

# Εφαρμογές Physical Computing με Raspberry Pi. Αξιοποίηση στη διδασκαλία του προγραμματισμού

Αναστάσιος Χατζηπαπαδόπουλος<sup>1</sup>, Δρ. Δημήτριος Λουκάτος<sup>2</sup>,  
Δρ. Βασίλης Σ. Μπελεσιώτης<sup>3</sup>

<sup>1</sup>Καθηγητής Πληροφορικής, 6ο ΕΠΑΛ/1ο ΕΚ Αθήνας  
[chatzipap@gmail.com](mailto:chatzipap@gmail.com)

<sup>2</sup>Καθηγητής Πληροφορικής ΠΣΠΑ  
[dlouka@sch.gr](mailto:dlouka@sch.gr)

<sup>3</sup>Σχολικός Σύμβουλος Πληροφορικής  
[vbelesiotis@sch.gr](mailto:vbelesiotis@sch.gr)

## Περίληψη

Η σύνδεση του πραγματικού κόσμου με τον ιδεατό του υπολογιστή, Physical Computing, αποτελεί έναν χώρο με μεγάλη συμβολή στην εκπαίδευση επιτυγχάνοντας υψηλό βαθμό μάθησης. Έτσι, έχει αναπτυχθεί μια πληθώρα εφαρμογών και προτάσεων ένταξης ανάλογων συστημάτων στη διδακτική διαδικασία, τόσο για την αύξηση του βαθμού της Υπολογιστικής Σκέψης του μαθητή, όσο για την ανάπτυξη δεξιοτήτων και ικανοτήτων επιτυχούς αντιμετώπισης προβλημάτων του πραγματικού κόσμου. Οι εμπορικές προσεγγίσεις με αντίστοιχο υλικό και λογισμικό, καθώς και οι τρόποι ένταξής τους σε θεματικές περιοχές της επιστήμης της Πληροφορικής και των άλλων μαθημάτων είναι πολλές. Στο άρθρο αυτό θα ασχοληθούμε με τη χρήση του συστήματος Raspberry Pi, μέσω του υποσυστήματος GPIO που διαθέτει, καθώς και τη συνεργασία του με το Arduino, σχετικά με τα παραπάνω και πιο συγκεκριμένα, με την περίπτωση που το σύστημα έχει ήδη αποκτηθεί για το εργαστήριο Πληροφορικής. Περίπτωση που δεν επιφέρει πρόσθετο κόστος για την υλοποίηση εφαρμογών Physical Computing και αυτό, σε εποχές που το κόστος απόκτησης και συντήρησης αποτελεί ρυθμιστικό παράγοντα στην προμήθεια εξοπλισμού ενός σχολείου.

**Λέξεις κλειδιά:** Physical Computing, Raspberry Pi, GPIO, διδασκαλία Πληροφορικής, κοντροκτιβισμός

## 1. Εισαγωγή

Στο πλαίσιο διδασκαλίας της Πληροφορικής, αλλά και των υπολοίπων μαθημάτων, προκειμένου να επιτευχθούν οι διδακτικοί στόχοι και να βελτιστοποιηθεί το γνωστικό αποτέλεσμα, οι εκπαιδευτικοί εφαρμόζουν διάφορες παιδαγωγικές προσεγγίσεις και διδακτικές μεθόδους, χρησιμοποιώντας κατάλληλα εργαλεία λογισμικού αλλά και υλικού. Αυτό, σε μια προσπάθεια κατάκτησης της γνώσης από το μαθητή, την απόκτηση δεξιοτήτων, την ανάπτυξη της δημιουργικότητας και για τα μαθήματα Πληροφορικής, την καλλιέργεια της Υπολογιστικής Σκέψης των μαθητών. Ο βασικός χώρος που προσφέρεται για την υποστήριξη τέτοιων δράσεων είναι το εργαστήριο Πληρο-

φορικής, που μέσω του κατάλληλου σύγχρονου υλικού και λογισμικού μπορεί να υποστηρίξει τις αντίστοιχες δραστηριότητες.

Στις μέρες μας που οι ανάγκες λόγω, τόσο των προγραμμάτων σπουδών όσο και των επιταγών της σύγχρονης διδακτικής των αντικειμένων έχουν αυξηθεί σε χρήση Νέων Τεχνολογιών, η κατάσταση του εργαστηρίου αποτελεί βασική συνιστώσα στη διδασκαλία των μαθημάτων. Η δομή του, καθώς και η ποιότητα και ποσότητα του παρεχόμενου εξοπλισμού, μπορεί να αποτελέσει ενισχυτικό ή όχι παράγοντα στις προσπάθειες επίτευξης των διδακτικών στόχων, ιδιαίτερα του εκπαιδευτικού Πληροφορικής. Αυτά, με το κόστος απόκτησης, αναβάθμισης και συντήρησης να αποτελεί σήμερα τον πλέον καθοριστικό παράγοντα. Με τον καθηγητή Πληροφορικής κύριο αποδέκτη προβλημάτων και δυσκολιών. Παράλληλα, μεμονωμένες προμήθειες πολλών σχολικών μονάδων από έργα επιτείνουν τα προβλήματα και οι αποσπασματικές αναβαθμίσεις παλαιού εξοπλισμού και δωρεές από διάφορους φορείς που δεν ακολούθησαν μια συνολικά ενιαία σχεδίαση, οδήγησαν σε μια κατάσταση μη ικανοποιητική τόσο ποσοτικά όσο ποιοτικά. Κατάσταση που έχει ως συνέπεια να επηρεάζει αρνητικά τη διδασκαλία.

