



חיל האוויר  
הולץ

בית הספר  
הרב תחומי  
רשת תיכונים ומכללות מקבוצת עמל

שם הפרוייקט: ירייה למטרה על ידי חיילים וארדואינו

שם המגיש: סידני סבן (עולה חדש!)

כיתה: יב 4

תעריך הגשה: 12/06/2018

שם המנחה: עודד נחמה



## תוכן עניינים

3.....	מבוא.....
4.....	תרשים מלבני.....
4.....	הסבר על תרשים מלבני.....
5.....	תרשים חשמלי של המשחק.....
6.....	תרשים חשמלי של הרובה לייזר.....
7.....	רשימת רכיבים.....
8.....	בניית הפרויקט – הרובה.....
10.....	בניית הפרויקט – המשחק.....
12.....	תרשים זרימה של הקוד.....
13.....	הסבר על תרשים זרימה לקוד.....
14.....	הקוד.....
18.....	הסבר על הקוד.....
19.....	איתור תקלות.....
20.....	הסבר טכני על הרכיבים המרכזים – Arduino.....
22.....	הסבר טכני על הרכיבים המרכזים – LCD.....
23.....	הסבר טכני על הרכיבים המרכזים – Laser.....
23.....	הסבר טכני על הרכיבים המרכזים – Micro-Servo.....
25.....	ביבליוגרפיה.....
26.....	דף צהרה.....

## מבוא

מטרת פרויקט זה היא להפוך את המשחק החיים האמיתיים של סדרת האנימציה  
הישנה 'Lucky Luke',

באמצעות מכשירים אלקטרוניים ורכיבים עם ארדואינו אוננו.

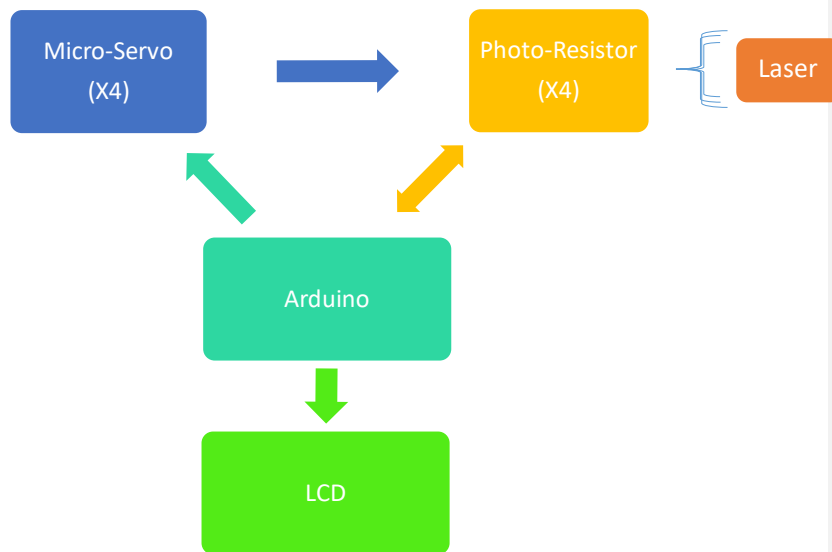
מטרת המשחק היא להכות את המטרות בזמן המוקצב, את הציון הוא הציג על מסך  
נוזלי.

אם אתה מצליח לירות ארבע מטרות בזמן, זה יוצג על המסך 'You Win!' .

אם לא תצליח לירות בארבעת היעדים בזמן, היא תופיע על המסך 'Game Over!' .

לצורך הפרויקט, בניתי וקניתי את כל הרכיבים והחומר לבדי.

