Communication
System Engeneering

תשתית תקשורת ברכב מרוץ אוטונומי

דו"ח מסכם

מס' פרויקט: 3711908 בפקולטה להנדסה

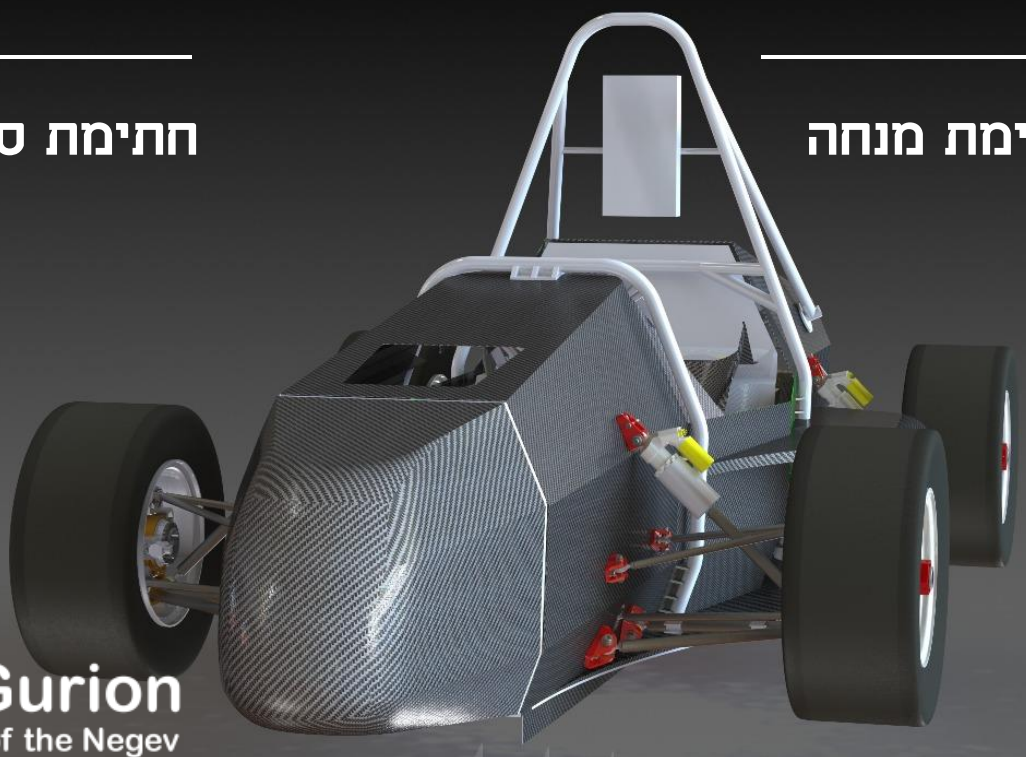
גיא פלח 201369998

בהנחיית פרופ' חן אבין

25.08.2019

חתימת מנחה

חתימת סטודנט



Ben-Gurion
University of the Negev

תוכן עניינים

רשימת טבלאות 6

רשימת קיצורים 7

1 הגדרת הפרויקט 8

1.1 הרקע והמוטיבציה לפרויקט 8

1.2 הצגת הבעיה 8

1.3 מטרת הפרויקט 9

2 רקע תאורטי 10

2.1 ממשקי תקשורת נפוצים ברכבים 10

2.2 מערכות אלקטרוניות ברכב 12

2.3 אתרנט ברכבים 14

2.4 פרוטוקולי אתרנט ליישומי זמן אמת 16

3 השיטה 20

3.1 מערכות הרכב האלקטרוניות BGR2019 22

3.2 סביבת עבודה 26

4 שלבים לביצוע הפרויקט 27

4.2 יומן פרויקט 30

5 שלבים במימוש המערכת 32

5.1 שיקולים לבחירת החומרה 32

5.2 שלבים בתכנון לוח PCB 32

5.3 מעבר כללי על הלוח ה Gateway 33

5.4 רכיבי ה-IC בלוח 33

5.5 מעגלים מרכזיים בלוח 38

5.5.1 Stm32F765 38

5.5.2 CAN-bus 42

5.5.3 Ethernet 44

5.5.4 חיבור MII 46

5.5.5 הוספת קריסטל קוורץ 46

5.5.6 ממשקי קינפוג ודיבאג 47
5.5.7 מקורות מתח 47
5.6 הלחמה ובדיקת הלוחות 48
5.6.1 תהליך בדיקת הלוח 49
5.7 אלגוריתמיקה ותוכנה 50
5.8 אינטגרציה ברכב – הטמעת המערכת ברכב 51
5.8.1 מערכות התקשורת ברכב 51
5.8.2פיתוח יכולות לפני התקנתן על הרכב 53
5.8.3 עבודה על הרכב והכנת הרכב לנסיעה 53
5.9 תחרות 54
6 מקורות 57

רשימת איורים

איור 1 - מבנה הודעת CAN Bus 10
איור 2 - סקירת טופולגיית מערכת Torque vectoring 13
איור 3 - תרשים בלוקים כללי במערכת ADS 14
איור 4 - מוסר/מקבל של סטנדרט IEEE 802.1 QAV 18
איור 5 - תעדוף וגישה מבוססת טריגרים ב-timed triggered ethernet 18
איור 6 - רכיבי הרשת 20
איור 7 - דוגמת תרחיש משתמש 20
איור 8 טופולוגיית הרשת 21
איור 9 - תרשים חוליות של ה-Gateway 21
איור 10 - סכימת המערכת 22
איור 11 - סכימת Drive-by-wire 23
איור 12 - סכימת המערכת ממשק אדם מכונה 24
איור 13 - סכימת מערכת traction control 25
איור 14 - סכימת מערכת ADS 25
איור 15 - ממשק משתמש של STM32CubeMX 26
איור 16 - ממשק מערכת Kicad 26
איור 19 - סכימת בלוקים של הלוח ..... **Error! Bookmark not defined.**
איור 20 - תרשים של בקר stm32f765 והרגליים שבשימוש בלוח 34
איור 21 - תרשים pinout של ה-Ethernet switch 35
איור 22 - תרשים בלוקים של המודולים מהם מורכב ה-Ethernet switch 35
איור 23 - סכימה חשמלית של ה-CAN-bus transceiver 36
איור 24 - סכימה חשמלית של המעגל ב-RJ45 37
איור 25 - שרטוט מכאני של RJ45 37
איור 26 - סכימת application note לגבי המלצת היצרן לאספקת מתח ללוח 38
איור 27 - סכימה חשמלית של אספקת המתח ל-stm32 במעגל שלנו 39
איור 28 - מימוש המעגל ב-layout 40
איור 29 - הדמיה של מימוש המעגל 40
איור 30 - תמונה של המעגל בפועל 41
איור31 - אספקת המתח לרכיב ה-ethernet switch 41
איור 32 - לוח פיתוח nucleo 144 אליו מחובר מודול CAN transceiver 42
איור 33 - הדמיה של מעגל ה-CAN-bus על הלוח 43
איור 34 - שרטוט חשמלי של מעגל ה-CAN-bus 43
איור 35 – המלצת היצרן לחיבור transformer אל ה-PHY של ה-Ethernet swtich 44
איור 36 - סכימה חשמלית של חיבור ה-RJ45 ל-PHY 45

איור 37 - חיבור ה-RJ45 ל-Ethernet switch 45
איור 38 - תרשים layout של חיבור MII 46
איור 39 - ממשק הדיבאגר וה-SPI 47
איור 40 - מקור המתח בלוח 48
איור 41 - תהליך בדיקת קצרים בין הרגליים של stm32f765 בעזרת מיקרוסקופ ו-וולטמטר 50
איור 42 - קופסת תקשורת ובתוכה מחשב הרכב 52
איור 43 - בקר דוושות 52
איור 44 - דשבורד 53
איור 45 - עבודה והכנת הרכב לקראת נסיעה – מתחילים בערב, מסיימים בבוקר 54
איור 46 - רגע ההודעה על זכייה בפרס driving the future 55

