מקיף ח' אשדוד | סמל מוסד:644484 |שנה"ל תשפ"ה 2024-25

***פרויקט גמר 5 יחידות במגמת אלקטרוניקה ומחשבים***

A black and white machine on a wooden surface

Description automatically generated

***רכב לנסיעה אוטונומית וזיהוי אובייקטים בדרך בעזרת מצלמה***

מגיש: רועי יהודה אלבז | מנחה: יוסי חזן | סמל שאלון 841589

תעודת זהות: 216715441

**תוכן עניינים**

* תודות --------------------------------------------------------------------------- עמ' 3

**תודות**

בפרוייקט זה, אני מסכם את מסלול הלימוד שלי במגמת מסמ"ת. במהלך 3 השנים האלו, התנסיתי במגוון תחומים בעולם האלקטרוניקה והמחשבים עם מיטב המורים שיש לבית הספר להציע.

ברצוני להוקיר תודה מקרב לב לכל מורי המגמה: יוסי חזן, טל מקדש ומאיר בוסקילה, אשר תמיד היו שם בשבילנו, עזרו ותמכו, והקנו לי ידע נרחב ועשיר בתחום שאני כל-כך אוהב. בנוסף, אני רוצה להקדיש תודה לרכזת המגמה, אפרת דיל, שתמיד דאגה שיהיה לנו את מה שצריך בשביל להצליח, על הדאגה, המקצועיות והאכפתיות כלפי התלמידים.

תודה מיוחדת למנחה שלי בפרוייקט, יוסי חזן, על הסבלנות, המקצועיות וההזדמנות לעבוד על פרוייקט כזה מעניין. תודה על ששיתפת אותנו בניסיונך הרב וקידמת אותנו בצורה מקצועית ביותר.

בס"ד

**הצעת פרוייקט**

***טופס 1101 - בקשה לאישור נושא עבודת גמר ופרוייקט גמר בחטיבה העליונה***

***במגמת הנדסת אלקטרוניקה ומחשבים***

**בית-הספר:**  
 סמל מוסד: 644484 שם ביה"ס: מקיף רחוב: ברק בן אבינועם מס' 10

ת.ד:5232 ישוב: אשדוד מיקוד: 5232 טל.088656917 פקס: 088641174

**שם וסמל השאלון:**  
 עבודת גמר (5 יח"ל) שם השאלון: \_\_\_מערכות מחשבים\_\_\_ סמל השאלון: 833599  
 פרויקט גמר (3 יח"ל) שם השאלון: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ סמל השאלון: \_ \_ \_ \_ \_ \_  
לצורך פשטות הניסוח נשתמש להלן במונח **פרויקט** הן לעבודת גמר והן לפרויקט גמר

**נושא הפרויקט**:  
תכנון רכב לנסיעה אוטונומית וזיהוי אובייקטים בדרך בעזרת מצלמה.

**תרשים מלבנים מצורף בנספח א'** (חובה לצרף)

**תרשים חשמלי מצורף בנספח ב'** (חובה לצרף)

**תיאור מפורט של נושא הפרויקט:**

בפרוייקט זה נתכנן רכב אוטונומי בסביבת עבודה של הבקר esp-32 לעקיבה אחרי פס שחור על גבי משטח לבן. כמו-כן נתכנת זיהוי אובייקטים לדוגמא: מצב רמזור, תמרורים וכו׳ ע"י שימוש ברכיב ה – ESP32-CAM. בהתאם לאובייקטים אותם מזהה הבקר מהמצלמה, יישלחו פקודות מתאימות למנועים לצורך התגברות על המכשולים המוצבים לרכב.

תיאור הבעיה או הצורך:

בימינו, המכוניות האוטונומיות משתלטות על העולם. בערב, טכנולוגיות לזיהוי מכשולים היו פרימיטיביות, אך כיום, מכוניות אוטונומיות כוללות בתוכן טכנולוגיות רבות, כגון נסיעה אוטונומית וזיהוי מכשולים בדרך. ישנן מספר גישות לזיהוי המכשולים בדרך, כגון חיישני מרחק, חיישני קרבה, מצלמות לצורך בקרת נסיעה ועיבוד ויזואלי של המכשולים או שילוב הגישות.

הצורך בטכנולוגיות שכאלו נובע מכמה סיבות מרכזיות:

1. נוחות למשתמש – בנסיעה אוטונומית, המשתמש לא צריך לנהוג בעצמו, מה שיכול לפנות אותו לביצוע פעולות אחרות בזמן הנהיגה.
2. בטיחות – בעזרת זיהוי המכשולים המפותח שיש ברכבים אוטונומיים, הם מספקים למשתמש בטיחות מוגברת, שכן הם מזהים בעצמם סכנות ומכשולים ולא מסתמכים על המשתמש להגבה נכונה ומהירה לסיטואציות.

המערכת נועדה לספק פתרון של עיבוד ויזואלי של מכשולים בדרך לרכבים אוטונומיים. המערכת שלנו, בשונה ממערכות אחרות, יכולה להציג למשתמש באיזה מכשול נתקל הרכב, על מסך OLED.

תפקיד הפרויקט:

תפקיד הפרויקט הוא לספק רכב אוטונומי, אשר יוכל להתגבר על המכשולים המוצגים לו בדרכו תוך מעקב אחר פס והצגת מידע מהמערכות למשתמש.

