδα επεξεργασίας σε ένα ενιαίο ολοκληρωμένο τσιπ με βάση το πυρίτιο.

δεν έχει RAM, ROM, μονάδες εισόδου-εξόδου, χρονόμετρα και άλλα περιφερειακά στο τσιπ.

χρησιμοποιεί έναν εξωτερικό δίαυλο για διασύνδεση με RAM, ROM και άλλα περιφερειακά.

Τα συστήματα που βασίζονται σε μικροεπεξεργαστές μπορούν να λειτουργούν με πολύ υψηλή ταχύτητα λόγω της τεχνολογίας που εμπλέκεται.

χρησιμοποιείται για εφαρμογές γενικού σκοπού που σας επιτρέπουν να χειρίζεστε πολλά δεδομένα.

Είναι πολύπλοκο και ακριβό, με μεγάλο αριθμό οδηγιών για επεξεργασία.

έχουν περισσότερες εγγραφές, έτσι τα προγράμματα είναι πιο εύκολο να γραφτούν

είναι ένα υποπροϊόν της ανάπτυξης μικροεπεξεργαστών με CPU μαζί με άλλα περιφερειακά

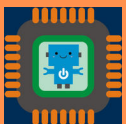
έχει CPU μαζί με RAM, ROM και άλλα περιφερειακά ενσωματωμένα σε ένα μόνο τσιπ.

χρησιμοποιεί έναν εσωτερικό δίαυλο ελέγχου.

Τα συστήματα που βασίζονται σε μικροελεγκτές λειτουργούν έως και 200 MHz ή περισσότερα ανάλογα με την αρχιτεκτονική.

χρησιμοποιείται για συστήματα ειδικά για εφαρμογές.

Είναι απλό και φθηνό με λιγότερο αριθμό εντολών για επεξεργασία



IF statement, declaration & call of void functions and const definition

Task: Program a device capable to read the state of a potentiometer (an analog input) and turns on an LED only if the potentiometer.

It must print the analog value regardless of the level.

```
const int analogPin = A0;
    // pin that the sensor is attached to
const int ledPin = 13;
    // pin that the LED is attached to
const int valmin = 400;
    // an arbitrary valmin level

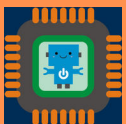
void setup() {
    // initialize the LED pin as an output:
    pinMode(ledPin, OUTPUT);
    // initialize serial communications:
    Serial.begin(9600); }
```

Κατανόηση των Εννοιών Προγραμματισμού με χρήση εφαρμογών Μικροελεγκτών

```
void loop() {
    // read the value of the potentiometer:
    int analogValue = analogRead(analogPin);

    // if the analog value is high enough,
    // turn on the LED:
    if (analogValue > valmin) {
        digitalWrite(ledPin, HIGH);
    } else {
        digitalWrite(ledPin, LOW);
    }

    // print the analog value:
    Serial.println(analogValue);
    delay(1);    // delay in between reads
}
```



Κατανόηση των Εννοιών Προγραμματισμού με χρήση εφαρμογών Μικροελεγκτών

Variable definition

```
<typeVar> nameVar [= value];
```

declares a variable of a specific type, which will determine the size of the values, the length and type of memory representation

Usual types of variables used in Arduino apps

int = numeric type for variables/constants; it is represented on 4 bytes and can store values between approx. $-2 \cdot 10^9 \dots 2 \cdot 10^9$

bool = boolean type for variables/constants; it is represented in 1 byte and can store values false (0) and true (1)

Constant definition

```
const <typeConst> nameConst = value;
```

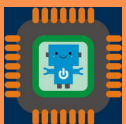
sets a constant of typeConst with specific value

Decision Structure

```
if (Condition) {  
    instructions_A  
    // do instructions_A if the  
    // Condition is true  
} else {  
    instructions_B  
    // do instructions_A if the  
    // Condition is false  
}
```

void Functions - declaration

```
void nameFunction(list of formal parameters)  
{ declaration of local variables  
  instructions  
}
```



void Functions - declaration

```
void nameFunction(formal parameters)
{ declaration of local variables
  instructions
}
```

non-void Functions - declaration

```
resultType nameFunction(formal parameters)
{ declaration of local variables
  instructions
  return expression; //
}
```

where **formal parameters** is a list of types and names of parameters used in function instructions

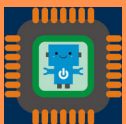
Κατανόηση των Εννοιών Προγραμματισμού με χρήση εφαρμογών Μικροελεγκτών

Call of a function

nameFunction(list of actual parameters)

- void functions - The call is an instruction
- non-void function - The call is an operand in expression with the same type as the resultType

! formal and actual parameters must have same type, number and must be in the same order



Arduino specific functions

setup()

void function - called when a sketch starts, and will only run once, after each powerup or reset of the Arduino board similar with main(). Use it to initialize variables, pin modes, start using libraries, etc.

loop()

void function - loops consecutively, allowing Arduino program to change and respond. It is called after setup() function, which initializes and sets the initial values.