רשימת טבלאות

טבלה 1 - פרוטוקל ממשק תקשורת CAN Bus 10
טבלה 2 - מבנה חוליית CAN Bus 10
טבלה 3 - מאפייני ממשק התקשורת LIN 11
טבלה 4 - מודל השכבות בממשק FlexRay 11
טבלה 5 - דרישות תקשורת עבור מכלולי הרכב השונים 15

רשימת קיצורים

BGR - Ben Gurion racing

FSAE – Formula SAE

SAE - Society of Automotive Engineers

Can - Controller Area Network

ADS – Automated driving system

HMI – Human Machine Interface

TDMA - Time-division multiple access

ECU - Electronic Control Unit

VCU - Velocity control unit

CSMA/CD + AMP - Carrier Sense Multiple Access / Collision Detection with Arbitration on Message Priority

UTP - Unshielded Twisted Pair

AUTOSAR - Automotive Open System Architecture

AVB – audio video bridge

RMII – reduced Media-independent interface, standard interface to connect a Fast Ethernet (i.e., 100 Mbit/s) media access control (MAC) block to a PHY chip

PWM – Pulse width modulation

PCB – Printed Circuit Board

RPI - Raspberry Pi

1 הגדרת הפרויקט

1.1 הרקע והמוטיבציה לפרויקט

משנת 2011 קיימת באוניברסיטה קבוצת רכב מרוץ, BGR. הקבוצה מתכננת ומייצרת רכבי מרוץ ושולחת אותם להתחרות בתחרויות Formula student אשר נערכות על ידי חברי ארגון Society of Automotive Engineers.

תחרויות ה-Formula student, או בקיצור FSAE, החלו כבר בשנת 1979 במטרה להכין סטודנטים שיוצאים מהאוניברסיטה לתעשיית הרכב, וכיום יש מעל 500 קבוצות פעילות בעולם.
התחרות נערכת בשלושה אופנים:

1. רכבי מרוץ עם מנוע בעירה
2. רכבי מרוץ חשמליים
3. רכבי מרוץ אוטונומים (החל משנת 2017)

רכב המרוץ החשמלי מסוג FSAE הינו המשכו של רכב מרוץ הבעירה ובתחילת דרכו ירש הרכב החשמלי את רוב התכונות של רכב הבעירה, ביניהן את שיטת התקשורת can bus. במעבר מרכב בעירה לרכב החשמלי נוספו יותר מערכות מחשוב, אך שיטת התקשורת נשארה אותו הדבר. עם זאת, כמות תעבורת המידע ברכב החשמלי גדלה ולכן ברכבים חשמליים נראה מספר רשתות can bus מקבילות, תכונה אשר לא הופיעה ברכב הבעירה.
בעקבות זאת, נרצה לארגן מחדש את שיטת התקשורת ברכב המרוץ,

1.2 הצגת הבעיה

ברכב הקבוצה קיימות מספר רשתות שרצות במקביל מבלי שקיימת מוסכמה איך לחבר ביניהן. בכל פעם שרוצים להוסיף מערכת חדשה לרכב, מתווספת רשת קטנה ללא חשיבה רחבה יותר.בהצעת חלופת פתרון התקשורת נרצה להתאים את הפתרון עבור המערכות הבאות:

1. Drive-by-wire
2. Traction control
3. ADS – Automated driving system
4. HMI – Human Machine Interface
5. Data Logging

1.3 מטרת הפרויקט

מטרת הפרויקט תכנון תשתית תקשורת לרכב אוטונומי מובסס Ethernet.

מטרות משנה:

- הכרת פתרונות התקשורת מבוסס Ethernet ברכבים
- הטמעת מערכת התקשורת ברכב המרוץ של קבוצת בן גוריון
- יצירת Framework המקל על הוספת מערכות מחשוב לרשת הרכב
- מתן תשובה אכותית לגבי תוצר המערכת , זמני השהייה, נצילות , רוכב פס , התמודדות בפני נפילות , signal integrity.

קריטריונים לטיב המערכת :

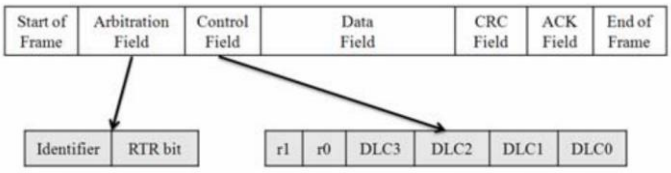
- התאמה לסביבת הרכב – נרצה שהשכבה הפיזית תתאים לסביבת הרכב.
- סקלאביליות – תאפשר הוספת והורדת רכיבים מהרשת באופן קל וגנרי.
- תמיכה במערכות Real time – חלק מהיישומים שהרשת משרתת הן מערכות המצריכות תעבורה דטרמנסטית של מידע.

2 רקע תאורטי

2.1 ממשקי תקשורת נפוצים ברכבים

בתעשיית הרכב השימוש במיקרו מעבדים גדל בעקביות. יותר ויותר רכיבים מכאניים מוחלפים במיקרו מעבדים. במאמר [1] קיימת סקירה אודות ממשקי התקשורת הנפוצים ביותר בתעשיית הרכב, נכון להיום. להלן הסבר אודות ממשקים אלו כפי שמובאים במאמר:

- (CAN) Controller Area Network – זהו ממשק תקשורת אשר פותח לראשונה על ידי רוברט בוש בשנת 1986. זוהי מערכת מרובת Master-ים התומכת ברוחב פס של עד 1 Mbits per second. ייחודה של המערכת טמון בשיטת הגישה לערוץ שלה, CSMA/CD+AMP. שיטה זו מאפשרת למספר חוליות CAN-bus לחלוק ערוץ משותף מבלי שיתקיימו התנגשויות, תוך מתן עדיפות לחוליות החשובות יותר. הפרוטוקול מאופיין בחבילות מידע קצרות הנעות בין בית אחד של מידע לבין 8 בתים. הרשת מיועדת עבור מרחקים שלא עולים על 40 מטר. תכונה נוספת של הממשק הוא חסינותו להפרעות אלקטרו-מגנטיות, דבר שמאפשר לטווח שלו להיות מורכב מכבל UTP. תכונות אלו הפכו אותו למועדף בחברות הרכב. בטבלה 1 ניתן לראות את הפרוטוקול של ממשק התקשורת CAN Bus ובטבלה 2 ניתן לראות את המבנה של חוליית (node) CAN Bus ובאיור 1 מוסבר אודות המבנה של הודעת CAN Bus.



איור 1 - מבנה הודעת CAN Bus

טבלה 1 - פרוטוקל ממשק תקשורת CAN Bus

- Hierarchical ranking of message - Guaranteed latencies - Flexible configuration - Reception of multiple sources with time synchronization - Multimaster operation - Error detection and signaling - Automatic retransmission of corrupted messages - Distinction made between timing errors - Permanent non-functionality at a node - Automatic disconnection of faulty nodes

טבלה 2 - מבנה חוליית CAN Bus

Application Layer	Description
Object	- Message Filtering - Message and Status handling
Transfer	- Fault Confinement - Error Detection and Signaling - Message Validation - Acknowledgment - Arbitration - Message Framing - Transfer Rate and Timing
Physical	- Signal Level and Bit Representation - Transmission Medium

- (LIN) Local Interconnect Network – ממשק זה הופיע לראשונה ביולי 1999. טופולגיית הממשק מורכבת מ-Master אחד ומספר Slave-ים. ה-LIN הוא פרוטוקול נחות יותר מה-CAN-bus ומטרתו היא להוזיל עלויות במקומות בהם הדרישה ל-real-time אינה נחוצה ותעבורת המידע נמוכה. כמו ה-CAN-bus, כך גם ה-LIN מורכב ממספר חוליות המחוברות בטווח משותף UTP. ה-LIN נפוץ ביישומים כמו פתיחת חלונות הרכב, מתן פקודה למגבים ועוד. בטבלה 3 ניתן לראות את המאפיינים של ממשק התקשורת LIN.