## תרשים מלבני

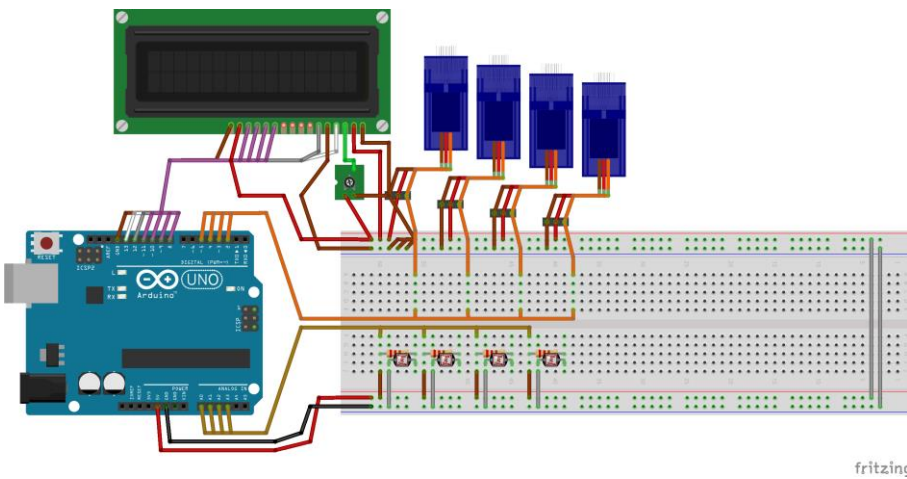


## הסבר על תרשים מלבני

כפי שניתן לראות בתרשים לעיל, הארדואינו נמצא במרכז הפיקוד, כאשר הוא שולח את המידע למנוע כדי לזוז ומקבל מהמקלט את המידע של הלייזר, הוא שולח טקסט מידע במסך.

## תרשים חשמלי

של המשחק:



fritzing

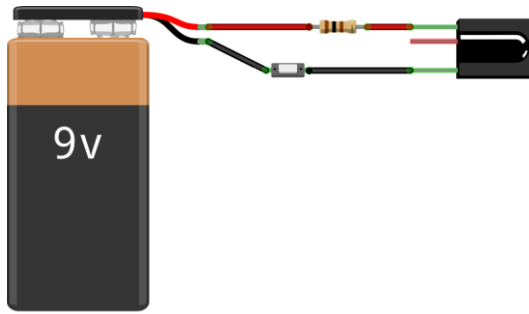
## חיבורים לפינים:

<b>Motors:</b>	
Motor 1	D2
Motor 2	D3
Motor 3	D4
Motor 5	D5

<b>Sensors:</b>	
Sensor 1	A0
Sensor 2	A1
Sensor 3	A2
Sensor 4	A3

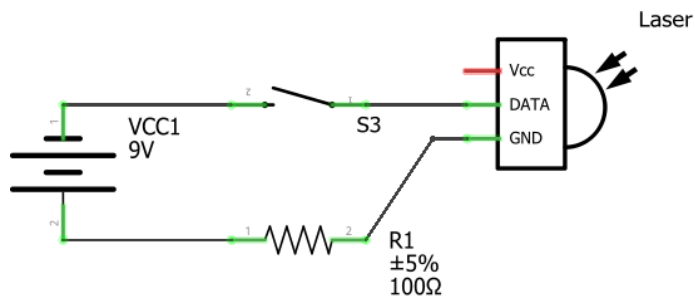
<b>Liquid Crystal:</b>	
VSS	Gnd
VDD	+5v
Vo	Potentiometer
D7	8
D6	9
D5	10
D4	11
RS	13
RW	Gnd
E	12
A	+5v
K	Gnd

של הרובה לייזר:



fritzing

חיבורים:

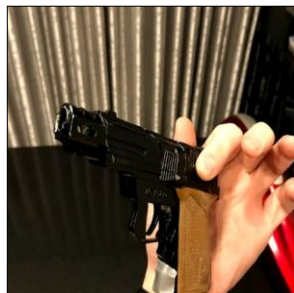
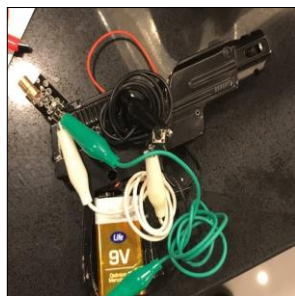


## רשימת רכיבים

1. Micro-Servo – x4
2. Photo-Resistor – x4
3. Resistor 100 Ohm – x5
4. Liquid Crystal – x1
5. Potentiometer – x1
6. Laser Diode - x1
7. Button – x2
8. Arduino Uno – x1
9. Battery 9v – x2

בניית של הפרוייקט

בניית הרובה לייזר:



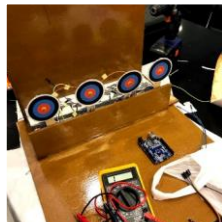
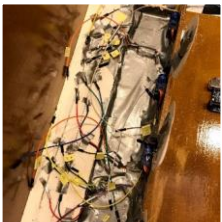
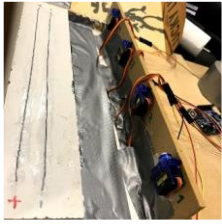


### הסבר של התמונות:

אני חותר את המתכת של האקדח שקניתי מראש עבור הפרויקט, כדי להיות מסוגל להזין את כל הרכיבים כולל לייזר.

לאחר שהצלחתי להיכנס לכל הרכיבים, אני מבצע מספר בדיקות וכאשר אני בטוח שהכל פועל כראוי, אני מכניס קצף בידוד ואני סוגר את המכסה מתכת פלסטיק עם דבק חזק.