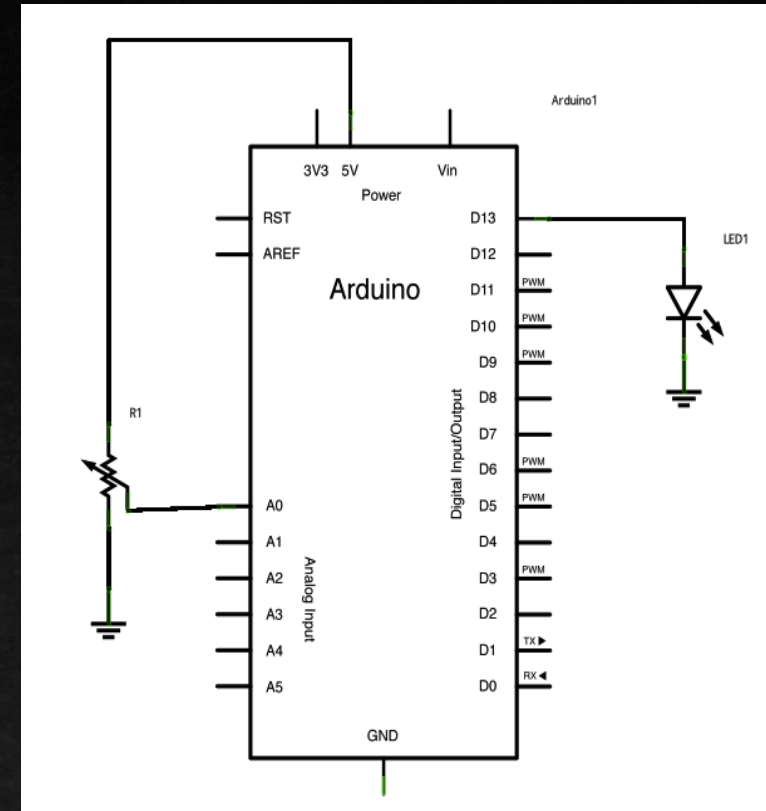
pinMode(pin, mode)

- void function with parameters
- **pin**: the Arduino pin number to set the mode of
 - **mode**: INPUT, OUTPUT, or INPUT_PULLUP

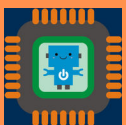
delay(milisc)

- void function with parameters
- **milisc**: number of milliseconds to pause the program

Κατανόηση των Εννοιών Προγραμματισμού με χρήση εφαρμογών Μικροελεγκτών



Ηλεκτρονικό σχήμα της συσκευής Arduino



Μια Εργαλειοθήκη για την προώθηση της χρήσης
Δεξιοτήτων STEM αξιοποιώντας Εφαρμογές Μικροελεγκτών

Project No. 2019-1-RO01-KA202-063965

Το έργο αυτό χρηματοδοτήθηκε με την υποστήριξη της Ευρωπαϊκής Επιτροπής. Το περιεχόμενο του παρόντος εγγράφου αντικατοπτρίζει μόνο τις απόψεις της συντακτικής του ομάδας και η Επιτροπή δε μπορεί να θεωρηθεί υπεύθυνη για οποιαδήποτε χρήση των πληροφοριών που περιέχονται σε αυτό.



Co-funded by the
Erasmus+ Programme
of the European Union

Arduino specific functions

digitalWrite(pin, value)

void function with parametrers

- pin: the Arduino pin number.
- value: HIGH or LOW

digitalRead(pin)

function with parametrers

- pin: the Arduino pin num
- return value: HIGH or LOW

analogRead(pin)

