



מכון טכנולוגי חולון
Holon Institute of Technology

ספר פרויקט

פרויקט גמר הנדסי לתואר B.Sc.

בהנדסת חשמל ואלקטרוניקה

**יישום תוצרי ראייה ממוחשבת
באמצעות רובוט הנשלט ע"י FPGA**

**Implementation of computer
vision products using a robot
controlled by FPGA**

אדיר ציון

שלומי רחימי

38084455

301287223

מנחה: ד"ר טירר נוניל

01/02/16

גולומב 52, ת.ד. 305, חולון 58102
טלפון 03-5026706, פקס' 03-5026643
52 Golomb St., Holon 58102 Israel
www.hit.ac.il Tel. 972-3-502-6706, Fax. 972-3-502-6643

הפקולטה להנדסה
המחלקה להנדסת חשמל ואלקטרוניקה
Faculty of Engineering
Department of Electrical and Electronics Engineering

תוכן עניינים

פרק 1 – הקדמה	6
1.1 רשימת קיצורים	6
1.2 תקציר	8
1.3 Abstract	9
פרק 2 – רקע תיאורטי	10
2.1 התמרת האף מעגלית (CHT)	10
2.2 FPGA	12
2.3 שפת VHDL	13
2.4 תוכנת ה-Quartus	13
2.5 GUI	14
2.6 עיבוד תמונה דיגיטלית	15
2.6.1 כיצד עובדת מצלמה דיגיטלית	15
2.6.2 עקרון עיבוד תמונה דיגיטלית	16
2.7 RGB	18
2.8 EEPROM	19
2.9 UART	20
2.9.1 מבוא	20
2.9.2 פונקציונאליות	20
2.9.3 UART ב-PC	22
2.10 Bluetooth	23
2.10.1 SparkFun Bluetooth Modem – BlueSMiRFSilver	24
2.11 מנוע DC-Servo	25
2.11.1 אמצעי שליטה בכיוון ומהירות מנועי DC (גשר H)	25
2.11.2 סכמה חשמלית של H-Bridge	26
2.11.3 סיכום אופן פעולת קווי בקרת המנוע לקביעת כיוון הסיבוב	26
2.12 מנוע SM-S4303R	27
2.13 MC74HC541A – octal buffer	28
2.14 אפנון רוחב פעימה – PWM	29

30	פרק 3 – מבנה המערכת	
31	GUI	3.1
31	עקרון בניית ממשק המשתמש	3.1.1
31	אופן פעולת ממשק המשתמש	3.1.2
32	מבנה ממשק המשתמש	3.1.3
33	שידור הוידאו מהסמארטפון ל- Matlab	3.1.4
35	פתיחת הוידאו בממשק המשתמש	3.1.5
36	יצירת תקשורת טורית בעזרת ממשק המשתמש	3.1.6
37	שליחת אותות ותפעול פקדי התנועה	3.1.7
38	זיהוי אובייקטים ע"פ צורה וצבע	3.2
38	רציונל	3.2.1
39	זיהוי מעגל ע"י התמרת CHT	3.2.2
39	זיהוי צבע המעגל	3.2.3
41	שילוב ניתוח הצורה ובדיקת הצבע לכדי זיהוי כולל	3.2.4
42	UART	3.3
42	חיבור המודם למחשב	3.3.1
45	בניית ה- UART וחיבורו	3.3.2
47	חיבור מוצאי המערכת ללוח ה- FPGA	3.4
47	חיבור מנועי סרוו	3.4.1
51	חיבור נוריות ה- Led	3.4.2
52	חיבור ה- Buffer	3.4.3
53	פרק 4 – סיכום ומסקנות	
53	סיכום	4.1
53	קשיים איתם התמודדנו במהלך הפרויקט	4.1.1
54	הצעות להמשך הפרויקט	4.1.2
55	ביבליוגרפיה	4.2
56	טופס הצעת הפרויקט	

רשימת איורים וטבלאות

10	משוואת מעגל	איור 1-
10	התמרת Hough של מעגל בעל רדיוס R	איור 2-
11	התמרת Hough של מעגל בעל רדיוס שאינו ידוע	איור 3-
11	מישור Hough	איור 4-
12	בלוק לוגי	איור 5-
12	רכיב FPGA של חברת Altera	איור 6-
17	ייצוג סוגי תמונות ב-Matlab	איור 7-
19	ייצוג הקסדצימלי ל-16 צבעים בתקן HTML 4.01	איור 8-
19	זיכרון מסוג EEPROM	איור 9-
20	חבילת מידע הנשלחת/נקלטת ע"י UART	איור 10-
21	סנכרון קצב שידור המידע המגיע מהמשדר למקלט	איור 11-
22	פירוט הפינים ביציאה הטורית של ה-PC	טבלה 12-
24	RN42/RN42N to MCU interface	איור 13-
24	BluteSMiRFSilver Bluetooth modem	איור 14-
24	תרשים רכיב ה-RN41 Bluetooth	איור 15-
25	מעגל בקרת מנועי DC בארכיטקטורת H-Brige	איור 16-
26	אופן פעולת קווי בקרת המנוע לקביעת כיוון הסיבוב	איור 17-
27	מפרט טכני של מנוע Servo מסוג SM-S4303R	איור 18-
28	Buffer function table	איור 19-
28	מבט כולל על כלל החוצצים ברכיב	איור 20-
28	Buffer pin assignment	איור 21-
28	המבנה הפנימי של חוצץ בודד	איור 22-
29	נוסחה לחישוב Duty Cycle	איור 23-
29	Pulse Width Modulation	איור 24-
30	תרשים זרימה של כלל המערכת	איור 25-
32	מבנה ממשק המשתמש	איור 26-
33	צד לקוח של האפליקציה DroidCam	איור 27-
33	האפליקציה DroidCam המותקנת במכשיר ה-Android	איור 28-
34	Matlab Support Package Installer	איור 29-
34	Matlab Image Acquisition Tool	איור 30-
35	הצגת הוידאו בממשק המשתמש	איור 31-

איור 32-	מקלט ה-Bluetooth במצב מחובר	36
איור 33-	האובייקטים שנבחרו לצורך הזיהוי	38
טבלה 34-	טווחי רמות הבהירות לזיהוי הגוונים השונים	40
איור 35-	אחוז שטח המעגל מהתמונה ה- "חתוכה"	41
איור 36-	תרשים זרימה של תהליך זיהוי האובייקט	41
איור 37-	דונגל Bluetooth דגם CSR 4.0	42
איור 38-	ביצוע Pairing בין המחשב ומודם ה-Bluetooth	42
איור 39-	ה- COM שנוצר במנהל ההתקנים	43
איור 40-	הגדרות יציאת ה- COM	43
איור 41-	אפשרויות הצגת התוכן המוקלד בתוכנת ה- Tera Term	44
איור 42-	הצגת הגדרות מודם ה-Bluetooth	44
איור 43-	שינוי ה- Baud rate בתוכנת ה- Tera Term	45
איור 44-	רכיב ה- UART	46
איור 45-	הצגת קוד ה- Ascii של תו בנוריות ה- DE-nano	46
איור 46-	כיווני תנועת מנוע ה- Servo	47
איור 47-	מנוע Servo מדגם SM-S4303R	47
איור 48-	רכיב ה- PWM בערוצים השונים	48
איור 49-	ערכי מחלק התדר	49
איור 50-	אותות PWM הנשלחים למנועי ה- Servo	49
איור 51-	סכמת RTL של רכיב ה- PWM	50
איור 52-	רכיב הדלקת וכיבוי נוריות ה- Led	51
איור 53-	ווקטור הדלקת וכיבוי נוריות ה- Led	51
איור 54-	סכמת הרכיב ששולט בהפעלת ה- Led	51
איור 55-	המבנה הפנימי של חוצץ בודד	52

פרק 1 – הקדמה

1.1 רשימת קיצורים

USB – Universal Serial Bus

RPM – Round Per Minute

VGA – Video Graphics Array

RoHS – Restriction of Hazardous Substances Directive

GUI – Graphical User Interface

RGB – Red Green Blue

EEPROM – Electrically Erasable Programmable Read Only Memory

UART – Universal Asynchronous Receiver Transmitter

PC – Personal Computer

DC – Direct Current

PWM – Pulse Width Modulation

FPGA – Field Programmable Gate Array

LED – Light Emitting Diode

CHT – Circular Hough Transform

SOC – System On Chip

VLSI – Very Large Scale Integration

ASIC – Application Specific Integrated Circuit

CCD – Charge Coupled Device

VGA – Video Graphics Array

JPEG – Joint Photographic Experts Group

LCD – Liquid Crystal Display

LSB – Least Significant Bit

MSB – Most Significant Bit

FHSS – Frequency Hopping Spread Spectrum

PAN – Personal Area Network

ISM – Industrial Scientific and Medical

AFH – Adaptive Frequency Hopping Spread Spectrum

SPP – Serial Port Profile

MOSFET – Metal Oxide Semiconductor Field Effect Transistors

WiFi – Wireless Fidelity

IP – Internet Protocol

ISI – Intersymbol Interference

COM – Component Object Model

Ascii – American Standard Code For Information Interchange

VHSIC – Very High Speed Integrated Circuits

VHDL – VHSIC Hardware Description Language

RTL – Register Transfer Logic

1.2 תקציר

תחום הראייה הממוחשבת תופס מקום הולך וגדל במוצרים שנמצאים בחזית הטכנולוגית בימנו. בין אם בתעשייה- באוטומציה של תהליכים אשר היו נחלתם של מפעילים אנושיים בעבר, כגון בדיקות ואבטחת איכות או בתקשורת חזותית, באינטרנט ובסרטי וידאו דיגיטליים. נושאי אבטחה משודרגים גם הם על ידי שימוש בטכניקות של עיבוד תמונה וראייה ממוחשבת לצרכים של זיהוי ועקיבה אחר אנשים, פנים ומטרות אחרות. ביצוע של משימות מהסוג הנ"ל באמצעות ראייה ממוחשבת משפר באופן משמעותי את איכות תוצאותיהן ומקטין מאוד את עלות ביצוען.

מטרת הפרויקט היא ביצוע עיבוד תוצרי ראייה ממוחשבת ויישומם ע"י רובוט הנשלט באמצעות FPGA. הדבר נעשה ע"י שרשרת תהליכים שראשיתה בסרט וידאו המצולם בזמן אמת ממצלמת Smartphone המותקן על גבי הרובוט ושליחתו למחשב למטרת עיבוד. העיבוד נעשה באמצעות תוכנת Matlab כאשר עיקרו, זיהוי וסיווג אובייקטים גיאומטריים (אשר מפוזרים בחלל החדר בו נמצא הרובוט), ע"פ צורה וצבע. תוצרי העיבוד מתורגמים למתן פקודות אשר מועברות לרובוט תוך שימוש בתקשורת טורית ובאופן אלחוטי ע"י Bluetooth. המידע מגיע למקלט Bluetooth הנמצא על גבי הרובוט ומוזרם ישירות ל- FPGA באמצעות UART אשר מהווה כמקלט בלבד ומסונכרן עם קצב שידור המידע המגיע מהמסדר (המחשב). ה- FPGA המשמש כבקר מחובר דרך יציאות I/O לגלגלי הרובוט המונעים באמצעות מנועי Servo ולנוריות LED בצבעים ירוק, אדום וצהוב, ומפעיל אותם בהתאם למידע המעובד. כל התהליך מנוהל ונשלט ע"י אפליקציית GUI אשר נכתבה בסביבת ה- Matlab ובעזרתה יכול המשתמש לראות את תנועת הרובוט מהמצלמה המותקנת עליו בזמן אמת, להתממשק עם הרובוט דרך יציאה טורית הנבחרת ע"י המשתמש, לשלוט בתנועתו ע"י פקדים ייעודיים ולאבחן זיהוי אובייקטים במרחב.

1.3 Abstract

These days, the field of computer vision is playing an increasingly growing part in the technological front. In the industry we are witnessing automation of processes which were shared by human operators in the past, such as tests, quality assurance of visual media, Internet and digital videos. Security issues too, are being upgraded with techniques of image processing and computer vision for the identification and tracking of people, faces and other purposes. These techniques significantly improve the quality of the results and greatly reduce costs.

This project's goal is to process computer vision products and their application, by using an FPGA-controlled robot. This is done by a chain of processes that begin with real time video footage taken with a Smartphone mounted on the robot and sent to a computer for processing. The processing is done using Matlab software that identifies and classifies geometric objects (scattered in the room where the robot is located), according to shape and color. The products of this processing are then translated into commands which are transferred to the robot (with wireless Bluetooth) using serial control media. The information is then broadcast to the Bluetooth receiver on the robot and sent directly to the FPGA via UART that acts as a receiver only and is synchronized with the transmission rate of information received from the transmitter (The computer). The FPGA is used as a controller that is connected via I/O ports to the robot's wheels which run with Servo motors and to LED lights (green, red and yellow), operating them according to the information processed.

The entire process is managed and controlled by a GUI application that was written in the Matlab environment and with which the user can see the robot's movement from the camera installed on it, in real-time. The user shares the interface with the robot through a serial port selected by the user, hereby controlling the motion with dedicated controls and diagnosing identification of the objects in the space.

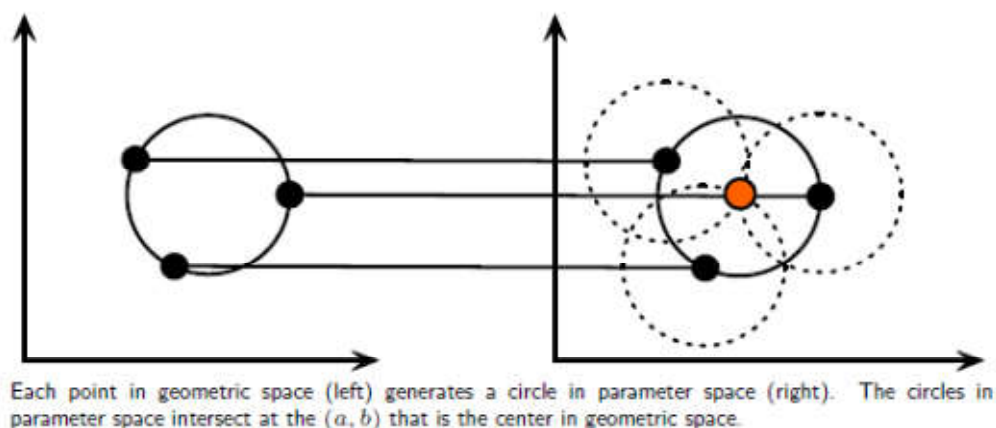
פרק 2 – רקע תיאורטי

2.1 התמרת האף מעגלית (CHT)

התמרת האף (Hough transform) הינה שיטה למציאת אלמנטים המשמשת בניתוח תמונה, ראייה ממוחשבת ועיבוד תמונה. מטרת האלגוריתם הינה מציאת מקרים לא מושלמים של אובייקטים בתוך סוג מסוים של צורות על ידי תהליך "הצבעה". הליך הצבעה זה מתבצע במרחב פרמטרים, שממנו אובייקטים "מועמדים" מתקבלים כמקסימום מקומי במרחב סוכם (Accumulator) הנבנה על ידי האלגוריתם. אלגוריתם האף בהגדרתו הראשונית עסק בזיהוי של קווים בתמונה, אך מאוחר יותר הורחב גם לזיהוי מיקומים של צורות כלשהן. הצורות הנפוצות ביותר בהקשר להתמרת האף הן מעגלים או אליפסות. מעגל הינה צורה גיאומטרית אשר מוגדת ע"י התבנית המתמטית הבאה:

$$(x - a)^2 + (y - b)^2 = R^2 \quad \text{איור 1 – משוואת מעגל}$$

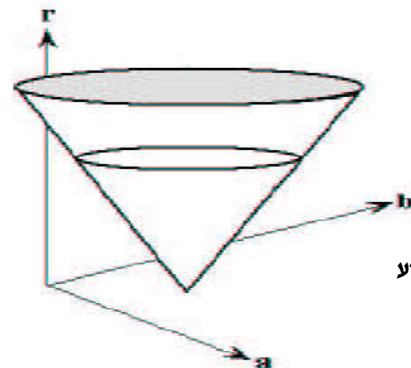
a, b הם פרמטרים המתארים את שיעורי מרכז המעגל ו- R הינו רדיוס המעגל. מטרת האלגוריתם הוא ביצוע זיהוי קצוות (edge detection) כגורם סינון ראשוני אשר ייצור תמונה ובה קווי המתאר של האובייקטים הנבדקים. כאשר ידוע מהו רדיוס המעגל (R) אותו מחפשים, אופן החיפוש יתבצע בצורה דו ממדית ע"י סריקת כל הנק' הנמצאות על ההיקף של האובייקט הנבדק כאשר כל זוג שיעורי נק' מסוימת (x, y) במישור התמונה, מהווה נק' מרכז מעגל (a, b) בעל רדיוס R במישור האף:



איור 2 – התמרת Hough של מעגל בעל רדיוס R

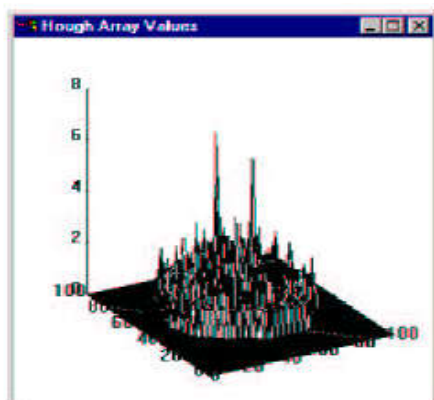
באמצעות שימוש במערך דו מימדי המכונה סוכם (Accumulator) אשר מכיל את הערכים של הפרמטרים a , b ניתן לבצע "הצבעה" לערכים לפי כמות הפעמים בה הם מופיעים. צמד הערכים שיקבל את ההצבעה המקסימאלית ואשר יעבור ערך סף (Threshold) שיקבע מראש, יהווה נק' מרכז המעגל בעל רדיוס R נתון. (הנק' הכתומה בשרטוט).

כאשר רדיוס המעגל (R) אותו מחפשים אינו ידוע, כל נק' במישור התמונה מתורגמת למשטח בצורת חרוט במישור האף. כאמור כל חרוט שכזה בנוי מאוסף של מעגלים בעלי מרכז משותף וברדיוסים שונים:



איור 3 – התמרת Hough של מעגל בעל רדיוס שאינו ידוע

מספר הממדים של הסוכם צריך להיות שווה למספר הפרמטרים הלא ידועים (a, b, R) , כלומר, שלוש. כל נק' במישור התמונה מתורגמת לתא בסוכם בעל הפרמטרים המתאימים וערכו מוגדל באחד. על ידי מציאת התאים עם הערכים הגבוהים ביותר בסוכם, בדרך כלל באמצעות מציאת מקסימום מקומי בו, ניתן לגלות את המעגלים בעלי ההסתברות הגבוהה ביותר. כאמור, הדרך הפשוטה ביותר למציאת פסגות אלו הינה באמצעות הפעלת סף (Threshold). עם זאת, קיימות טכניקות אחרות אשר עשויות להניב תוצאות טובות יותר בנסיבות שונות. בשל חוסר דיוק של אלגוריתם זיהוי הקצוות, יש בדרך כלל שגיאות בסוכם, אשר עשויות להקשות על מציאת הפסגות המתאימות, ובכך גם את המעגלים המתאימים. התוצאה הסופית של האלגוריתם הינה מערך תלת ממדי, בעל ציר אחד המייצג את R , ציר אחד המייצג את a וציר נוסף המייצג את b . לכל תא במערך ערך מסוים והתאים בעלי הערכים הגבוהים ביותר מייצגים את המעגלים האינדוקטיביים ביותר בתמונה. במישור האף, הנק' בעלת הערכים המייצגים את המעגל המבוקש, תבוטא כנק' בה נחתכים מס' מקסימאלי של קונוסים.



איור 4 – מישור Hough

FPGA 2.2

FPGA (Field Programmable Gate Array) הינו רכיב אלקטרוני מתוכנת הבנוי כמעגל משולב (Integrated Circuit) המכיל בתוכו מספר רב מאוד של "בלוקים לוגיים" הניתנים לתכנות. בלוקים לוגיים הם אוסף של שערים לוגיים (כמו שער NOT ו-AND), רכיבים מורכבים יותר כגון בוררים ומפענחים ואף רכיבי זיכרון מ-Flip-flops בודדים ועד רגיסטרים ובלוקים שלמים של תאי זיכרון. מעל שכבה זו נמצאת שכבה שבעת צריבת התוכנית שכתב המשתמש לרכיב FPGA, יוצרת "קשרים" חשמליים בין הרכיבים האלה ובין הבלוקים הלוגיים עצמם בהתאם להגדרות שקבע המתכנת ובכך נוצר מעגל חשמלי המתאים לקוד. ה-FPGA התחיל את דרכו כמקביל ומתחרה ל-CPLD, אך ככל שהגודל, היכולות, ומהירות הפעולה של ה-FPGA גדלו, גדל גם נתח השוק של רכיבים מסוג זה, וכיום מקובל למצוא רכיבי FPGA כמערכת על שבב (SoC).

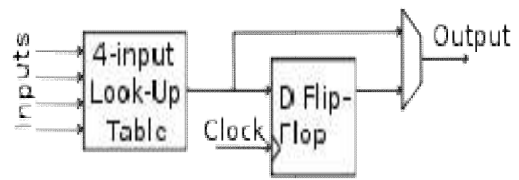
נהוג להשתמש ברכיבי FPGA עבור מעגלים שנועדו לבצע משימות בעלות אותו אופי, אך שונות אחת מהשנייה בלוגיקה הפנימית (ולא בממשקים החיצוניים), ואשר מיוצרים בכמויות קטנות. במעגלים אלו, העלות של התכנון והייצור של כל דגם היא גבוהה יחסית, ולכן יש היגיון לשאוף לפיתוח של מעגל חשמלי בעל ממשקים משותפים שיתאימו למספר מוצרים שונים, ובכך להגדיל את כמות המעגלים שמיוצרים.