מבנה המערכת ואופן הפעולה:

כל הרכיבים מחוברים ומתקשרים עם בקר ה – ESP32. אל הבקר תחובר גם מצלמה מסוג ESP32-CAM, ותעביר מידע לגבי המכשולים בדרכו של הרכב לבקר. המצלמה מונחת על מנוע סרוו לצורך הזזתה על צירה וקליטת מידע מסביבה רחבה יותר. אל הבקר מחובר גם מסך OLED, אשר יציג למשתמש איזה אובייקטים נמצאים למול עדשת המצלמה, אליהם יגיב הרכב. אל הרכב מחובר גם חיישן פס, אשר ישמש לצורך נסיעה אוטומוטית ומעקב אחריו.

לבקר מחוברות נורות NeoPixel אשר ישמשו לצורך איתות של הרכב.

**המפרט הטכני של הפרויקט:**

1. מיקרו בקר – ESP32

2. מצלמה – ESP32-CAM אשר תשמש לזיהוי תמרורים כגון: מצב הרמזור  
3. מנוע סרוו (Servo) להטיית המצלמה

4. חיישני פס לצורך מעקב אחרי הפס השחור

5. 4 מנועי DC לכיוון נסיעת הרכב

6. מסך OLED לצורך הצגת מידע למשתמש

7. נורות NeoPixel אשר ישמשו לצורך איתות, תאורה וחיווי עצירה.

8. חיישן מרחק אולטרה סוני **כאופציה** לזיהוי מכשולים

**פרטי התלמיד והמשימות אותן יבצע:** (יש למלא סעיף זה לפרויקט שיבוצע על-ידי תלמיד יחיד. במידה ויותר מתלמיד אחד יבצע פרויקט נוסף בעל המאפיינים שפורטו בסעיפים 3-7 יש לשכפל את עמוד 2 ולמלא סעיף 8 עבור כל תלמיד.) **א. התלמיד**: שם פרטי: רועי יהודה  **שם משפחה**: אלבז  **מס' זהות 216715441**:  
פרוט המשימות אותן יבצע: (ניתן לצרף נספח)  
  
  
1. לימוד נושאי העבודה ושפת תוכנה עבור הבקר ESP32

2. לימוד נושאי העבודה וסביבת העבודה של המצלמה ESP-32-CAM  
3. תכנון ובניית חומרת המערכת  
4. תכנון המערכת והבנתה

5. בניית מערכת על כל מרכיביה  
6. ניסויים מדידות ואיתור תקלות  
7. הכרת האמצעים לפיתוח מעגל המשלב תוכנה וחומרה

8. כתיבת תוכנית בשפת C לבקר ESP32 בסביבת עבודה Arduino IDE

9. כתיבת ספר פרוייקט  
  
  
חתימת התלמיד: רועי יהודה אלבז  
  
תאריך: 29/10/2024

**ביבליוגרפיה:**א. תכנות ESP32 בסביבת Arduino IDE - ספרו של שי מלול

ב. דפי נתונים מהיצרן על הרכיבים השונים במעגל

**צוות ההוראה בבית-הספר המטפל בבקשה:  
המנחה:** שם פרטי: \_יוסי שם משפחה: \_חזן\_\_ טל. בבית: \_\_\_\_\_\_\_\_\_\_\_\_\_ טל נייד: \_\_\_0507930920\_\_

דוא"ל: yossi.mitsu@gmail.com

השכלה:\_ \_מהנדס אלקטרוניקה ומחשבים\_\_ תחום השכלה: \_\_\_\_\_\_\_\_\_\_\_\_\_\_ חתימה: \_\_יוסי חזן\_\_  
**מרכז המגמה:** שם פרטי: אפרת\_ שם משפחה: \_\_דיל\_\_ טל. בבית: \_\_\_\_\_\_\_\_\_\_\_\_\_ טל נייד:  0524232199

דוא"ל: [efrat1705@walla.com](mailto:efrat1705@walla.com)

חתימה: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
**מנהל ביה"ס**: שם פרטי: אווה שם משפחה: טבת חתימה: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
תאריך:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ חותמת ביה"ס: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

