4 Εισαγωγή στο Arduino

Τι είναι το Arduino;

To arduino είναι ένας μικροελεγκτής, προσαρμοσμένος σε μια πλακέτα και έτοιμος προς χρήση.

Μας φώτισες. Και τι είναι ένας μικροελεγκτής;

Φανταστείτε το μικροελεγκτή σαν ένα τσιπάκι που περιέχει μέσα του ένα ολόκληρο μικρό υπολογιστή. Είναι ένας υπολογιστής ειδικού σκοπού: δεν θα τον χρησιμοποιήσετε για να γράψετε ένα κείμενο αλλά μπορείτε να τον προγραμματίσετε να ελέγχει άλλες συσκευές: να αναβοσβήνει φωτάκια (LED), να περιστρέφει κινητήρες κλπ.

Που χρησιμοποιείται;

Παντού γύρω μας, χωρίς συχνά να του δίνουμε σημασία. Σχεδόν όλες οι ηλεκτρονικές συσκευές που διαθέτετε έχουν τουλάχιστον ένα μέσα τους. To touchpad στο laptop που γράφω τις σημειώσεις αυτή τη στιγμή ελέγχεται από ένα μικροελεγκτή: διαβάζει την επιφάνεια αφής και στέλνει τα αντίστοιχα δεδομένα στην κεντρική πλακέτα του υπολογιστή. Το ψυγείο σας, ο φούρνος μικροκυμάτων, το κινητό, η τηλεόραση, το αυτοκίνητο, ο διεθνής διαστημικός σταθμός και τα ρομποτάκια που στείλαμε στον Άρη είναι γεμάτα μικροελεγκτές.

Σε τι διαφέρει από ένα κανονικό υπολογιστή;

Σε αντίθεση με τον υπολογιστή που χρησιμοποιούμε για τις καθημερινές μας εργασίες, ένας μικροελεγκτής έχει τα παρακάτω χαρακτηριστικά και διαφορές:

- Δεν διαθέτει λειτουργικό σύστημα. Δεν πρόκειται να βάλετε Windows σε ένα μικροελεγκτή!
- Τρέχει κάθε φορά ένα και μοναδικό πρόγραμμα που πρόκειται να γράψουμε εμείς και κάνει μια συγκεκριμένη λειτουργία.
- Κάθε φορά που τον ενεργοποιούμε αυτόματα αρχίζει να εκτελεί το πρόγραμμα που του βάλαμε πριν. Το πρόγραμμα αποθηκεύεται μόνιμα μέσα στο ίδιο το κύκλωμα. Οι μικροελεγκτές δεν διαθέτουν σκληρούς δίσκους αλλά flash μνήμη (όπως το usb stick που έχετε στη τσέπη σας) και μια μικρή ποσότητα μνήμης RAM για χρήση από το πρόγραμμα.
- Οι μικροελεγκτές συνήθως λειτουργούν με ελάχιστη ενέργεια. Οι περισσότεροι μπορούν να δουλέψουν για αρκετό διάστημα με μια μπαταρία ρολογιού. Πολλά μοντέλα διαθέτουν κατάσταση χαμηλής ενέργειας και παραμένουν σχεδόν σβηστά αν δεν χρειάζεται να κάνουν κάτι.

- Οι μικροελεγκτές είναι συνήθως πολύ φτηνότεροι από οποιοδήποτε υπολογιστή.
- Διαθέτουν ακροδέκτες που μπορούμε να χρησιμοποιήσουμε ως "εισόδους" (για να συνδέσουμε αισθητήρες θερμοκρασίας, διακοπτάκια κλπ) και "εξόδους" που μπορούμε να συνδέσουμε μικρούς ηλεκτρικούς κινητήρες, LED ή ακόμα και μικρές οθόνες.

Δηλ. το Arduino είναι ένας μικροελεγκτής;

Μικροελεγκτής είναι μόνο το τσιπάκι. Αυτό από μόνο του δεν θα ήταν εύκολο να το κάνουμε κάτι: χρειαζόμαστε ένα τρόπο για να το προγραμματίσουμε (απαιτεί να το συνδέσουμε σε ένα κανονικό υπολογιστή) και κάποιο εύκολο σύστημα για να του συνδέσουμε έξτρα εξαρτήματα (LED, κινητήρες, διακοπτάκια κλπ). Αν και θα μπορούσαμε να ξεκινήσουμε από το μηδέν για να τα φτιάξουμε όλα αυτά, είναι πολύ πιο εύκολο να αγοράσουμε ένα σύστημα που να μας δίνει ένα μικροελεγκτή έτοιμο προς χρήση. Ένα τέτοιο σύστημα είναι το Arduino το οποίο χρησιμοποιεί συνήθως τους μικροελεγκτές AVR της εταιρίας Atmel. Μας παρέχεται ως πλακέτα έτοιμη για χρήση.

