



«ΜΙΑ ΝΕΑ ΑΡΧΗ ΣΤΑ ΕΠΑΛ»

ΕΚΘΕΣΗ ΠΕΠΡΑΓΜΕΝΩΝ ΣΧΕΔΙΟΥ ΔΡΑΣΗΣ

ΤΙΤΛΟΣ ΕΡΓΟΥ (συνοπτικός)

«...“Νέα” αρχή σε “Νέο” ΕΠΑΛ !..»

ΑΝΑΔΟΧΟ ΣΧΟΛΕΙΟ

Εσπερινό ΕΠΑΛ Κομοτηνής.

ΤΙΤΛΟΣ ΔΡΑΣΗΣ

Νέα Έξυπνα Συστήματα.

ΣΧΟΛΙΚΟ ΕΤΟΣ 2018-2019

στο πλαίσιο της δράσης του έργου MIS 5011051 , Υποστήριξη και Διαχείριση των Σχεδίων Δράσης του έργου "Μια Νέα Αρχή στα ΕΠΑΛ"



3.1 ΠΕΡΙΓΡΑΦΗ ΤΗΣ ΔΡΑΣΗΣ

Οι μαθητές σε συνεργασία με τους εκπαιδευτικούς θα δημιουργήσουν μία σειρά από έξυπνα συστήματα - αυτοματισμούς στο μάθημα «Ερευνητική Εργασία στην Τεχνολογία» που είναι δύο ώρες εβδομαδιαίως. Τα συστήματα αυτά θα χρησιμοποιούν έξυπνους μικροελεγκτές της σειράς Arduino. Το Arduino είναι μια ηλεκτρονική πλατφόρμα ανοικτού κώδικα και σχεδιασμού, που βασίζεται σε ευέλικτο και εύκολο στη χρήση υλικό και λογισμικό. Προορίζεται για καλλιτέχνες, σχεδιαστές, υλοποίηση χόμπι και δραστηριοτήτων, και γενικότερα για οποιονδήποτε ενδιαφέρεται να δημιουργήσει αλληλεπιδραστικά αντικείμενα ή περιβάλλοντα. Τελικός σκοπός είναι να εισαχθεί ο μαθητής με ευχάριστο τρόπο στα ψηφιακά κυκλώματα και τον προγραμματισμό

3.2 ΣΤΟΧΟΙ ΔΡΑΣΗΣ ΚΑΙ ΤΡΟΠΟΙ ΕΠΙΤΕΥΞΗΣ ΤΟΥΣ

Το Arduino είναι μια ηλεκτρονική πλατφόρμα ανοικτού κώδικα και σχεδιασμού, που βασίζεται σε ευέλικτο και εύκολο στη χρήση υλικό και λογισμικό. Υπάρχουν ελευθερα στο διαδίκτυο πολλά εγχειρίδια που μπορούν να χρησιμοποιηθούν σε μια ερευνητική εργασία στο λύκειο , όπου οι μαθητές θα μπορούν να ψάχνουν περισσότερες πληροφορίες σε κάθε μάθημα σχετικά με αυτά που καλούνται να υλοποιήσουν και τελικά να προτείνουν μόνοι τους λύσεις στα προβλήματα και κυκλώματα με τα οποία ασχολούνται. Επίσης, με τη γνώση και εμπειρία που θα προκύψει από τα φύλλα εργασίας μπορεί μετά να υλοποιηθεί μια μεγαλύτερη κατασκευή που θα συγκεντρώνει αρκετά από τα επιμέρους τμήματα που έχουν ήδη τύχει επεξεργασίας και προγραμματισμού από την τάξη στην οποία απευθυνόμαστε. Για το λόγο αυτό προτείνεται και μια σύνθετη κατασκευή ανάλογα με την εμπειρία κάθε μαθησιακής ομάδας (π.χ. αυτοκίνητο που κινείται μόνο του, ελέγχοντας για εμπόδια).

Εκτός από τη βασική έκδοση του περιβάλλοντος Arduino IDE, υπάρχει και μια παραλλαγμένη έκδοση του Scratch*, η οποία μπορεί να χρησιμοποιηθεί για να γράψουμε προγράμματα για το Arduino, η S4A - Scratch For Arduino*, η οποία επίσης είναι ανοικτού κώδικα και δωρεάν. Το πλεονέκτημα της έκδοσης αυτής είναι ο οπτικός προγραμματισμός (blocks όπως στο Scratch) σε σχέση με το γράψιμο εντολών στο κλασσικό περιβάλλον. Παρόμοιας λογικής είναι και το ArduBlock*, το οποίο επίσης χρησιμοποιεί οπτικό προγραμματισμό μέσω έτοιμων blocks για τον προγραμματισμό του. Ακόμα, υπάρχουν οπτικές εκδόσεις στο διαδίκτυο (web περιβάλλοντα), όπως το BlocklyDuino* ή το ArduinoMio*. Ο προγραμματισμός λοιπόν του Arduino μπορεί να γίνεται και με το λογισμικό S4A (Scratch for Arduino)!

Για την παρούσα δράση λαμβάνοντας υπόψιν τον αριθμό των μαθητών 16 προτείνεται η δημιουργία τεσσάρων μαθησιακών ομάδων έτσι ώστε να έχουν όλοι οι μαθητές πρόσβαση στο υλικό και να μπορούν να δουλέψουν πρακτικά και πιο άνετα. Θα χρειαστεί η προμήθεια 4 σετ Arduino Uno R3 Starter / Basic Kit with Motors όπου θα γίνει και η πρώτη και πιο βασική προσέγγιση στον προγραμματιζόμενο ελεγκτή Arduino. Καθ' όλη την διάρκεια της δράσης οι μαθητές της Α τάξης θα έχουν την αμέριστη βοήθεια από τους μαθητές των μεγαλύτερων τάξεων Β-Γ των Τομέων της Πληροφορικής σε θέματα προγραμματισμού, του Μηχανολογικού τομέα σε κατασκευαστικά θέματα όπως: ίδιο-κατασκευασμένες θήκες από plexiGlass για την τοποθέτηση και λειτουργία των ελεγκτών. Οι εργαστηριακές ασκήσεις στα αντίστοιχα μαθήματα ειδικότητας των τάξεων Β-Γ μπορούν να τροποποιηθούν κατάλληλα έτσι ώστε να καλύψουν τις ανάγκες της δράσης χωρίς επίσης να αλλάζει ή να παρακωλύεται ο χαρακτήρας και η παρακολούθηση της διδακτικής τους ύλης. Θα υλοποιηθούν – κατασκευαστούν, τουλάχιστον τέσσερα, μεγάλα έξυπνα συστήματα όπως: Οπτικός διαχωριστής αντικειμένων, Ρομποτικό όχημα με απομακρυσμένο έλεγχο από κινητή συσκευή, Έξυπνη συσκευή κυλιόμενου μηνύματος, Ραντάρ με αισθητήρα υπερήχων, Αυτόματη κλειδαριά με ασύρματη κάρτα, Έξυπνος έλεγχος ηλεκτρικού φορτίου από κινητή συσκευή.

Τα έργα αυτά θα αποτελέσουν μία από τις βασικές θεματικές της ετήσιας έκθεσης με τα έργα των μαθητών του ΕΠΑΛ που θα πραγματοποιηθεί με σκοπό το άνοιγμα του ΕΠΑΛ στην κοινωνία και την επικοινωνία με φορείς ώστε να φανεί το καινοτόμο έργο που γίνεται στο ΕΠΑΛ. Πρόκειται για ομαδικά έργα σε τεχνολογία αιχμής που υποδεικνύουν ότι τα ΕΠΑΛ δίνουν γερές βάσεις για την τεχνολογία του μέλλοντος ειδικά στον αυτοματισμό και την ασύρματη επικοινωνία με κινητές συσκευές

3.4 ΠΑΡΑΔΟΤΕΑ ΤΗΣ ΔΡΑΣΗΣ

Τα παραδοτέα της δράσης είναι:

1. Νέα έξυπνα συστήματα
 - Κώδικας προγραμματισμού που αναπτύχθηκε
 - Φωτογραφικό υλικό από την εκτέλεση των ασκήσεων και υλοποίησης
 - Βίντεο της λειτουργίας των έξυπνων συσκευών.

1. Έξυπνη συσκευή κυλιόμενου μηνύματος

- **Φυλλάδιο παρουσίασης**
- **Βοηθητική εφαρμογή σε AppInventor2**
- **Κώδικας προγραμματισμού που αναπτύχθηκε**
 - **Κώδικας Arduino IDE**
 - **Κώδικας Βιβλιοθηκών Arduino IDE**
- **Φωτογραφικό υλικό από την εκτέλεση των ασκήσεων και υλοποίησης**
- **Βίντεο της λειτουργίας των έξυπνων συσκευών.**



«ΜΙΑ ΝΕΑ ΑΡΧΗ ΣΤΑ ΕΠΑΛ»

“Νέα” αρχή σε “Νέο” ΕΠΑΛ !

Νέα έξυπνα συστήματα



Οι μαθητές σε συνεργασία με τους εκπαιδευτικούς δημιούργησαν μία σειρά από έξυπνα συστήματα - αυτοματισμούς στο μάθημα «Ερευνητική Εργασία στην Τεχνολογία» και «Αρχές Προγραμματισμού Υπολογιστών». Τα συστήματα αυτά χρησιμοποιούν έξυπνους μικροελεγκτές της σειράς Arduino. Το Arduino είναι μια ηλεκτρονική πλατφόρμα ανοικτού κώδικα και σχεδιασμού, που βασίζεται σε ευέλικτο και εύκολο στη χρήση υλικό και λογισμικό. Προορίζεται για καλλιτέχνες, σχεδιαστές, υλοποίηση χόμπι και δραστηριοτήτων, και γενικότερα για οποιονδήποτε ενδιαφέρεται να δημιουργήσει αλληλεπιδραστικά αντικείμενα ή περιβάλλοντα. Τελικός σκοπός είναι να εισαχθεί ο μαθητής με ευχάριστο τρόπο στα ψηφιακά κυκλώματα και τον προγραμματισμό. Πρόκειται για ομαδικά έργα σε τεχνολογία αιχμής, που υποδεικνύουν ότι τα ΕΠΑΛ δίνουν γερές βάσεις για την τεχνολογία του μέλλοντος, ειδικά στον αυτοματισμό και την ασύρματη επικοινωνία με κινητές συσκευές

#1 Έξυπνη συσκευή κυλιόμενου μηνύματος

Η έξυπνη αυτή συσκευή εμφανίζει ένα κυλιόμενο μήνυμα σε έναν φωτεινό πίνακα LED.

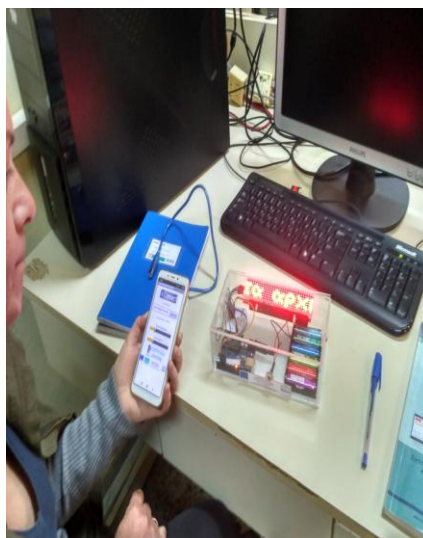
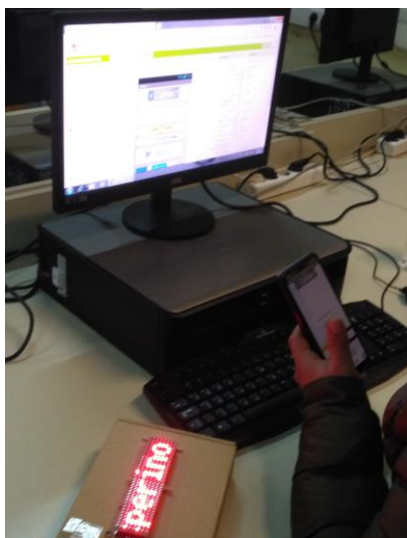


Χρησιμοποιήθηκαν:

Μικροελεγκτής ARDUINO, Άρθρωμα πίνακας LED (max7219-4x4x8), Άρθρωμα επικοινωνίας Bluetooth(HC06)

Αναπτύχθηκαν εφαρμογές κώδικα:

Κώδικας ελέγχου ARDUINO (S4A, ARDUINO IDE, C++), Βοηθητική εφαρμογή κινητής συσκευής ANDROID (appInventor)



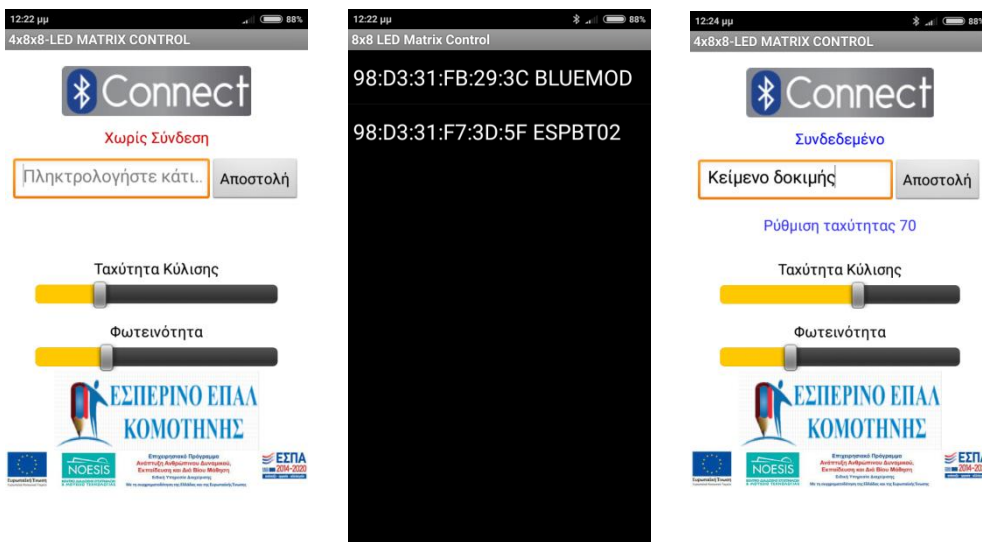
Ομάδα Εργασίας Εσπερινού ΕΠΑΛ Κομοτηνής:

ΧΑΚΑΝ Α., ΟΛΥΜΠΙΟΣ Α., ΧΑΚΑΝ Α., ΠΑΥΛΟΣ Γ., ΟΥΡΑΝΙΑ Μ., ΕΛΕΝΗ Τ, ΔΗΜΗΤΡΗΣ Δ..



#1B Βοηθητική εφαρμογή έξυπνης συσκευής κυλιόμενου μηνύματος

Η εφαρμογή αυτή (για συσκευές με λειτουργικό android), αναπτύχθηκε με την βοήθεια των μαθητών του τομέα Πληροφορικής του Εσπερινού ΕΠΑΛ Κομοτηνής, στο πλαίσιο της δράσης του έργου MIS 5011051 «Υποστήριξη και Διαχείριση των Σχεδίων Δράσης του έργου "Μια Νέα Αρχή στα ΕΠΑΛ"». Η εφαρμογή συνδέεται μέσω δικτύου bluetooth με τον ελεγκτή arduino.



Μπορεί να αποσταλεί μικρό κείμενο (ελληνικό ή/και λατινικό αλφάβητο) στον ελεγκτή και να εμφανιστεί κυλιόμενο στον συνδεδεμένο πίνακα 4x8x8 LED. Μπορεί να ρυθμιστεί η ένταση της φωτεινότητας καθώς και η ταχύτητα κύλισης του κειμένου.

Μπορείτε να κατεβάσετε την εφαρμογή και να δοκιμάσετε την έξυπνη συσκευή από το Google Play Store:

https://play.google.com/store/apps/details?id=appinventor.ai_anon16410901270233.Arduino_8x8_LED_Matrix_Control



Ομάδα Εργασίας Εσπερινού ΕΠΑΛ Κομοτηνής:

ΧΑΚΑΝ Α., ΟΛΥΜΠΙΟΣ Α., ΧΑΚΑΝ Α., ΠΑΥΛΟΣ Γ., ΟΥΡΑΝΙΑ Μ., ΕΛΕΝΗ Τ. ΔΗΜΗΤΡΗΣ Δ

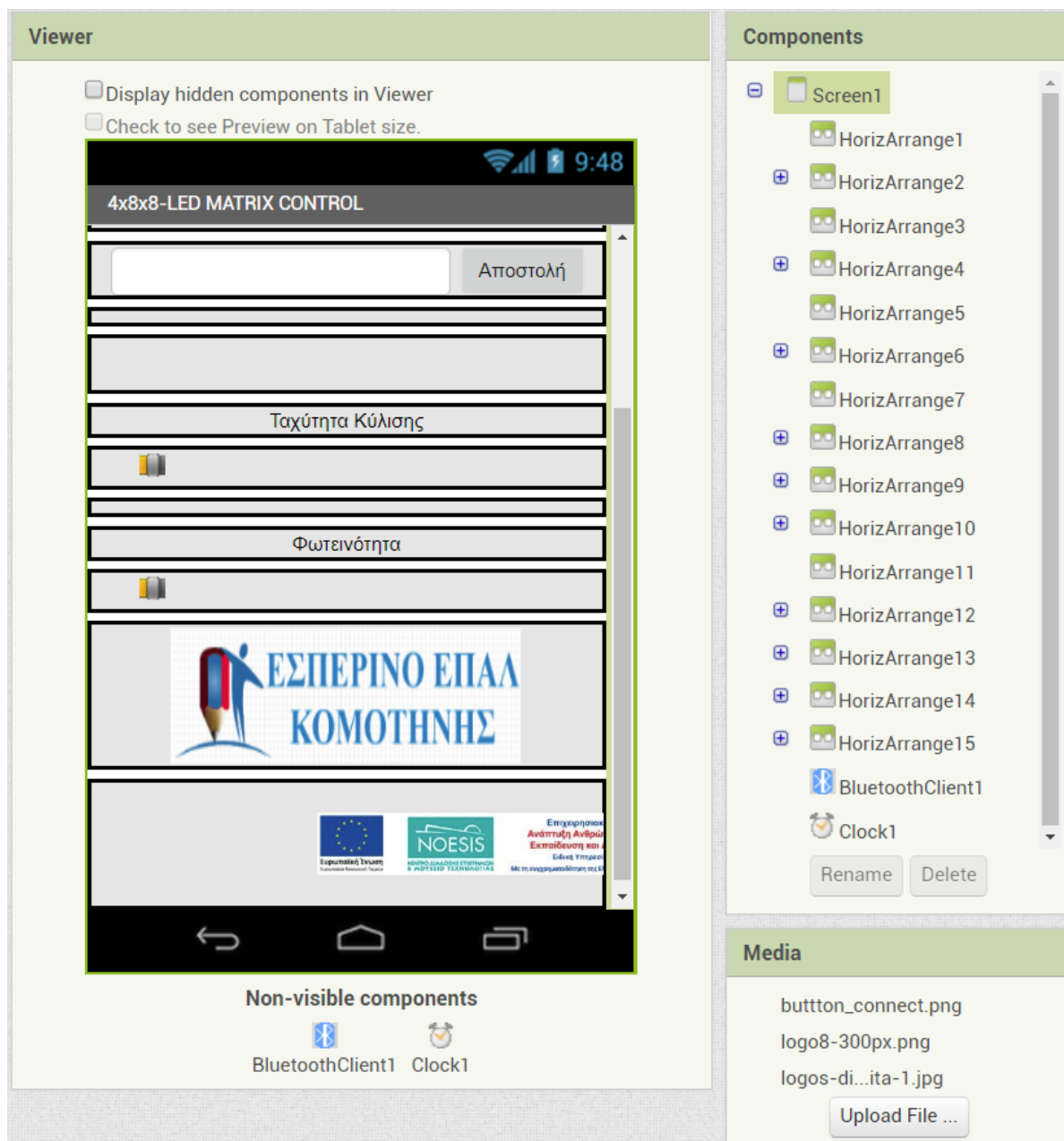
Βοηθητική εφαρμογή σε AppInventor2

Κώδικας προγραμματισμού που αναπτύχτηκε:

Βοηθητική εφαρμογή αποστολής μηνυμάτων

Για την υλοποίηση της βοηθητικής εφαρμογής μηνυμάτων, θα χρησιμοποιηθεί η πλατφόρμα **appinventor**, που έχει χρησιμοποιηθεί πολλές φορές και για άλλες εφαρμογές και σε άλλα μαθήματα.