Οι νέες τάσεις στο χώρο της Πληροφορικής και των συσκευών επικοινωνίας, ολοένα και περισσότερο βασίζονται στη χρήση, φορητών συσκευών και στο αντίστοιχο λογισμικό τους. Οι συσκευές αυτές αποτελούνται από μία κάρτα, με όλες τις μονάδες ενσωματωμένες σε αυτήν. Σχεδίαση που ανάλογα με τις δυνατότητές της, έχει μικρό κόστος, ανυπαρξία σχεδόν βλαβών και μικρές απαιτήσεις ενέργειας, παρέχοντας αυτονομία για ορισμένες χρήσεις τους. Πολλές από τις συσκευές αυτές, επικεντρώνονται στην επικοινωνία με τον πραγματικό κόσμο, άλλες αποτελούν υπολογιστικά συστήματα, ενώ κάποιες καλύπτουν και τις δύο κατηγορίες. Εδώ, συναντάμε διάφορες εταιρείες που δραστηριοποιούνται στον ευρύτερο χώρο της τεχνολογίας και κατασκευάζουν εμπορικά τέτοια προϊόντα, όπως ρομποτικά, συστήματα υποστήριξης αναγνωρισμένων μεθοδολογιών, όπως η STEM, με υπολογιστές ή μικροελεγκτές κάρτας και χαμηλού κόστους.

Στην ελληνική εκπαίδευση, με βάσει τα παραπάνω, υπάρχει η ανάγκη για τη διάδοση της λύσης απόκτησης: α) υπολογιστικών συστημάτων κάρτας χαμηλού κόστους και β) μικροελεγκτών παρόμοιας αρχιτεκτονικής. Αυτό, παράλληλα με τη συνεχή κατάθεση αξιόλογων προτάσεων ένταξής τους στη διδακτική πράξη, καθώς και την ορθή εκμετάλλευση των αποτελεσμάτων χρήσης τους.

Στους παραπάνω δύο άξονες έρχεται να συνεισφέρει το άρθρο αυτό, με προτάσεις ένταξης στο διδακτήριο του υπολογιστικού συστήματος και μικροελεγκτή Raspberry Pi, αλλά και του μικροελεγκτή Arduino. Το άρθρο διαρθρώνεται ως εξής. Στην ενότητα 2, Θεωρητικό & τεχνικό υπόβαθρο, αναλύονται θέματα σχετικά με τα αναφερόμενα αντικείμενα, ενώ στην ενότητα 3, Προτάσεις εφαρμογών Physical Computing, παρουσιάζονται προτάσεις εφαρμογών σε προγραμματιστικό περιβάλλον Python και Scratch με τις αναφερόμενες μονάδες, Τέλος στην ενότητες 4 κατατίθενται στοιχεία

από την εμπειρία χρήσης τους, ενώ στην 5 ακολουθούν τα αντίστοιχα συμπεράσματα και η μέλλουσα εργασία.

## **2. Θεωρητικό & τεχνικό υπόβαθρο**

### **2.1 Physical Computing**

Με σκοπό την ανάπτυξη υψηλών αποτελεσμάτων μάθησης, οι δραστηριότητες όπου ο μαθητής *Δημιουργεί* (Anderson et all, 2001) πρέπει να έχουν κυρίαρχη θέση στο σχεδιασμό μαθημάτων. Μεταξύ των δραστηριοτήτων με υψηλό ποιοτικά επίπεδο, είναι αυτές που σχετίζονται με χειροπιαστά αποτελέσματα του φυσικού κόσμου, πέραν αυτών του ιδεατού του υπολογιστή, κατάσταση που καλύπτεται, από τον καθολικό όρο Physical Computing-PhC (O'Sullivan, 2014). Ο όρος αυτός πιστεύουμε ότι είναι ευρύτερος του όρου Ρομποτική, όπου συχνά περιορίζεται στο αυτόνομο αντικείμενο. Μια και ο όρος δεν έχει ακόμη μια επικρατούσα ελληνική μετάφραση, θα χρησιμοποιούμε στη συνέχεια τον αγγλικό όρο. Για να επιτευχθούν καταστάσεις PhC απαιτείται από το σύστημα υλικού να έχει τη δυνατότητα επικοινωνίας με το φυσικό κόσμο, πράγμα που γίνεται μέσω ειδικών διαύλων επικοινωνίας του συστήματος που δέχονται ή στέλνουν σήματα από/προς διάφορους αισθητήρες.

### **2.2 Συστήματα κάρτας, χαμηλού κόστους**

Κατά καιρούς έχουμε συναντήσει διάφορα συστήματα που συγκεντρώνουν τις μονάδες τους σε μία βασική κάρτα. Τέτοια συστήματα συχνά βρίσκονταν ενσωματωμένα σε άλλα (embedded systems) με ποικίλους ρόλους, ή και αποτελούσαν τη βάση ενός προσωπικού υπολογιστή. Πρόσφατα η τεχνολογία αυτή έχει επικρατήσει σε καθημερινές συσκευές, όπως για παράδειγμα είναι και τα κινητά τηλέφωνα. Πολλά δε από τα συστήματα αυτής της κατηγορίας συγκροτούν έναν μικροελεγκτή, με δυνατότητες υλοποίησης ποικίλων εφαρμογών, όπως ρομποτικά συστήματα ή γενικότερα συστήματα αυτοματισμού και ελέγχου, αναγνώρισης καταστάσεων του φυσικού κόσμου. Στην κατηγορία αυτή επικρατεί ο μικροελεγκτής Arduino (Arduino, 2016), στον οποίο αναφέρεται ήδη ένας μεγάλος αριθμός διδακτικών προτάσεων και εκπαιδευτικών εφαρμογών. Άλλα, αποτελούν ένα υπολογιστικό σύστημα περιέχοντας και υποσύστημα επικοινωνίας με το φυσικό κόσμο, με προφανή τη δυνατότητα λειτουργίας τους υπό όρους και στο ρόλο του μικροελεγκτή. Ένα τέτοιο σύστημα είναι το Raspberry Pi (RPI), που έχει δυνατότητα υλοποίησης PhC μέσω του υποσυστήματός του GPIO.