Πόσο κοστίζει ένα Arduino; Που θα βρούμε;

Ανάλογα με το μοντέλο κοστίζει μέχρι μερικές δεκάδες ευρώ. Το πιο συνηθισμένο μοντέλο είναι το Arduino Uno το οποίο στοιχίζει γύρω στα 10 ευρώ (και λιγότερο σε μεγάλες παραγγελίες από Κίνα!) Μεγαλύτερα μοντέλα μπορεί να κάνουν μέχρι και 50-60 ευρώ. Τα ακριβά μοντέλα διαθέτουν ταχύτερους μικροελεγκτές με περισσότερη μνήμη και είναι κατάλληλα για πολύ πιο πολύπλοκες εφαρμογές (έλεγχος πυραυλικών συστημάτων!)

Θα σας δώσω από ένα Arduino και μερικά ακόμα εξαρτήματα **στον καθένα** σας για να πάρετε σπίτι σας και να πειραματιστείτε. Στις σημειώσεις αυτές θα βρείτε προγραμματάκια για να παίξετε και να δοκιμάσετε. Η τελική μας εφαρμογή του Enigma θα χρησιμοποιήσει κάποιο μεγαλύτερο μικροελεγκτή ή συνδυασμό μικροελεγκτών.

Έχουμε δηλαδή δέκα Arduino; Όταν τελειώσει το Enigma τι θα τα κάνουμε; Είκοσι, για να είμαι ακριβής :D

Όποιος μου αποδείξει ότι τον ενδιαφέρει και πρόκειται να συνεχίσει να το χρησιμοποιεί για να μαθαίνει, θα του το χαρίσω. Οι υπόλοιποι θα τα επιστρέψετε και θα γίνουν δωρεά στο εργαστήριο ηλεκτρονικών και πληροφορικής του σχολείου.

Arduino ακούω και Arduino δε βλέπω...

Αυτό είναι το Arduino Uno (έχουμε 10):



Και αυτό το Arduino Nano (έχουμε 10):



Πως συνδέουμε εξαρτήματα;

Χρησιμοποιούμε τους ακροδέκτες που βλέπετε και συνδέουμε καλωδιάκια (jumper cables) τα οποιά καταλήγουν σε μια breadboard όπου φτιάχνουμε το υπόλοιπο κύκλωμα. Το Arduino Nano μπορεί να τοποθετηθεί το ίδιο απευθείας πάνω στη breadboard.

Breadboard; Jumper cables;

Breadboard είναι το παρακάτω εξάρτημα:



Μας επιτρέπει να συνδέουμε και να ξεσυνδέουμε εξαρτήματα χωρίς να τα χαλάμε και τη χρησιμοποιούμε για να κάνουμε πειραματικές διατάξεις. Τα jumper cables είναι καλωδιάκια που χρησιμοποιούνται για να συνδέσουν τα εξαρτήματα μεταξύ τους.

Οι τρυπούλες στη breadboard ενώνονται μεταξύ τους κάθετα. Υπάρχει όμως ένα χώρισμα στη μέση. Π.χ. το παρακάτω κύκλωμα δείχνει πως θα φτιάχναμε ένα κύκλωμα με ένα LED, μια αντίσταση και μια μπαταρία:





Αντίσταση;

Είναι ένα βασικό εξάρτημα της ηλεκτρονικής – χρειάζεται για να περιορίζει

το ρεύμα και είναι απαραίτητη για να λειτουργήσει το LED. Οι αντιστάσεις βγαίνουν σε διάφορες τιμες (τα χρώματα που έχουν πάνω χρησιμοποιούνται για την αναγνώριση τους). Θα σας δώσω τις κατάλληλες αντιστάσεις για τα πειράματα σας και δεν χρειάζεται να μάθετε περισσότερες λεπτομέρειες (αν δεν θέλετε. Αν θέλετε εδώ είμαστε).

4.1 Ελέγχοντας ένα LED με ένα Arduino

Τώρα που είδατε πως μπορείτε να ενεργοποιήσετε ένα LED με μια μπαταρία, ας βάλουμε και το Arduino στο κύκλωμα μας. Εδώ είναι το παράδειγμα μας με το Arduino Nano:



fritzing

Θα θέλαμε να κάνουμε το LED να αναβοσβήνει χρησιμοποιώντας το μικροελεγκτή μας. Αυτό είναι συνήθως το πρώτο απλό πρόγραμμα που γράφουμε σε ένα μικροελεγκτή για να δούμε πως δουλεύει. Στα προγράμματα που κατεβάσατε υπάρχει ένα με την ονομασία Arduino IDE. Αυτό πρέπει να το εγκαταστήσετε στον υπολογιστή σας, πιέζοντας απλά μερικά Next. Το πρόγραμμα αυτό χρησιμοποιείται για να γράφουμε κώδικα για το Arduino.