Σύντομη περιγραφή: Για την εφαρμογή θα χρησιμοποιηθεί μια μόνο οθόνη με τα αντικείμενα που φαίνονται στις παρακάτω εικόνες, η οποία θα συνδέεται με το άρθρωμα bluetooth του arduino και θα αποστέλλει το κείμενο που θέλουμε να εμφανίζεται, καθώς και τις ρυθμίσεις φωτεινότητας και ταχύτητας κύλισης.



Έξυπνη συσκευή κυλιόμενου μηνύματος

| | | Components TYPE | Components NAME | Properties | Value |
|----|----------------|-----------------|----------------------|---|---|
| 1 | | | Screen1 | AppName Icon | 8x8 LED Matrix Control ledmatrixico.png |
| 2 | User Interface | ListPicker | BluetoothList | Image | button_connect.png |
| 3 | User Interface | Label | Connected | Text | Χωρίς Σύνδεση |
| 4 | User Interface | TextBox | TextBox1 | Width Hint | 65 percent Πληκτρολογήστε κάτι.. |
| 5 | User Interface | Button | Send_Button | Text | Αποστολή |
| 6 | User Interface | Label | Message_Status | | |
| 7 | User Interface | Slider | SliderScrollingSpeed | MinValue MaxValue ThumbPosition | 0 120 30 |
| 8 | User Interface | Slider | SliderBrightness | MinValue MaxValue ThumbPosition | 1 15 5 |
| 9 | User Interface | Image | Image1 | Height Width Picture | 84 pixel 217 pixel logo8-300px.png |
| 10 | User Interface | Image | Image1 | Height Width Picture | Automatic Automatic logos-dimosiotita-1.jpg |
| 11 | Connectivity | BluetoothClient | BluetoothClient1 | CharacterEncoding | ISO-8859-7 |
| 12 | Sensors | Clock | Clock1 | TimerAlwaysFires TimerEnabled TimerInterval | checked checked 1000 |

Θα χρησιμοποιηθούν οι εικόνες:

Εικόνα

Αρχείο

| | |
|---|-------------------------|
|  | ledmatrixico.png |
|  | button_connect.png |
|  | logo8-300px.png |
|  | logos-dimosiotita-1.jpg |

Έξυνη συσκευή κυλιόμενου μηνύματος

Ο κώδικας των blocks φαίνεται παρακάτω:

```
when BluetoothList .BeforePicking
do
  set BluetoothList . Elements to BluetoothClient1 . AddressesAndNames

when BluetoothList .AfterPicking
do
  if call BluetoothClient1 .Connect address BluetoothList . Selection
  then
    set BluetoothList . Elements to BluetoothClient1 . AddressesAndNames

when Send_Button .Click
do
  call BluetoothClient1 .SendText text " 1 "
  call BluetoothClient1 .SendText text join TextBox . Text of component TextBox1 " "
  set Message_Status . Text to TextBox . Text of component TextBox1
  set Message_Status . TextColor to green

when Clock1 .Timer
do
  if BluetoothClient1 . IsConnected
  then
    set Connected . Text to " Συνδεδεμένο "
    set Connected . TextColor to blue
  if not BluetoothClient1 . IsConnected
  then
    set Connected . Text to " Χωρίς Σύνδεση "
    set Connected . TextColor to red

when SliderBrightness .PositionChanged thumbPosition
do
  call BluetoothClient1 .SendText text " 3 "
  call BluetoothClient1 .SendText text join round get thumbPosition " "
  set Message_Status . Text to join " Ρύθμιση φωτεινότητας " round get thumbPosition
  set Message_Status . TextColor to orange

when SliderScrollingSpeed .PositionChanged thumbPosition
do
  call BluetoothClient1 .SendText text " 2 "
  call BluetoothClient1 .SendText text join round get thumbPosition " "
  set Message_Status . Text to join " Ρύθμιση ταχύτητας " round get thumbPosition
  set Message_Status . TextColor to blue
```

Κώδικας Arduino IDE

```
/*
  4x8x8 LED Matrix MAX7219 Scrolling Text
  Android Control via Bluetooth

  By Esperino EPAL KOMOTINIS

  Implemented during classes of project
  "MNAE" Mia Nea Arxi sta Epal
  KOMOTINI - GREECE 1-3-2019

  Based on a work of Dejan Nedelkovski, www.HowToMechatronics.com
  Based on the following library:
  GitHub | riyas-org/max7219 https://github.com/riyas-org/max7219

  With the help of the font builder of MD_MAX72XX-master library,

  Max4x8Matrix modified library by dimdimakop
*/
#include <Max4x8Matrix.h>
#include <SoftwareSerial.h>
#include <avr/pgmspace.h>

PROGMEM const unsigned char CH[] = {
  3, 8, B00000000, B00000000, B00000000, B00000000, B00000000, // space
  1, 8, B01011111, B00000000, B00000000, B00000000, B00000000, // !
  3, 8, B00000011, B00000000, B00000011, B00000000, B00000000, // "
  5, 8, B00010100, B00111110, B00010100, B00111110, B00010100, // #
  4, 8, B00100100, B01101010, B00101011, B00010010, B00000000, // $
  5, 8, B01100011, B00010011, B00001000, B01100100, B01100011, // %
  5, 8, B00110110, B01001001, B01010110, B00100000, B01010000, // &
  1, 8, B00000011, B00000000, B00000000, B00000000, B00000000, // '
  3, 8, B00011100, B00100010, B01000001, B00000000, B00000000, // (
  3, 8, B01000001, B00100010, B00011100, B00000000, B00000000, // )
  5, 8, B00101000, B00011000, B00001110, B00011000, B00101000, // *
  5, 8, B00001000, B00001000, B00111110, B00001000, B00001000, // +
  2, 8, B10110000, B01110000, B00000000, B00000000, B00000000, // ,
  4, 8, B00001000, B00001000, B00001000, B00001000, B00000000, // -
  2, 8, B01100000, B01100000, B00000000, B00000000, B00000000, // .
  4, 8, B01100000, B00011000, B00000110, B00000001, B00000000, // /
  4, 8, B00111110, B01000001, B01000001, B00111110, B00000000, // 0
  3, 8, B01000010, B01111111, B01000000, B00000000, B00000000, // 1
  4, 8, B01100010, B01010001, B01001001, B01000110, B00000000, // 2
  4, 8, B00100010, B01000001, B01000101, B00110110, B00000000, // 3
  4, 8, B00011000, B00010100, B00010010, B01111111, B00000000, // 4
  4, 8, B00100111, B01000101, B01000101, B00111001, B00000000, // 5
  4, 8, B00111110, B01001001, B01001001, B00110000, B00000000, // 6
  4, 8, B01100001, B00010001, B00001001, B00000111, B00000000, // 7
  4, 8, B00110110, B01001001, B01001001, B00110110, B00000000, // 8
  4, 8, B00000110, B01001001, B01001001, B00111110, B00000000, // 9
  2, 8, B01010000, B00000000, B00000000, B00000000, B00000000, // :
  2, 8, B10000000, B01010000, B00000000, B00000000, B00000000, // ;
  3, 8, B00010000, B00101000, B01000100, B00000000, B00000000, // <
  3, 8, B00010100, B00010100, B00010100, B00000000, B00000000, // =
  3, 8, B01000100, B00101000, B00010000, B00000000, B00000000, // >
  4, 8, B00000010, B01011001, B00001001, B00000110, B00000000, // ?
  5, 8, B00111110, B01001001, B01010101, B01011101, B00001110, // @
  4, 8, B01111110, B00010001, B00010001, B01111110, B00000000, // A
  4, 8, B01111111, B01001001, B01001001, B00110110, B00000000, // B
  4, 8, B00111110, B01000001, B01000001, B00100010, B00000000, // C
  4, 8, B01111111, B01000001, B01000001, B00111110, B00000000, // D
  4, 8, B01111111, B01001001, B01001001, B01000001, B00000000, // E
  4, 8, B01111111, B00001001, B00001001, B00000001, B00000000, // F
  4, 8, B00111110, B01000001, B01001001, B01111010, B00000000, // G
  4, 8, B01111111, B00001000, B00001000, B01111111, B00000000, // H
  3, 8, B01000001, B01111111, B01000001, B00000000, B00000000, // I
  4, 8, B00110000, B01000000, B01000001, B00111111, B00000000, // J
  4, 8, B01111111, B00001000, B00010100, B01100011, B00000000, // K
  4, 8, B01111111, B01000000, B01000000, B01000000, B00000000, // L
  5, 8, B01111111, B00000010, B00001100, B00000010, B01111111, // M
  5, 8, B01111111, B00000100, B00001000, B00010000, B01111111, // N
  4, 8, B00111110, B01000001, B01000001, B00111110, B00000000, // O
  4, 8, B01111111, B00001001, B00001001, B00000110, B00000000, // P
  4, 8, B00111110, B01000001, B01000001, B10111110, B00000000, // Q
  4, 8, B01111111, B00001001, B00001001, B01110110, B00000000, // R
  4, 8, B01000110, B01001001, B01001001, B00110010, B00000000, // S
  5, 8, B00000001, B00000001, B01111111, B00000001, B00000001, // T
  4, 8, B00111111, B01000000, B01000000, B00111111, B00000000, // U
  5, 8, B00001111, B00110000, B01000000, B00110000, B00001111, // V
  5, 8, B00111111, B01000000, B00111000, B01000000, B00111111, // W
  5, 8, B01100011, B00010100, B00001000, B00010100, B01100011, // X
  5, 8, B00000111, B00001000, B01110000, B00001000, B00000111, // Y
  4, 8, B01100001, B01010001, B01001001, B01000111, B00000000, // Z
  2, 8, B01111111, B01000001, B00000000, B00000000, B00000000, // [
```

Έξυνη συσκευή κλιόμενου μηνύματος

```
4, 8, B00000001, B00000110, B00011000, B01100000, B00000000, // \ backslash
2, 8, B01000001, B01111111, B00000000, B00000000, // ]
3, 8, B00000010, B00000001, B00000010, B00000000, B00000000, // hat
4, 8, B01000000, B01000000, B01000000, B01000000, B00000000, // ^
2, 8, B00000001, B00000010, B00000000, B00000000, B00000000, // v
4, 8, B00100000, B00101000, B01010100, B01111000, B00000000, // a
4, 8, B01111111, B01000100, B01000100, B00111000, B00000000, // b
4, 8, B00111000, B01000100, B01000100, B00101000, B00000000, // c
4, 8, B00111000, B01000100, B01000100, B01111111, B00000000, // d
4, 8, B00111000, B01010100, B01010100, B00011000, B00000000, // e
3, 8, B00000100, B01111110, B00000101, B00000000, B00000000, // f
4, 8, B10011000, B01000100, B01000100, B01111000, B00000000, // g
4, 8, B01111111, B00000100, B00000100, B01111000, B00000000, // h
3, 8, B01000100, B01111101, B01000000, B00000000, B00000000, // i
4, 8, B01000000, B10000000, B10000100, B01111101, B00000000, // j
4, 8, B01111111, B00010000, B00101000, B01000100, B00000000, // k
3, 8, B01000001, B01111111, B01000000, B00000000, B00000000, // l
5, 8, B01111100, B00000100, B01111100, B00000100, B01111000, // m
4, 8, B01111100, B00000100, B00000100, B01111000, B00000000, // n
4, 8, B00111000, B01000100, B01000100, B00111000, B00000000, // o
4, 8, B11111100, B00100100, B00100100, B00011000, B00000000, // p
4, 8, B00011000, B00100100, B00100100, B11111100, B00000000, // q
4, 8, B01111100, B00001000, B00000100, B00000100, B00000000, // r
4, 8, B01001000, B01010100, B01010100, B00100100, B00000000, // s
3, 8, B00000100, B00111111, B01000100, B00000000, B00000000, // t
4, 8, B00111100, B01000000, B01000000, B01111100, B00000000, // u
5, 8, B00011100, B00100000, B01000000, B00100000, B00011100, // v
5, 8, B00111100, B01000000, B00111100, B01000000, B00111100, // w
5, 8, B01000100, B00101000, B00010000, B00101000, B01000100, // x
4, 8, B10011100, B01000000, B10100000, B01111100, B00000000, // y
3, 8, B01100100, B01010100, B01001100, B00000000, B00000000, // z
3, 8, B00001000, B00110110, B01000001, B00000000, B00000000, // {
1, 8, B01111111, B00000000, B00000000, B00000000, B00000000, // |
3, 8, B01000001, B00110110, B00001000, B00000000, B00000000, // }
4, 8, B00001000, B00000100, B00001000, B00000100, B00000000, // ~
//start of ISO-8859-7 encoding of Greek Alphabet
2, 8, B00000000, B00000000, B00000000, B00000000, B00000000, //Non-breaking space
3, 8, B00001110, B00001101, B00001101, B00000000, B00000000, //161
3, 8, B00001011, B00001011, B00000111, B00000000, B00000000, //162
5, 8, B01001000, B01111110, B01001001, B01000001, B01100110, //Pound sign
5, 8, B00101000, B01111110, B01010100, B01010100, B01000010, //euro
5, 8, B00101011, B00101111, B11111100, B00101111, B00101011, //drm
1, 8, B01110111, B00000000, B00000000, B00000000, B00000000, //|
4, 8, B01100110, B10001001, B10010101, B01101010, B00000000, //Section sign
3, 8, B00000001, B00000000, B00000001, B00000000, B00000000, //Spacing diaeresis
5, 8, B00111110, B01011101, B01010101, B01000001, B00111110, //Copyright
3, 8, B00001101, B00001101, B00001111, B00000000, B00000000, //Feminine Ordinal Ind.
5, 8, B00001000, B00010100, B00101010, B00010100, B00100010, //<<
5, 8, B00001000, B00001000, B00001000, B00001000, B00111000, //Not sign
0, 8, B00000000, B00000000, B00000000, B00000000, B00000000, //Soft Hyphen
5, 8, B00111110, B01111111, B01001011, B01110111, B00111110, //Registered Trademark
5, 8, B00000001, B00000001, B00000001, B00000001, B00000001, //Spacing Macron Overline
3, 8, B00000010, B00000010, B00000010, B00000000, B00000000, //Degree
5, 8, B01000100, B01000100, B01011111, B01000100, B01000100, //+/-
3, 8, B00011001, B00010101, B00010011, B00000000, B00000000, //Superscript 2
3, 8, B00010001, B00010101, B00011111, B00000000, B00000000, //Superscript 3
2, 8, B00000010, B00000001, B00000000, B00000000, B00000000, //Acute accent
3, 8, B00000100, B00000011, B00000100, B00000000, B00000000, //dialytika tonos
5, 8, B01111101, B000001010, B000001001, B000001010, B01111100, //ALPHA tonos
2, 8, B00011000, B00011000, B00000000, B00000000, B00000000, //Middle Dot
5, 8, B01111111, B01001001, B01001001, B01001001, B01000001, //EPSILON
5, 8, B01111111, B00001001, B00001000, B00001000, B01111111, //ETA tonos
4, 8, B00000011, B01000000, B01111111, B01000001, B00000000, //IOTA tonos
5, 8, B00100010, B00010100, B00101010, B00010100, B00001000, //>>
5, 8, B00111101, B01000010, B01000001, B01000001, B00111110, //OMICRON tonos
5, 8, B00101111, B00010000, B11001000, B01010100, B01110101, //1/2
5, 8, B00000001, B00000110, B00001000, B01110000, B00001000, //UPSILON tonos
5, 8, B01011110, B01100001, B00000001, B01100001, B01011110, //OMEGA tonos
5, 8, B00000010, B00111101, B01000010, B01000000, B00100000, //iota dialytika tonos
5, 8, B01111100, B00001010, B00001001, B00001010, B01111100, //ALPHA
5, 8, B01111111, B01001001, B01001001, B01001001, B00110110, //BETA
5, 8, B01111111, B00000001, B00000001, B00000001, B00000011, //GAMMA
5, 8, B01111100, B01000010, B01000001, B01000010, B01111100, //DELTA
5, 8, B01111111, B01001001, B01001001, B01001001, B01000001, //EPSILON
5, 8, B01100011, B01010001, B01001001, B01001001, B01000101, B01100011, //ZETA
5, 8, B01111111, B00001000, B00001000, B00001000, B01111111, //ETA
5, 8, B00111110, B01001001, B01001001, B01001001, B00111110, //THETA
3, 8, B01000001, B01111111, B01000001, B00000000, B00000000, //IOTA
5, 8, B01111111, B00001000, B00001000, B00101000, B00100001, //KAPPA
5, 8, B01111100, B00000010, B00000001, B00000010, B01111100, //LAMBDA
5, 8, B01111111, B00000010, B00001100, B00000010, B01111111, //MU
5, 8, B01111111, B00000010, B00000100, B00001000, B01111111, //NU
5, 8, B01100011, B00001001, B01001001, B01001001, B01100011, //XI
5, 8, B00111110, B01000001, B01000001, B01000001, B00111110, //OMICRON
5, 8, B00000001, B01111111, B00000001, B01111111, B00000001, //PI
```

