

# ENGINE

Teaching online electronics, microcontrollers and programming in Higher Education

---

**Output 2: Online Course for Microcontrollers:  
syllabus, open educational resources**

Practice leaflet: Module\_1-2 pins as inputs

---

**Lead Partner: International Hellenic University (IHU)**

# Δήλωση

Αυτό το αρχείο συντάχθηκε στο πλαίσιο του έργου ENGINE. Όπου έχουν χρησιμοποιηθεί άλλα δημοσιευμένα και αδημοσίευτα υλικά, αυτά έχουν αναγνωρισθεί.

## Πνευματική ιδιοκτησία

© Copyright 2021 - 2023 the [ENGINE](#) Consortium

Warsaw University of Technology (Poland)

International Hellenic University (IHU) (Greece)

European Lab for Educational Technology- EDUMOTIVA (Greece)

University of Padova (Italy)

University of Applied Sciences in Tarnow (Poland)

Όλα τα δικαιώματα διατηρούνται.



Αυτό το έγγραφο έχει άδεια Creative Commons Attribution-NonCommercial- NoDerivatives 4.0 International License.

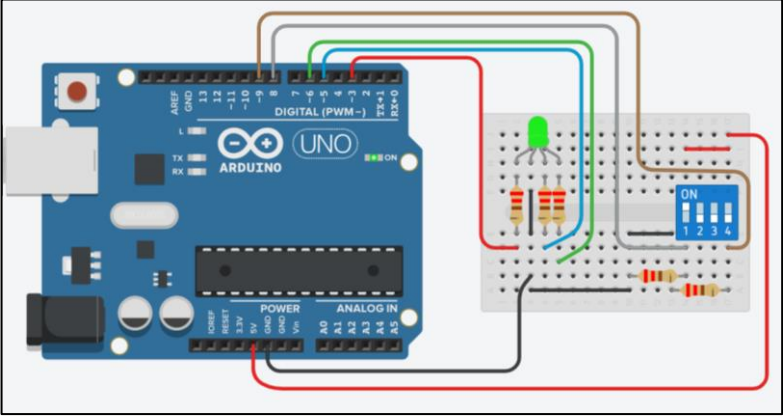
Αυτό το έργο έχει χρηματοδοτηθεί με την υποστήριξη της Ευρωπαϊκής Επιτροπής. Αυτή η έκθεση αντικατοπτρίζει μόνο τις απόψεις του συγγραφέα και η Επιτροπή δεν μπορεί να θεωρηθεί υπεύθυνη για οποιαδήποτε χρήση των πληροφοριών που περιέχονται σε αυτήν.

# Πίνακας Περιεχομένων

Δραστηριότητα 1. Switches .....	1
Δραστηριότητα 2. Push-buttons.....	3
Ανακεφαλαίωση.....	4

## Δραστηριότητα 1. Switches

Αυτή η δραστηριότητα χρησιμοποιεί διακόπτες που προκαλούν σήματα εισόδου στο Arduino Uno. Βασικός στόχος είναι η κατανόηση της συνδεσμολογίας του κυκλώματος αλλά και του κώδικα για την αναγνώριση των σημάτων εισόδου από διακόπτες.

