

Διδασκαλία των εννοιών «Εισόδου/Εξόδου» (I/O) προγραμματίζοντας τον Arduino σε Wiring C και σε εναλλακτικές γλώσσες προγραμματισμού, S4A, C# και Python

Γ. Γιάννακας¹, Ι. Δήμος²

¹Δρ. Φυσικός Ρ/Η (Μ.Σc.), Ph.D. Η.Μ.Μ.Υ. Π.Θ., Υπ. Ε.Κ.Φ.Ε. Φθ/δας
ggiannakas@gmail.com

²Πληροφορικός (ΠΕ19), Μ.Σc in Engineering, Υπ. ΚΕ.ΠΑΗ.ΝΕ.Τ. Φθ/δας
idimos@sch.gr

Περίληψη

Στην παρούσα εργασία, παρουσιάζονται τα χαρακτηριστικά και οι δυνατότητες του Arduino ως ένα χαμηλού κόστους αξιόπιστο σύστημα διδασκαλίας Προγραμματισμού στο Σχολικό Εργαστήριο Πληροφορικής και Εφαρμογών Η/Υ (ΣΕΠΕΗΥ), σε συνδυασμό με την απαραίτητη, σήμερα στην εποχή του IoT (Internet of Things), ανάγκη αλληλεπίδρασης με πραγματικά υπολογιστικά συστήματα. Πιο συγκεκριμένα, η εργασία εστιάζει την προσοχή της στις έννοιες Είσοδο (I) και Έξοδο (O), ώστε οι μαθητές να διακρίνουν τις δύο έννοιες (I/O) και να κατανοήσουν τις δυνατότητες των αισθητήρων, ενεργοποιητών και σερβομηχανισμών, που μπορούν να αποτελέσουν εισόδους ή/και εξόδους σε ένα σύστημα επεξεργασίας και διαχείρισης δεδομένων όπως ο Arduino, με τα κατάλληλα ηλεκτρονικά κυκλώματά του. Στο πλαίσιο αυτό, γίνεται αναφορά στη μορφή του κώδικα Wiring C (IDE) που χρησιμοποιεί ο Arduino καθώς και εναλλακτικών μεθόδων προγραμματισμού, όπως με τη γλώσσα S4A (Scratch for Arduino), C# (C Sharp) & Python, εφαρμοσμένων σε ένα πραγματικό σενάριο.

Λέξεις κλειδιά: arduino, S4A, C#, python, ΣΕΠΕΗΥ, IoT, STEM

1. Εισαγωγή

Το σενάριο σχεδιάστηκε με σκοπό τη χρήση και αξιοποίηση των Τεχνολογιών της Πληροφορίας και των Επικοινωνιών (ΤΠΕ) στην εκπαιδευτική διαδικασία. Η συγκεκριμένη εργασία, αποτελεί μια διδακτική παρέμβαση που στόχο έχει να βοηθήσει μαθητές ή αρχάριους προγραμματιστές να βελτιώσουν τις επιδόσεις τους στον προγραμματισμό, ενώ μέσω της ανακαλυπτικής μεθόδου διδασκαλίας και της καθοδηγούμενης διερευνητικής μάθησης, με έμφαση στη σχεδίαση και με σαφή εμπλοκή των μαθητών στην εκπαιδευτική διαδικασία, μπορεί εύκολα να προσεγγίσει και τις απαιτήσεις της νέας STEM (Science, Technology, Engineering, Mathematics) εκπαίδευσης. Οι εργασίες, με τη χρήση της Scratch (S4A), της C#, της Python και του

Arduino IDE, στοχεύουν στην εξοικείωση των μαθητών με βασικές εντολές προγραμματισμού εφαρμοσμένων σε ένα πραγματικό σενάριο, με διάφορους τύπους εισόδων (PWM, Button, Distance Sensor) και εξόδων (Led, Serial Output) στον Arduino. Ο μικροεπεξεργαστής Arduino επιλέχθηκε για την ευκολία του στη χρήση, για τις δυνατότητες του, την χαμηλή τιμή του και για το γεγονός ότι ο προγραμματισμός πραγματοποιείται εναλλακτικά με τη βοήθεια πολλών διαφορετικών γλωσσών προγραμματισμού. Κύρια εστιάζει την προσοχή της στις έννοιες Είσοδο (I) και Έξοδο (O), ώστε οι μαθητές να διακρίνουν τις δύο έννοιες (I/O) και να κατανοήσουν τις δυνατότητες των αισθητήρων, ενεργοποιητών και σερβομηχανισμών, που μπορούν να αποτελέσουν εισόδους ή/και εξόδους σε ένα σύστημα επεξεργασίας και διαχείρισης δεδομένων όπως ο Arduino, με τα κατάλληλα ηλεκτρονικά κυκλώματά του.

