



**ΠΑΝΕΠΙΣΤΗΜΙΟ
ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ**

Ενσωματωμένα Συστήματα

Ενότητα: ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ: CIRC-02

Δρ. Μηνάς Δασυγένης

mdasyg@ieee.org

Τμήμα Μηχανικών Πληροφορικής και Τηλεπικοινωνιών

Εργαστήριο Ψηφιακών Συστημάτων και Αρχιτεκτονικής Υπολογιστών

<http://arch.ict.e.uowm.gr/mdasyg>

Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ψηφιακά Μαθήματα του Πανεπιστημίου Δυτικής Μακεδονίας**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Περιεχόμενα

1.Σκοπός της άσκησης.....	4
2.Παραδοτέα.....	4
3.Κατασκευή του κυκλώματος CIRC-02.....	5
4.Επαλήθευση παραμέτρων στο προγραμματιστικό περιβάλλον του Arduino.....	5
5.Προγραμματισμός του κυκλώματος CIRC-02.....	6
6.Παραμετροποίηση του κυκλώματος CIRC-02.....	6
6.1Αλλαγή συνάρτησης ελέγχου των LEDs.....	6
6.2Δημιουργία μιας δικιάς μας συνάρτησης για τον έλεγχο των LEDs.....	7

1. Σκοπός της άσκησης

- Χρήση του αναπτυξιακού περιβάλλοντος του Arduino¹ για τη συγγραφή και τη μεταφόρτωση προγραμμάτων στην πλακέτα του Arduino Uno.
- Δημιουργία κυκλώματος ελέγχου ενός LED.

2. Παραδοτέα

- Ένα αρχείο zip με τα project **CIRC_02**, **CIRC_02_d1**, **CIRC_02_d2** που θα δημιουργήσετε.
- Ένα screenshot από το Arduino IDE που δείχνει ότι η μεταγλώττιση (compile) έγινε με επιτυχία και ταυτόχρονα εμφανίζει το μέγεθος του δυαδικού σχεδίου για κάθε κύκλωμα.
- Ένα video 5-10 sec επίδειξης του κυκλώματος μαζί με ηχητική περιγραφή για κάθε κύκλωμα.




Χρόνος ολοκλήρωσης εργαστηρίου: 25 λεπτά.

Στην προηγούμενη εργαστηριακή άσκηση υλοποιήθηκε ένα κύκλωμα που ενεργοποιούσε και απενεργοποιούσε ένα LED. Σ' αυτή την εργαστηριακή άσκηση θα υλοποιηθεί ένα κύκλωμα που εφαρμόζει την ίδια λειτουργία σε οκτώ LEDs. Θα υλοποιηθούν διάφορα LED animations και θα κατανοηθούν καλύτερα οι βασικές αρχές λειτουργίας και προγραμματισμού του Arduino, αφού θα χρησιμοποιηθούν πίνακες για ευκολότερη διαχείριση των μεταβλητών και η for() για επαναλήψεις.

¹ Χρησιμοποιήθηκε υλικό από το [SparkFun Inventors Kit for Arduino](#)

3. Κατασκευή του κυκλώματος CIRC-02

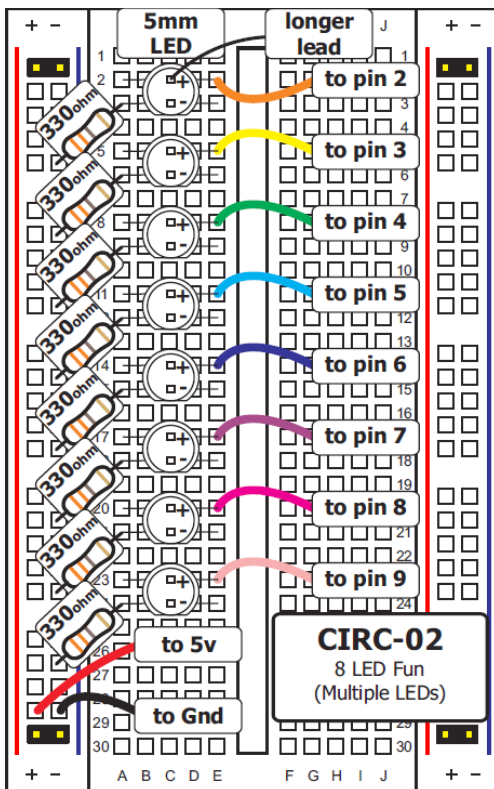
Για την εκπόνηση του κυκλώματος CIRC-02 απαιτούνται τα εξής μέρη:

Στοιχείο	Περιγραφή και ποσότητα
	8x 5mm κίτρινο LED
	8x Αντίσταση 330 Ohm (Πορτοκαλί – Πορτοκαλί – Καφέ)
	10x Καλώδια

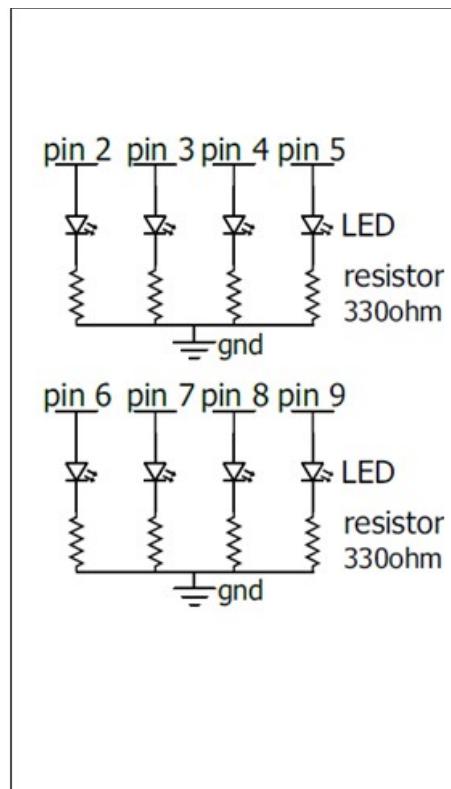
Τα φυλλάδια δεδομένων (datasheets) των υλικών βρίσκονται παρακάτω:

- [LED](#)
- [Αντίσταση 330 Ohm](#)

Κατασκευάστε το κύκλωμα, σύμφωνα με την παρακάτω σχηματική αναπαράσταση:



Σχήμα 1



Σχήμα 2

Συνδέστε το θετικό ακροδέκτη του LED στο αναγραφόμενο pin του Arduino και μια αντίσταση 330 Ohm στον κάθε αρνητικό ακροδέκτη, που να πηγαίνει στη γείωση.

Επισημαίνεται ότι ο θετικός ακροδέκτης του LED είναι αυτός με το μεγαλύτερο μήκος.

Τέλος συνδέστε, στις κατάλληλες θέσεις του breadboard, ένα καλώδιο στο pin τροφοδοσίας 5V του Arduino και ένα σε κάποιο pin γείωσης (Gnd).

4. Επαλήθευση παραμέτρων στο προγραμματιστικό περιβάλλον του Arduino

Συνδέστε το Arduino Uno και ανοίξτε το προγραμματιστικό περιβάλλον του.

Επιβεβαιώστε ότι η επιλεγμένη σειριακή θύρα είναι η θύρα στην οποία είναι συνδεδεμένο το Arduino.

Βεβαιωθείτε ότι η επιλεγμένη πλακέτα είναι το Arduino Uno

Βεβαιωθείτε ότι ο επιλεγμένος προγραμματιστής είναι ο AVRISP mkII

Όλα τα παραπάνω έχουν περιγραφεί αναλυτικά στην Ενότητα 2 του CIRC-01.

5. Προγραμματισμός του κυκλώματος CIRC-02

Αντιγράψτε τον παρακάτω κώδικα στο προγραμματιστικό περιβάλλον του Arduino.

(Εναλλακτικά κατεβάστε τον από [εδώ](#))

```

                                Κώδικας προγραμματισμού του κυκλώματος
/* -----
 * | Arduino Experimentation Kit Example Code|
 * | CIRC-02 .: 8 LED Fun .: (Multiple LEDs)|
 * -----
 *
 * A few Simple LED animations
 *
 * For more information on this circuit http://tinyurl.com/d2hrud
 *
 */
//LED Pin Variables
int ledPins[] = {2,3,4,5,6,7,8,9}; //An array to hold the pin each LED is connected
to
                                //i.e. LED #0 is connected to pin 2, LED #1, 3 and so on
                                //to address an array use ledPins[0] this would equal 2
                                //and ledPins[7] would equal 9

/*
 * setup() - this function runs once when you turn your Arduino on
 * We the three control pins to outputs
 */
void setup()
{
  //Set each pin connected to an LED to output mode (pulling high (on) or low (off)
  for(int i = 0; i < 8; i++){ //this is a loop and will repeat eight times
    pinMode(ledPins[i],OUTPUT); //we use this to set each LED pin to output
  } //the code this replaces is below
/* (commented code will not run)
 * these are the lines replaced by the for loop above they do exactly the
 * same thing the one above just uses less typing
  pinMode(ledPins[0],OUTPUT);
  pinMode(ledPins[1],OUTPUT);
  pinMode(ledPins[2],OUTPUT);