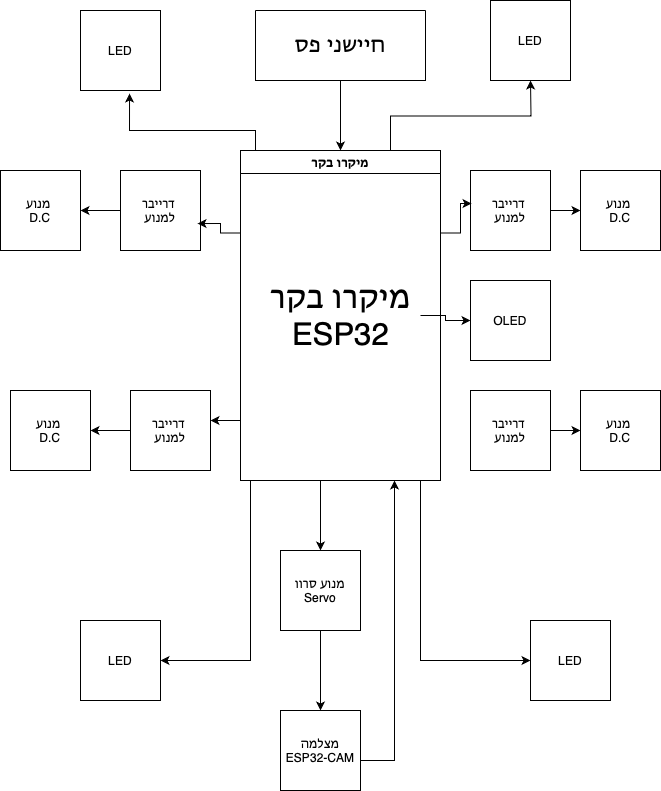
**מבוא**

אנו חיים בימים אשר בהם הטכנולוגיה משנה את חיינו בכל התחומים. ההתקדמות הטכנולוגית מקלה על חיינו והופכת אותם לקלים ונוחים יותר. דוגמה לחידוש טכנולוגי שכזה הוא מכונית אוטנומית. מכונית שכזו הינה כלי רכב אשר מקבל מידע מחיישנים שונים ונוסע במסלול קבוע מראש. על המכונית האוטונומית להתגבר על המכשולים המוצבים לה בדרכה בכדי לעבור את מסלולה בצורה מיטבית.

בפרוייקט זה נתכנן רכב אוטונומי בסביבת עבודה של הבקר ESP-32 לעקיבה אחרי פס שחור על גבי משטח לבן. כמו-כן נתכנת זיהוי אובייקטים לדוגמא: מצב רמזור, תמרורים וכו׳ ע"י שימוש ברכיב ה – .ESP32-CAM בהתאם לאובייקטים אותם מזהה הבקר מהמצלמה, יישלחו פקודות מתאימות למנועים לצורך התגברות על המכשולים המוצבים לרכב.

בפרוייקט זה נשלב תוכנה וחומרה, הרכב האוטונומי מרכב מבקר ESP32, אשר אליו מחוברים חיישנים נוספים כגון חיישן פס ומצלמה מסוג ESP32-CAM, מסך OLED לצורך הצגת מידע מהחיישנים ו- 4 מנועים לצורך הסעת הרכב.

**תרשים מלבנים**



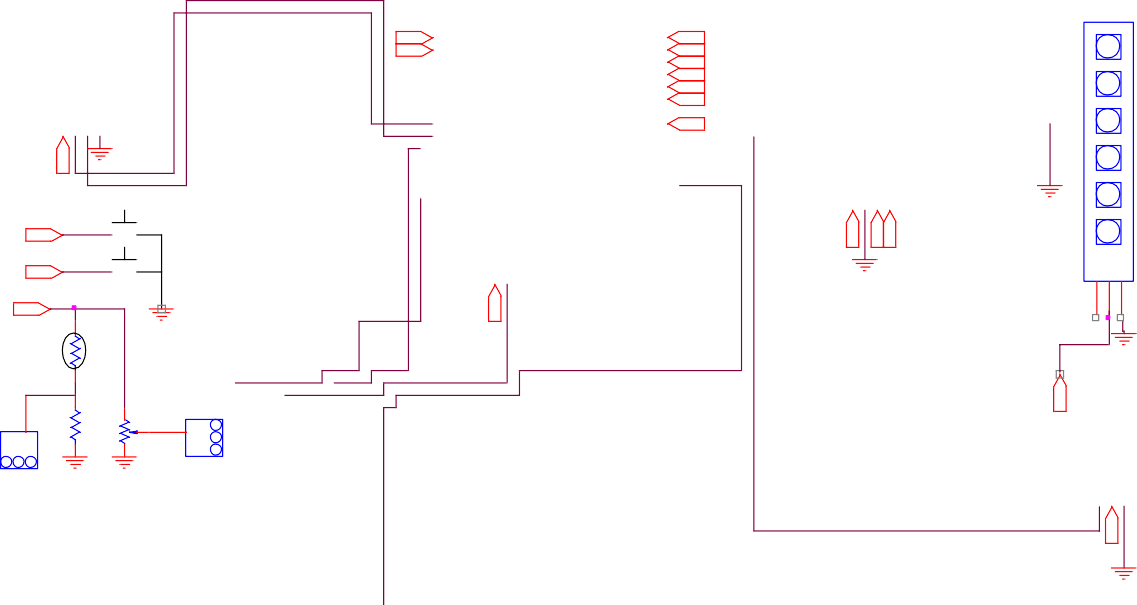
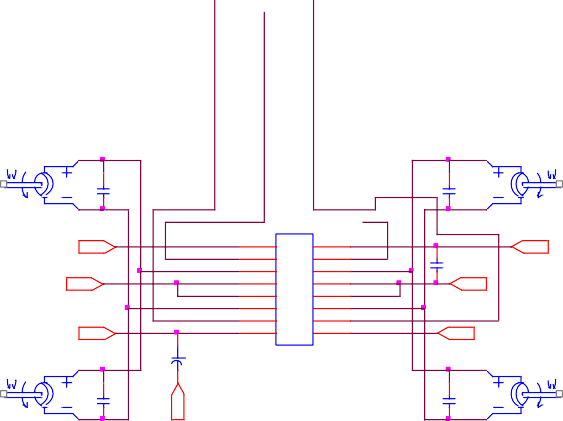
**הסבר תרשים המלבנים:**

בתרשים המלבנים ניתן לראות רכיבים שונים, אשר מחוברים כולם לבקר ה – ESP32.