## בניית המשחק:



## הסבר על התמונות:

חשבתי ועיצבתי את כל ביצוע הפרויקט, התחלתי על ידי חיתוך לוחות של קרטון דחוס, ולאחר מכן למדוד אותם דבק אותם יחד על ידי יצירת מבנה מוצק.

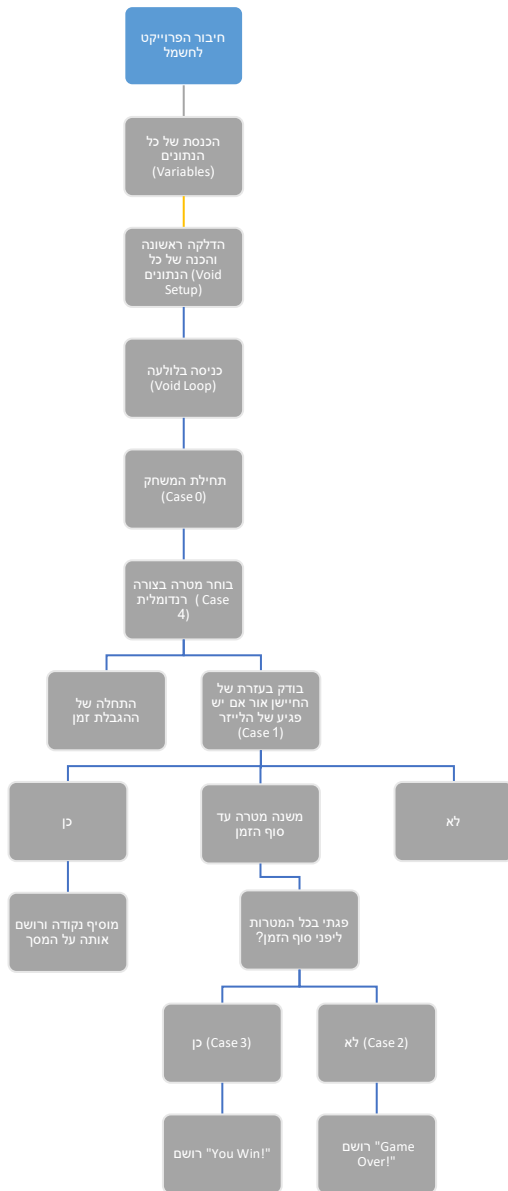
ואז התחלתי על ידי דבק חזק המנועים ריתוך המלחינים הראשונים.  
ואז להפוך את המטרות שהצמדתי את המנועים.

בעקבות זה, התחלתי לצייר את הקרטון דחוס על ידי נותן לו מרקם יפה יותר של צבע עץ ולהחיל שרף כי לרכך את המרקם.

לאחר מכן הודפסתי את התמונות שאני צריך עבור העיצוב והחיל אותו על הפרויקט.

אחר עידכון התוכנית, אני גזור והוסיף מסך ורמקול לפרוייקט.

## תרשים זרימה של הקוד



## הסבר על תרשים זרימה של הקוד

מרגע שהפרויקט מחובר לחשמל, הקוד מתחיל על ידי כתיבת המשתנים ועל ידי החלל הראשון, הוא מאתחל את כל המלחינים, ולאחר מכן על ידי במקרה הראשון, הוא מאותחל את מנועי סרוו, ואז זה שינויים של תיבת החל אקראיות כדי להיות מסוגל לבחור מנוע כי יתחיל את המשחק

המשחק מתחיל, השחקן חייב לכוון אל המטרה (נגד אור) עם אקדח לייזר

אם היעד מקבל את המידע שיש לו המשחק מתחיל, השחקן חייב לכוון אל המטרה (נגד אור) עם אקדח לייזר בזמן המוקצב

אם היעד מקבל את המידע כי הוא נגע, זה מוסיף את הציון למסך ולאחר מכן ממשיך לשנות את היעד.

אם היא לא נגעה, היא ממשיכה לשנות את היעד שלה עד סוף הזמן המוקצב.

אם השחקן פגע היעד היטב לפני שנגמר הזמן, יצג על המסך "You Win!"

אם השחקן לא הצליח להכות מספר היעד מספיק בזמן שהוקצה, יצג על המסך "Game Over!"