טבלה 3 - מאפייני ממשק התקשורת LIN

- Single master with multiple slave nodes - Low cost implementation based on a UART interface - Self synchronization without quartz in slave node - Deterministic signal transmission - Low cost single wired implementation - Signal -based application interaction - Re-configurability - Speed up to 20Kbits per second - Predictable behavior - No arbitration necessary - Guaranteed latency times for signal transmission

- FlexRay – פתרון המתעלה על הביצועים של ה-CAN-bus, והומצא מאוחר יותר במטרה לתת פתרון טוב יותר מה-CAN-bus. פרוטוקל זה גם כן מרובה master-ים ומבוסס על חלוקת הערוץ לחלונות זמן מחזוריים (TDMA). כל חלון זמן מחולק ל-2 תתי חלונות זמן: הראשון שמור להודעות סטטיות והשני להודעות דינאמיות. בכך הפרוטוקול משלב שירותים דטרמיניסטיים (הודעות המיועדות לחלון הזמן הסטטי) והודעות ארעיות (המיועדות לחלון הדינמי, בדומה ל-CAN-bus). בטבלה 4 ניתן לראות את מודל השכבות המקובל בממשק FlexRay.

טבלה 4 - מודל השכבות בממשק FlexRay

Layer	Description
Application	- User-Programmed Application
Presentation	- Frame Filtering and Masking - Frame and Status Handling
Transfer	- Fault Confinement - Error Detection and Signaling - Message Validation - Message Framing - Communication Cycle - Synchronization
Physical	- Fault Confinement - Error Detection and Signaling - Error Confinement in Time Domain - Signal Level and Bit Representation - Transmission Medium

2.2 מערכות אלקטרוניות ברכב

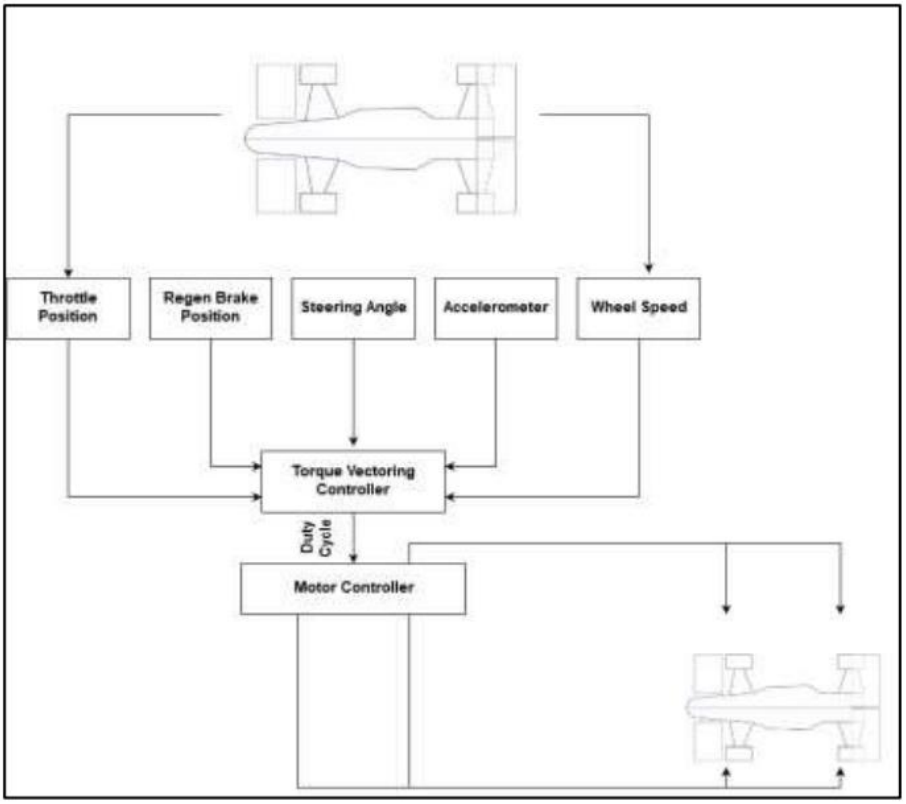
1. Drive-by-wire – המונח drive-by-wire נוגע לכל המערכות האלקטרוניות המרחיבות או מחליפות את מנגנוני השליטה המכאניים המסורתיים. במקום להשתמש בשיטות מכאניות כגון משאבות הידראוליות וכבלים, המספקות גישה ישירה בין הנהג לבין הרכב, טכנולוגית ה-drive-by-wire משתמשת בחלופה אלקטרונית על מנת לפקד על הרכב [2].

אופן פעולת המערכת [3]: אל ממשקי הנהיגה מחברים חיישנים אשר מודדים את מצבם, דוגמא לכך היא זווית מיקום דוושת התאוצה. במקום שדוושת התאוצה תמשוך את מצערת המנוע באופן מכאני, חיישן אשר ממקום על דוושת התאוצה מתרגם את זווית הדוושה להודעה דיגיטאלית ושולח אותה אל בקר המנוע, לפי הודעה זו פוקד בקר המנוע על פתיחת המצערת.
בשיטה זו ניתן להטמיע לוגיקה נוספת, שכן בקר המנוע יכול לקבל קלטים נוספים ולפיהם להחליט על כמות פתיחת מצערת עדיפה.

2. Traction control – מערכת בקרת האחיזה מבקרת על כמות המומנט שמספק המנוע לגלגלים בכדי למנוע החלקה בעת תאוצה. למערכת זו יש יחידת שליטה אלקטרונית, ECU – Electronic Control Unit. יחידה זו עוקבת אחר מצב אחיזת הגלגלים ברכב. זיהוי ההחלקה מתבצע משערוך מצב הרכב בעזרת מידע המגיע מקבוצת חיישנים הפזורים עליו, למשל חיישני תאוצה מהירות גלגל, מהירות סיבוב המנוע וכיו"ב. המידע מתנקז אל בקר ה-ECU. בקר ה-ECU מעדכן בכול איטרציה את ערך המומנט המקסימאלי שניתן לדרוש מהמנוע.
בעיית ההחלקה ברכבי מרוץ היא בעיה נפוצה כיוון שלרכבים אלו יש כוח מתפרץ רב.
מערכת Traction control משפרת באופן ניכר את ביצועיו הדינאמים.