לכל רכיב בפרוייקט מטרה משלו:

* בקר ה – ESP32 – משמש לצורך שליטה כללית במערכת, אחראי על עיבוד הנתונים, חיבור לאינטרנט, שליחת פקודות לרכיבים השונים וביצוע משימות.
* חיישני פס – מטרתם היא לעקוב אחרי פס שחור, על גבי משטח לבן. בעזרתם ידע הרכב לאן לנסוע.
* מנועי DC: אחראיים על התזוזה של המערכת. מנועים אלו יניעו את גלגלי המערכת בכיוון הדרוש לצורך מעקב של הרכב האוטונומי אחרי הפס.
* מצלמה ESP32-CAM – בעזרת המצלמה ושילוב של ניתוח תמונה באמצעות בינה מלאכותית, ידע הבקר להתגבר על המכשולים המוצבים בפניו. המצלמה תורכב על מנוע SERVO, אשר יאפשר לסובב את המצלמה על צירה ולקלוט מידע מסביבה רחבה יותר.
* מנוע SERVO: המנוע משמש לסיבוב של המצלמה על צירה, כדי שתוכל לראות מכשולים בסביבתה ולשלוח פקודות לבקר לצורך התגברות עליהם.
* OLED – מסך OLED אשר ישמש לצורך הצגת מידע מהחיישנים על גביו.
* מנורות LED - ישמשו לצורך סימוני איתות בעבור הרכב.

שרטוט חשמלי של המערכת



U11

HC-SR04

GND

extrn v cc

U8

OLED display

1

esp32-WROOM

U9

RGB LED

U16

5 led stripe

0

extrn v cc

1. GND
2. 3V3
3. EN
4. GPIO36
5. GPIO39
6. GPIO34
7. GPIO35
8. GPIO32
9. GPIO33
10. GPIO25
11. GPIO26
12. GPIO27
13. GPIO14 GPIO12

38

GND 37

GPIO23 36

GPIO22 35

GPIO1 34

GPIO3 33

GPIO21 32

NC 31

GPIO19 30

GPIO18 29

GPIO5 28

GPIO17 27

GPIO16 26

GPIO4 25 GPIO0

GND

key 2 scl uart0 tx uart0 rx sda

key 1

uart1 tx uart1 rx

0

key 1

key 2

scl sda

extrn v cc

0

extrn v cc

0

GND

U10

serv o

0

R4 1K

extrn v cc

in

0

0

U4

U2

extrn v cc

C4

1n

C1

1n

0

DCMOTOR

U6 L298D

5V

1 16

1 16

2 15

DCMOTOR

5V

GND

12

VIN\_RAW

8 7 10 9

8 9

5V

U5

C6

1n

U3

C3

1n

C2

1n

DCMOTOR DCMOTOR

הסבר השרטוט החשמלי

בשרטוט זה ניתן לראות את כל החיבורים לבקר ה – ESP32:

* מסך OLED – לא מחובר להדקים ספציפיים, אלא להדקי SCL ו- SDA, אשר באמצעותם מתבצעת תקשורת I2C.
* מנועי DC: בחרתי במנועים אלו מכיוון שהם מאפשרים פעולה של המנוע שלא לפי זווית, אלא בהינתן זרם למנוע, הוא ינוע בכיוון מסוים, עד שיפסיק לקבל זרם. מנועי ה DC מחוברים בעזרת רכיב ה – D298L. רכיב זה הינו דרייבר אשר מאפשר תקשורת בין הבקר למנועי ה DC.
* מנוע סרוו – בחרתי ברכיב זה בכדי שתהיה אפשרות למצלמה לקלוט מידע גם מסביבתה ולא רק מדברים שנמצאים בדיוק מולה.
* חיישן LDR – לא מחובר ישירות לבקר עצמו, אך מחובר באופן חיצוני ומוכן מראש לאדמה ולמקור מתח. מתחבר להדק דיגיטלי מספר 34 של הבקר.
* חיישני פס – בהם בחרתי בכדי לאפשר מעקב אחרי פס שחור.
* טבעת לדים – תאפשר חיווי של כיוון הפנייה ואיתות.

**הסבר על הבקר:**

**32ESP**

הבקר איתו אנו עובדים הוא ESP32 הפותח ע"י חברת Espressif Systems. הבקר הינו בעל 2 ליבות ורב עוצמה ביחס לאחרים בתחום המיקרו בקרים. הבקר תומך בפרוטוקולי תקשורת אלחוטית כמו WiFi ו Bluetooth ללא צורך ברכיבים חיצוניים נוספים. בנוסף, קיימים על הבקר רכיבי חומרה נוספים כגון ממירי ADC (Analog to Digital Converter). הבקר הינו חסכוני באנרגיה ויעיל ביחס לאחרים.

כדי לתכנת את הבקר לבצע את הפעולות שאנחנו רוצים שהוא יבצע, יש לצרוב אליו קוד אשר כתוב במגוון שפות כגון C או MicroPython למשל. הקוד נצרב לבקר באמצעות מספר כלים וביניהם: Arduino IDE, ESP-IDF ו- MicroPython. בעזרת הקוד ניתן לקבל מידע בפורמטים שונים (אנאלוגי/דיגיטלי) מהכניסות של הבקר, ולשלוח אותות לרכיבים (גל ריבועי PWM למשל).

הקוד נשמר בזיכרון ה Flash של המעבד והוא מבוצע בצורה אוטומטית כאשר המעבד מופעל באמצעות קריאת הקוד מזיכרון ה – Flash.