## TIKZ

```
1 // SERVO pin definitions
2
3 #include <Servo.h>
4 #include <LiquidCrystal.h>
5
6 LiquidCrystal lcd(13, 12, 11, 10, 9, 8);
7
8 // Pin Assignment
9 const int targetLightSensor_0 = A0;
10 const int targetLightSensor_1 = A1;
11 const int targetLightSensor_2 = A2;
12 const int targetLightSensor_3 = A3;
13
14 const int targetServoPin_0 = 2;
15 const int targetServoPin_1 = 3;
16 const int targetServoPin_2 = 4;
17 const int targetServoPin_3 = 5;
18
19 const int MAX_LIGHT_ALLOWED = 75;
20 const int MAX_GAME_SCORE = 5;
21 const int MAX_TIME_ALLOWED = 8;
22
23 // Define Attached Components
24 Servo targetServo_0, targetServo_1, targetServo_2, targetServo_3;
25
26 // Servo Positions
27 const int TARGET_UP = 80; // Awake
28 const int TARGET_DOWN = 0; // Sleep
29
30 // Game States
31 const int GAME_STATE_RESET = 0;
32 const int GAME_STATE_RUN = 1;
33 const int GAME_STATE_GAME_OVER = 2;
34 const int GAME_STATE_WIN = 3;
35 const int GAME_STATE_TARGET = 4;
36
37 int targetLightValue;
38
39 Servo targetServoList[4] = { targetServo_0, targetServo_1, targetServo_2, targetServo_3 };
40 int targetSensorList[4] = { targetLightSensor_0, targetLightSensor_1, targetLightSensor_2, targetLightSensor_3 };
41
42
```

```
43 // Game Variables
44 int randomNumber;
45 int currentGameState;
46 int gameScore = 0;
47 int scoreConfirm = 0;
48 int timeElapsed = 0;
49 int timeInMilli = 0;
50
51 void setup() {
52   Serial.begin(9600); //Begin serial communication
53   lcd.begin(16, 2);
54
55   randomSeed(analogRead(0));
56   currentGameState = GAME_STATE_RESET;
57
58   targetServo_0.attach(targetServoPin_0);
59   targetServo_1.attach(targetServoPin_1);
60   targetServo_2.attach(targetServoPin_2);
61   targetServo_3.attach(targetServoPin_3);
62 }
63
64 void loop() {
65   switch(currentGameState) {
66     // Set targets to down position and initialize game variables
67     case GAME_STATE_RESET:
68       for(int i = 0; i < 4; i++) {
69         targetServoList[i].write(TARGET_DOWN);
70       }
71       gameScore = 0;
72       timeElapsed = 0;
73       currentGameState = GAME_STATE_TARGET;
74       break;
75     case GAME_STATE_TARGET:
76       // Pick a target
77       randomNumber = random(1, 4);
78       targetServoList[randomNumber].write(TARGET_UP);
79       targetLightValue = 0;
80
81       currentGameState = GAME_STATE_RUN;
82       break;
83     case GAME_STATE_RUN:
```

```

84
85 // Check reset button state, if on then change currentGameState to GAME_STATE_RESET
86
87 // check sensor value: if hot then add 1 to score and change game state to GAME_STATE_TARGET
88 targetLightValue = analogRead(targetSensorList[randNumber]);
89
90 if(targetLightValue > MAX_LIGHT_ALLOWED) {
91
92     if(scoreConfirm > 3) {
93         gameScore++;
94         scoreConfirm = 0;
95         targetServoList[randNumber].write(TARGET_DOWN);
96         currentGameState = GAME_STATE_TARGET;
97     } else {
98         scoreConfirm++;
99     }
100
101 }
102 // Keep track of time: if too long then change game state to GAME_STATE_GAME_OVER
103 timeInMili = millis();
104 if((int)(timeInMili / 100) % 10 == 0) {
105     timeElapsed++;
106 }
107
108 if(timeElapsed >= MAX_TIME_ALLOWED) {
109     currentGameState = GAME_STATE_GAME_OVER;
110 }
111
112 // Keep track of score: if higher than 30 then change game state to GAME_STATE_WIN
113 if(gameScore >= MAX_GAME_SCORE) {
114     currentGameState = GAME_STATE_WIN;
115 }
116
117 Serial.print("Target = ");
118 Serial.print(randNumber);
119 Serial.print(" - ");
120 Serial.print(targetServoList[randNumber].read());
121 Serial.print(" - light: ");
122 Serial.print(targetLightValue);
123 Serial.print(" - score: ");
124 Serial.print(gameScore);
125 Serial.print(" - timeElapsed: ");
126 Serial.println(timeInMili);
127

