

# Εργασία Ενσωματωμένων Συστημάτων

Σαλαχώρης Μαυρουδής<sup>1</sup> και Κυριακόπουλος Αριστομένης<sup>1</sup>

Πανεπιστήμιο Πατρών

Τμήμα Μηχανικών Ηλεκτρονικών Υπολογιστών και Πληροφορικής  
up1054436@upnet.gr up1054307@upnet.gr

**Περίληψη** Σκοπός της εργασίας είναι η δημιουργία ενός αυτόματα ελεγχόμενου τραπέζιού μπιλιάρδου που λειτουργεί με νόμισμα, αλλά με ειδοποιό διαφορά την ηλεκτρονική υλοποίησή του αντί της μηχανικής. Ο μικροελεγκτής που θα χρησιμοποιηθεί είναι ο ATmega328P και η υλοποίηση του λογισμικού θα γίνει στην γλώσσα C++. Το σύστημα θα μπορεί να ανιχνεύει χρώματα μπαλών όπως επίσης και κέρματα. Διαθέτει, ακόμα, μία οθόνη 7 τμημάτων για την εμφάνιση του ποσού. Τέλος, το τραπέζι μπιλιάρδου θα έχει και μία έξυπνη λειτουργία που θα ενημερώνει τον διαχειριστή του πόσα παιχνίδια έχουν παιχτεί κάθε ώρα, όπως επίσης και πόσα κέρματα έχει μέσα.

## 1 Περιγραφή Προβλήματος

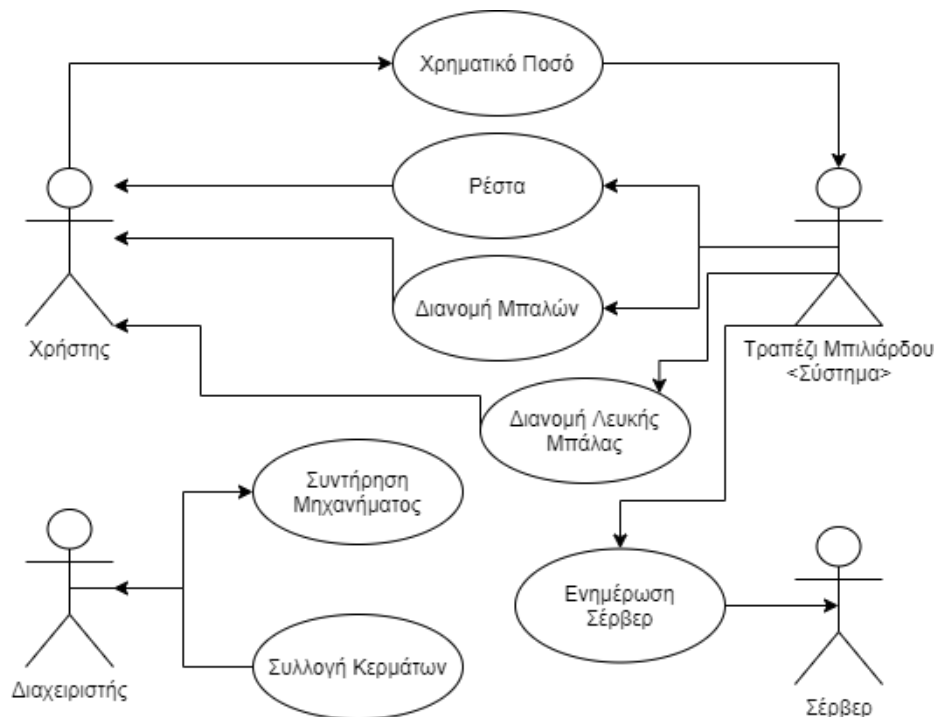
Τα περισσότερα τραπέζια χρησιμοποιούν μηχανικούς τρόπους για να ξεχωρίσουν την άσπρη μπάλα από τις υπόλοιπες, συνήθως κάνοντάς την ελαφρώς βαρύτερη ή μεγαλύτερη από τις άλλες. Οι μηχανικοί τρόποι όμως δημιουργούν μια ανομοιομορφία (έστω και μικρή) μεταξύ των μπαλών και φθείρονται με τη πάροδο του χρόνου, το οποίο μακροπρόθεσμα καθιστά τον μηχανισμό άχρηστο. Η συντήρηση ενός μηχανικού τραπέζιου είναι δυσκολότερη, αφού απαιτεί την ενεργή παρουσία κάποιου ατόμου, το οποίο είναι υπεύθυνο για την επίλυση πιθανών εμπλοκών του μηχανισμού, την ύπαρξη σκόνης και άλλων προβλημάτων που μειώνουν την κερδοφορία της μηχανής.

Ένας ηλεκτρονικός μηχανισμός για το τραπέζι του μπιλιάρδου μπορεί να παρέχει πολλές λειτουργίες που ο μηχανικός σχεδιασμός δεν μπορεί να πετύχει. Ο ιδιοκτήτης του τραπέζιου μπορεί να ενημερώνεται για το πόσες φορές έχει χρησιμοποιηθεί το τραπέζι κατά τη διάρκεια της ημέρας χωρίς να βρίσκεται στον ίδιο χώρο με αυτό. Υπάρχουν ακόμα και σχέδια για έξυπνα μπιλιάρδα που γνωρίζουν πότε μια μπάλα μπήκε και ποια μπάλα ήταν αυτή, πράγμα που καθιστά ένα μπιλιάρδο ικανό να γνωρίζει τον νικητή χωρίς ανθρώπινη παρέμβαση[2]. Με διάφορους συνδυασμούς νέων τεχνολογιών, ένα τραπέζι θα μπορούσε να προσφέρει εκπτώσεις σε συχνούς πελάτες, να αποθηκεύει το σκορ του κάθε πελάτη χωρίς ανθρώπινη παρέμβαση και άλλες παρόμοιες λειτουργίες.