```

Κώδικας προγραμματισμού του κυκλώματος

```
pinMode(ledPins[3],OUTPUT);
pinMode(ledPins[4],OUTPUT);
pinMode(ledPins[5],OUTPUT);
pinMode(ledPins[6],OUTPUT);
pinMode(ledPins[7],OUTPUT);
(end of commented code)*/
}
/*
* loop() - this function will start after setup finishes and then repeat
* we call a function called oneAfterAnother(). if you would like a different
behaviour
* uncomment (delete the two slashes) one of the other lines
*/
void loop() // run over and over again
{
  oneAfterAnotherNoLoop(); //this will turn on each LED one by one then turn each
off
  //oneAfterAnotherLoop(); //does the same as oneAfterAnotherNoLoop but with
//much less typing
//oneOnAtATime();
//this will turn one LED on then turn the next one
//on turning the
//former off (one LED will look like it is scrolling
//along the lin//inAndOut();
//lights the two middle LEDs then moves them out then back
//in again
}
/*
* oneAfterAnotherNoLoop() - Will light one LED then delay for delayTime then light
* the next LED until all LEDs are on it will then turn them off one after another
*
* this does it without using a loop which makes for a lot of typing.
* oneOnAtATimeLoop() does exactly the same thing with less typing
*/
void oneAfterAnotherNoLoop(){
  int delayTime = 100; //the time (in milliseconds) to pause between LEDs
//make smaller for quicker switching and larger for slower
  digitalWrite(ledPins[0], HIGH); //Turns on LED #0 (connected to pin 2 )
  delay(delayTime); //waits delayTime milliseconds
  digitalWrite(ledPins[1], HIGH); //Turns on LED #1 (connected to pin 3 )
  delay(delayTime); //waits delayTime milliseconds
  digitalWrite(ledPins[2], HIGH); //Turns on LED #2 (connected to pin 4 )
  delay(delayTime); //waits delayTime milliseconds
  digitalWrite(ledPins[3], HIGH); //Turns on LED #3 (connected to pin 5 )
  delay(delayTime); //waits delayTime milliseconds
  digitalWrite(ledPins[4], HIGH); //Turns on LED #4 (connected to pin 6 )
  delay(delayTime); //waits delayTime milliseconds
  digitalWrite(ledPins[5], HIGH); //Turns on LED #5 (connected to pin 7 )
  delay(delayTime); //waits delayTime milliseconds
  digitalWrite(ledPins[6], HIGH); //Turns on LED #6 (connected to pin 8 )
  delay(delayTime); //waits delayTime milliseconds
  digitalWrite(ledPins[7], HIGH); //Turns on LED #7 (connected to pin 9 )
  delay(delayTime); //waits delayTime milliseconds
//Turns Each LED Off
  digitalWrite(ledPins[7], LOW); //Turns on LED #0 (connected to pin 2 )
  delay(delayTime); //waits delayTime milliseconds
  digitalWrite(ledPins[6], LOW); //Turns on LED #1 (connected to pin 3 )
  delay(delayTime); //waits delayTime milliseconds
  digitalWrite(ledPins[5], LOW); //Turns on LED #2 (connected to pin 4 )
  delay(delayTime); //waits delayTime milliseconds
  digitalWrite(ledPins[4], LOW); //Turns on LED #3 (connected to pin 5 )
  delay(delayTime); //waits delayTime milliseconds
  digitalWrite(ledPins[3], LOW); //Turns on LED #4 (connected to pin 6 )
  delay(delayTime); //waits delayTime milliseconds
  digitalWrite(ledPins[2], LOW); //Turns on LED #5 (connected to pin 7 )
  delay(delayTime); //waits delayTime milliseconds
```