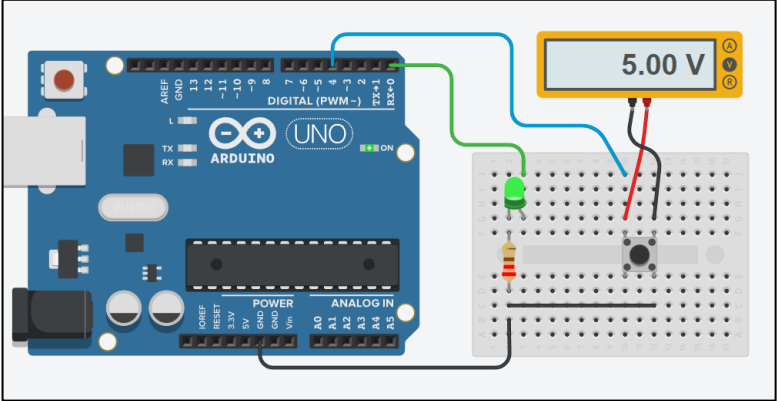
<p>Δραστηριότητα 1 (25 λεπτά)</p>	<p>Στόχος είναι το Arduino Uno να διαβάζει τις καταστάσεις σε 2 pins εισόδου, και να οδηγεί ένα RGB LED ανάλογα με τους συνδυασμούς των εισόδων:</p> <p>Πίνακας 1. Πιθανές καταστάσεις εισόδων και RGB LED</p> <table border="1" data-bbox="683 730 1139 929"> <thead> <tr> <th>Switch_1</th> <th>Switch_4</th> <th>RGB LED</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>OFF</td> </tr> <tr> <td>0</td> <td>1</td> <td>Κόκκινο</td> </tr> <tr> <td>1</td> <td>0</td> <td>Πράσινο</td> </tr> <tr> <td>1</td> <td>1</td> <td>Μπλε</td> </tr> </tbody> </table> <p><b>Βήμα 1.</b> Υλοποιείται το κύκλωμα  <b>Βήμα 2.</b> Γράφεται ο κώδικας του μικροελεγκτή  <b>Βήμα 3.</b> Γίνεται έλεγχος λειτουργίας του κυκλώματος</p>	Switch_1	Switch_4	RGB LED	0	0	OFF	0	1	Κόκκινο	1	0	Πράσινο	1	1	Μπλε
Switch_1	Switch_4	RGB LED														
0	0	OFF														
0	1	Κόκκινο														
1	0	Πράσινο														
1	1	Μπλε														
<p>Βήμα 1 (8 λεπτά)</p>	<p>Να υλοποιηθεί το επόμενο κύκλωμα.</p>  <p>Εικόνα.1.1 Switches και RGB LED</p>															
<p>Βήμα 2 (15 λεπτά)</p>	<p>Να μελετηθεί ο κώδικας, <b>να συμπληρωθεί</b> και να γραφτεί στον μικροελεγκτή:</p> <hr/> <p>/* Switches and RGB LED</p> <p>Συνδεσμολογία κυκλώματος:  PIN_3 =&gt; Αντίσταση 220Ω =&gt; Κόκκινο pin από το RGB LED  PIN_5 =&gt; Αντίσταση 220Ω =&gt; Μπλε pin από το RGB LED  PIN_6 =&gt; Αντίσταση 220Ω =&gt; Πράσινο pin από το RGB LED</p>															

## 2. Pins as inputs - Practice leaflet

	<pre>PIN_8 =&gt; Pull down Αντίσταση (220Ω) =&gt; switch_1 (Vcc) PIN_9 =&gt; Pull down Αντίσταση (220Ω) =&gt; switch_4 (Vcc) */  #define R_pin 3 //ονομάζουμε "R_pin" το PIN_3 #define G_pin 6 //ονομάζουμε "G_pin" το PIN_6 #define B_pin 5 //ονομάζουμε "B_pin" το PIN_5 #define Sw1_pin 8 //ονομάζουμε "Sw1_pin" το PIN_8 #define Sw4_pin 9 //ονομάζουμε "Sw4_pin" το PIN_9  //Η συνάρτηση setup() περιέχει αρχικοποιήσεις //Εκτελείται μόνο μία φορά κατά την τροφοδοσία του Arduino Uno void setup() {   //Ρυθμίζεται το pin 3, pin 5, pin 6 ως output (έξοδος)   //Ρυθμίζεται το pin 8, pin 9 ως input (είσοδος)   pinMode(R_pin, OUTPUT);   pinMode(G_pin, OUTPUT);   pinMode(B_pin, OUTPUT);   pinMode(Sw1_pin, INPUT);   pinMode(Sw4_pin, INPUT); }  //Αυτή η συνάρτηση εκτελείται διαρκώς void loop() {   if(digitalRead(Sw1_pin)==0 &amp;&amp; digitalRead(Sw4_pin)==0){     //RGB LED = OFF     analogWrite(R_pin, 0); // 0% PWM στο pin 3     analogWrite(G_pin, 0); // 0% PWM στο pin 6     analogWrite(B_pin, 0); // 0% PWM στο pin 5     delay(1000); // Αναμονή για 1 δευτερόλεπτο   }   else if(digitalRead(Sw1_pin)==0 &amp;&amp; digitalRead(Sw4_pin)==1){     //κόκκινο χρώμα RGB =&gt; R=255, G=0, B=0     analogWrite(R_pin, 255); // 100% PWM στο pin 3     // 0% PWM στο pin 6     analogWrite(B_pin, 0); // 0% PWM στο pin 5     delay(1000); // Αναμονή για 1 δευτερόλεπτο   }   else if(digitalRead(Sw1_pin)==1 &amp;&amp; digitalRead(Sw4_pin)==0){     //πράσινο χρώμα RGB =&gt; R=0, G=255, B=0     analogWrite(R_pin, 0); // 0% PWM στο pin 3     // 100% PWM στο pin 6     analogWrite(B_pin, 0); // 0% PWM στο pin 5     delay(1000); // Αναμονή για 1 δευτερόλεπτο   }   else{ //if(digitalRead(Sw1_pin)==1 &amp;&amp; digitalRead(Sw4_pin)==1)     //μπλε χρώμα =&gt; RGB=0,0,255     analogWrite(R_pin, 0); // 0% PWM στο pin 3     analogWrite(G_pin, 0); // 0% PWM στο pin 6     analogWrite(B_pin, 255); // 100% PWM στο pin 5     delay(1000); // Αναμονή για 1 δευτερόλεπτο   } } }</pre>
Βήμα 3 (2 λεπτά)	Έλεγχος σωστής λειτουργίας του κυκλώματος

## Δραστηριότητα 2. Push-buttons

Αυτή η δραστηριότητα χρησιμοποιεί push-buttons που προκαλούν σήματα εισόδου στο Arduino Uno. Βασικός στόχος είναι η κατανόηση της συνδεσμολογίας του κυκλώματος αλλά και του κώδικα για την αναγνώριση των σημάτων εισόδου από push-buttons.