ישנן מספר שיטות למימוש מערכת Traction control, אחת מהן היא torque vectoring [4] – שיטה שפותחה בשנת 2006 ומקובלת ברכבים בעלי מספר מנועים (מנוע אחד לכל גלגל) וכוללת את הרכיבים הבאים:

- Throttle position sensor
- Brake position sensor
- Half effect sensors
- Inertial measurement unit
- Steering angle sensor
- Controller Software
- Electronic control unit
- Motor controller



איור 2 - סקירת טופולגיית מערכת Torque vectoring

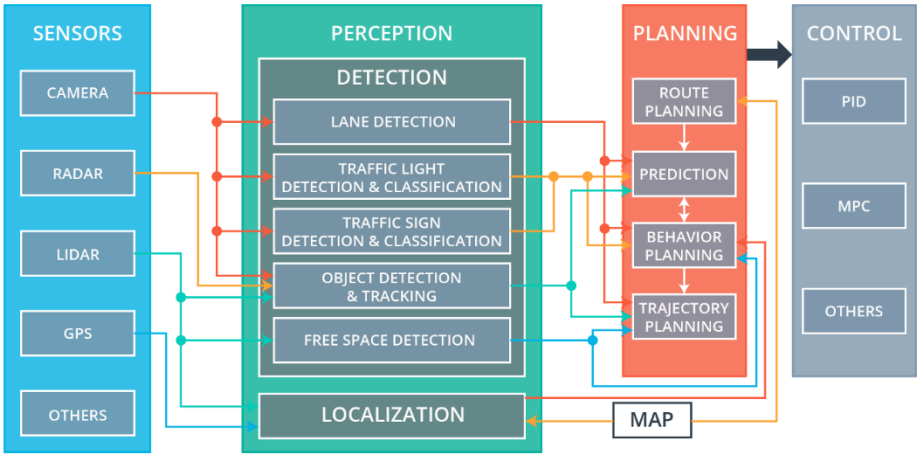
כל הרכיבים מתקשרים ביניהם בדרך כלל בממשק CAN-bus. תדירות עדכון ערך המומנט המקסימלי תלויה בתדירות פקודת הנסיעה שנשלחת למחשבי המנוע.
בשיטה זו משתמשים בקבוצת הרכב של בן גוריון, באיור 2 ניתן לראות תרשים בלוקים למערכת כזו מתוך המאמר מתוך [4].

3. ADS – Automated driving system – מערכת אלקטרונית שמחקה את פעילות הנהג ומחליפה אותו.

ניתן לחלק את את אופן פעולת המערכת לארבעה שלבים עיקריים (ראה איור 3):

א. שלב החישה – דגימת המידע מהחיישנים ושליחתו אל יחידת ה-perception.
ב. שלב התפיסה – בשלב זה כל המידע מהחיישנים עובר עיבוד, ובסופו המערכת יודעת להבין איפה היא נמצאת במרחב וביחס לאובייקטים אחרים.
ג. שלב התכנון – בשלב זה המערכת מחליטה על פקודת הנסיעה הבאה: כיצד לכוון את ההגה, אילו חיוויים לתת לדוושות וכד', ושולחת את הפקודה אל יחידות השליטה
ד. שלב השליטה – מקבל את החיווי ומנהג את הרכב בהתאם.

מאחר והמערכת מחקה באופן מלאכותי את פעילות הנהג, נוסיף לחיישנים הקיימים מצלמות אשר מקבילות לפעולת הראייה. נהוג להעביר את התמונות ללא שימוש בדחיסה, על מנת להקטין את ה-delay. מנגד, רוחב הפס גדל [5].



איור 3 - תרשים בלוקים כללי במערכת ADS

4. HMI – Human Machine Interface – ממשקי אדם מכונה. ממשקים אשר מספקים מידע על מצב הרכב.
5. Data Logging – יחידה האוגרת את כל המידע העובר ברשתות הרכב למטרות דיאגנוסטיקה ומעקב אחר מצב מערכות הרכב.

2.3 אתרנט ברכבים

מאמר [7] הינו מאמר סקירה אודות אתרנט ברכבים – automotive ethernet – ובחלק זה אביא את עיקרי הדברים אודות נושא זה.

הצורך במערכות מחשוב ברכב עולה, מה שמוסיף לסיבוכיות הכבילה. רשת Ethernet ייעודית ברכבים יכולה להפחית את עלות וסיבוכיות הרשת.

ניתן לחלק את המערכות האלקטרוניות ברכב לתחומים:

- הנעה – כולל את המנועים ומערכת התמסורת.
- שלדה – כוללת את הבלמים, היגוי ומתלים.
- Driver Assistance – מערכות לזיהוי נתיב, זיהוי הולכי רגל.
- ממשק אדם-מכונה – מערכות אלו כוללות מנגנונים המסייעים לנהג ונועדו להעלאת בטיחותו.
- פנים הרכב – מושבים, מערכת מולטימדיה, כריות אוויר.

לכל אחד מהתחומים קיימת רשת מקומית משלו, ולכל רשת יש את הדרישות שלה. בטבלה 5 מובאות הדרישות למכלולים השונים.

טבלה 5 - דרישות תקשורת עבור מכלולי הרכב השונים

Domain	Description	End-to-End Latency Requirements	Bandwidth Requirements
Powertrain	Controls the components that generate power and transmit to the road.	< 10 us	Low
Chassis	Controls steering, brakes, suspension	< 10 us	Low
Body and Comfort	Radio, A/C, window, seat, and light controls	< 10 ms	Low
Driver Assistance and Driver Safety	Controls systems designed to increase safety	< 250 us or < 1 ms depending on system	20 – 100 Mbps per camera
Human-Machine Interface	Controls displays and other interfaces that interface with the driver or passengers	< 10 ms	Varies by system, but this growing

Automotive Ethernet זוהי רשת פיזית אשר מתוכננת להתאים לצרכי תעשיית הרכב ועתידה לשמש כ-backbone לכל תתי הרשתות.

הדרישות מ-Automotive Ethernet הן:

- עמידות בפני הפרעות אלקטרו-מגנטיות
- רוחב פס מתאים
- עמידה בדרישות זמן ההשהייה
- סינכרון
- דרישות ניהול הרשת

כדי להתאים את ה-Ethernet לרכב נעשה שימוש בטכנולוגיות הבאות:

- AUTOSAR (Automotive Open System Architecture) – AUTOSAR הוא סטנדרט הנוגע לארכיטקטורת תוכנה במערכות מחשוב ברכבים. הארכיטקטורה כוללת דרישות עבור מערכת ההפעלה היישומים ודגש למימוש ה ל-Automotive TCP/UDP/IP stack, כדי שיתאים לסביבת הרכב.

- One-Pair Ether-net (OPEN) – טכנולוגיה המתייחסת לטווח הפיזי של ה-Ethernet (PHY), המסוגלת להעביר כמות מידע העולה על 1 GB per second לשני הכיוונים במקביל על טווח של UTP יחיד.
- Time Synchronization – עבור יישומי זמן אמת קיים סטנדרט המובא ב-[6] לסנכרון בין רשתות מקומיות
- AV Bridging – פרוטוקולים המאפשרים שידור של וידיאו על גבי רשת ethernet תוך שמירה על קצב שידור קבוע ו-delay מינימאלי, ועומדים בדרישות של מערכות ADS.

2.4 פרוטוקולי אתרנט ליישומי זמן אמת

[9] HDbaseT

- הגדרה – פרוטוקול מיתוג מנות אשר פותח על ידי חברת Valens. הפרוטוקול מיועד לשרת יישומי זמן אמת, בדגש על תעבורת וידיאו.