## 2 Απαιτήσεις Συσκευής

### 2.1 Απαιτήσεις Χρήστη

Σε πολλές εργασίες αυτοματοποιημένων συστημάτων χρησιμοποιούνται τα UML (Unified Modeling Language) διαγράμματα για την καλύτερη αναπαράσταση λειτουργίας ενός συστήματος. Ένα από αυτά είναι το διάγραμμα use case το οποίο συμπεριλαμβάνει όλες τις αλληλεπιδράσεις μεταξύ του συστήματος και όλων των χρηστών [3]. Παρατίθεται ένα διάγραμμα για το τραπέζι μπιλιάρδου.



Εικ 1: Διάγραμμα περιπτώσεων χρήσης.

### 2.2 Απαιτήσεις Λειτουργιών

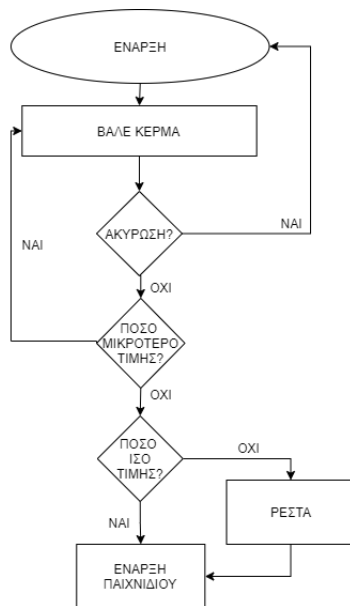
Το τραπέζι μπιλιάρδου θα πρέπει να είναι αυτοματοποιημένο στο μεγαλύτερο μέρος του. Οι λειτουργίες που θεωρούνται απαραίτητες αναλύονται σε:

1. Ένα μηχανισμό κερματοδέκτη στον οποίο εισάγεται το χρηματικό ποσό κάθε παιχνιδιού. Ο κερματοδέκτης θα είναι ικανός να αναγνωρίσει αν εισήλθε το σωστό ποσό και να επιστρέψει ρέστα αν αυτά υπάρχουν.

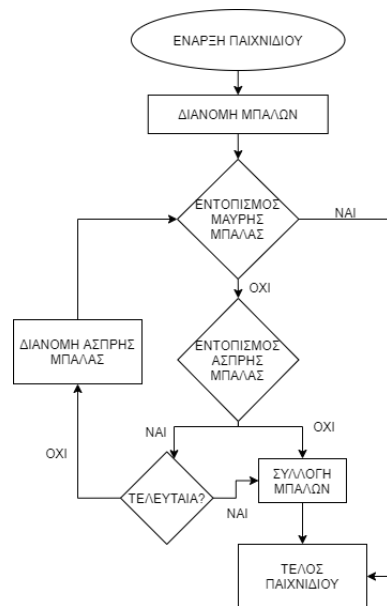
2. Ένα μηχανισμό χρώματος ο οποίος θα καταλαβαίνει τι χρώμα είναι η μπάλα που βρίσκεται μπροστά του και θα λειτουργεί αντίστοιχα.
  - (a) Οποιαδήποτε μπάλα εκτός της λευκής δεν εξέρχεται από τον μηχανισμό μέχρι την έναρξη καινούριου παιχνιδιού.
  - (b) Όταν εντοπίζεται η άσπρη μπάλα, αναδιανέμεται πάλι στον χρήστη.
  - (c) Με τον εντοπισμό της μαύρης το παιχνίδι τελειώνει. Έπειτα, όλες οι μπάλες δεν αναδιανέμονται συμπεριλαμβανόμενης και της άσπρης.
3. Ένα διακόπτη που αλλάζει την λειτουργία του τραπεζιού σε ελεύθερο παιχνίδι. Όλες οι μπάλες διανέμονται συνεχόμενα έως ότου αλλάξει πάλι η λειτουργία του τραπεζιού.
4. Ένα προσαρμογέα WiFi που κάθε 1 ώρα στέλνει πληροφορίες σε έναν server σχετικά με το πόσα παιχνίδια έχουν παιχτεί και αριθμό κερμάτων στον κερματοδέκτη.

### 3 Προδιαγραφές

Παρακάτω αναλύονται τα διαγράμματα ροής των δύο βασικών διαδικασιών του μηχανισμού. Η πρώτη διαδικασία προηγείται της δεύτερης και υπάρχει ένας διακόπτης για ελεύθερο παιχνίδι που δεν σχεδιάστηκε στα διαγράμματα γιατί αλληλεπιδρά μόνο με τον διαχειριστή του μηχανισμού.



Διάγραμμα ροής κερματοδέκτη.