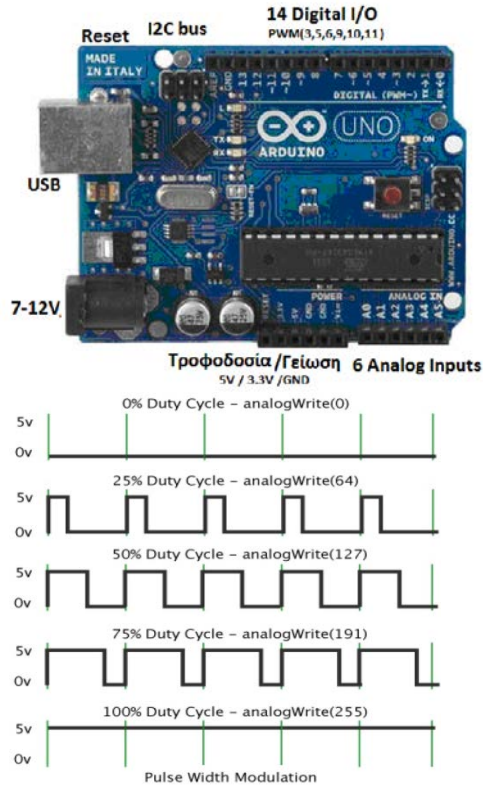
Είσοδος/Έξοδος (I/O) θεωρείται στην Πληροφορική, η επικοινωνία μεταξύ ενός πληροφοριακού συστήματος επεξεργασίας (όπως ένας υπολογιστής) και του εξωτερικού κόσμου, όπως πιθανά ένας άνθρωπος ή ένα άλλο πληροφοριακό σύστημα επεξεργασίας. Είσοδοι (Inputs) αποκαλούνται τα σήματα ή τα δεδομένα που εισέρχονται στο πληροφοριακό σύστημα, ενώ Έξοδοι (Outputs) είναι τα σήματα ή τα δεδομένα που εξέρχονται από το πληροφοριακό σύστημα.

2. Ο MCU Arduino & ο Προγραμματισμός του

2.1 Arduino

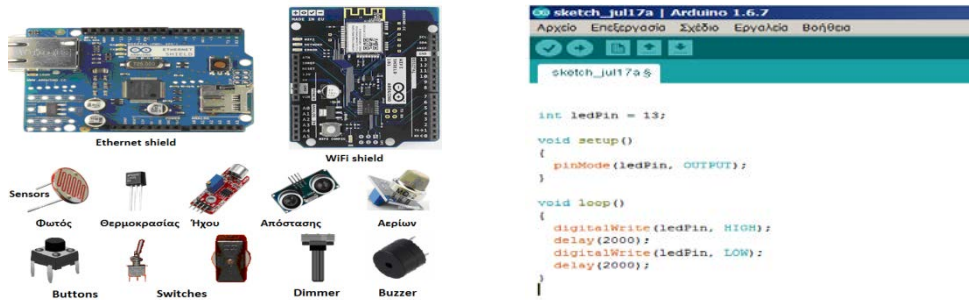
Ο Arduino, είναι μια «ανοικτού κώδικα» (αναφέρεται στο λογισμικό του μικροελεγκτή MCU ATmega-AVR/ATMEL) πλατφόρμα ηλεκτρονικών πολύ χαμηλού κόστους, βασισμένη σε ευέλικτο και εύκολο στη χρήση hardware και software, που προορίζεται για οποιονδήποτε έχει λίγη προγραμματιστική εμπειρία, στοιχειώδεις γνώσεις ηλεκτρονικών και ενδιαφέρεται να δημιουργήσει διαδραστικά αντικείμενα ή περιβάλλοντα. Έχει τη δυνατότητα αλληλεπίδρασης (physical computing) με το φυσικό κόσμο μέσω αισθητήρων, για την καταγραφή, επεξεργασία και απεικόνιση δεδομένων διάφορων φυσικών μεγεθών όπως είναι η θερμοκρασία, η πίεση, η απόσταση, η ένταση ρεύματος ή φωτός κ.α.. Έτσι, μπορεί να αποτελέσει ένα ελκυστικό εργαλείο για την προώθηση και της STEM (Science Technology Engineering Mathematics) μεθοδολογίας, προωθούμενης τελευταία για τη διδασκαλία των θετικών επιστημών (Zieris et al, 2014). Αποτελεί μία λύση (όχι τη μόνη) για τη δημιουργία διαδραστικών ηλεκτρονικών συσκευών για κάθε είδους εφαρμογή.

Ειδικότερα, ο Arduino (in USA, Genuino εκτός USA), υποστηρίζει σειριακή επικοινωνία με τον H/Y μέσω θύρας USB (COM#).