Στη συνέχεια θα αναφερθούμε, για λόγους δυνατής έκτασης του άρθρου, επιλεκτικά μόνο σε ειδικά χαρακτηριστικά του συστήματος Raspberry Pi χρήσιμα για τους σκοπούς του άρθρου.

### **2.2.1 Raspberry Pi, η διεπαφή GPIO και PhC**

Το σύστημα εμφανίστηκε το 2012, και σήμερα βρίσκεται στην έκδοση Raspberry Pi3, ModelB, (RaspberryPi, 2016), ενώ διαθέτει χαρακτηριστικά τα οποία σε κάθε έκδοσή του βελτιώνονται, με το κόστος του να παραμένει σταθερό. Μια από τις σημαντικότερες καινοτομίες του RPi, όπως προαναφέραμε, είναι η διεπαφή του GPIO (General Purpose input/output), που στα τελευταία μοντέλα RPi2, RPi3 και RPi Zero αποτελείται από 40 ακίδες (Pins) (Raspberry GPIO, 2016). Μέσω αυτής επιτυγχάνεται και η επικοινωνία με το φυσικό κόσμο, δυνατότητα που μεταφράζεται άμεσα και σε κόστος.

Η τρέχουσα διανομή ΛΣ Raspbian (Raspbian, 2016) υποστηρίζει το χειρισμό της διεπαφής GPIO σε διάφορα προγραμματιστικά περιβάλλοντα, ανάμεσα στα οποία είναι και τα περιβάλλοντα-γλώσσες προγραμματισμού Scratch και Python που χρησιμοποιούνται στα τρέχοντα αναλυτικά προγράμματα σπουδών του Γυμνασίου και του Επαγγελματικού Λυκείου αντίστοιχα.

### **2.2.2 Raspberry Pi και Arduino**

Το Raspberry Pi μπορεί να συνεργασθεί πολύ αποτελεσματικά και με τον μικροελεγκτή Arduino. Διασύνδεση που κρίνεται αρκετά σκόπιμη όταν έχουν ήδη αναπτυχθεί εφαρμογές υλικού/λογισμικού πάνω στο Arduino ή απαιτείται μια γρήγορη μέθοδος ψηφιοποίησης των δεδομένων από τον αναλογικό κόσμο που μας περιβάλλει. Η παραπάνω διασύνδεση ενδείκνυται να γίνεται, τουλάχιστον για τους άπειρους χρήστες, μέσω κάποιας από τις USB θύρες του RPi. Στην περίπτωση ακριβώς της διασύνδεσης του Arduino με το RPi, ο Arduino γίνεται συνήθως αντιληπτός από το περιβάλλον του Raspbian ως μία συμβατική θύρα η οποία και μπορεί να τύχει της κατάλληλης διαχείρισης μέσω του περιβάλλοντος της Python, με χρήση του ανάλογου αρθρώματος, για παράδειγμα του pySerial (pySerial, 2016).

### **2.3. Διδακτήριο, υπολογιστές και μικροελεγκτές κάρτας χαμηλού κόστους**

Έτσι, πιστεύουμε ότι μεταξύ των λύσεων ένταξης νέου εξοπλισμού -υλικού και λογισμικού- στο εργαστήριο, πρέπει, πέραν της συμβατότητας ένταξης στο πρόγραμμα σπουδών, να ακολουθούμε μοντέλα τα οποία να είναι, ευέλικτα ώστε να προσαρμόζονται στην ποικιλία του υφιστάμενου εξοπλισμού, διαχειρίσιμα ώστε να ελαχιστοποιούν το απαιτούμενο χρόνο συντήρησης και χαμηλού κόστους, ώστε να μην επιβαρύνουν σημαντικά το διαθέσιμο προϋπολογισμό του σχολείου.

Κατά συνέπεια, και μετά από αρκετές προτάσεις χρήσης του, καθώς και εκείνη της εφαρμογής του σε σχολείο (Λουκάτος, Χατζηπαπαδόπουλος, Μπελεσιώτης 2016), θεωρούμε πια το σύστημα RPi είναι κατάλληλο, υπό προϋποθέσεις βέβαια, για εξοπλισμό διδακτηρίων.

### 3. Προτάσεις εφαρμογών *Physical Computing*

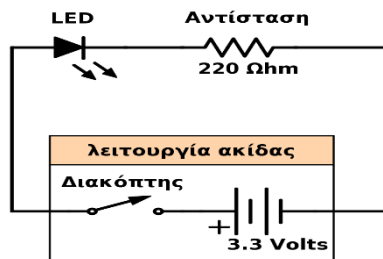
Στην ενότητα αυτή παρατίθενται και περιγράφονται σενάρια δραστηριοτήτων εφαρμογής PhC με RPi/Arduino και Python/Scratch.

#### 3.1 Χρήση RPi και Python σε ΑΠΣ προγραμματισμού ΕΠΑ.Λ.

Πριν την περιγραφή ορισμένων δραστηριοτήτων σε Python αναφέρουμε εισαγωγικά, ότι η γλώσσα προγραμματισμού Python χρησιμοποιείται ήδη από το 2015-16 τόσο στη Β΄ τάξη των Επαγγελματικών Λυκείων (ΕΠΑ.Λ.) και στο μάθημα Αρχές Προγραμματισμού του Τομέα Πληροφορικής όσο και στη Γ' τάξη, στο μάθημα γενικής παιδείας Εισαγωγή στις Αρχές Επιστήμης των ΗΥ όλων των τομέων και ειδικοτήτων. Από τη σχολική χρονιά 2016-17, θα χρησιμοποιείται και στο πανελλαδικά εξεταζόμενο μάθημα της Γ' τάξης του τομέα Πληροφορικής, Προγραμματισμός Υπολογιστών.