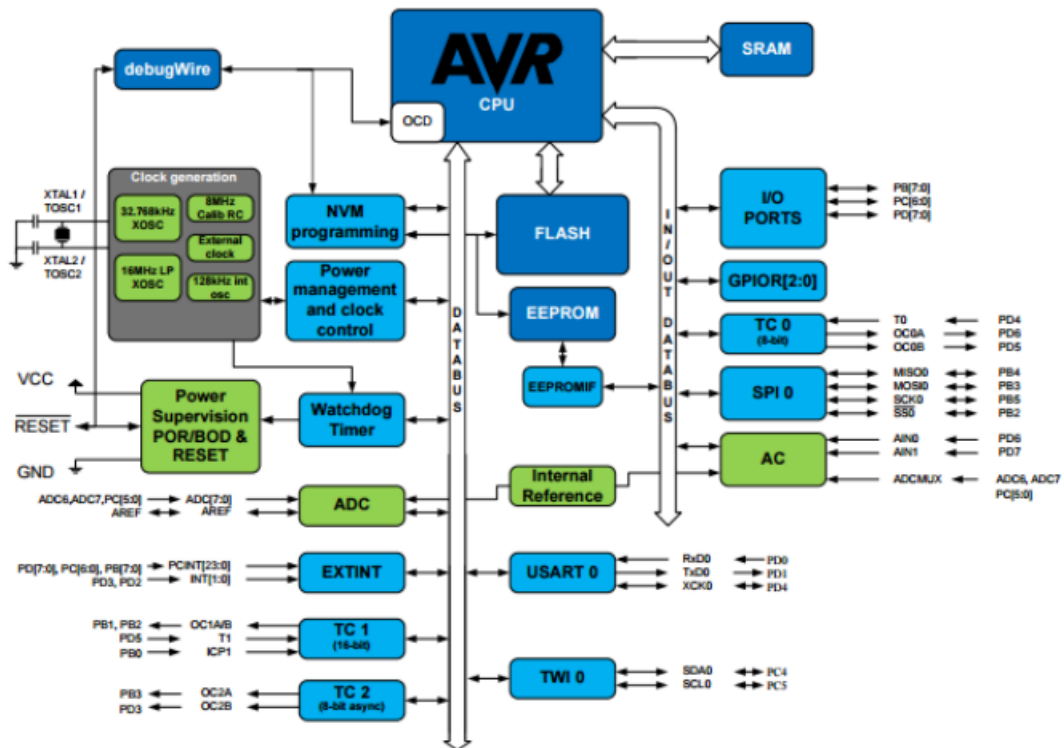
Διάγραμμα ροής παιχνιδιού.

## 4 Αρχιτεκτονική Συστήματος

### 4.1 Αρχιτεκτονική Υλικού

Για την υλοποίηση του συστήματος στο τραπέζι θα χρησιμοποιηθεί η πλακέτα Arduino Uno [1], που θα είναι υπεύθυνη για τον έλεγχο όλων των αισθητήρων και μοτέρ που θα χρειαστούν για την σωστή λειτουργία του παιχνιδιού. Η συγκεκριμένη πλακέτα χρησιμοποιεί τον μικροελεγκτή ATmega328P ο οποίος είναι υψηλής απόδοσης και ταυτόχρονα χαμηλής κατανάλωσης ενέργειας.

Χρησιμοποιεί την αρχιτεκτονική Harvard η οποία είναι μία προηγμένη RISC αρχιτεκτονική που διαχωρίζει την μνήμη προγραμμάτων και δεδομένων. Οι εντολές AVR είναι μήκους 16 bit και οι διευθύνσεις μνήμης μπορεί να είναι των 16 ή 32 bit. Ο μικροελεγκτής λειτουργεί με ένα ρολόι κρυσταλικού ταλαντωτή χαλαζία με συχνότητα 16 Mhz, και περιλαμβάνει 3 είδη μνήμης, μία SRAM των 2KB, μία EEPROM των 1KB και μία μνήμη Flash των 32KB από τα οποία τα 0.5KB χρησιμοποιούνται για το bootloader. Η πλακέτα Arduino Uno έχει 20 pin εισόδου/εξόδου τα οποία θα χρειαστούν για την σύνδεση με τους αισθητήρες.

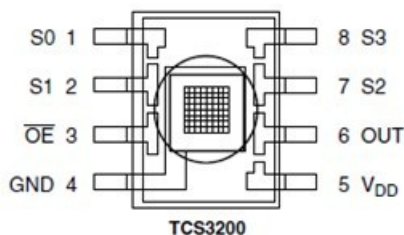


Εικ 2: Διάγραμμα αρχιτεκτονικής του Arduino

Θα χρειαστούν οι ακόλουθες περιφερειακές συσκευές για την δημιουργία ολόκληρου του συστήματος:

1. Έναν αισθητήρα χρώματος τύπου TCS3200

Χρησιμοποιεί 3 προγραμματιζόμενες φωτοδιόδους για να αναγνωρίσει κόκκινο, πράσινο, και μπλε χρώμα, πληροφορία την οποία εξάγει με ψηφιακό σήμα (τιμής 0-255 ανά χρώμα), που θα χρησιμοποιηθεί για αναγνώριση του χρώματος της μπάλας που πέρασε από μπροστά του. Έτσι, μπορούμε όχι μόνο να μετρήσουμε αν μπήκε μια μπάλα, αλλά και να αναγνωρίσουμε το χρώμα της



Πίνακας 1: Πίνακας Pin TCS3200

Pin	I/O	Περιγραφή
GND (4)		Γείωση Τροφοδοσίας
OE (3)	I	Ενεργοποίηση για σήμα εξόδου (ενεργό με 0)
OUT(6)	O	Σήμα εξόδου
S0, S1 (1, 2)	I	Είσοδοι επιλογής κλίμακας σήματος εξόδου
S2, S3 (7, 8)	I	Είσοδοι επιλογής τύπου φωτοδιόδου
VDD (5)		Παροχή ρεύματος