Έξυπνη συσκευή κυλιόμενου μηνύματος

```
5,8,B01111111,B00001001,B00001001,B00001001,B00000110,//RHO
2,8,B00000000,B00000000,B00000000,B00000000,B00000000, //
5,8,B01100011,B01010101,B01001001,B01000001,B01100011, //SIGMA
5,8,B00000011,B00000001,B01111111,B00000001,B00000011, //TAU
5,8,B00000111,B00001000,B01110000,B00001000,B00000111, //UPSILON
5,8,B00001000,B01010101,B01111111,B01010101,B00001000, //PHI
5,8,B01100011,B00010100,B00001000,B00010100,B01100011, //CHI
5,8,B00000111,B00001000,B01111111,B00001000,B00000111, //PSI
5,8,B01011110,B01100001,B00000001,B01100001,B01011110, //OMEGA
3,8,B01000101,B01111100,B01000101,B00000000,B00000000, //IOTA dialytika
5,8,B00001100,B00010001,B01100000,B00010001,B00001100, //UPSILON dialytika
5,8,B00111000,B01000100,B01001001,B00110000,B01001000, //alpha tonos
4,8,B00101000,B01010100,B01010101,B01000100,B00000000, //epsilon tonos
4,8,B01111100,B00001000,B00000101,B11111000,B00000000, //eta tonos
3,8,B00111101,B01000000,B01000000,B00000000,B00000000, //iota tonos
5,8,B00111101,B01000000,B01000011,B01000000,B00111101, //upsilon dialytika tonos
5,8,B00111000,B01000100,B01001000,B00110000,B01001000, //alpha
4,8,B11111000,B01010100,B01010100,B01010100,B00101000,B00000000, //beta
5,8,B00000010,B00110100,B01001000,B00110100,B00000010, //gamma
4,8,B00110000,B01001010,B01001101,B00110001,B00000000, //delta
4,8,B00101000,B01010100,B01010100,B01000100,B00000000, //epsilon
4,8,B00100010,B00101000,B01010100,B10101010,B11000110,B00000000, //zeta
4,8,B01111100,B00001000,B00000100,B11111000,B00000000, //eta
5,8,B00111000,B01010100,B01010100,B01010100,B00111000, //theta
3,8,B00111100,B01000000,B01000000,B00000000,B00000000, //iota
4,8,B01111100,B00010000,B00101000,B01000100,B00000000, //kappa
4,8,B01111010,B00001001,B00010001,B01111110,B00000000, //lambda
5,8,B11111100,B00100000,B00100000,B00010000,B01111100, //mu
5,8,B00001100,B00110000,B01000000,B00110000,B00001100, //nu
5,8,B00010100,B00101010,B00101010,B01010100,B01000000, //xi
5,8,B00111000,B01000100,B01000100,B01000100,B00111000, //omicron
5,8,B00000100,B01111100,B00000100,B01111100,B00000100, //pi
4,8,B11111000,B00100100,B00100100,B00011000,B00000000, //rho
5,8,B01001000,B01010100,B01010100,B01010100,B00100100, //sigma (final)
5,8,B00111000,B01000100,B01000100,B00111100,B00000100, //sigma
5,8,B00000100,B00000100,B00111100,B01000100,B00000100, //tau
5,8,B00111100,B01000000,B01000000,B01000000,B00111100, //upsilon
5,8,B00001100,B00010000,B01111100,B00010010,B00001100, //phi
5,8,B01000100,B00101000,B00010000,B00101000,B01000100, //chi
5,8,B00001100,B00010000,B01111100,B00010000,B00001100, //psi
5,8,B00111000,B01000100,B00110000,B01000100,B00111000, //omega
5,8,B00000001,B00111100,B01000001,B01000000,B00100000, //iota dialytika
5,8,B00111100,B01000001,B01000000,B01000001,B00111100, //upsilon dialytika
5,8,B00111000,B01000100,B01000101,B01000100,B00111000, //omicron tonos
5,8,B00111100,B01000000,B01000001,B01000000,B00111100, //upsilon tonos
5,8,B00111000,B01000100,B00110001,B01000100,B00111000, //omega tonos
};

int dIn = 7; // DIN pin of MAX7219 module
int clk = 6; // CLK pin of MAX7219 module
int cs = 5; // CS pin of MAX7219 module

int maxInUse = 4; // Number of MAX7219's connected

Max4x8Matrix m(dIn, cs, clk, maxInUse);
SoftwareSerial Bluetooth(3, 4); // Bluetooth RX | TX

byte buffer[10];
byte incomebyte;
byte incomeunbyte;
int scrollSpeed = 100;
byte text[100] = "Esperino EPAL Komotinis "; // Initial text message
byte brightness = 15;
int count = 0;
char indicator;

void setup() {
    m.init(); // MAX7219 initialization
    m.setIntensity(brightness); // initial led matrix intensity, 0-15
    Serial.begin(9600); // Default communication rate of the Bluetooth module
    Bluetooth.begin(9600); // Default communication rate of the Bluetooth module
}

void loop() {
    // Printing the text
    printStringWithShift(text, scrollSpeed);

    if (Bluetooth.available()) { // Checks whether data is coming from the serial port
        indicator = Bluetooth.read(); // Starts reading the serial port, the first byte from the incoming data
        // If we have pressed the "Send" button from the Android App, clear the previous text
        if (indicator == '1') {
            for (int i = 0; i < 100; i++) {
                text[i] = 0;
                m.clear();
            }
        }
    }
}
```

```

// Read the whole data/string coming from the phone and put it into text[] array.
while (Bluetooth.available()) {
    incomebyte = Bluetooth.read();
    incomeunbyte=incomebyte;
    Serial.print(incomeunbyte);
    Serial.print(",");
    text[count] = incomebyte;
    count++;
}
count = 0;
Serial.print("\n end of bluetooth trans\n");
}
// Adjusting the Scrolling Speed
else if (indicator == '2') {

    String sS = Bluetooth.readString();
    Serial.print("\n read speed to "+String(sS)+"\n");
    Serial.print("\n read speed len "+String(sS.length()));
    long ldot=sS.lastIndexOf('.');
    long pldot=sS.lastIndexOf('.',ldot-1);
    sS=sS.substring(pldot+2, ldot);

    Serial.print("\n calc speed ldot "+String(ldot)+" and pldot "+String(pldot)+" mystr "+sS);
    scrollSpeed = 130 - sS.toInt(); // Milliseconds, subtraction because lower value means higher scrolling
speed
    Serial.print("\n set speed to "+String(scrollSpeed)+"\n");
}
// Adjusting the brightness
else if (indicator == '3') {
    String sB = Bluetooth.readString();
    Serial.print("\n read brightness to "+String(sB));
    Serial.print("\n read brightness len "+String(sB.length()));
    long ldot=sB.lastIndexOf('.');
    long pldot=sB.lastIndexOf('.',ldot-1);
    sB=sB.substring(pldot+2, ldot);
    Serial.print("\n calc brightness ldot "+String(ldot)+" and pldot "+String(pldot)+" mystr "+sB);
    brightness = sB.toInt();
    m.setIntensity(brightness);
    Serial.print("\n set brightness to "+String(brightness)+"\n");
}
}
}

void printCharWithShift(byte c, int shift_speed) {
    if (c < 32) return;
    if (c > 126 and c < 160) return;
    if (c < 127) c -= 32;
    if (c > 159) c -= 65;
    memcpy_P(buffer, CH + 7 * c, 7);
    m.writeSprite(32, 0, buffer);
    m.setColumn(32 + buffer[0], 0);

    for (int i = 0; i < buffer[0] + 1; i++)
    {
        delay(shift_speed);
        m.shiftLeft(false, false);
    }
}

void printStringWithShift(byte* s, int shift_speed) {
    while (*s != 0) {
        printCharWithShift(*s, shift_speed);
        s++;
    }
}

void printString(char* s)
{
    int col = 0;
    while (*s != 0)
    {
        if (*s < 32) continue;
        char c = *s - 32;
        memcpy_P(buffer, CH + 7 * c, 7);
        m.writeSprite(col, 0, buffer);
        m.setColumn(col + buffer[0], 0);
        col += buffer[0] + 1;
        s++;
    }
}
}

```

Κώδικας Βιβλιοθηκών Arduino IDE

```

/*
 * MaxMatrix
 * Version 1.0 Feb 2013
 * Copyright 2013 Oscar Kin-Chung Au
 */

#ifndef Max4x8Matrix_H
#define _Max4x8Matrix_H_

#include "Arduino.h"

#define max7219_reg_noop          0x00
#define max7219_reg_digit0       0x01
#define max7219_reg_digit1       0x02
#define max7219_reg_digit2       0x03
#define max7219_reg_digit3       0x04
#define max7219_reg_digit4       0x05
#define max7219_reg_digit5       0x06
#define max7219_reg_digit6       0x07
#define max7219_reg_digit7       0x08
#define max7219_reg_decodeMode   0x09
#define max7219_reg_intensity    0x0a
#define max7219_reg_scanLimit    0x0b
#define max7219_reg_shutdown     0x0c
#define max7219_reg_displayTest  0x0f

class Max4x8Matrix
{
private:
    byte data;
    byte load;
    byte clock;
    byte num;
    byte buffer[80];
    byte outbuffer[80];

    void reload();

public:
    Max4x8Matrix(byte data, byte load, byte clock, byte num);

    void init();
    void clear();
    void setCommand(byte command, byte value);
    void setIntensity(byte intensity);
    void setColumn(byte col, byte value);
    void setColumnAll(byte col, byte value);
    void setDot(byte col, byte row, byte value);
    void writeSprite(int x, int y, const byte* sprite);

    void shiftLeft(bool rotate = false, bool fill_zero = true);
    void shiftRight(bool rotate = false, bool fill_zero = true);
    void shiftUp(bool rotate = false);
    void shiftDown(bool rotate = false);
};

#endif

/*
 * Max4x8Matrix
 * Version 1.3 Nov 2019
 * Copyright 2013 Oscar Kin-Chung Au + 2019 dimdimakop
 * Based on Max4x8Matrix of Oscar Kin-Chung Au
 * modified by dimdimakop 2019 to work with 4x8x8panel module
 */

#include "Arduino.h"
#include "Max4x8Matrix.h"

Max4x8Matrix::Max4x8Matrix(byte _data, byte _load, byte _clock, byte _num)
{
    data = _data;
    load = _load;
    clock = _clock;
    num = _num;
    for (int i=0; i<80; i++){
        buffer[i] = 0;
        outbuffer[i] = 0;
    }
}

```

```
void Max4x8Matrix::init()
{
    pinMode(data, OUTPUT);
    pinMode(clock, OUTPUT);
    pinMode(load, OUTPUT);
    digitalWrite(clock, HIGH);

    setCommand(max7219_reg_scanLimit, 0x07);
    setCommand(max7219_reg_decodeMode, 0x00); // using an led matrix (not digits)
    setCommand(max7219_reg_shutdown, 0x01); // not in shutdown mode
    setCommand(max7219_reg_displayTest, 0x00); // no display test
    // empty registers, turn all LEDs off
    clear();
    setIntensity(0x0f); // the first 0x0f is the value you can set
}

void Max4x8Matrix::setIntensity(byte intensity)
{
    setCommand(max7219_reg_intensity, intensity);
}

void Max4x8Matrix::clear()
{
    for (int i=0; i<80; i++){
        buffer[i] = 0;
        outbuffer[i] = 0;
    }
    reload();
}

void Max4x8Matrix::setCommand(byte command, byte value)
{
    digitalWrite(load, LOW);
    for (int i=0; i<num; i++)
    {
        shiftOut(data, clock, MSBFIRST, command);
        shiftOut(data, clock, MSBFIRST, value);
    }
    digitalWrite(load, LOW);
    digitalWrite(load, HIGH);
}

void Max4x8Matrix::writeSprite(int x, int y, const byte* sprite)
{
    int w = sprite[0];
    int h = sprite[1];
    if (h == 8 && y == 0){
        for (int i=0; i<w; i++)
        {
            int c = x + i;
            if (c>=0 && c<80)
                buffer[c] = sprite[i+2];
        }
        reload();
    }
}

void Max4x8Matrix::reload()
{
    byte ccol;
    for (int N=0; N<num; N++){
        for (int j=0; j<8; j++){
            ccol=buffer[N*8+j];
            for (int i=0; i<8; i++){
                bitWrite(outbuffer[N*8+i], 7-j, bitRead(ccol, i));
            }
        }
        for (int i=0; i<8; i++)
        {
            int col = 24+i;
            digitalWrite(load, LOW);
            for (int j=0; j<num; j++)
            {
                shiftOut(data, clock, MSBFIRST, i + 1); //address of column
                shiftOut(data, clock, MSBFIRST, outbuffer[col]); //val for column

                col -= 8;
            }
            digitalWrite(load, LOW);
            digitalWrite(load, HIGH);
        }
    }
}
```

```
void Max4x8Matrix::setColumn(byte col, byte value)
{
    buffer[col] = value;
}

void Max4x8Matrix::setColumnAll(byte col, byte value)
{
    digitalWrite(load, LOW);
    for (int i=0; i<num; i++)
    {
        shiftOut(data, clock, MSBFIRST, col + 1);
        shiftOut(data, clock, MSBFIRST, value);

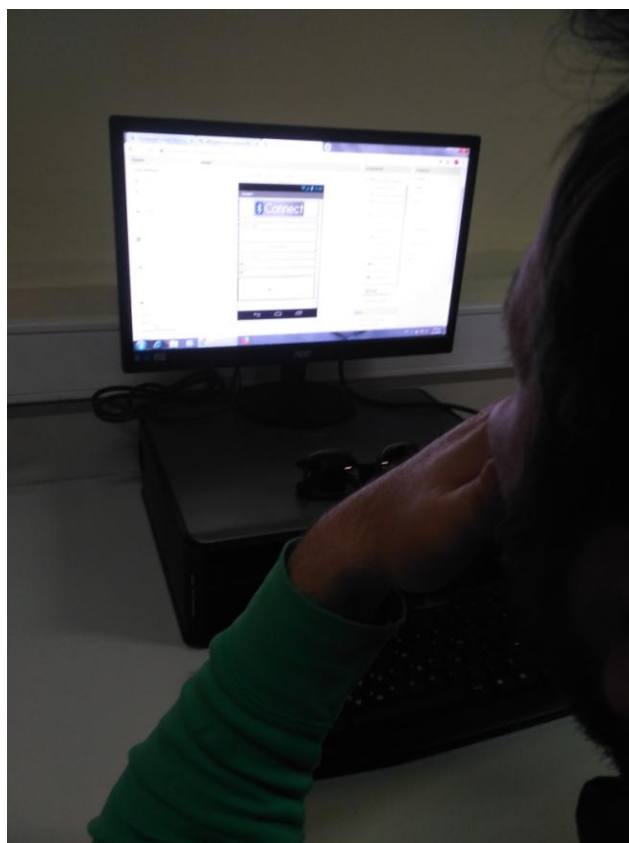
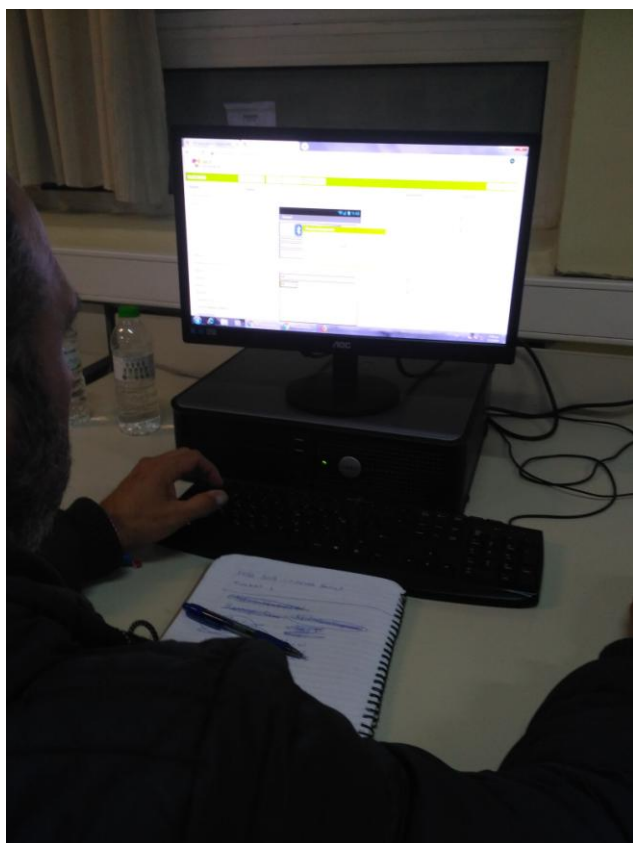
        buffer[col * i] = value;
    }
    digitalWrite(load, LOW);
    digitalWrite(load, HIGH);
}

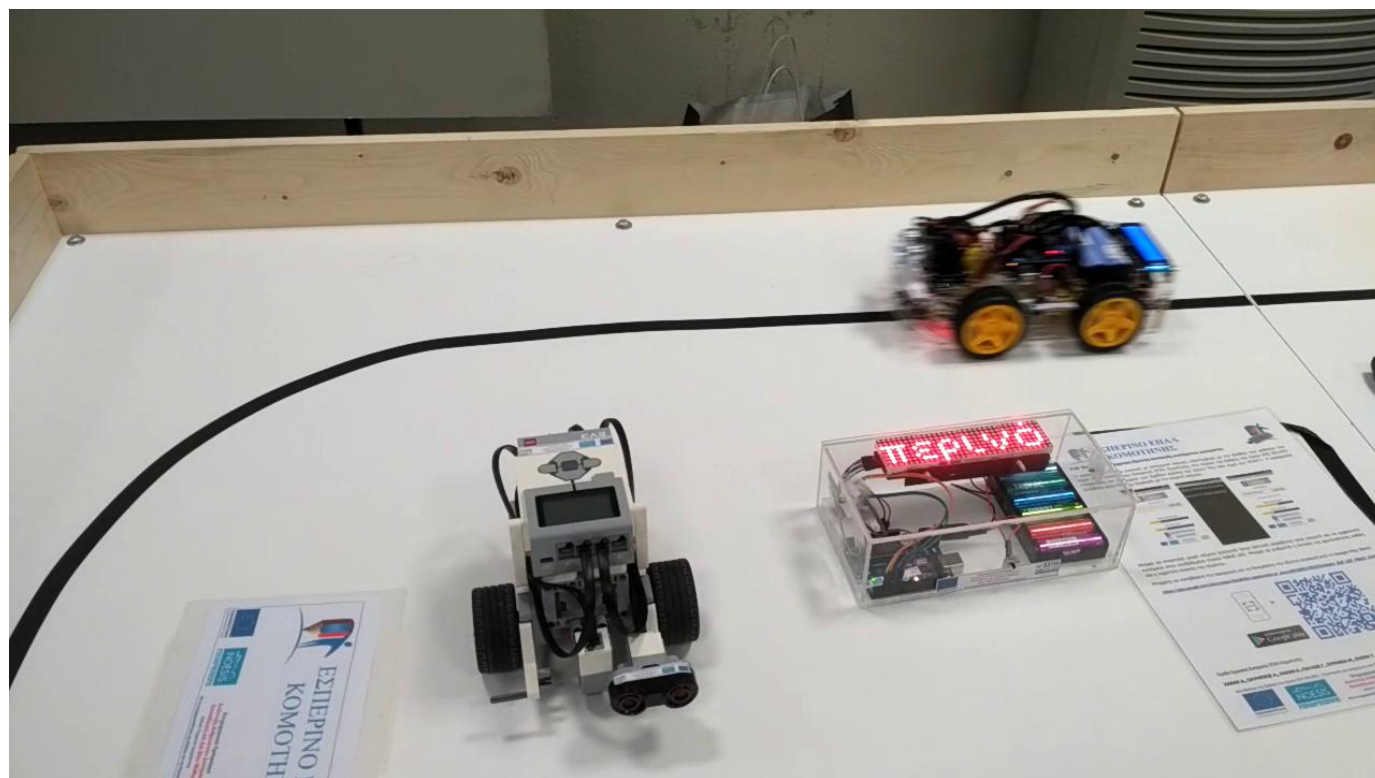
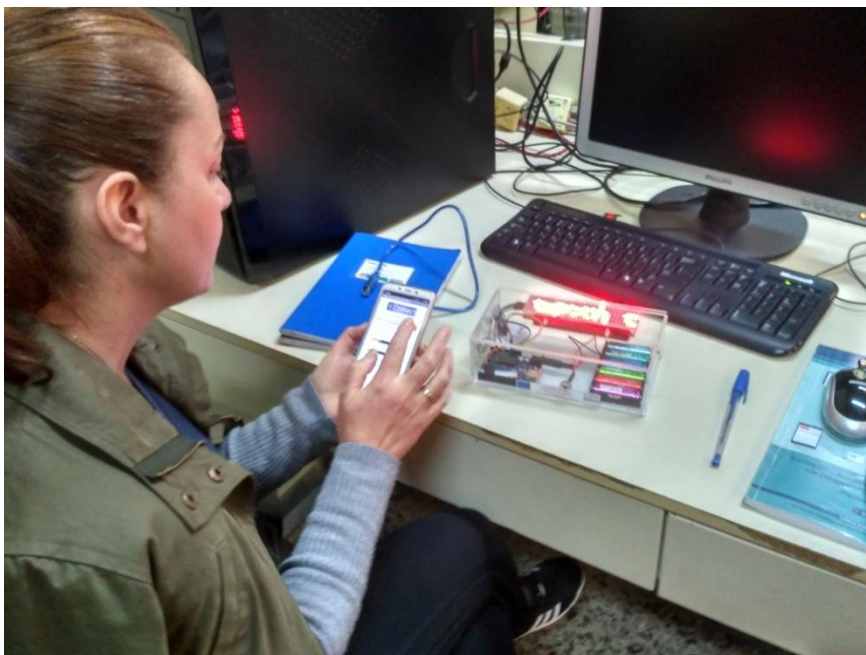
void Max4x8Matrix::shiftLeft(bool rotate, bool fill_zero)
{
    byte old = buffer[0];
    int i;
    for (i=0; i<79; i++)
    {
        buffer[i] = buffer[i+1];
        if (rotate) buffer[num*8-1] = old;
        else if (fill_zero) buffer[num*8-1] = 0;
    }
    reload();
}

void Max4x8Matrix::shiftRight(bool rotate, bool fill_zero)
{
    int last = num*8-1;
    byte old = buffer[last];
    int i;
    for (i=79; i>0; i--)
    {
        buffer[i] = buffer[i-1];
        if (rotate) buffer[0] = old;
        else if (fill_zero) buffer[0] = 0;
    }
    reload();
}

void Max4x8Matrix::shiftUp(bool rotate)
{
    for (int i=0; i<num*8; i++)
    {
        bool b = buffer[i] & 1;
        buffer[i] >>= 1;
        if (rotate) bitWrite(buffer[i], 7, b);
    }
    reload();
}

void Max4x8Matrix::shiftDown(bool rotate)
{
    for (int i=0; i<num*8; i++)
    {
        bool b = buffer[i] & 128;
        buffer[i] <<= 1;
        if (rotate) bitWrite(buffer[i], 0, b);
    }
    reload();
}
```