##### 3.1.1 Διαχείριση GPIO σε Python

Τεχνικά, για την υλοποίηση εφαρμογών PhC με τη γλώσσα προγραμματισμού Python και την ανάπτυξη σχετικών δραστηριοτήτων στα παραπάνω μαθήματα, πρέπει να μπορούμε να χρησιμοποιήσουμε σχετικές βιβλιοθήκες της γλώσσας. Έτσι, θα αναφέρουμε δύο τέτοιες βιβλιοθήκες. Τη βιβλιοθήκη που αρχικά αναπτύχθηκε, **RPi.GPIO** (RPi.GPIO, 2016), μέσω της οποίας είναι δυνατός ο έλεγχος της διεπαφής GPIO και τη νεώτερη, **gpiozero** (gpiozero, 2016), η οποία επιτυγχάνει έναν υψηλότερο βαθμό αφαίρεσης από την πρώτη και στην οποία θα αναφερθούμε παρακάτω. Η χρήση της τελευταίας προσφέρει απλό χειρισμό σε κυκλώματα όπως αυτό της εικόνας 1, ή και πιο σύνθετα με στοιχεία όπως: Button και RGB LED, Buzzer, Motor, Motion Sensor, Light Sensor, Distance Sensor, ADC Converters (MCP3004 & MCP3008).



Εικόνα 1. Κύκλωμα LED και έλεγχος μέσω ακίδας GPIO

##### 3.1.2 Προτάσεις δραστηριοτήτων PhC με Python

Στη συνέχεια θα δοθεί ένα σενάριο δραστηριότητας εφαρμογής Physical Computing, που περιλαμβάνει ένα απλό ηλεκτρονικό κύκλωμα σύμφωνα με τη λογική του σχήματος της εικόνας 1. Επίσης ο τελικός συνοδός κώδικας, ο οποίος προτείνεται να οι-

κοδομείται σταδιακά με ολοένα και με πιο σύνθετα ερωτήματα, ώστε με αυτόν τον τρόπο να εστιάζουμε περισσότερο στη συγγραφή και βελτιστοποίησή του και λιγότερο στην ίδια την κατασκευή. Κάθε λάθος, αλλαγή ή τροποποίηση στον κώδικα έχει πέραν ενός μηνύματος κόκκινου χρώματος στην οθόνη του διερμηνέα και άμεσες συνέπειες στο ανάπτυγμα του φυσικού κόσμου. Έτσι, οι μαθητές προσπαθώντας να λύσουν το πρόβλημα και να επιτύχουν το φυσικό αποτέλεσμα, οδηγούνται αναγκαστικά στη βελτίωση του κώδικά τους, ανακαλύπτοντας, τις συνέπειες της αλλαγής στη ροή εκτέλεσης των εντολών, των συντακτικών και λογικών λαθών, τη χρησιμότητα των συναρτήσεων, τον τρόπο λειτουργίας των αλγοριθμικών δομών και των δομών δεδομένων. Στις ενδεικτικές λύσεις χρησιμοποιείται η Python στην έκδοση 3 και η βιβλιοθήκη gpiozero.

**Το αρχικό πρόβλημα** το οποίο στη συνέχεια, όπως θα δούμε, μπορεί να επεκταθεί με διάφορους τρόπους, είναι η υλοποίηση ενός **φωτεινού σηματοδότη οχημάτων**. Για την κατασκευή απαιτούνται 3-6 leds διαφορετικού χρώματος, 1 Buzzer και 3-6 αντιστάσεις 220 Ω.

**Πίνακας 1.** Ενδεικτική λύση του βήματος 5 του φωτεινού σηματοδότη

<pre> from gpiozero import LED from time import sleep from datetime import datetime # Αντιστοίχιση ακίδων GPIO red = ( LED(27), LED(21) ) orange = ( LED(19), LED(13) ) green=( LED(5), LED(22) ) buzzer = LED(10) # Αρχικοποίηση καταστάσεων red[0].off() ; red[1].on() orange[0].off() ; orange[1].off() green[0].on() ; green[1].off() sleep(5) # Συνάρτηση λειτουργίας buzzer def buzzerRing(t):     buzzer.on()     sleep(t)     buzzer.off() # Λειτουργία Πορτοκαλί Φαναριού def orangeLight(num,t):     orange[num].on() </pre>	<pre> sleep(t) buzzerRing(t) orange[num].off() # Συνάρτηση λειτουργίας σηματοδότη def trafficLight(tRed, tGreen):     red[1].on() ; green[1].off() ; red[0].off()     ; green[0].on()     print ('Κίνηση οχημάτων στο εργοστά- στο.')     sleep(tRed - tRed / 10)     orangeLight(0,tRed/10)     red[1].off() ; green[1].on() ; red[0].on()     ; green[0].off()     print ('Κίνηση Οχημάτων στη Λεωφόρο')     sleep(tGreen)     orangeLight(1,tRed/10)     green[1].off()     print ('Έναρξη Λειτουργίας Σηματοδότη.') # Κύρια Επαναληπτική Διαδικασία while True :     trafficLight(10,5) </pre>
--	--

**Βήμα 1ο.** Να κατασκευαστεί ανάπτυγμα που να εξομοιώνει ένα φωτεινό σηματοδότη για τα διερχόμενα οχήματα μιας οδού που οδηγεί σε ένα εργοστάσιο.

**Βήμα 2ο.** Προσθέστε στη συνέχεια ένα Buzzer και κάντε αλλαγές στον κώδικα, ώστε να προειδοποιεί με ηχητικό σήμα τους πεζούς με προβλήματα όρασης λίγο πριν ανάψει το πράσινο φανάρι και αρχίσει η κίνηση των οχημάτων.

**Βήμα 3ο.** Το φανάρι βρίσκεται σε οδό όπου λειτουργεί εργοστάσιο με 3 οκτώωρες βάρδιες (7:00,15:00,23:00) των 200 περίπου εργαζομένων. Να γίνεται αυτόματη αύξηση του χρόνου διέλευσης των οχημάτων στις παραπάνω ώρες.

