

## Εργαστήριο Εκπαιδευτικής Ρομποτικής I (Arduino)

## Φύλλο Εργαστηριακής Άσκησης 3:

**Ξεκαθαρίζουμε Βασικές έννοιες και διαδικασίες**

Όνοματεπώνυμο:

Ημερομηνία:

Ομάδα:

**Σύνδεση με τις προηγούμενες συναντήσεις****Η βασική ιδέα των αυτοματισμών / physical computing**

Με βάση ένα πρόγραμμα, ο μικροελεγκτής δέχεται (συνήθως) κάποια είσοδο ή εισόδους και ελέγχει μια ή περισσότερες εξόδους. Μπορεί όμως (όπως στην περίπτωση των συνηθισμένων φαναριών κυκλοφορίας) να μην υπάρχει κάποια είσοδος.

Το πρόγραμμα το γράφουμε στο περιβάλλον IDE και το ανεβάζουμε στον μικροελεγκτή.

**Το κάθε ΠΡΟΓΡΑΜΜΑ συνεργάζεται με τα ΣΥΓΚΕΚΡΙΜΜΕΝΑ κάθε φορά ΚΥΚΛΩΜΑΤΑ**

**Και στο πρόγραμμα και στα κυκλώματα ΥΠΑΡΧΟΥΝ πάντα ΠΟΛΛΕΣ ΛΥΣΕΙΣ που κάνουν τη δουλειά, φυσικά με πλεονεκτήματα και μειονεκτήματα σχεδιασμού κάθε φορά.**

Υπάρχουν Αναλογικές και Ψηφιακές Είσοδοι και Έξοδοι.

Καλό είναι να τα ξεκαθαρίσουμε, ώστε να μας είναι εύλογο σε ποιες υποδοχές (pins) του Arduino συνδέουμε κάθε κύκλωμα, για να πάρουμε πληροφορίες ή για το ελέγξουμε.



**Τα υλικά**

Εκτός από τα βασικά μας υλικά (μικροελεγκτής Arduino Uno, breadboard, καλώδια, μπαταρίες) έχουμε χρησιμοποιήσει αντιστάσεις (σταθερής τιμής), led διαφόρων χρωμάτων και διακόπτες πίεσης (pushbuttons).

Υπενθυμίζουμε ότι τα led (φωτοдиодοι) έχουν πολικότητα (+ και -) στη συνδεσμολογία τους.

Επίσης θυμίζουμε την συνδεσμολογία του διακόπτη πίεσης (pushbutton).

Σήμερα θα προσθέσουμε στη συλλογή μας δυο μεταβλητές αντιστάσεις

	<b>Η φωτοαντίσταση (LDR)</b> είναι ένας μεταβλητός αντιστάτης του οποίου η αντίσταση μεταβάλλεται ανάλογα με την φωτεινότητα.
	<b>Το ποτενσιόμετρο</b> είναι ένας μεταβλητός αντιστάτης (αντίσταση) του οποίου η τιμή της αντίστασης μεταβάλλεται στρέφοντας μια επαφή.

**Εξασκούμαστε στη βασική λογική Εισόδου/Εξόδου Αναλογικού/Ψηφιακού**

Στις ακόλουθες περιπτώσεις, συμπληρώστε το πλήθος (0, 1, 2, 3, ...) και το είδος (Analog/ Digital / - ) των κυκλωμάτων εισόδου και εξόδου σε έναν «αυτοματισμό».

Παρατηρήστε πως ενώ μπορεί να μην υπάρχει «είσοδος», δεν έχει νόημα να μην υπάρχει «έξοδος».