בזכות המבנה הפנימי של רכיבי ה-FPGA, הם שימושיים במיוחד בתחומים הדורשים עיבוד מקבילי, וניתן לבצע בהם כמות גדולה יחסית של פעולות וחישובים, אפילו כאשר קצב אות השעון נמוך יחסית. יכולת זו שימושית במיוחד בתחום פיצוח הצפנות, ואף פתחה אפשרות לסוג חדש של עיבוד, שבו רכיבי ה-FPGA מסייעים למעבד הראשי בביצוע פעולות מסוימות, לפעמים על פי הגדרת תצורה ספציפית, שנטען לתוך ה-FPGA מיד לפני ביצוע החישוב ב-FPGA.

המגבלה העיקרית בשילוב רכיבי FPGA במחשבים היא יכולת הביצועים מבחינת מהירות והספק חשמלי בהשוואה למעגלי VLSI בתכנון מלא וב-ASIC ומגבלת הסיבוכיות האפשרית הנמוכה ב-FPGA, בהשוואה לאפשרויות בתוכנה רגילה. אחת הדרכים להתמודד עם מגבלה זו היא להפחית את הדיוק היכן שניתן, על מנת לנצל את המקביליות של ה-FPGA. מגבלה נוספת היא זמן ההכנה הארוך יחסית (בהשוואה לתוכנה רגילה) מרגע שהוגדר שינוי מסוים בתצורת הטעינה של ה-FPGA, עד לרגע שניתן להשתמש בתצורה הזו.



איור 6 – רכיב FPGA של חברת Altera



איור 5 – בלוק לוגי

2.3 שפת VHDL

שפת VHDL (Very High Speed Integrated Circuits Hardware Description Language) נוצרה כתוצאה מפרויקט מחקר גדול שערכו בשנות ה-80 משרד ההגנה האמריקאי. הפרויקט נקרא VHSIC (Very High Speed Integrated Circuit).

המטרה הייתה ליצור שפת תיעוד וסימולציה סטנדרטית לתיאור חומרה ספרתית מורכבת. במשך השנים הצטרפו חברות מהתעשייה לפיתוח השפה, כגון: IBM, TI וכו', ולאחר שיפורים שנערכו הוגדר בשנת 1987 תקן IEEE 1076 שמגדיר באופן בסיסי את השפה. עם השנים התווספו תקנים לסוגי מידע ספציפיים, כגון: לסוגי מידע סטנדרטים-1164, לחבילות מתמטיות-1076.2 וכו'.

כיום השפה מכילה בנוסף לכלי הסימולציה, כלי סינתזה והינה אחת משפות תיאור החומרה הנפוצות בתעשייה ובאקדמיה. התכנון בעזרת השפה נעשה בהתאם לתכנון הספרתי המקובל שהינו בניית ארכיטקטורה של המערכת, הגדרת יחסי גומלין בין החלקים המבצעים לבין החלקים המבקרים בתוך המערכת, כתיבת קוד, סימולציה וסינתזה.

2.4 תוכנת ה- Quartus

תוכנת ה- Quartus פותחה ע"י חברת ALTERA לשימושים עבור רכיבים ברי-תכנות כדוגמת FPGA. בעזרת התוכנה ניתן לערוך סימולציות וסינתזות לרכיבים ובנוסף ניתן לוודא את מימוש החומרה הנוצר ע"י סכמת RTL ואף לצרוב את הקוד ברכיבים. כיום חברת ALTERA מספקת גרסת תוכנה ללא תשלום אשר נקראת "Quartus II Web Edition".

GUI 2.5

GUI (Graphical User Interface) הוא ממשק משתמש לתוכנה או לאתר אינטרנט, המבוסס על עיצוב גרפי (להבדיל מעיצוב טקסטואלי בלבד) של המסך המוצג למשתמש. הוא מכיל פקדים המשמשים כרכיבים (components) אשר מאפשרים למשתמש לבצע משימות אינטראקטיביות. המשתמש אינו צריך ליצור תסריט (script) או להקליד פקודות בשורת הפקודה כדי לבצע את המשימות. שלא כמו קידוד תוכניות למטרת ביצוע משימות, המשתמש אינו צריך להבין את הפרטים של איך המשימות מבוצעות.

רכיבי ממשק המשתמש יכולים לכלול תפריטים, סרגלי כלים, לחצנים, כפתורי רדיו, תיבות רשימה, מחוונים ועוד. ממשק הנוצר באמצעות כלים ייעודיים כדוגמת MATLAB, יכול גם לבצע כל סוג של חישוב, לקרוא ולכתוב קבצי נתונים, לתקשר עם ממשקי משתמש אחרים, ולהציג נתונים באמצעות טבלאות או גרפים.

לרוב, מממשק משתמש מחכה לקריאות מהמשתמש לביצוע פעולות, ואז מגיב לכל קריאה לפי התור. לכל פקד ופקד יש פונקציית Callback אשר מבצעת סט של פקודות שנכתבו מראש ומבוצעות בעת קבלת קריאה מהמשתמש כדוגמת לחיצה על כפתור רדיו או בחירת ערך מסוים ברכיב מסוג תפריט. ישנן 2 דרכים ליצירת ממשק משתמש בסביבת תוכנת MATLAB:

- באמצעות GUIDE:

בגישה זו יוצרים תרשים ריק אשר מאוכלס ברכיבים שונים הניתנים לבחירה מתוך עורך גרפי. ניתן למקם כל רכיב ורכיב בתרשים הריק ולשנות את מאפייניו הויזואליים כגון צבע, גופן, גודל וכו', מה שמאפשר שליטה דינמית ופשוטה בנראות האפליקציה אותה רוצים לבנות. GUIDE יוצר קובץ קוד ובו פונקציית Callback לכל רכיב בממשק המשתמש. ניתן להפעיל את האפליקציה הן מהתרשים הנבנה (קובץ FIG) והן מקובץ הקוד שנוצר.

- יצירת ממשק משתמש "תכנותי":

בגישה זו יוצרים קובץ קוד בלבד, אשר מגדיר את כל התכונות וההתנהגויות של כל רכיב ורכיב. כאשר משתמש מפעיל את קובץ הקוד, נוצר תרשים המכיל את כל הרכיבים אשר נשלטים ע"י פעולות המשתמש. לרוב, התרשים אינו נשמר מאיטרציה לאיטרציה מכיוון שקובץ הקוד יוצר תרשים חדש בכל פעולת הרצה.

קבצי הקוד של כל גישה נראים שונה. קבצי קוד של ממשקי משתמש "תכנותיים" הם ארוכים יותר משום שהם מגדירים את כל המאפיינים של כל רכיב ורכיב בתרשים, כמו גם את פונקציות ה-Callbacks לרכיבים. שימוש ב-GUIDE לעומת זאת, מכיל קובץ קוד יותר מצומצם, משום שכל הגדרות הרכיבים נעשים דרך התרשים עצמו בעת בנייתו. עיקר הקוד עוסק בפונקציות ה-Callback של הרכיבים ואתחול ממשק המשתמש בעת פתיחתו. ניתן ליצור ממשק משתמש באמצעות GUIDE ואז לשנות אותו תכנותית. לעומת זאת, שימוש הפוך בגישות, בלתי ניתן לביצוע.

2.6 עיבוד תמונה דיגיטלית

עיבוד תמונה (Image Processing) הוא תחום טכנולוגי-הנדסי, העוסק בעיבוד של אותות המייצגים תמונה או הדמאה (image) – בין אם היא תמונה בודדת (stills) או תמונה (frame) מסרטון וידאו. תחום עיבוד התמונה כולל אלגוריתמים לשיפור תמונות, חילוץ מידע מתוך תמונות, התאמת תמונה, גילוי שינויים, זיהוי תנועה בוידאו, ראייה ממוחשבת ועוד.

עיבוד תמונה נעשה לשתי מטרות עיקריות:

- לשם שיפור התמונה: הפחתת "רעשים", הדגשת פרטים, חידוד, ריטוש תמונה, יצירת אפקטים, תיקון צבעים ורמות אפור וכדומה.
- לשם הסקת מסקנות: מדידות, זיהוי פגמים, גילוי תנועה, זיהוי צורות וסיווג אובייקטים.

לעיבוד תמונה שימושים רבים:

- בתעשייה עיבוד התמונה משמש למדידות ולזיהוי פגמים. ישנן גם מכונות ש"רואות" את העובד המשתמש בהן, מכוונות את כלי העבודה ואחר בודקות את התוצאה.
- בביטחון לאומי, בביטחון פנים ובאבטחה עיבוד התמונה משמש לזיהוי תנועה וסיווג צורות.
- בציבור הרחב משתמשים בעיבוד תמונה על מנת לשפר את התמונות שצולמו.

2.6.1 כיצד עובדת מצלמה דיגיטלית

מקור עיקרי לצילום תמונה דיגיטלית הוא מצלמה דיגיטלית, כגון מצלמות הוידאו הביתיות האופייניות, ומצלמות ה-Still החדישות בהן במקום סרט צילום ישנה מדיה דיגיטלית. מצלמות כאלו מורכבות סביב מטריצת חיישנים הרגישים לאור (כגון גלאי CCD), אשר קובעים מתחזרם חשמלי באופן יחסי ישיר לכמות הפוטונים הפוגעת בהם. כל חיישן מייצר ערך עבור פיקסל בודד, ולכן מטריצות חיישנים אלו צריכות להיות גדולות (מאות על מאות של חיישנים על chip בודד). המתחזרם הנוצרים מומרים ישירות ל-L הסיביות המייצגות את רמת האפור בנקודה זו וכך מתקבל ערכו של הפיקסל. פרמטר מרכזי בקביעת איכותה של מצלמה כזו הוא מספר הגלאים במטריצת החיישנים. מצלמות Still דיגיטליות עם מערך של 480 על 640 נחשבות לרמה הבסיסית ביותר (זוהי רזולוציית VGA), ואיכותן רחוקה מלהתחרות במצלמת סרט קונבנציונלית. לשם קבלת איכות מתחרה בסרט צילום, על המערך להכיל כ-2000 על 2000 חיישנים ויותר. כאלטרנטיבה למערך דו-מימדי של חיישנים, ניתן להציע מערך חד-מימדי וסריקה מכאנית (של המערך, מראה מטילה או פריזמה) לקבלת אוסף שורות היוצרות את התמונה. גישה זו מקובלת כאשר נדרשת רזולוציה

גבוהה מאוד ולפיכך מערך חיישנים ארוך (בסדר גודל של אלפים). הבעיה העיקרית עם גישה זו הוא הצורך במרכיבי מכאניקה מדויקת במצלמה.

על מנת לקלוט תמונת צבע נדרש בעיקרון שימוש בשלושה מערכי חיישנים, כשכל אחד מהם מחובר למסנן אופטי לצבע הרלוונטי. כך ייצא למשל כי שכבת ה-RED תיוצר ע"י מערך גלאים עם פילטר אופטי המונע מכל צבע למעט אדום לחדור. לכן, מצלמת צבע כזו תהיה בפועל שלוש מצלמות שונות על אותו ציר אופטי לשם קבלת התאמה מוחלטת ביניהן במובן הגיאומטרי.

כל האמור לעיל מתייחס לדרך בה פועלת מצלמה דיגיטלית אופיינית הן לתמונות Still והן לוידאו (סדרות של תמונות ברצף). חשוב לציין כי מצלמה אלקטרונית אינה מוגבלת לחישה של אור נראה. בהינתן גלאים החשים כל תופעה פיזיקאלית אחרת וממירים אותה לאותות חשמליים, ניתן לסדרם במערך דו-מימדי ולהגדיר את תוצאת החישה כתמונה דיגיטלית לכל דבר. כך הדבר בצילום אינפרא-אדום, המספק תמונות של אורך גל אחר הקשור לחום של עצמים, כך גם בצילום רנטגן, צילומים סיסמיים של רעידות על-פני כדור הארץ, ועוד.

2.6.2 עקרון עיבוד תמונה דיגיטלית


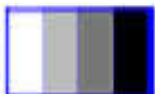


כל קובץ תמונה מורכב מנקודות זעירות הקרויות פיקסלים. ככל שיש יותר נקודות ביחידת שטח, כך הרזולוציה של התמונה גבוהה יותר, והתמונה תראה חדה יותר. ככל שנקודה יכולה לתת טווח יותר גדול של צבעים, כך עומק הצבע גדול יותר, והתמונה תראה ריאליסטית יותר. הנתונים הללו מאוחסנים כמו שהם בקובץ Bitmap (תמונת מפת סיביות), שבו יש ייצוג מספרי לכל נקודה בתמונה, או שנדחסים באמצעות אלגוריתם מיוחד ונשמרים בקבצים דחוסים כדוגמת JPEG. בכל תהליך של שמירה ופתיחה בקבצים דחוסים, התוכנה משתמשת בתהליך עיבוד מיוחד, שיועד לדחוס את נתוני התמונה תוך איבוד מינימאלי של מידע.

מערכת עבוד תמונה בסיסית מורכבת מדוגם תמונה המחובר למחשב המסוגל לאחסן את התמונה בזיכרון ואחר כך לעבד אותו באמצעות תוכנה מתאימה המסוגלת לנהל את הדגימה, את התצוגה של התמונה ולבצע את האלגוריתמים השונים. התמונה הנדגמת תופיע בזיכרון בצורת מטריצה דו ממדית $p[x][y]$, כל איבר במטריצה הוא פיקסל. כל פיקסל מכיל אינפורמציה על רמת האור שהפיקסל הזה נחשף לה בזמן הצילום. הרזולוציה נקבעת על ידי צפיפות הפיקסלים על חיישן המצלמה, ואם דוגמים ומייצגים פיקסל בבייט (8 ביט) זה אומר שכל פיקסל מייצג אחד מ-256 רמות אפור.

במקרה של טיפול בתמונה צבעונית משתמשים לרוב ב- RGB (Red-Green-Blue) ואז ישנן שלוש מטריצות:

$$p_{\text{red}}[x][y], p_{\text{green}}[x][y], p_{\text{blue}}[x][y]$$

Overview: How Images are Represented?

Image Type	Indexed 	Intensity 	Binary 	RGB 
Double Data	Image is an M-by-N array of integers in the range [1,P]. Colormap is a P-by-3 array of floating-point values in the range [0,1].	Image is an M-by-N array of floating-point values. The conventional dynamic range is [0,1].	Image is an M-by-N logical array containing only 0's and 1's.	Image is an M-by-N-by-3 array of floating-point values in the range [0,1].
Uint8 Data	Image is an M-by-N array of integers in the range [0,P-1]. Colormap is a P-by-3 array of floating-point values in the range [0,1].	Image is an M-by-N array of unsigned 8-bit integers. The conventional dynamic range is [0,255].	Image is an M-by-N logical array containing only 0's and 1's. Unlike uint8 intensity images, 1 represents white	Image is an M-by-N-by-3 array of floating-point values in the range [0,255].

RGB 2.7

אדום ירוק כחול או בשם הלועזי GREEN RED ו-BLUE (ובקיצור אותם RGB מפורסמים) הם צבעי יסוד היוצרים את התמונות שעל מסכי המחשב והטלוויזיה. בין אם מדובר במסכים מיושנים וכבדים או במסכים LCD ופלזמה שטוחים, ואפילו במקרני וידאו. חיבור עצמות מתאימות של קרני אור בצבעים אלו מאפשר לנו לקבל כמעט כל גוון צבע שנרצה.

קביעת העוצמות של מרכיבי הצבע השונים נעשית על ידי השמת המספרים השלמים בין 0 ל 255 לכל אחד מן המשתנים R , G , B , כאשר 0 פירושו לא להאיר בכלל את הפיקסל, ו-255 מסמל הארה במלוא העוצמה. (לעיתים מסמנים עוצמה יחסית על ידי חלוקה ב-255, ואז הערכים הם בין 0 ל-1). לשם הנוחות נהוג לתרגם את מספרים אילו לבסיס הקסדצימלי (בסיס 16) בו יש צורך רק ב-2 תווים כדי לייצג מספרים בטווח הנ"ל. באופן זה מקצים לכל אחד משלושת צבעי היסוד (RGB) שני תווים משמאל לימין. בכל צבע הטווח נע בין 00-FF (בבסיס דצימלי) שמבטאים רוויה של אותו הצבע. מכיוון שבכל מרכיב ניתן ליצור 256 גוונים שונים, מספר הצבעים השונים שניתן לייצג באופן זה הוא 256 בחזקת שלוש שהם 16,777,216 צבעים שונים.

דרך מקובלת לקביעת עוצמת מרכיביו של צבע מסוים, היא באמצעות שלשה הקסדצימלית. כל שלשה הקסדצימלית בנויה משש ספרות המייצגות מספר הקסדצימלי בן שלושה בתים, המייצגים את הצבעים אדום, ירוק וכחול אשר מרכיבים את הצבע הסופי:

- בית מס' 1- צבע אדום.
- בית מס' 2- צבע ירוק.
- בית מס' 3- צבע כחול.

כל אחד משלושת הבתים הללו מייצג מספר בטווח הנע בין 00-FF (ייצוג הקסדצימלי), המקביל לסולם הנע בין 0-255 בייצוג עשרוני ומגדיר רמת עוצמה לכל צבע. בסה"כ מתקבל עומק צבע של 24 סיביות (Truecolor) ומס' הצבעים המיוצגים בשיטה זו עומד על $256^3 = 2^{24} = 16,777,216$. לדוגמה, צבע המורכב מהערכים העשרוניים 36, 104, 160 (אדום, ירוק וכחול בהתאמה), מקבל את הערכים ההקסדצימליים 24, 68 ו-A0 בהתאמה, ומיוצג על ידי השלשה 2468A0.

את הכתיבה המקוצרת לעתים מכנים RRGGBB שכן שתי הספרות השמאליות מייצגות את כמות האדום, שתי האמצעיות את כמות הירוק והימניות את הכחול. תקן HTML 4.01 מגדיר שישה עשר שמות צבעים על פי המפורט בטבלה בעמוד הבא :

Color	Hexadecimal	Color	Hexadecimal	Color	Hexadecimal	Color	Hexadecimal
aqua	#00FFFF	Gray	#808080	navy	#000080	silver	#C0C0C0
black	#000000	green	#008000	olive	#808000	teal	#008080
blue	#0000FF	lime	#00FF00	purple	#800080	white	#FFFFFF
fuchsia	#FF00FF	maroon	#800000	red	#FF0000	yellow	#FFFF00

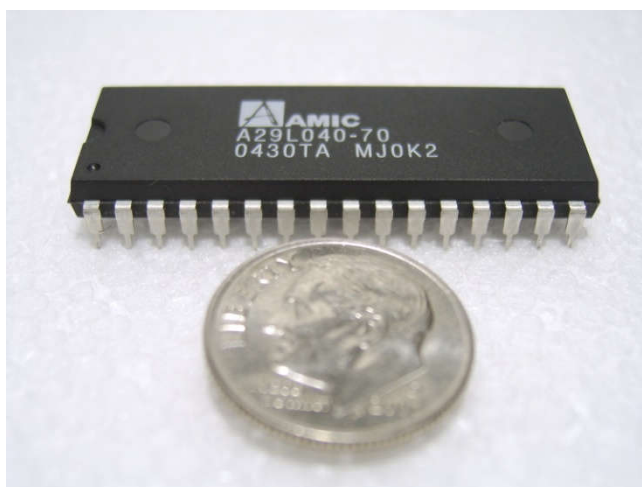
איור 8 – ייצוג הקסדצימלי ל- 16 צבעים בתקן HTML 4.01

EEPROM 2.8

EEPROM (Electrically Erasable Programmable Read-Only Memory) הינו זיכרון לקריאה בלבד אשר ניתן למחיקה ולתכנות. סוג הזיכרון הוא "בלתי נדיף" ומשמש במחשבים ומכשירים אלקטרוניים אחרים בכדי לאחסן כמויות קטנות יחסית של נתונים ובנוסף, מאפשר מחיקה ותכנות מחדש של "בתים" ייעודיים.

EEPROM בנוי כמערך של טרנזיסטורים בעלי שער צף. ניתן למחוק ולתכנת EEPROM, על ידי מתן אותות מיוחדים. במקור, EEPROM היה מוגבל לפעולות על "בית" אחד, מה שגרם לו להיות איטי יותר, אבל EEPROM מודרני מאפשר פעולות "מרבובות בתים". כמו כן, קיימת מגבלת אורך חיים למחיקה ותכנות מחדש, אשר עומדת על סדר גודל של מיליון פעולות עבור EEPROM מודרני. אם כן, אורך חיי ה- EEPROM אשר מתוכנת לעיתים קרובות כאשר המחשב נמצא בשימוש, הינו שיקול חשוב בבנייה ותכנון המערכת כולה.

שלא כמו רוב הסוגים של זיכרון בלתי נדיף, EEPROM מאפשר בדרך כלל ל- "בתים" לקרוא, להימחק, ולהיכתב מחדש באופן אינדיבידואלי. EEPROM נמחק על ידי חשיפת השבב לאור אולטרה סגול אשר מוחק את כל התוכן שלו. זיכרון FLASH לעומת זאת, ניתן לתכנות ומחיקה רק בקבוצות של "בתים", הנעות בין עשרות, לעשרות אלפי "בתים" עבור מכשירים שונים.