**Βήμα 4ο.** Το Βήμα 3 να λαμβάνεται υπόψη μόνο κατά τις εργάσιμες ημέρες, το Σαββατοκύριακο το εργοστάσιο είναι κλειστό.

**Βήμα 5ο.** (Τροποποίηση Σεναρίου). Να γίνει συζήτηση και μελέτη της περίπτωσης, υλοποίησης και ενός 2ου σηματοδότη ο οποίος θα βρίσκεται σε κάθετη λεωφόρο θα λειτουργεί συγχρονισμένα με τον 1ο σηματοδότη (πίνακας 1).

Λόγω έκτασης του άρθρου η αναλυτική περιγραφή και ο κώδικας όλων των βημάτων της άσκησης βρίσκεται στο σύνδεσμο: <https://github.com/chatzipap/CIE2016>

**Πίνακας 2.** Κώδικας εφαρμογής καταγραφής διερχομένων οχημάτων(HC-SR04)

<pre> from gpiozero import LED, Distance- Sensor from time import sleep from datetime import datetime # Αντιστοίχιση ακίδων GPIO red = LED(27); buzzer = LED(10) sensor = DistanceSensor(echo=16, trigger=20) # Αρχικοποίηση καταστάσεων red.off(); sleep(1) # Calibrate Distance sum = 0; print ('Calibrating Dis- tance...') for i in range(10):     d = int(sensor.distance * 100)     sum += d     sleep(0.5) # Αρχικοποίηση Μεταβλητών default = sum /10; count = 0; state = 0 print ('Default=', default) </pre>	<pre> fo = open('log.txt','a') # Άνοιξε αρχείο try :     while True:         d = int(sensor.distance * 100)         if abs(d-default) &lt;= 2 :             state = 0             sleep(0.2)         if abs(d-default) &gt; 2 and state ==0:             count += 1             red.on() ; buzzer.on(); sleep(0.1)             buzzer.off();red.off();             now = datetime.now()             rec = ', '.join( [str(item) for item in [now,d,count]])             print (rec); fo.write(rec + '\n')             state = 1 except :     print ('I/O Error') finally :     fo.close()     print ('the end') </pre>
--	--

Εάν επιθυμούμε, μπορούμε να επεκτείνουμε το κύκλωμα καθώς και το συνοδευόμενο κώδικα προσθέτοντας:

α) Ένα **button**, ώστε να προκαλέσουμε ένα εξωτερικό συμβάν από το φυσικό κόσμο ως είσοδο και όχι εσωτερικό, όπως με την ώρα της ημέρας ή την ημέρα εβδομάδας. Αυτό το κουμπί, για παράδειγμα, θα το πατούσε ένας πεζός όταν θα επιθυμούσε να διασχίσει ένα πολυσύχναστο δρόμο για να προκαλέσει μεταβολή στους αρχικούς χρόνους (κόκκινο/πράσινο φανάρι) για να διευκολυνθεί η διέλευσή του.

β) Έναν **αισθητήρα απόστασης υπερήχων** (sensor HC-SR04). Αυτό για να υλοποιήσουμε ένα μετρητή διερχομένων οχημάτων και καταγραφή της κυκλοφοριακής κίνησης. Στη συνέχεια αφού εκτελεστεί ο κώδικας (βλ. πίνακα 2) μπορούμε να αναλύσουμε τα δεδομένα της κίνησης και να τα μελετήσουμε στατιστικά, ώστε ανάλογα την ώρα της ημέρας ή την ημέρα της εβδομάδας, τα αποτελέσματα να λειτουργήσουν ως ανατροφοδότηση για αυτόματη διαμόρφωση του χρόνου διάρκειας των φαναριών και καλύτερη διαχείριση της κίνησης.

### Παρατήρηση

Εάν θέλουμε να δώσουμε **έμφαση σε δημιουργία άλλης κατασκευής** μπορεί να δοθεί ως άσκηση το ίδιο κύκλωμα με άλλο σενάριο (άρα και άλλος κώδικας) για να κατασκευάσουμε, για παράδειγμα, ένα βοηθό στάθμευσης (parking assistant) ο οποίος θα περιλαμβάνει, ηχητική (Buzzer), φωτεινή (LEDs) ένδειξη, καθώς και προβολή της υπολειπόμενης απόστασης με character LCD Display.

Όλα τα παραπάνω υλικά έχουν από ελάχιστο ως μικρό κόστος και είναι επαναχρησιμοποιούμενα, ενώ τα κυκλώματα είναι εύκολα στην κατασκευή τους και έχουν λογικό χρόνο υλοποίησης. Έτσι μπορούμε να επεκταθούμε με μοναδικό όριο αυτό της φαντασίας και της δημιουργικότητας. Προσθέτοντας δε επιπλέον στοιχεία στην κατασκευή ή στο σενάριο, μπορούμε να εμβαθύνουμε σε αλγοριθμικό και προγραμματιστικό επίπεδο ανάλογα βέβαια και με τη σύνθεση και τις δυνατότητες του μαθητικού δυναμικού αλλά και το σχεδιασμό του εκπαιδευτικού και τους διδακτικούς του στόχους.