ה 32ESP משומש כיום במקומות רבים וביניהם:

* מכשירי IoT (לדוגמה: שלטים לנורות LED) – בפרוייקט זה, תקשורת אלחוטית בין רכיב המצלמה לבקר.
* שימוש תעשייתי (למטרות אוטומציה תעשייתית ובקרת מכשור)
* פריטי לבוש (כגון שעונים חכמים)
* ועוד

נתונים:

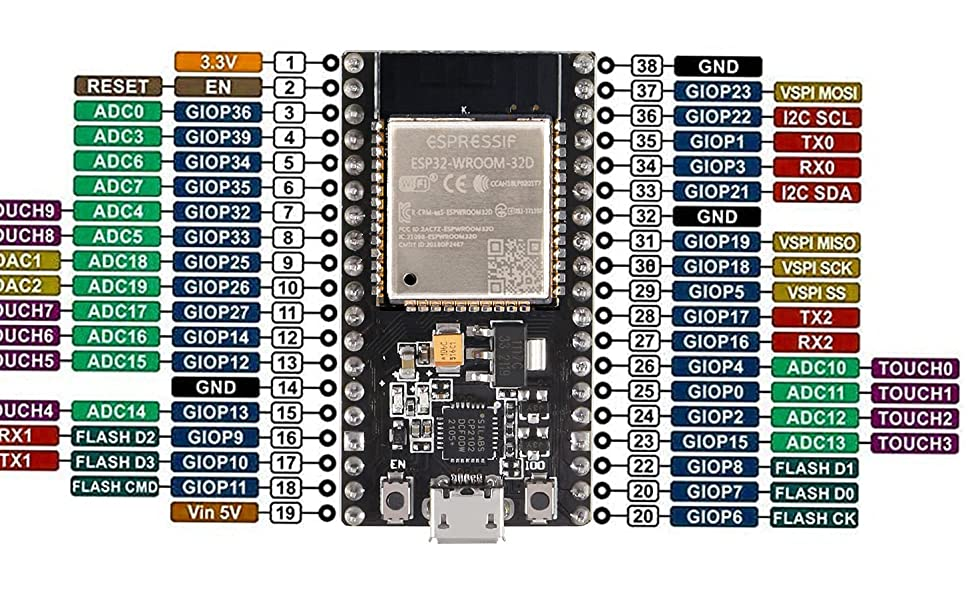
מעבד:

* בקר ה – ESP32 מבוסס על מעבד Xtensa® Dual-Core 32-bit LX6, עם תדר שעון של עד כ – 240 MHz. כל ליבה היא בת 32 סיביות.
* המעבד יכול לפעול בביצועים של עד כ- 600 DIMPS.

זכרון:

ה – ESP32 כולל בתוכו זיכרונות פנימיים מסוגי SRAM ו- ROM. בתוך זיכרון ה – ROM, מאוכסנות מערכת ההפעלה ותוכניות שצרב המשתמש. זיכרון ה – RAM משמש לאכסון נתונים זמניים.

* קיבולת זכרון ה – SRAM – 520KB
* קיבולת זכרון ה – ROM – 448KB



ממשקים:

בקר ה – ESP32 כולל מגוון של ממשקים בעבור המשתמש, לצורכים שונים:

* 36 יציאות וכניסות כלליות (GPIO) הניתנות לתכנות – דרך יציאות וכניסות אלה, ניתן לכתוב לרכיבים המחוברים לבקר, ולקבל מהן מידע (מכאן השם - general-purpose input/output). לחלק מפינים אלה יש תכונות נוספות אשר יכול המשתמש לנצל בשביל מטרותיו.
* תמיכה ב PWM – בעזרת PWM, ניתן לשנות עוצמה בה פועל רכיב מסויים. שינוי עוצמה זה נעשה בעזרת שליחת גל ריבועי ב Duty Cycle שקובע המפתח, מה שמאפשר לרכיבים שונים (כמו מנועים) לפעול במהירות אותה רוצה המשתמש, ולא בהכרח במהירות הגבוהה ביותר בלבד.

* 2 ממירי ADC של 12 סיביות, עם עד 18 ערוצים -ממיר אותות דיגיטליים לאנאלוגיים. מאפשרים למשתמש לקרוא נתונים אנאלוגיים (כמו טמפרטורה ולחות). את הנתונים אפשר לקרוא מ 18 פינים שונים.
* 2 ממירי DAC של 8 סיביות – ממיר אותות דיגיטליים לאנאלוגיים.
* עד 3 ממשקי UART – מאפשרים למשתמש לדבר לרכיבים אליהם מחובר הבקר, ולרכיבים בחזרה לבקר באמצעות תקשורת טורית אסינכרונית. ל – UART אין הדקי שעון טורי ועל כן הינו אסינכרוני (אין סנכרון בתקשורת. הבקר יודע מתי מתחיל אות ומתי הוא מסתיים בעזרת סיביות שמטרתן להודיע לו על סיום/ תחילת שידור אות).
* עד 4 ממשקי SPI – פרוטוקול תקשורת סינכרונית המאפשר תקשורת בין רכיבים במערכת. מהירותו גבוהה יותר ביחס לאופציות אחרות הזמינות בבקר. עובד בעקרון של –

Master – Slave.