Βίντεο της λειτουργίας των έξυπνων συσκευών

<https://www.youtube.com/watch?v=29qRMqQxcws>

2. Έξυπνη συσκευή ελέγχου συσκευών ισχύος (με ρελέ)

- **Φυλλάδιο παρουσίασης**
- **Βοηθητική εφαρμογή σε AppInventor2**
- **Κώδικας προγραμματισμού που αναπτύχθηκε**
 - **Κώδικας Arduino IDE**
- **Φωτογραφικό υλικό από την εκτέλεση των ασκήσεων και υλοποίησης**
- **Βίντεο της λειτουργίας των έξυπνων συσκευών.**



«ΜΙΑ ΝΕΑ ΑΡΧΗ ΣΤΑ ΕΠΑΛ»

“Νέα” αρχή σε “Νέο” ΕΠΑΛ !

Νέα έξυπνα συστήματα



Οι μαθητές σε συνεργασία με τους εκπαιδευτικούς δημιούργησαν μία σειρά από έξυπνα συστήματα - αυτοματισμούς στο μάθημα «Ερευνητική Εργασία στην Τεχνολογία» και «Αρχές Προγραμματισμού Υπολογιστών». Τα συστήματα αυτά χρησιμοποιούν έξυπνους μικροελεγκτές της σειράς Arduino. Το Arduino είναι μια ηλεκτρονική πλατφόρμα ανοικτού κώδικα και σχεδιασμού, που βασίζεται σε ευέλικτο και εύκολο στη χρήση υλικό και λογισμικό. Προορίζεται για καλλιτέχνες, σχεδιαστές, υλοποίηση χόμπι και δραστηριοτήτων, και γενικότερα για οποιονδήποτε ενδιαφέρεται να δημιουργήσει αλληλεπιδραστικά αντικείμενα ή περιβάλλοντα. Τελικός σκοπός είναι να εισαχθεί ο μαθητής με ευχάριστο τρόπο στα ψηφιακά κυκλώματα και τον προγραμματισμό. Πρόκειται για ομαδικά έργα σε τεχνολογία αιχμής, που υποδεικνύουν ότι τα ΕΠΑΛ δίνουν γερές βάσεις για την τεχνολογία του μέλλοντος, ειδικά στον αυτοματισμό και την ασύρματη επικοινωνία με κινητές συσκευές

#2 Έξυπνη συσκευή ελέγχου συσκευών ισχύος (με ρελέ)

Η έξυπνη αυτή συσκευή επιτρέπει τον έλεγχο τεσσάρων ηλεκτρονόμων (ρελέ) που βρίσκονται σε ειδικό άρθρωμα (module) μέσω της επικοινωνίας του μικροελεγκτή Arduino με δίκτυο Bluetooth .



Χρησιμοποιήθηκαν:

Μικροελεγκτής ARDUINO, Άρθρωμα ηλεκτρονόμων (4relaymodule), Άρθρωμα επικοινωνίας Bluetooth(HC06)

Αναπτύχθηκαν εφαρμογές κώδικα:

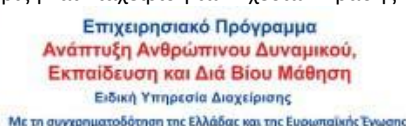
Κώδικας ελέγχου ARDUINO (S4A, ARDUINO IDE, C++), Βοηθητική εφαρμογή κινητής συσκευής ANDROID (appInventor)



Ομάδα Εργασίας Εσπερινού ΕΠΑΛ Κομοτηνής:

ΧΑΚΑΝ Α., ΟΛΥΜΠΙΟΣ Α., ΧΑΚΑΝ Α., ΠΑΥΛΟΣ Γ., ΟΥΡΑΝΙΑ Μ., ΕΛΕΝΗ Τ, ΔΗΜΗΤΡΗΣ Δ..

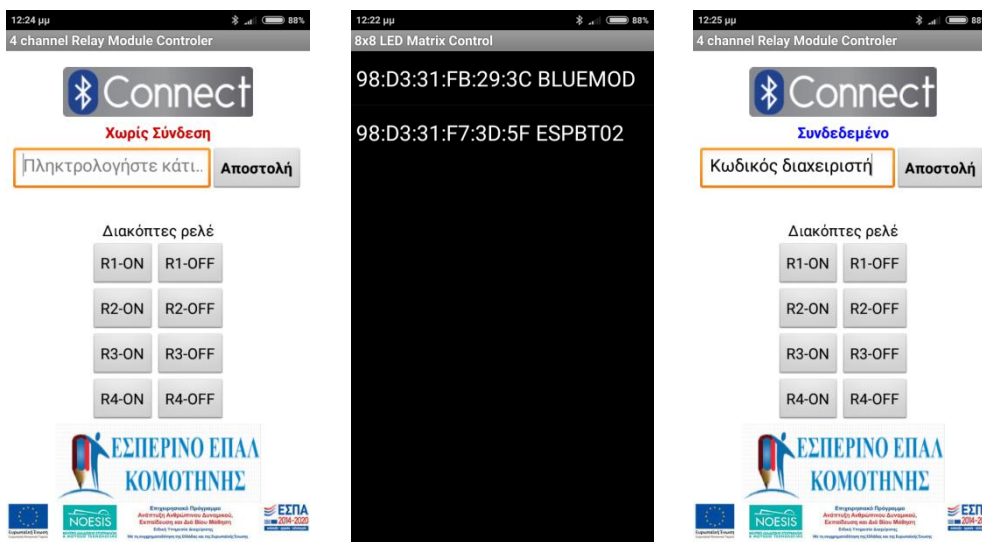
στο πλαίσιο της δράσης του έργου MIS 5011051, Υποστήριξη και Διαχείριση των Σχεδίων Δράσης του έργου "Μια Νέα Αρχή στα ΕΠΑΛ"





#2B Βοηθητική εφαρμογή έξυπνης συσκευής ελέγχου συσκευών ισχύος

Η εφαρμογή αυτή (για συσκευές με λειτουργικό android), αναπτύχθηκε με την βοήθεια των μαθητών του τομέα Πληροφορικής του Εσπερινού ΕΠΑΛ Κομοτηνής, στο πλαίσιο της δράσης του έργου MIS 5011051 «Υποστήριξη και Διαχείριση των Σχεδίων Δράσης του έργου "Μια Νέα Αρχή στα ΕΠΑΛ"». Η εφαρμογή συνδέεται μέσω δικτύου bluetooth με τον ελεγκτή arduino.



Μπορεί να ελέγχει την ενεργοποίηση-απενεργοποίηση και των τεσσάρων ηλεκτρονόμων του αρθρώματος ξεχωριστά. Μπορεί επίσης να αποστέλλει ειδικά μηνύματα ελέγχου, ασφάλειας και διαχείρισης.

Μπορείτε να κατεβάσετε την εφαρμογή και να δοκιμάσετε την έξυπνη συσκευή από το Google Play Store:

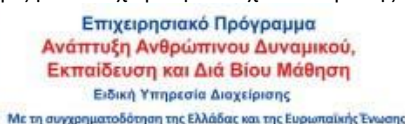
https://play.google.com/store/apps/details?id=appinventor.ai_anon16410901270233.Arduino_8x8_LED_Matrix_Control



Ομάδα Εργασίας Εσπερινού ΕΠΑΛ Κομοτηνής:

ΧΑΚΑΝ Α., ΟΛΥΜΠΙΟΣ Α., ΧΑΚΑΝ Α., ΠΑΥΛΟΣ Γ., ΟΥΡΑΝΙΑ Μ., ΕΛΕΝΗ Τ. ΔΗΜΗΤΡΗΣ Δ.

στο πλαίσιο της δράσης του έργου MIS 5011051 , Υποστήριξη και Διαχείριση των Σχεδίων Δράσης του έργου "Μια Νέα Αρχή στα ΕΠΑΛ"



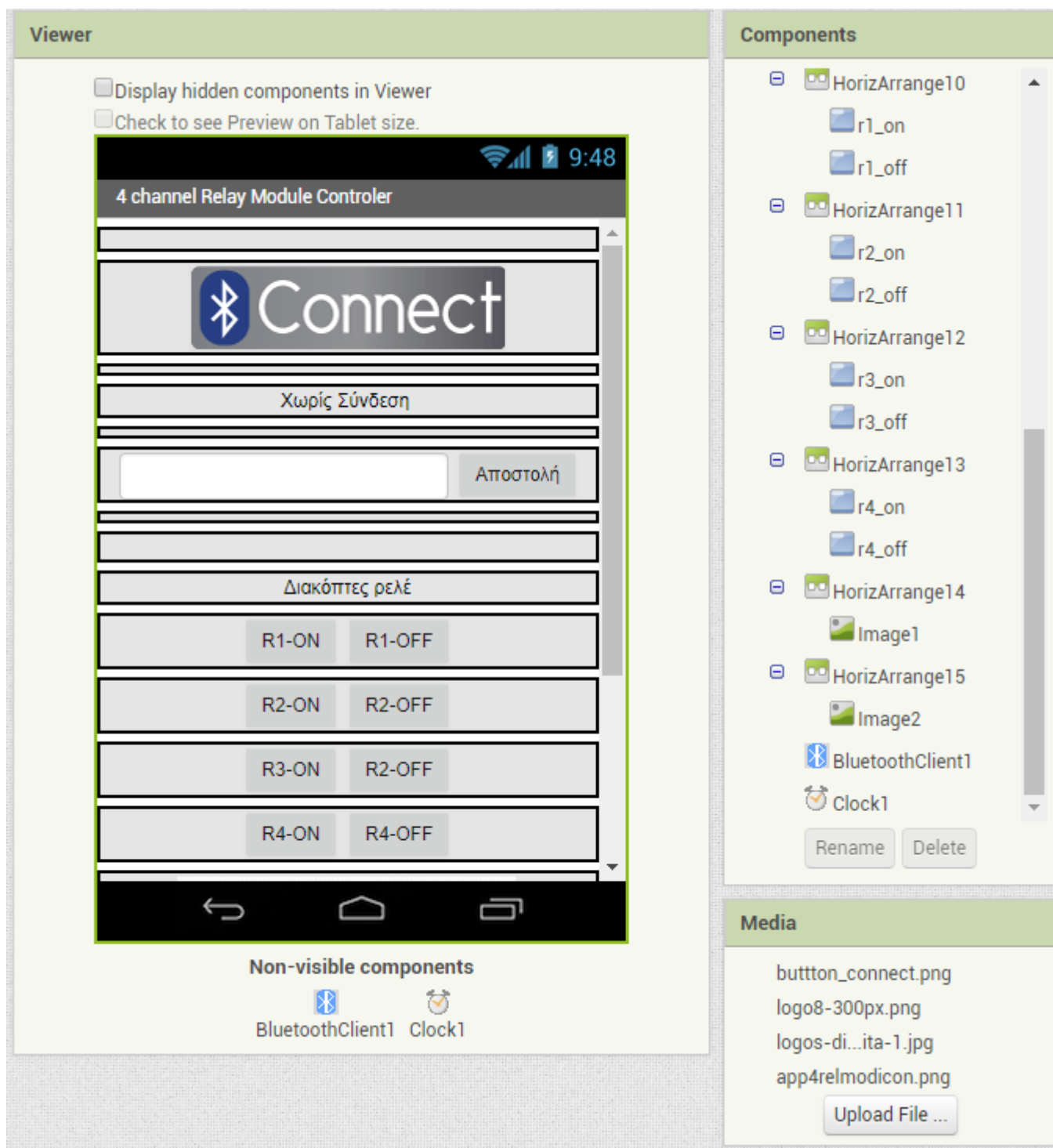
Βοηθητική εφαρμογή σε AppInventor2

Κώδικας προγραμματισμού που αναπτύχτηκε:

Βοηθητική εφαρμογή αποστολής μηνυμάτων

Για την υλοποίηση της βοηθητικής εφαρμογής μηνυμάτων, θα χρησιμοποιηθεί η πλατφόρμα **appinventor**, που έχει χρησιμοποιηθεί πολλές φορές και για άλλες εφαρμογές και σε άλλα μαθήματα.

Σύντομη περιγραφή: Για την εφαρμογή θα χρησιμοποιηθεί μια μόνο οθόνη με τα αντικείμενα που φαίνονται στις παρακάτω εικόνες, η οποία θα συνδέεται με το άρθρωμα bluetooth του arduino και θα αποστέλλει σε κείμενο τις εντολές ελέγχου των ηλεκτρονόμων.