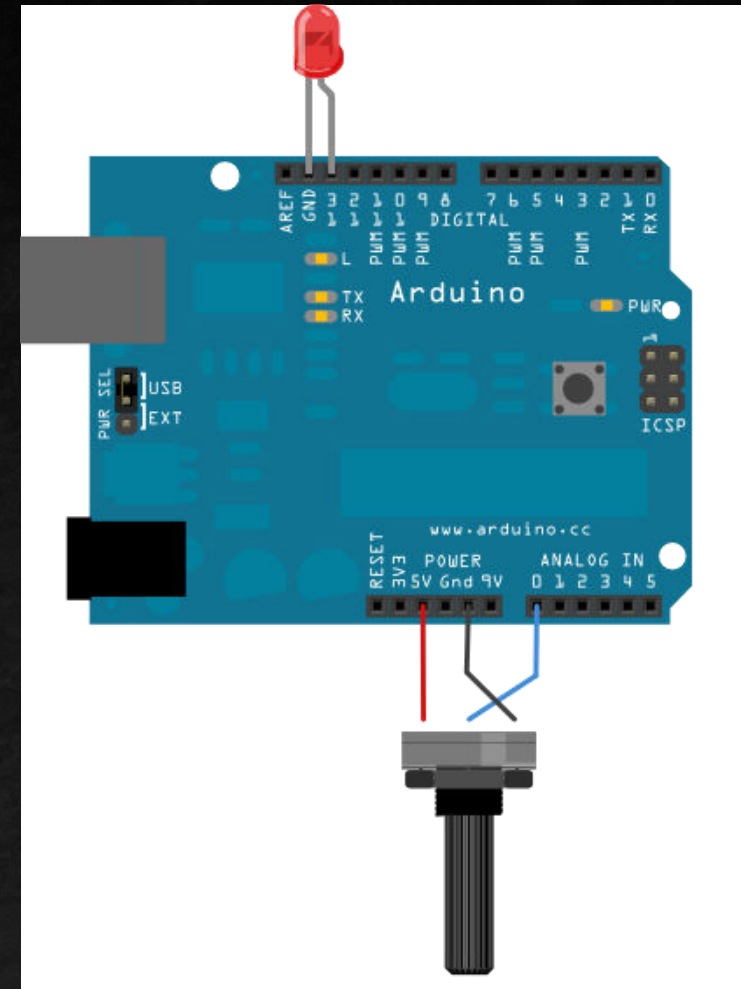
int function with parametrer

- pin: the name of the analog input pin to read from (A0 to A5 on most boards)
- Returns the analog reading on the pin.

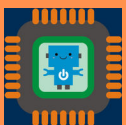
Arduino Pin Levels Constants HIGH και LOW

pin	Είσοδος	ΕΞΟΔΟΣ
HIGH	τάση > 3.0V	5V
LOW	τάση > 3.0V	0V

Κατανόηση των Εννοιών Προγραμματισμού με χρήση εφαρμογών Μικροελεγκτών



Συσκευή Arduino



Μια Εργαλειοθήκη για την προώθηση της χρήσης
Δεξιοτήτων STEM αξιοποιώντας Εφαρμογές Μικροελεγκτών

Project No. 2019-1-RO01-KA202-063965

Το έργο αυτό χρηματοδοτήθηκε με την υποστήριξη της Ευρωπαϊκής Επιτροπής. Το περιεχόμενο του παρόντος εγγράφου αντικατοπτρίζει μόνο τις απόψεις της συντακτικής του ομάδας και η Επιτροπή δε μπορεί να θεωρηθεί υπεύθυνη για οποιαδήποτε χρήση των πληροφοριών που περιέχονται σε αυτό.



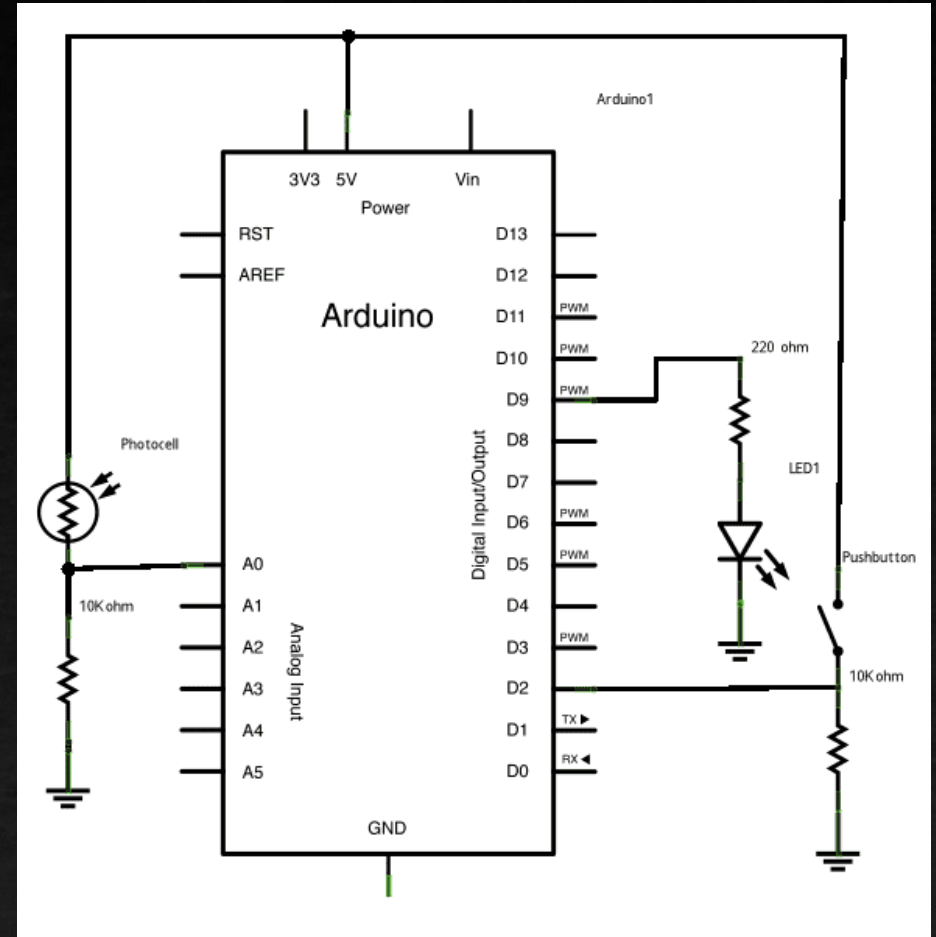
Co-funded by the
Erasmus+ Programme
of the European Union