Εικόνα 1. α) MCU Arduino Uno, β) PWM (ΨευδοΑναλογικές έξοδοι)

Επικοινωνεί με τον έξω κόσμο (αισθητήρες, ενεργοποιητές κλπ) μέσω αναλογικών σειριακών θυρών εισόδου - αμιγείς αναλογικές έξοδοι δεν υπάρχουν-, (6 στο Arduino Uno, A0-A5, Εικόνα 1α) που χρησιμοποιούν την ADC (Αναλογικο-Ψηφιακή) μετατροπή της πλακέτας (ανάλυση 10bit ή 1024 τιμών, από 0 όταν η τάση εισόδου είναι 0 Volts, μέχρι 1023 όταν η τάση εισόδου είναι 5 Volts, για αναλογικούς αισθητήρες), καθώς και μέσω (14 για το Arduino Uno, D0-D13) ψηφιακών (High/Low) εισόδων/εξόδων (Digital I/O) που μπορούν όμως να έχουν και άλλες λειτουργίες (π.χ. τα 3,5,6,9,10,11 λειτουργούν και ως ψευδο-αναλογικές έξοδοι μέσω της διαμόρφωσης εύρους παλμών (PWM, Εικόνα 1β) σε 256 (από 0-255) στάθμες τάσης (ανάλυση 8bit), όπως για παράδειγμα όταν ελέγχουμε τη φωτεινότητα ενός Led συνδεδεμένου π.χ. στο pin 10, όπου η μέση τιμή 127 θα φωτίζει κατά το ήμισυ τη φωτοдиодο, όχι επειδή θα λαμβάνει τη μισή τάση (2.5V) αλλά επειδή θα λαμβάνει την αρχική τάση (5V) με ένα duty cycle 50% (ίσους δηλ. χρόνους εναλλαγής ON/OFF).



Εικόνα 2. α) Arduino Shields & Sensors, β) Arduino IDE (Led's On/Off)

Τέλος, καθένα από τα 6 αναλογικά pin, με κατάλληλη εντολή μέσα από το πρόγραμμα μπορεί να μετατραπεί σε ψηφιακό pin εισόδου/εξόδου όπως τα 14 που βρίσκονται στην απέναντι πλευρά και τότε μετονομάζονται από A0~5 σε D14~19 αντίστοιχα.

Οι αισθητήρες (ψηφιακοί ή αναλογικοί, σε ολοκληρωμένο κύκλωμα) συνδέονται στις ψηφιακές ή/και αναλογικές εισόδους του Arduino, που κάνοντας χρήση των κατάλληλων βιβλιοθηκών, διαβάζει τις τιμές τάσης στην είσοδό του και τις μετατρέπει σε κατάλληλες και αναγνωρίσιμες τιμές των φυσικών μεγεθών που μετρά. Επίσης υπάρχουν και έτοιμες πλακέτες και κυκλώματα ηλεκτρονικών ως επεκτάσεις, (shields, Εικόνα 2α) χρήσιμα ειδικά σε όσους δεν έχουν ένα στοιχειώδες υπόβαθρο περί ηλεκτρονικής, με στόχο την επέκταση των δυνατοτήτων και της λειτουργικότητας του Arduino, με εύχρηστο και γρήγορο τρόπο (WiFi shield, Ethernet Shield, GPS shield, servo-motor shields, etc.).

2.2 Arduino IDE (Wiring C)

Για την επικοινωνία, διαχείριση και τον προγραμματισμό του Arduino, υπάρχει το Arduino IDE (INTEGRATED DEVELOPMENT ENVIRONMENT) (Arduino/Genuino, 2016) που παρέχει μέσω της Java, ένα περιβάλλον προγραμματισμού (Εικόνα 2β) σε wiring C, βιβλιοθήκες (lib) και παραδείγματα, μεταγλωττιστή (AVR gcc compiler) και τη δυνατότητα να φορτωθεί το μεταγλωττισμένο πρόγραμμα στο Arduino. Η γλώσσα προγραμματισμού βασίζεται στη C, υπάρχουν όμως και κάποιες ειδικές εντολές, συναρτήσεις και σταθερές που βοηθούν στη διαχείριση του ειδικού hardware του Arduino (pinMode, DigitalRead, AnalogRead, AnalogWrite, DigitalWrite, millis, delay, Serial.begin, Serial.println κλπ.). Το κάθε πρόγραμμα σε C, αποτελείται από 2 βασικές ρουτίνες, την setup() που εκτελείται μια μόνο φορά και την loop() που αποτελεί τον κορμό του προγράμματος και εκτελείται συνέχεια (ή όσο ισχύει κάποια συνθήκη). Αρχικά βέβαια, θα πρέπει να έχουν ενσωματωθεί οι απαιτούμενες βιβλιοθήκες και να έχουν οριστεί όλες οι προς χρήση μεταβλητές και σταθερές. Ότι ακολουθεί τα //, είναι σχόλιο και δεν εκτελείται (Πουλάκης, 2015).