Το πρόγραμμα για να κάνουμε το LED να αναβοσβήνει υπάρχει στα έτοιμα παραδείγματα. Πηγαίνετε στο File -> Examples -> Basics -> Blink. Δεν χρειάζεται να το πληκτρολογήσετε!



Από το μενού Tools -> Board επιλέξτε Arduino Nano και από το μενού Tools -> Port την επιλογή COM που εμφανίζει (π.χ. COM4). Πριν εκτελεστεί ο κώδικας, θα πρέπει να μεταγλωττιστεί και να ελεγχθεί πιέζοντας το εικονίδιο Verify και τέλος να μεταφερθεί στο Arduino, πιέζοντας το εικονίδιο Upload.

Θα δείτε το LED να αναβοσβήνει. Εκτός από αυτό που έχουμε συνδέσει εμείς θα δείτε και ένα LED στην πλακέτα να αναβοσβήνει. Στην πραγματικότητα είναι συνδεμένα και τα δύο στην ίδια έξοδο (ψηφιακή έξοδος 13) οπότε αναβοσβήνουν μαζί.

Τρεις απλές ασκήσεις:

- 1. Μπορείτε να μετατρέψετε το πρόγραμμα και το κύκλωμα ώστε να αναβοσβήνουν το LED αν το συνδέσετε στην έξοδο 12; Τι θα πρέπει να αλλάξετε;
- Μπορείτε να κάνετε το LED να αναβοσβήνει πιο γρήγορα (hint: δείτε την εντολή delay);
- Μπορείτε να προσθέσετε γραμμές στο πρόγραμμα ώστε να αναβοσβήνουν το LED στην έξοδο 12 και το LED πάνω στο Arduino στην έξοδο 13;

4.2 Πως δουλεύει το πρόγραμμα;

Το πρόγραμμα μας, χωρίς τα σχόλια, δείχνει κάπως έτσι:



Θα εξετάσουμε όλες τις εντολές του αναλυτικά.

Το πρόγραμμα είναι γραμμένο στη γλώσσα **Wiring** που με τη σειρά της βασίζεται στη γλώσσα **C** η οποία είναι από τις πλέον βασικές γλώσσες προγραμματισμού υπολογιστών. Η Wiring προσθέτει κατάλληλες εντολές που σχετίζονται με το Arduino κάνοντας εύκολη τη διαχείριση των λειτουργιών του. Η **C** είναι αρκετά δυνατή γλώσσα αλλά μπορεί να γίνει αρκετά δυσνόητη για ένα αρχάριο. Εμείς θα χρειαστούμε λίγες και σχετικά απλές εντολές για τα παραδείγματα που θα παρουσιάσουμε στις επόμενες ενότητες.

Κάθε πρόγραμμα Arduino χωρίζεται σε δύο ενότητες (συναρτήσεις):

- 1. Ενότητα **setup**. Εκτελείται μόνο μια φορά, στην εκκίνηση του προγράμματος (όταν δηλαδή συνδέσουμε παροχή ρεύματος στο Arduino).
- Ενότητα loop. Εκτελείται συνέχεια όσο τροφοδοτείται με ρεύμα το Arduino. Όταν τελειώσουν οι εντολές που περιέχονται μέσα στην ενότητα loop, εκτελείται ξανά από την αρχή.

Ας δούμε τώρα το δικό μας πρόγραμμα. Στο setup έχουμε:

```
void setup() {
   pinMode(13, OUTPUT);
}
```

Η μόνη γραμμή που περιέχει το setup είναι:

pinMode(13, OUTPUT);

Η γραμμή αυτή καθορίζει ότι ο ακροδέκτης D13:



θα λειτουργεί ως έξοδος.

Στο Arduino πολλοί από τους ακροδέκτες είναι δυνατόν να λειτουργούν τόσο ως είσοδοι όσο και ως έξοδοι. Διαθέτει δώδεκα ακροδέκτες D2 – D13 για αυτό το σκοπό. Όταν λειτουργούν ως είσοδοι, συνδέουμε πάνω τους διακοπτάκια, αισθητήρες κλπ. Ως έξοδοι, δίνουν ρεύμα για να ανάψουν LED, να δώσουν εντολές σε άλλες συσκευές κλπ. Αν θέλαμε ο ακροδέκτης 13 να λειτουργεί ως είσοδος θα γράφαμε:

```
pinMode(13, INPUT);
```

Στην ενότητα Loop έχουμε:

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

Έχοντας καθορίσει τον ακροδέκτη 13 ως έξοδο, μπορούμε τώρα να ενεργοποιήσουμε μια από τις δύο καταστάσεις του:

- Την κατάσταση HIGH όπου ο ακροδέκτης θα βγάζει τάση 5V.
- Την κατάσταση LOW όπου ο ακροδέκτης δεν θα βγάζει τάση!