Περιγραφή περίπτωσης	Είσοδος		Έξοδος	
	Πλήθος (-, 1, 2, ...)	Είδος (- / A / D)	Πλήθος (1, 2, ...)	Είδος (A / D)
Ένα κόκκινο led ανάβει και σβήνει κάθε ένα δευτερόλεπτο	-	-	1	D
Ένα κόκκινο led και ένα πράσινο led, <b>ανάβουν και σβήνουν εναλλάξ</b> , κάθε ένα δευτερόλεπτο				
Δυο κίτρινα led <b>συνδεδεμένα σε σειρά</b> , ανάβουν και σβήνουν (προφανώς μαζί) κάθε ένα δευτερόλεπτο				
Με τον κατάλληλο προγραμματισμό ελέγχω την λειτουργία ενός φωτεινού σηματοδότη (οχημάτων και πεζών), σε μια διασταύρωση				
Όσο πατάω έναν <b>διακόπτη πίεσης</b> (pushbutton), ανάβει ένα led, και όταν δεν τον πατάω το led παραμένει σβηστό				
Ένα led ανάβει <b>αυξάνοντας σταδιακά την φωτεινότητα</b> του και μετά την μειώνει σταδιακά. Αυτό επαναλαμβάνεται συνέχεια.				
Όταν με έναν αισθητήρα φωτός, <b>θεωρώ πως είναι μέρα</b> , σβήνουν τα φώτα (ένα led), και <b>όταν θεωρώ πως είναι νύχτα</b> (είναι δηλ. αρκετά σκοτεινά) τότε ανάβουν τα φώτα (ένα led)				
Με έναν αισθητήρα φωτός, ελέγχω το επίπεδο φωτισμού του χώρου και <b>ανάλογα ανάβουν με μεταβαλλόμενη ένταση φωτισμού</b> τα φώτα (ένα led)				
Με έναν αισθητήρα θερμοκρασίας μετρώ την θερμοκρασία του χώρου και αν είναι πάνω από 20°C <b>αρχίζει να λειτουργεί</b> ένας ανεμιστήρας (αλλιώς σβήνει)				
Με έναν αισθητήρα θερμοκρασίας μετρώ την θερμοκρασία του χώρου και αν είναι πάνω από 20°C αρχίζει να λειτουργεί ένας ανεμιστήρας, αν είναι πάνω από 30 °C αρχίζει να λειτουργεί <b>και ένας δεύτερος</b> ανεμιστήρας				
Με ένα ποτενσιόμετρο * (μεταβλητή αντίσταση) <b>ελέγχω πόσο πολύ ή λίγο θα φωτοβολεί ένα led</b>				
Με τρία ποτενσιόμετρα (μεταβλητές αντιστάσεις) ελέγχω πόσο πολύ ή λίγο θα φωτοβολούν τρία led (κόκκινο, πράσινο, μπλε) με σκοπό να φτιάξω πολλές διαφορετικές αποχρώσεις (χρωματικό μοντέλο RGB/Red-Green-Blue)				
Όταν <b>πλησιάζω σε μια απόσταση</b> από την πόρτα του super market, αυτή ανοίγει αυτόματα				
Όταν περάσω <b>μπροστά από έναν ανιχνευτή</b> , λίγο πριν την πόρτα του super market, η πόρτα αυτή ανοίγει αυτόματα				
Με ένα σύστημα που μετράω την υγρασία στο έδαφος, ανάβει ένα πράσινο led αν είναι ποτισμένο ή ένα κόκκινο led αν χρειάζεται πότισμα				
Όταν πλησιάσω το χέρι μου κοντά σε έναν «έξυπνο» κάδο ανακύκλωσης, ανοίγει το καπάκι του αυτόματα				
<i>παρακάτω περιγράψτε δυο δικές σας ιδέες αυτοματισμών</i>				

(\*) Ποτενσιόμετρο είναι μια αντίσταση που αλλάζει τιμές από μηδέν μέχρι μια συγκεκριμένη τιμή π.χ. 10 ΚΩ

### Επαναληπτικές δραστηριότητες (εμπέδωση)

Δεν ξεχνάμε ότι χρειάζεται κάθε φορά να φτιάξουμε τα κατάλληλα κυκλώματα και να φτιάξουμε στο περιβάλλον IDE, το κατάλληλο πρόγραμμα που μετά θα το ανεβάσουμε στην πλακέτα του Arduino.

Μπορείτε να γράψετε όσο απλά θέλετε το πρόγραμμα (πηγαίο κώδικα), αρκεί να το καταλαβαίνετε.

#### Επαναληπτική δραστηριότητα 1: Φτιάχνουμε ξανά ένα φανάρι οχημάτων (μόνο)

Σχεδιάστε το κύκλωμα / τα κυκλώματα που απαιτούνται όσο καλύτερα μπορείτε.