Έτσι, προτείνεται η χρήση του μικροελεγκτή (MCU) Arduino και των αισθητήρων που τον συνοδεύουν, προγραμματιζόμενοι όμως τόσο με την εύχρηστη και φιλική γλώσσα S4A (Διαμαντόπουλος κ.α., 2015), (Ορφανάκης & Παπαδάκης, 2014) αποφεύγοντας έτσι τη δυσκολία που ενέχει (ειδικά για μαθητές) η χρήση της γλώσσας wiring C του Arduino IDE, όσο και με τις γλώσσες C# και Python ως εφαρμογές γλωσσών υψηλού επιπέδου για πιο έμπειρους χρήστες.

2.3 S4A

S4A (S4A, MIT) είναι μια συμβατή τροποποίηση του γνωστού Scratch (<https://scratch.mit.edu/>) που επιτρέπει τον απλό προγραμματισμό της ανοιχτής πλατφόρμας Arduino παρέχοντας και νέα block εντολών για τη διαχείριση αισθητήρων και άλλων διατάξεων που συνδέονται στο Arduino, όπως μοτέρ, servo κλπ.. Ο βασικός λόγος δημιουργίας της S4A είναι να προσελκύσει ανθρώπους στον κόσμο του προγραμματισμού (Rosenbaum et.al., 2010).

Η γλώσσα S4A έχει ελεγχθεί ώστε να λειτουργεί ορθά με Arduino Uno, Diecimila & Duemilanove. Η διασύνδεση γίνεται αυτόματα μέσω θύρας USB, αφού βέβαια εγκατασταθεί αρχικά το κατάλληλο firmware μέσω της κανονικής διεπαφής του περιβάλλοντος Arduino IDE. Η S4A επιτρέπει και ασύρματη διασύνδεση, ενώ διατίθεται έκδοση τόσο για Windows, όσο και για Linux & Mac.

2.4 C# (C Sharp)

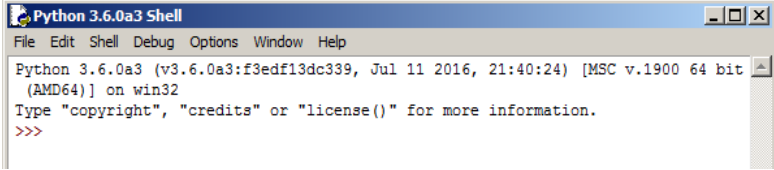
Η C# (C Sharp) (Hejlsberg, 2011) είναι γλώσσα προγραμματισμού H/Y που δημιουργήθηκε από την Microsoft μέσα από την πλατφόρμα .NET και αργότερα αναγνωρίστηκε επισήμως από την Ecma (ECMA-334, 2006) και την ISO (ISO/IEC 2327:2006). Είναι μια από τις γλώσσες προγραμματισμού που δημιουργήθηκαν για την Common Language Infrastructure. Η C# (Hejlsberg, 2011) σχεδιάστηκε κυρίως ως μια απλή αντικειμενοστραφής γλώσσα γενικής χρήσης. Στις 15/8/2012 κυκλοφόρησε η έκδοση 5.0 η οποία είναι η πιο πρόσφατη μέχρι σήμερα.

2.5 Python

Η Python (Python) είναι μια εύκολη στην εκμάθηση, ισχυρή γλώσσα προγραμματισμού υψηλού επιπέδου. Έχει αποδοτικές δομές δεδομένων και μια απλή αλλά αποτελεσματική προσέγγιση στον αντικειμενοστραφή προγραμματισμό. Η κομμητή σύνταξη της Python (Python) και οι δυναμικοί τύποι της, μαζί με τη λειτουργία της ως διερμηνεύσιμη (αντί μεταγλωττιζόμενης όπως η C/C++) γλώσσας, την καθιστούν ιδανική γλώσσα για δημιουργία σεναρίων εντολών και για ταχεία ανάπτυξη εφαρμογών σε πολλούς τομείς (ως ΕΛΛΑΚ) και στις περισσότερες πλατφόρμες.