WHILE statement

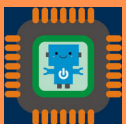
Task: Program a device capable to read for five seconds the sensor input and calibrate by defining the minimum and maximum of expected values for the readings taken during the loop.

```
const int sensorPin = A0;  
    // pin that the sensor is attached to  
const int ledPin = 9;  
    // pin that the LED is attached to  
    // variables:  
int sensorValue = 0;  
    // the sensor value  
int sensorMin = 1023;  
    // minimum sensor value  
int sensorMax = 0;  
    // maximum sensor value
```

Κατανόηση των Εννοιών Προγραμματισμού με χρήση εφαρμογών Arduino



Σχήμα της συσκευής Arduino



Μια Εργαλειοθήκη για την προώθηση της χρήσης
Δεξιοτήτων STEM αξιοποιώντας Εφαρμογές Μικροελεγκτών

Project No. 2019-1-RO01-KA202-063965

Το έργο αυτό χρηματοδοτήθηκε με την υποστήριξη της Ευρωπαϊκής Επιτροπής. Το περιεχόμενο του παρόντος εγγράφου αντικατοπτρίζει μόνο τις απόψεις της συντακτικής του ομάδας και η Επιτροπή δε μπορεί να θεωρηθεί υπεύθυνη για οποιαδήποτε χρήση των πληροφοριών που περιέχονται σε αυτό.

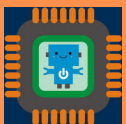


Co-funded by the
Erasmus+ Programme
of the European Union

WHILE statement

```
void setup() {  
  // turn on LED to signal the start  
  // of the calibration period:  
  pinMode(13, OUTPUT);  
  digitalWrite(13, HIGH);  
  // calibrate during the first five seconds  
  while (millis() < 5000) {  
    sensorValue = analogRead(sensorPin);  
    if (sensorValue > sensorMax){  
      sensorMax = sensorValue;  
    }  
    if (sensorValue < sensorMin){  
      sensorMin = sensorValue;  
    }  
  }  
  digitalWrite(13, LOW);    // signal the end of the calibration period  
}
```

Κατανόηση των Εννοιών Προγραμματισμού με χρήση εφαρμογών Μικροελεγκτών



Κατανόηση των Εννοιών Προγραμματισμού με χρήση εφαρμογών Μικροελεγκτών

Determin the Minimum and Maximum value from a set of values

Algorithm

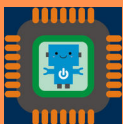
- Step 1.** Set the variables for sensorMin with the maximum value possible and sensor Max with the minimum value possible
- Step 2.** Compare current value with sensorMin and, if it is smaller, update sensorMin
- Step 3.** Compare current value with sensorMax and, if it is greater, update sensorMax