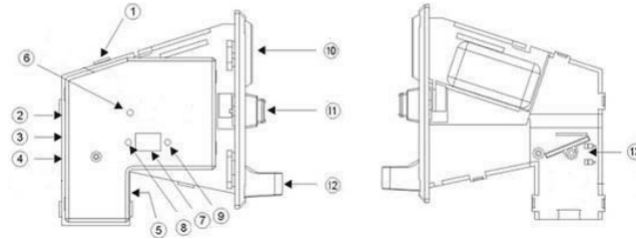
Για την επιλογή του χρώματος που θα ανιχνεύσει η φωτοδιόδος, χρησιμοποιούνται τα pin S2 και S3. Οι φωτοδιόδοι είναι συνδεδεμένοι παράλληλα, όποτε η επιλογή τους γίνεται με το εξής φίλτρο:

Πίνακας 2: Φίλτρο Επιλογής χρώματος

Τύπος Φωτοδιόδου	S2	S3
Κόκκινο	LOW	LOW
Μπλε	LOW	HIGH
Κανένα Φίλτρο (καθαρισμός)	HIGH	LOW
Πράσινο	HIGH	HIGH

## 2. 1 Μηχανισμός ανίχνευσης κερμάτων τύπου CH-926

Ολοκληρωμένο κύκλωμα με κωδικό CH 926. Αναγνωρίζει μέχρι 6 διαφορετικούς τύπους νομισμάτων και έχει ως έξοδο αναλογικό σήμα που μπορούμε να διαβάσουμε μέσω του μικροελεγκτή



Εικ 3: Διάγραμμα εισόδων/εξόδων κερματοδέκτη

- (1) φωτάκι LED ενδείξεων
- (2) υποδοχή 4 pin (VDD, GND και παλμός εξόδου
- (3) Διακόπτης για σήμα εξόδου NO. ή NC.
- (4) Διακόπτης για σήμα εξόδου ταχύτητας (20ms, 50ms, 100ms)
- (5) Ηλεκτρομαγνητική βαλβίδα 12V
- (6) κουμπί setup
- (7) LED οθόνη
- (8) κουμπί πρόσθεσης παλμών
- (9) κουμπί αφαίρεσης παλμών
- (10) υποδοχή νομισμάτων
- (11) κουμπί για την αφαίρεση κολλημένου νομίσματος
- (12) σχισμή για ρέστα

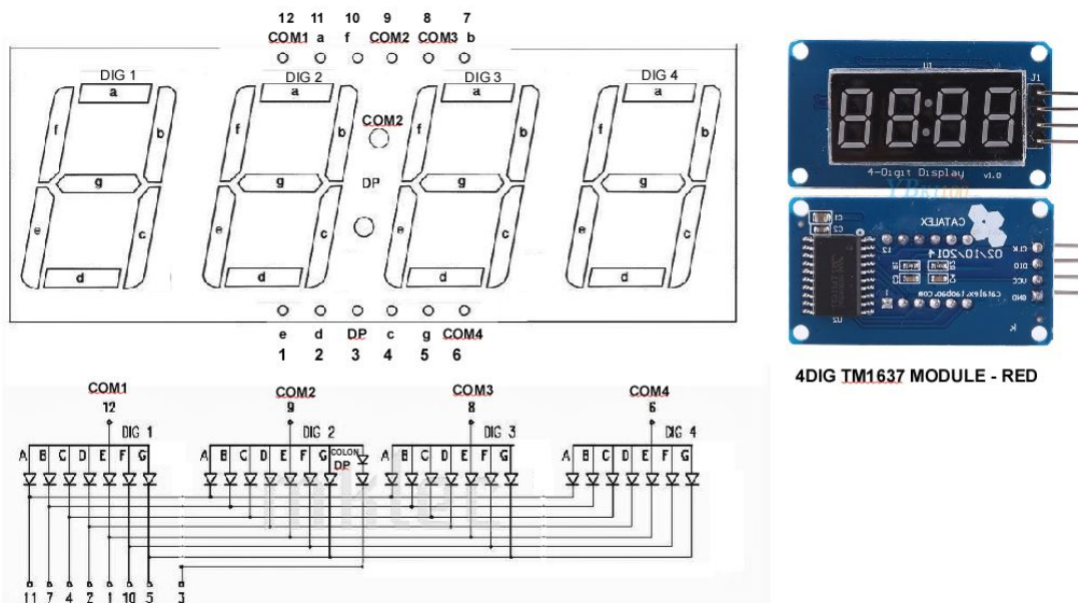
## 3. Οθόνη TM1637Display

Θα χρησιμοποιήσουμε ένα 7 segment display των 4 ψηφίων που έχει ήδη υποστήριξη από το Arduino IDE και χρησιμοποιεί 2 εισόδους ελέγχου από τις 12 που κανονικά θα ήταν απαραίτητες. Αυτό συμβαίνει επειδή εμπεριέχει ένα μικρό ενσωματωμένο στο πίσω μέρος του ολοκληρωμένου.