### 3.2 Χρήση RPi και Scratch σε ΑΠΣ Γυμνασίου

Για σχετικά μικρότερες ηλικίες, όπως οι μαθητές Γυμνασίου, ο προγραμματισμός με χρήση εντολών σε περιβάλλον κειμένου ενέχει δυσκολίες και δεν είναι πάντα τόσο ελκυστικός όσο θα περίμενε κανείς. Για το λόγο αυτό περιβάλλοντα οπτικού προγραμματισμού πλακιδίων, όπως το Scratch (Scratch, 2016) αποδεικνύονται διδακτικά, ιδιαίτερα αποτελεσματικά και είναι αγαπητά στους μαθητές. Το Raspberry Pi, στις Raspbian εκδόσεις λογισμικού που παρέχει, περιλαμβάνει και τον προγραμματιστικό μικρόκοσμο του Scratch και φυσικά υποστηρίζει τις προφανείς δυνατότητες του καθόλα πετυχημένου αυτού εργαλείου, δηλαδή σε μορφή περιβάλλοντος διάδρα-



σης συμβατικού υπολογιστή, με πληκτρολόγιο, ποντίκι και οθόνη. Συνεπώς μπορούν οι μαθητές να χρησιμοποιούν το Scratch μέσα από το Raspberry Pi, με μεγάλο όφελος σε κατανάλωση ενέργειας, όγκο και κόστος, μεγέθη στα οποία πλεονεκτεί το Raspberry Pi σε σχέση με έναν συμβατικό υπολογιστή.

Φυσικά αυτό που ενδιαφέρει ιδιαίτερα, σχετικά με το Scratch, είναι η δυνατότητα να επεκταθεί η χρήση του, πέρα από ένα πετυχημένο εργαλείο προγραμματισμού για περιβάλλον διάδρασης συμβατικού υπολογιστή και σε ένα αποτελεσματικό μέσο προγραμματισμού σεναρίων αλληλεπίδρασης με το φυσικό κόσμο. Αυτό επιτυγχάνεται, για την περίπτωση του Arduino, με περιβάλλοντα προγραμματισμού όπως το S4A (S4A, 2016), ενώ είναι ιδιαίτερα ελκυστική η ύπαρξη αντίστοιχης λύσης και για την περίπτωση του Raspberry. Η τελευταία περίπτωση αναλύεται ακριβώς στα επόμενα.

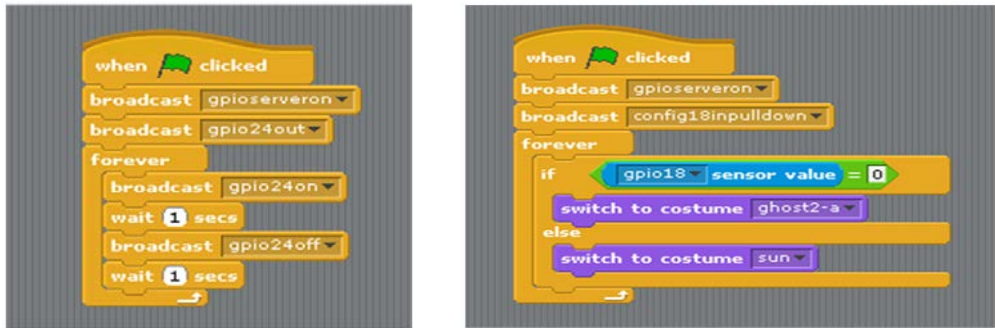
### **3.2.1 Διαχείριση GPIO σε Scratch**

Στο σημείο αυτό έρχεται το Raspberry Pi να συνεισφέρει δυναμικά, καθώς με την τελευταία έκδοση της διανομής Raspbian, την Raspbian Jessie, απλοποιεί αρκετά τη διαδικασία προγραμματισμού των GPIO ακίδων του, μέσα ακριβώς από το περιβάλλον του Scratch που φιλοξενεί το ίδιο (Scratch GPIO, 2016). Με τον τρόπο αυτό, χωρίς σύνθετα πρόσθετα συστήματα, μπορούν οι μαθητές να υλοποιήσουν προγραμματιστικά και κατασκευαστικά ευφάνταστα σενάρια διάδρασης με το φυσικό κόσμο. Πιο συγκεκριμένα, στο Raspberry Pi η διαχείριση των GPIO ακίδων, μέσα από το Scratch, γίνεται με την ευρεία χρήση broadcast μηνυμάτων. Η σύνταξη και η χρήση των εντολών ελέγχου των GPIO ακίδων μέσω του Scratch είναι καλά τεκμηριωμένη στο δικτυακό τόπο του Raspberry Pi, ενώ υπάρχουν και κατατοπιστικά παραδείγματα και μέσα στο περιβάλλον του Scratch. Η συνήθης διαδικασία χρήσης των GPIO ξεκινά με ενεργοποίηση του GPIO εξυπηρετητή, τον ορισμό μίας συγκεκριμένης ακίδας ως είσοδο ή έξοδο δεδομένων και την ανάθεση τιμής σε μία ακίδα, όταν αυτή είναι ακίδα εξόδου. Πέρα από τις προφανείς GPIO εντολές παρέχονται και άλλες που επιστρέφουν την τιμή της μεταβλητής χρόνος συστήματος, την IP διεύθυνση του Raspberry ή μπορούν να τραβήξουν μία φωτογραφία με την κάμερα που ενδέχεται να έχει συνδεθεί στο Raspberry.

### **3.2.2 Προτάσεις δραστηριοτήτων PhC με Scratch**

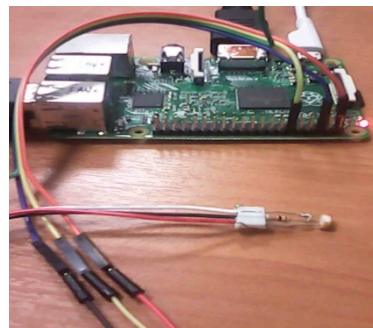
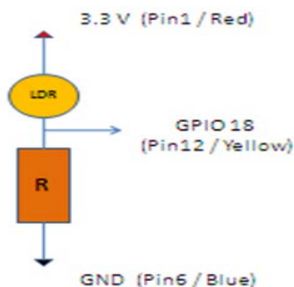
Ακολουθούν δύο πολύ απλά αλλά και κατατοπιστικά παραδείγματα PhC με την αξιοποίηση των GPIO ακίδων του Raspberry Pi, μέσω του Scratch, τα οποία για λόγους έκτασης του άρθρου, θα δοθούν με μικρή διδακτική ανάλυση. Στο πρώτο από αυτά, απλά ανάβουμε και σβήνουμε ένα LED. Αναλυτικότερα το LED είναι συνδεδεμένο ανάμεσα στην GPIO ακίδα 24 και τη γείωση, μαζί με 220 Ω αντίσταση σε σειρά με αυτό και ανάβει και σβήνει για ένα δευτερόλεπτο, όπως το παράδειγμα blink που εί-

ναι διαθέσιμο στο περιβάλλον του Arduino. Λεπτομέρεια του κώδικα σε Scratch φαίνεται στην εικόνα 2, στο αριστερό μέρος.