WHILE statement

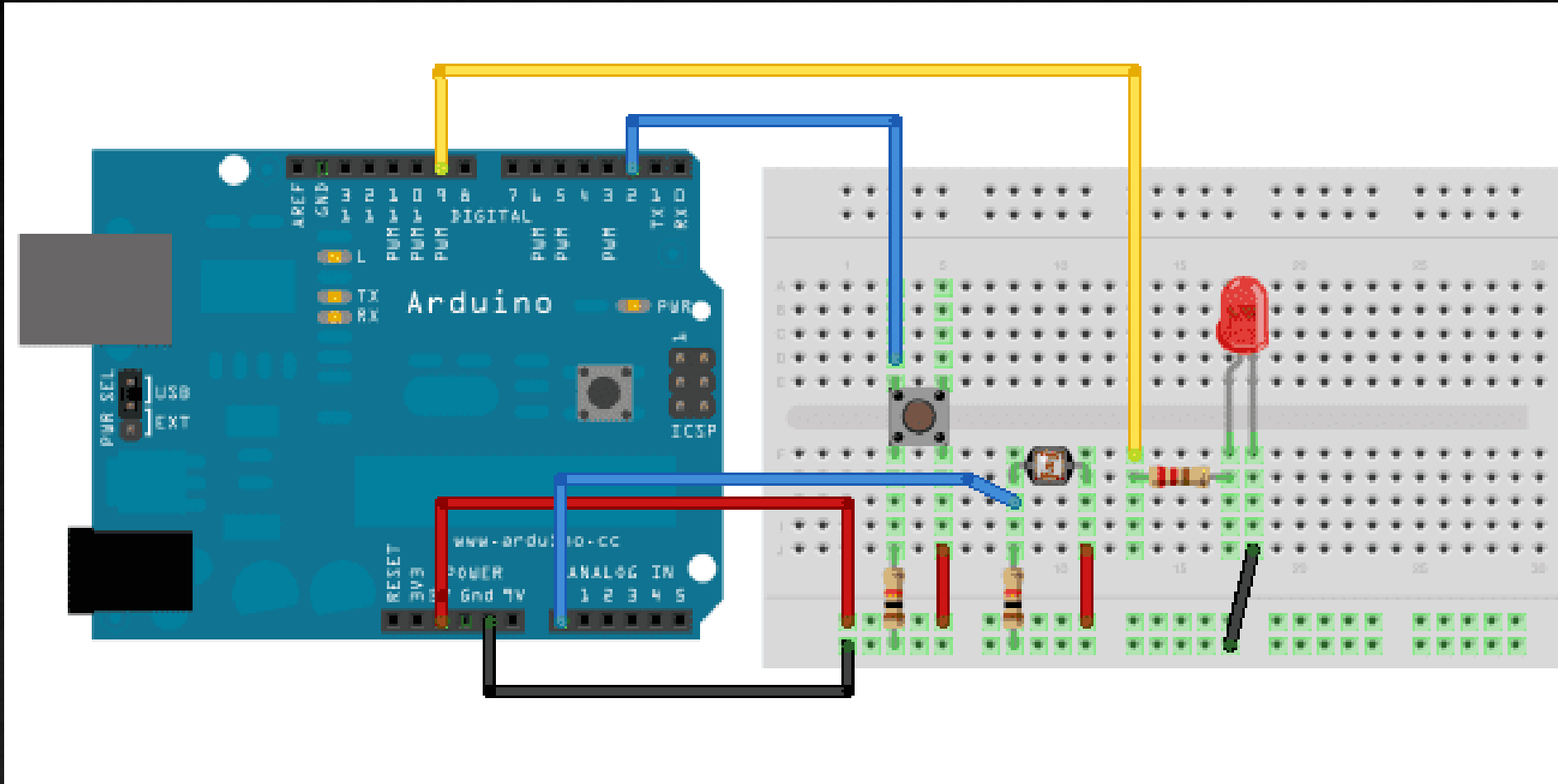
```
while (Condition) {  
    instructions_A  
}
```

Execution

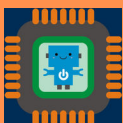
- Step 1.** The Condition is evaluated
- Step 2.** If the Condition is True
 - 2.1. Instructions A will be executed
 - 2.2. Go to Step 1.
- If Condition is False, the program execution leave the loop



Κατανόηση των Εννοιών Προγραμματισμού με χρήση εφαρμογών Μικροελεγκτών



Συσκευή Arduino
που εξηγεί το WHILE



Μια Εργαλειοθήκη για την προώθηση της χρήσης
Δεξιοτήτων STEM αξιοποιώντας Εφαρμογές Μικροελεγκτών

Project No. 2019-1-RO01-KA202-063965

Το έργο αυτό χρηματοδοτήθηκε με την υποστήριξη της Ευρωπαϊκής Επιτροπής. Το περιεχόμενο του παρόντος εγγράφου αντικατοπτρίζει μόνο τις απόψεις της συντακτικής του ομάδας και η Επιτροπή δε μπορεί να θεωρηθεί υπεύθυνη για οποιαδήποτε χρήση των πληροφοριών που περιέχονται σε αυτό.



Co-funded by the
Erasmus+ Programme
of the European Union

FOR statement and ARRAYS manipulation

Κατανόηση των Εννοιών Προγραμματισμού με χρήση εφαρμογών Μικροελεγκτών

```
int timer = 100;
    // The higher the number,
    // the slower the timing.
int ledPins[] = { 2, 7, 4, 6, 5, 3};
    // an array of pin numbers to which LEDs are attached
int pinCount = 6;
    // the number of pins (the length of the array)

void setup() {
    // the array elements are numbered from 0 to (pinCount-1)
    // use a for loop to initialize each pin as an output:
    for (int thisPin = 0; thisPin < pinCount; thisPin++) {
        pinMode(ledPins[thisPin], OUTPUT);
    }
}
```