Εγκατάσταση Python για Arduino (Mcwhorte, 2014): Για να κάνουμε ένα Arduino να επικοινωνεί με τη γλώσσα Python, θα πρέπει να εγκαταστήσουμε μια σειρά απαραίτητων λογισμικών. Τα βήματα και τα λογισμικά φαίνονται παρακάτω:

Κατεβάστε την **python 3.6** από την παρακάτω διεύθυνση, ανάλογα με την έκδοση του λειτουργικού σας συστήματος (<https://www.python.org/downloads/windows/>). Μετά την επιτυχή εγκατάσταση, θα εμφανιστεί στην επιφάνεια εργασίας το σχετικό εικονίδιο το οποίο αφού το τρέξουμε θα εμφανιστεί το terminal της python:



```
Python 3.6.0a3 Shell
File Edit Shell Debug Options Window Help
Python 3.6.0a3 (v3.6.0a3:f3edf13dc339, Jul 11 2016, 21:40:24) [MSC v.1900 64 bit
(AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
```

Εγκαθιστούμε τους drivers pyserial (Liechti, 2001-2016) για την σειριακή επικοινωνία της γλώσσας με το Arduino, από (<https://pypi.python.org/pypi/pyserial, 2016>).

3. Σενάριο Διδασκαλίας εννοιών I/O (Εισόδου/Εξόδου)

3.1 Οι βασικές συναρτήσεις ορισμού I/O των ασκήσεων του σεναρίου

Από τις πιο βασικές λειτουργίες του μικροελεγκτή είναι ο έλεγχος των θυρών που διαθέτει ώστε, είτε να δίνει ρεύμα είτε να παίρνει ρεύμα από αυτές. Στην αρχικοποίηση κάθε προγράμματος (setup) χαρακτηρίζουμε τα Pins που χρησιμοποιούμε ως είσοδο (I) ή ως έξοδο (O). Η συνάρτηση `pinMode(Pin, Mode)` χρησιμοποιείται με το όνομά της και ορίσματα α) τον αριθμό Pin και β) την κατάσταση (Mode) λειτουργίας που χαρακτηρίζεται με τη λέξη INPUT (είσοδος) ή OUTPUT (έξοδος), π.χ., `pinMode(12, OUTPUT)`, `pinMode(ledPin, OUTPUT)`. Για να μπορέσουμε να δώσουμε ρεύμα προς τα έξω μέσω μιας θύρας (pin) θα πρέπει πρώτα να έχει αυτή οριστεί ως θύρα εξόδου. Αντίστοιχα, για να «διαβάσουμε» από μια είσοδο, θα πρέπει αρχικά να την ορίσουμε ως είσοδο. Και τα 14 pins του Arduino μπορούν δουλεύουν ως ψηφιακές εξόδου, δηλαδή δίνουν έξοδο 0 ή 5V. Αυτό γίνεται με χρήση της συνάρτησης `digitalWrite(Pin, Value)`, όπου το όρισμα Pin αναφέρεται στο νούμερο της θύρας για την οποία θα δώσουμε τάση εξόδου, ενώ η τάση εξόδου μπορεί να είναι 0 V ή 5 V, οι οποίες αναπαρίστανται με προκαθορισμένες τιμές στην παράμετρο value (LOW: θα δώσει 0 V στην έξοδο (pin), HIGH: θα δώσει 5 V στην έξοδο (pin)). Και τα 14 ψηφιακά pins του Arduino μπορούν να δουλεύουν και ως ψηφιακές εισόδου, δηλαδή να “διαβάσουν” ως είσοδο τάση με τιμή είτε 0 είτε 5V. Αυτό γίνεται με χρήση της συνάρτησης `digitalRead(Pin)`, όπου το όρισμα Pin αναφέρεται στο νούμερο της θύρας για την οποία θα πάρουμε είσοδο, ενώ η συνάρτηση επιστρέφει με το όνομά της την τιμή εισόδου. Η τάση εισόδου μπορεί να είναι 0V ή 5V, οι οποίες αναπαρίστανται με προκαθορισμένες τιμές στην τιμή που διαβάζουμε (LOW: όταν λάβει τάση 0 V στην είσοδο (pin), HIGH: όταν λάβει τάση 5 V στην είσοδο (pin)). Κάποια από τα 14 Pins