Με την πρώτη εντολή:

digitalWrite(13, HIGH);

Ο ακροδέκτης βγάζει τάση 5V και το LED ανάβει.

Με την εντολή:

digitalWrite(13, LOW);

Ο ακροδέκτης δεν βγάζει τάση και το LED σβήνει.

Ενδιάμεσα έχουμε την εντολή:

delay(1000);

η οποία προκαλεί καθυστέρηση 1000 χιλιοστά του δευτερολέπτου δηλ. ένα δευτερόλεπτο. Αν δεν την είχαμε, δεν θα προλαβαίναμε να δούμε το LED να αναβοσβήνει! Η ενότητα loop έχει μόνο τέσσερις εντολές, αλλά θυμηθείτε ότι επαναλαμβάνονται από την αρχή μόλις τελειώσουν. Εύκολο δεν είναι;

4.3 Τρεχαντήρι

Μπορούμε τώρα να φτιάξουμε ένα ελαφρά πιο πολύπλοκο κύκλωμα, το οποίο όμως θα χρησιμοποιήσουμε και για όλα τα παραδείγματα της μηχανής Enigma. Για το σκοπό αυτό θα χρειαστούμε πέντε LED και τις αντίστοιχες πέντε αντιστάσεις. Θα συνδέσουμε τα LED στις εξόδους D2 ως D6 του arduino. Η σειρά δεν θα είναι τυχαία: το πρώτο LED (από αριστερά) θα συνδεθεί στο D6, το δεύτερο στο D5 κ.ο.κ. Το κύκλωμα μας θα μοιάζει κάπως έτσι:



Και το πρόγραμμα που θα πρέπει να γράψετε (ή να πάρετε από το φάκελο runninglights) είναι το παρακάτω:

```
void setup() {
  for (int i=2; i<=6; i++)
    pinMode(i, OUTPUT);
}
void loop() {
  delay(50);
  digitalWrite(6, LOW);
  digitalWrite(2, HIGH);
  for (int i=3; i<=6; i++) {
    delay(50);
    digitalWrite(i, HIGH);
    digitalWrite(i-1, LOW);
  }
}</pre>
```

Αν συνδέσετε το κύκλωμα σωστά και στείλετε το παραπάνω πρόγραμμα, θα δείτε τα LEDs να αναβοσβήνουν διαδοχικά και γρήγορα!

Για να δούμε τώρα πως λειτουργεί:

Στην ενότητα loop έχουμε:

```
void setup() {
  for (int i=2; i<=6; i++)
     pinMode(i, OUTPUT);
}</pre>
```

Με αυτό ορίζουμε ότι οι ακροδέκτες D2-D6 θα γίνουν έξοδοι. Είναι εντελώς αντίστοιχο με το να γράφαμε:

```
void setup() {
    pinMode(2, OUTPUT);
    pinMode(3, OUTPUT);
    pinMode(4, OUTPUT);
    pinMode(5, OUTPUT);
    pinMode(6, OUTPUT);
}
```

αλλα είναι πολύ πιο σύντομο! Πως λειτουργεί; Ορίζουμε μια μεταβλητή i η οποία θα παίρνει ακέραιες τιμές από το 2 ως το 6:

for (int i=2; i<=6; i++)</pre>

Η εντολή αυτή δεν τελειώνει εδώ: για κάθε τιμή που παίρνει το i εκτελείται το παρακάτω:

```
pinMode(i, OUTPUT);
```

Με αυτό το τρόπο έχουμε ορίσει αρκετά σύντομα ότι οι ακροδέκτες D2-D6 είναι έξοδοι. Και πάμε στην κύρια ενότητα:

delay(50); digitalWrite(6, LOW); digitalWrite(2, HIGH);

Κάνουμε μια μικρή αρχική καθυστέρηση. Σβήνουμε το αριστερό LED (που είναι συνδεμένο στην D6) και ανάβουμε το δεξιότερο (στην D2). Έπειτα χρησιμοποιούμε το προηγούμενο κόλπο μας για να διατρέξουμε όλα τα υπόλοιπα:

```
for (int i=3; i<=6; i++) {
    delay(50);
    digitalWrite(i, HIGH);
    digitalWrite(i-1, LOW);
}</pre>
```

Η τιμή του i ξεκινάει από το 3 και πάει μέχρι το 6. Αρχικά, έχει την τιμή 3 οπότε είναι σαν να εκτελεί:

```
delay(50);
digitalWrite(3, HIGH);
digitalWrite(2, LOW);
```

Ουσιαστικά δηλ. ανάβει το LED στη D3 και σβήνει το LED στη D2 (που είχαμε ανάψει αρχικά). Έπειτα το i παίρνει τιμή 4, και είναι σαν να εκτελεί:

delay(50); digitalWrite(4, HIGH); digitalWrite(3, LOW);

ανάβει το LED στη D4 και σβήνει το LED στη D3. Αυτό επαναλαμβάνεται μέχρι το i να φτάσει στην τιμή 6 οπότε ο βρόχος τελειώνει. Όμως επειδή όλο αυτό είναι γραμμένο μέσα στην ενότητα loop, εκτελείται ξανά αυτόματα από την αρχή.