Γράψτε ένα προσχέδιο του προγράμματος / πηγαίου κώδικα

**ΠΡΙΝ ΤΑ  
ΥΛΟΠΟΙΗΣΕΤΕ**

---

Θυμηθείτε να κρατάτε τα στιγμιότυπα της ομάδας σας από την εργασία σας  
φωτογραφίες, βίντεο, πηγαίο κώδικα  
(έχετε πλέον σταθερές θέσεις και υπολογιστές)

## Επαναληπτική δραστηριότητα 2: Πειραματιζόμαστε με ένα διακόπτη pushbutton

---

Φτιάχνουμε έναν αυτοματισμό που ανάβει ένα πράσινο led όταν πατάμε ένα διακόπτη, ενώ όταν δεν τον έχουμε πατημένο ανάβει συνέχεια ένα κόκκινο led (που σβήνει όταν ανάβει το πράσινο).

Σχεδιάστε το κύκλωμα / τα κυκλώματα που απαιτούνται όσο καλύτερα μπορείτε.

Γράψτε ένα προσχέδιο του προγράμματος / πηγαίου κώδικα

**ΠΡΙΝ ΤΑ  
ΥΛΟΠΟΗΣΕΤΕ**

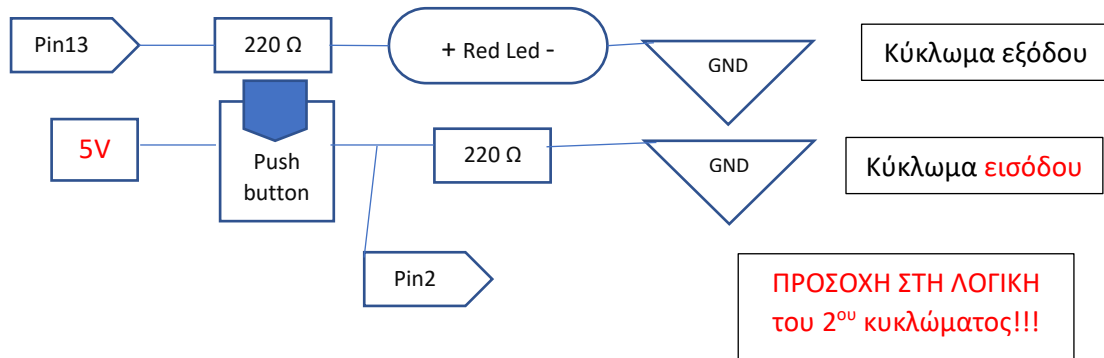
---

Θυμηθείτε να κρατάτε τα στιγμότυπα της ομάδας σας από την εργασία σας  
φωτογραφίες, βίντεο, πηγαίο κώδικα  
(έχετε πλέον σταθερές θέσεις και υπολογιστές)

ΘΥΜΑΜΑΙ ΑΠΟ ΤΟ ΠΡΗΓΟΥΜΕΝΟ ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ

Το push button ως ψηφιακή είσοδος

Πρόχειρο σχέδιασμα κυκλώματος



Πηγαίος κώδικας (με σχόλια) &amp; ΚΥΚΛΩΜΑ

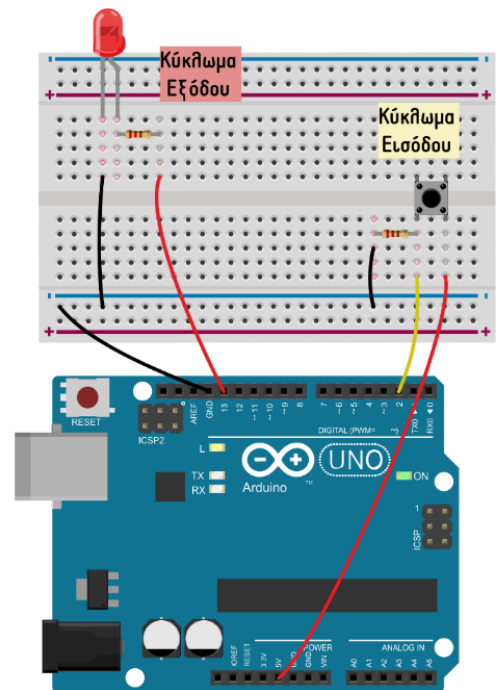
```
// Οι σταθερές ΔΕΝ μπορούν να αλλάξουν τιμή μέσα στο
// πρόγραμμα
const int redledPin = 13; // LED pin (έστω κόκκινο)
const int buttonPin = 2; // pushbutton pin

int buttonStatus = 0; // μεταβλητή (variable) για
// ανάγνωση (είσοδο) του
// pushbutton status

void setup() {
  pinMode(redledPin, OUTPUT); // LED pin output
  pinMode(buttonPin, INPUT); // pushbutton pin input
}



void loop() {
  buttonStatus = digitalRead(buttonPin); // «διαβάζω»
  // την κατάσταση του
  // pushbutton

  // Ελέγγω εάν το pushbutton πατήθηκε,
  // οπότε το buttonState γίνεται HIGH:
  if (buttonStatus == HIGH) {
    digitalWrite(redledPin, HIGH); // LED on
  } else {
    digitalWrite(redledPin, LOW); // LED off
  }
}
```