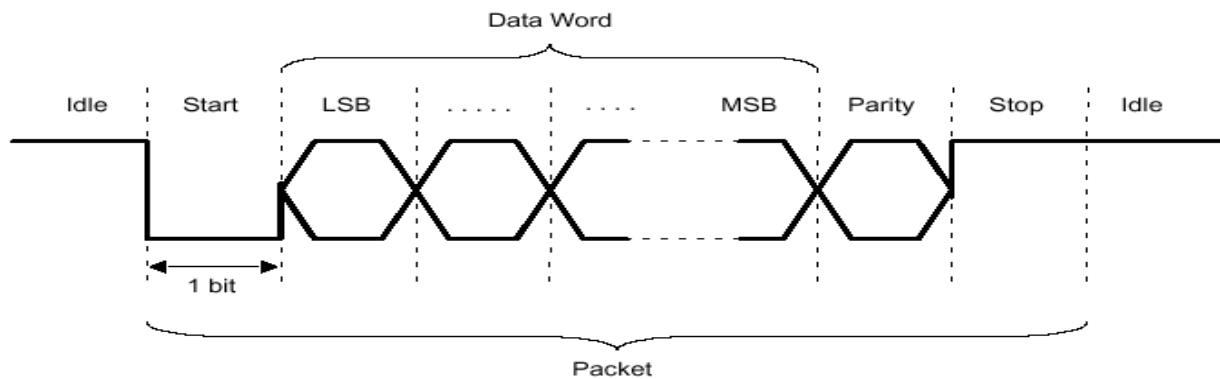


איור 9 – זיכרון מסוג EEPROM

UART – Universal Asynchronous Receiver Transmitter 2.9

2.9.1 מבוא

UART הינו ממשק סריאלי אסינכרוני, האחראי לקידוד מידע לשידור ופענוח בעת הקליטה. מידע הנקלט או משודר ע"י ה-UART דורש סיבית התחלה (logic low) בתחילת החבילה וסיבית סוף (logic high). חבילת מידע נשלחת החל מביט ה-LSB עד לביט ה-MSB. למטרות גילוי שגיאות, ניתן להוסיף את ביט הזוגיות. האות על הקו תמיד גבוה עד אשר מתחילה זרימת מידע. דוגמא של חבילת מידע כזו ניתן לראות בציור הבא:



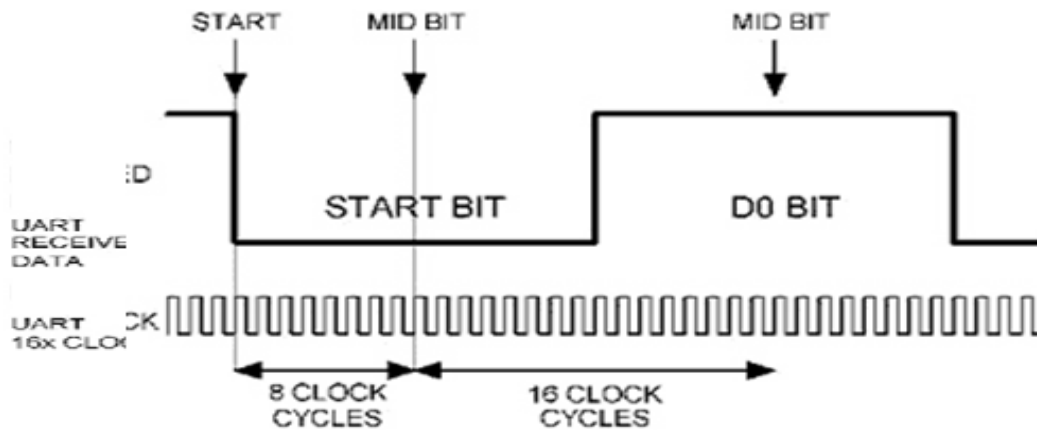
איור 10 – חבילת מידע הנשלחת/נקלטת ע"י UART

2.9.2 פונקציונאליות

באופן כללי יש ל-UART את היכולות הבאות:

- גודל מידע משתנה - התוו הנשלח יכול להיות בעל 5,6,7 או 8 סיביות.
- מספר משתנה של סיביות עצירה - יכולות להיות 1, 1.5 או 2 סיביות עצירה.
- קצב תקשורת (Baud rate) הניתן לוויסות ע"י תכנות.
- גילוי אוטומטי של קצב התקשורת - זה מאפשר ל-UART לגלות את הקצב התקשורת של המשדר שמשדר אליו
- יצירת זוגיות - כאשר משדרים תו ישנה אפשרות לשלוח את סיבית הזוגיות לגילוי שגיאות בשידור. אחת האפשרויות היא Even parity כלומר מספר האחדים בחבילה הינו זוגי.
- גילוי זוגיות - כאשר מתקבל תו, ה-UART יבדוק את סיבית הזוגיות ויוודא שזה תואם את קביעת הזוגיות (Parity setting) של התקשורת.
- Set Break - זה יגרום לשליחת רצף של אפסים הגדול בהרבה מגודל חבילה תקנית. אופציה זו מאפשרת אינדיקציה של מאורעות מיוחדים כגון שינוי קצב התקשורת

- Break Indicator - ל - UART יש יכולת לגלות Break condition (רצף של אפסים). אירוע זה גורם לעדכון רגיסטר סטאטוס בערך שנקבע מראש.
 - גילוי מסגרת שגויה - מאפשר לאותת אם התגלתה סיבית עצירה לא תקינה.
 - גילוי דריסה - מאותת על שגיאה אם מילה מסוימת התקבלה לפני שהמילה הקודמת נקראה בכלל מהחוצץ. מחזיר שגיאה לרגיסטר הסטאטוס.
- כאשר שני UARTs צריכים לתקשר, הם יודעים את קצב האיתות המוסכם ביניהם אך המקלט אינו יודע מתי חבילה תישלח (אין שעון מקלט). לכן, בפרוטוקול שמכונה "אסינכרוני" קביעת קצב דגימת המקלט הוא יותר מורכב מזה של המשדר. משדר פשוט משדר מסגרת של נתונים בקצב סיביות מוגדר ומוסכם. אך עבור המקלט יש לזהות את תחילת שידור המסגרת כדי לסנכרן את הדגימה, וכך לקבוע את נקודת הדגימה הטובה ביותר עבור המקלט.



איור 11 – סנכרון קצב שידור המידע המגיע מהמשדר למקלט

איור 11 מציג שיטה נפוצה לסנכרון קצב הדגימה במקלט בהתאם למסגרת שהתקבלה. ה-UART מקבל למעשה קצב שעון שהוא פי 16 מקצב שידור הנתונים. זיהוי מסגרת חדשה מתרחש כאשר האות משתנה ממצב גבוה לנמוך ולאחר 8 מחזורי שעון, מכאן נקודת האמצע של כל ביט תופיע כל 16 מחזורי שעון לאחר מכן. שיפור נוסף הוא לדגום רק שלוש פעמים (7, 8, ו-9, מתוך 16) במקום שמונה דגימות (8 מתוך 16). הקונפיגורציה הבסיסית של ה-serial port של ה-PC היא כדלקמן:

Baud rate : 9600 bits/sec
 Data bits : 8 bits
 Parity : Even
 Stop Bits : 1
 Flow control: Hardware

פרוטוקול UART נתמך ב- PC ע"י יציאה טורית RS-232 בהנחה שעשינו את הקונפיגורציה המתאימה לתקשורת. פירוט הפינים ביציאה הטורית של ה- PC :

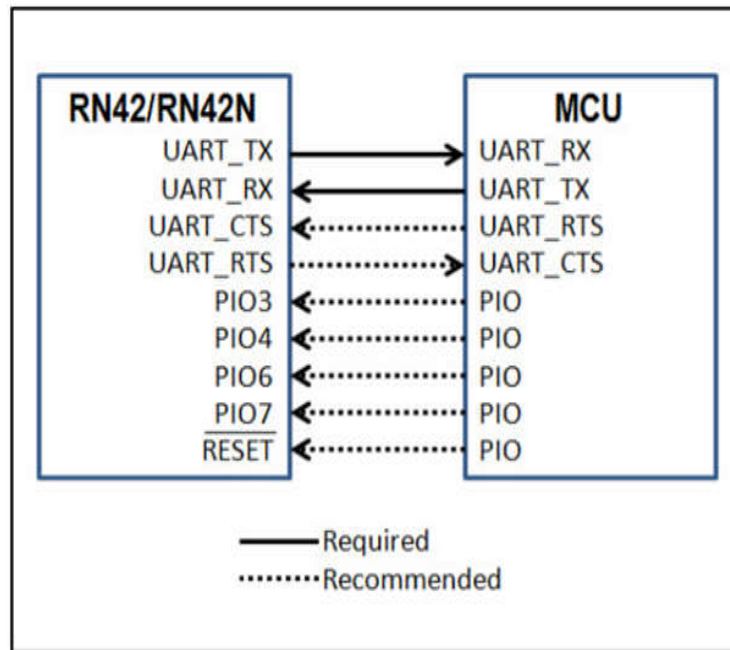
D-Type-9 Pin No.	Abbreviation	Full Name	Function
Pin 3	TD	Transmit Data	Serial Data Output (TXD)
Pin 2	RD	Receive Data	Serial Data Input (RXD)
Pin 7	RTS	Request To Send	This line informs the Modem that the UART is ready to exchange data
Pin 8	CTS	Clear To Send	This line indicates that the Modem is ready to exchange data
Pin 6	DSR	Data Set Ready	This tells the UART that the modem is ready to establish a link.
Pin 5	SG	Signal Ground	
Pin 1	CD	Carrier Detect	When the modem detects a "Carrier" from the modem at the other end of the phone line, this Line becomes active.
Pin 4	DTR	Data Terminal Ready	This is the opposite to DSR. This tells the Modem that the UART is ready to link.
Pin 9	RI	Ring Indicator	Goes active when modem detects a ringing signal from the PSTN

Bluetooth 2.10

Bluetooth (או שן כחולה) הוא תקן תקשורת אלחוטית ליצירת רשת במרחב האישי (PAN), ברמת אבטחה גבוהה על ידי שימוש בגלי רדיו באורכי גל קצרים. Bluetooth משמש לתקשורת נתונים למרחקים קצרים בקצבים קטנים ובהספק חשמלי נמוך ויכול לחבר עד 8 מכשירים. רשתות Bluetooth משמשות לרוב לשימושים אישיים, כמו קשר בין מחשב לציוד היקפי או בין טלפון סלולארי לדיבורית. Bluetooth פועל בטווח התדרים 2400-2483.5 מגה הרץ שזה טווח התדרים קצרי-הטווח (ISM) שחופשיים מרישוי, המשמשים לתעשייה, רפואה ומחקר.

Bluetooth משתמש בטכנולוגיית רדיו שנקראת FHSS, המידע המועבר מחולק לחבילות כאשר כל חבילה מועברת באחד מ 79 ערוצי Bluetooth שונים ולכל חבילה רוחב פס של 1 מגה הרץ. בלוטות' 4.0 משתמש ברוחב פס של 2 מגה הרץ שמאפשר 40 ערוצים. הערוץ הראשון מתחיל בתדר 2402 מגה הרץ וממשיך עד לתדר 2480 מגה הרץ בקפיצות של 1 מגה הרץ והוא מבצע בדרך כלל 1600 קפיצות לשנייה כאשר AFH מופעל. בלוטות' הינו פרוטוקול מבוסס מיתוג מנות עם מבנה של אדון-משרת (Master-Slave). בכל פיקונט כל אדון יכול לתקשר עם 7 משרתים (למרות שלא כל המכשירים תומכים במספר זה) כאשר כל המשרתים חולקים את השעון של האדון. חילוף המנות מבוסס על השעון הבסיסי שנקבע על ידי האדון שעובד בטווחים של 312.5 מיקרו שנייה. שתי פעימות שעון מהוות חריץ נתונים אחד של 625 מיקרו שנייה, שני חריצים מהווים זוג חריצים של 1250 מיקרו שנייה, במקרה הפשוט של חבילות בחריץ אחד האדון משדר בזוגות וקולט ביחידים, ובהתאם המשרת משדר ביחידים וקולט בזוגות. חבילות יכולות להיות באורך של 1, 3 או 5 חריצים אבל בכל המקרים האדון צריך להתחיל את השידור בחריצים זוגיים והמשרת צריך לשדר בבודדים. במהלך ההתקשרות שני מכשירים, הם יכולים בתיאום להחליף תפקידים ביניהם כך שהאדון נהיה המשרת והמשרת נהיה האדון.

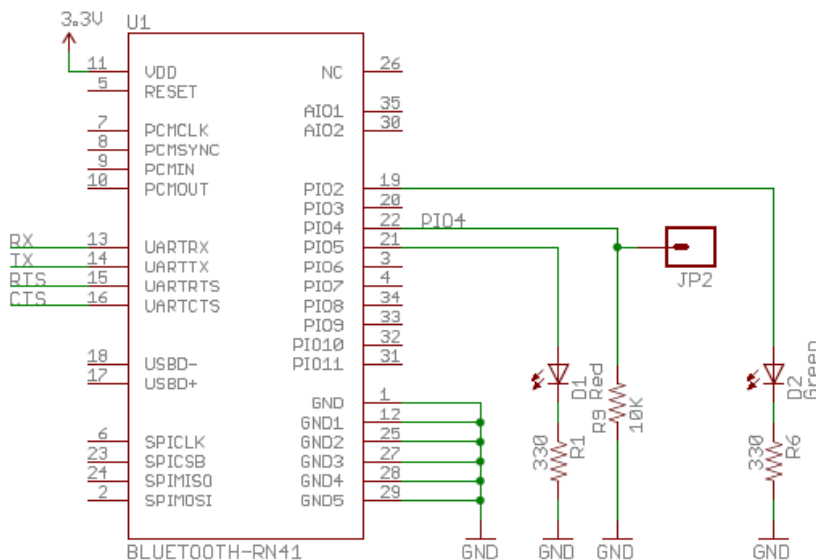
RN42 הוא רכיב תקשורת אלחוטית קטן בעל הספק נמוך ופשוט להטמעה שמשתמש בטכנולוגיית Bluetooth כדי לקשר בין מכשירים. הוא מתאים לעבודה ביישומים מבוססי בטרייה וברירת המחדל שלו היא עבודה בקונפיגורציית SPP (Serial Port Profile). צריכת האנרגיה שלו במנוחה היא 250 Micro Farad. הרכיב יכול לעבוד בקצב של עד 3 Mbps ולטווח של עד כ-20 מטר.



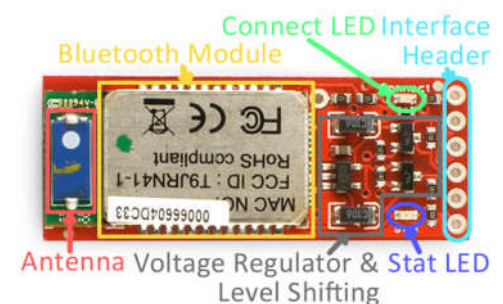
איור 13 – RN42/RN42N to MCU interface

SparkFun Bluetooth Modem – BlueSMiRFSilver 2.10.1

ה- "BlueSMiRFSilver" מתוצרת "SparkFun" הוא רכיב תקשורת שנועד להחליף שימוש בכבלים בטכנולוגיית Bluetooth. ברכיב מותקן מודול ה- Bluetooth אותו סקרנו קודם- RN42. הוא מגיע עם אנטנה מובנית והוא יכול לשדר בקצבים של בין 2400 עד 115200 ביטים לשנייה. על מנת להפעיל את הרכיב נדרש לספק לו ממתח חיצוני בין 3.3 ל- 6 וולט ונדרשת קונפיגורציה ראשונית מול מחשב.



איור 15 – תרשים רכיב ה- Bluetooth RN41



איור 14 – BlueSMiRFSilver Bluetooth modem

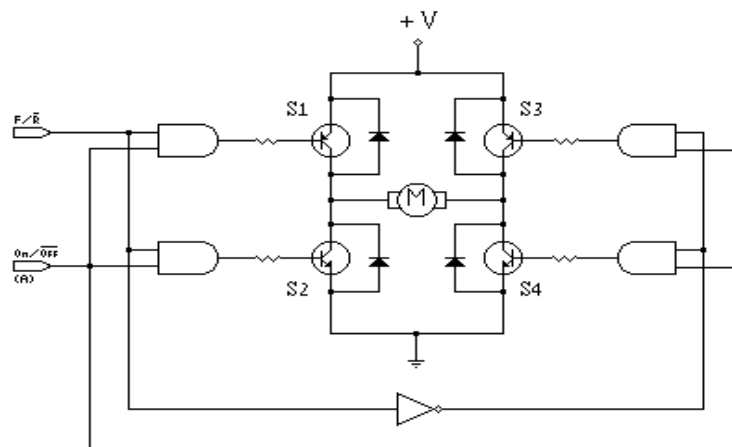
2.11 מנוע DC-Servo

למנועי Servo מסוג זרם יש פיר הקולט\מסור מידע למנוע, הפיר ימוקם בעזרת שליחת מידע מקודד למנוע. כאשר המידע הנקלט במנוע, גם המיקום הזוויתי של הפיר משתנה. סוגי מנועי זרם ישר Servo הם בדרך כלל חסרי מברשות או מנועי גיר מסוגים שונים. מנועים אלו הם קטנים וחזקים מאוד יחסית לגודלם. ברובוטיקה מכאטרוניקה משתמשים במנועים אלה כשילוב מנוע זרם ישר עם תיבת הילוכים ועם Potentiometer\Encoder לסגירת מעגל בקרה (בקרה בחוג סגור). כשבחרים מנוע Servo זרם ישר חשוב לשים לב למפרט הביצועים של המנוע הכולל:

- מהירות הפיר- מהירות סיבובית לא מועמסת של הפיר החיצוני.
- מתח סופי- מתח של מנוע זרם ישר.
- זרם מתמשך- הזרם המקסימאלי שיכול להיות מסופק לסלילי הרוטור בלי לגרום להם לחימום יתר.
- מומנט מתמשך- מומנט היציאה האפשרי של המנוע תחת תנאים קבועים.
- אנרגיה מתמשכת- האנרגיה המכאנית המסופקת ע"י המנוע.

2.11.1 אמצעי שליטה בכיוון ומהירות מנועי DC (גשר H)

הבקר איננו מסוגל להניע את המנועים ישירות, כיוון שהוא לא מספק מספיק זרם. כדי להתגבר על כך, הזרם מסופק למנועים ע"י בקר המנועים, והמעבד רק מעביר את אותות השליטה והבקרה לבקר זה. בקרים אופייניים של מנועי DC משתמשים בטופולוגיה בסיסית הקרויה גשר- H (H-Bridge), שהיא למעשה 4 מפסקים המחוברים בצורה דמוית H למנוע. ניתן ליישם מפסקים אלו באמצעות ממסרים, טרנזיסטור (ראה איור 16) MOSFET (Metal-Oxide semiconductor field effect transistors), או מעגל משולב (שבב) לבקרת מנועים.



איור 16 – מעגל בקרת מנועי DC בארכיטקטורת H-Bridge

2.11.2 סכמה חשמלית של H-Bridge

המעגל באיור 16 מתאר ארכיטקטורה פשוטה של מעגל בקרת מנועים. המעגל מורכב למעשה משני חלקים: חלק לוגי צירופי, ו-H-Bridge. תחומי העבודה של הטרנזיסטורים במעגל זה הם תחומי הרוויה והקטעון, כך שהם משמשים כמתגים.

הטרנזיסטורים ב-H-Bridge משנים את תחומי העבודה שלהם באופן כזה שיפעיל את המתח על המנוע בקוטביות מסוימת, שתגרום למנוע להסתובב בכיוון אחד, או בקוטביות ההפוכה, שתסובב את המנוע בכיוון ההפוך. לדוגמה, כאשר הטרנזיסטורים S1 ו-S4 באיור מצויים בתחום הרוויה, (משמשים כמתגים סגורים) ואילו הטרנזיסטורים S2 ו-S3 מצויים בתחום הקטעון (משמשים כמתגים פתוחים), הזרם יעבור במנוע משמאל לימין. כאשר הטרנזיסטורים S1 ו-S4 באיור בתחום הקטעון, והטרנזיסטורים S2 ו-S3 בתחום הרוויה, הזרם יעבור במנוע מימין לשמאל. אם כל הטרנזיסטורים בתחום הקטעון, המנוע יסתובב בחופשיות. וכאשר כולם בתחום הרוויה (הזרם עובר בחופשיות בכל חלקי המעגל), המנוע יבלום. הטרנזיסטורים S1, S3 הם מסוג NPN, ואילו הטרנזיסטורים S2, S4 הם מסוג PNP.

למעגל הלוגי נכנסים שני קווי קלט: קו האפשר, וקו הכיוון. המעגל, המורכב מ-4 שערי AND ושער NOT אחד, מספק את פונקצית הבסיס לכל אחד מהטרנזיסטורים ב-H-Bridge, כך שיפעלו בהתאם לטבלה שלמטה. כאשר קו האפשר במצב לוגי '0', הפלט של כל שערי ה-AND יהיה '0', ולא יעבור זרם ב-H-Bridge. כאשר קו האפשר במצב לוגי '1', זוג הטרנזיסטורים שימצאו בתחום הרוויה תלויים בערך המצוי בקו הכיוון.