* עד 2 ממשקי I2C - מאפשר תקשורת סינכרונית (בעזרת הדק SCL – שעון סריאלי), בין התקנים במערכת. בכדי לדבר להתקנים המשתמשים בפרוטוקול תקשורת זה, חשוב לדעת מהי הכתובת שלהם, ופחות משנה לאיזה פין הם מחוברים (אם כי חשוב שיחוברו לפין אשר תומך בתקשורת I2C. במקרה שלנו אלו פינים 22 ו – 21).
* ממשק I2S (ליישומי אודיו) – פרוטוקול תקשורת סינכרוני המאפשר למשתמש לתקשר עם רכיבי שמע ולהעביר נתוני שמע.
* תמיכה ב - CAN BUSפרוטוקול תקשורת המאפשר לרכיבים שונים במערכת לתקשר אחד עם השני, במהירויות גבוהות יחסית.

ממשקי תקשורת אלחוטיים:

* WiFi – הבקר תומך בקישרויות WiFi 2.4 GHz בתקן 802.11 b/g/n, בקצב העברת נתונים של עד 150 Mbps ובמצבי Station, SoftAP, Wi-Fi Direct ו-P2P.
* Bluetooth – הבקר תומך בקישוריות Bluetooth 4.2 ו- BLE (Bluetooth Low Energy).

צריכת זרם:

הבקר מאוד יעיל בצריכת הזרם שלו. הוא צורך כ – 100mA בפעילות רגילה. כאשר פעולתו מוגברת, הוא צורך כ – 240mA.

מתח פעולה:

הבקר שלנו פועל במתח של .3.3V

תחום המתחים בהם יכול לפעול הבקר: 3.0V – 3.6V

טמפ׳ עבודה:

40°C - 85°C

שיקולים בבחירת הרכיב:

בחרתי ברכיב ה – ESP32 מכיוון שהוא בקר רב עוצמה ובעל אפשרות לחיבור WiFi ו Bluetooth מובנות לצורך תקשורת אלחוטית עם רכיבים אחרים. הבקר מאפשר לי לחבר אליו הרבה רכיבים גם בחיבור חוטי, ובסופו של דבר ליצור את התוצר הסופי: רכב אוטונומי.

**דו״ח ניסוי -רועי אלבז**

**ניסוי מס׳ 1:**

שם הניסוי: הדלקת נורת RGB בסביבת עבודה של ה – 32ESP ע"י 2 כפתורים

מטרת הניסוי: התנסות בעבודה עם כפתורים בסביבת העבודה של ה – ESP32 לצורך הדלקה של נורת ה RGB בצבע יחיד, בהתאם לכפתור שנלחץ. כל כפתור ידליק את הנורה בצבע אחר.

מהלך הניסוי:

רשימת ציוד:

* אתר סימולציות Wokwi
* בקר ESP32
* 2 כפתורים
* נורת RGB

בניסוי זה, נעזרתי באתר הסימולציות Wokwi. חיברתי לבקר 2 כפתורים, אחד בצבע ירוק ואחד בצבע אדום, אשר כל אחד ידליק את הצבע שלו בנורה. את הריכים בניסוי זה חיברתי על מטריצה, אותה חיברתי להדק ה5V של הבקר. לכפתורים הוספתי נגדים, כמו גם לרגלי המבוא של הנורה. לאחר מכן הרצתי את הקוד שכתבתי, שקובע שכאשר הנורה האדומה תלחץ, ידלק אור אדום בנורה, וכאשר ילחץ הכפתור הירוק, ידלק אור ירוק בנורה. לאחר מכן יהיה ניתן להריץ את הקוד ולצפות בתוצאות.

תוצאות: ניתן לראות כי כאשר מריצים את הקוד, ולוחצים על כפתור בצבע מסויים, ידלק צבע הכפתור בנורה.

מסקנות: מניסוי זה למדתי כיצד לעבוד עם כפתורים בסביבת העבודה של הבקר. למדתי איך לחבר אותם ואיך לגרום לעם לעבוד עם חלקים נוספים כמו נורת RGB.

ברקוד לצפייה בניסוי בסימולציה:

A qr code on a white background

Description automatically generated

תמונת סימולציה:

A circuit board with wires

Description automatically generated

קוד הניסוי: בעמוד הבא

// Define pin numbers

#define redButton 23

#define greenButton 19

#define blue\_pin 2

#define redPin 26

#define greenPin 27

void setup() {

  pinMode(redButton, INPUT\_PULLUP);

  pinMode(greenButton, INPUT\_PULLUP);

  pinMode(redPin, OUTPUT);

  pinMode(greenPin, OUTPUT);

  digitalWrite(redPin, LOW);

  digitalWrite(greenPin, LOW);

  digitalWrite(blue\_pin, LOW);

}

void loop() {

  if (digitalRead(redButton) == LOW) {

    digitalWrite(redPin, HIGH);

    digitalWrite(greenPin, LOW);

    digitalWrite(blue\_pin, LOW);

  } else if

  (digitalRead(redButton) == HIGH) {

    digitalWrite(redPin, LOW);

    digitalWrite(greenPin, LOW);

    digitalWrite(blue\_pin, LOW);

  }

  if (digitalRead(greenButton) == LOW) {

    digitalWrite(redPin, LOW);

    digitalWrite(greenPin, HIGH);

    digitalWrite(blue\_pin, LOW);

  } else if (digitalRead(greenButton) == HIGH) {

    digitalWrite(redPin, LOW);