του Arduino που έχουν την ένδειξη PWM (pseudo-Analog Outputs), μπορούν όπως αναφέραμε, να προσομοιώσουν την αναλογική έξοδο μέσω διαμόρφωσης εύρους (διάρκειας) παλμών. Έτσι, με τιμές από το 0 μέχρι το 255 (8bits), προσομοιώνουμε (αναλογικά) το διάστημα από 0 έως 5V. Αυτό γίνεται με χρήση της συνάρτησης `analogWrite(Pin, Value)`, όπου το όρισμα `Pin` αναφέρεται στο νούμερο της θύρας για την οποία θα δώσουμε ρεύμα εξόδου, ενώ η τάση εξόδου κυμαίνεται από 0 V μέχρι και 5 V, οι οποίες τιμές της τάσης αναλογικά αναπαρίστανται με τιμές στη μεταβλητή `Value`. Το Arduino έχει και 6 αναλογικές εισόδους, οι οποίες χαρακτηρίζονται με τα σύμβολα `A0`, `A1`, `A2`, `A3`, `A4`, `A5`. Μπορούμε να συνδέσουμε κάποιο αναλογικό εξάρτημα (π.χ. ποτενσιόμετρο) και να το διαβάσουμε ως είσοδο. Αυτό γίνεται με χρήση της συνάρτησης `analogRead(Pin)`, όπου το όρισμα `Pin` αναφέρεται στο νούμερο της θύρας για την οποία θα πάρουμε είσοδο, ενώ η συνάρτηση επιστρέφει με το όνομά της την τιμή εισόδου. Η τιμή εισόδου κυμαίνεται από 0 μέχρι και 1023 (10bits ADC) (Πουλάκης, 2015).

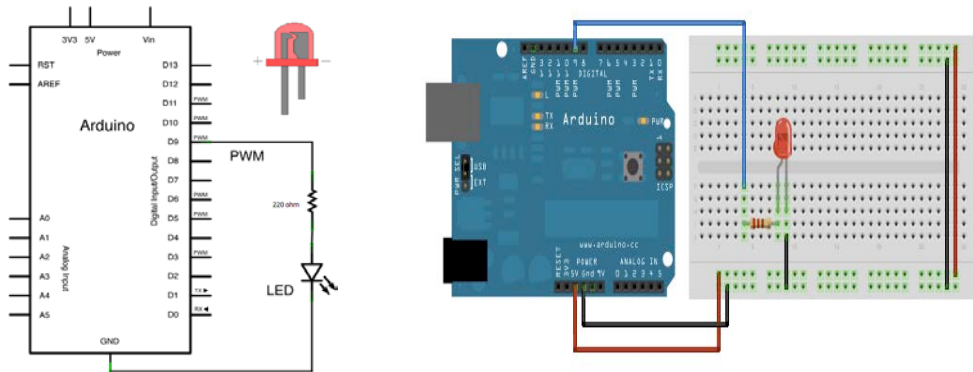
Ακολούθως θα εφαρμόσουμε τις 4 μεθόδους προγραμματισμού (IDE wiring-C, S4a, C#, Python) του Arduino (Uno), σε ένα πραγματικό παράδειγμα ψηφιακής εξόδου οδηγούμενης από ψευδοαναλογική είσοδο (PWM).

3.2 Κυκλωμάτωση και Προγραμματισμός σε 4 περιβάλλοντα προγραμματισμού (IDE, S4A, C#, Python) σεναρίου I/O

Υλοποιήσαμε το ακόλουθο εργαστηριακό παράδειγμα, κάνοντας χρήση βασικών ηλεκτρονικών διατάξεων, με 4 γλώσσες προγραμματισμού (IDE wiring-C, C4A, C#, Python):

Analog Output (Pseudoanalog Input / PWM, control lightning (fading) of LED)

Χρησιμοποιήσαμε την ψευδοαναλογική έξοδο 9 (D9/PWM) ώστε να τροφοδοτήσουμε μια Led, η φωτεινότητα της οποίας λαμβάνει τιμές από 0-5V, ανάλογα με την τιμή εξόδου PWM που λαμβάνει τιμές από 0-255 (256 τιμές). Έτσι, αν η έξοδος 9 δίνει τιμή 127, τότε η led είναι μισοαναμμένη στα 2.5V. Όπως πάντοτε συμβαίνει, έχουμε θέσει μια αντίσταση των 220Ω στο κύκλωμα της φωτοδίοδου, ώστε να την προστατεύσουμε από υπέρταση. Η τιμή αυτή εύκολα προκύπτει, αν σκεφτούμε ότι τροφοδοτούμε με 5V μια φωτοδίοδο που μπορεί να δεχθεί μέχρι 3V στα άκρα της, ενώ αντέχει ένταση ρεύματος μέχρι 10mA. Οπότε $R = (V_{usb} - V_{led}) / I = (5-3)/0.01 = 200\Omega$. Προσοχή απαιτείται και για την πόλωση με ορθή πολικότητα, της Led. Ανάστροφη πόλωσή της, την αφήνει σβηστή, εάν δεν την καταστρέψει. Ακολουθεί το κύκλωμα της άσκησης (**Εικόνα 3**), καθώς και ο προγραμματισμός, τόσο στη γλώσσα Arduino IDE όσο και σε S4A (Scratch) στο Παράρτημα.