## Μεταβλητές αντιστάσεις

Εκτός από τις (σταθερές) αντιστάσεις που έχουμε ήδη συναντήσει, υπάρχουν πολλοί ακόμη τύποι αντιστάσεων, κάποιοι από αυτούς είναι μεταβλητοί. Σήμερα θα ασχοληθούμε με δυο από αυτούς.

	<p><b>Η φωτοαντίσταση (LDR)</b> είναι ένας μεταβλητός αντιστάτης του οποίου η αντίσταση μεταβάλλεται ανάλογα με την φωτεινότητα. Το πρώτο πράγμα που θα πρέπει να ανιχνεύσετε είναι αν η τιμή της αυξάνεται ή μειώνεται με την αύξηση του προσπίπτοντος φωτός.</p>
	<p><b>Το ποτενσιόμετρο</b> είναι ένας μεταβλητός αντιστάτης (αντίσταση) του οποίου η τιμή της αντίστασης μεταβάλλεται στρέφοντας μια επαφή.</p>

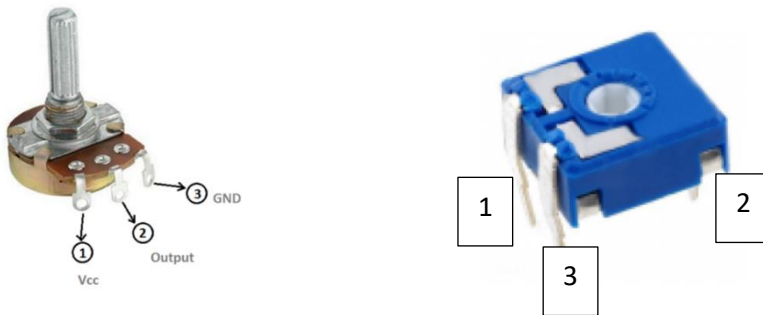
Χρησιμοποιώντας μπαταρίες (ή την τροφοδοσία από το Arduino), καλώδια, breadboard, led και ότι άλλο χρειαστείτε, **ανακαλύψτε τον τρόπο που μεταβάλλεται η αντίσταση μιας φωτοαντίστασης σε σχέση με το φως.** Αυξάνεται ή μειώνεται με την αύξηση του φωτισμού; Η μεταβολή είναι αναλογική/γραμμική ή όχι;

Σχεδιάστε το κύκλωμα που υλοποιήσατε για να διερευνήσετε τον τρόπο λειτουργίας.