Εικ 4: Οθόνη TM1637

Η υλοποίηση του εσωτερικού κυκλώματος του TM1637 περιγράφεται παρακάτω:

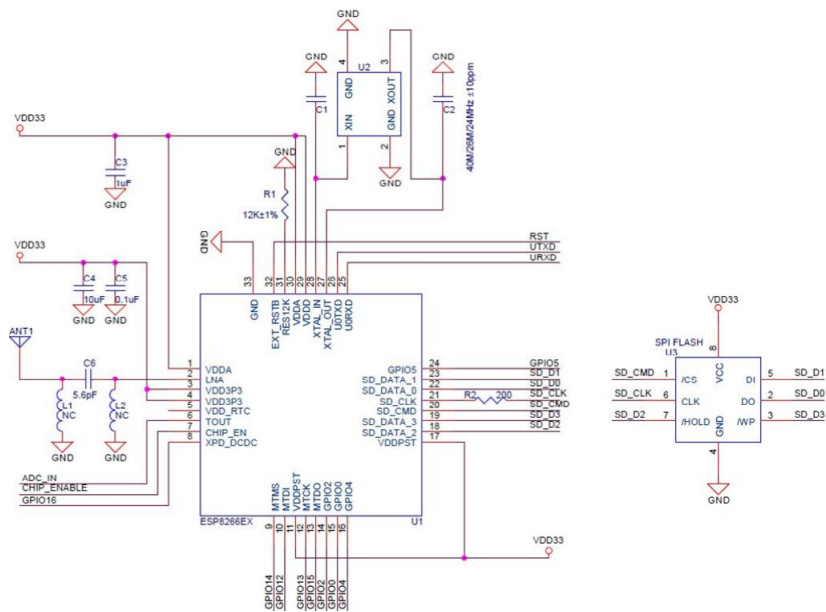


Εικ 5: Διάγραμμα εσωτερικού κυκλώματος TM1637

#### 4. Κάρτα wifi (ESP8266)

Μικρό ολοκληρωμένο κύκλωμα που ενσωματώνει όλη τη στοιβία πρωτοκόλλου TCP/IP, επιτρέποντας στον μικροελεγκτή να στείλει και να λάβει πληροφορίες. Θα χρησιμοποιηθεί και ένα μικρό κύκλωμα μετατροπής σήματος από 5V σε 3.3V καθώς η κάρτα δέχεται σήμα 3.3V.

Η υλοποίηση του εσωτερικού κυκλώματος του ESP8266 περιγράφεται παρακάτω:



Εικ 6: Διάγραμμα εσωτερικού κυκλώματος ESP8266

Τα εξωτερικά Pin του κυκλώματος φαίνονται στο παρακάτω σχήμα:



Εικ 7: Διάγραμμα εξωτερικού κυκλώματος ESP8266

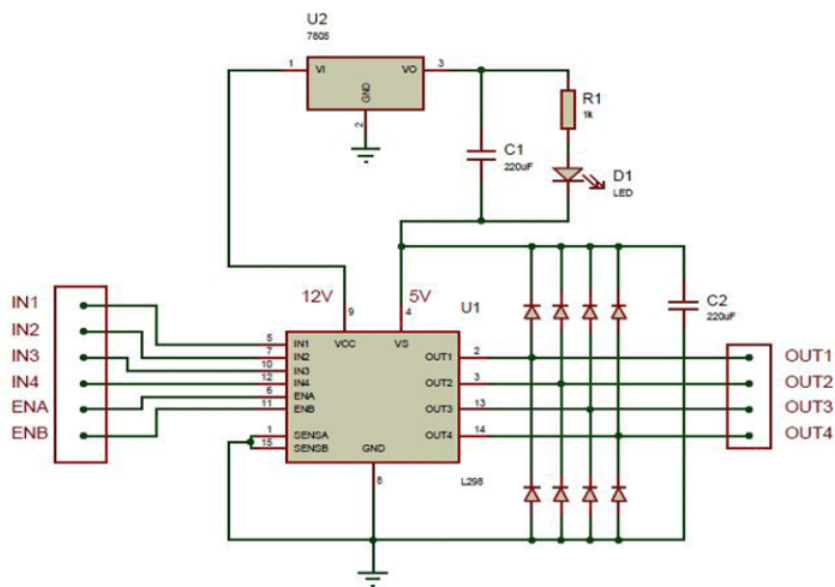


Πίνακας 3: Πίνακας Pin ESP8266-01

Pin	I/O Περιγραφή
GND (1)	Γείωση Τροφοδοσίας
GPIO2 (2) I	GPIO
GPIO0 (3) I	GPIO
RXD(4) I	UART0, λήψη δεδομένων από pin RXD
VCC (5)	Παροχή ρεύματος 3.3V
RST (6) I	εξωτερικό pin reset (ενεργό με 0)
CH-PD (7) I	pin ενεργοποίησης κυκλώματος
TXD (8) O	UART0, αποστολή δεδομένων στο pin RXD

5. Μονάδα οδηγού μοτέρ DC μοντέλο: L298N

Κατά την εκκίνηση μιας παρτίδας, πρέπει να διαμοιραστούν οι μπάλες στους παίκτες. Ένα μοτέρ θα ανοίγει μια μικρή "πόρτα" για να πέφτουν οι μπάλες σε μέρος προσβάσιμο προς το χρήστη. Τη δουλειά αυτή αναλαμβάνει ένα μοτέρ (για τις ανάγκες της εργασίας ένα 12V), που συνδέεται με το Arduino μέσω ενός ελεγκτή (μοντέλο L298N). Μέσω του ελεγκτή μπορούμε να ρυθμίσουμε καλύτερα το άνοιγμα και το κλείσιμο της πόρτας μετά την έναρξη του παιχνιδιού.