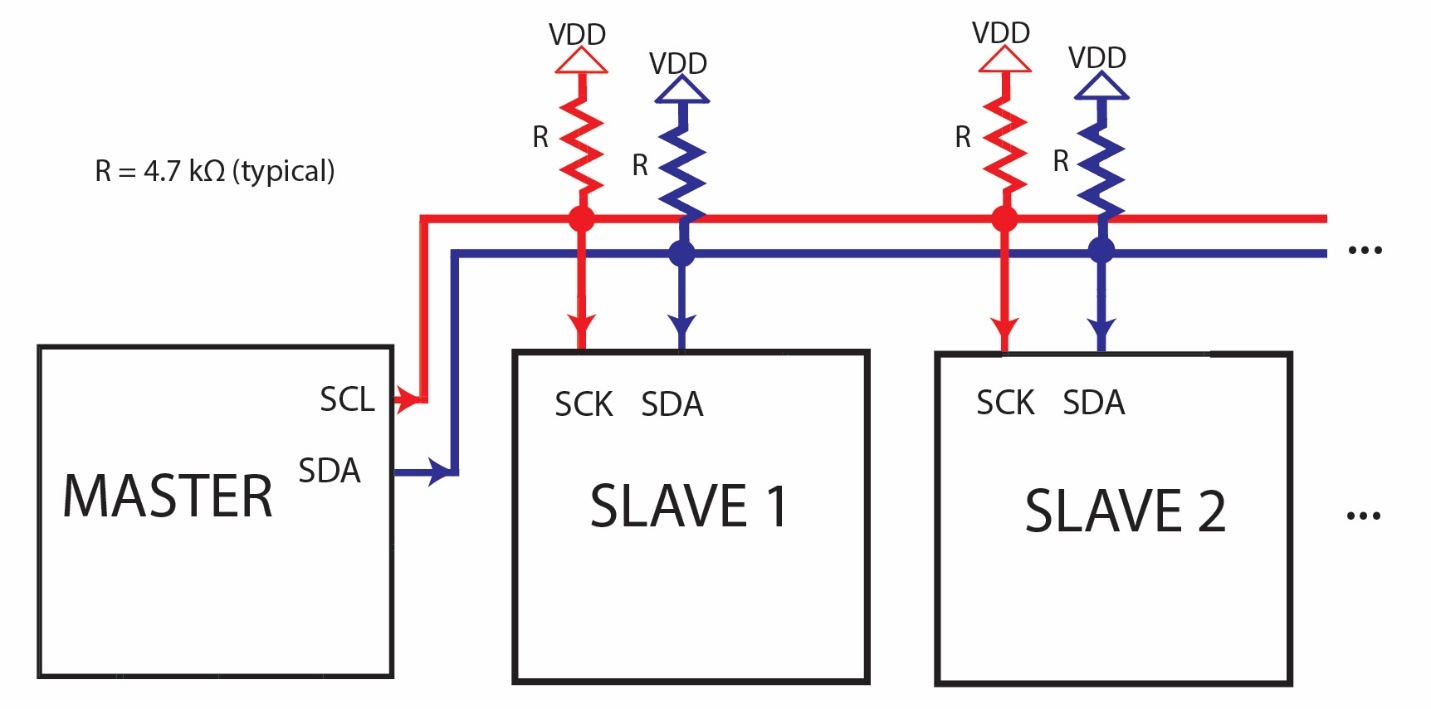
    digitalWrite(greenPin, LOW);

    digitalWrite(blue\_pin, LOW);

  }

}

**פרוטוקול**



פרוטוקול I2C (Inter-Integrated Circuit) הוא פרוטוקול תקשורת טורית דו-כיוונית, המיועד להעברת נתונים בין רכיבים אלקטרוניים. הפרוטוקול פותח על ידי חברת פיליפס (כיום NXP) ומאפשר חיבור של מספר רב של רכיבים על אותו ערוץ תקשורת בעזרת שני חוטים בלבד.

הפרוטוקול עובד בעקרון Master – Slave (מאסטר – עבד). ה Master יוזם את התקשורת עם העבדים במערכת.

מאפיינים עיקריים של התקשורת:

* שני חוטים בלבד:

1. SDA – Serial Data (נתונים טוריים) – בעזרת קו זה יכולים העבדים לשלוח נתונים למאסטר ולהפך. במקרה שלנו זהו פין 21.
2. SCL – Serial Clock (השעון הטורי) – בעזרת קו זה המאסטר מסנכרן את העברת הנתונים. הסנגרון מתרחש בעליית מתח בגל ריבועי ששולח המאסטר בהדק השעון הטורי. במקרה שלנו זהו פין 22.

* כתובות ייחודיות: כל מכשיר המחובר לרשת I2C מקבל כתובת ייחודית המאפשרת תקשורת עם מספר מכשירים על אותו קו.
* קצב העברת נתונים: התקשורת מאפשרת 3 מצבי העברת נתונים:

1. מצב סטנדרטי: עד 100 Kbps בו אנו נשתמש בפרויקט זה.
2. מצב מהיר: עד 400 Kbps.
3. מצב מהיר מאוד: עד 3.4 Mbps.

אופן הפעולה של התקשורת:

התקשורת מתחילה באיתות התחלה, כאשר המאסטר שולח כתובת סלאב וביט קריאה/כתיבה. הסלאב המתאים שולח אישור (ACK) אם הוא מזהה את עצמו בכתובת. הנתונים מועברים בבייטים של 8 ביטים, כאשר כל בייט מלווה באישור מהצד המקבל. התקשורת מסתיימת באיתות סיום.