4.4 Επικοινωνία με τον Υπολογιστή

Όλα καλά μέχρι εδώ, αλλά τι γίνεται αν θέλουμε εμείς να κάνουμε την κατασκευή μας να επικοινωνεί με τον υπολογιστή που την τροφοδοτεί; Για παράδειγμα, στο Enigma θα θέλαμε να χρησιμοποιήσουμε το πληκτρολόγιο του υπολογιστή μας για να στέλνουμε τα γράμματα στη μηχανή. Αντίστροφα, η μηχανή θα μπορούσε να χρησιμοποιήσει (εκτός από τα LED) και την οθόνη του υπολογιστή μας για να μας στείλει μηνύματα.

Στην πραγματικότητα η διαδικασία αυτή δεν είναι δύσκολη: υπάρχει ήδη κανάλι επικοινωνίας μεταξύ του Arduino και του υπολογιστή, μέσω της USB θύρας. Πως άλλωστε του στέλνουμε το πρόγραμμα κάθε φορά; Μένει μόνο να δούμε πως θα χρησιμοποιήσουμε την επικοινωνία αυτή την ώρα που το πρόγραμμα μας εκτελείται. Η επικοινωνία αυτή είναι γνωστή ως *σειριακή*.

Θα γράψουμε ένα πρόγραμμα στο οποίο θα στέλνουμε στο arduino ένα γράμμα από το πληκτρολόγιο και θα μας το απεικονίζει στα πέντε LEDs που έχουμε συνδέσει ήδη. Βέβαια με πέντε LEDs δεν μπορούμε να φτιάξουμε οθόνη. Μπορούμε όμως να κάνουμε μια απλή αντιστοίχιση γραμμάτων με αριθμούς:

A => 1 B => 2 C => 3 ... Z => 26

Και πάλι όμως, πως θα δείξουμε 26 νούμερα χρησιμοποιώντας 5 LEDs; Απλά θα χρησιμοποιήσουμε δυαδικό σύστημα! Έχουμε να δείξουμε τους αριθμούς από το 1 ως το 26 στο δυαδικό χρησιμοποιώντας 5 LEDs. Με 5 LEDs έχουμε 32 διαφορετικούς συνδυασμούς – μας περισσεύουν και έξι. Ποιοι είναι αυτοί οι συνδυασμοί; Θα πρέπει να μετρήσουμε από το 1 ως το 26 χρησιμοποιώντας πέντε δυαδικά ψηφία. Αν αντιστοιχίσουμε ένα αναμμένο LED στο ψηφίο '1' και ένα σβηστό στο ψηφίο '0' θα έχουμε τον παρακάτω πίνακα:

Γράμμα	Αριθμός	Δυαδικό	LEDs
А	1	00001	00000
В	2	00010	00000
С	3	00011	$\bigcirc \bigcirc $
D	4	00100	00000
E	5	00101	$\bigcirc \bigcirc $
F	6	00110	$\bigcirc \bigcirc $
G	7	00111	$\bigcirc \bigcirc $
Н	8	01000	00000
I	9	01001	$\bigcirc \bigcirc $
J	10	01010	$\bigcirc \bigcirc $
K	11	01011	
L	12	01100	$\bigcirc \bigcirc $
М	13	01101	$\bigcirc \bigcirc $
Ν	14	01110	$\bigcirc \bigcirc $
0	15	01111	
Р	16	10000	00000
Q	17	10001	$\bigcirc \bigcirc $
R	18	10010	$\bigcirc \bigcirc $
S	19	10011	$\bigcirc \bigcirc $
Т	20	10100	$\bigcirc \bigcirc $
U	21	10101	$\bigcirc \bigcirc $
V	22	10110	$\bigcirc \bigcirc $
W	23	10111	
Х	24	11000	$\bigcirc \bigcirc $
Y	25	11001	
Z	26	11010	

Και το πρόγραμμα μας (που μπορείτε επίσης να βρειτε στο φάκελο serial-alphabet) είναι το παρακάτω:

```
void setup() {
 Serial.begin(9600);
 Serial.println("Enigma Ready");
 for (int i=2; i<=6; i++)</pre>
   pinMode(i,OUTPUT);
Serial.print("Enter letter: ");
}
void loop() {
  char c;
  if (Serial.available()) {
    c = Serial.read();
    Serial.println(c);
    Serial.print("Result: ");
    showInt(c);
    Serial.print("Enter letter: ");
  }
}
void showInt(int c)
{
  char s[6];
  int i=0;
  if (c >= 95 && c <= 122)
    c = c - 96;
  else
    c = c - 64;
  if (c >= 1 && c <= 26) {
    for (i = 0; i <= 4; i++)</pre>
      s[i]='0';
```

```
s[5]='\0';
  i=4;
  while (c >= 1) {
    if (c % 2 == 1)
      s[i--] = '1';
    else
      s[i--] = '0';
    c = c / 2;
  }
  for (i = 2; i <= 6; i++)</pre>
    if (s[6-i] == '0')
      digitalWrite(i, LOW);
    else
      digitalWrite(i, HIGH);
  Serial.println(s);
} else
  Serial.println("Error!");
```