```



```
128     lcd.setCursor(0, 0);
129     lcd.print("Score: ");
130     lcd.setCursor(10, 0);
131     lcd.print(gameScore);
132
133     delay(500);
134
135     break;
136 case GAME_STATE_GAME_OVER:
137     // Play loser sound
138     Serial.println("Game Over");
139     lcd.setCursor(0, 1);
140     lcd.print("Game Over");
141     delay(20000);
142     break;
143 case GAME_STATE_WIN:
144     // Play winner sound
145     Serial.println("Player Wins");
146     lcd.setCursor(0, 1);
147     lcd.print("You Won");
148     delay(20000);
149     break;
150
151 //Initialize the Servo
152 //targetServo_0.write(TARGET_DOWN);
153 //delay(500);
154
155 //Reading the Sensors values
156 //targetLightValue = analogRead(targetLightSensor_0);
157
158 //Writing the Sensors values
159 //Serial.print("Sensor 1 = ");
160 //Serial.println(targetLightValue);
161 //delay(200);
162
163 //if (targetLightValue > 90) {
164 //    targetServo_0.write(TARGET_UP);
165 //    delay(500);
166 //}
167
168 //else {
169 //    Serial.print("waiting for laser");
170 //}
171 }
172
173
174
175
```

## הסבר על הקוד

- שורות 3-50: הכנסת של כל הנתונים (Variables).
- שורות 51-62: הדלקה ראשונה והכנה של כל הנתונים (Void Setup).
- שורות 64-150: כניסה בלולעה (Void Loop).
- שורות 67-74: הכנת המשחק למצב התחלתי (Case 0).
- שורות 75-82: מקבל מספר רנדומלי ולפי המספר הזה בוכר את המטרה הראשונה של המשחק (Case 4).
- שורות 83-135: מתחיל את ספירת הזמן ובודק בכל מטרה שהיא במצב פעילה אם היא קיבלה את האור של הרובה לייזרת אם כן היא מוסיפה נקודה ורושמת אותה על המסך וממשיכה להפעיל את המטרה הבאה עד סוף הזמן, אם לא היא פשוט ממשיכה להפעיל את מטרה הבאה עד סוף הזמן. ואם השחקן מצליח לפגוע במספיק מטרות ליפני סוף הזמן: הובר ל(Case 3). אם הוא לא הספיק בזמן הוא הובר ל(Case 2). כל השורות אלו זה (Case 1).
- שורות 136-142: רושם על נמסך "Game Over!" (Case 2).
- שורות 143-149: רושם על נמסך "You Win!" (Case 3).
- שורות 152-172: שורות אלו נמצאים רק במצב טסט של הפרוייקט.

### שורות חשובות:

- שורה 77 ו55: אלו שורות של פקודות כדי להפעיל את קבלת של המספר הרנדומלי.
- שורות 103-105: אלו פקודות של הברת הזמן במילי-שניות לשניות.
- שורה 19 ו90: אלו פקודות של בדיקת הלייזר.

## איתור תקלות

במהלך בניית הפרויקט מצאתי את עצמי במספר דילמות ובעיות.

כשהתחלתי לבנות את אקדח הלייזר (שהיה הדבר הכי קשה לעשות), היה לי הרעיון שכאשר השחקן משתמש בלייזר, עליו ללחוץ על ההדק שבו הוא מפעיל את לייזר.

הבעיה היתה שזה היה מסובך מאוד למצמץ את המתכת להחזיר את המרכיבים בחזרה פנימה בלי שהם זז ואת ריתוך נשאר.

מצאתי את עצמי מתמודד עם בחירה, כפתור הלחמה לא מפסיק, אז הייתי צריך להוסיף כפתור חיצוני לעשות שוב את החוטים.

בעיה נוספת שנתקלתי בה היא שבבסיס רציתי להוסיף רמקול, אבל הבעיה, שבסופו של דבר, הכל עבד בצורה מושלמת לכמה שניות ופתאום כלום.

לקח לי זמן רב להבין מהיכן באה הבעיה, ומה שסיכמתי היה שהבעיה באה רמקול, אז החלטתי להסיר אותה מהפרויקט, אבל בשביל הטופס אני השארתי אותו במקומות המתוכננים שלו.

Commenté [1]:

## הסבר טכני של הרכיבים המרכזים

:Arduino Uno

### **מה זה ארדואינו ומה אפשר לעשות איתו?**

ארדואינו הוא לוח פיתוח ופלטפורמה בקוד פתוח שכולל סביבת פיתוח קלה ונוחה למשתמש.

הוא מיועד למהנדסי אלקטרוניקה, סטודנטים, אומנים, חובבי רובוטיקה, ילדים וכל מי שמתעניין באלקטרוניקה ומחשבים.

כדי להפעיל ארדואינו אין צורך בידע מוקדם, והרעיון המרכזי הוא לשחק איתו ולבנות פרויקטים ותוך כדי כך ללמוד על עולם האלקטרוניקה והמחשבים. זוהי דרך מצוינת ללמוד בעיקר בזכות הקלות של השימוש בו.

השימושים בארדואינו מוגבלים רק ע"י הדמיון שלכם: הוא יכול להיות מערכת השקיה ממוחשבת, שעון מעורר, שרת אינטרנט, מערכת בית חכם, ואפילו רובוט!! ואלו רק חלק מהשימושים.