מסך OLED – SH1106

תצוגת OLED (Organic LED) הינה טכנולוגיה בה כל פיקסל פולט אור משלו (בניגוד לתצוגת LCD בה קיימת תאורה אחורית). תכונה זו מאפשר לו להיות דק וקל משקל ואף להציג שחור מוחלט (הפיקסלים ישארו כבויים).

מכיוון שבתצוגה זו אנו מדליקים רק את הפיקסלים שבהם אנו רוצים להציג מידע, צריכת החשמל תהיה נמוכה יותר ביחס לצגי LCD, בהם יש תאורה אחורית אשר דולקת כל עוד רוצים להציג מידע על המסך.

על הרכיב:

בפרוייקט זה אנו משתמשים במסך SH1106 מבית Adafruit Industries. המסך עובד בתקשורת I2C. בהתאם לפרוטוקול תקשורת זה, יש לחבר את המסך רק לפיני SCL ו- SDK (מלבד חיבורי VCC ו- GND).

שימוש רכיב זה בפרוייקט:

בעזרת מסך OLED, אנו יכולים להציג בצורה נוחה למשתמש מידע מהמערכת. על התצוגה ניתן להציג מרחק מעצמים ומצב של רמזור.

ספריות בהן משתמש הרכיב:

* Wire
* Adafruit\_GFX
* Adafruit\_SH110X

פונקציות עיקריות לשימוש במסך ה OLED:

* Adafruit\_SH1106G – משמשת להגדרת המסך כאובייקט בקוד. מקבל את רוחב ואורך המסך, ציון לעבודה עם פרוטוקול I2Cוציון להדק ריסט (במקרה שלנו אין לכן במקום ייכתב -1).

מחזירה אובייקט מסוג Adafruit\_SH1106G.

* .begin() – מאתחלת את התצוגה. מקבלת את כתובת המסך בתקשורת I2C (במקרה שלנו: 0x3C), והוראה האם לאתחל את המסך או לא.
* clearDisplay() – פונקציה לניקוי כל מה שכתוב על המסך.
* setTextColor() – מגדירה את הצבע של הטקסט שנדפיס על המסך
* setCursor(x,y) – מגדירה את המיקום בו יודפס טקסט על המסך.
* setTextSize() – מגדירה את גודל הטקסט שיוצג על המסך (יכולה לקבל ערכים בין 1 ל 5).
* println() – מדפיסה את הערך שניתן לה על המסך.
* display() – מעדכנת את המסך להצגת השינויים שביצענו בו.

**דו״ח ניסוי -רועי אלבז**

**ניסוי מס׳ 2:**

שם הניסוי: שימוש במסך OLED

מטרת הניסוי: בניסוי זה עלינו להדליק את מסך ה – OLED, ולרשום את השם הפרטי בצד שמאל למעלה בקטן, ואת שם המשפחה בגדול מתחתיו.

מהלך הניסוי:

רשימת ציוד:

* אתר סימולציות Wokwi
* מיקרו בקר ESP32
* מסך OLED

בניסוי זה, נשתמש במסך ה OLED. ראשית אשתמש באתר הסימולציות WOKWI לצורך דימוי השימוש במסך. המסך אשר נמצא בסימולטור הינו מסוג SSD1306, בעוד שהמסך שאנו נשתמש בו בפרוייקט הינו מסוג SH1106G. דבר זה יקשה על העברת הקוד בצורה זהה לבקר עצמו. בהרצת הקוד בסימולציה, התוכנית עובדת כראוי, אך בצריבה לכרטיס, המסך אינו מגיב. לצורך כך היה עליי לחפש באינטרנט כיצד להתגבר על הבעיה הזאת, והבנתי כי אני צריך להוריד ספרייה נפרדת אשר כוללת את הפקודות של המסך שלנו.

תוצאות: לאחר המרת הקוד, מקוד שעובד עם מסך מסוג SSD1306 לקוד שעובד עם המסך SH1106G, על המסך הוצג השם שלי בקטן, ושם המשפחה שלי בגדול, בשוררות נפרדות.

מסקנות: מניסוי זה למדתי איך להשתמש במסך OLED, איך להתגבר על מכשולים ולהתקין ספריות, ואיך להמיר קוד שעובד בסימולציה, לקוד שעובד עם הבקר האמיתי.

קוד התוכנית:

#include <Wire.h>

#include <Adafruit\_GFX.h>

#include <Adafruit\_SH110X.h>

#define SCREEN\_WIDTH 128 // OLED width, in pixels

#define SCREEN\_HEIGHT 64 // OLED height, in pixels

// create an OLED display object connected to I2C

Adafruit\_SH1106G oled = Adafruit\_SH1106G(SCREEN\_WIDTH, SCREEN\_HEIGHT, &Wire, -1);

void setup() {

Serial.begin(9600);

// initialize OLED display with I2C address 0x3C

if (!oled.begin(0x3C, 1)) {

Serial.println(F("failed to start OLED"));

while (1);

}

delay(2000); // wait two seconds for initializing

oled.clearDisplay(); // clear display

oled.setTextSize(1); // set text size

oled.setTextColor(SH110X\_WHITE); // set text color

oled.setCursor(2, 10); // set position to display (x,y)

oled.println("Roy"); // set text

oled.setTextSize(2); // set text size

oled.setCursor(5, 25); // set position to display (x,y)

oled.println("Elbaz"); // set text

oled.display(); // display on OLED

}

void loop() {

}

קישור לסרטון המעשי: קישור לסימולציה:



תמונת מסך של הסימולציה:

A screenshot of a computer

Description automatically generated