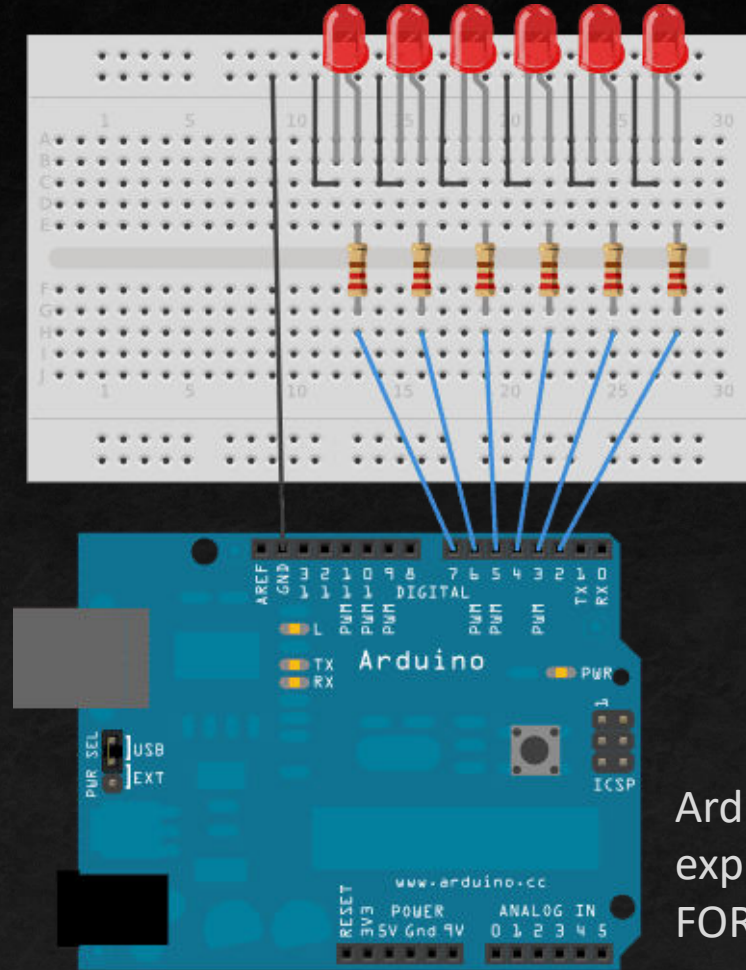
Task: Program a device capable to light up a series of LEDs attached to pins whose numbers are neither contiguous nor necessarily sequential. To do this is, pin numbers will be store in an ARRAY and then use FOR loops to iterate over the array.



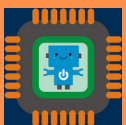
FOR statement and ARRAYS manipulation

Κατανόηση των Εννοιών Προγραμματισμού με χρήση εφαρμογών Μικροελεγκτών

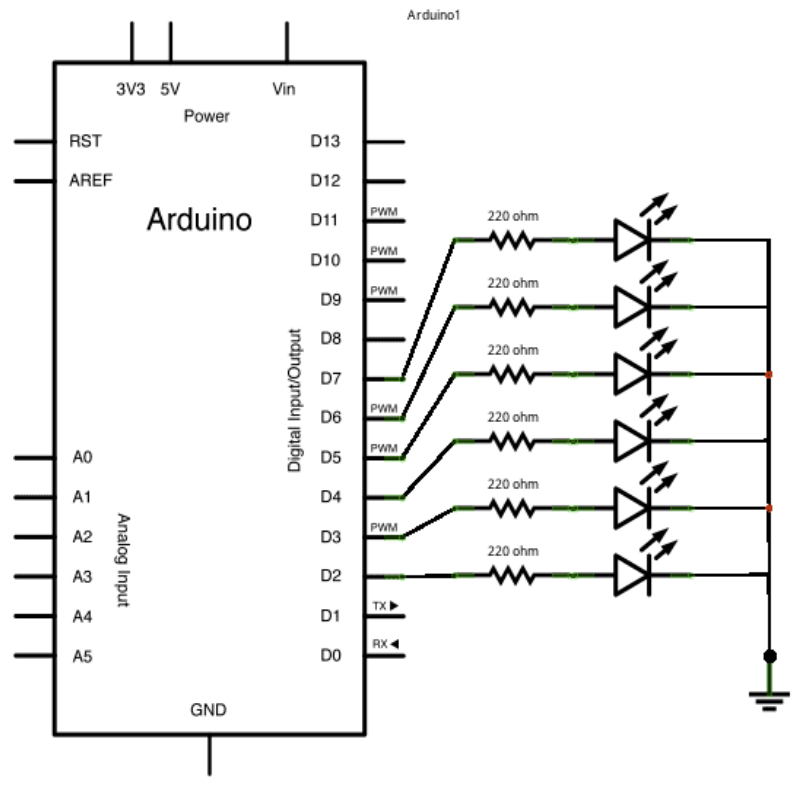
```
void loop() {  
  // loop from the lowest pin to the highest:  
  for (int thisPin = 0; thisPin < pinCount; thisPin++) {  
    digitalWrite(ledPins[thisPin], HIGH);  
    // turn the pin on:  
    delay(timer);  
    digitalWrite(ledPins[thisPin], LOW);  
    // turn the pin off:  
  }  
  
  // loop from the highest pin to the lowest:  
  for (int thisPin=pinCount-1; thisPin >= 0; thisPin--){  
    // turn the pin on:  
    digitalWrite(ledPins[thisPin], HIGH);  
    delay(timer);  
    // turn the pin off:  
    digitalWrite(ledPins[thisPin], LOW);  
  }  
}
```