<p>Δραστηριότητα 2 (25 λεπτά)</p>	<p>Ζητούμενο είναι το Arduino Uno να διαβάσει τις καταστάσεις από ένα push-button. Ένα LED <b>ακολουθεί τις καταστάσεις</b> από το pin εισόδου στο οποίο είναι συνδεδεμένο το push-button.</p> <ul style="list-style-type: none"> <li>• Το push-button συνδέεται στο PIN_4. Για να μην τοποθετηθεί εξωτερική αντίσταση, ενεργοποιείται η ενσωματωμένη <b>pullup</b> αντίσταση με αντίστοιχη εντολή στην pinMode()</li> <li>• Στο κύκλωμα προστέθηκε ένα βολτόμετρο για την μέτρηση της τάσης που εμφανίζεται στο pin εισόδου</li> </ul> <p><i>Tip.</i> Με αυτήν την συνδεσμολογία όταν πατηθεί το push-button δίνει 0 στην είσοδο και το LED απενεργοποιείται</p> <p><b>Βήμα 1.</b> Υλοποιείται το κύκλωμα  <b>Βήμα 2.</b> Γράφεται ο κώδικας του μικροελεγκτή  <b>Βήμα 3.</b> Γίνεται έλεγχος λειτουργίας του κυκλώματος  <b>Βήμα 4.</b> Τροποποιήσεις και συζήτηση</p>
<p>Βήμα 1 (6 λεπτά)</p>	<p>Να υλοποιηθεί το επόμενο κύκλωμα.</p>  <p>Εικόνα.1.2 Push-button και LED</p>
<p>Βήμα 2 (6 λεπτά)</p>	<p>Να μελετηθεί ο κώδικας, <b>να συμπληρωθεί</b> και να γραφτεί στον μικροελεγκτή:</p> <hr/> <p>/* <b>Push-button και LED</b></p> <p>Συνδεσμολογία κυκλώματος:          PIN_0 =&gt; LED_Anode - LED_Cathode =&gt; Αντίσταση 220Ω =&gt; Gnd          PIN_4 =&gt; Pull up αντίσταση (built in) =&gt; push-button (Gnd)</p>

## 2. Pins as inputs - Practice leaflet

	<pre>*/  #define led_pin 0 //ονομάζουμε "led_pin" το PIN_0 #define pb_pin 4 //ονομάζουμε "pb_pin" το PIN_4  //Η συνάρτηση setup() περιέχει αρχικοποιήσεις //Εκτελείται μόνο μία φορά κατά την τροφοδοσία του Arduino Uno void setup() {   //Ρυθμίζεται το PIN_0 ως output    //Ρυθμίζεται το PIN_4 ως input με ενεργοποιημένη την pullup αντίσταση   pinMode(pb_pin, INPUT_PULLUP); }  //Αυτή η συνάρτηση εκτελείται διαρκώς void loop() {   //Το LED ακολουθεί την κατάσταση του push-button   digitalWrite(led_pin, digitalRead(pb_pin)); }</pre>
Βήμα 3 (3 λεπτά)	Έλεγχος σωστής λειτουργίας του κυκλώματος
Βήμα 4 (10 λεπτά)	<p>Προτεινόμενες τροποποιήσεις και συζήτηση:</p> <ul style="list-style-type: none"><li>• Πως θα έπρεπε να τροποποιηθεί ο κώδικας ώστε το LED να εμφανίζει την αντίστροφη κατάσταση από αυτήν που υπάρχει στο pin εισόδου;</li><li>• Τι θα άλλαζε για το pin εισόδου αν δεν ήταν ενεργοποιημένη η ενσωματωμένη pull up αντίσταση;</li><li>• Τι θα έπρεπε να αλλάξει ώστε όταν πατηθεί το push-button το Arduino Uno να βλέπει 5V;</li><li>• Χρειάζεται ή όχι debounce time;</li></ul>

### Ανακεφαλαίωση

Χρησιμοποιήθηκαν βασικές εντολές προγραμματισμού του Arduino, όπως:

- delay()
- analogWrite()
- digitalWrite()
- digitalRead()

Μέσα από τις δραστηριότητες αξιοποιήθηκαν pins του Arduino ως είσοδοι για να διαβάσουν καταστάσεις από

- switches
- push-buttons