Εικ 8: Διάγραμμα εσωτερικού κυκλώματος L298N

Πίνακας 4: Πίνακας Pin L298N

Pin	I/O Περιγραφή
IN1 και IN2	I pin εισόδων για μοτέρ A (φορά περιστροφής)
IN3 και IN4	I pin εισόδων για μοτέρ B (φορά περιστροφής)
ENA	I Επίτρεψη σήματος PWM για μοτέρ A (ταχύτητα)
ENB	I Επίτρεψη σήματος PWM για μοτέρ B (ταχύτητα)
OUT1 και OUT2	O pin εξόδων για μοτέρ A (σύνδεση με μοτέρ)
OUT3 και OUT4	O pin εξόδων για μοτέρ B (σύνδεση με μοτέρ)
VCC 12V	παροχή ρεύματος 12V
VS 5V	παροχή ρεύματος 5V
GND	Γείωση Τροφοδοσίας

## 4.2 Αρχιτεκτονική Λογισμικού

Το Arduino Uno έχει έναν μικροελεγκτή, τον ATmega328P, άρα δεν εμπεριέχεται μαζί ένα λειτουργικό σύστημα. Χρησιμοποιείται κυρίως για απλές διεργασίες που γίνονται σε πραγματικό χρόνο και ενδύκνεται για τον έλεγχο συσκευών και ανιχνευτών αφού δεν υπάρχει καθυστέρηση από την ύπαρξη λειτουργικού συστήματος. Επιπλέον, στην περίπτωση διακοπής της παροχής ρεύματος, η συσκευή κάνει reset αντί να χρειάζεται τερματισμός λειτουργίας του λειτουργικού συστήματος. Το λογισμικό που θα χρησιμοποιηθεί για τον προγραμματισμό της συσκευής είναι το Arduino IDE 2.0, το οποίο εκδόθηκε το 2019 και είναι ανοιχτού κώδικα. Με αυτό το τρόπο η επέκταση του λογισμικού είναι εύκολη με χρήση βιβλιοθηκών της C++ και την τροποποίηση του ολοκληρωμένου για την δημιουργία πλήρως εξατομικευμένης πλακέτας Arduino.

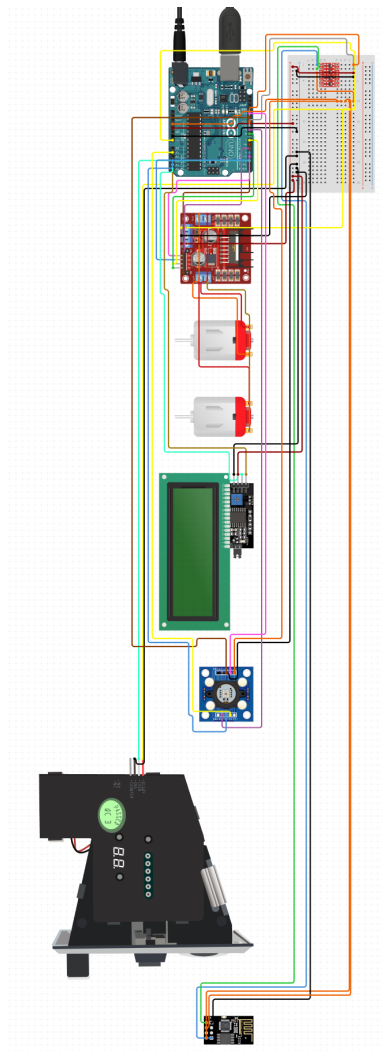
Οι μικροελεγκτές μπορούν να προγραμματιστούν με έναν εξωτερικό προγραμματιστή (π.χ. AVR-ISP), ο οποίος εγκαθιστά προγράμματα που ονομάζονται Sketches και είναι γραμμένα στο Arduino IDE. Αυτοί οι προγραμματιστές χρησιμοποιούνται αν απαιτείται η χρήση όλης της μνήμης flash ή η εξάλειψη καθυστερήσεων από το bootloader. Συνήθως, χρησιμοποιείται ο bootloader της πλακέτας, ο οποίος αποτελεί ένα μικρό κομμάτι λογισμικού για την αρχικοποίηση των ρυθμίσεων της πλακέτας και παρέχεται μαζί με την πλακέτα.

Τα Sketches έχουν την κατάληξη αρχείου .ino και χρειάζονται τουλάχιστον 2 συναρτήσεις για να λειτουργεί ο μικροελεγκτής, την `setup()` και `loop()`.

1. Η `setup()` είναι πρακτικά σα την συνάρτηση `main()` και χρησιμοποιείται για την αρχικοποίηση μεταβλητών και pin εισόδου/εξόδου.
2. Η συνάρτηση `loop()` εκτελείται μετά το τέλος της `setup()` και εκτελείται επ'άπειρον μέχρι η πλακέτα να υποστεί reset ή να σταματήσει η τροφοδοσία της. Πρακτικά έχει την λειτουργία της συνάρτησης `while(1)`.

## 5 Σχεσιασμός Συνολικού Συστήματος

Ο σχεδιασμός του συνολικού συστήματος φαίνεται στο παρακάτω διάγραμμα, το οποίο όμως δεν αναπαριστά απαραίτητα την τελική συνδεσμολογία αφού ο προγραμματισμός του δεν έχει ακόμα υλοποιηθεί. Παρόλα αυτά, τα pin εισόδων/εξόδων του κάθε υποσυστήματος έχουν συνδεθεί με τρόπο ώστε η λειτουργία τους να είναι θεωρητικά σωστή.