Ισως φαίνεται κάπως μεγάλο, αλλά δεν είναι ανάγκη για αρχή να κατανοήσετε κάθε λεπτομέρεια του. Ας το εκτελέσουμε για να δούμε το αποτέλεσμα. Αφού το στείλετε στο Arduino, δεν θα παρατηρήσετε κάποιο LED να ανάβει. Για να δούμε κάτι πρέπει να του στείλουμε ένα γράμμα. Στο περιβάλλον του Arduino, πηγαίνετε στο μενού Tools και επιλέξτε Serial Monitor:



}



Στο κάτω δεξιό μέρος επιλέξτε 'No line ending' και '9600 baud'. Αυτοί είναι οι παράμετροι της επικοινωνίας μας με το Arduino. Στο παράθυρο τώρα θα πρέπει να βλέπετε τα μηνύματα:

Enigma Ready Enter letter:

(Αν δεν έχουν εμφανιστεί με το που αλλάξατε τις ρυθμίσεις, πιέστε enter μια φορά.)

Είμαστε τώρα έτοιμοι να στείλουμε γράμματα στο arduino. Δέχεται μόνο κεφαλαία αγγλικά. Αν γράψετε μικρά αγγλικά θα τα μετατρέψει σε κεφαλαία. Σε οποιοδήποτε άλλο χαρακτήρα θα απαντήσει με 'Error!'. Αν πιέσετε το γράμμα 'E' θα δείτε το τρίτο και το πέμπτο LED να ανάβουν. Συμβουλευτείτε τον πίνακα που δώσαμε προηγουμένως για να βεβαιωθείτε ότι δείχνει τα γράμματα σωστά!

🕺 сом4	
Enigma Ready	
Enter letter:	E
Result: 00101	
Enter letter:	



Πως λειτουργεί το πρόγραμμα; Χωρίς να μπούμε σε πολλές λεπτομέρειες θα εξετάσουμε μερικά τμήματα:

```
void setup() {
  Serial.begin(9600);
  Serial.println("Enigma Ready");
  for (int i=2; i<=6; i++)
    pinMode(i,OUTPUT);
  Serial.print("Enter letter: ");
}</pre>
```

Εδώ κανονίζονται οι παράμετροι της σειριακής επικοινωνίας και ορίζονται οι ακροδέκτες D2-D6 ως έξοδοι, με τον ίδιο τρόπο που κάναμε πριν.

```
void loop() {
  char c;
  if (Serial.available()) {
    c = Serial.read();
    Serial.println(c);
    Serial.print("Result: ");
    showInt(c);
    Serial.print("Enter letter: ");
  }
}
```

Στο σημείο αυτό, το πρόγραμμα μας διαβάζει ένα χαρακτήρα από το πληκτρολόγιο του υπολογιστή και τον αποθηκεύει στη μεταβλητή c. Έπειτα τον στέλνει στη συνάρτηση showInt που έχουμε γράψει παρακάτω και η οποία:

• Ελέγχει αν το γράμμα που δώσαμε είναι κεφαλαίο αγγλικό. Αν είναι μικρο αγ-

γλικό, το μετατρέπει σε κεφαλαίο. Αν είναι οτιδήποτε άλλο τυπώνει 'Error!'.

- Αν το γράμμα είναι έγκυρο, το μετατρέπει σε ένα αριθμό από το 1 ως το 26 (A=1, Z=26).
- Μετατρέπει τον αριθμό στο δυαδικό κάνοντας διαδοχικές διαιρέσεις με το 2 και αποθηκεύοντας κάθε φορά το υπόλοιπο.
- Χρησιμοποιεί τα υπόλοιπα που έχει αποθηκεύσει για να δείξει τις αντίστοιχες τιμές στα LEDs.

Το παραπάνω πρόγραμμα βέβαια δεν είναι κρυπτογράφηση. Είναι όμως μια καλή αρχή:

- Μάθαμε πως μπορούμε να επικοινωνούμε με το arduino και να του στέλνουμε και να διαβάζουμε δεδομένα από αυτό.
- Σκεφτήκαμε ένα τρόπο για να κωδικοποιήσουμε τα γράμματα Α Ζ χρησιμοποιώντας αριθμούς.
- Μετατρέψαμε τους αριθμούς σε δυαδικό σύστημα.
- Είδαμε πως να απεικονίσουμε τον αριθμό χρησιμοποιώντας 5 LEDs.