OSI 7 Layers	Function	HDBaseT Implementation
Application	Responsible for determining when access to the network is required.	HDBaseT T-Adaptors for Video, Audio, USB and Control
Presentation	Ensures data is received in a useable format.	HDBaseT protocol for secured and reliable delivery
Session	Establishing and maintaining connections. Responsible for port assignments.	
Transport	Breaks data into frames and assigns sequence numbers.	HDBaseT Session Multiplexing
Network	Source-destination addressing. Subnets and path determination exist at this layer.	HDBaseT Network (Switching) Layer
Data Link	Frames exist here. This layer handles flow control. Specifies topology and provides hardware addressing.	HDBaseT Link Layer
Physical	Transmission of the raw bit stream. Electrical signaling and hardware interface.	HDBaseT Copper/Fiber PHY

Table1 - HDbaseT OSI Layers [9]

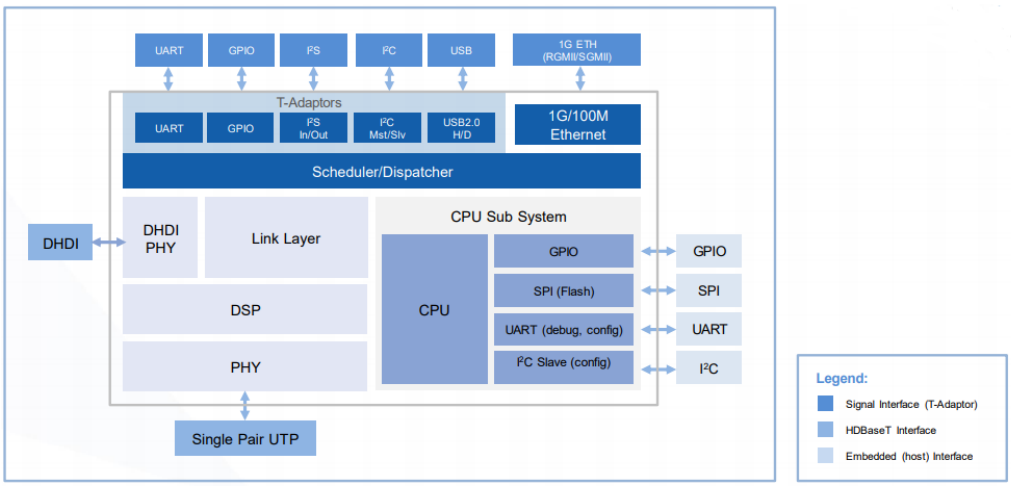


Figure1 - VA6000 Block Diagram [9]

- מאפיינים –
  - השכבה הפיזית (PHY) – HdbaseT משתמש בשיטת PAM16/PAM8/PAM4 להעברת symbol-ים. ל-HDbaseT יש את היכולת לשנות את מספר ה-symbol-ים בהתאם לאחוז השגיאה. ה-PHY של HDbaseT הוא automotive prove.
  - תכונת ה-tunneling – HdbaseT משתמש בשיטת tunneling בכדי להעביר מידע של הפרוטוקולים הבאים: USB, I2C, UART. הוא עושה זאת ע"י הוספת header של HDbaseT, ושולח אותו על גבי הרשת. חבילות ethernet נשלחות לרשת בצורתן המקורית.
  - מאפייני שכבת הרשת – פרוטוקל HDbaseT מממש במקביל 2 סוגי רשתות על אותו טווח, שהן:
    - T-Network – רשת המנתבת מידע בשיטת circuit switching.
    - E-Network – רשת המנתבת מידע בשיטת packet switching.

בכדי לשלוח הודעה על גבי ה-T-Network תחילה יש לממש session בין שני רכיבי HDbaseT, כל session מקבל ID ייחודי ומוקצה לו רוחב פס ייעודי על גבי נתיב מסוים ברשת.

- טופולוגיית הרשת – ניתן לבנות את הרשת בטופולוגית Daisy-chaining, כאשר רכיבי ה-HDbaseT מסוגלים לתקשר ביניהם באמצעות הודעות בקרה (HD-CMP).

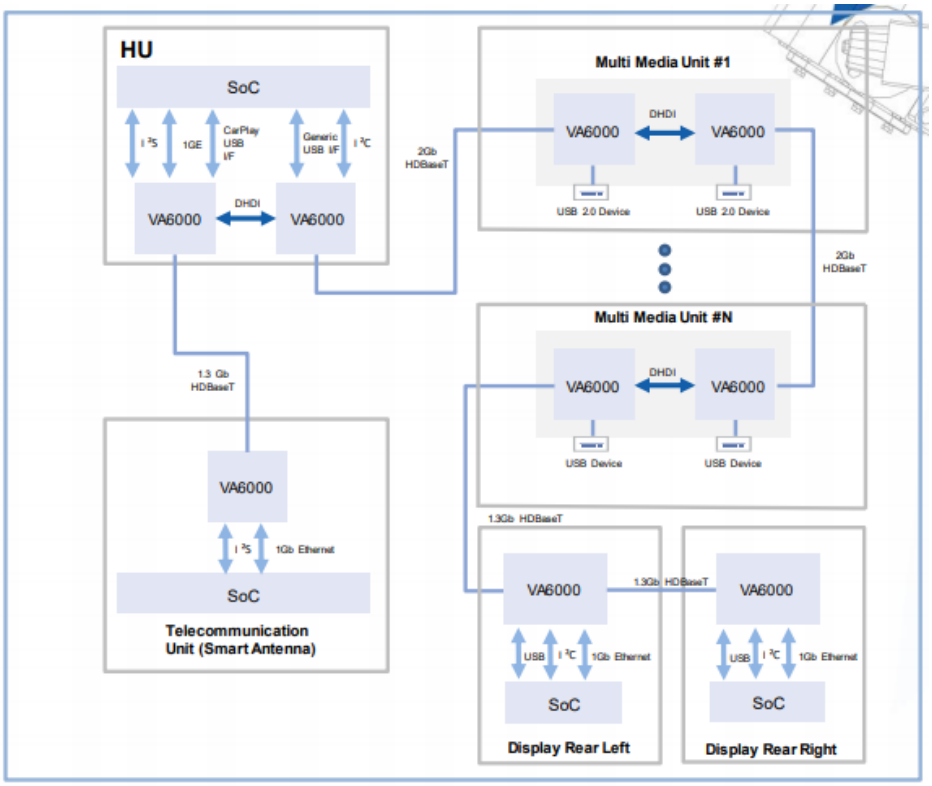
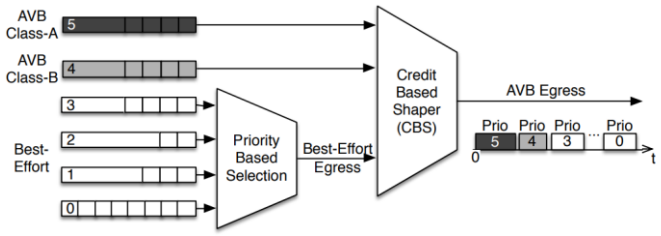


Figure2 - HDbaseT Topology Example [9]

Time-Sensitive Networking (IEEE 802.1 AVB)

זהו סטנדרט המשמש להעברת הודעות עבור יישומי זמן אמת על גבי תשתית של Ethernet (IEEE 802), על ידי הוספת הפרוטוקולים הבאים (כפי שמובא ב-[8]):

1. IEEE 802.1 AS – פרוטוקול סנכרון רשתות המאפשר לסנכרן את רכיבי הרשת בדיוק של עד 1µs ובמרחק של 7 hop-ים.
2. IEEE 802.1 QAV – סטנדרט מגדיר את מדיניות התורים ב-AVB Ethernet ומבוסס על אלגוריתם CBS (Credit Base Shaping). מחלק את התור ל-3 סוגי שירות: A, B ו- Best Effort. ה-delay של שירות A מקצה לקצה הוא הקצר ביותר, אחריו B ולבסוף Best Effort. באיור 4 ניתן לראות תרשים של מוסר/מקבל הסטנדרט.



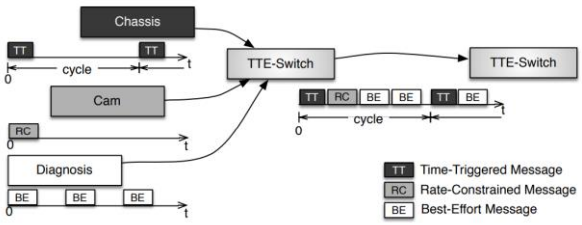
איור 4 - מוסר/מקבל של סטנדרט IEEE 802.1 QAV

3. IEEE 802.1 QAT – Stream Reservation Protocol. בהינתן switch-ים שתומכים ב-AVB, מאפשר להירשם ל-stream, לפרסם stream ולבטל רישום ל-stream באופן דינאמי.