2.11.3 סיכום אופן פעולת קווי בקרת המנוע לקביעת כיוון הסיבוב

מצב לוגי של קו האפשר	מצב לוגי של קו הכיוון	פעולת המנוע
1	1	סיבוב לכיוון אחד.
1	0	סיבוב לכיוון שני.
0	X	עצירת המנוע

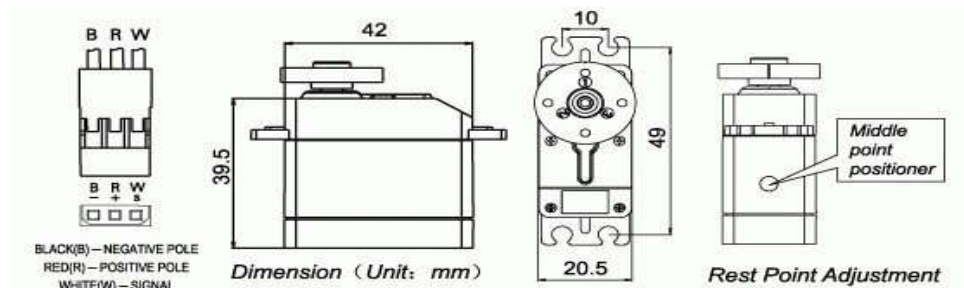
טבלה 17 – אופן פעולת קווי בקרת המנוע לקביעת כיוון הסיבוב

2.12 מנוע SM-S4303R

הוא מנוע Servo בגודל סטנדרטי המיוצר ע"י חברת SpringRC, והוא מנוע Servo רציף. במתח של שישה וולט הוא מגיע למהירות מרבית של 54 סיבובים לדקה (RPM) ומומנט של 5.1 ק"ג לס"מ. במנוע קיים פוטנציומטר עמו ניתן לשנות את נקודת המנוחה, ברירת המחדל היא 1.5 מילי שנייה מתוך מחזור של 20 מילישניות. עבור רוחב אות הגדול מ-1.5, המנוע יסתובב נגד כיוון השעון ועבוד רוחב אות מתחת ל-1.5, הוא יסתובב עם כיוון השעון. ניתן לשלוט ב-Servo עם קו בקרה בודד מבקר כלשהו במידה וניתן לו מתח ממקור חיצוני.

JR interface, 360°with Robot Servo, 4plastic gears+1metal gear.double ball bearing.

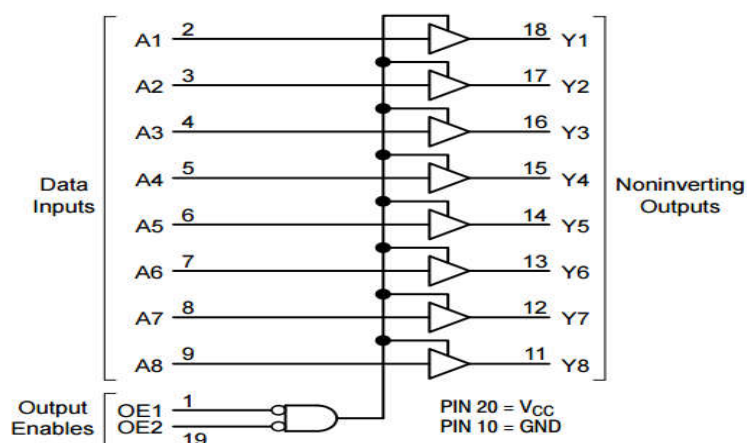
Products specification								Technical parameters						
Size (mm)					Weight		Wire	4.8V			6V			Rotation angle
A	B	C	D	E	g	oz		Speed	Torque		Speed	Torque		
R/M	kg-cm	oz-in	R/M	kg-cm	oz-in									
42.0	20.5	39.5	49.0	10.0	44	1.55	30.0	60.0	3.3	45.8	70	4.8	66.7	360°



איור 18 – מפרט טכני של מנוע Servo מסוג SM-S4303R

MC74HC541A – octal buffer 2.13

חוצץ שמונה רגלים, ממיר כניסת מתח בטווח מצוי למתח V_{CC} מבוקש. רכיב זה הוא למעשה אוסף של שמונה רכיבי Tri-state משורשרים המאפשרים מעבר לא מהפך של עד שמונה אותות. בנוסף לכניסות A1-A8 ולשמונת המוצאים המתאימים, לרכיב כניסת V_{CC} (מתח מבוקש), כניסת אדמה ושני Enables. בהכנסת 'L' (LOW) בשתי כניסות ה- Enables, החוצץ עובד כמגבר מתח, אחרת תתקבל במוצא עכבה גבוהה - 'Z'.



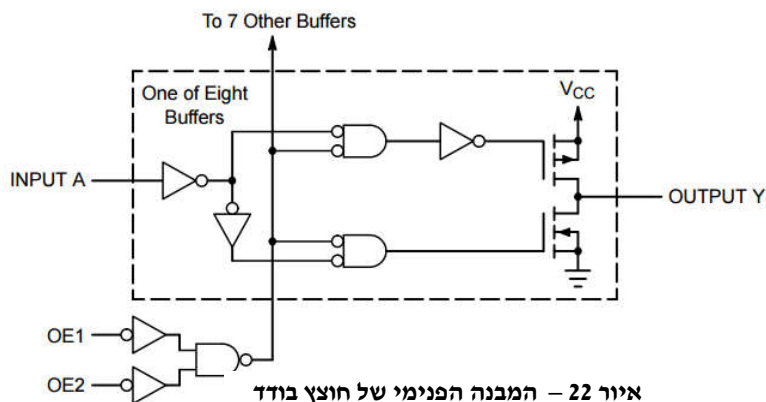
איור 20 – מבט כולל על כלל החוצצים ברכיב

Inputs			Output Y
OE1	OE2	A	
L	L	L	L
L	L	H	H
H	X	X	Z
X	H	X	Z

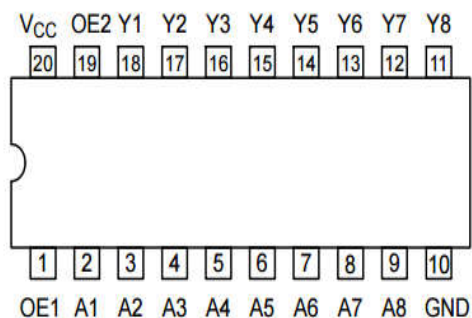
X = Don't Care

Z = High Impedance

איור 19 – Buffer function table



איור 22 – המבנה הפנימי של חוצץ בודד



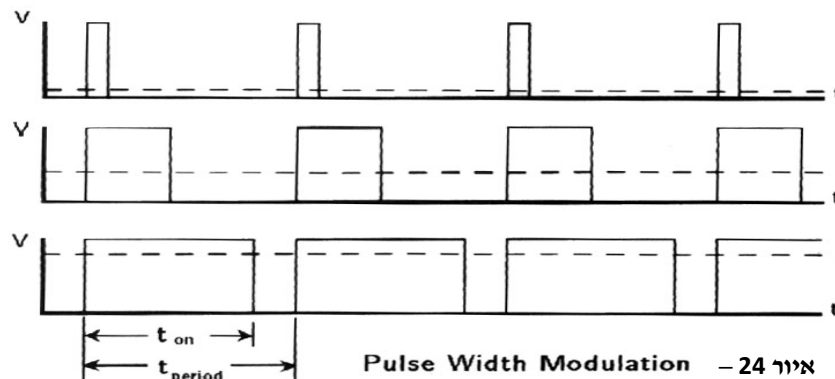
איור 21 – Buffer pin assignment

2.14 אפנון רוחב פעימה – PWM

על מנת לשלוט במהירות המנוע, המפסקים נפתחים ונסגרים בקצב שונה בכדי לספק מתח ממוצע שונה למנוע. טכניקה זו קרויה PWM (Pulse Width Modulation) ומודגמת באיור שלמטה, כאשר V הוא המתח על המנוע ו- t הוא הזמן. למשל, כאשר טרנזיסטורים S1 ו- S4 משמשים ל- PWM וטרנזיסטורים S2 ו- S3 נשארים בתחום הקטעון, המתח במנוע יהיה זהה ובעל אותה הקוטביות כמו מתח הכניסה כאשר S1 ו- S4 סגורים, ו- 0V כאשר הם פתוחים. בשיטה זו מוכנס לקו האפשרור גל ריבועי, בעל זמן המחזור t_{period} , כאשר הזמן בו הגל נמצא במתח לוגי '1' הוא t_{on} . וכתוצאה מכך, מהירות המנוע, ניתנת לשליטה ע"י שינוי רוחב-הפעימה (Duty Cycle או pulse-width ratio):

$$\text{Duty Cycle} = \frac{t_{on}}{t_{period}} \quad \text{איור 23 – נוסחה לחישוב Duty Cycle}$$

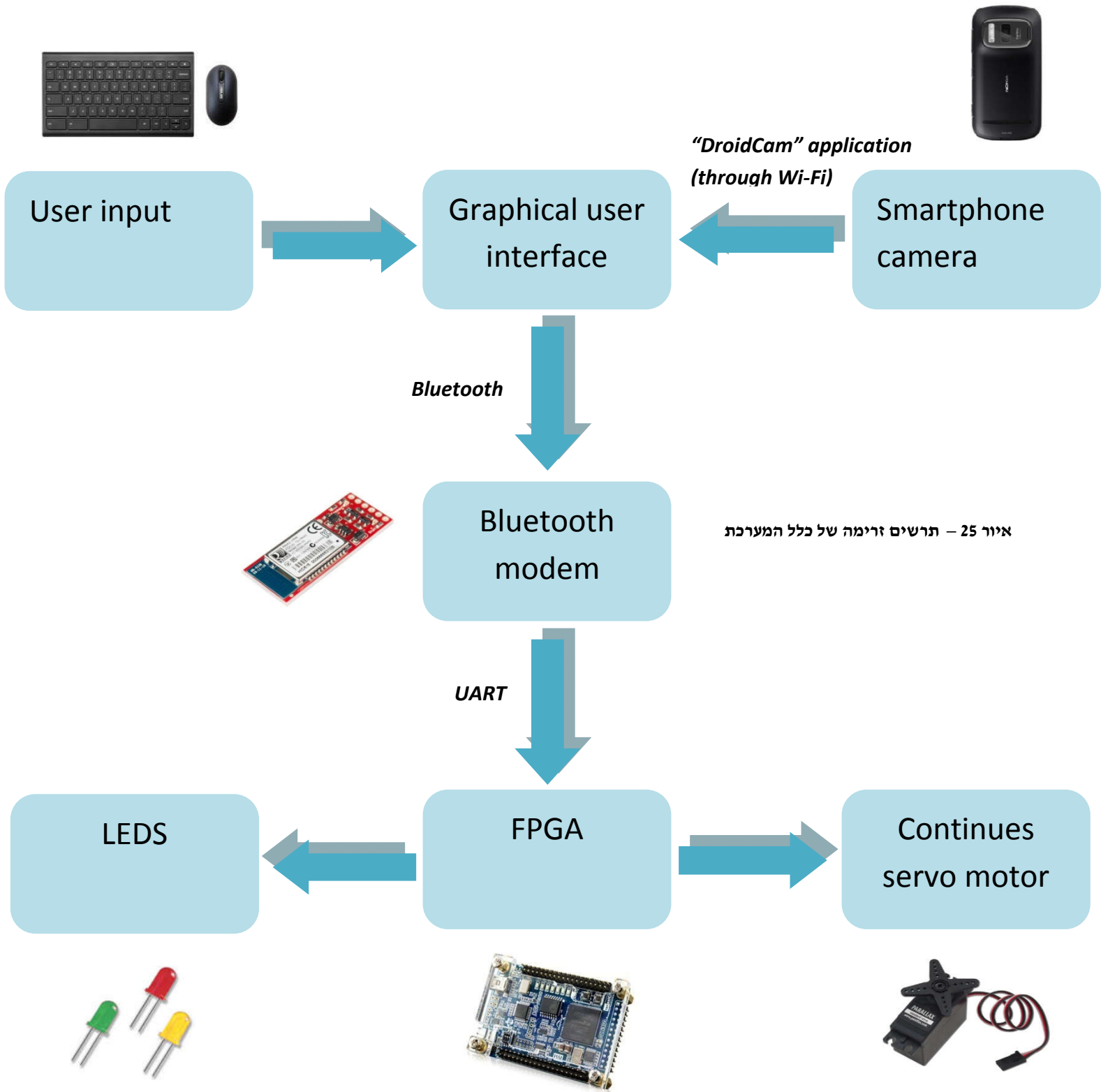
הזנת גל ה- PWM מבוצעת ע"י בקר המתוכנת מראש ע"פ הצרכים הנדרשים:



אפנון רוחב פעימת מתח הכניסה יוצר מתחים ממוצעים שונים. באיור, הקו המלא הוא הגל הריבועי המסופק לקו האפשרור, והקו המקווקו מייצג את המתח הממוצע המתקבל במנוע.

הפשטת המתגים ב- H-Bridge יכולה להיות ממומשת במספר דרכים. ממסרים יכולים להפעיל ולכבות מנועים, ואף לשלוט על כיוונם, אולם אין להשתמש בהם במערכות הדורשות בקרה על מהירות המנוע בשיטת PWM, בשל מהירותם התגובה האיטית והבלאי הגבוה שלהם. מימושים של מצב מוצק (Solid state), כמו טרנזיסטורים דו-קוטביים (Bipolar) ו-MOSFET נפוצים יותר במערכות כאלו.

פרק 3 – מבנה המערכת



GUI 3.1

3.1.1 עיקרון בניית ממשק המשתמש

היות והמערכת מכילה מס' מרכיבים אשר יש להם השפעה אחד על השני, מומלץ לרכז את כלל מרכיביה למקום מרכזי אשר יהווה תא שליטה ובקרה על תהליכי הביניים המבוצעים בה. פתרון נוח ויעיל הוא שימוש בממשק משתמש (GUI) בסביבת העבודה בה נכתב הקוד ומתבצעות הפעולות המרכזיות.

המחשב, באמצעות תוכנת ה-Matlab, משמש כמרכז העצבים של המערכת היות והוא מקבל קלט מהמשתמש וממצלמת סמארטפון המותקנת על גבי רובוט, מבצע עיבוד ושולח פלט לרובוט דרך משדר Bluetooth להמשך ביצוע משימות. בכדי לנהל את התהליך בצורה ידידותית ונוחה למשתמש, נבנה ממשק משתמש באמצעות כלים ייעודיים הקיימים בתוכנת ה-Matlab. ממשק המשתמש מכיל מס' פקדים ורכיבים בעיצוב גרפי מותאם אישית, אשר מאפשרים למשתמש לשלוט במערכת בצורה פשוטה וללא ידע בקוד או בפרטים טכניים שעומדים מאחורי בניית הממשק.

3.1.2 אופן פעולת ממשק המשתמש

GUIDE יוצר קובץ קוד ובו פונקציות Callback לכל רכיב בממשק המשתמש. ניתן להפעיל את האפליקציה הן מהתרשים הנבנה (קובץ FIG) והן מקובץ הקוד שנוצר. לכל פקד יש שם (tag name) אשר מופיע בפונקציה ה-Callback שלו וניתן לשינוי. בעת הפעלת פקד מסוים בממשק המשתמש, מתבצעת קריאה לפונקציות Callback המשויכות אליו ומכילה סט של פקודות הכתובות בה בצורה היררכית. לכל פקד מס' מאפיינים אשר שמורים במבנה הנקרא handles ובו שדה לכל פקד לפי ה-tag name שלו.

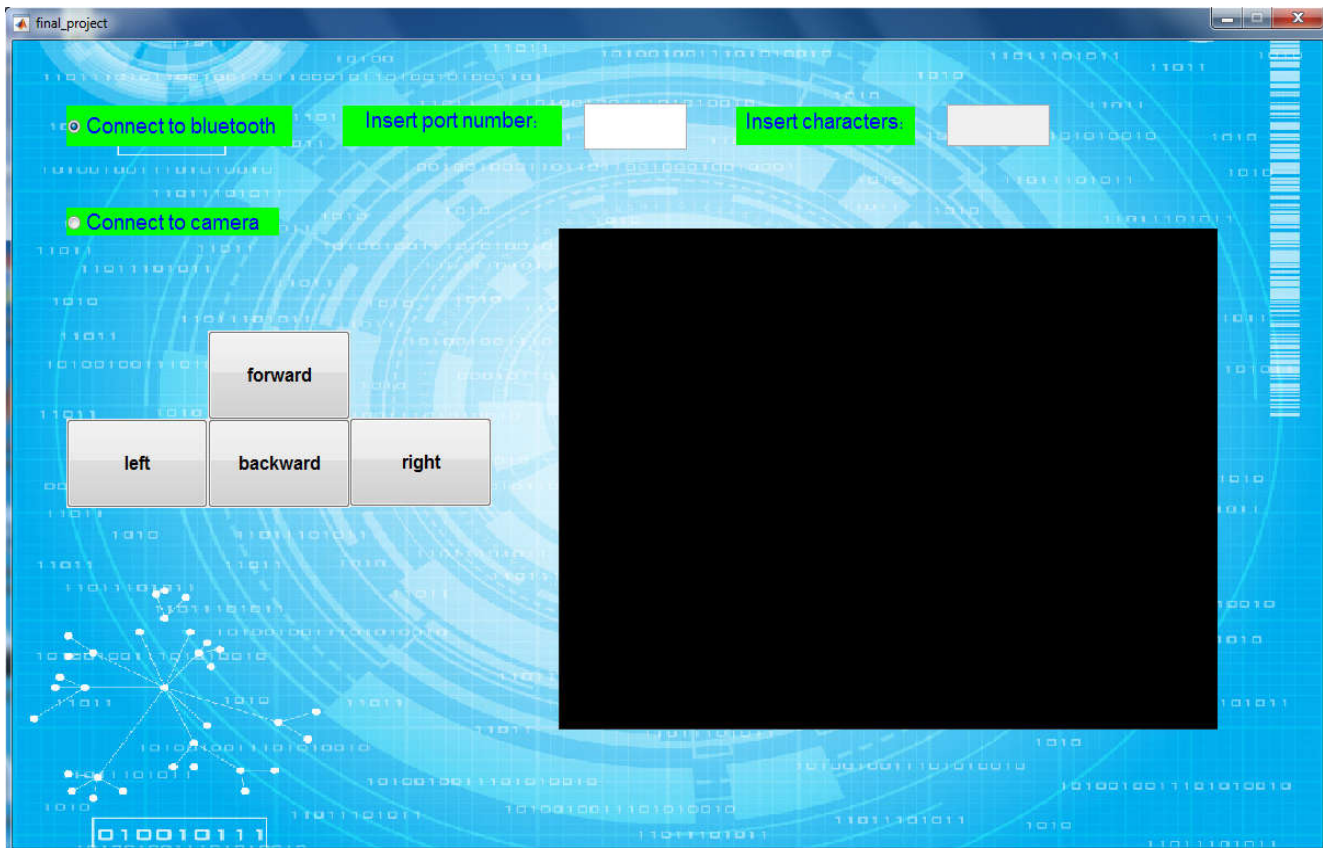
עקרון ביצוע הפונקציה מזכיר במידת מה תכנות בשפת C, אשר עושה שימוש ניכר במשתנים מקומיים (local) המשמשים לביצוע חישובים בגוף הפונקציה ונמחקים בעת סיומה. עם זאת, קיימים גם משתנים אשר ברצוננו לשמור לעת ביצוע פונקציות ה-callbacks הבאות ולא לאבד אותם ביציאה מהפונקציה הנוכחית, לכן הם מוגדרים ולאחר מכן נשמרים במבנה ה-handles וניתנים לקריאה ושינוי במהלך ביצוע פונקציות ה-callbacks הבאות. פקודת ה-get הינה פקודה המאפשרת לפנות לשדה בעל מאפיין של פקד מסוים ולקלוט את ערכו של מאפיין זה, ואילו פקודת ה-set מאפשרת השמה של ערך בשדה בעל מאפיין של פקד מסוים. בכל השמה של ערך ע"י הפקודה set, יש לשמור את כלל המבנה handles באמצעות הפקודה guidata.

בתהליך יצירת ממשק המשתמש ע"י פונקציה ה-GUIDE, נוצרת פונקציה פתיחה אשר מבצעת סט של פקודות הכתובות בה, בעת ההרצה הראשונית של ממשק המשתמש. בפונקציה הפתיחה מופיעים ומאותחלים כל המשתנים אשר יצרנו וחלק מהמאפיינים של הפקדים בממשק המשתמש ונשמרים תחת המבנה handles. בנוסף, קיימת פונקציה סגירה שבה מתבצעת ניתוק התקשורת של משתנה המחזיק בתקשורת טורית (s). כמו כן מתבצעת מחיקה של כלל האלמנטים והפקדים המרכיבים את המבנה handles, על מנת לבצע שחרור זיכרון שאינו בשימוש יותר.