Όποιος ενδιαφέρεται για τον ακριβή τρόπο που γίνονται τα παραπάνω, θα μπορέσει να με ρωτήσει στις συναντήσεις.

Ουσιαστικά δηλ. έχουμε έτοιμο το κομμάτι που διαβάζει τα δεδομένα, τα μετατρέπει σε αριθμούς και τα απεικονίζει. Μένει να κάνουμε την πραγματική κρυπτογράφηση. Αυτό προφανώς είναι το πιο σύνθετο κομμάτι, ωστόσο με την ολοκλήρωση του θα μπορούμε να μεταφέρουμε τον κώδικα σε μια πιο σύνθετη μηχανή που θα έχει οθόνη, κινητήρες για τους ρότορες και πολλά περισσότερα από τα αγαπημένα μας LEDs. Και θα θέλει σχετικά λίγες αλλαγές για να λειτουργήσει εκεί.

4.5 Nano Enigma - Πρώτη Εκδοχή

Όπως κάθε πολύπλοκο πρόγραμμα (ή πρόβλημα), θα πρέπει να αντιμετωπίσουμε την Enigma τμηματικά. Δεν θα προσπαθήσουμε να φτιάξουμε κατευθείαν μια μηχανή με όλη τη πολυπλοκότητα της κανονικής Enigma, αλλά θα προσπαθήσουμε αρχικά μια ιδιαίτερα απλοποιημένα εκδοχή. Ποια είναι όμως η πιο απλή Enigma που μπορούμε να φτιάξουμε; Φανταστείτε ότι αφαιρούμε από την κανονική Enigma τα πάντα και αφήνουμε μόνο το πληκτρολόγιο, τον ένα ρότορα (τον δεξιότερο, αυτόν που συνδέεται άμεσα με το πληκτρολόγιο) και τον ανακλαστήρα. Και φυσικά τα λαμπάκια για να βλέπουμε τι βγαίνει!

Στη μηχανή αυτή, όταν πιέζουμε ένα πλήκτρο ο ρότορας αρχικά περιστρέφεται μια θέση και έπειτα το ηλεκτρικό ρεύμα από το πληκτρολόγιο συνδέεται στην αντίστοιχη δεξιά θέση του ρότορα. Το σήμα αυτό 'κρυπτογραφείται' από τον ρότορα και βγαίνει σε μια άλλη θέση από την αριστερή του πλευρά.

Για παράδειγμα: Έστω ότι ο δεξιός μας ρότορας είναι τύπου ΙΙΙ. Ο ρότορας αυτός έχει την παρακάτω αντιστοίχιση επαφών:

1	2	3	4	5	6	7	8	9	10) 11	12	2 13	3 14	11	5 16	5 17	7 18	5 19	20) 21	. 22	23	24	- 25	5 26
A	В	С	D	Ε	F	G	Η	Ι	J	Κ	L	Μ	Ν	Ο	Р	Q	R	S	Т	U	V	W	Х	Y	Ζ
В	D	F	Η	J	L	С	Р	R	Т	Χ	V	Ζ	Ν	Y	E	Ι	W	G	А	Κ	Μ	U	S	Q	0

Ας υποθέσουμε ότι ο ρότορας βρίσκεται στην θέση Α πριν πιέσουμε οποιοδήποτε πλήκτρο. Σε αυτή τη θέση αντιστοιχεί η επαφή 1. Και έστω ότι πιέζουμε το πλήκτρο Α. Ο ρότορας έχει 26 επαφές δεξιά σε κυκλική μορφή και εφάπτεται με τις αντίστοιχες 26 του πληκτρολογίου. Το Α αντιστοιχεί στην πρώτη επαφή του πληκτρολογίου. Αλλά θυμηθείτε ότι ο ρότορας πρώτα περιστρέφεται και μετά κρυπτογραφεί. Θα έχουμε το εξής αποτέλεσμα:

- Ο χειριστής πιέζει το γράμμα 'Α' στο πληκτρολόγιο.
- Ο ρότορας περιστρέφεται κατά μια θέση. Τώρα αντιστοιχεί το γράμμα Β στην πρώτη επαφή.
- Το Α είναι το πρώτο γράμμα, έτσι το ρεύμα από το πληκτρολόγιο μεταφέρεται στην πρώτη επαφή του πληκτρολογίου με το ρότορα (στο B).
- Το Β κρυπτογραφείται στο γράμμα D σύμφωνα με τον παραπάνω πίνακα.
- Το D κανονικά αντιστοιχεί στην επαφή 4, όμως ο ρότορας έχει περιστραφεί μια θέση και έτσι βρίσκεται τώρα στην επαφή 3.