Έξυπνη συσκευή ελέγχου συσκευών ισχύος (με ρελέ)

| | | Components TYPE | Components NAME | Properties | Value |
|----|----------------|-----------------|------------------|---|--|
| 1 | | | Screen1 | AppName Icon | 8x8 LED Matrix Control ledmatrixico.png |
| 2 | User Interface | ListPicker | BluetoothList | Image | button_connect.png |
| 3 | User Interface | Label | Connected | Text | Χωρίς Σύνδεση |
| 4 | User Interface | TextBox | TextBox1 | Width Hint | 65 percent Πληκτρολογήστε κάτι.. |
| 5 | User Interface | Button | Send_Button | Text | Αποστολή |
| 6 | User Interface | Label | Message_Status | Text | |
| 7 | User Interface | Label | RelaysLabel | Text | Διακόπτες ρελέ |
| 8 | User Interface | Button | r1_on | Text | R1-ON |
| 9 | User Interface | Button | r1_off | Text | R1-OFF |
| 10 | User Interface | Button | r2_on | Text | R2-ON |
| 11 | User Interface | Button | r2_off | Text | R2-OFF |
| 12 | User Interface | Button | r3_on | Text | R3-ON |
| 13 | User Interface | Button | r3_off | Text | R3-OFF |
| 14 | User Interface | Button | r4_on | Text | R4-ON |
| 15 | User Interface | Button | r4_off | Text | R4-OFF |
| 16 | Connectivity | BluetoothClient | BluetoothClient1 | CharacterEncoding | ISO-8859-7 |
| 17 | Sensors | Clock | Clock1 | TimerAlwaysFires TimerEnabled TimerInterval | checked checked 1000 |

Θα χρησιμοποιηθούν οι εικόνες:

Εικόνα

Αρχείο

| | |
|---|-------------------------|
|  | ledmatrixico.png |
|  | button_connect.png |
|  | logo8-300px.png |
|  | logos-dimosiotita-1.jpg |

Ο κώδικας των blocks φαίνεται παρακάτω:

The image displays a collection of code blocks for a Bluetooth device control system, organized into several functional groups:

- BluetoothList Initialization:**
 - BeforePicking:** A 'do' block that sets 'BluetoothList.Elements' to 'BluetoothClient1.AddressesAndNames'.
 - AfterPicking:** An 'if' block that calls 'BluetoothClient1.Connect' with 'BluetoothList.Selection' as the address. If successful, it sets 'BluetoothList.Elements' to 'BluetoothClient1.AddressesAndNames'.
- Send Button:** A 'when Send_Button.Click' block that:
 - Calls 'BluetoothClient1.SendText' with text '1'.
 - Calls 'BluetoothClient1.SendText' with text '2'.
 - Joins 'TextBox.Text' of component 'TextBox1'.
 - Sets 'Message_Status.Text' to 'TextBox.Text' of component 'TextBox1'.
 - Sets 'Message_Status.TextColor' to green.
- Timer (Clock1):** A 'when Clock1.Timer' block that:
 - Checks 'BluetoothClient1.IsConnected'. If true, sets 'Connected.Text' to 'Συνδεδεμένο' and 'Connected.TextColor' to blue.
 - Checks 'BluetoothClient1.IsConnected'. If false, sets 'Connected.Text' to 'Χωρίς σύνδεση' and 'Connected.TextColor' to red.
- Radio Buttons (r1-r4):** Eight 'when [radio].Click' blocks, each calling 'BluetoothClient1.SendText' with a specific text value:
 - r1_off: text '1'
 - r1_on: text '2'
 - r2_off: text '3'
 - r2_on: text '4'
 - r3_off: text '5'
 - r3_on: text '6'
 - r4_off: text '7'
 - r4_on: text '8'

Κώδικας Arduino IDE

```
/*

    Arduino 4 relays module control

    By Esperino EPAL KOMOTINIS

    Implemented during classes of project
    "MNAE" Mia Nea Arxi sta Epal
    KOMOTINI - GREECE 29-3-2019

NOTES!!!!

AT SIM800L
AT+IPR=9600 Set baud rate at 9,600 bps
AT+CBC Inquire charging status and remaining battery capacity
AT+CSQ Inquire GSM signal quality
AT+GSV Display product identification information
*/

#include <SoftwareSerial.h>
char indicator;
String bufstr = "";
String BTbufstr = "";
String bufsmsdata = "";
String lastsmsid = "";
String lastcallid = "";
int ringcount=0;
String callids[5]={"", "", "", "", ""};

//SoftwareSerial Bluetooth(10, 11); // Bluetooth RX | TX
SoftwareSerial GSMSerial(3, 4); // SIM800L RX | TX

void setup(){
    pinMode(6, OUTPUT); // sets the digital pin 6 as output RED
    pinMode(7, OUTPUT); // sets the digital pin 7 as output YELLOW
    pinMode(8, OUTPUT); // sets the digital pin 8 as output GREEN
    pinMode(9, OUTPUT); // sets the digital pin 9 as output BLUE

    Serial.begin(9600);
    // Bluetooth.begin(9600);
    GSMSerial.begin(9600);

    delay(2000); // give time to log on to network.
    //Bluetooth.print("AT\r");
    delay(9000);
    GSMSerial.print("AT+CLIP=1\r"); //CALLER ID ENABLE
    GSMSerial.print("AT+CMGF=1\r"); //TEXT MODE SMS ENABLE
    GSMSerial.print("AT+CNMI=2,2,0,0,0\r"); //SMS DELIVERY ENABLE
    digitalWrite(8, HIGH);
}

void loop() {

    if (GSMSerial.available()){
        while(GSMSerial.available()){
            bufstr = GSMSerial.readStringUntil('\n');

            //Serial.println("Get_it_from_GSM:"+bufstr+":endof("+bufstr.length()+")");
```

```

    if(bufstr.startsWith("+CLIP: \")){
        //handle clip
        lastcallid = bufstr.substring(bufstr.indexOf("+CLIP: \")+8,
bufstr.indexOf("+CLIP: \")+18);
        //Serial.println("lastcallid:"+lastcallid+":endof("+lastcallid.length()+")");
    }
    if(bufstr.startsWith("RING")){
        //handle RING
        ringcount++;
    }
    if(bufstr.startsWith("NO CARRIER")){
        //handle finished Call
        fincallhandlerFEST();
        //Serial.println("ringcount:"+String(ringcount)+":endof");
        //reset RING counter
        ringcount = 0 ;
    }
    if(bufstr.startsWith("+CMT: \")){
        //handle CMT and get next line that is SMS DATA
        lastsmsid = bufstr.substring(bufstr.indexOf("+CMT: \")+7,
bufstr.indexOf("+CMT: \")+20);
        //Serial.println("lastsmsid:"+lastsmsid+":endof("+lastsmsid.length()+")");
        while(GSMSerial.available()){
            bufsmsdata = GSMSerial.readStringUntil('\n');
        }

//Serial.println("bufsmsdata:"+bufsmsdata+":endof("+bufsmsdata.length()+")");
        //call smsdata handler
        finsmshandlerFEST();
    }

}

//Bluetooth

if (Serial.available()) {
    while(Serial.available()){
        BTbufstr = Serial.readStringUntil('\n');
        //Serial.println("Get from BT...:"+BTbufstr);
    }
    //Serial.println();

    if(BTbufstr.indexOf("R1ON")>-1){SetRelayPin(2);}
    if(BTbufstr.indexOf("R1OFF")>-1){SetRelayPin(1);}
    if(BTbufstr.indexOf("R2ON")>-1){SetRelayPin(4);}
    if(BTbufstr.indexOf("R2OFF")>-1){SetRelayPin(3);}
    if(BTbufstr.indexOf("R3ON")>-1){SetRelayPin(6);}
    if(BTbufstr.indexOf("R3OFF")>-1){SetRelayPin(5);}
    if(BTbufstr.indexOf("R4ON")>-1){SetRelayPin(8);}
    if(BTbufstr.indexOf("R4OFF")>-1){SetRelayPin(7);}

}

/*

if (Serial.available()){
    while(Serial.available()){
        //Bluetooth.write(Serial.read());
        GSMSerial.write(Serial.read());
    }
    //Bluetooth.println();
    GSMSerial.println();
    // Serial.println("Send to GSM...");
}
*/

```

```
}  
  
void SetRelayPin(int relayval) {  
    switch (relayval) {  
        case 1:  
            digitalWrite(6, LOW);  
            break;  
        case 2:  
            digitalWrite(6, HIGH);  
            break;  
        case 3:  
            digitalWrite(7, LOW);  
            break;  
        case 4:  
            digitalWrite(7, HIGH);  
            break;  
        case 5:  
            digitalWrite(8, LOW);  
            break;  
        case 6:  
            digitalWrite(8, HIGH);  
            break;  
        case 7:  
            digitalWrite(9, LOW);  
            break;  
        case 8:  
            digitalWrite(9, HIGH);  
            break;  
        default:  
            // statements  
            break;  
    }  
}  
  
}  
void fincallhandler(){  
    switch (ringcount-1) {  
        case 1:  
            if (lastcallid==callids[1]){ SetRelayPin(1);}  
  
            break;  
        case 2:  
            if (lastcallid==callids[1]){ SetRelayPin(2);}  
            break;  
        case 3:  
            if (lastcallid==callids[2]){ SetRelayPin(3);}  
            break;  
        case 4:  
            if (lastcallid==callids[2]){ SetRelayPin(4);}  
            break;  
        case 5:  
            if (lastcallid==callids[3]){ SetRelayPin(5);}  
            break;  
        case 6:  
            if (lastcallid==callids[3]){ SetRelayPin(6);}  
            break;  
        case 7:  
            if (lastcallid==callids[4]){ SetRelayPin(7);}  
            break;  
        case 8:  
            if (lastcallid==callids[4]){ SetRelayPin(8);}  
            break;  
        default:  
            // statements  
            break;  
    }  
}  
  
}
```

```
void fincallhandlerFEST() {

    if (ringcount > 3) {
        SetRelayPin(8);
    } else {
        SetRelayPin(7);
    }

}

void finsmshandler() {
    if (callids[1].indexOf(lastsmsid)) {
        if (bufsmsdata.indexOf("R1ON")) {SetRelayPin(1);}
        if (bufsmsdata.indexOf("R1OFF")) {SetRelayPin(2);}
    }
    if (callids[2].indexOf(lastsmsid)) {
        if (bufsmsdata.indexOf("R2ON")) {SetRelayPin(3);}
        if (bufsmsdata.indexOf("R2OFF")) {SetRelayPin(4);}
    }
    if (callids[3].indexOf(lastsmsid)) {
        if (bufsmsdata.indexOf("R3ON")) {SetRelayPin(5);}
        if (bufsmsdata.indexOf("R3OFF")) {SetRelayPin(6);}
    }
    if (callids[4].indexOf(lastsmsid)) {
        if (bufsmsdata.indexOf("R4ON")) {SetRelayPin(7);}
        if (bufsmsdata.indexOf("R4OFF")) {SetRelayPin(8);}
    }

}

void finsmshandlerFEST() {

    if (bufsmsdata.indexOf("R1ON")) {SetRelayPin(1);}
    if (bufsmsdata.indexOf("R1OFF")) {SetRelayPin(2);}

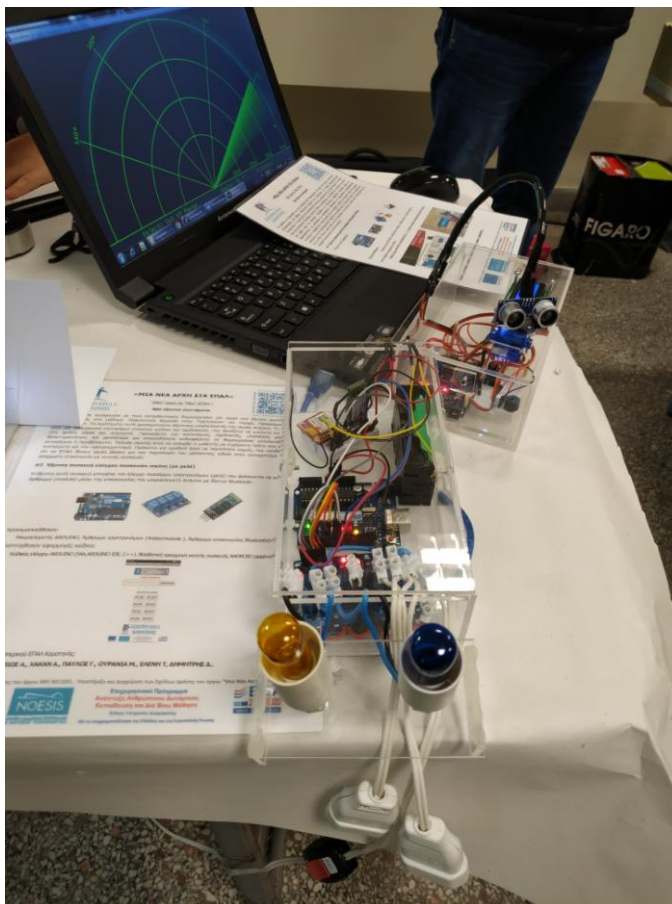
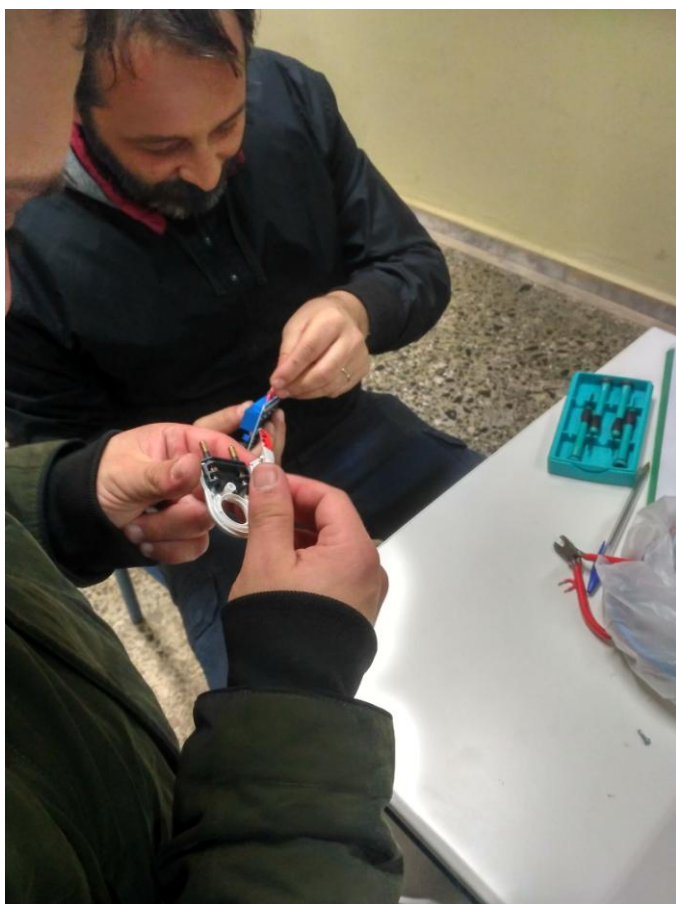
    if (bufsmsdata.indexOf("R2ON")) {SetRelayPin(3);}
    if (bufsmsdata.indexOf("R2OFF")) {SetRelayPin(4);}

    if (bufsmsdata.indexOf("R3ON")) {SetRelayPin(5);}
    if (bufsmsdata.indexOf("R3OFF")) {SetRelayPin(6);}

    if (bufsmsdata.indexOf("R4ON")) {SetRelayPin(7);}
    if (bufsmsdata.indexOf("R4OFF")) {SetRelayPin(8);}

}

}
```



Βίντεο της λειτουργίας των έξυπνων συσκευών

<https://www.youtube.com/watch?v=c83-2QKWlqo>

3. Έξυπνο αυτοκινούμενο όχημα (4WD Robot Car)

- **Φυλλάδιο παρουσίασης**
- **Βοηθητική εφαρμογή σε AppInventor2**
- **Κώδικας προγραμματισμού που αναπτύχτηκε**
 - **Κώδικας Arduino IDE**
- **Φωτογραφικό υλικό από την εκτέλεση των ασκήσεων και υλοποίησης**
- **Βίντεο της λειτουργίας των έξυπνων συσκευών.**



«ΜΙΑ ΝΕΑ ΑΡΧΗ ΣΤΑ ΕΠΑΛ»

“Νέα” αρχή σε “Νέο” ΕΠΑΛ !

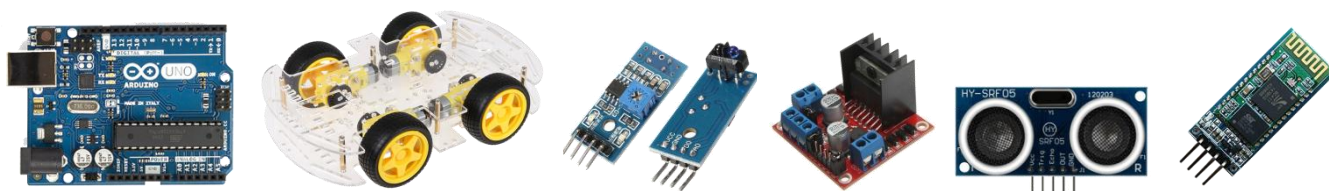
Νέα έξυπνα συστήματα



Οι μαθητές σε συνεργασία με τους εκπαιδευτικούς δημιούργησαν μία σειρά από έξυπνα συστήματα - αυτοματισμούς στο μάθημα «Ερευνητική Εργασία στην Τεχνολογία» και «Αρχές Προγραμματισμού Υπολογιστών». Τα συστήματα αυτά χρησιμοποιούν έξυπνους μικροελεγκτές της σειράς Arduino. Το Arduino είναι μια ηλεκτρονική πλατφόρμα ανοικτού κώδικα και σχεδιασμού, που βασίζεται σε ευέλικτο και εύκολο στη χρήση υλικό και λογισμικό. Προορίζεται για καλλιτέχνες, σχεδιαστές, υλοποίηση χόμπι και δραστηριοτήτων, και γενικότερα για οποιονδήποτε ενδιαφέρεται να δημιουργήσει αλληλεπιδραστικά αντικείμενα ή περιβάλλοντα. Τελικός σκοπός είναι να εισαχθεί ο μαθητής με ευχάριστο τρόπο στα ψηφιακά κυκλώματα και τον προγραμματισμό. Πρόκειται για ομαδικά έργα σε τεχνολογία αιχμής, που υποδεικνύουν ότι τα ΕΠΑΛ δίνουν γερές βάσεις για την τεχνολογία του μέλλοντος, ειδικά στον αυτοματισμό και την ασύρματη επικοινωνία με κινητές συσκευές

#3 Έξυπνο αυτοκινούμενο όχημα (4WD Robot Car)

Το έξυπνο αυτοκινούμενο όχημα μπορεί να κινηθεί με τέσσερις διαφορετικούς τρόπους ελέγχου :1) Έλεγχο με απλό τηλεχειριστήριο τηλεόρασης, 2) έλεγχο από εφαρμογή κινητού και σύνδεση Bluetooth , 3) αυτόματη οδήγηση με αισθητήριο υπερήχων, 4) αυτόματη παρακολούθηση ίχνους.