Κώδικας προγραμματισμού του κυκλώματος

```
digitalWrite(ledPins[1], LOW); //Turns on LED #6 (connected to pin 8 )
delay(delayTime); //waits delayTime milliseconds
digitalWrite(ledPins[0], LOW); //Turns on LED #7 (connected to pin 9 )
delay(delayTime); //waits delayTime milliseconds
}
/*
 * oneAfterAnotherLoop() - Will light one LED then delay for delayTime then light
 * the next LED until all LEDs are on it will then turn them off one after another
 *
 * this does it using a loop which makes for a lot less typing.
 * than oneOnAtATimeNoLoop() does exactly the same thing with less typing
 */
void oneAfterAnotherLoop(){
    int delayTime = 100; //the time (in milliseconds) to pause between LEDs
    //make smaller for quicker switching and larger for slower
    //Turn Each LED on one after another
    for(int i = 0; i <= 7; i++){
        digitalWrite(ledPins[i], HIGH); //Turns on LED #i each time this runs i
        delay(delayTime); //gets one added to it so this will repeat
    } //8 times the first time i will = 0 the final
    //time i will equal 7;

//Turn Each LED off one after another
    for(int i = 7; i >= 0; i--){
        //same as above but rather than starting at 0 and counting
        up
        //we start at seven and count down
        digitalWrite(ledPins[i], LOW); //Turns off LED #i each time this runs i
        delay(delayTime); //gets one subtracted from it so this will repeat
    } //8 times the first time i will = 7 the final
    //time it will equal 0
}

/*
 * oneOnAtATime() - Will light one LED then the next turning off all the others
 */
void oneOnAtATime(){
    int delayTime = 100; //the time (in milliseconds) to pause between LEDs
    //make smaller for quicker switching and larger for slower
    for(int i = 0; i <= 7; i++){
        int offLED = i - 1; //Calculate which LED was turned on last time through
        if(i == 0) { //for i = 1 to 7 this is i minus 1 (i.e. if i = 2 we will
            offLED = 7; //turn on LED 2 and off LED 1)
        } //however if i = 0 we don't want to turn of led -1 (doesn't exist)
        //instead we turn off LED 7, (looping around)
        digitalWrite(ledPins[i], HIGH); //turn on LED #i
        digitalWrite(ledPins[offLED], LOW); //turn off the LED we turned on last time
        delay(delayTime);
    }
}

/*
 * inAndOut() - This will turn on the two middle LEDs then the next two out
 * making an in and out look
 */
void inAndOut(){
    int delayTime = 100; //the time (in milliseconds) to pause between LEDs
    //make smaller for quicker switching and larger for slower

//runs the LEDs out from the middle
    for(int i = 0; i <= 3; i++){
        int offLED = i - 1; //Calculate which LED was turned on last time through
        if(i == 0) { //for i = 1 to 7 this is i minus 1 (i.e. if i = 2 we will
            offLED = 3; //turn on LED 2 and off LED 1)
        } //however if i = 0 we don't want to turn of led -1 (doesn't exist)
```



Κώδικας προγραμματισμού του κυκλώματος


```
//instead we turn off LED 7, (looping around)
int onLED1 = 3 - i; //this is the first LED to go on ie. LED #3 when i = 0 and
LED
    //#0 when i = 3
int onLED2 = 4 + i; //this is the first LED to go on ie. LED #4 when i = 0 and
LED
    //#7 when i = 3
int offLED1 = 3 - offLED; //turns off the LED we turned on last time
int offLED2 = 4 + offLED; //turns off the LED we turned on last time

digitalWrite(ledPins[onLED1], HIGH);
digitalWrite(ledPins[onLED2], HIGH);
digitalWrite(ledPins[offLED1], LOW);
digitalWrite(ledPins[offLED2], LOW);
delay(delayTime);
}

//runs the LEDs into the middle
for(int i = 3; i >= 0; i--){
    int offLED = i + 1; //Calculate which LED was turned on last time through
    if(i == 3) { //for i = 1 to 7 this is i minus 1 (i.e. if i = 2 we will
        offLED = 0; //turn on LED 2 and off LED 1)
    } //however if i = 0 we don't want to turn of led -1 (doesn't exist)
//instead we turn off LED 7, (looping around)
int onLED1 = 3 - I; //this is the first LED to go on ie. LED #3 when i = 0 and
LED //#0 when i = 3
int onLED2 = 4 + i; //this is the first LED to go on ie. LED #4 when i = 0 and LED
    //#7 when i = 3
int offLED1 = 3 - offLED; //turns off the LED we turned on last time
int offLED2 = 4 + offLED; //turns off the LED we turned on last time

digitalWrite(ledPins[onLED1], HIGH);
digitalWrite(ledPins[onLED2], HIGH);
digitalWrite(ledPins[offLED1], LOW);
digitalWrite(ledPins[offLED2], LOW);
delay(delayTime);
}
}
```

Πατήστε  **Αποθήκευση (Ctrl+s)**. Θα σας ζητηθεί να δώσετε όνομα φακέλου (*Sketch Folder*). Δώστε **"CIRC-02"** και πατήστε **"Αποθήκευση"**. Το πρόγραμμα θα δημιουργήσει τον φάκελο **CIRC-02** μέσα στον οποίο θα περιέχεται το αρχείο **CIRC-02.ino** με τον κώδικα που γράψατε παραπάνω.