Εικόνα 3. Analog Output (fading of a LED by a PWM pseudoanalog input)

Σε αρκετές περιπτώσεις προγραμματισμού Arduino, είναι σύνηθες από έμπειρους προγραμματιστές να χρησιμοποιείται η προσέγγιση γλωσσών προγραμματισμού υψηλού επιπέδου όπως οι C# και Python, με τον Arduino. Ο λόγος, έχει να κάνει με τις δυνατότητες που παρέχονται από τις γλώσσες αυτές, όπως η χρήση του αντικειμενοστραφούς προγραμματισμού (κληρονομικότητα, πολυμορφικότητα και ενθυλάκωση), όπως επίσης και η χρήση προχωρημένων βιβλιοθηκών για web εφαρμογές.

Στο τρέχον παράδειγμα, μεταφέρουμε το κομμάτι της λογικής επεξεργασίας τόσο στη C# όσο και στην Python και προγραμματίζουμε το Arduino (με wiring C) απλώς να «ακούει» τη σειριακή πόρτα και ανάλογα με την τιμή που θα διαβάζει θα ενεργοποιεί/απενεργοποιεί την τάση σε οποιοδήποτε από τις αναλογικές/ψηφιακές εισόδους/εξόδους του Arduino UNO. Στο παράδειγμα μας, ορίζουμε την τάση του led (Αναλογική έξοδος 9) από τη C# ή την Python και όχι από τη wiring C. Η υλοποίηση με τους τρόπους αυτούς, ακολουθεί στο Παράρτημα.

Δυστυχώς, ο περιορισμένος χώρος του άρθρου δεν επέτρεψε την αναφορά και σε άλλα σενάρια υλοποίησης, κάνοντας χρήση διάφορων αισθητήρων και σερβομηχανισμών, με στόχο να διαφανεί εκτενέστερα η χρήση, τόσο Αναλογικών όσο και Ψηφιακών Εισόδων και Εξόδων.

4. Συμπέρασμα

Οι μαθητές μαθαίνουν πώς να κατασκευάζουν και να προγραμματίζουν (με 4 τρόπους, επιλέγοντας όποιον τους βολεύει) διαδραστικά συστήματα, χρησιμοποιώντας αισθητήρες και δομικά ηλεκτρονικά στοιχεία. Το όφελος έγκειται τόσο στην εκμάθηση των βασικών αρχών των Φυσικών Επιστημών, της Τεχνολογίας, της Εφαρμοσμένης Μηχανικής και των Μαθηματικών, αρχές που συνοψίζονται στο ακρωνύμιο STEM (Science, Technology, Engineering, Mathematics), όσο και στη γνωριμία τους με τον κόσμο των διαδραστικών κατασκευών, κατασκευάζοντας μοντέλα που αντι-

στοιχούν στον φυσικό κόσμο, ενώ ταυτόχρονα εισάγονται στις αρχές και τις βασικές τεχνικές του προγραμματισμού και της αλγοριθμικής σκέψης.

Αναφορές

- Arduino/Genuino. (2016, 7). Ανάκτηση από <https://www.arduino.cc>
- ECMA-334. (2006). Standard ECMA-334, C# Language Specification. Ανάκτηση από Standard ECMA-33 <http://www.ecma-international.org/publications/files/ECMA-ST/Ecma-334.pdf>
- Hejlsberg, A. (2011). "Future directions for C# and Visual Basic". C# lead architect. Microsoft. Ανάκτηση από Hejlsberg, Anders. "Future directions for C# and Visual Basic". C# lead architect. Microsoft. Retrieved September 21, 2011
- <https://pypi.python.org/pypi/pyserial>. (2016). <https://pypi.python.org/pypi/pyserial>. Ανάκτηση από <https://pypi.python.org/pypi/pyserial>
- Liechti, C. (2001-2016). Python Serial Port Extension for Win32, OSX, Linux, BSD. Ανάκτηση από <https://github.com/pyserial/pyserial>
- Mcwhorte, P. (2014, 6 18). Using Python with Arduino. Ανάκτηση από Using Python with Arduino: <http://www.toptechboy.com/using-python-with-arduino-lessons/>
- Python, C. (n.d.). <https://www.python.org/>. Ανάκτηση από <https://www.python.org/>
- Rosenbaum, E., Eastmond, E. & Mellis, D. (2010, 1 24-27). Empowering programmability for tangibles. In Proceedings of the 4th international Conference on Tangible Embedded, and Embodied interaction. TEI '10. ACM, New York, NY, 357-360. Cambridge, Massachusetts, USA. Ανάκτηση από <http://web.media.mit.edu/~ericr/papers/rosenbaum-Scratch4Arduino-TEI.pdf>
- S4A, MIT. (n.d.). Scratch for Arduino, 1.6. Ανάκτηση από Scratch for Arduino: <http://s4a.cat>
- Zieris, H., Gerstberger, H. & Müller, W. (2014). Conference: KEYCIT 2014 – Key Competencies in Informatics and ICT. Στο Zieris, H. & Gerstberger, H. & Müller, W., Using Arduino-Based Experiments to Integrate Computer Science Education and Natural Science. (pp. 238-243). Potsdam.
- Διαμαντόπουλος, Χ. Βλάχου, Χ., & Μπερδούσης, Ι. (2015). Πρόταση διαθεματικής διδασκαλίας της δομής επιλογής με αξιοποίηση του Scratch S4A και του Arduino. «Η εκπαίδευση στην εποχή των Τ.Π.Ε.», (σσ. 1288- 1297, ISBN: 978-960-99435-8-1). Αθήνα.
- Ορφανάκης, Β. & Παπαδάκης, Σ. (2014). Μια δραστηριότητα διδασκαλίας προγραμματισμού με τη χρήση του Scratch για Arduino (S4A). Proceedings of 6th Conference on Informatics in Education CIE, (σσ. 540-549). Κέρκυρα.