Time-Triggered Ethernet (AS6802)

סטנדרט המאפשר לספק שירותי רשת ליישומי זמן אמת, בכך שהוא מגדיר פרוטוקול סנכרון רשת ו-3 רמות של סוגי הודעות (כפי שמובא ב-[8]):

1. Time Triggered – שומרים slot-ים מראש עבור ההודעות מקבוצה זו (TDMA). באיור 5 ניתן לראות תרשים של תעדוף וגישה להודעות בסוג הודעה זה.
2. Rate Construction – קבוצת הודעות זו תמיד תהיה מתועדפת מעל קבוצת הודעות Best Effort.
3. Best Effort



איור 5 - תעדוף וגישה מבוססת טריגרים ב-timed triggered ethernet

(IEEE 1588-2008) Precision-Time Protocol

סטנדרט לסנכרון שעונים ברשתות LAN. עובד בשיטה החלפת הודעות עם timestamp ובנוסף מכיל חומרה ייעודית להוספת וקריאת חתימת הזמן, ובכך מדייק את הסנכרון. אודות סטנדרט זה ניתן לקרוא ב-[8].

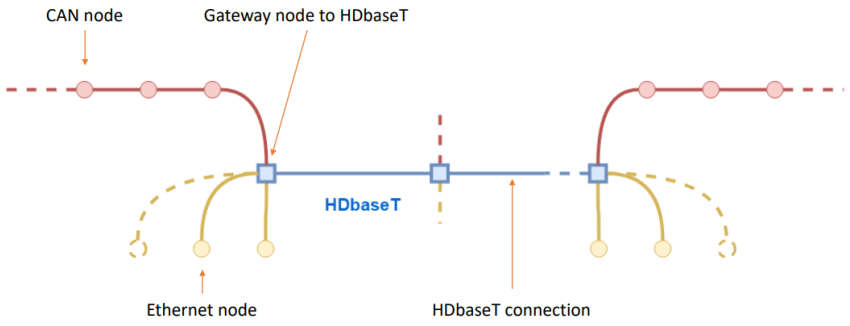
3 השיטה

המטרה בפתרון היא להוסיף יכולת העברת מידע ברכב מכל רכיב לכל רכיב, והשיטה היא לקשר בין כל רשתות ה-CAN-bus המקוריות אחת לשנייה. בכדי לעשות זאת נעשה שימוש בהוספת רכיבים יעודיים, Gateway nodes.

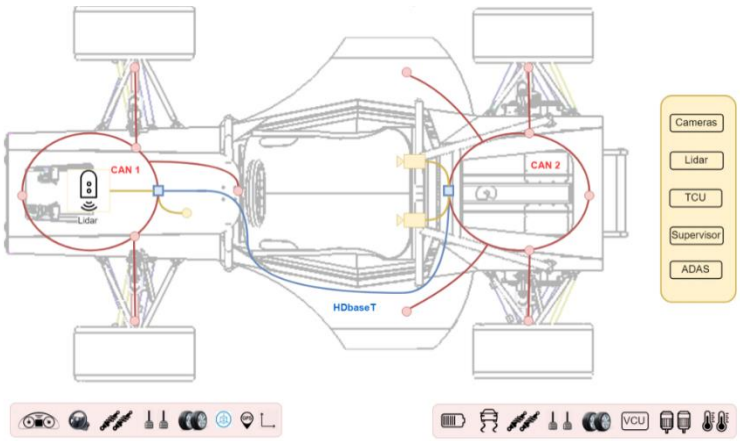
הרכיב תומך בממשקים בהם אנו משתמשים ברכב: Ethernet , CAN-bus , וכל תפקידו הוא לנהל את תעבורת המידע ברכב.

ה-Gateway-ים מנהלים את תהליך תעבורת המידע ברכב, מספקים שירותי סנכרון ובודקים את תקינות יכולת הכיבול ברשתות ה-CAN-bus. כמו כן, מאחר ומדובר ברשתות המיועדות לשרת יישומי זמן אמת, על ה-Gateway-ים לפעול ביעילות מבלי ליצור עיכובים בתעבורת המידע.

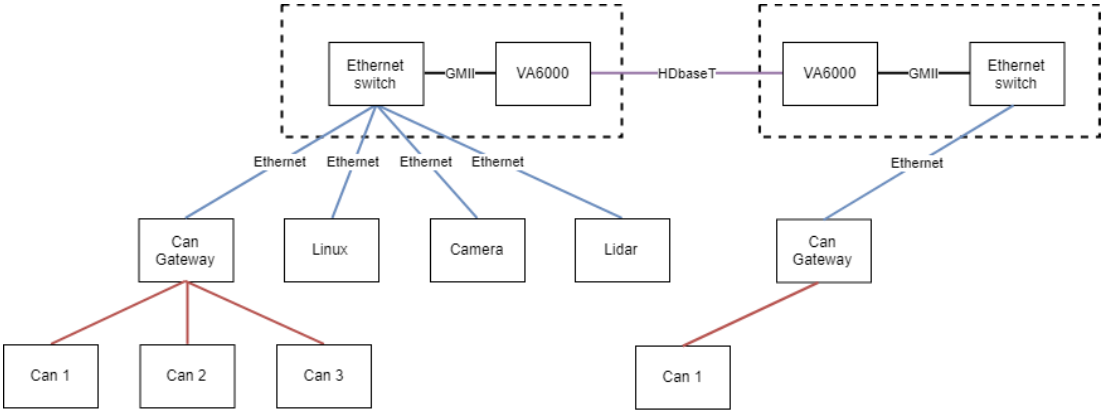
באיור 6 ניתן לראות תרשים של רכיבי הרשת ואילו באיור 7 ניתן לראות דוגמה לאיך יהיו הרכיבים מחוברים ברכב.



איור 6 - רכיבי הרשת



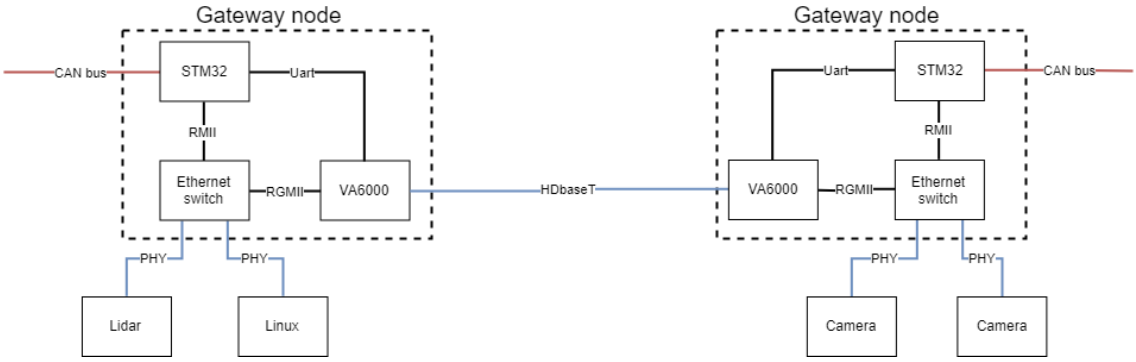
איור 7 - דוגמת תרחיש משתמש



איור 8 טופולוגיית הרשת

מבנה ה-Gateway

ה-Gateway מורכב מ-3 מודולים עיקריים (באיור 8 ניתן לראות תרשים של ה-node-ים ב-gateway):



1. בקר Stm32F767zi – מתממשק בצד אחד עם רשת CAN-bus ומצד שני מתקשר עם ה-Ethernet switch, באמצעות ממשק RMII. ה-CAN-bus וה-RMII מובנים כמודולים בתוך הבקר. לבקר תמיכה חומרתית בפרוטוקול PTP, הוא מסוגל לשרת עד 3 רשתות CAN-bus שונות במקביל והוא בעל ביצועים גבוהים. יש לו 2MB זיכרון ועובד בתדירות שעון של 216mhz. הקבוצה בעבר עבדה עם מספר בקרים שונים, אך לבסוף הוחלט כי בקרים של חברת stm הינם המתאימים ביותר עבורה.