### *Πρόχειρο σχέδιασμα κυκλώματος φωτοαντίστασης*

### Παρατηρήσεις:

**Μπορείτε να υλοποιήσετε το αντίστοιχο κύκλωμα με το ποτενσιόμετρο;**



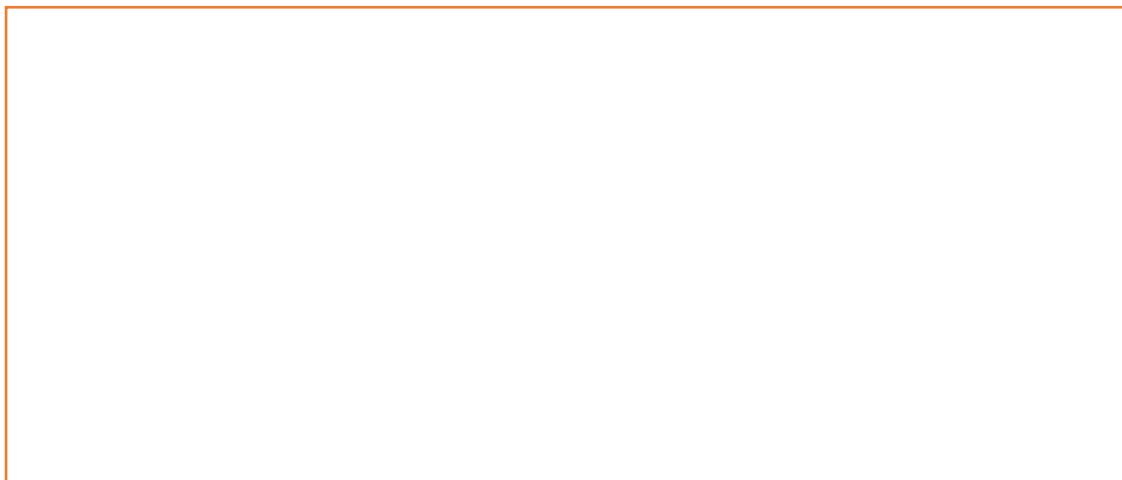
**Αντικαταστήστε την φωτοαντίσταση με ένα ποτενσιόμετρο.**

**ΠΡΟΣΟΧΗ**, θα συνδέσουμε το ένα άκρο και το μεσαίο άκρο του ποτενσιόμετρου. Το τρίτο άκρο θα πρέπει (*by the book*) να το συνδέσουμε στη γείωση/GND. Αλλά θα θέλαμε να πειραματιστείτε και χωρίς αυτή την τελευταία σύνδεση με την γείωση.

**ΠΡΟΣΟΧΗ!** Θα πρέπει να συνδέσετε σε σειρά και μια αντίσταση 220 Ω (γιατί ΑΠΛΑ μπορεί να ρυθμίσουμε το ποτενσιόμετρο στο μηδέν και να κάψουμε το led)

**Πρόχειρο σχεδίασμα κυκλώματος**

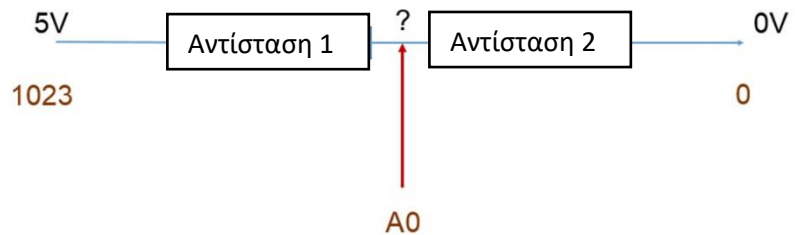
Περιστρέψτε το ποτενσιόμετρο και παρατηρήστε τι συμβαίνει με τη φωτεινότητα του LED.



## Η ιδέα του «Διαιρέτη τάσης»

Κυκλώματα όπως το παραπάνω, το οποίο αποτελείται από δύο αντιστάτες συνδεδεμένους σε σειρά, στα άκρα των οποίων εφαρμόζεται η τάση εισόδου (π.χ. 5V), ονομάζονται διαιρέτες τάσης.

Τάση εξόδου είναι η διαφορά δυναμικού ανάμεσα στους ακροδέκτες της μίας από τις δύο αντιστάσεις. Οι τιμές που μπορεί να πάρει η τάση εξόδου κυμαίνονται από το 0 έως την τάση εισόδου.



## Λαμβάνουμε ενημέρωση από τον «Διαιρέτη τάσης»

Προκειμένου να διαβάζουμε την έξοδο του ποτενσιόμετρου, ας συνδέσουμε με ένα καλώδιο τη στήλη του μεσαίου ακροδέκτη στην αναλογική είσοδο A0 του Arduino (ισοδύναμα μπορείτε να χρησιμοποιήσετε οποιαδήποτε άλλη αναλογική είσοδο).