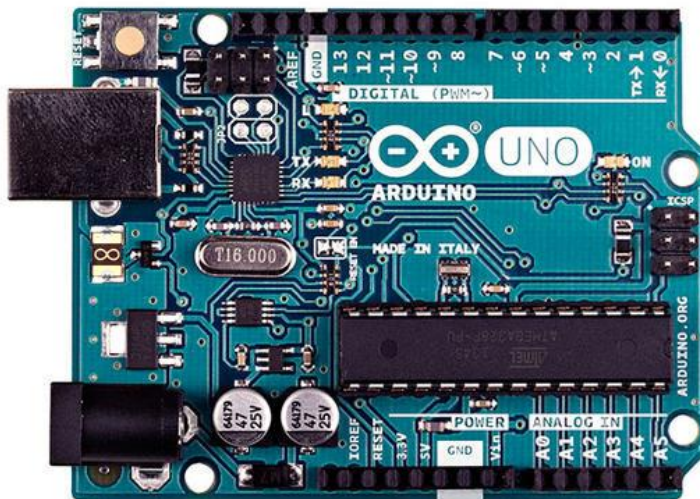
הדרך הטובה ביותר לחשוב על ארדואינו הוא כמוח קטן... המוח של האדם מקבל קלט מהחושים שלנו(עיניים, אוזניים, אף) מעבד את הנתונים שהוא קיבל ועושה איתם משהו, לדוגמה מפעיל איברים בגוף כמו ידיים ורגליים.

הארדואינו מקבל קלט מחיישנים(חיישן קרבה, חיישן אור, כפתור) מעבד את הנתונים, ולפי התוכנה שכתובה בו מחליט האם ואיך להפעיל פלט מסוים כמו נורה, מנוע או מסך.

לכן הוא נקרא בקר כי הוא מבקר על הפעולות שנעשות על פי התוכנה שאנחנו מכניסים אליו.

ארדואינו הוא מחשב קטן שיכול להתממשק אל העולם הפיזי. הוא חלש יותר ממחשב ביתי אבל יש לו גישה אל רכיבים חיצוניים בניגוד למחשב הביתי. גם המחשב הביתי מחובר אל אמצעי קלט ופלט כמו מקלדת ומסך אבל בארדואינו האפשרויות הן בלתי מוגבלות.

אין לו שימוש ספציפי מסוים אלא הוא מכונה כללית עם אינספור שימושים ואפליקציות. הוא מזכיר במקצת את הסמארטפון שהוא היבריד בין בקר למחשב, אבל גם הסמארטפון מוגבל במה



שהמשתמש יכול לעשות איתו. ארדואינו הוא לא מוצר מוגמר אלא כלי לבניית מוצרים ופרויקטים.

אם אתם אוהבים לקנות מוצר מוכן כמו סמארטפון שמישהו בנה בשבילכם ורוב האפשרויות סגורות בפניכם אז זה גם בסדר גמור אבל הארדואינו הוא שונה – אתם קובעים איך להשתמש בו ומה לעשות איתו. יש לזה מחיר כמובן שהוא הקשיים הראשוניים להבין איך הוא עובד אבל הסיפוק של לבנות משהו בעצמך הוא כל כך גדול שזה שווה את כל הטרחה.

כמו כל מחשב, את הארדואינו ניתן לחלק לחומרה ותוכנה – החומרה היא כל הרכיבים הפיזיים שנמצאים על המעגל המודפס, והשפה שהם מבינים היא חשמל.

התוכנה היא רצף הוראות שכתוב בשפת תכנות ארדואינו (מבוססת על ++C והיא אומרת ללוח הארדואינו מה לעשות). המשתמש הוא זה שכותב את התוכנה.

הארדואינו הוא לוח פיתוח ופלטפורמת חומרה בקוד פתוח ולכן שרטוט המעגלים וכל הקבצים נגישים לכל אחד באופן חופשי. עובדה זו הפכה אותו לפופולרי במיוחד בעולם המייקרים ולמעשה מאפשרת לכל אדם לשנות את העיצוב שלו ואפילו לייצר תואם ארדואינו משלו ולמכור אותו.

LCD:

## איך פועל מסך LCD?

מסך – LCD (Liquid Crystal Display) תצוגת גביש נוזלי (הוא מסך שתצוגתו פועלת על פי העקרונות הפיזיקליים של גביש נוזלי. רבים ממסכי הטלוויזיה, מסכי המחשב, תצוגות בתא הטייס, שעונים, מחשבוני הנמצאים בשימוש כיום פועלים על עקרון זה. מסכים כאלו הם לרוב נוזלים, ניידים, קלים ואמינים. בנוסף, צריכת האנרגיה שלהם נמוכה, כך שהם יכולים לפעול בעזרת סוללות. חסרונם העיקרי הוא בכך שזווית הצפייה בהם מוגבלת.