Στη συνέχεια φορτώστε το πρόγραμμα στην πλακέτα πατώντας  **Φόρτωση (Ctrl+u)**.

Μόλις ολοκληρωθεί η διαδικασία φόρτωσης το πρόγραμμα θα εκτελεστεί στην πλακέτα, κάνοντας το LED του κυκλώματος και της πλακέτας να αναβοσβήνει.

Σε περίπτωση που κάποια LEDs δεν ανάβουν, ελέγξτε την πολικότητα τους. Αν κάποια LEDs ανάβουν εκτός της ακολουθίας που περιγράφεται στη συνάρτηση `oneAfterAnotherNoLoop()` ελέγξτε μήπως τα καλώδια τους έχουν τοποθετηθεί στη λάθος θύρα.

6. Παραμετροποίηση του κυκλώματος CIRC-02

6.1 Αλλαγή συνάρτησης ελέγχου των LEDs

Στη συνάρτηση `loop()` υπάρχουν τέσσερις συναρτήσεις και οι τελευταίες τρεις ξεκινούν με `"/`. Αυτό σημαίνει ότι η γραμμή αντιμετωπίζεται ως σχόλιο και δεν εκτελείται. Για να αλλάξετε τη

συνάρτηση ελέγχου των LEDs από την **oneAfterAnotherNoLoop()**, σε μία από τις άλλες τρεις, βάλτε σχόλιο στην **oneAfterAnotherNoLoop()**, και σβήστε τα από μια άλλη, όπως παρακάτω.

Πατήστε Αποθήκευση ως (**Ctrl+Shift+s**), και μεταβείτε έξω από το φάκελο του project CIRC_02 αν είστε μέσα σ' αυτόν.

Δώστε όνομα φακέλου CIRC_02_d1 για να αποθηκεύσετε το πρόγραμμα.

Στη συνέχεια, φορτώστε το στην πλακέτα (**Ctrl+u**).

Επιβεβαιώστε την ορθή λειτουργία του κυκλώματος.

*(Σημείωση: Οι **oneAfterAnotherNoLoop()** και **oneAfterAnotherLoop()** έχουν την ίδια λειτουργία, απλά διαφέρουν ως προς την υλοποίηση)*

6.2 Δημιουργία μιας δικιάς μας συνάρτησης για τον έλεγχο των LEDs

Στο τέλος του προγράμματος, μετά τη συνάρτηση inAndOut(), ορίστε μια συνάρτηση με όνομα my_animation() η οποία θα υλοποιεί ένα animation που θα γράψετε εσείς.

Ενδεικτικά η δομή μιας συνάρτησης είναι:

```
void <όνομα συνάρτησης>()  
{  
    <Ορισμοί μεταβλητών>  
    <Εντολές>  
}
```

Μόλις ολοκληρώσετε τη συγγραφή της συνάρτησης, πηγαίνετε στη συνάρτηση **loop()** και καλέστε την γράφοντας **my_animation()**;

Το animation που πρέπει να υλοποιήσετε είναι το animation που υπάρχει στο [video](#). Θα πρέπει να χρησιμοποιήσετε τουλάχιστον ένα for-loop.

Μην παραλείψετε να κάνετε σχόλια όλες τις άλλες κλήσεις συναρτήσεων που περιέχονται στην loop().

Πατήστε Αποθήκευση ως (**Ctrl+Shift+s**), μεταβείτε έξω από το φάκελο του ήδη αποθηκευμένου project, αν είστε μέσα σ' αυτόν, (όμοια με πριν θα πρέπει να βλέπετε τους φακέλους CIRC_02, CIRC_02_d1) και δώστε όνομα φακέλου **CIRC_02_d2** για να αποθηκεύσετε το πρόγραμμα.

Στη συνέχεια, φορτώστε το στην πλακέτα (**Ctrl+u**). Επιβεβαιώστε ότι η συνάρτησή σας δουλεύει όπως θα έπρεπε.