Εκκινήστε το Arduino IDE, συνδέστε τη συσκευή σας και επιλέξτε το πρόγραμμα:

**File > Examples > Basics > ReadAnalogVoltage**

**Αρχείο > Παραδείγματα > Basics > ReadAnalogVoltage**

Αφού τρέξετε το πρόγραμμα δείτε το αποτέλεσμα στο Serial Monitor επιλέγοντας:

**Tools > Serial Monitor**

**Εργαλεία > Παρακολούθηση Σειριακής**

Τι παρατηρείτε;



## Μια ματιά στον πηγαίο κώδικα.

Παρακάτω απομονώσαμε τον πηγαίο κώδικα που χρησιμοποιήσατε, χωρίς τα σχόλια.

Υπογραμμίστε τα νέα στοιχεία που χρησιμοποιούνται σε αυτό το πρόγραμμα;

```
void setup() {
  Serial.begin(9600);
}

void loop() {
  int sensorValue = analogRead(A0);
  float voltage = sensorValue * (5.0 / 1023.0);
  Serial.println(voltage);
}
```

Καταλαβαίνετε τη λειτουργία τους;

- `Serial.begin(9600);`
- `analogRead(A0);`
- `float voltage = sensorValue * (5.0 / 1023.0);`
- `Serial.println(voltage);`

Αν όχι ρωτήστε!!!

## Λίγο παραπάνω κώδικας

Σύγκριση δύο «λύσεων». Παρατηρήστε και κατανοήστε τις διαφορές

<pre>void setup() {   Serial.begin(9600); }  void loop() {   int sensorValue = analogRead(A0);   float voltage = sensorValue * (5.0 / 1023.0);   Serial.println(voltage); }</pre>	<pre>int sensorValue; float voltage;  void setup() {   Serial.begin(9600); }  void loop() {   sensorValue = analogRead(A0);   voltage = sensorValue * (5.0 / 1023.0);   Serial.println(voltage); }</pre>
---	--

Υπάρχουν δυο (παρόμοιες) εντολές οι `serial.print` και `serial.println`

Δοκιμάστε και βρείτε σε τι διαφέρουν.

Έχουμε μέχρι τώρα δει τις εντολές `DigitalWrite()` και `AnalogRead()`

Μπορείτε να υποθέσετε τι κάνουν οι εντολές ....

`DigitalRead()`

.....

`AnalogWrite()`

.....

Μπορείτε να γράψετε μια (σωστή συντακτικά) υποθετική σύνταξή τους;

Συσχετίστε κάθε μια από τις 4 εντολές read/write με τα ηλεκτρονικά που έχουμε δει ως τώρα

	<code>DigitalWrite</code>	<code>DigitalRead</code>	<code>AnalogWrite</code>	<code>AnalogRead</code>
led				
pushbutton				
ποτενσιόμετρο				
φωτοαντίσταση				

**Το παρακάτω project ΔΕΝ περιμένουμε να το υλοποιήσουμε στη σημερινή μας συνάντηση. Αν όμως κάποια ομάδα μπορέσει μπορεί να ασχοληθεί μαζί του καταρχήν σε επίπεδο σχεδιασμού**

## **PROJECT - Έλεγχος φωτισμού**

---

Τώρα αυτό που ζητάμε είναι για πρώτη φορά να φτιάξουμε έναν αυτοματισμό που ελέγχεται από την κατάσταση στο περιβάλλον του. **Επιτέλους μια πραγματική εισαγωγή στο physical computing!!!**

**ΠΡΟΒΛΗΜΑ: Θέλουμε όταν σκοτεινιάζει να ανάβουν τα φώτα.**

Τι υλικά νομίζετε πως θα χρειαστούν;

Πόσα κυκλώματα θα χρειαστούμε;

Η είσοδος θα είναι Αναλογική ή Ψηφιακή;

Η έξοδος θα είναι Αναλογική ή Ψηφιακή;

Κάντε την απαραίτητη προεργασία και κρατήστε τις απαραίτητες σημειώσεις.  
(Ας πούμε πως θα «μεταφράζατε» την απαίτηση «όταν σκοτεινιάζει»?)

Θυμηθείτε, πως **κρατάτε αρχείο με τις προσπάθειές σας**  
καταρχήν στον υπολογιστή σας ή με όποιο δικό σας τρόπο σας βολεύει,  
με τελικό στόχο να τα αναρτήσετε στα *Εγγραφα* της *Ομάδας* σας, στην *e-class*