Πάμε στο κομμάτι του ανακλαστήρα. Αυτή τη στιγμή το ρεύμα φεύγει από την επαφή 3 του ρότορα και εισέρχεται στην επαφή 3 του ανακλαστήρα. Αν δούμε τον αντίστοιχο πίνακα του ανακλαστήρα, θα ανακαλύψουμε ότι το σήμα από την επαφή 3 βγαίνει στην επαφή 21. Το σήμα αυτό θα ξαναμπεί στην αριστερή επαφή 21 του ρότορα. Στη θέση 21 κανονικά θα είχαμε το γράμμα U (το 21ο γράμμα του λατινικού αλφαβήτου), αλλά επειδή ο ρότορας έχει μετακινηθεί μια θέση, έχουμε το V. Το γράμμα V αντιστοιχίζεται στην δεξιά μεριά στο γράμμα L που βρίσκεται κανονικά στη θέση 12. Όμως, καθώς ο ρότορας έχει περιστραφεί κατά μια θέση, είναι τώρα στη θέση 11. Η επαφή 11 συνδέεται με το λαμπάκι αριθμός 11 και το ανάβει. Το 11 αντιστοιχεί στο γράμμα K.



Σίγουρα είναι λίγο μπέρδεμα να το ακολουθήσει κάποιος αυτό με το χέρι. Γι'αυτό πιάστε το paper Enigma που σας έδωσα και δείτε το.

Ας κάνουμε ένα ακόμα παράδειγμα. Συνεχίζοντας από το προηγούμενο, χωρίς να αλλάξουμε ρυθμίσεις, ας πιέσουμε τώρα το F. Το F αντιστοιχεί στην 6η επαφή του πληκτρολογίου.

- Ο χειριστής πιέζει το 'F' στο πληκτρολόγιο.
- Ο ρότορας περιστρέφεται κατά μια θέση. Τώρα αντιστοιχεί το C στην πρώτη επαφή.
- Το F είναι το 6ο γράμμα, το ρεύμα μεταφέρεται από την 6η επαφή του πληκτρολογίου στην 6 επαφή του ρότορα.
- Κανονικά στην 6η επαφή του ρότορα θα είχαμε το F, όμως ο ρότορας έχει κάνει ήδη δύο βήματα άρα στη θέση 6 βρίσκεται το H.
- Το Η κρυπτογραφείται στο Ρ σύμφωνα με τον παραπάνω πίνακα. Το Ρ είναι κανονικά στην επαφή 16, όμως επειδή έχουμε κάνει δύο βήματα είναι τώρα στη 14.
- Το ρεύμα εξέρχεται από την επαφή 14 του ρότορα και εισέρχεται στην 14 του ανακλαστήρα.

Στο κομμάτι του ανακλαστήρα, έχουμε:

- Η επαφή 14 του ανακλαστήρα συνδέεται με την 11.
- Το ρεύμα εισέρχεται στην αριστερη επαφή 11 του ρότορα.
- Στην 11 κανονικά είναι το K, όμως επειδή έχουμε μετακινηθεί δυο θέσεις έχουμε πλέον το M.
- Το Μ κρυπτογραφείται στο V σύμφωνα με τον πίνακα, το οποίο κανονικά είναι στην θέση 22. Τώρα όμως μετά από δύο βήματα είναι στην 20.
- Η επαφή 20 ανάβει το λαμπάκι 20 που αντιστοιχεί στο Τ.



Προφανώς το να προσπαθούμε να κρυπτογραφήσουμε με το χέρι στην Enigma είναι χειρότερο και από την Enigma των Γερμανών :D

Τα παραπάνω τώρα μπορείτε να τα βρείτε ως πρόγραμμα στο φάκελο enigma-nano1. Φορτώστε το στο Arduino και ανοίξτε το serial monitor όπως κάνατε και στο προηγούμενο πρόγραμμα της επικοινωνίας. Θα δείτε το ίδιο μήνυμα:

Enigma Ready Enter letter:

Όμως όταν γράψετε ένα γράμμα, θα το κρυπτογραφήσει σύμφωνα με την παραπάνω διαδικασία και θα δείτε το αποτέλεσμα τόσο στην οθόνη σας όσο και στα LED με τη μορφή 0/1. Επιβεβαιώστε τα αποτελέσματα που είδαμε πριν:

👳 СОМ4
I
Enigma Ready
Enter letter: A
Results:
Binary: 01011
Letter: K
Enter letter: F
Results:
Binary: 10100
Letter: T
Enter letter:

Έχουμε μια πρώτη πολύ απλή κρυπτογράφηση, με ένα και μοναδικό ρότορα. Στην επόμενη μας συνάντηση, θα φτιάξουμε μια πλήρη Enigma με τρεις ρότορες και με όλα τα περίφημα ελαττώματα της αρχικής μηχανής! Μέχρι τότε μπορείτε να πειραματιστείτε όσο θέλετε με αυτή – και φυσικά να συμπληρώσετε τις ασκήσεις που θα σας δώσω!