Εικ 9: Διάγραμμα συστήματος

## 6 Υλοποίηση Συστήματος

Η υλοποίηση του συστήματος θα δοθεί σε έναν προσομοιωτή του Arduino επειδή δεν έχουμε κάποια από τα ολοκληρωμένα συστήματα που θα χρησιμοποιηθούν (το Coin Acceptor και το Color Sensor). Αρχικά, το color sensor TCS3200 επιστρέφει τιμές συχνότητας όταν κάποιο αντικείμενο υπάρχει μπροστά του. Γι' αυτό το λόγο πρέπει να λάβουμε τις τιμές που καταλαβαίνει ο σένσορας για κόκκινο, πράσινο και μπλε χρώμα. Οι μεταβλητές που θα χρησιμοποιηθούν αντιστοιχούν στα pin των schematic που έχουμε αναλύσει παραπάνω. Επίσης, τα pin του arduino που θα χρησιμοποιηθούν φαίνονται στο σχέδιο του συνολικού συστήματος. Πρώτο βήμα είναι το upload ενός sketch για να ληφθούν οι τιμές των χρωμάτων: (εδώ μπαίνει το sketch του tcs config) Οι τιμές που θα λάβουμε κυμαίνονται μεταξύ κάποιων τιμών ανάλογα με την απόσταση της διόδου. Αφού λοιπόν λάβουμε τις τιμές συχνότητας, κρατάμε το πάνω και κάτω όριο για κάθε τιμή και το μεταφράζουμε σε τιμές RGB που κυμαίνονται μεταξύ 0-255. Οι τιμές που θα χρησιμοποιήσουμε είναι:

Πίνακας 5: Τιμές συχνότητας

Χρώμα	Κάτω Όριο	Πάνω Όριο
Κόκκινο	50	100
Πράσινο	200	500
Μπλε	300	1000

Έπειτα περιγράφεται η υλοποίηση του συστήματος. Οι main συναρτήσεις setup() και loop() υπάρχουν στο αρχείο (firmware) και το κάθε ολοκληρωμένο περιγράφεται στα 2 δικά του αρχεία header και source. Αυτό έχει γίνει για λόγους καθαρότητας. Αρχικά τα αρχεία (TCS3200) λαμβάνουν τις τιμές από τις φωτοδιόδους, τις μετατρέπουν σε RGB τιμές και τα όρια που έχουν τοποθετηθεί είναι RGB μικρότερο του 10 για το μαύρο χρώμα (έλλειψη χρώματος) και RGB μεγαλύτερο του 220 για λευκό χρώμα(ύπαρξη). Επίσης υπάρχει εντοπισμός της κάθε διαφορετικής μπάλας για τον έλεγχο του παιχνιδιού και την ενημέρωση των παιχτών όσων αφορά πόσες μπάλες έχουν απομείνει. Οι ενδεικτικές τιμές που έχουν χρησιμοποιηθεί είναι οι εξής:

Πίνακας 6: Τιμές χρώματος					
Αριθμός Μπάλα	Ομάδα	Κόκκινο Πράσινο Μπλε			
0	Άσπρη	Ουδέτερη	220	220	220
1	Κίτρινη	Μονόχρωμη	245	245	20
2	Μπλε	Μονόχρωμη	20	20	245
3	Κόκκινη	Μονόχρωμη	245	20	20
4	Μωβ	Μονόχρωμη	110	20	245
5	Πορτοκαλί	Μονόχρωμη	245	110	20
6	Πράσινη	Μονόχρωμη	20	245	20
7	Μπορντώ	Μονόχρωμη	110	20	20
8	Μαύρη	Ουδέτερη	5	5	5
9	Κίτρινη	Δίχρωμη	205	205	20
10	Μπλε	Δίχρωμη	20	20	205
11	Κόκκινη	Δίχρωμη	205	20	20
12	Μωβ	Δίχρωμη	90	20	205
13	Πορτοκαλί	Δίχρωμη	205	90	20
14	Πράσινη	Δίχρωμη	20	205	20
15	Μπορντώ	Δίχρωμη	80	20	20

Ταυτόχρονα για την προσομοίωση των παλμών από το Coin Acceptor χρησιμοποιείται ένα χειροκίνητο κουμπί που θα πατιέται αναλόγως το νόμισμα που προσωμοιώνεται. Στον παρακάτω πίνακα αναλύονται οι παλμοί για κάθε νόμισμα:

Πίνακας 7: Προσομοίωση Coin Acceptor

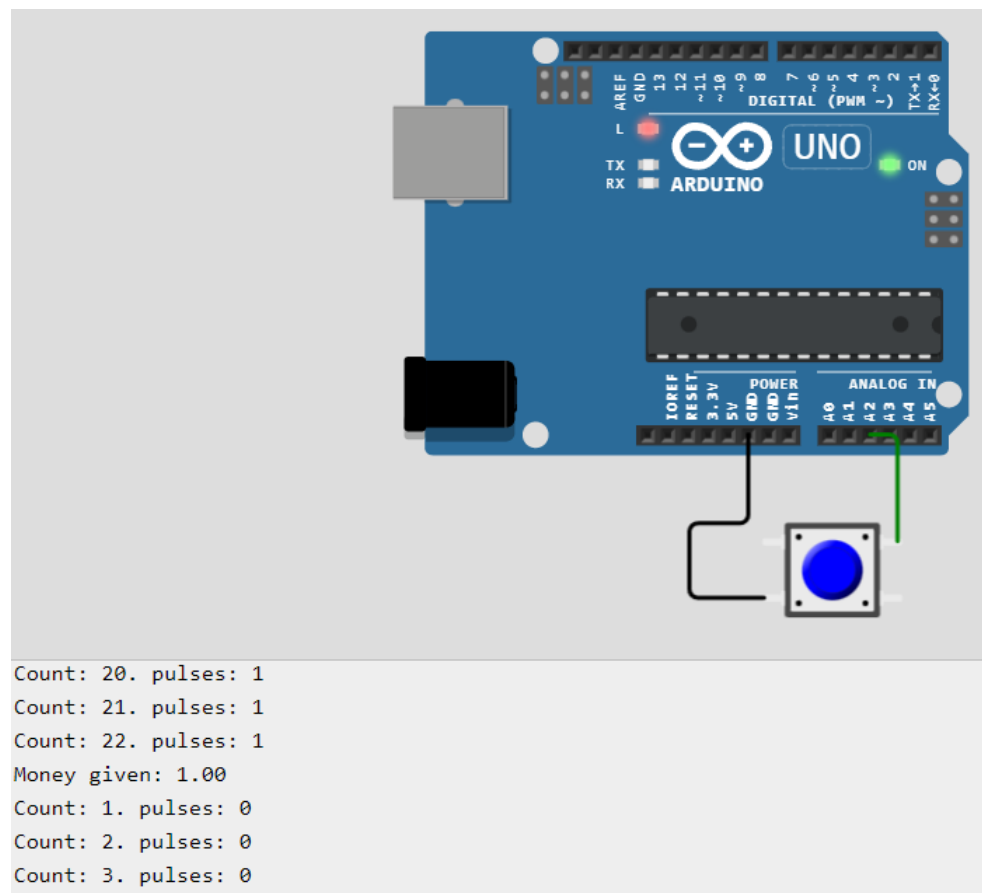
Παλμοί Νόμισμα	
1	2 Ευρώ
2	1 Ευρώ
3	50 Λεπτά
4	20 Λεπτά
5	10 Λεπτά

Στο τέλος κάθε παιχνιδιού, προσμετρούνται τα συνολικά χρήματα και παιχνίδια και στέλνονται σε έναν mySQL server μέσω του ESP8266.

## 7 Προσομοίωση λειτουργίας συστήματος

Το σύστημα θα προσομοιωθεί στο Wokwi Simulator επειδή δεν έχουμε τα εξαρτήματα που χρειαζόμαστε για την υλοποίηση του συστήματος. Λόγω αυτής της έλλειψης, έχουν δημιουργηθεί συναρτήσεις που παρέχουν την λειτουργία του συστήματος μέσω ενός κουμπιού και του Serial Monitor του Arduino. Τα βήματα αναλύονται παρακάτω:

1. Στην δεξιά μεριά της οθόνης υπάρχει πράσινο κουμπί που ξεκινάει την προσομοίωση.
2. Ένα counter αρχίζει να μετράει κάθε 0.3 δευτερόλεπτα και περιμένει παλμούς από το κουμπί που παρομοιάζουν την λειτουργία του κερματοδέκτη. Φαίνεται στην παρακάτω εικόνα η προσομοίωση λήψης ενός νομίσματος αξίας 2 ευρώ:



Εικ10: Προσομοίωση λήψης 1 ευρώ

3. Μετά την λήψη της αποδεκτής τιμής ( 5 ευρώ ), αρχίζει καινούριο παιχνίδι στο οποίο η σειρά που πέφτουν οι μπάλες είναι τυχαία. Ένα τέτοιο παράδειγμα φαίνεται παρακάτω:

```
Count: 23. pulses: 0
Count: 24. pulses: 0
Money given: 5.00
Game is starting!
Opening hatch A to release all balls...
Closing hatch A
Blue solid ball detected. Remaining solids: 6
white ball detected, opening hatch B
closing hatch B
Dark red solid ball detected. Remaining solids: 5
Yellow solid ball detected. Remaining solids: 4
Green solid ball detected. Remaining solids: 3
Black ball has been detected. Game Ended!
-----
Noone won :/
Total amount in machine: 5.00.Total games played: 1
```

Εικ 11: Προσομοίωση τυχαίου παιχνιδιού

4. Το σύστημα επιστρέφει στην αρχική του λειτουργία, περιμένοντας την λήψη της αποδεκτής τιμής για να ξεκινήσει ένα καινούριο παιχνίδι.

Το πρόγραμμα διατηρεί πληροφορίες για τον συνολικό αριθμό παιχνιδιών που έχουν παιχτεί όπως επίσης και τον συνολικό αριθμό παιχνιδιών που έχουν παιχτεί. Επίσης με το ESP8266 κάθε μερικές ώρες αυτά τα στοιχεία μπορούν να σταλθούν σε μία βάση δεδομένων μέσω wifi.

## Αναφορές

1. Prakash, Amrita and Singh, Ajit.:  
Development of an Arduino-Based Embedded System  
*Available at SSRN 3422103 (2019)*
2. Ko, Mink-Kuan.:  
*Design and Implementation of Billiard Auto-scoring System.*  
Master thesis in Department of Electrical Engineering, National Chungcheng University, Taiwan (2006)
3. Kaur, Aman.:  
Application of UML in real-time embedded systems.  
*International Journal of Software Engineering & Applications (IJSEA) 3.2 (2012)*