Arduino device explaining FOR loop



Κατανόηση των Εννοιών Προγραμματισμού με χρήση εφαρμογών Μικροελεγκτών



Σχήμα της συσκευής Arduino

FOR statement

```
for(int counter = initialVal; counter <= finalVal; counter++) {  
    instructions_A  
}
```

Execution

Step 1. The counter is set with initialValue

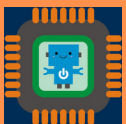
Step 2. If counter <= finaValue is True

2.1. Instructions A will be executed

2.2. counter is increased with 1

2.3. Go to Step 2

Step 3. If counter <= finaValue is False, the program execution leave the loop



Organizing data in ARRAYS

Array = a collection of data with the same type, organized in a contiguous memory zone and referred with a single name, which is a pointer (memory address) of the first element in the array.

Declaration:

```
elementType arrayName[numberOfElements];
```

Initialization:

- along with the declaration

```
int ledPins[] = { 2, 7, 4, 6, 5, 3};
```
- by assignation

```
digitalWrite(ledPins[thisPin], HIGH);
```
- by reading values from input

Κατανόηση των Εννοιών Προγραμματισμού με χρήση εφαρμογών Μικροελεγκτών

Referring to a specific value from the array

`A[expressionIndex]`

`expressionIndex` is an integer from `[0, count-1]`, indicating the position of the element in array

Parsing the ARRAY in order to analyze and process its elements:

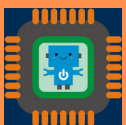
left-right

```
for (int it = 0; it < count; it++)  
    process(A[it]);
```

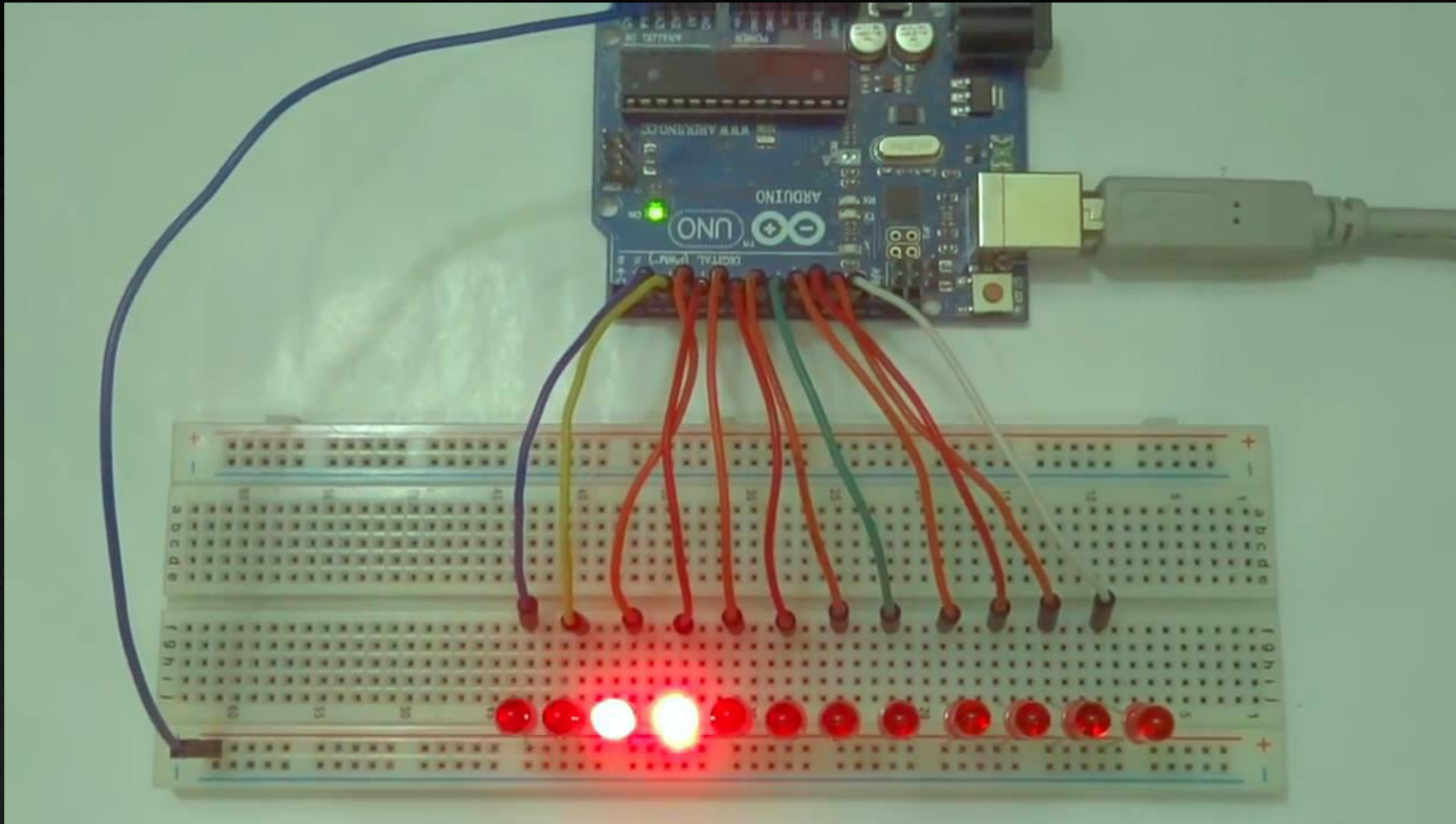
right-left

```
for (int it = count - 1; it >= 0; it--)  
    process(A[it]);
```