2. 1Gb Ethernet switch – נועד בכדי לאפשר בניית רשת Ethernet על הרכב.

3. HDbaseT controller and PHY VA6000 – משמש כ-PHY עבור ה-port המנתב אל הקצה המרוחק של הרכב. ל-HDbaseT יש עמידות גבוהה בפני רעשים אלקטרו-מגנטיים.

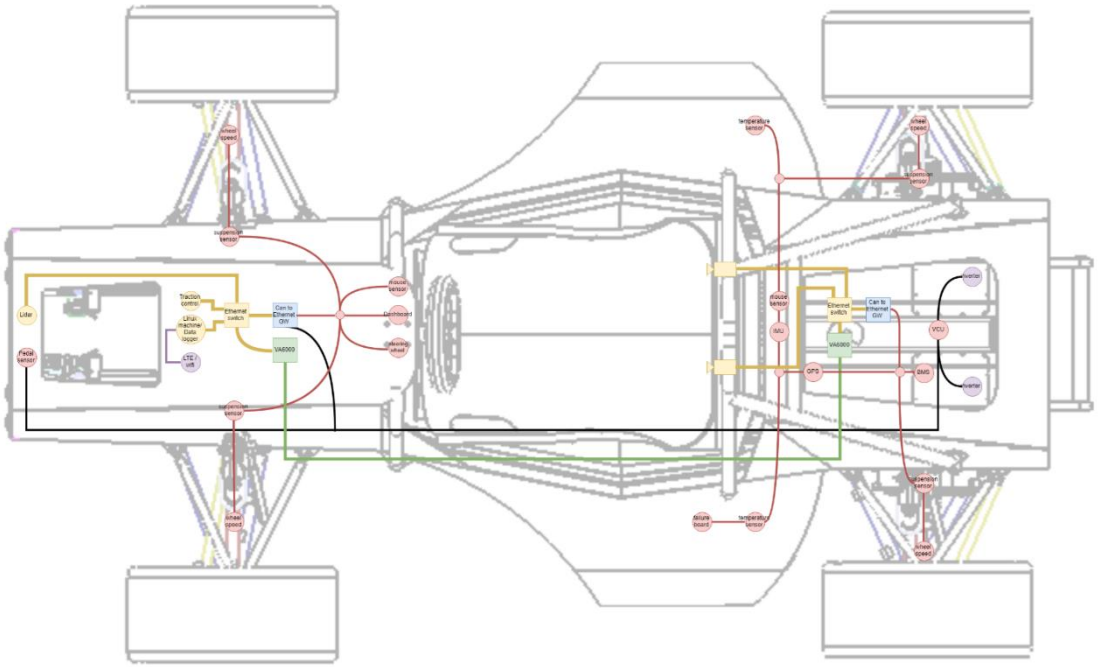
איור 9 - תרשים חוליות של ה-Gateway

בכדי שהמערכות האלקטרוניות ברכב יוכלו להיתמך ברכיבי ה-Gateway, קיים צורך להוסיף להן Framework ייעודי.

שיטת ניהול תעבורת המידע תקבע בהמשך.

3.1 מערכות הרכב האלקטרוניות BGR2019

באיור 9 ניתן לראות סכימה של המערכות האלקטרוניות ברכב.



איור 10 - סכימת המערכת

מערכת drive-by-wire

תפקיד המערכת הוא המרת זווית דוושת התאוצה לכמות המומנט שמספקים המנועים.

רכיבי המערכת:

- Pedal controller – רכיב אשר אחראי על דגימת מצב הדוושות ושליחת המידע באמצעות ממשק CAN-bus אל רשת הרכב.
- Velocity control unit – אחראי על מתן פקודת נסיעה למנועים. תדירות רענון פקודת הנסיעה הוא 200Hz. ה-VCU מצפה לקבל בכל חלון של 5msec שתי הודעות: הודעה אחת ישירה מבקר הדוושות והודעה נוספת מה-ECU. ה-VCU מוודא את תקינות ההודעות, ובהתאם למידע שקיבל נותן חיווי לבקרי המנועים. בחיווי דורש ה-VCU כמה מומנט על כל מנוע להפעיל על

הגלגלים. דרישת המומנט מתבצעת על ידי חיווי PWM.

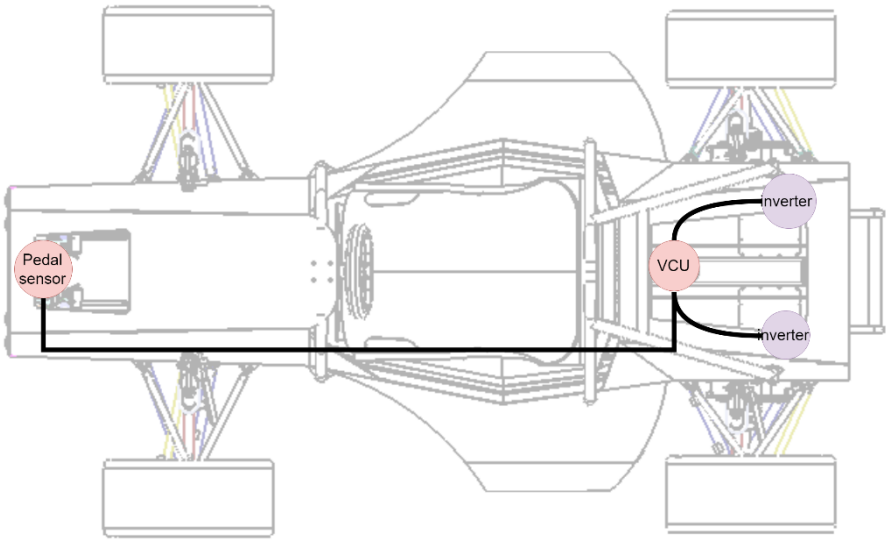
ה-VCU גם מתשאל את מחשבי המנועים בעזרת תקשורת טורית, Uart.ה-VCU מתשאל באופן מחזורי את המנועים לגבי ערכי הפעולה שלהם: צריכת זרם, מתח, טמפרטורה, סל"ד , ומפיץ את המידע ברשת. ה-VCU הוא הרכיב היחידי שיש לו הסמכות לתת חיווי למנועים.

ל-VCU 3 מצבים:

- Neutral – במצב זה הבקר מקבל מתח ומתקשר עם רשת הרכב, אך אינו נותן חיווי למנועים. דהיינו, לחיצה על דוושת התאוצה לא תפעיל את המנועים.
- Ready to drive – במצב זה הבקר דולק וגם מתאפשר מתן חיווי למנועים.
- Impossibility – הבקר דולק אך זוהתה תקלה ולכן לא תתאפשר שליטה על המנועים. במצב זה יפרסם הבקר הודעה ברשת הרכב, ו לחיצה על דוושת התאוצה לא תפעיל את המנועים.

הן ה-VCU והן ה-Pedal controller פותחו על ידי הקבוצה בכדי לענות על דרישות הרכב. שני הלוחות מובססים על בקרי Stm32F303RE.

בכדי להתאים את הבקרים לתפקידם הייעודי, ה-VCU ו-Pedal controller , ולסביבת הרכב נדרש תכנון תוספות חומרתיות ותכנותיות. סכימה של המערכת ורכיביה ניתן לראות באיור 10.