Χρησιμοποιήθηκαν:

Μικροελεγκτής ARDUINO, Άρθρωμα επικοινωνίας Bluetooth(HC06), Αισθητήριο υπερήχων, Αισθητήριο Υπέρυθρων , Κιτ οχήματος με κινητήρες

Αναπτύχθηκαν εφαρμογές κώδικα:

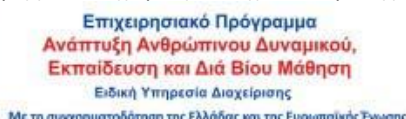
Κώδικας ελέγχου ARDUINO (S4A, ARDUINO IDE, C++), Βοηθητική εφαρμογή κινητής συσκευής ANDROID (appInventor)



Ομάδα Εργασίας Εσπερινού ΕΠΑΛ Κομοτηνής:

ΧΑΚΑΝ Α., ΟΛΥΜΠΙΟΣ Α., ΧΑΚΑΝ Α., ΠΑΥΛΟΣ Γ., ΟΥΡΑΝΙΑ Μ., ΕΛΕΝΗ Τ, ΔΗΜΗΤΡΗΣ Δ

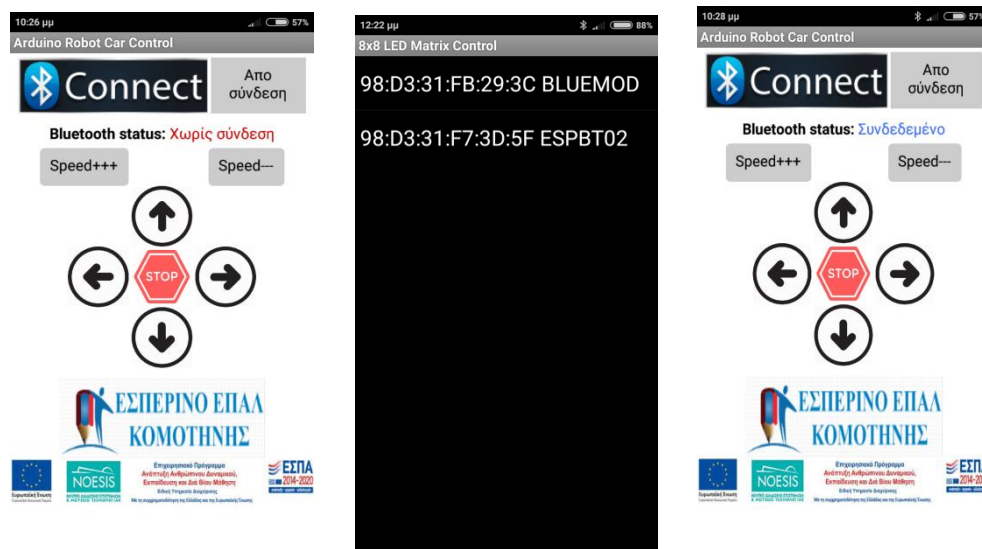
στο πλαίσιο της δράσης του έργου MIS 5011051 , Υποστήριξη και Διαχείριση των Σχεδίων Δράσης του έργου "Μια Νέα Αρχή στα ΕΠΑΛ"





#3B Βοηθητική εφαρμογή αυτοκινούμενου οχήματος (4WD Robot Car)

Η εφαρμογή αυτή (για συσκευές με λειτουργικό android), αναπτύχθηκε με την βοήθεια των μαθητών του τομέα Πληροφορικής του Εσπερινού ΕΠΑΛ Κομοτηνής, στο πλαίσιο της δράσης του έργου MIS 5011051 «Υποστήριξη και Διαχείριση των Σχεδίων Δράσης του έργου "Μια Νέα Αρχή στα ΕΠΑΛ"». Η εφαρμογή συνδέεται μέσω δικτύου bluetooth με τον ελεγκτή arduino.



Μπορεί να ελέγχει την οδήγηση του έξυπνου οχήματος με τα κατάλληλα πλήκτρα.

Μπορείτε να κατεβάσετε την εφαρμογή και να δοκιμάσετε την έξυπνη συσκευή από το Google Play Store:

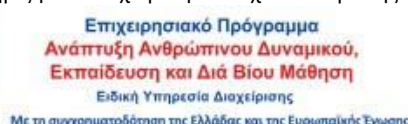
https://play.google.com/store/apps/details?id=appinventor.ai_anon16410901270233.RobotCarDrive



Ομάδα Εργασίας Εσπερινού ΕΠΑΛ Κομοτηνής:

ΧΑΚΑΝ Α., ΟΛΥΜΠΙΟΣ Α., ΧΑΚΑΝ Α., ΠΑΥΛΟΣ Γ., ΟΥΡΑΝΙΑ Μ., ΕΛΕΝΗ Τ. ΔΗΜΗΤΡΗΣ Δ.

στο πλαίσιο της δράσης του έργου MIS 5011051, Υποστήριξη και Διαχείριση των Σχεδίων Δράσης του έργου "Μια Νέα Αρχή στα ΕΠΑΛ"



Κώδικας Arduino IDE

```
/*
  Arduino 4WD Robot car

  By Esperino EPAL KOMOTINIS

  Implemented during classes of project
  "MNAE" Mia Nea Arxi sta Epal
  KOMOTINI - GREECE 12-3-2019

*/
#include <LiquidCrystal_I2C.h> //including libraries of I2C-LCD1602 liquid crystal
#include <Wire.h> //including libraries of I2C
#include <IRremote.h> //including libraries of remote control
#define RECV_PIN 12 //pin 12 of IR remoter control receiver
IRrecv irrecv(RECV_PIN); //defining pin 12 of IR remoter control
decode_results results; //cache of decode of IR remoter control
#define IR_Go 0x9986485A //going forward
#define IR_Back 0x1C6A919C //going backward
#define IR_Left 0x4EBA0422 //turning left
#define IR_Right 0x5EA08104 //turning right
#define IR_Stop 0x6AF21320 //stop
#define IR_Servo_L 0x51CEC000 //motor turning left (SCAN)
#define IR_Servo_R 0xD380803A //motor turning right
#define IR_Speed_UP 0x1E6BA628 //increasing speed (SKIP)
#define IR_Speed_DOWN 0xBE14D11A //decreasing speed
#define IR_Tracking_Mode 0xB8BAFCA0 //line detect (4)
#define IR_Self_Control 0x72C8CD04 //ultrasonic distance detecting (2)
#define IR_IR_Control 0xDB2CD85E //(1)
#define IR_Bluetooth_Control 0xBFA12A36 //(3)
#define IR_ESC 0x22CF7E40 //quitting from remote control (PAUSE)
////////////////////////////////////
#define SensorLeft 6 //sensor left pin of line tracking module
#define SensorMiddle 9 //sensor middle pin of line tracking module
#define SensorRight 11 //sensor right pin of line tracking module
unsigned char SL; //state of left sensor of line tracking module
unsigned char SM; //state of middle sensor of line tracking module
unsigned char SR; //state of right sensor of line tracking module
int inputPin=A0; // ultrasonic module ECHO to A0
int outputPin=A1; // ultrasonic module TRIG to A1
unsigned char Bluetooth_val; // ultrasonic module TRIG to A1
LiquidCrystal_I2C lcd(0x3f,16,2); //defining liquid crystal
#define Lpwm_pin 5 //pin of controlling speed---- ENA of motor driver board
#define Rpwm_pin 10 //pin of controlling speed---- ENB of motor driver board
int pinLB=4; //pin of controlling turning---- IN1 of motor driver board
int pinLF=2; //pin of controlling turning---- IN2 of motor driver board
int pinRB=8; //pin of controlling turning---- IN3 of motor driver board
int pinRF=7; //pin of controlling turning---- IN4 of motor driver board
unsigned char Lpwm_val = 100; //initialized left wheel speed at 150
unsigned char Rpwm_val = 100; //initialized right wheel speed at 150
int Car_state=0; //the working state of car
int servopin=3; //defining digital port pin 3, connecting to signal line of servo
motor
//int myangle; //defining variable of angle
int pulsewidth; //defining variable of pulse width
int ServoAngle=95; //initialized angle of motor at 95° ServoAngle
void Sensor_IO_Config() //IO initialized function of three line tracking , all setting at
input
{
  pinMode(SensorLeft,INPUT);
  pinMode(SensorMiddle,INPUT);
  pinMode(SensorRight,INPUT);
}
void Sensor_Scan(void) //function of reading-in signal of line tracking module
{
  SL = digitalRead(SensorLeft);
  SM = digitalRead(SensorMiddle);
  SR = digitalRead(SensorRight);
}
void servopulse(int servopin,int myangle) //defining a function of pulse
{
  pulsewidth=(myangle*11)+500; //converting angle into pulse width value at 500-2480
  digitalWrite(servopin,HIGH); //increasing the level of motor interface to upmost
  delayMicroseconds(pulsewidth); //delaying microsecond of pulse width value
```

```
digitalWrite(servopin,LOW); //decreasing the level of motor interface to the least
delay(20-pulsewidth/1000);

}
void Set_servopulse(int set_val)
{
  noInterrupts();
  for(int i=0;i<=100;i++) //giving motor enough time to turn to assigning point
  {
    servopulse(servopin,set_val); //invokimg pulse function
  }
  interrupts();
}
void M_Control_IO_config(void)
{
  pinMode(pinLB,OUTPUT); // /pin 2
  pinMode(pinLF,OUTPUT); // pin 4
  pinMode(pinRB,OUTPUT); // pin 7
  pinMode(pinRF,OUTPUT); // pin 8
  pinMode(Lpwm_pin,OUTPUT); // pin 11 (PWM) -->5
  pinMode(Rpwm_pin,OUTPUT); // pin10(PWM)
}
void Set_Speed(unsigned char Left,unsigned char Right) //function of setting speed
{
  analogWrite(Lpwm_pin,Left);
  analogWrite(Rpwm_pin,Right);
}
void advance() // going forward
{
  digitalWrite(pinRB,LOW); // making motor move towards right rear
  digitalWrite(pinRF,HIGH);
  digitalWrite(pinLB,LOW); // making motor move towards left rear
  digitalWrite(pinLF,HIGH);
  Car_state = 1;
  show_state();
}
void turnR() //turning right(dual wheel)
{
  digitalWrite(pinRB,HIGH); //making motor move towards right rear
  digitalWrite(pinRF,LOW);
  digitalWrite(pinLB,LOW);
  digitalWrite(pinLF,HIGH); //making motor move towards left front
  Car_state = 4;
  show_state();
}
void turnL() //turning left(dual wheel)
{
  digitalWrite(pinRB,LOW);
  digitalWrite(pinRF,HIGH); //making motor move towards right front
  digitalWrite(pinLB,HIGH); //making motor move towards left rear
  digitalWrite(pinLF,LOW);
  Car_state = 3;
  show_state();
}
void stopp() //stop
{
  digitalWrite(pinRB,HIGH);
  digitalWrite(pinRF,HIGH);
  digitalWrite(pinLB,HIGH);
  digitalWrite(pinLF,HIGH);
  Car_state = 5;
  show_state();
}
void back() //back up
{
  digitalWrite(pinRB,HIGH); //making motor move towards right rear
  digitalWrite(pinRF,LOW);
  digitalWrite(pinLB,HIGH); //making motor move towards left rear
  digitalWrite(pinLF,LOW);
  Car_state = 2;
  show_state();
}
void show_state(void)
```

```

{
  lcd.setCursor(0, 1);
  switch(Car_state)
  {
    case 1:lcd.print(" Go ");Serial.print(" \r\n GO");
    break;
    case 2:lcd.print("Back ");Serial.print(" \r\n Back");
    break;
    case 3:lcd.print("Left ");Serial.print(" \r\n Left");
    break;
    case 4:lcd.print("Right");Serial.print(" \r\n Right");
    break;
    case 5:lcd.print("Stop ");Serial.print(" \r\n Stop");
    break;
    default:
    break;
  }
}
void LCD1602_init(void) //including initialized function of liquid crystal
{
  lcd.init(); //invoking initialized function of LCD in LiquidCrystal_I2C.h
  delay(10); //delaying for 10 millisecond
  lcd.backlight(); //open backlight of LCD1602
  lcd.clear(); //clear screen
}
void Show_V(unsigned char V)
{
  lcd.setCursor(11, 0);
  lcd.print("V= ");
  lcd.setCursor(13, 0);
  lcd.print(V,DEC);
  Serial.print("\n Speed = ");
  Serial.print(V,DEC);
}
void Show_ServoAngle(int Angle)
{
  lcd.setCursor(6,1);
  lcd.print("C= ");
  lcd.setCursor(8, 1);
  lcd.print(Angle,DEC);
  Serial.print("\n Angle = ");
  Serial.print(Angle,DEC);
}
void Tracking_Mode(void) //function of line tracking
{
  if(Ultrasonic_Ranging(0) < 5){
    stopp();
    delay(100);
    back();
    delay(50);
    stopp();
  }else{

  lcd.setCursor(0, 0); //setting cursor in the first row and column
  lcd.print("Infrared Tracking_Mode ");
  Sensor_Scan();
  if (SM == HIGH)// middle sensor in black area
  {
    if (SL == LOW & SR == HIGH) // black on left, white on right, turn left
    {
      turnR();
    }
    else if (SR == LOW & SL == HIGH) // white on left, black on right, turn
right
    {
      turnL();
    }
    else // white on both sides, going forward
    {
      advance();
    }
  }
  }
  else // middle sensor on white area
  {
    if (SL== LOW & SR == HIGH)// black on left, white on right, turn left

```

```
        {
            turnR();
        }
        else if (SR == LOW & SL == HIGH) // white on left, black on right, turn
right
        {
            turnL();
        }
        else // all white, stop
        {
            back();
            delay(100);
            stopp() ;
        }
    }
}
}
void Self_Control(void)//self-going, ultrasonic obstacle avoidance
{
    int H;
    lcd.setCursor(0, 0); //setting cursor in the first row and column
    lcd.print("Ultrasonic_Self_Ctr ");
    Show_V(Lpwm_val);
    Set_servopulse(ServoAngle);
    Show_ServoAngle(ServoAngle);
    H = Ultrasonic_Ranging(1);
    delay(100);
    if(Ultrasonic_Ranging(1) < 35)
    {
        stopp();
        delay(100);
        back();
        delay(50);
    }

    if(Ultrasonic_Ranging(1) < 60)
    {
        stopp();
        delay(100);
        Set_servopulse(45);
        Show_ServoAngle(45);
        int L = ask_pin_L(2);
        delay(300);
        Set_servopulse(140);
        Show_ServoAngle(140);
        int R = ask_pin_R(3);
        delay(300);

        if(L > R)
        {
            back();
            delay(100);
            turnL();
            delay(400);
            stopp();
            delay(50);
            Set_servopulse(ServoAngle);
            Show_ServoAngle(ServoAngle);
            H = Ultrasonic_Ranging(1);
            delay(500);
        }

        if(L <= R)
        {
            back();
            delay(100);
            turnR();
            delay(400);
            stopp();
            delay(50);
            Set_servopulse(ServoAngle);
            Show_ServoAngle(ServoAngle);
            H = Ultrasonic_Ranging(1);
            delay(300);
        }
    }
}
```

```
        if (L < 35 && R < 35)
        {
            stopp();
            delay(50);
            back();
            delay(50);
        }
    }
    else
    {
        advance();
    }
}
int Ultrasonic_Ranging(unsigned char Mode)//function of ultrasonic distance detecting
, MODE=1, displaying, no displaying under other situation
{
    int old_distance;
    digitalWrite(outputPin, LOW);
    delayMicroseconds(2);
    digitalWrite(outputPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(outputPin, LOW);
    int distance = pulseIn(inputPin, HIGH); // reading the duration of high level
    distance= distance/58; // Transform pulse time to distance
    if(Mode==1){
        lcd.setCursor(11, 1);
        lcd.print("H= ");
        lcd.setCursor(13, 1);
        lcd.print(distance,DEC);
        Serial.print("\n H = ");
        Serial.print(distance,DEC);
        return distance;
    }
    else return distance;
}
int ask_pin_L(unsigned char Mode)
{
    int old_Ldistance;
    digitalWrite(outputPin, LOW);
    delayMicroseconds(2);
    digitalWrite(outputPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(outputPin, LOW);
    int Ldistance = pulseIn(inputPin, HIGH);
    Ldistance= Ldistance/58; // Transform pulse time to distance
    if(Mode==2){
        lcd.setCursor(11, 1);
        lcd.print("L= ");
        lcd.setCursor(13, 1);
        lcd.print(Ldistance,DEC);
        Serial.print("\n L = ");
        Serial.print(Ldistance,DEC);
        return Ldistance;
    }
    else return Ldistance;
}
int ask_pin_R(unsigned char Mode)
{
    int old_Rdistance;
    digitalWrite(outputPin, LOW);
    delayMicroseconds(2);
    digitalWrite(outputPin, HIGH); //
    delayMicroseconds(10);
    digitalWrite(outputPin, LOW);
    int Rdistance = pulseIn(inputPin, HIGH);
    Rdistance= Rdistance/58; // Transform pulse time to distance
    if(Mode==3){
        lcd.setCursor(11, 1);
        lcd.print("R= ");
        lcd.setCursor(13, 1);
        lcd.print(Rdistance,DEC);
        Serial.print("\n R = ");
        Serial.print(Rdistance,DEC);
        return Rdistance;
    }
}
```

```

    }
    else return Rdistance;
}

void IR_Control(void) //remote control, when pressing"#", it quitting from the mode
{
    unsigned long Key;
    lcd.setCursor(0,0); //setting cursor in the first row and column
    lcd.print("IR_Ctr      ");

    while(Key!=IR_ESC)
    {
        if(irrecv.decode(&results)) //to judge whether serial port receive data
        {
            Key = results.value;
            switch(Key)
            {

                case IR_Go:advance(); //UP
                break;
                case IR_Back: back(); //back
                break;
                case IR_Left:turnL(); //Left
                break;
                case IR_Right:turnR(); //Righ
                break;
                case IR_Stop:stopp(); //stop
                break;
                case IR_Servo_L: if(ServoAngle<=180){ //motor turning left
                    ServoAngle=ServoAngle+10;

                    Set_servopulse(ServoAngle);
                    Show_ServoAngle(ServoAngle);
                    //Set_servopulse(45);
                    //Show_ServoAngle(45);
                }

                break;
                case IR_Servo_R: if(ServoAngle-10>=0){ //motor turning right
                    ServoAngle-=10;
                    Set_servopulse(ServoAngle);
                    Show_ServoAngle(ServoAngle);
                    //Set_servopulse(140);
                    //Show_ServoAngle(140);
                }

                break;
                case IR_Speed_UP:if(Rpwm_val+10<=250&&Rpwm_val+10<=250){ //increasing speed
                    Lpwm_val+=10; Rpwm_val+=10;
                    Set_Speed(Lpwm_val,Rpwm_val);
                    Show_V(Lpwm_val);
                }

                break;
                case IR_Speed_DOWN:if(Rpwm_val-10>=0&&Rpwm_val-10>=0){ //decreasing speed
                    Lpwm_val-=10; Rpwm_val-=10;
                    Set_Speed(Lpwm_val,Rpwm_val);
                    Show_V(Lpwm_val);
                }

                break;
                default:
                break;
            }
            irrecv.resume(); // Receive the next value
        }
    }
    lcd.clear();
    lcd.setCursor(0, 0); //setting cursor in the first row and column
    lcd.print(" Wait Signal ");
    stopp();
}

void Bluetooth_Control() //Bluetooth remote control
{
    lcd.setCursor(0, 0); //setting cursor in the first row and column
    lcd.print("BluetoothControl");

    if(Serial.available()) //to judge whether serial port receive data
    {