*Εικόνα 2. Κύκλωμα LED και έλεγχος μέσω ακίδας GPIO*

Στο δεύτερο παράδειγμα, έχοντας συνδέσει έναν αισθητήρα φωτεινότητας πάνω στο Raspberry, αλλάζουμε την εμφανιζόμενη μορφή από Ήλιο σε Φάντασμα, ανάλογα με το αν έχουμε μέρα ή νύχτα, αντίστοιχα. Ο αισθητήρας φωτός (φωτοαντίσταση) βρίσκεται με την έξοδό του συνδεδεμένη στην GPIO ακίδα 18. Λεπτομέρεια του κώδικα φαίνεται στην εικόνα 2 στο δεξί μέρος, ενώ στην εικόνα 3 φαίνεται ο τρόπος σύνδεσης του αισθητήρα φωτεινότητας στο Raspberry. Ο αισθητήρας φωτεινότητας παρέχει αναλογική έξοδο μεταξύ 0 Volt και 3.3 Volt και έχει συνδεθεί πάνω σε ψηφιακή είσοδο. Το γεγονός αυτό έχει σαν αποτέλεσμα τιμές εξόδου του αισθητήρα που είναι πιο κοντά στο 0 να θεωρούνται λογικό "0", ενώ τιμές που είναι πιο κοντά στα 3.3V να θεωρούνται λογικό "1", συμπεριφορά που αποτελεί ένα πολύ καλό παράδειγμα στο να εμβαθύνει κανείς στη διαφορά φιλοσοφίας ανάμεσα σε ένα αναλογικό και ένα ψηφιακό σύστημα.



*Εικόνα 3. Κύκλωμα LED και έλεγχος μέσω ακίδας GPIO*

#### **4. Εμπειρία από την υλοποίησης του PhC με RPi**

Η περίπτωση υλοποίησης του PhC, αμιγώς με το Raspberry Pi και με χρήση του περιβάλλοντος του Scratch, παρουσιάζει το πλεονέκτημα ότι οι μαθητές λόγω της εξοικείωσης το προγραμματιστικό αυτό περιβάλλον μπορούν εύκολα να συντάξουν απλά προγράμματα. Προσοχή πρέπει να δοθεί στην αντιστοίχιση των διάφορων ακίδων του Raspberry με τους αριθμούς των GPIO διεπαφών. Αξίζει ακόμα να σημειωθεί ότι, ειδικά όταν δεν χρησιμοποιείται κάποια ειδική πρόσθετη πλακέτα με αισθητήρες ή ενεργοποιητές σχεδιασμένους για το Raspberry Pi, απαιτείται αυξημένη προσοχή κατά τις συνδέσεις διακριτών στοιχείων πάνω στη μητρική πλακέτα του Raspberry, η οποία θυμίζουμε ότι λειτουργεί στα 3.3 Volt και όχι στα 5.0 Volt, όπως ένα συμβατικό σύστημα Arduino.

Η λύση διασύνδεσης Raspberry Pi και Arduino δοκιμάστηκε από τους μαθητές στο ΠΣΠΑ σχολείο, στα πλαίσια του *ομίλου δημιουργικότητας* με τίτλο Καινοτόμα Υπολογιστικά Συστήματα και Εφαρμογές, καθώς και στο μάθημα της Ερευνητικής Εργασίας στο Λύκειο, κατά το σχολικό έτος 2015-2016.

Το RPi χρησιμοποιήθηκε επίσης σε Δ.Ι.ΕΚ κατά το προηγούμενο σχολικό έτος και στο μάθημα Πρακτική Εφαρμογή, στην Ειδικότητα των Τεχνικών Δικτύων και Τηλεπικοινωνιών για τη διδασκαλία βασικών αλλά και πιο προχωρημένων εννοιών των δικτύων υπολογιστών (Λουκάτος, Χατζηπαπαδόπουλος, Μπελεσιώτης 2016), αλλά και για τη δημιουργία κατασκευών PhC σχετικών με την ειδικότητά τους. Η ανταπόκριση των μαθητών ήταν ιδιαίτερα θετική. Η διδασκαλία είναι σε εξέλιξη, θα συνεχιστεί η χρήση του και στα επόμενα εξάμηνα και θα επανέλθουμε με τα τελικά αποτελέσματα της εφαρμογής του.

#### **5. Συμπεράσματα - Μελλοντικοί Στόχοι**

Η Επιστήμη των Υπολογιστών έχει αποκτήσει ιδιαίτερη σημασία στην εκπαίδευση σε πάρα πολλές αναπτυσσόμενες και μη χώρες και χρησιμοποιείται ως μέσο απόκτησης Υπολογιστικής Σκέψης, υλοποιώντας με μέσα λογισμικού αλλά και υλικού το κονστрукτιβιστικό μοντέλο μάθησης. Οι εφαρμογές Physical Computing αποτελούν ισχυρά εργαλεία στην κατανόηση των αρχών του προγραμματισμού, αλλά και άλλων θεματικών περιοχών του κλάδου της Πληροφορικής, όπως για παράδειγμα στην Αρχιτεκτονική Συστημάτων, στα Συστήματα Ελέγχου, Σχεδιασμού Υλικού, Συλλογής & Επεξεργασίας Δεδομένων. Ιδιαίτερα τα συστήματα *ανοιχτής σχεδίασης* προσαρμόζονται πιο εύκολα στις ανάγκες του εκπαιδευτικού και στα προγράμματα σπουδών έχοντας ταυτόχρονα και χαμηλότερο κόστος κτήσης. Κυριότερα όμως οφέλη είναι τα καθαρά παιδαγωγικά που αφορούν στο υψηλό μαθησιακό επίπεδο, στην ενεργοποίηση της συμμετοχής και του ενδιαφέροντος των μαθητών, στην καλλιέργεια της δημιουργικής σκέψης και της φαντασίας τους. Ίσως να απέχουμε ακόμα αρκετά από τα εκπαιδευτικά μοντέλα άλλων χωρών και την ένταξη του PhC ως μοντέλο δόμησης