3.1.3 מבנה ממשק המשתמש

ממשק המשתמש מכיל את הרכיבים הבאים (ראה איור 26):

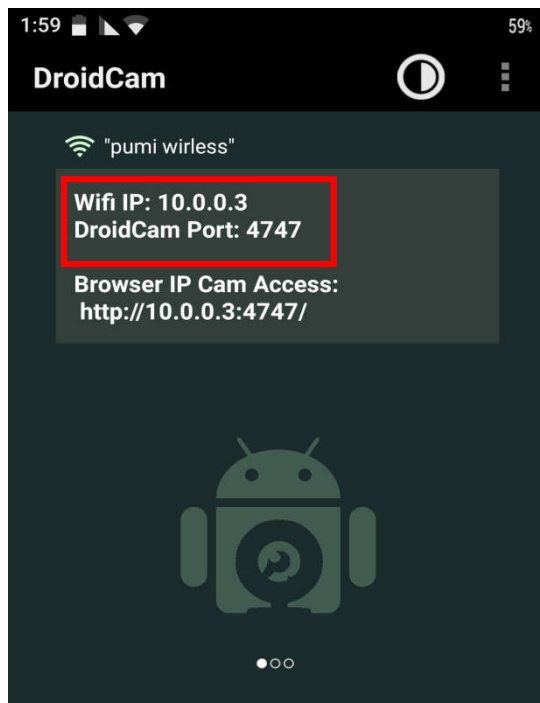
- Connect/disconnect from bluetooth : מתחבר או מתנתק ממקלט ה-Bluetooth שנמצא על הרובוט. ביצוע חיבור חיוני ליצירת קשר עם הרובוט למטרת ביצוע משימות.
- Insert port number : יש להזין את מס' היציאה המוגדרת כיציאת תקשורת טורית במחשב.
- Insert characters : ניתן להכניס בשדה בכל פעם תו אחד ולשלוח אותו למקלט ה-Bluetooth שנמצא על הרובוט. השדה פעיל רק לאחר שבוצעה התחברות למקלט ה-Bluetooth.
- פקדי תנועה : פקדים אשר מאפשרים שליטה מלאה בתנועתו של הרובוט קדימה, אחורה, ימינה ושמאלה. כאשר פקד נלחץ, הוא נצבע באפור כהה והוא יחדול מפעילות רק בעת לחיצה נוספת עליו, או ע"י לחיצה על פקד תנועה אחר.
- Connect to camera : מתחבר או מתנתק ממצלמת הסמארטפון שנמצאת על גבי הרובוט. תצוגת הקלט המגיע מהמצלמה, תופיע על גבי החלונית השחורה שבמרכז ממשק המשתמש. לפני הלחיצה של הפקד, יש לוודא כי המחשב והסמארטפון מחוברים על אותה רשת אינטרנט וכי האפליקציה הייעודית בה אנו משתמשים כמצלמת רשת (DroidCam) פתוחה ועובדת על גבי המחשב וקולטת את הוידאו המשודר מהסמארטפון.



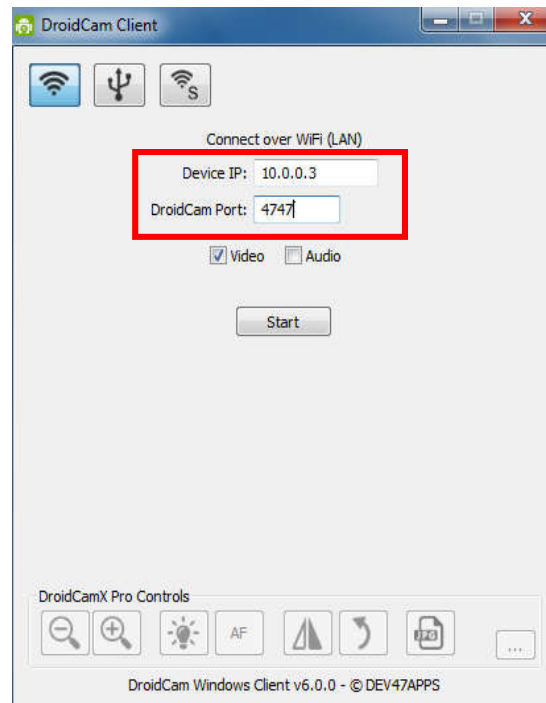
איור 26 – מבנה ממשק המשתמש

3.1.4 שידור הוידאו מהסמארטפון ל- Matlab

מצלמת הסמארטפון מהווה מצלמת רשת שכן היא מעבירה וידאו בזמן אמת למחשב באמצעות אפליקציה ייעודית למכשירי אנדרואיד אשר נקראת DroidCam. בנוסף לאפליקציה המותקנת בסמארטפון, יש לאפליקציה צד לקוח (Client side) המותקן במחשב. שידור הוידאו נעשה באמצעות רשת ה- WiFi כאשר שתי הצדדים מחוברים אליה ויש להם כתובת IP משותפת.

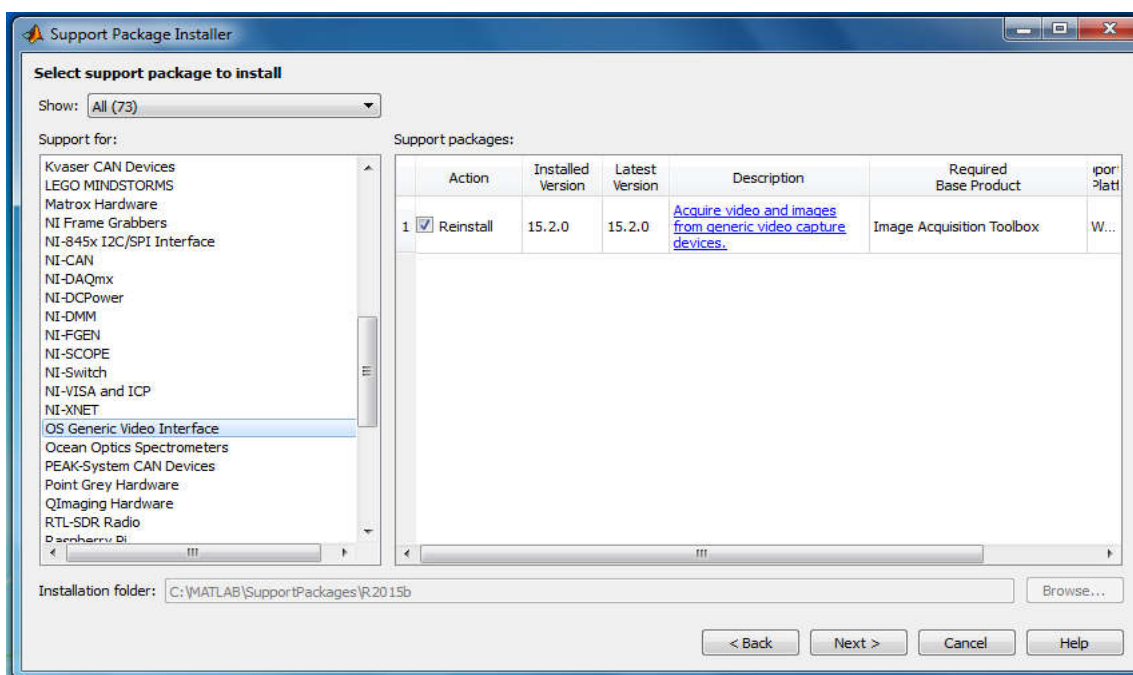


איור 28 – האפליקציה DroidCam המותקנת במכשיר ה- Android



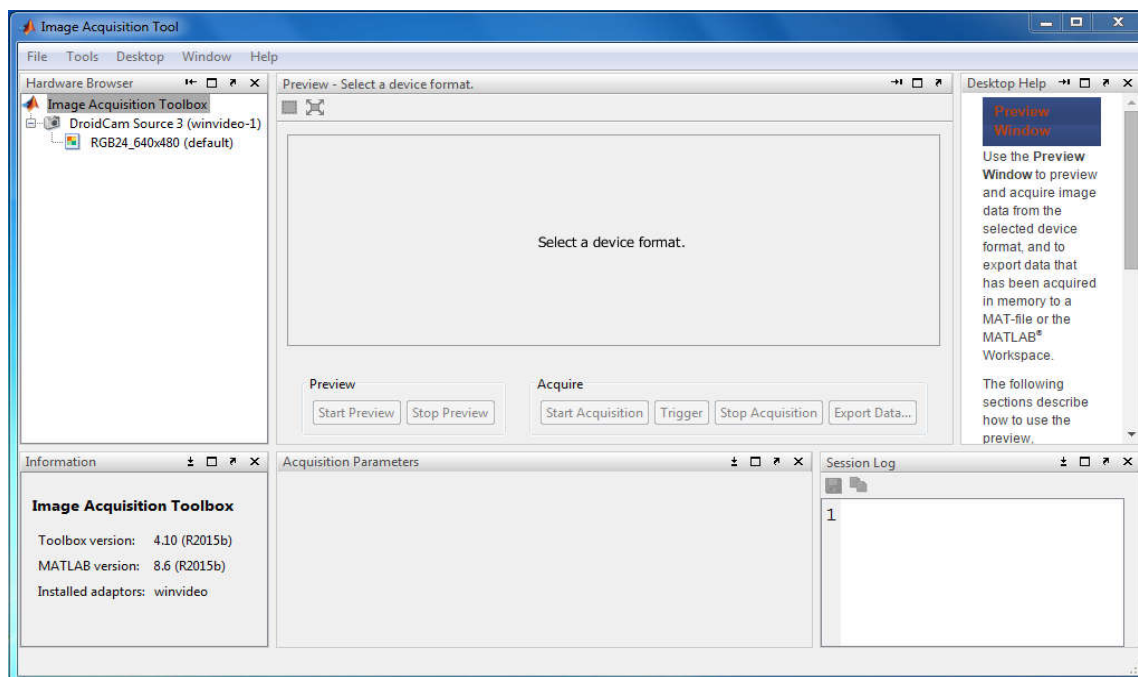
איור 27 – צד לקוח של האפליקציה DroidCam

בכדי לפתוח את הוידאו המשודר בצד הלקוח של האפליקציה באמצעות תוכנת ה- Matlab יש להגדיר תחילה את האפליקציה כמצלמת רשת. התהליך נעשה באמצעות ארגז כלים מיוחד של ה- Matlab לעבודה עם סרטוני וידאו אשר נקרא "Image Acquisition Toolbox". תחילה יש לדאוג למתאם וידאו אשר מתאים לשידור סרטון ברזולוציה המבוקשת שהיא 640X480. המתאם הרצוי לפורמט נקרא 'winvideo' והוא מותקן באמצעות חבילת תמיכה ייעודית של Matlab בשם "Os Generic Video Interface" אשר ניתן להוריד אותה מתוך שלל חבילות תמיכה אשר מוצעות ע"י Matlab (לצורך התהליך יש להתחבר לאתר של Matlab באמצעות חשבון משתמש אישי).



איור 29 – Matlab Support Package Installer

לאחר התקנת חבילת התמיכה, ניתן לראות את שם המתאם ואת שמה של המצלמה בחלונית הצדדים. כמו כן ניתן לבצע הצגה של הוידאו המגיע ממצלמת הרשת ולשלוט במסך מאפיינים כגון קצב שידור הוידאו (פריימים לשנייה), ניגודיות, בהירות, Zoom ועוד.



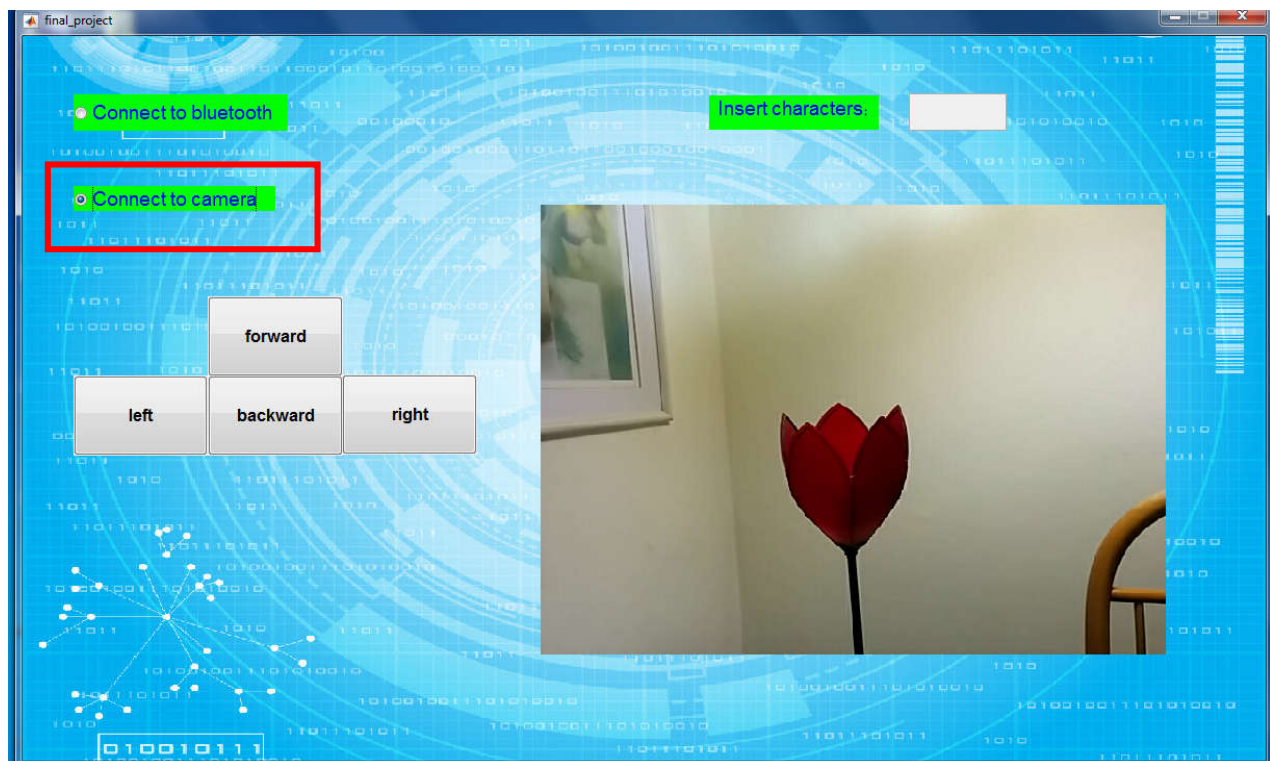
איור 30 – Matlab Image Acquisition Tool

3.1.5 פתיחת הוידאו בממשק המשתמש

לאחר ייבוא הוידאו ממצלמת הסמארטפון למחשב והצגתו דרך ה- "Image Acquisition Toolbox" ב-Matlab, ברצוננו להציגו על גבי מסך הנמצא בתוך ממשק המשתמש. בכדי ליצור מסך אשר יהיה תואם לרזולוציית הוידאו המתבקשת (640X480), יצרנו מטריצת אפסים בגודל 640x680 ובנינו רכיב מסוג מערכת צירים (axes) אשר עליו מוצגת המטריצה.

ע"י שימוש בפקודה הקרויה videoinput או שומרים את קלט הוידאו המגיע ממצלמת הרשת לתוך משתנה בשם vid אשר נשמר במבנה handles לצורך המשך פעילותו לפעולות נוספות מחוץ לפונקציה בה הוא נוצר. ע"י שימוש בפקודה preview או מציגים את תוכן המשתנה (להלן, הוידאו) על גבי המטריצה שבנינו, אשר מוצגת על גבי מסך שחור (מערכת הצירים).

הוידאו מוצג על גבי המסך השחור בעת לחיצה על על הכפתור "connect to camera" ונסגר בעת לחיצה נוספת עליו. לאחר הצגת הוידאו על גבי המסך השחור, טרם שלב ביצוע הקוד העוסק בעיבוד התמונה המופיעה בו, נשלחת פקודת "hold on" אשר מאפשרת הצגת אלמנטים נוספים על גבי התמונה ללא מחיקתה.



איור 31 – הצגת הוידאו בממשק המשתמש

3.1.6 יצירת תקשורת טורית בעזרת ממשק המשתמש

בכדי להעביר את המידע המעובד מממשק המשתמש למקלט ה-Bluetooth אשר נמצא על גבי הרובוט השתמשנו בתקשורת טורית. כאמור, ניתן לשדר מידע דיגיטלי בשתי דרכים עיקריות: תקשורת טורית או תקשורת מקבילית. יתרונה העיקרי של תקשורת מקבילית הוא המהירות שבה מועבר האות, שכן ניתן להעביר בשיטה זו מס' ביטים במקביל. לעומת זאת, בתקשורת טורית המידע עובר ביט אחר ביט.

למרות יתרון זה ומכיוון שאין ברצוננו להעביר כמויות מאסיביות של מידע, בחרנו לשדר את המידע בתקשורת טורית. בנוסף, לתקשורת טורית ישנם מס' יתרונות בולטים, כגון יחס אות לרעש נמוך ומרחק שידור מקסימלי גדול יחסית לשיטת התקשורת המקבילית. עוד נקודה חשובה היא שבתקשורת מקבילית קיים ה- ISI (Intersymbol interference) אשר גורם לאות להתעוות ובכך בעצם מוריד את קצב העברה (חבילת תווים) שנשלחת בצורה משובשת, נשלחת שוב עד שהיא מתקבלת בהצלחה.

טרם יצירת התקשורת הטורית בין המחשב ומקלט ה-Bluetooth אשר על גבי הרובוט, יש לאתחל מס' הגדרות כגון, פורט היציאה לתקשורת טורית (COM), קצב שידור הנתונים (Baud rate), מס' סיביות המידע אשר נשלחות בכל חבילת מידע, סיביות זוגיות, וסיביות עצירה. פירוט נוסף על צורת התקשורת הטורית, תהליך האתחול ואופן שליחת המידע וקליטתו, מוצג בהמשך בפרק 3.3.1.

בדומה לפתיחת הוידאו בממשק המשתמש, הוגדר משתנה אשר יוצר את התקשורת הטורית ונשמר במבנה ה- handles. בין שדות המשתנה ניתן לראות ולשלוט בהגדרות החיבור וולדהליק/לכבות אותו. יצירת המשתנה נעשית ע"י קבלת קלט מהמשתמש לאחר לחיצה על הפקד "connect to camera" אשר פותח שדה שאליו יש להזין את מס' היציאה הטורית שהוגדרה מבעוד מועד בהגדרות המחשב. כאן המקום לציין, כי יצירת החיבור לא תמיד צולחת בפעם הראשונה משום שתוכנת ה- Matlab אינה ייעודית לתקשורת טורית, בניגוד לתוכנות ייעודיות אשר עשינו בהם שימוש לצורך כיוול ובדיקה, כדוגמת תוכנת "Tera Term". באם, ניסיון החיבור לא צלח, תצא הודעת שגיאה ויש לנסות שוב. במידה והחיבור צלח, כפתור החיבור יהפך ל- disconnect וכמו כן יופיע אור ירוק בנורית ה- Led שבמקלט ה- Bluetooth. כעת מממשק המשתמש ערוך ומוכן לשליחת אותות לרובוט.



איור 32 – מקלט ה-Bluetooth במצב מחובר

3.1.7 שליחת אותות ותפעול פקדי התנועה

ממשק המשתמש מאפשר תפעול פשוט בכיוון נסיעתו של הרובוט. השליטה מתבצעת ע"י ארבעה כפתורי תנועה (Forward, Backword, Right, Left) אשר מזכירים שליטה ברכב במשחק מחשב. בניגוד למשחק מחשב אשר בו הרכב נוסע רק בעת לחיצה ממושכת על אחד מחצאי הכיוון, תנועת הרובוט מתבצעת ע"י לחיצה בודדת על אחד מפקדי התנועה בממשק המשתמש.

הפקד אשר נלחץ נצבע באפור כהה ויפסיק את תנועת הרובוט רק בעת לחיצה נוספת עליו או על פקד כיוון אחר שיצבע כעת הוא באפור כהה. כל פקד אשר נלחץ שולח אות למשדר ה- Bluetooth אשר נמצא על גבי הרובוט ומתורגם לתנועה בכיוון הרצוי. כאמור, בכדי להשתמש בפקדי התנועה יש לבצע חיבור ע"י תקשורת טורית בין ממשק המשתמש ובין הרובוט.

התווים הנשלחים בעת לחיצה על הפקדים הם:

- 'f' – בעת לחיצה על פקד Forward.
- 'b' – בעת לחיצה על פקד Backward.
- 'r' – בעת לחיצה על פקד Right.
- 'l' – בעת לחיצה על פקד Left.
- '0' – בעת לחיצה נוספת על פקד פעיל.

בנוסף לפקדי התנועה יש אפשרות לשליחה ידנית של אותות מממשק המשתמש לרובוט, דרך שדה אשר נמצא בחלקו העליון של ממשק המשתמש ורשום לידו "Insert characters". יש לשים לב כי השדה פעיל רק לאחר ביצוע התקשורת הטורית בין ממשק המשתמש והרובוט וכי ניתן לשלוח בכל פעם רק תו אחד בלבד. לאחר שליחת התו השדה מתנקה אוטומטית מתווים. כאמור, שליחת התווים אשר הוזכרו לעיל תגרור את תנועת הרובוט בכיוון המתאים ושליחת כל תו אחר, תחדול את תנועת הרובוט (זהה לשליחת התו '0' אשר מתבצעת אוטומטית בעת לחיצה על פקד תנועה פעיל).

נוסף על התווים הנ"ל, ישנם שלושה תווים אשר מדליקים את שלושת נוריות ה- Led שנמצאות על גוף הרובוט בהתאמה. אופן הדלקת הנוריות יידון בהרחבה בפרק שעוסק בזיהוי האובייקטים ע"פ צורה וצבע.

התווים האחראים להדלקת נוריות ה- Led הם:

- 'G' – מדליק את ה- Led הירוק.
- 'R' – מדליק את ה- Led האדום.
- 'Y' – מדליק את ה- Led הצהוב.