```

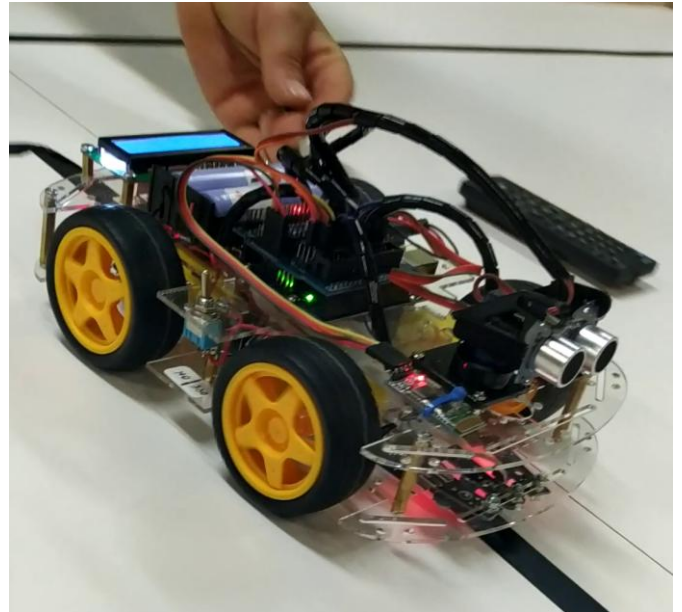
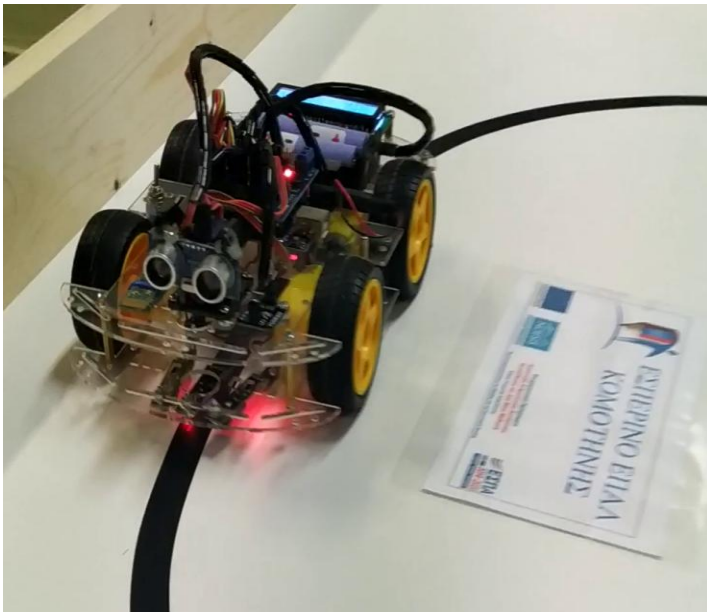
```
Bluetooth_val=Serial.read(); //reading value of Bluetooth serial port, giving the value to
val
  lcd.setCursor(1, 0); //setting cursor in the first row and column
  lcd.print("BT:"+Bluetooth_val);

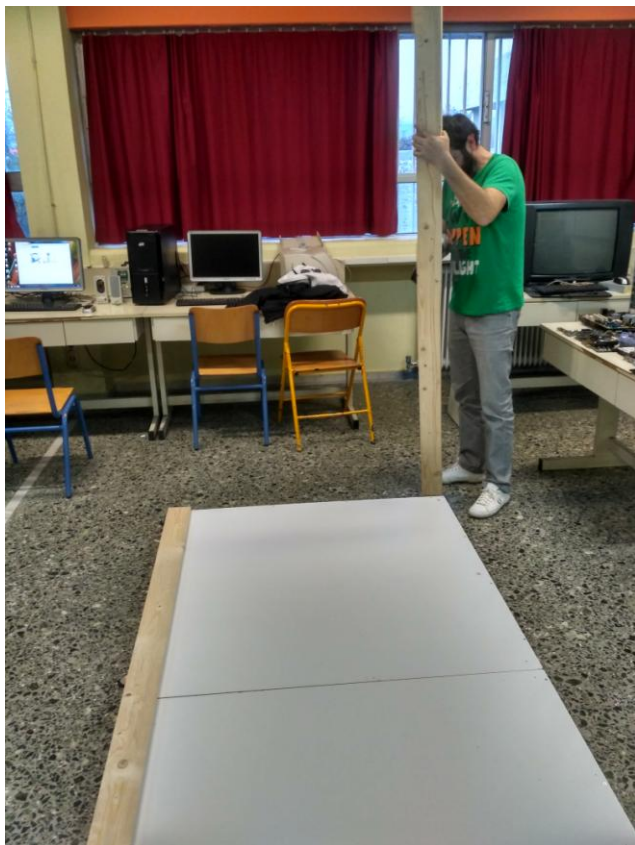
switch(Bluetooth_val)
{
  case 'G':advance(); //GO
  break;
  case 'B': back(); //back
  break;
  case 'L':turnL(); //Left
  break;
  case 'R':turnR(); //Right
  break;
  case 'S':stopp(); //stop
  break;
  case 'U':SpeedUp(); //SpeedUp
  break;
  case 'D':SpeedDown(); //SpeedDown
  break;
}
}
}
void SpeedUp()
{
if(Rpwm_val+10<=250&&Rpwm_val+10<=250){ //increasing speed
  Lpwm_val+=10; Rpwm_val+=10;
  Set_Speed(Lpwm_val,Rpwm_val);
  Show_V(Lpwm_val);
}
}
void SpeedDown()
{
if(Rpwm_val-10>=0&&Rpwm_val-10>=0){ //decreasing speed
  Lpwm_val-=10; Rpwm_val-=10;
  Set_Speed(Lpwm_val,Rpwm_val);
  Show_V(Lpwm_val);
}
}
void setup()
{
  Serial.begin(9600); //initialized serial port , using Bluetooth as serial port,
  setting baud
  pinMode(servopin,OUTPUT); //setting motor interface as output
  LCD1602_init(); //initializing 1602
  M_Control_IO_config(); //motor controlling the initialization of IO
  Set_Speed(Lpwm_val,Rpwm_val); //setting initialized speed
  Set_servopulse(ServoAngle); //setting initialized servomotor angle
  Sensor_IO_Config(); //initializing IO of line tracking module
  irrecv.enableIRIn(); //starting receiving IR remote control signal
  pinMode(inputPin, INPUT); //IO of ultrasonic module
  pinMode(outputPin, OUTPUT); //IO of ultrasonic module
  lcd.setCursor(0, 0); //setting cursor at 0.0
  lcd.print(" Wait Signal "); //LCD printing character string
  stopp(); //stop
}
void loop()
{
  if (irrecv.decode(&results))
  { //when receiving a button
    if(results.value == IR_Tracking_Mode)
    {
      Set_Speed(90,90);
      while(IR_Tracking_Mode)
      {
        Tracking_Mode();
      } //pressing "OK" on remote controller, and entering remote control mode
    }
    if(results.value == IR_Self_Control)
    {
      while(IR_Self_Control)
```

```
    {
      Self_Control(); //pressing "OK" on remote controller, and entering remote control mode
    }
  }
  if(results.value == IR_IR_Control)
  {

    IR_Control(); //pressing "OK" on remote controller, and entering remote control mode

  }
  if(results.value == IR_Bluetooth_Control)
  {
    while(IR_Bluetooth_Control)
    {
      Bluetooth_Control(); //pressing "OK" on remote controller, and entering remote control
mode
    }
  }
  irrecv.resume(); // Receive the next value
}
delay(10);
}
```



Βίντεο της λειτουργίας των έξυπνων συσκευών

<https://www.youtube.com/watch?v=fWdS0nkealE>

4. Έξυπνη συσκευή ραντάρ υπερήχων (Ultrasonic Radar)

- **Φυλλάδιο παρουσίασης**
- **Κώδικας προγραμματισμού που αναπτύχθηκε**
 - **Κώδικας Arduino IDE**
 - **Κώδικας Processing**
- **Φωτογραφικό υλικό από την εκτέλεση των ασκήσεων και υλοποίησης**
- **Βίντεο της λειτουργίας των έξυπνων συσκευών.**



«ΜΙΑ ΝΕΑ ΑΡΧΗ ΣΤΑ ΕΠΑΛ»

“Νέα” αρχή σε “Νέο” ΕΠΑΛ !

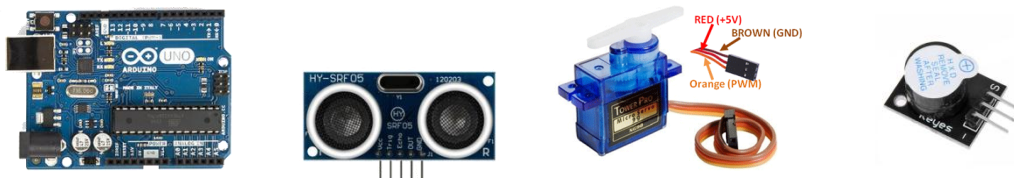
Νέα έξυπνα συστήματα



Οι μαθητές σε συνεργασία με τους εκπαιδευτικούς δημιούργησαν μία σειρά από έξυπνα συστήματα - αυτοματισμούς στο μάθημα «Ερευνητική Εργασία στην Τεχνολογία» και «Αρχές Προγραμματισμού Υπολογιστών». Τα συστήματα αυτά χρησιμοποιούν έξυπνους μικροελεγκτές της σειράς Arduino. Το Arduino είναι μια ηλεκτρονική πλατφόρμα ανοικτού κώδικα και σχεδιασμού, που βασίζεται σε ευέλικτο και εύκολο στη χρήση υλικό και λογισμικό. Προορίζεται για καλλιτέχνες, σχεδιαστές, υλοποίηση χόμπι και δραστηριοτήτων, και γενικότερα για οποιονδήποτε ενδιαφέρεται να δημιουργήσει αλληλεπιδραστικά αντικείμενα ή περιβάλλοντα. Τελικός σκοπός είναι να εισαχθεί ο μαθητής με ευχάριστο τρόπο στα ψηφιακά κυκλώματα και τον προγραμματισμό. Πρόκειται για ομαδικά έργα σε τεχνολογία αιχμής, που υποδεικνύουν ότι τα ΕΠΑΛ δίνουν γερές βάσεις για την τεχνολογία του μέλλοντος, ειδικά στον αυτοματισμό και την ασύρματη επικοινωνία με κινητές συσκευές

#4 Έξυπνη συσκευή ραντάρ υπερήχων (Ultrasonic Radar)

Η έξυπνη αυτή εφαρμογή

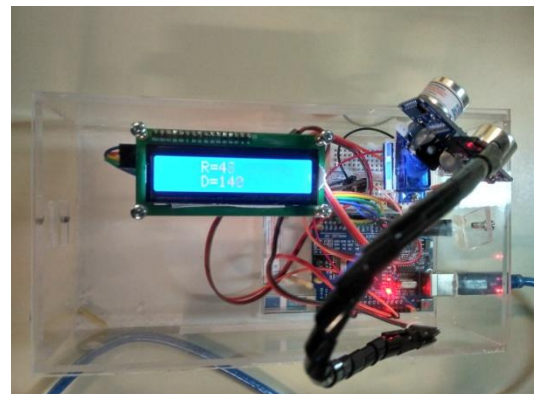
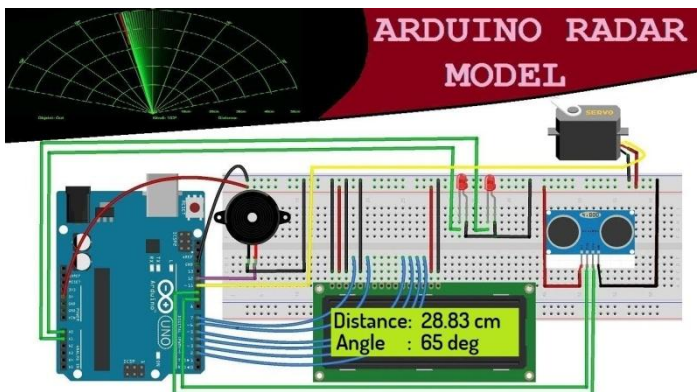


Χρησιμοποιήθηκαν:

Μικροελεγκτής ARDUINO, Αισθητήριο υπερήχων, Σερβοκινητήρας, Ενεργός Βομβητής

Αναπτύχθηκαν εφαρμογές κώδικα:

Κώδικας ελέγχου ARDUINO (S4A, ARDUINO IDE, C++), Βοηθητική εφαρμογή επεξεργασίας δεδομένων Processing



Ομάδα Εργασίας Εσπερινού ΕΠΑΛ Κομοτηνής:

ΧΑΚΑΝ Α., ΟΛΥΜΠΙΟΣ Α., ΧΑΚΑΝ Α., ΠΑΥΛΟΣ Γ., ΟΥΡΑΝΙΑ Μ., ΕΛΕΝΗ Τ, ΔΗΜΗΤΡΗΣ Δ..

στο πλαίσιο της δράσης του έργου MIS 5011051, Υποστήριξη και Διαχείριση των Σχεδίων Δράσης του έργου "Μια Νέα Αρχή στα ΕΠΑΛ"

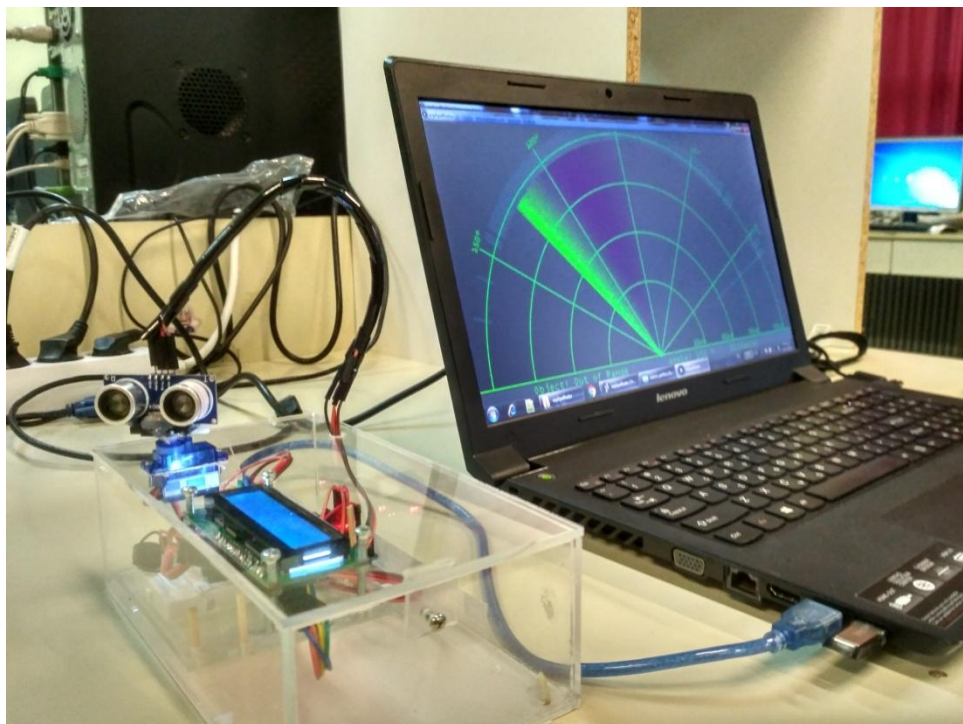




#2B Βοηθητική εφαρμογή ραντάρ υπερήχων (Ultrasonic Radar)

Η εφαρμογή αυτή (για συσκευές με λειτουργικό android), αναπτύχθηκε με την βοήθεια των μαθητών του τομέα Πληροφορικής του Εσπερινού ΕΠΑΛ Κομοτηνής, στο πλαίσιο της δράσης του έργου MIS 5011051 «Υποστήριξη και Διαχείριση των Σχεδίων Δράσης του έργου "Μια Νέα Αρχή στα ΕΠΑΛ"». Η εφαρμογή συνδέεται με τη θύρα USB2Serial με τον ελεγκτή arduino.

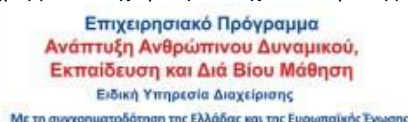
Χρησιμοποιήθηκε η πλατφόρμα ανάπτυξης **Processing**, όπου με την βοήθεια της γλώσσας προγραμματισμού **Java** τα δεδομένα που στέλνει ο μικροελεγκτής μέσω της σειραϊκής θύρας οπτικοποιούνται στο συνηθισμένο ημικυκλικό γράφημα ραντάρ.



Ομάδα Εργασίας Εσπερινού ΕΠΑΛ Κομοτηνής:

ΧΑΚΑΝ Α., ΟΛΥΜΠΙΟΣ Α., ΧΑΚΑΝ Α., ΠΑΥΛΟΣ Γ., ΟΥΡΑΝΙΑ Μ., ΕΛΕΝΗ Τ. ΔΗΜΗΤΡΗΣ Δ.