των προγραμμάτων σπουδών (National curriculum in England, 2013) για την κατανόηση θεματικών περιοχών της επιστήμης της Πληροφορικής.

Προς αυτή την κατεύθυνση δόθηκαν στο άρθρο ενδεικτικές δραστηριότητες ένταξης συστημάτων κάρτας στη διδασκαλία, ενώ κατατέθηκαν συμπεράσματα τάξης.

Οι ενδείξεις που έχουν οι συγγραφείς από την εμπειρία του PhC με συστήματα όπως το Raspberry Pi, αλλά και Arduino, στη διδασκαλία είναι ιδιαίτερα ενθαρρυντικές. Έτσι, θα γίνει προσπάθεια να συνεχιστούν οι αναφερόμενες εφαρμογές σε πιο ευρεία κλίμακα, όπου η πειστική σχολική καθημερινότητα μας το επιτρέπει, ενημερώνοντας παράλληλα για τα αποτελέσματα την εκπαιδευτική κοινότητα.

## **Αναφορές**

*Οι αναφορές σε Διαδικτυακές πηγές, προσπελάστηκαν τον 7/2016*

Anderson L. W., Krathwohl D. R., Airasian P. W., Cruikshank K. A., Mayer R E., Pintrich P. R, Rath,J. & Wittrock M. C. (2001). *A taxonomy for learning, teaching, and assessing: A revision of Bloom 's taxonomy of educational objectives*, New York: Longman.

Arduino (2016). Δικτυακός τόπος υποστήριξης του Arduino, <http://www.arduino.org/gpiozero> (2016). Πρόσφατη βιβλιοθήκη ελέγχου του GPIO με Python στο δικτυακό τόπο υποστήριξης, <https://gpiozero.readthedocs.io/en/v1.3.1/>

National curriculum in England (2013). *National curriculum in England computing programmes of study (2013)*, <https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study>.

O'Sullivan, D., Igoe, T. (2014). *Physical Computing: Sensing and Controlling the Physical World with Computers*, Thomson, ISBN-13: 008-2039503462 ISBN-10: 159200346X, Edition: 1st.

pySerial (2016). Δικτυακός τόπος υποστήριξης της βιβλιοθήκης pySerial, <http://pyserial.readthedocs.io/en/latest/pyserial.html>

Raspberry GPIO (2016). Λειτουργία της διεπαφής GPIO στο δικτυακό τόπο του Raspberry Pi, <https://www.raspberrypi.org/documentation/usage/gpio/>

Raspberry Pi (2016). Εκδόσεις συστημάτων υλικού από τον επίσημο δικτυακό τόπο του Raspberry Pi, <https://www.raspberrypi.org/products/>

Raspbian (2016). Διανομές της έκδοσης Raspbian του Debian Linux, <https://www.raspberrypi.org/downloads/raspbian/>

RPi.GPIO (2016). Αρχική βιβλιοθήκη ελέγχου του GPIO με Python από τον δικτυακό τόπο υποστήριξης και ανάπτυξης, <https://pypi.python.org/pypi/RPi.GPIO>.

S4A (2016). Λογισμικό που επιτρέπει τον προγραμματισμό του Arduino μέσω του περιβάλλοντος του MIT Scratch (Scratch For Arduino), <http://s4a.cat/>

Scratch GPIO (2016). Έκδοση του Scratch για το Raspberry που αυξάνει την ευχρηστία διαχείρισης των GPIO ακίδων, διαθέσιμες οδηγίες στον δικτυακό τόπο, <https://www.raspberrypi.org/documentation/usage/scratch/gpio/README.md>

Λουκάτος Δ. Χατζηπαπαδόπουλος Α. & Μπελεσιώτης Β. (2016). Πρακτικές Πιλοτικής Αξιοποίησης του Raspberry Pi στην Εκπαίδευση. *10<sup>ο</sup> Πανελλήνιο Συνέδριο Καθηγητών Πληροφορικής «Η Πληροφορική στην Πρωτοβάθμια και Δευτεροβάθμια Εκπαίδευση, Ρόλος και Εφαρμογές»* Ναύπλιο 16-17 Απριλίου 2016.

### Abstract

Physical Computing, which connects the real world with the virtual computer world, is a field with great contribution to education and the achievement of high-level learning. A variety of applications and proposals for the integration of Physical Computing to the teaching process have been developed. These aim to increase the students' computational thinking and develop their skills and real-world problem-solving abilities. Numerous commercial approaches exist, including both hardware and software solutions, as well as methodologies to integrate them into the teaching process of various thematic areas of computer science and other topics. This article deals with the use of the Raspberry Pi system, via the GPIO embedded subsystem and its connection with Arduino System, when it is already acquired and functioning in a computer lab. In this case, no additional costs are required for the implementation of Physical Computing applications. This is very important, particularly at a time in which the acquisition and maintenance costs are the most important factors related to the procurement of school equipment.

**Keywords:** Physical Computing, Raspberry Pi, GPIO, teaching informatics/teaching computer science, constructivism.