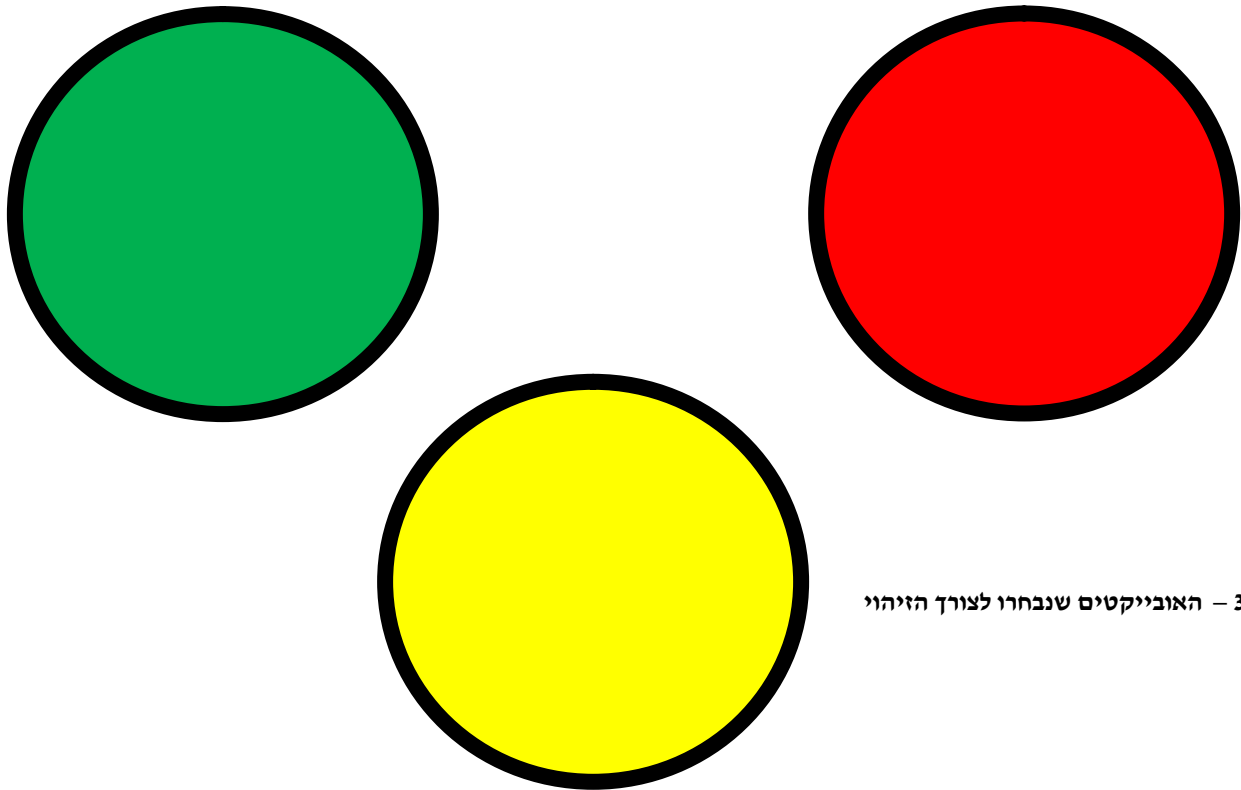
3.2 זיהוי אובייקטים ע"פ צורה וצבע

3.2.1 רציונל

אחד האתגרים הטכנולוגיים בתחום הראייה הממוחשבת הוא זיהוי אובייקטים וסיווגם. קיימות מס' טכניקות מרכזיות בעולם עיבוד התמונה בכדי לזהות אובייקט, אשר המשותף למרביתן הוא ביצוע ניתוח באמצעות אלגוריתמים מתמטיים. מרבית הטכניקות מתמקדות בשני תכונות מרכזיות של העצם המזוהה והן צורתו והצבע שלו.

זיהוי צורתו של האובייקט יכולה להיות פעולה מורכבת אשר דורשת את איתורו במרחב התמונה וסיווגו ע"פ המבנה הגיאומטרי שלו. לעומת זאת, זיהוי מרכיבי הצבע של האובייקט, הינה משימה פשוטה יותר שכן מלבד פענוח גווני הצבע המיוצגים במטריצת התמונה הנבדקת ע"י ערכים מספריים, אינה דורשת פעולות מתמטיות מורכבות. אך למרות הפשטות היחסית, זיהוי מרכיבי הצבע הינו תהליך הרבה פחות מדויק מזיהוי צורתו של האובייקט, היות והוא מושפע מרעשים לא רצויים המופיעים בתמונה ושינויי תאורה בין סביבה אחת לאחרת.

בכדי לבחון את עניין זיהוי האובייקטים ולממשו בצורה אופטימאלית, החלטנו לשלב בין שתי הטכניקות של זיהוי ע"פ צורה וע"פ צבע. המשימה שהוגדרה הייתה לזהות אובייקטים בצורת מעגל בשלושה צבעים שונים הפזורים במרחב החדר. צבעי המעגלים אשר נבחרו לצורך הזיהוי הם אדום, צהוב וירוק וזו הייתה בחירה שרירותית לחלוטין. הצורה אשר נבחרה לצורך הזיהוי היא מעגל, שכן זהו אובייקט גיאומטרי נפוץ בעל תבנית מתמטית, מה שיוצר אתגר בדמות מציאת אלגוריתם מתמטי אשר ימומש כקוד לצורך זיהויו.



איור 33 – האובייקטים שנבחרו לצורך הזיהוי

3.2.2 זיהוי מעגל ע"י התמרת CHT

לצורך זיהוי המעגלים השמשנו בפונקציה ייעודית של Matlab בשם "imfindcircle" אשר נמצאת בספריית עיבוד התמונה- Image Processing Toolbox. הפונקציה מבצעת התמרת Hough מעגלית כפי שהוסבר בחלק התיאורטי. היא מקבלת כקלט את התמונה בה יש ניסיון לזהות אובייקט מעגלי, את טווח הרדיוס הרצוי לזיהוי, את מידת הניגודיות של האובייקט ביחס לרקע בו הוא מופיע (בהיר או כהה ממנו) ואת סף הרגישות הדרוש לזיהוי. הפונקציה מחזירה כפלט, ווקטור ובו הרדיוסים של המעגלים המזוהים ואת שיעורי קואורדינטות נק' המרכז שלהם.

הפונקציה עושה שימוש באלגוריתם לזיהוי קצוות (Edge detection) אשר מבוסס על פעולה מתמטית שמטרתה למצוא את קווי הקצה של עצמים בתמונה. קווי קצה (שפות) מוגדרים כגבול בין רמות בהירות שונות. הקצוות שזוהו, מהווים את קו המתאר של האובייקט הנבדק, אותו הפונקציה סורקת ובודקת ע"י תהליך הצבעה אשר מתבצע בתוך סוכסם תלת מימדי, כפי שכבר הוסבר.

ובכן, החלק העיקרי שנבדק במעגלים, הוא קו המתאר שלהם ולכן המעגלים בהם השתמשנו (איור 33) הם בעלי קו מתאר שחור אשר נמצא ברמת ניגודיות גבוהה לחלק הפנימי של המעגל ולסביבתו החיצונית. כמו כן העלנו מעט את רמת סף הרגישות של הזיהוי, מה שמאפשר זיהוי פחות "קשוח".

3.2.3 זיהוי צבע המעגל

זיהוי צבע המעגל מתבצע ע"י ניתוח ערכי ה- RGB של התמונה הנבדקת. כאמור, כל תא במטריצת התמונה מכיל 3 ערכים מספריים אשר משקפים את רמת הבהירות של כל גוון יסוד (אדום, ירוק, כחול). כל ערך הוא בטווח המספרים '255' - '0', כאשר '0' זוהי רמת הבהירות הכי נמוכה (שחור) ו-'255' הכי גבוהה (לבן).

בכדי לבדוק את הצבע של התמונה בנינו שלוש פונקציות משנה, אחת לכל גוון. עקרון הפעולה של שלושת הפונקציות זהה, כל פונקציה מקבלת כקלט, חלק בתמונה (מטריצה) אשר זוהה עוד קודם כמעגל, סוכמת את ערכי הפיקסלים בטווח הגוון שהגדרנו בהתאם למדדים אשר בדקנו מראש, ומחזירה את מס' הפיקסלים שסכמה. בנוסף הגדרנו פרמטר סף זיהוי אשר קובע את סף רגישות זיהוי הגוון בדמות אחוז הפיקסלים שהוחזרו מפונקציה מתוך כלל הפיקסלים שנשלחו אליה (גודל המטריצה המכילה את המעגל שזוהה בלבד). קבענו את סף הזיהוי ל- 80% אך ניתן לשנותו דרך הקוד.

הטווחים אשר הוגדרו בכל פונקציה לזיהוי גוון צבע, הם על סמך רמות הבהירות של צבעי היסוד כמפורט בטבלאות הבאות:

• Red:

גוון יסוד	Blue (3)	Green (2)	Red (1)
טווח ערכים	$\text{Image}(i,j,3) \leq 150$	$\text{Image}(i,j,2) \leq 150$	$\text{Image}(i,j,1) \geq 100$
הפרש מקסימלי בין גווני היסוד	$ \text{Image}(i,j,3) - \text{Image}(i,j,2) \leq 50$		

• Green:

גוון יסוד	Blue (3)	Green (2)	Red (1)
טווח ערכים	$\text{Image}(i,j,3) \leq 150$	$\text{Image}(i,j,2) \geq 100$	$\text{Image}(i,j,1) \leq 150$
הפרש מקסימלי בין גווני היסוד	$ \text{Image}(i,j,3) - \text{Image}(i,j,1) \leq 50$		

• Yellow:

גוון יסוד	Blue (3)	Green (2)	Red (1)
טווח ערכים	$\text{Image}(i,j,3) \leq 150$	$\text{Image}(i,j,2) \geq 150$	$\text{Image}(i,j,1) \geq 150$
הפרש מקסימלי בין גווני היסוד	$ \text{Image}(i,j,2) - \text{Image}(i,j,1) \leq 50$		

טבלה 34 – טווחי רמות הבהירות לזיהוי הגוונים השונים

3.2.4 שילוב ניתוח הצורה ובדיקת הצבע לכדי זיהוי כולל

בכדי לבחון את סרט הוידאו המגיע מהסמארטפון למחשב ומשם ל-Matlab, היה עלינו לדגום אותו בפרק זמן מסוים שיאפשר ניתוח תמונה בודדת ללא שיבושים ותקיעת התוכנה אשר פועלת ברקע. דגימת התמונה וניתוחה, התבצעה בעזרת לולאה אשר פועלת עד כיבוי המצלמה בממשק המשתמש וחוזרת לפעולה בעת הפעלתה מחדש. הלולאה מכילה סדר פעולות היררכי עם השהייה מכוונת של 400 מילישניות בין פעולה לפעולה על מנת לאפשר פעולה חלקה ושוטפת.

זה המקום לציין כי לצורך ביצוע המשימות, השתמשנו במצלמת סמארטפון ומחשב אישי פשוטים ללא יכולות עיבוד מיוחדות כמו הכלים המקובלים בעולם התעשייה ולכן על אף המאמץ לייעל את הקוד, נתקלנו לעיתים בבעיות של עיבוד איטי ועיכוב בהגעת הסרט מהמצלמה למחשב.

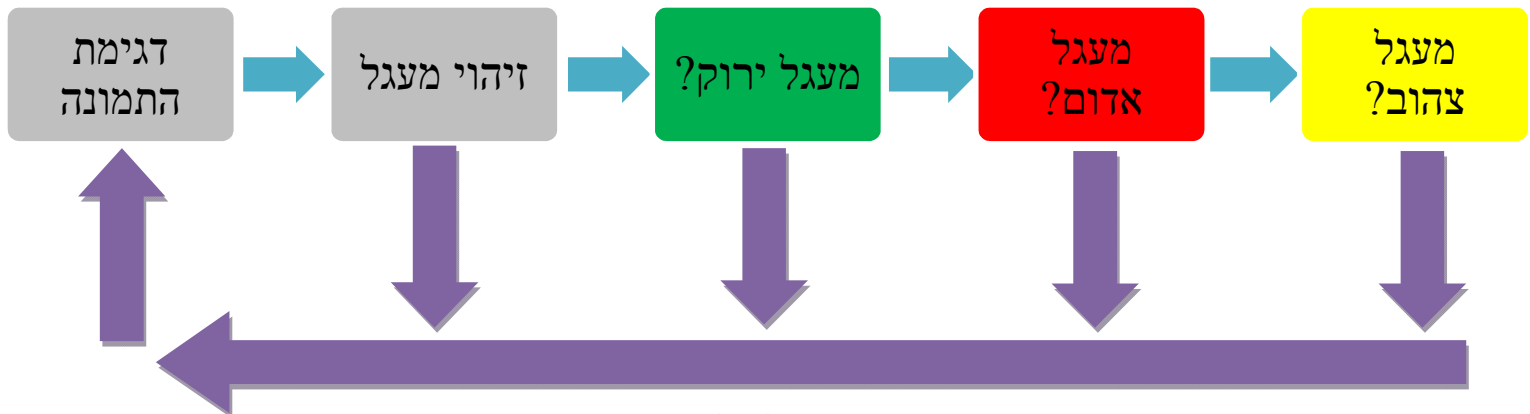
כחלק ממאמצי הייעול, ביצענו את ניתוח הצורה למציאת מעגל לפני ניתוח הצבע. באם התקבל זיהוי של מעגל יצרנו תמונה ריבועית חדשה אשר חוסמת את המעגל בתוכה וממוקדת לאזור שבו תתבצע בדיקת הצבע. זהו חלק קריטי שכן רוב חלקי התמונה אינם רלוונטיים לשלב ניתוח הצבע ומה שכן רלוונטי הוא פנים המעגל. ניתן לייצג את אחוז השטח אותו תופס המעגל, מכלל התמונה החדשה שיצרנו ע"פ הפקטור הבא:

$$\text{איור 35 – אחוז שטח המעגל מהתמונה ה- "חתוכה"} \quad factor = \frac{S_{\text{מעגל}}}{S_{\text{ריבוע}}} = \frac{\pi R^2}{4R^2} = \frac{\pi}{4} = 0.7853 = 78.53\%$$

התמונה ה- "חתוכה" נשלחת לפונקציות לזיהוי גוון אשר סקרנו, ע"פ הסדר: Green → Red → Yellow. באם התבצע זיהוי מוצלח של המעגל בגוון הצבע המסוים, תנועת הרובוט מופסקת ונורית ה-Led שעליו מתחילה לרצד בצבע המתאים לגוון שזוהה. כמו כן, האובייקט שזוהה מוקף במעגל בצבע שלו. באם לא התבצע זיהוי מוצלח, הלולאה עוברת לבדיקת הגוון הבא וכן הלאה עד לגוון האחרון, בו היא חוזרת לתחילת רצף הפעולות שראשיתו, איתור מעגל בתמונה שנדגמה (ראה איור 36).

לאחר ביצוע זיהוי מוצלח של אובייקט, והנעתו של הרובוט בכיוון מסוים, תכונה הנורית המרצדת וימחק המעגל אשר מקיף את האובייקט ותהליך הזיהוי יחול מחדש.

חשוב לציין שגם אם התבצע זיהוי מוצלח והרובוט עומד במנוחה, הניתוח ממשיך להתבצע ברקע וניתן לראות זאת ע"י הזזת האובייקט כל עוד הוא בטווח אשר קולטת עדשת המצלמה ואינו מרוחק מידי ממנה. ניתן לראות את המעגל המצויר סביב האובייקט עוקב אחריו כל עוד לא יצא מהפריים. זהו מימוש ידוע ונפוץ בשימוש בתחום הראייה הממוחשבת שמאפשר מעקב אחר אובייקטים דינאמיים (Object Tracking).



איור 36 – תרשים זרימה של תהליך זיהוי האובייקט

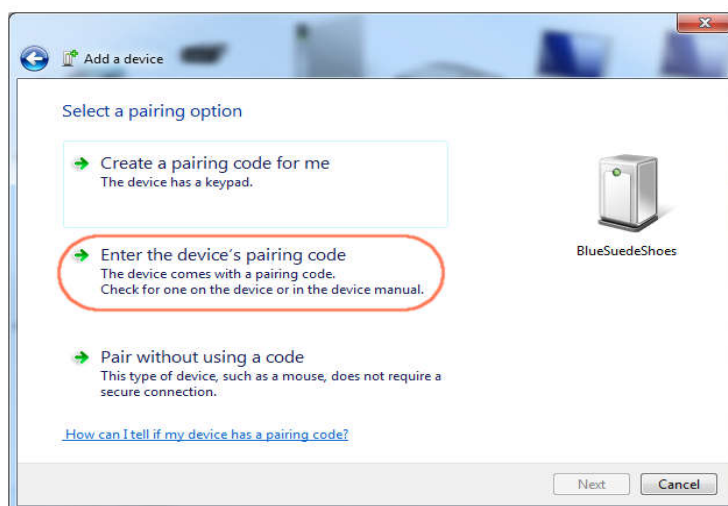
UART 3.3

3.3.1 חיבור המודם למחשב

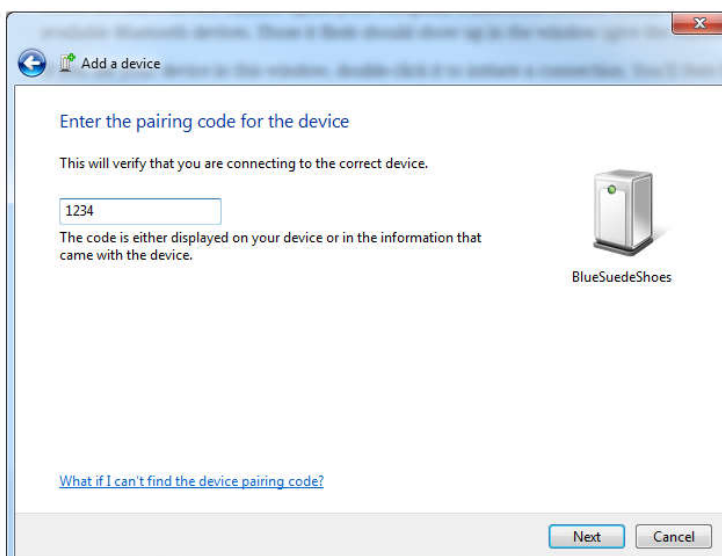
בפרויקט אנחנו עבדנו עם מודם Bluetooth תוצרת microchip בשם RN42 אותו סרקנו בפרק המבוא, המודם מורכב על ציף של חברת "Sparkfun" בשם "Bluesmirf silver". שימוש במודם זה מאפשר לנו לחסוך את השימוש בכבלים ומאפשר תנועה פשוטה יותר וחופשייה ממגבלות חוטיות. לפני שמחברים את המודם למחשב יש לוודא כי המחשב יכולת להתחבר למכשירי Bluetooth, במידה ואין, ניתן לרכוש דונגל חיצוני שמתחבר בחיבור USB למחשב כדוגמת הדונגל בו השתמשנו ולאחר מכן יש לבצע אתחול ראשוני של המודם.



איור 37 – דונגל Bluetooth דגם CSR 4.0

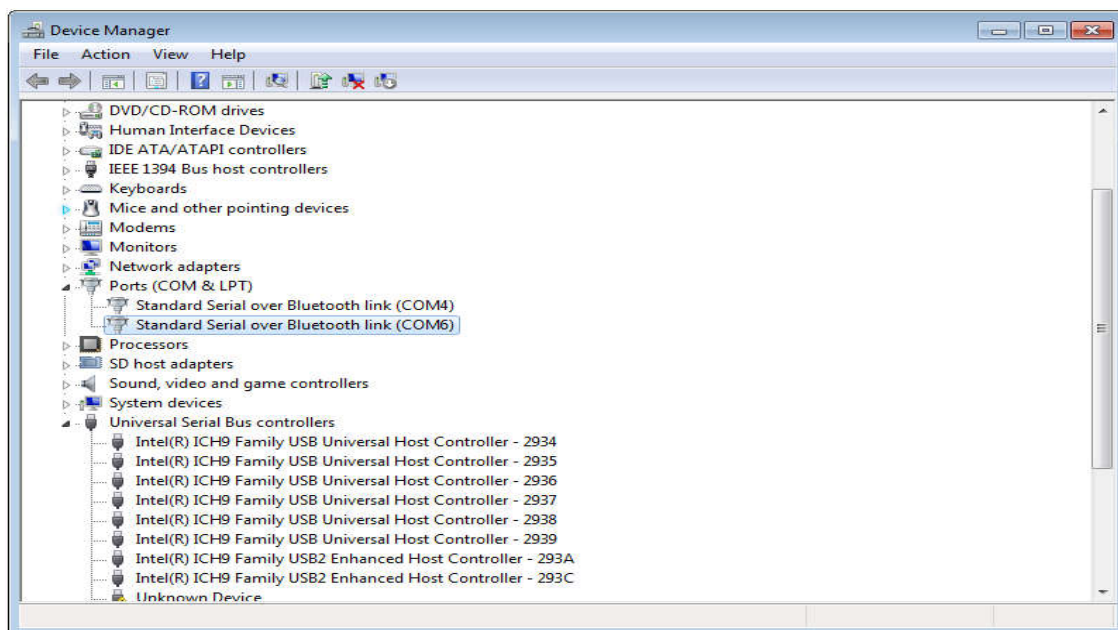


השלב הראשון הוא ביצוע pairing (זיווג) בין המחשב עמו עובדים ובין המודם כמו בכל מכשיר Bluetooth. הקוד הראשוני הוא 1234 וניתן לשינוי בהמשך.



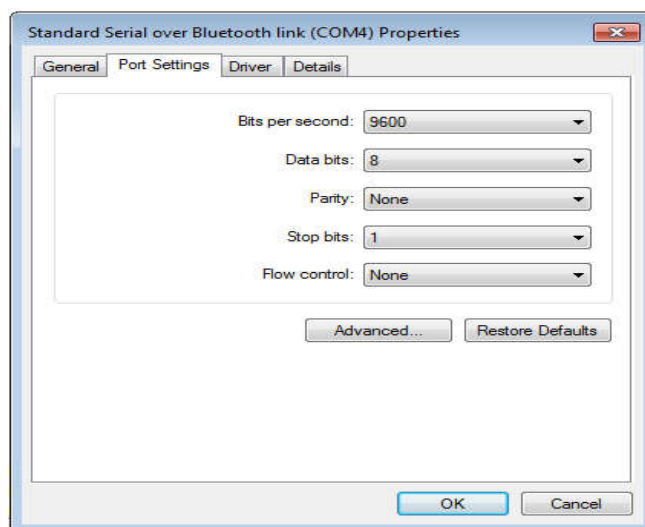
איור 38 – ביצוע Pairing בין המחשב ומודם ה- Bluetooth

בשלב זה המודם מחובר למחשב עם שני מספרי COM אחד לכתיבה והשני לקריאה, וניתן להתחיל להתאים אותו לדרישותינו.