στο πλαίσιο της δράσης του έργου MIS 5011051, Υποστήριξη και Διαχείριση των Σχεδίων Δράσης του έργου "Μια Νέα Αρχή στα ΕΠΑΛ"



Κώδικας Arduino IDE

```
/*
  Arduino Radar

  By Esperino EPAL KOMOTINIS

  Implemented during classes of project
  "MNAE" Mia Nea Arxi sta Epal
  KOMOTINI - GREECE 12-3-2019

  used:
  Ultrasonic Sensor HC-SR04
  SG90 Mini Gear Micro Servo 9g
  IIC / I2C 1602 Green Backlight LCD Display Screen

  optional:
  Processing IDE
*/
// Includes
#include <Servo.h>
#include <LiquidCrystal_I2C.h> //including libraries of I2C-LCD1602 liquid crystal
#include <Wire.h> //including libraries of I2C
#include <SoftwareSerial.h>

// Defines Trig and Echo pins of the Ultrasonic Sensor
const int trigPin = 10;
const int echoPin = 11;
// I2C-LCD1602 liquid crystal module SDA to A4
// I2C-LCD1602 liquid crystal module SCL to A5
const int servopin=12; //defining digital port pin 12, connecting to signal line of
servo motor
const int buzzer = 5;

// Variables for the duration and the distance
long duration;
int distance;
int DistanceSec;

SoftwareSerial Bluetooth(3, 4); // Bluetooth RX | TX
LiquidCrystal_I2C lcd(0x27,16,2); //defining liquid crystal
Servo myServo; // Creates a servo object for controlling the servo motor

void LCD1602_init(void) //including initialized function of liquid crystal
{
  lcd.init(); //invoking initialized function of LCD in LiquidCrystal_I2C.h
  delay(10); //delaying for 10 millisecond
  lcd.backlight(); //open backlight of LCD1602
  lcd.clear(); //clear screen
}

void setup() {
  pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
  pinMode(echoPin, INPUT); // Sets the echoPin as an Input
  pinMode(buzzer, OUTPUT);
  Serial.begin(9600);
  Bluetooth.begin(9600);
  myServo.attach(servopin); // Defines on which pin is the servo motor attached
  LCD1602_init(); //initializing 1602
  DistanceSec=15;
}

void loop() {
  // rotates the servo motor from 15 to 165 degrees
  String s1 = "";
  for(int i=15;i<=165;i=i+1){
    myServo.write(i);
    delay(60);
    distance = calculateDistance();// Calls a function for calculating the distance measured by
the Ultrasonic sensor for each degree

    Serial.print(i); // Sends the current degree into the Serial Port
    Serial.print(","); // Sends addition character right next to the previous value needed later
in the Processing IDE for indexing
```

Έξυπνη συσκευή ραντάρ υπερήχων (Ultrasonic Radar)

```
Serial.print(distance); // Sends the distance value into the Serial Port
Serial.print("."); // Sends addition character right next to the previous value needed later
in the Processing IDE for indexing

s1=String(i)+","+String(distance)+".";
Bluetooth.println(s1);

lcd.setCursor(4, 0);
lcd.print("R= ");
lcd.setCursor(6, 0);
lcd.print(i,DEC);
lcd.setCursor(4, 1);
lcd.print("D= ");
lcd.setCursor(6, 1);
lcd.print(distance,DEC);
if (distance <= DistanceSec)
{
    tone(buzzer, 10); // Send 1KHz sound signal...
    //digitalWrite(ledPin1, LOW);
    //digitalWrite(ledPin2, HIGH);
    delay(200);
    noTone(buzzer); // Stop sound...
}

}
// Repeats the previous lines from 165 to 15 degrees
for(int i=165;i>15;i=i-1){
myServo.write(i);
delay(60);
distance = calculateDistance();
Serial.print(i);
Serial.print(",");
Serial.print(distance);
Serial.print(".");
lcd.setCursor(4, 0);
lcd.print("R= ");
lcd.setCursor(6, 0);
lcd.print(i,DEC);
lcd.setCursor(4, 1);
lcd.print("D= ");
lcd.setCursor(6, 1);
lcd.print(distance,DEC);

if (distance <= DistanceSec)
{
    tone(buzzer, 10); // Send 1KHz sound signal...
    //digitalWrite(ledPin1, LOW);
    //digitalWrite(ledPin2, HIGH);
    delay(200);
    noTone(buzzer); // Stop sound...
}
}
}
// Function for calculating the distance measured by the Ultrasonic sensor
int calculateDistance(){

digitalWrite(trigPin, LOW);
delayMicroseconds(2);
// Sets the trigPin on HIGH state for 10 micro seconds
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
duration = pulseIn(echoPin, HIGH); // Reads the echoPin, returns the sound wave travel time in
microseconds
distance= duration*0.034/2;
return distance;
}
```

Κώδικας Processing

```
/* Arduino Radar Project
 *
 * Updated version. Fits any screen resolution!
 * Just change the values in the size() function,
 * with your screen resolution.
 *
 * by Dejan Nedelkovski,
 * www.HowToMechatronics.com
 */
import processing.serial.*; // imports library for serial communication
import java.awt.event.KeyEvent; // imports library for reading the data from the serial port
import java.io.IOException;
Serial myPort; // defines Object Serial
// defubes variables
String angle="";
String distance="";
String data="";
String noObject;
float pixsDistance;
int iAngle, iDistance;
int index1=0;
int index2=0;
PFont orcFont;
void setup() {

    size (1360, 700); // ***CHANGE THIS TO YOUR SCREEN RESOLUTION***
    smooth();
    myPort = new Serial(this,"COM4", 9600); // starts the serial communication
    myPort.bufferUntil('.'); // reads the data from the serial port up to the character '.'. So
    actually it reads this: angle,distance.
    orcFont = loadFont("OCRAExtended-30.vlw");
}
void draw() {

    fill(98,245,31);
    textFont(orcFont);
    // simulating motion blur and slow fade of the moving line
    noStroke();
    fill(0,4);
    rect(0, 0, width, height-height*0.065);

    fill(98,245,31); // green color
    // calls the functions for drawing the radar
    drawRadar();
    drawLine();
    drawObject();
    drawText();
}
void serialEvent (Serial myPort) { // starts reading data from the Serial Port
    // reads the data from the Serial Port up to the character '.' and puts it into the String
    variable "data".
    data = myPort.readStringUntil('.');
    data = data.substring(0,data.length()-1);

    index1 = data.indexOf(","); // find the character ',' and puts it into the variable "index1"
    angle= data.substring(0, index1); // read the data from position "0" to position of the
    variable index1 or thats the value of the angle the Arduino Board sent into the Serial Port
    distance= data.substring(index1+1, data.length()); // read the data from position "index1" to
    the end of the data pr thats the value of the distance

    // converts the String variables into Integer
    iAngle = int(angle);
    iDistance = int(distance);
}
void drawRadar() {
    pushMatrix();
    translate(width/2,height-height*0.074); // moves the starting coordinats to new location
    noFill();
    strokeWeight(2);
    stroke(98,245,31);
    // draws the arc lines
    arc(0,0,(width-width*0.0625),(width-width*0.0625),PI,TWO_PI);
    arc(0,0,(width-width*0.27),(width-width*0.27),PI,TWO_PI);
}
```



```
arc(0,0,(width-width*0.479),(width-width*0.479),PI,TWO_PI);
arc(0,0,(width-width*0.687),(width-width*0.687),PI,TWO_PI);
// draws the angle lines
line(-width/2,0,width/2,0);
line(0,0,(-width/2)*cos(radians(30)),(-width/2)*sin(radians(30)));
line(0,0,(-width/2)*cos(radians(60)),(-width/2)*sin(radians(60)));
line(0,0,(-width/2)*cos(radians(90)),(-width/2)*sin(radians(90)));
line(0,0,(-width/2)*cos(radians(120)),(-width/2)*sin(radians(120)));
line(0,0,(-width/2)*cos(radians(150)),(-width/2)*sin(radians(150)));
line((-width/2)*cos(radians(30)),0,width/2,0);
popMatrix();
}
void drawObject() {
  pushMatrix();
  translate(width/2,height-height*0.074); // moves the starting coordinats to new location
  strokeWeight(7);
  stroke(255,10,10); // red color
  pixsDistance = iDistance*((height-height*0.1666)*0.025); // covers the distance from the
  sensor from cm to pixels
  // limiting the range to 40 cms
  if(iDistance<40){
    // draws the object according to the angle and the distance
    line(pixsDistance*cos(radians(iAngle)),-pixsDistance*sin(radians(iAngle)),(width-
width*0.505)*cos(radians(iAngle)),-(width-width*0.505)*sin(radians(iAngle)));
  }
  popMatrix();
}
void drawLine() {
  pushMatrix();
  strokeWeight(9);
  stroke(30,250,60);
  translate(width/2,height-height*0.074); // moves the starting coordinats to new location
  line(0,0,(height-height*0.12)*cos(radians(iAngle)),-(height-
height*0.12)*sin(radians(iAngle))); // draws the line according to the angle
  popMatrix();
}
void drawText() { // draws the texts on the screen

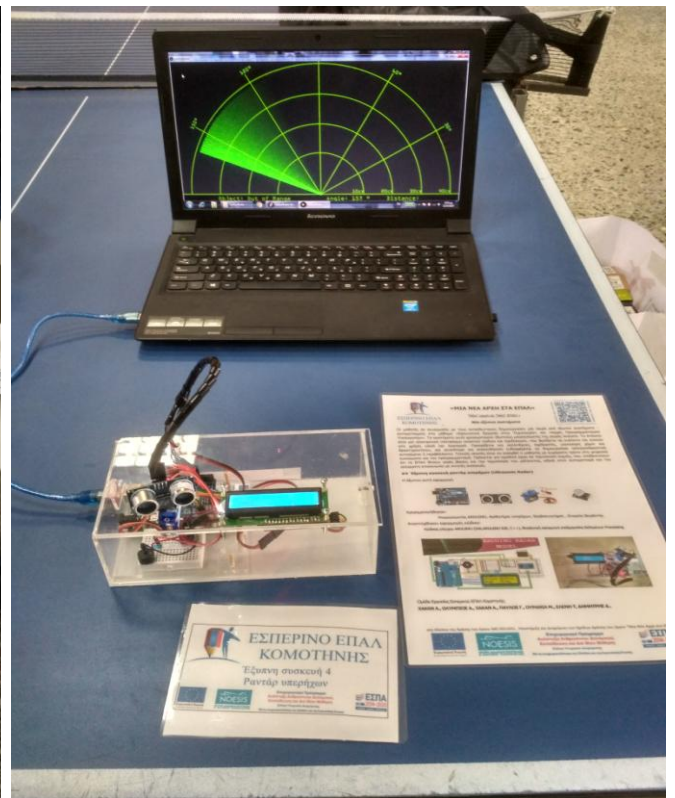
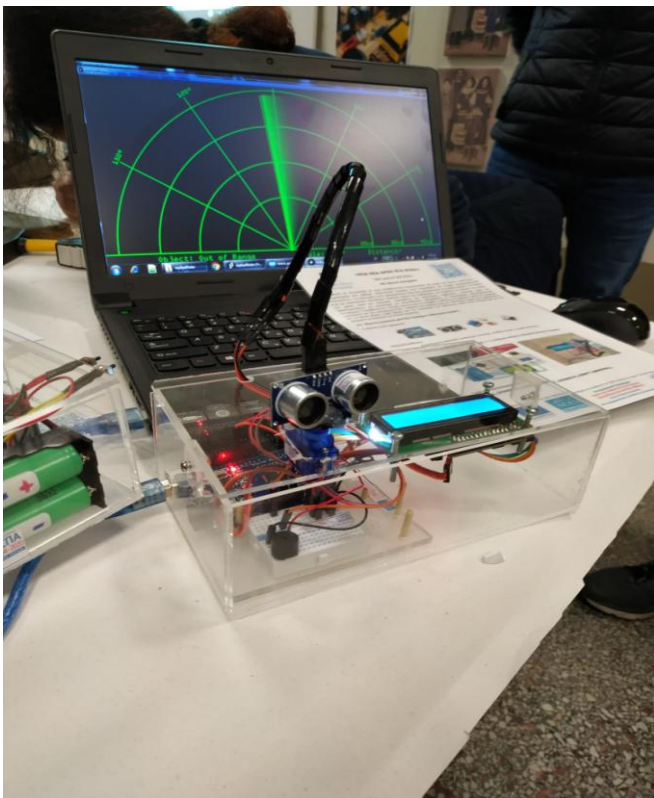
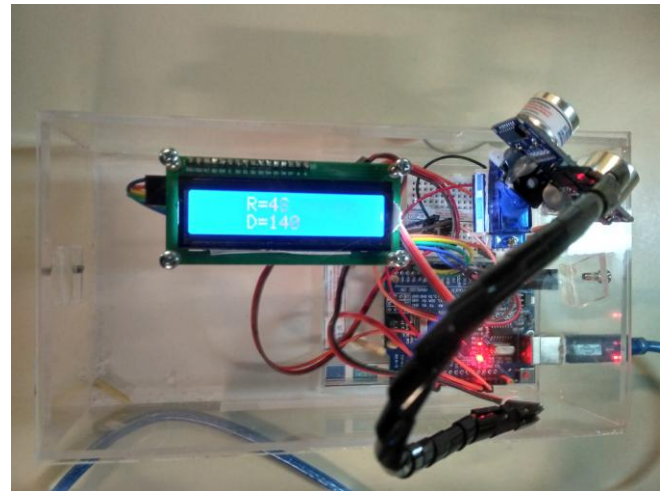
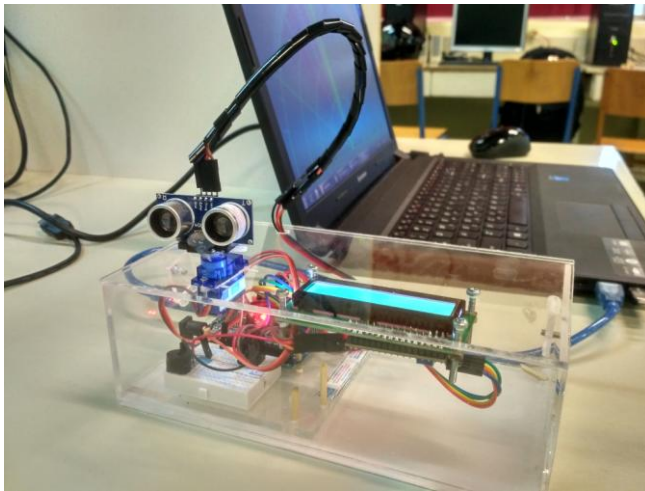
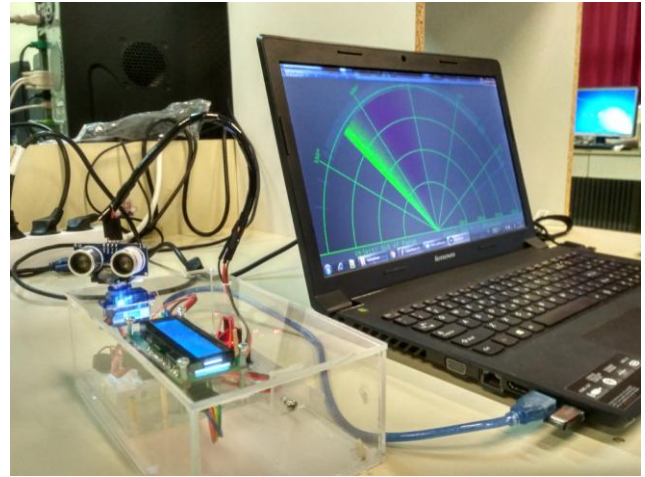
  pushMatrix();
  if(iDistance>40) {
    noObject = "Out of Range";
  }
  else {
    noObject = "In Range";
  }
  fill(0,0,0);
  noStroke();
  rect(0,height-height*0.0648,width,height);
  fill(98,245,31);
  textSize(25);

  text("10cm",width-width*0.3854,height-height*0.0833);
  text("20cm",width-width*0.281,height-height*0.0833);
  text("30cm",width-width*0.177,height-height*0.0833);
  text("40cm",width-width*0.0729,height-height*0.0833);
  textSize(30);
  text("Object: " + noObject,width-width*0.875,height-height*0.0277);
  text("Angle: " + iAngle + " °",width-width*0.48,height-height*0.0277);
  text("Distance: ",width-width*0.26,height-height*0.0277);
  if(iDistance<40) {
    text(" " + iDistance + " cm",width-width*0.225,height-height*0.0277);
  }
  textSize(25);
  fill(98,245,60);
  translate((width-width*0.4994)+width/2*cos(radians(30)),(height-height*0.0907)-
width/2*sin(radians(30)));
  rotate(-radians(-60));
  text("30°",0,0);
  resetMatrix();
  translate((width-width*0.503)+width/2*cos(radians(60)),(height-height*0.0888)-
width/2*sin(radians(60)));
  rotate(-radians(-30));
  text("60°",0,0);
  resetMatrix();
  translate((width-width*0.507)+width/2*cos(radians(90)),(height-height*0.0833)-
```

Έξυνη συσκευή ραντάρ υπερήχων (Ultrasonic Radar)

```
width/2*sin(radians(90)));
  rotate(radians(0));
  text("90°",0,0);
  resetMatrix();
  translate(width-width*0.513+width/2*cos(radians(120)),(height-height*0.07129)-
width/2*sin(radians(120)));
  rotate(radians(-30));
  text("120°",0,0);
  resetMatrix();
  translate((width-width*0.5104)+width/2*cos(radians(150)),(height-height*0.0574)-
width/2*sin(radians(150)));
  rotate(radians(-60));
  text("150°",0,0);
  popMatrix();
}
```

Φωτογραφικό υλικό από την εκτέλεση των ασκήσεων και υλοποίησης



Βίντεο της λειτουργίας των έξυπνων συσκευών

<https://www.youtube.com/watch?v=XmBitRsodDg>



ΕΣΠΕΡΙΝΟ ΕΠΑΛ ΚΟΜΟΤΗΝΗΣ

ΣΧΕΔΙΑ ΔΡΑΣΗΣ ΜΙΑ ΝΕΑ ΑΡΧΗ ΣΤΑ ΕΠΑΛ (Μ.Ν.Α.Ε.)

"ΝΕΑ" ΑΡΧΗ ΣΕ "ΝΕΟ" ΕΠΑΛ



Επιχειρησιακό Πρόγραμμα
Ανάπτυξη Ανθρώπινου Δυναμικού,
Εκπαίδευση και Διά Βίου Μάθηση

Ειδική Υπηρεσία Διαχείρισης

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



Επισκεφθείτε το ιστολόγιο μας

ΕΣΠΕΡΙΝΟ ΕΠΑΛ ΚΟΜΟΤΗΝΗΣ



ΕΣΠΕΡΙΝΟ ΕΠΑΛ ΚΟΜΟΤΗΝΗΣ

<https://blogs.sch.gr/epal-esp-komot>