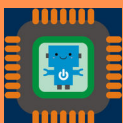
where `count` is the number of array elements and `it` is an index used for parsing the array



Κατανόηση των Εννοιών Προγραμματισμού με χρήση εφαρμογών Μικροελεγκτών



Μαθαίνοντας για τους
βρόχους ΓΙΑ και τους Πίνακες
μέσω συσκευών Arduino



Μια Εργαλειοθήκη για την προώθηση της χρήσης
Δεξιοτήτων STEM αξιοποιώντας Εφαρμογές Μικροελεγκτών

Project No. 2019-1-RO01-KA202-063965

Το έργο αυτό χρηματοδοτήθηκε με την υποστήριξη της Ευρωπαϊκής Επιτροπής. Το περιεχόμενο του παρόντος εγγράφου αντικατοπτρίζει μόνο τις απόψεις της συντακτικής του ομάδας και η Επιτροπή δε μπορεί να θεωρηθεί υπεύθυνη για οποιαδήποτε χρήση των πληροφοριών που περιέχονται σε αυτό.



Co-funded by the
Erasmus+ Programme
of the European Union

Καλυπτόμενες Επιστημονικές Περιοχές

Αρχιτεκτονική υλικού για υπολογιστές και ενσωματωμένα συστήματα

Δομημένος Προγραμματισμός – τύποι δεδομένων, καταστάσεις (IF, WHILE, FOR), συναρτήσεις που καθορίζονται από τον χρήστη

Προγραμματισμός συσκευών που βασίζονται σε μικροελεγκτή (π.χ. Arduino)



Μια Εργαλειοθήκη για την προώθηση της χρήσης
Δεξιοτήτων STEM αξιοποιώντας Εφαρμογές Μικροελεγκτών

Project No. 2019-1-RO01-KA202-063965

Το έργο αυτό χρηματοδοτήθηκε με την υποστήριξη της Ευρωπαϊκής Επιτροπής. Το περιεχόμενο του παρόντος εγγράφου αντικατοπτρίζει μόνο τις απόψεις της συντακτικής του ομάδας και η Επιτροπή δε μπορεί να θεωρηθεί υπεύθυνη για οποιαδήποτε χρήση των πληροφοριών που περιέχονται σε αυτό.



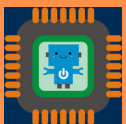
Co-funded by the
Erasmus+ Programme
of the European Union

- Test πολλαπλής επιλογής
- mini-project σε ομάδα 2-3 μαθητών - προγραμματισμός συσκευών Arduino που :
 - ✓ Θα περιγράφουν τη λειτουργία άλλων συγκεκριμένων εντολών στον δομημένο προγραμματισμό
 - ✓ Θα εφαρμόζονται σε πραγματικές καταστάσεις - για παράδειγμα, τη λειτουργία ενός φαναριού RGV.



Βιβλιογραφία

- <https://creativecommons.org/2008/10/22/wired-on-arduino-and-open-source-computing/>
- <https://www.arduino.cc/en/Tutorial/BuiltInExamples/WhileStatementConditional>
- <https://www.arduino.cc/en/Tutorial/BuiltInExamples/ForLoopIteration>
- <https://commons.wikimedia.org/wiki/Category:Fritzing>
- <https://www.arduino.cc/en/Tutorial/BuiltInExamples/ForLoopIteration>
- https://commons.wikimedia.org/wiki/Category:Arduino_projects



Μια Εργαλειοθήκη για την προώθηση της χρήσης
Δεξιοτήτων STEM αξιοποιώντας Εφαρμογές Μικροελεγκτών

Project No. 2019-1-RO01-KA202-063965

Το έργο αυτό χρηματοδοτήθηκε με την υποστήριξη της Ευρωπαϊκής Επιτροπής. Το περιεχόμενο του παρόντος εγγράφου αντικατοπτρίζει μόνο τις απόψεις της συντακτικής του ομάδας και η Επιτροπή δε μπορεί να θεωρηθεί υπεύθυνη για οποιαδήποτε χρήση των πληροφοριών που περιέχονται σε αυτό.



Co-funded by the
Erasmus+ Programme
of the European Union