איור 39 – ה- COM שנוצר במנהל ההתקנים

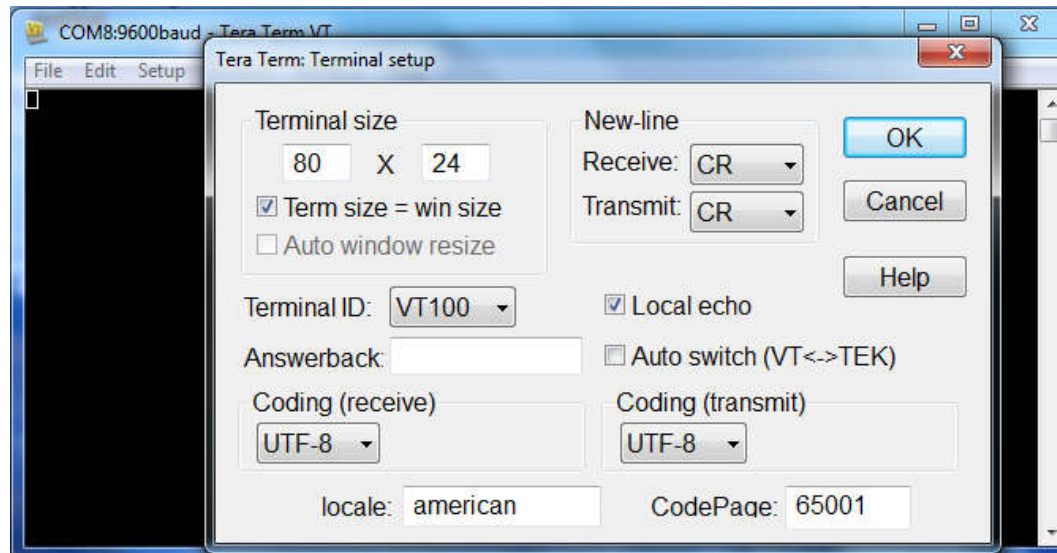
כל כתיבה וקריאה מהמודם תבוצע דרך הפורטים הללו וניתן לראות ולשנות את ההגדרות של המודם דרך מנהל ההתקנים ב- Windows.



איור 40 – הגדרות יציאת ה- COM

בשלב זה השתמשנו בתוכנת "Tera Term" כדי לשנות את הגדרות ברירת המחדל של המודם אך ניתן להשתמש בכל תוכנת תקשורת טורית (כדוגמת putty).

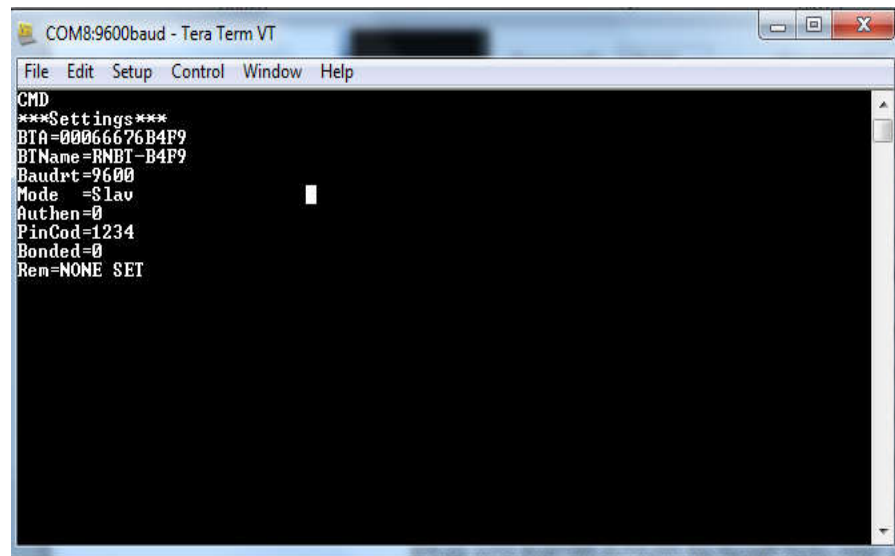
בשלב הראשון על מנת לראות את התווים שאנו שולחים למודם בחרנו באפשרות Local echo :



איור 41 – אפשרור הצגת התוכן המוקלד בתוכנת ה- Tera Term

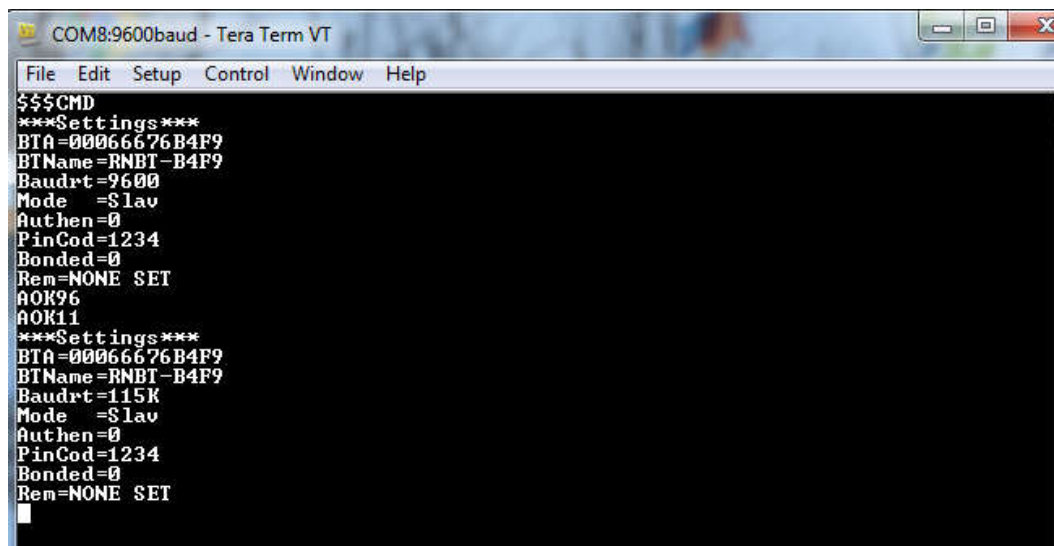
בשלב הבא שלחת רצף התווים "\$\$\$" מכניסה אותנו לתפריט ההגדרות ומאפשרת לנו שליטה בכל הפרמטרים שנוגעים לשידור התווים ע"י המודם החל בקצב השידור, דרך בחירת סיבית הזוגיות או שם המודם, כמו גם הקוד לביצוע הזיווג עם מחשבים. כאשר יקבל המודם את הפקודה \$\$\$, התשובה שנקבל תהיה CMD והנורה האדומה תהבהב בקצב מהיר. דבר זה מאותת לנו שאנו נמצאים בתוך מצב הקונפיגורציה.

בשליחת האות D, התפריט יפתח ונקבל למעשה את ההגדרות הקיימות כרגע במודם :



איור 42 – הצגת הגדרות מודם ה- Bluetooth

על מנת לשנות את ברירת המחדל של baud rate = 115200, ניתנת פקודה אשר צורתה היא <,>SU, כאשר במקום סוגריים נכתוב את שתי הספרות הראשונות של הקצב המבוקש. בכל שינוי שכזה, תשובת המודם תהיה <aok>, כלומר התקבל השינוי לקצב המבוקש.



```
COM8:9600baud - Tera Term VT
File Edit Setup Control Window Help
$$$CMD
***Settings***
BTn=00066676B4F9
BTName=RNBT-B4F9
Baudrt=9600
Mode=Slav
Authen=0
PinCod=1234
Bonded=0
Rem=NONE SET
AOK96
AOK11
***Settings***
BTn=00066676B4F9
BTName=RNBT-B4F9
Baudrt=115K
Mode=Slav
Authen=0
PinCod=1234
Bonded=0
Rem=NONE SET
```

איור 43 – שינוי ה-Baud rate בתוכנת ה-Tera Term

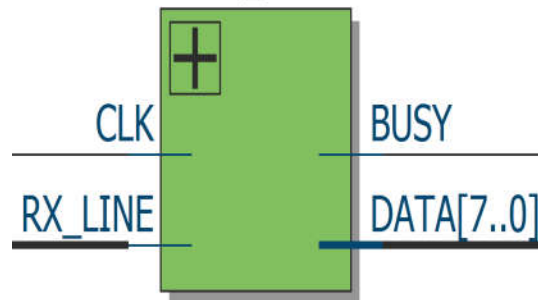
בכדי לסיים את העבודה בתפריט השינויים, נשלח את הרצף "----". איתות זה יפסיק את ההבהוב של הנורה האדומה והמודם יכנס למצב עבודה עם השינויים שהוזנו. בשלב זה ניתן להתחיל בחיבור המודם ללוח ה-FPGA ולהתחיל לשלוח אותות מהמחשב אל הלוח בלי שימוש בכבלים.

3.3.2 בניית ה-UART וחיבור

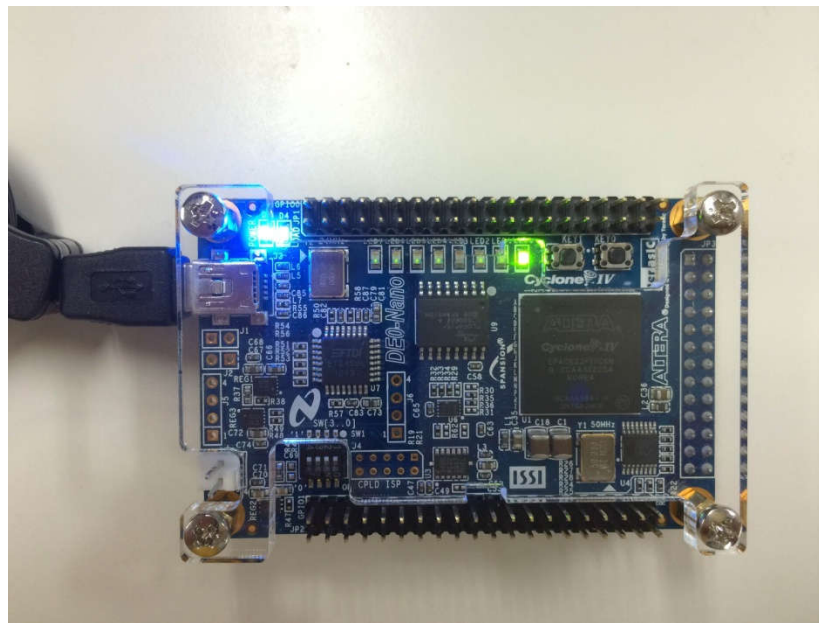
בבואנו לחבר את מודם ה-Bluetooth כמחבר בין המחשב לרכיב ה-FPGA עמו אנחנו עובדים, היינו זקוקים ל-UART שיתאים לעבודתנו ולצרכינו. ראשית מבחינת שידור/קליטה, ב-UART שלנו לא היה לנו צורך במשדר מכיוון שהשידור מבוצע בצורה חד כיוונית מממשק המשתמש ב-Matlab לכיוון ה-FPGA. שנית, מבחינת קצב שידור, בחרנו לעבוד בקצב של 9600 ביטים לשנייה ולכן התאמנו את ה-UART לקצב זה. כמו כן בחרנו לעבוד ללא סיבית זיהוי מסגרת וסיבית זוגיות.

UART_RX:C1

איור 44 – רכיב ה-UART



מבחינת הביצוע, מרגע שזיהינו סיבית 0 כהתחלת אות, אנחנו דוגמים תשע דגימות נוספות ולבסוף לפני שליפת האות מוודאים כי הביט הראשון הוא 0 והאחרון 1 (כך אנו מוודאים שנגמר שידור האות והקו חזר למתח גבוה), אז נשלף האות באורך שמונה סיביות להמשך עבודה. במידה והתנאי לא מתקיים אז סימן שהאות "נשבר" או שדגמנו רעש ולכן נעביר "00000000" להמשך ונמנע העברת מידע שגוי הלאה. על מנת לוודא כי האות הרצוי הגיעה בצורה תקינה ללוח ה-FPGA, ביצענו את הצגתו של הקוד ה-Ascii של האות הנשלח דרך שמות נוריות ה-Led המצויות בלוח.



איור 45 – הצגת קוד ה-Ascii של תו בנוריות ה-DE-nano

חיבור המודם לFPGA הושלם. את החיבור הפיזי יצרנו בין אחת מרגלי ה-FPGA לרגל TX-O במודם, בנוסף חיברנו מתח ואדמה למודם. בשלב זה המודם מחובר לרכיב ה-UART עובד. ניתן להתחיל בשליחת אותות מהמחשב אל ה-FPGA.

3.4 חיבור מוצאי המערכת ללוח ה-FPGA

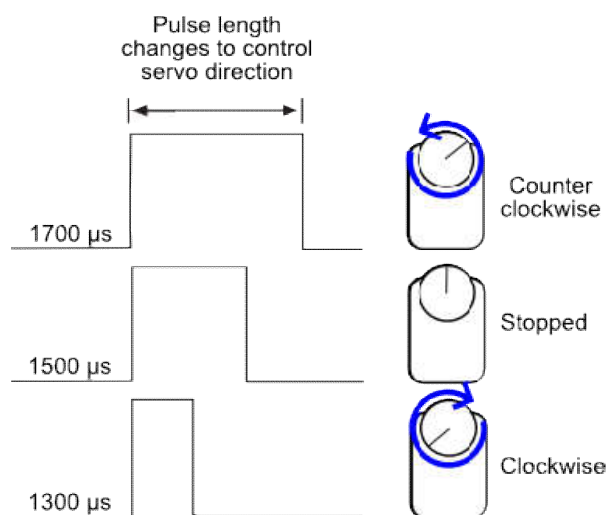
בפרק זה נסקור את דרך החיבור וההוראות השונות אשר ניתנות למוצאי המערכת. כאמור, מוצאי המערכת הם שלוש נורות LED בצבעים אדום, ירוק וצהוב בהתאמה לזיהוי הצבעים הנ"ל באלגוריתם הזיהוי שבנינו, ושני מנועי Servo הנשלטים דרך ממשק המשתמש במחשב ובאמצעותם נע הרכב במרחב. כל הוראות התנועה והחיווי מגיעות אך ורק מהמחשב, פעולת ה-FPGA היא רק כמעין מכונת מצבים משוכללת המעבירה כל העת מתח רצוי למוצאים הרלוונטים.

3.4.1 חיבור מנועי סרוו

בפרויקט שלנו עבדנו עם שני מנועי Servo של חברת "Springrc" אותם סקרנו בפרק המבוא, המנועים מסוג "continuous rotation", כלומר אין להם מגבלת תנועה סיבובית והם בעלי שלוש נקודות חיבור.



איור 47 – מנוע Servo מדגם SM-S4303R



איור 46 – כיווני תנועת מנוע ה-Servo

המחבר ב-Servo שלנו מכיל חוטים בצבעי אדום, שחור ולבן כפי שניתן לראות בתמונה:

- כבל אדום- 5 וולט.
- שחור- אדמה.
- לבן- PWM (pulse width modulation).

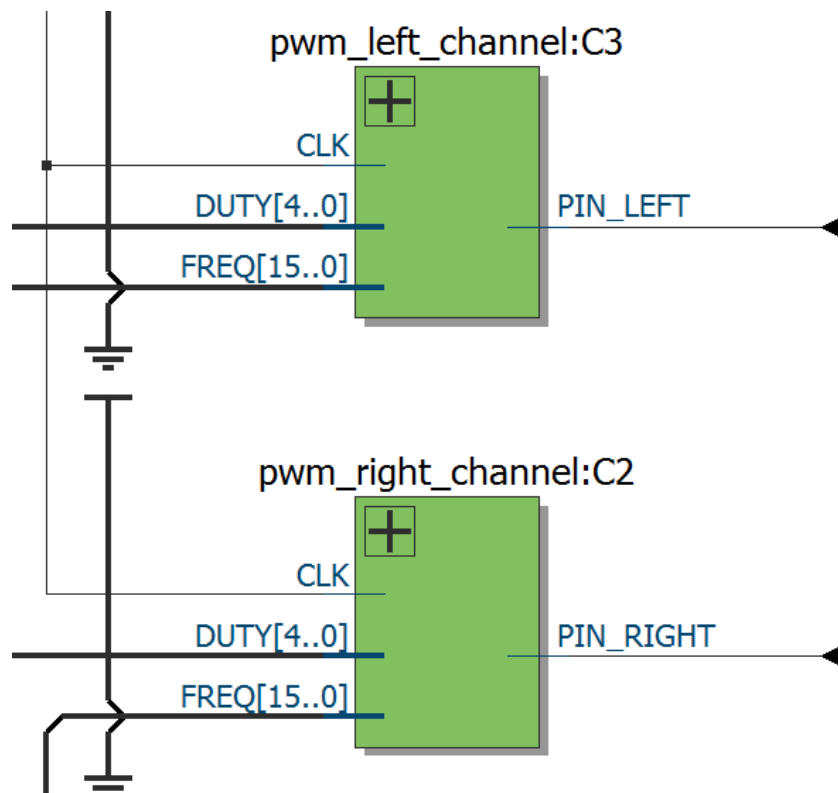
הבחירה במצב המבוקש מתבצעת לפי פסיקות המגיעות מממשק המשתמש. האותיות f, b, r, l מכתיבות את הכיוונים בהם המשתמש בוחר להניע את הרובוט.

עבור תנועה נוחה במרחב בנינו ארבע פונקציות נסיעה:

- 'f' – קדימה: ה-Servo הימני מסתובב עם כיוון השעון. ה-Servo השמאלי מסתובב נגד כיוון השעון.
- 'b' – אחורה: ה-Servo הימני מסתובב נגד כיוון השעון. ה-Servo השמאלי מסתובב עם כיוון השעון.
- 'r' – ימינה: ה-Servo השמאלי מסתובב נגד כיוון השעון. ה-Servo הימני עוצר.
- 'l' – שמאלה: ה-Servo הימני מסתובב עם כיוון השעון. ה-Servo השמאלי עוצר.

כיוון הסיבובים אמנם לא אינטואיטיבי, אך הדבר נובע משום שמנועי ה-Servo מחוברים אחד מול השני ולכן פעולתם היא כעיקרון מראה. בתנועה קדימה ה-Servo ימין מסתובב עם כיוון השעון, אך ה-Servo שמאל חייב להסתובב נגד כיוון השעון כדי שינועו יחדיו.

כדי לייצר את התנועה בנינו Component שמקבל שני פרמטרים - Duty cycle ומחלק תדר וע"י כך מייצר אות PWM אשר מועבר ל-Servo המבוקש.