גביש נוזלי הוא מצב של חומר בעל תכונות שחלקן מתאימות לנוזלים וחלקן מתאימות למוצקים. מדובר בחומרים שהמולקולות שלהם דמויות "חוט". למשל, גביש כזה יכול להימצא במצב בו המולקולות חופשיות לנוע כמו בנוזל, אך מסתדרות בכיוון אחד כמו במוצק.

אחד ממצבי הצבירה הנפוצים ביותר של גבישים נוזליים נקרא "נמטי". במצב זה כאשר מפעילים שדה חשמלי המולקולות מסתדרות על ציר אחד מקביל בקירוב.

## סידור המולקולות במצב צבירה נמטי

במסכי ה-LCD – כל פיקסל (נקודה המרכיבה את התמונה) מורכב בד"כ משכבת מולקולות המונחות בין שתי אלקטרודות שקופות, ובין שני מקטבים. מקטב הוא התקן המאפשר רק לאור המקוטב בכיוון מסוים לעבור דרכו. (קיטוב הוא כיוון השדה החשמלי של האור). אור שמקוטב בכיוון אנכי לכיוון זה יחסם על ידי המקטב. כך, ללא הגביש הנוזלי, אור שצליח לעבור דרך המקטב הראשון ייחסם על ידי המקטב השני, המכוון לחסום אור בכיוון האנכי לכיוון המקטב הראשון.

שתי האלקטרודות הנוגעות בגביש הנוזלי גורמות למולקולות שלו להסתדר בכיוון מסוים. ברוב המסכים שתי האלקטרודות מסדרות את המולקולות כך שכיוונן משתנה בהדרגה על פני עובי הגביש הנוזלי. כך קורה שאור המגיע מצד אחד של האלקטרודה משנה את כיוונו בהדרגה עד שיגיע לצד השני. סידור המולקולות הוא כזה, שכאשר אור מגיע לצד השני הוא אנכי לכיוונו המקורי, וכך יוכל לצאת מהמקטב השני. הפיקסל הנמצא על המסך אשר דרכו יצא האור יהיה בצבע לבן. כאשר יופעל שדה חשמלי על הגבישים הנוזליים בעזרת האלקטרודות, יישר הפיתול של הגבישים, המולקולות לא ישנו את כיוונם בהדרגה לכיוון אנכי, והאור לא יקוטב ב-90 מעלות כפי שקרה קודם. לכן, האור לא יעבור את המקטב השני והפיקסל על גבי המסך לא יואר. זהו הצבע השחור.

בעזרת שינוי השדה החשמלי אשר מקבל הגביש ניתן לגרום לחלק מסוים של האור לעבור את המקטב וכך ניתן להשיג כמעט כל גוון שרוצים. כל פיקסל בנוי משלושה תאים עם מסננים בצבעים אדום, ירוק וכחול ועל ידי שילוב שלושת צבעי היסוד הללו ניתן להציג כל צבע.

## :Laser

### מבנה לייזר דיודה

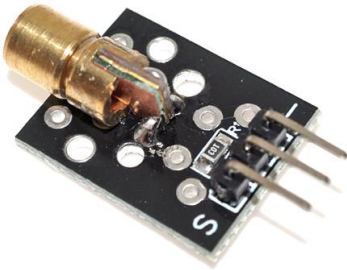
לייזר דיודה הוא לייזר מיניאטורי שממדיו קטנים מ-1 מילימטר. ההגברה בלייזר דיודה היא גבוהה ביותר, ולכן מספיק תווך פעיל קצר לקבלת לזירה. אורכים מקובלים של התווך הפעיל נמצאים בתחום: 100-1000 מיקרון.

המבנה הבסיסי של לייזר דיודה כולל רצועה של תווך פעיל שממדיה:

- רחב: 1-10 מיקרון,
- אורך: 200-300 מיקרון,
- גובה: 50-2000 אנגסטרם, שהם 5-200 ננומטר.

למעשה, מנפח של כ- 50 מילימטרים מעוקבים, ניתן להפיק קרינה רציפה בהספק של: 100 מיליוואט.

ניתן להשתמש במהוד כה קצר, עקב ההגברה הגבוהה) מעל 100 (cm-1) שניתן למצוא בחומרים אלו.