Πουλάκης, Ε. (2015, 1). Προγραμματίζοντας με τον μικροελεγκτή Arduino. (Ε. Πουλάκης, Επιμ.) Ανάκτηση από Προγραμματίζοντας με τον μικροελεγκτή Arduino: <http://users.sch.gr/manpoul/docs/arduino/ProgrammingArduino.pdf>

ΠΑΡΑΡΤΗΜΑ

Wiring C (Arduino IDE)

```
int led = 9;           // the PWM pin the LED is attached to
int brightness = 0;   // how bright the LED is
int fadeAmount = 5;   // how many points to fade the LED by

// the setup routine runs once when you press reset:
void setup() {
  // declare pin 9 to be an output:
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  // set the brightness of pin 9:
  analogWrite(led, brightness);

  // change the brightness for next time through the loop:
  brightness = brightness + fadeAmount;

  // reverse the direction of the fading at the ends of the fade
  if (brightness <= 0 || brightness >= 255) {
    fadeAmount = -fadeAmount;
  }
  // wait for 30 milliseconds to see the dimming effect
  delay(30);
}
```

S4A



Κώδικας σε wiring C (Arduino IDE). (ο Arduino περιμένει να «διαβάσει» μια τιμή από τη σειριακή θύρα και τη στέλνει ως είσοδο στο led)

```

int data_rx;
int led = 9;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  pinMode(led,OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  data_rx = Serial.read();
  analogWrite(led,data_rx);
  delay(300);
}

```

Κώδικας σε C#. (Το παρακάτω πρόγραμμα στέλνει αδιαλλείτως τιμές φωτεινότητας στο led τις οποίες διαβάζει ο Arduino)

```

using System;
using System.Text;
using System.Threading;
using System.Threading.Tasks;
using System.IO.Ports;

namespace ReadAnalogInputA9
{
  class Program
  {
    static void Main(string[] args)
    {
      int step = 5;
      int brightness = 1;
      SerialPort myport = new SerialPort();
      myport.BaudRate = 9600;
      myport.PortName = "COM9";
      myport.Open();

      while (true)
      {
        myport.WriteLine((brightness).ToString());
        brightness = brightness + step;

        if(brightness<=0 || brightness >=255)
          step = step * -1;
        Console.WriteLine(brightness.ToString());
        Thread.Sleep(
          (int)TimeSpan.FromSeconds(0.08).TotalMilliseconds);
      }
    }
  }
}

```

Κώδικας σε Python. (Το παρακάτω πρόγραμμα στέλνει αδιαλλείτως τιμές φωτεινότητας στο led τις οποίες διαβάζει ο Arduino)

```
import serial
import time

brightness = 0
step = 5

ard = serial.Serial('com9',9600)

while 1:
    ard.write(str(brightness).encode('utf-8'))
    brightness += step
    if (brightness <= 0 or brightness >= 255):
        step = step * (-1)
    time.sleep(0.08)
```

Abstract

In this paper, Arduino UNO main characteristics and capabilities as a low cost and trustworthy teaching solution are presented. Our approach is mainly intended for IoT (Internet of Things) systems that their usage is dramatically increased especially at Informatics School Laboratories. Interaction with robotics has already come true not only to industry field but the education itself. More specifically, this project is dealing with Input/Output (I/O) programming notion clarifications for Secondary School students. Students are called to understand what is the meant by “Input” and “Output” for an IoT system, working with sensors and servo motors connected on Arduino Uno board. Our example is implemented with Wiring C, S4A and C#, Python as high level programming languages alternatives.

Keywords: arduino, S4A, C#, Python, ΣΕΠΕΗΥ, IoT, STEM