איור 48 – רכיבי ה-PWM בערוצים השונים

עבדנו עם שלושה מצבי רוחב אות: 1, 1.5 ו 2 מילישניות. העבודה בפרויקט התבצעה על לוח FPGA של חברת ALTERA בשם DE0-nano בעל תדירות שעות של 50 מגה הרץ. בכדי ליצור רוחב אות מסוים, השתמשנו במחלק תדר (משתנה בשם FREQ הנשלח כפרמטר) בהתאם לתדר המבוקש. אות ה-PWM מחולק למצב בו הוא נמצא ב-'high' חלק מהזמן וב-'low' ביתר הזמן, ע"י ערך Duty cycle הנשלח כפרמטר. מכיוון שה-Duty cycle נוצר דרך PROCESS אשר פועל ע"פ שעות בתדר המשתנה FREQ ומופעל רק בעלייתו, יש להשתמש במחלק תדר (FREQ) הגדול פי 2 מהתדר הרצוי. מכאן הערכים בהם השתמשנו ליצירת מחלקי התדר הם:

$$1\text{ ms} \leq \frac{50\text{ MHz}}{2 \cdot 25000} = 1\text{ KHz}$$

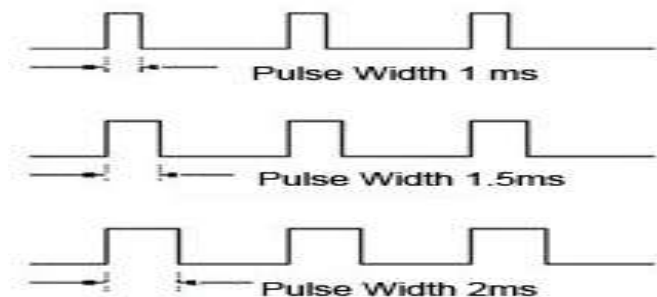
איור 49 – ערכי מחלק התדר

$$1.5\text{ ms} \leq \frac{50\text{ MHz}}{2 \cdot 37500} = \frac{2}{3}\text{ KHz}$$

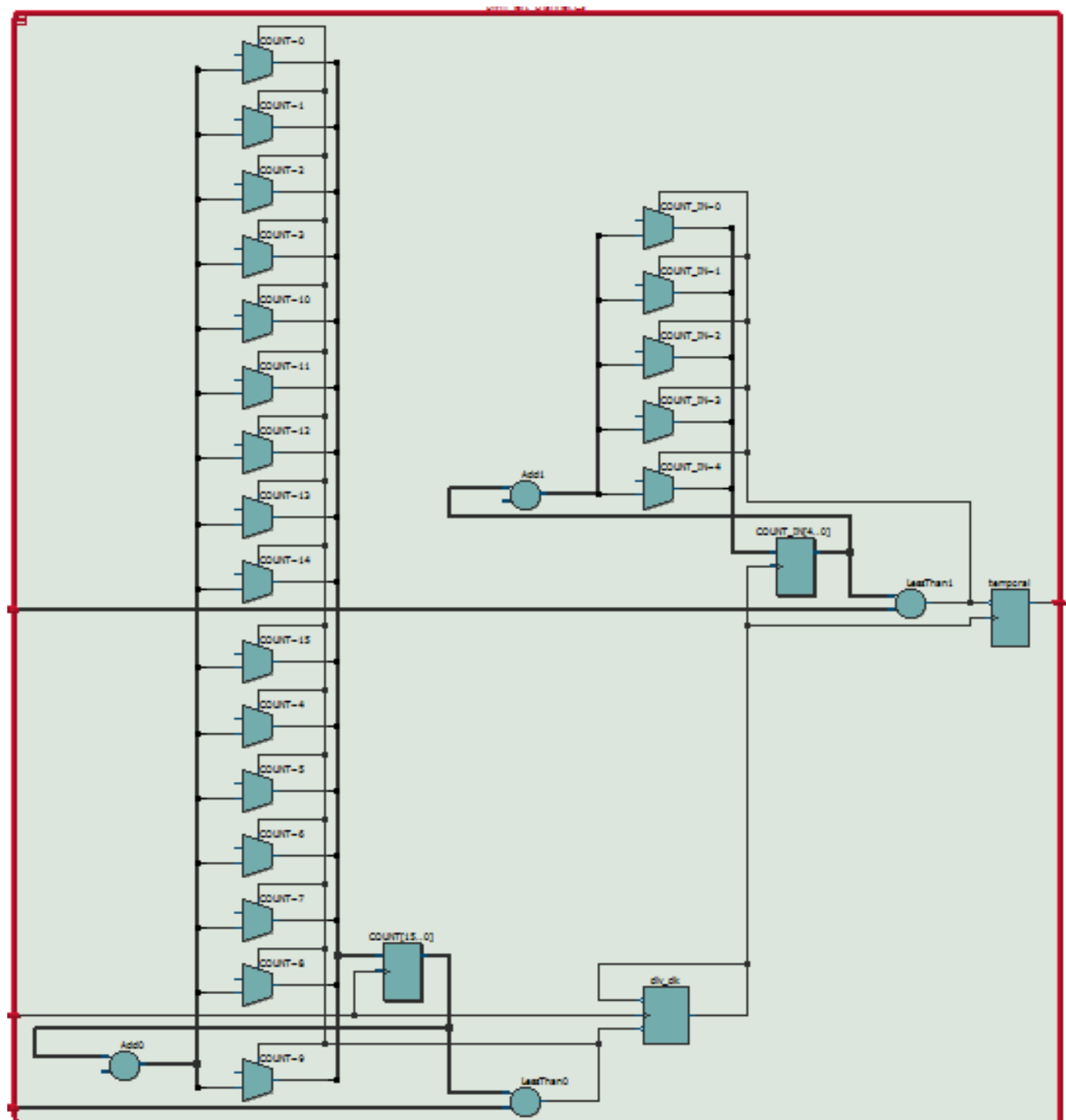
$$2\text{ ms} \leq \frac{50\text{ MHz}}{2 \cdot 50000} = 0.5\text{ KHz}$$

פרמטר ה-Duty cycle סוכם את השעות המחולק בתדר FREQ על מנת להגיע לזמן מחזור כולל של 20 מילישניות:

- עבור רוחב פולס של 1ms (סיבוב עם כיוון השעון), Duty cycle=20.
- עבור רוחב פולס של 2ms (סיבוב נגד כיוון השעון), Duty cycle=10.
- עבור רוחב פולס של 1.5ms (נקודת המנוחה), Duty cycle=13. זמן המחזור יוצא 19.5ms, אמנם יש הפרש קל מזמן המחזור הרצוי, אבל מנועי Servo "סלחנים" לשינויים עד לטווח של 10% ברוחב האות.

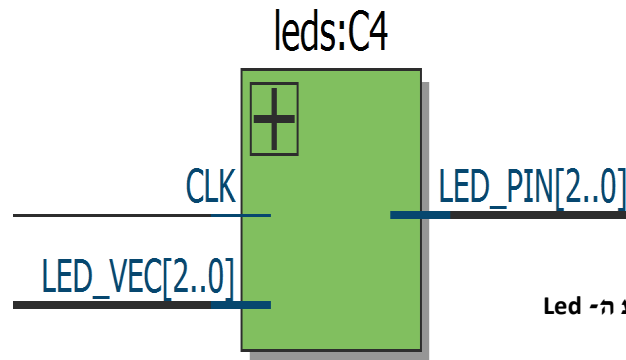


איור 50 – אותות PWM הנשלחים למנועי ה-Servo



איור 51 – סכמת RTL של רכיב ה- PWM

כפי שניתן לראות בתרשים ישנם שני מערכי סכימה, הגדול והראשי משמש ליצירת מחלק התדר FREQ והשני והקטן יותר משמש לסכימה לייצור ה- Duty cycle הרצוי.



איור 52 – רכיב הדלקת וכיבוי נוריות ה- Led

עבור נוריות ה- Led הגדרנו ווקטור (COLOR_VEC) באורך 3 סיביות שמקבל ערך שונה בכל פסיקה.

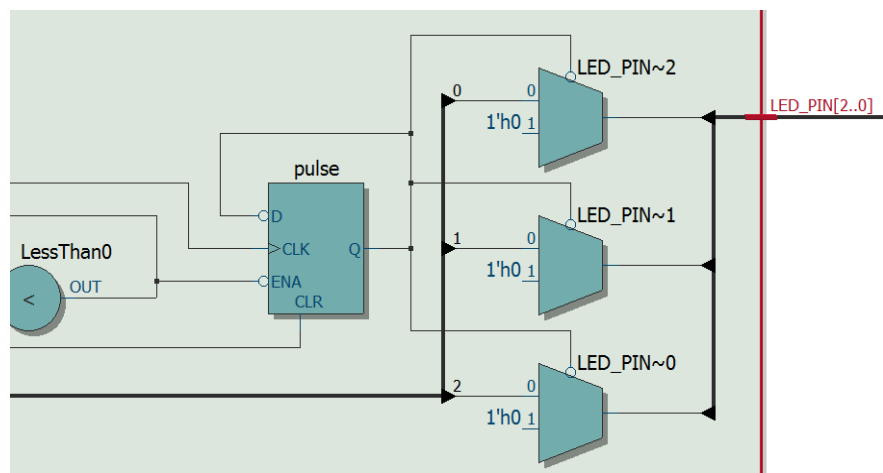


איור 53 – ווקטור הדלקת וכיבוי נוריות ה- Led

נוריות ה- Led יידלקו בהתאם לווקטורים הבאים :

- "100" – מדליק את ה- Led הירוק.
- "010" – מדליק את ה- Led האדום.
- "001" – מדליק את ה- Led הצהוב.

עבור כל פסיקה אחרת, מוצא ה- "COLOR_VEC" יהיה "000", כלומר עבור כל פקודה אחרת מלבד זיהוי עיגול באחד משלושת הצבעים אשר הגדרנו, יכובו את כל הנוריות. הדלקת הנוריות מתבצעת באופן של הבהוב. בתוך הרכיב שממפה את הנוריות, הכנסנו סוכם המייצר שעון שמדליק ומכבה את הנורית המתאימה אחת לשניה.

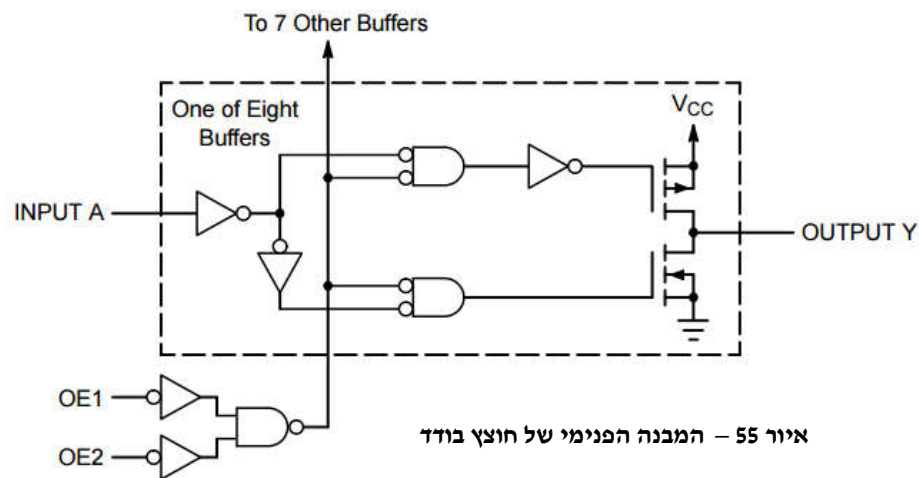


איור 54 – סכמת הרכיב ששולט בהפעלת ה- Led

3.4.3 חיבור ה-Buffer

רגלי ה I/O ברכיב מוציאות מתח של 3.3 וולט ואילו מנועי ה- Servo צורכים מתח שעולה על 4.8 וולט. הפתרון לבעיה זו הוא חוצץ מגביר מתח אותו סקרנו בפרק המבוא.

MC74HC541A – octal buffer הינו חוצץ בעל 8 כניסות ושמונה מוצאים. את כניסות ה- OE1 : Enable ו- OE2 חיברנו יחדיו אל רגל מלוח ה- FPGA אשר אנו מוציאים ממנה "מתח נמוך" (low) בכדי לאפשר את פעולת החוצץ.



חיברנו לחוצץ מקור מתח חיצוני בגובה של 5 וולט כ- V_{CC} . את אות המוצא של לוח ה- FPGA המיועד להפעלת ה- Servo, חיברנו אל מבוא החוצץ (A בתרשים) ואת המוצא (Y בתרשים) חיברנו אל ה- Servo כסיגנל שלו או כאות ה- PWM.

פרק 4 – סיכום ומסקנות

4.1 סיכום

בתחילת העבודה תכננו לייצר פרויקט שידמה משחק או סביבה משחקית ושיהיה רחב מבחינת תחומי המימוש, דבר שיקנה לנו ניסיון והיכרות בתחומים שונים אליהם רצינו להיחשף. השתדלנו שהפרויקט ייגע בטכנולוגיה הרלוונטית ביותר היום, מתוך הרצון לשלב את העולמות שנהנינו ללמוד במהלך התואר והיינו רוצים לעסוק בהם בעתיד.

בפרויקט זה רכשנו ניסיון במספר תחומים :

1. עבודה עם רכיב FPGA וכתובה ב-VHDL.
2. כתיבת ממשק משתמש והטמעת שידור וידאו בתוך הממשק.
3. עיבוד וידאו ודגימתו בזמן אמת כחלק מהתמודדות עם תהליכים הקורים בזמן אמת.
4. שימוש בתחליפי כבלים כדוגמת Bluetooth ו-WiFi.
5. שליטה במנועי Servo, כולל הקושי לייצר תנועה שונה במנועים שונים.

ברצוננו להודות למכללה ולמרצים אשר נתנו לנו את הכלים וההזדמנות לבצע את הפרויקט המעניין הזה.

4.1.1 קשיים איתם התמודדנו במהלך הפרויקט

במהלך הפרויקט התמודדנו עם לא מעט קשים , להלן רשימה חלקית :

1. הקושי הראשון היה להתמקצע בכתיבת UART שיותאם למשימה הספציפית אותה רצינו לממש, החיבור עם מודם ה-Bluetooth וצריבת הקוד לזיכרון ה-EEPROM ב-FPGA כך שלא ימחק בניתוק מהחשמל. הספר של אייל חברבר עזר לנו להבין את הפתרונות האפשריים וכך לפתור את הבעיה.
2. בחירת מצלמה : לקח לנו זמן לבחור מצלמה כי לא ידענו איך להתחבר לשידור ולמשוך אותו להמשך עיבוד ב-MATLAB. פרמטר נוסף שהנחה אותנו היה הרצון לעשות שימוש במכשירים שזמינים לאנשים רבים ואינם מוזמנים במיוחד לצורך הפרויקט.
- מבחינה כלכלית, ככל שדרישותינו היו גבוהות כך גם המצלמה עלתה יותר ולכן אנו בחרנו במצלמת הסמארטפון, מוצר נגיש שעונה על הדרישות הרלוונטיות לפרויקט.

3. בשלב כתיבת ממשק המשתמש ב-MATLAB, לקח לנו זמן וניסויי וטעייה רבים על מנת לחבר את הוידאו המגיע מהמצלמה לתוך ה-GUI, ליצור את הפקדים השונים ולייצר ממשק עמו ניתן לנווט את הרובוט במרחב. נעזרנו בעיקר בשלושת המקורות הבאים בכדי להתגבר על הבעיות הללו:
 - HELP - פונקצית העזרה האוטומטית הקיימת ב-MATLAB.
 - <http://www.mathworks.com> – אתר הבית של יצרני MATLAB. שימושי ביותר, מפורט ומלא בדוגמאות.
 - פורומים ובלוגי חובבים שונים- הרשת מלאה בהתנסויות של אנשים מהם ניתן להחכים וללמוד.
4. כיצד לעבד וידאו בכלי עיבוד תמונה: בפרויקט עיבדנו בכל פעם תמונה אחת מהוידאו. בתחילה הדבר הוביל לקריסות מרובות היות וטרם גמרנו עיבוד של תמונה אחת וכבר אנחנו דוגמים תמונה נוספת. על מנת להתגבר על קושי זה, ייעלנו את משך הליך הבדיקה והעיבוד של כל תמונה, הקטנו את גודל התמונה המעובדת ולבסוף גם הוספנו 'פאזזה' מאולצת בכדי לאפשר עבודה רציפה.
5. זיהוי אובייקטים מרובים: באלגוריתם שכתבנו בתחילה, לא שמנו דגש על מצב בו בתמונה ישנם מספר אובייקטים. במקרה זה הפיתרון היה לאפשר זיהוי של אובייקט אחד בלבד בכל ריצה.
6. בעיות צל: בזיהוי צבע יש בעיה של הסטת ערכי ה-RGB בכל שינוי של מידת החשיפה לאור. הפתרון שמצאנו היה לאפשר קשת רחבה של ערכים לייצוג כל צבע.

4.1.2 הצעות להמשך הפרויקט

- משום שבפרויקט עבדנו במישורים רבים, ראשית ניתן לקחת כל אלמנט בנפרד ולשכלל אותו לרמות דיוק וביצועיות טובות יותר, ובהקשר הרחב יותר של הפרויקט בשלמותו:
1. ניתן לשנות את התוכנה בה השתמשנו לשידור הוידאו למחשב, לתוכנה שתשתמש ביכולות נוספות של הטלפון, בין אם זה הקלטת סאונד או חיישני גירוסקופ. כך יהיה ניתן לקבל מידע רחב יותר על המתרחש בסביבת הרובוט.
 2. ניתן לשנות, להוסיף ולהרחיב את יכולות הזיהוי של האלגוריתם שבנינו לצבעים נוספים ולצורות גיאומטריות נוספות.
 3. ניתן להוסיף חיישנים חיצוניים לרובוט כגון חיישן טמפרטורה או חיישן אור וכך להחזיר מידע אודות הסביבה בה הרובוט נמצא דרך ה-UART, כך שנוכל לקבל מידע בצינור נוסף מלבד המצלמה, במקרה של תקלה ברשת ה-WiFi.

1. <https://learn.sparkfun.com/tutorials/using-the-bluesmirf/all>
2. <http://www.idogendel.com/whitebyte>
3. <http://www.microchip.com/wwwproducts/Devices.aspx?dDocName=en558330>
4. <http://www.mathworks.com>
5. <https://www.wikipedia.org>
6. <http://www.bluetooth.com/Pages/Bluetooth-Home.aspx>
7. <http://www.lirtex.com/he>
8. <https://www.pololu.com/product/1248>
9. <https://github.com>
10. Terasic DE0-Nano User Manual.

**הצעה לפרויקט הנדסי – מכון טכנולוגי חולון- H.I.T.
הפקולטה להנדסת חשמל, אלקטרוניקה ותקשורת
(50076)**

שנה

|| 2 | 0 | 1 | 5 ||

15.04.15 : תאריך הגשה ראשונית
: תאריך אישור ההצעה
01.01.16 : תאריך סיום משוער

א. פרטי הסטודנט/ים:

2	שם מלא	אדיר ציון
	תעודת זהות	38084455
	טלפון	050-9944497
	דואר אלקטרוני	adirdido@hotmail.com



חתימת הסטודנט
אדיר ציון

1	שם מלא	שלומי רחימי
	תעודת זהות	301287223
	טלפון	052-4622935
	דואר אלקטרוני	Shlomir25@gmail.com



חתימת הסטודנט
שלומי רחימי

ב. שם הפרויקט

שם הפרויקט בעברית: (עד 8 מילים):

יישום תוצרי ראייה ממוחשבת באמצעות רובוט הנשלט ע"י FPGA.

Project Name: (Up to 8 words):

Implementation of computer vision products using a robot controlled by FPGA.

ג. שמות המנחים:

מנחה אקדמי: ד"ר טירר נוגל חתימה: _____

מנחה נוסף: _____ חתימה: _____

ד. רקע ורציונאל לפרויקט

על הרקע והרציונאל להכיל רקע כללי, מוטיבציה ורעיונות בהיקף של **לפחות 10 שורות**. יש להתייחס לשיקולי שיווק ותיאור הצורך בהתאם ליישומיים השונים (שימו לב! רקע אינו מטרת הפרויקט ואינו תיאור הפרויקט!)

רקע כללי- (מקסימום 5 שורות)

במהלך השנים האחרונות ניתן לראות את השימוש היומיומי ההולך וגובר ברובוטים כתחליף לפעולות אנושיות. דוגמא לכך הוא ה- IROBOT (רובוט לניקוי הבית) והרכב הרובוטי של גוגל (שנמצא בתהליכי רישוי).

רציונאל- מוטיבציה לפרויקט- מקסימום 5 שורות

המוטיבציה לפרויקט היא לתת מענה מכני בעזרת רובוט למקרים ומצבים פיזיים שונים. כך לדוגמא, התמצאות הרובוט במרחב בהתאם לאילוצים שמוצגים בפניו.

ה. מטרת הפרויקט / היקף הפרויקט

מטרה עיקרית

עיבוד תמונות המכילות צורות שונות בצבעים שונים ומתן פקודות לרובוט לביצוע

משימות בהתאם למידע שיופק מהתמונות.

ו. תוצרי הפרויקט: (מדדים כמותיים להצלחת הפרויקט). יש לוודא קורלציה בין מטרת הפרויקט לתוצרים

לגבי סימולציה לפרט מה יוצג כתוצר סופי ומה יודגם כמו כן לציין מהי הפלטפורמה לסימולציה ואם קיימת. לגבי בנייה לפרט מה המעגל/מערכת שיבנה ומה יודגם.

1. הפקת מידע בזמן אמת מתמונות שעברו עיבוד דיגיטלי.

2. שליטה וסנכרון בין החלקים השונים בשרשרת הביצוע (עיבוד, שליחת המידע ותפעול הרובוט)

ז. מקורות

יש לצרף מקור אחד עיקרי שעליו מתבססת העבודה. בציון המקורות הנוספים יש לכתוב: שם מחבר, **שם ספר/מאמר ממנו נלקח המקור, הוצאה, שנה ועמוד**

1. <http://www.mathworks.com>
2. <https://learn.sparkfun.com/tutorials/using-the-bluesmirf/all>
3. Terasic DE0-Nano User Manual

ח. פרט את רשימת הציוד ורכיבים ייחודיים הנדרשים לשם ביצוע הפרויקט ואת אופן השגת הציוד/רכיבים

	הציוד/רכיב	אופן השגתו
1.	Bluetooth Modem	ע"י המכון
2.	DE0-Nano board	ע"י המכון
3.	WIFI	מצלמת WIFI, או בשימוש במצלמת הסמארטפון האישי
4.	חוטים, מחברים וציוד אלקטרוני פשוט	ציוד שקיים ברשותנו והיתר ע"י המכון

ט. גאנט פעילות בפרויקט

פעילות ואבני דרך בביצוע הפרויקט. על הסטודנט למלא טבלת הגאנט

פעילות/חודש	חודש 1	חודש 2	חודש 3	חודש 4	חודש 5	חודש 6	חודש 7	חודש 8
חקר הנושא וסביבת העבודה	X	X						
תכנון המעגל החשמלי על כל רכיביו		X	X					
בחירה של רכיבים מתאימים והזמנתם	X							
פיתוח אלגוריתם (במידת הצורך)			X	X				
דו"ח ביניים				X				
כתיבת תוכנה (במידה וקיימת)			X	X	X			
סימולציה תוכנה/אלגוריתם				X	X			
בניית המעגל החשמלי (במידה וקיים) ואינטגרציה בין רכיביו				X	X	X		
אינטגרציה בין התוכנה ורכיבי החומרה					X	X		
בדיקת תפקוד ותקינות המערכת						X	X	
כתיבת ספר פרויקט						X	X	X

י. סיכונים

1. סיכונים בהשגת מטרות הפרויקט (אם קיימים)

קושי בביצוע עיבוד תמונה מדויק והפקת מידע רלוונטי ממנה.

2. כיצד תתמודד (הסטודנט) עם סיכונים אלו

פישוט מנגנון עיבוד התמונה.

פרויקט מאושר	פרויקט מאושר בתנאי	פרויקט לא מאושר
הערות	הערות	הערות

שם וחתימת רכז פרויקטים