## :Micro-Servo

### מהו מנוע סרוו?



מנוע סרוו הוא מנוע זרם ישר (DC Motor) בעל מערכת תמסורת פנימית של גלגלי שיניים ובקרה אלקטרונית על מיקום המנוע. מה שמיחד מנועי סרוו היא העובדה שהם אינם מסתובבים בצורה חופשית כמו מנועי DC, אלא נעים על פי זווית – לרוב בין 0 ל-180 מעלות.

מנועי סרוו פועלים בחוג סגור, כלומר הינם בעלי בקרה על מיקום המנוע, ובעלי יכולת תיקון פערים מהמיקום הרצוי.

### שימושים שונים למנועי סרוו ברובוטיקה

מנועי סרוו נמצאים בשימוש בסוגים רבים מאד של רובוטים ובניהם זרועות רובוטיות, מכוניות הנשלטות בשלט רחוק, רובוטי-רכב, מטוסים ומסוקים (לשליטה על זווית הכנף \ רוטור). ישנן סיבות

רבות לכך שמנועי סרוו נפוצים כל כך באפליקציות רובוטיקה, וביניהן קלות השליטה במנועי סרוו, דרישות האנרגייה הנמוכות (יעילות), הכח הגבוה, רמת מתח TTL, והגודל והמשקל הנמוכים.

### יתרונות וחסרונות של שימוש במנועי סרוו

מנועי סרוו שימושיים מאד עבור אפליקציות רובוטיקה, בשל סיבות רבות:

- לרוב מנועי סרוו הינם מנועים בעלי גודל פיזי קטן
- מנועי סרוו מספקים כח זוויתי (מומנט) חזק מאד בהשוואה לגודלם
- מנועי סרוו פועלים בחוג סגור ולכן נחשבים אמינים מאד
- למנועי סרוו יש מעגל שליטה ובקרה פנימי
- מנועי סרוו צורכים זרם בצורה פרופורציונאלית למטען אותו הם נושאים (לכן סרוו שאינו נושא מטען רב לא יצרוך הרבה זרם)
- מנועי סרוו פועלים במתח נמוך יחסית (כ-4 עד 6 וולט)

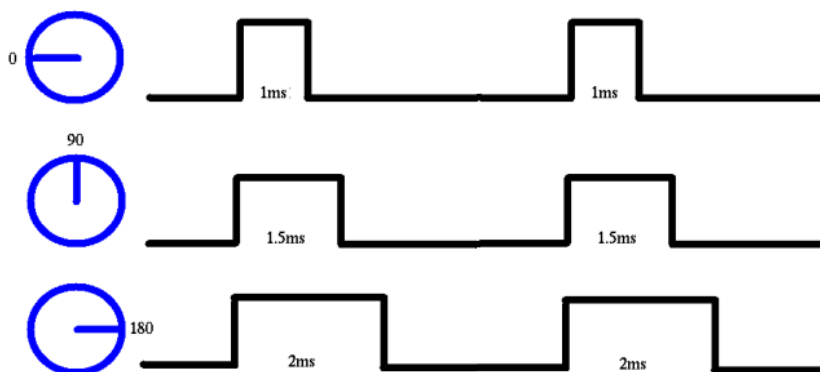
### כיצד שולטים במנוע סרוו?

שליטה במנועי סרוו מבוצעת על ידי שליחת אות דיגיטאלי אל חוט הבקרה של המנוע. הרעיון הכללי הוא שליחת גל מרובע (Square Wave) אל המנוע, כאשר אורך הגל הוא זה שקובע את הזווית אליה ינוע המנוע.

לדוגמה, כאשר נספק למנוע גל בו רוחב הפולס הוא 1 מילי-שנייה, המנוע ינוע אל זוויתו המינימאלית – 0 מעלות.

כאשר נספק למנוע גל בו רוחב הפולס הוא 1.5 מילי-שנייה, המנוע ינוע אל זוויתו האמצעית – 90 מעלות.

כאשר נספק למנוע גל בו רוחב הפולס הוא 2 מילי-שנייה, המנוע ינוע אל זוויתו הגדולה ביותר – 180 מעלות.





## ביבליוגרפיה

- הקוד: [https://github.com/Darkfall48/Cardiac\\_Memory/blob/master/Project%20Code/Working/Complete%20Work/One\\_Shot/One\\_Shot.ino](https://github.com/Darkfall48/Cardiac_Memory/blob/master/Project%20Code/Working/Complete%20Work/One_Shot/One_Shot.ino)
- הקישור GitHub של הפרוייקט: <https://github.com/Darkfall48/Lucky-Star-Shooter>

### קישורים אחרים:

- <https://stwww1.weizmann.ac.il>
- <http://ptc.weizmann.ac.il/?CategoryID=1623&ArticleID=4640>
- <https://stwww1.weizmann.ac.il/lasers/?p=1695>
- <https://hackstore.co.il/books/arduino-for-beginners-1>