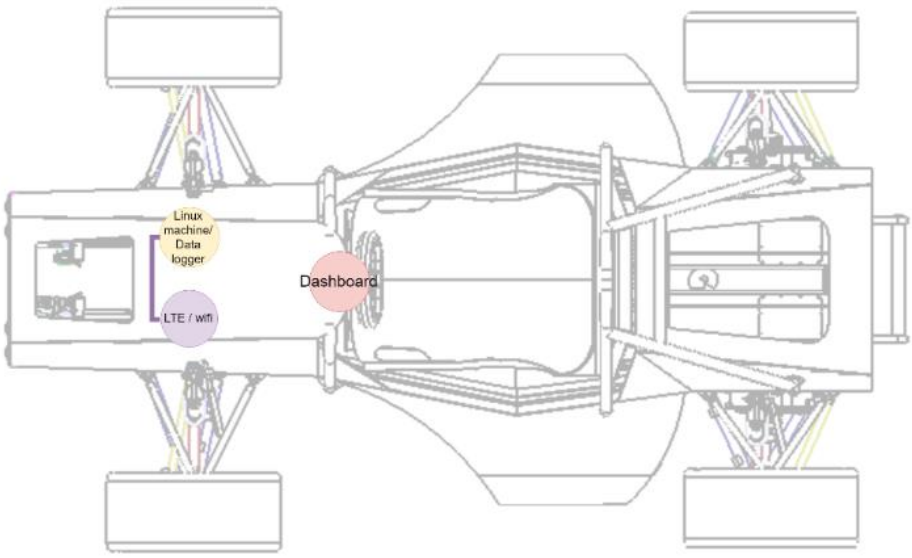
איור 11 - סכימת Drive-by-wire

Human machine interface

מטרת המערכת היא מתן חיווי לנהג ולצוות הרכב על מצב מערכות הרכב. המערכת מאפשרת לשקף על פעולת המערכות.

רכיבי המערכת :

- Dashboard – רכיב המציג חיוויים לנהג ומתריע בפני תקלות. ה-dashboard מקבל את המידע שלו באופן סיריאלי דרך רשת הרכב.
- Telemetria – מטרת ה-Telemetria היא לספק גישה קלה לכל חברי הקבוצה לכל המידע שנמצא ברכב, מבלי להתחבר אל הרכב באופן ישיר ואפילו מבלי להיות קרובים אליו פיזית. ל-telemetria שימושים הן במהלך עבודה על הרכב והן במהלך הנסיעה.ה-Telemetria מאפשרת לפקח אחר תקינות הרכב גם במהלך נסיעתו ובכך לאפשר לנהג להתמקד במלאכת הנהיגה. ה-Telemetria מתבצעת על ידי מימוש של שרת Http, אשר רץ על Rasppbery pi. אל ה-Raspberry pi נחבר דאנגל usb שמשדר WIFIוגם מתחבר אל הרשת הסלולארית. תוכנית השרת שרצה על ה-Raspberry pi פותחת אתר ברשת ה- WIFI המקומית, ה-raspberry pi מקבל את המידע על הרכב דרך ממשק ה-ethernet ומפיץ אותו החוצה באמצעות ה-WIFI.



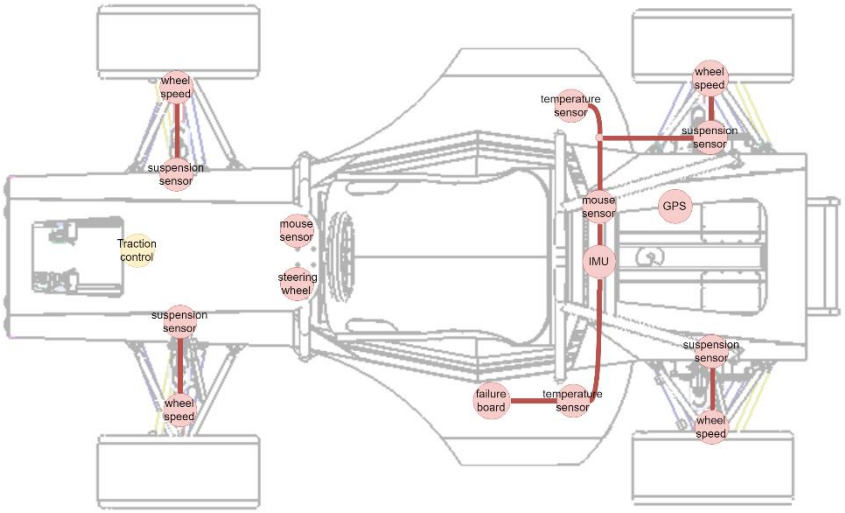
איור 12 - סכימת המערכת ממשק אדם מכונה

רשת החיישנים עבור מערכת ה-Traction control

רשת real time של חוליות CAN-bus המודדות את המצב הדינאמי של הרכב ושולחות את המידע לרשת. בקר ה-ECU מקבל את המידע, מעבד אותו ומספק ערך מומנט מקסימלי לכל מנוע.

רשת החיישנים בנויה מבקרי stm32f102t שמתקשרים באמצעות רשת CAN-bus. על הבקרים יש CAN-bus stack שפותח על ידי הקבוצה. הבקרים דוגמים את המידע ושולחים אותו לרשת באופן מחזורי. תדירות רענון ההודעות תלוי בתדירות החיישן המחובר אליהם.

באיור 12 מובאת סכימת המערכת ורכיביה

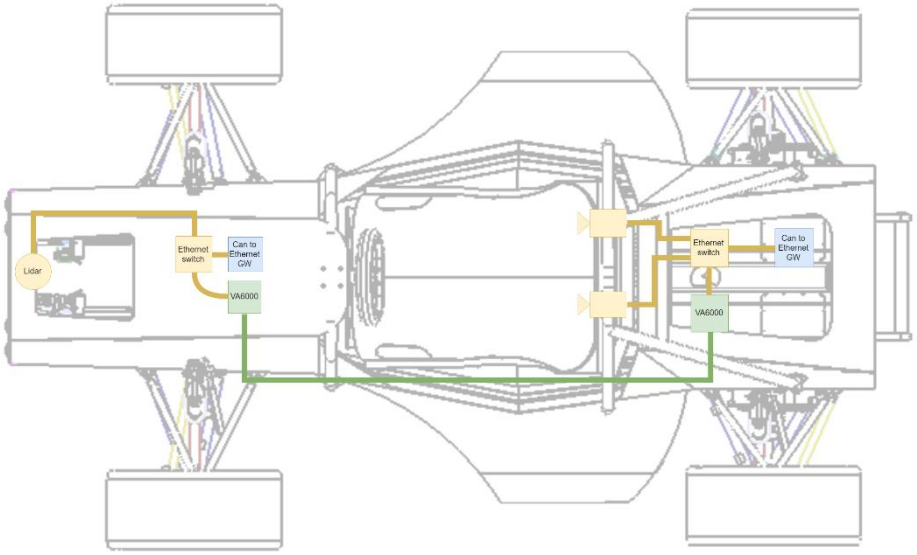


איור 13 - סכימת מערכת traction control

<u>תשתית התקשורת ותשתית למערכות ADS :</u>

המטרה היא הוספת backbone לרשת הרכב והתאמתו ליישומי ADS, הצורכים רוחב פס רב ביחס לרשתות ה-CAN-bus.

התשתית בנויה מ-2 Gateway nodes, אחד נמצא בחלקו הקדמי של הרכב והשני בחלקו האחורי, והם מחוברים באמצעות ממשק HDbaseT. אל רשת ה-ethernet נחבר מצלמות וחיישני Lidar.

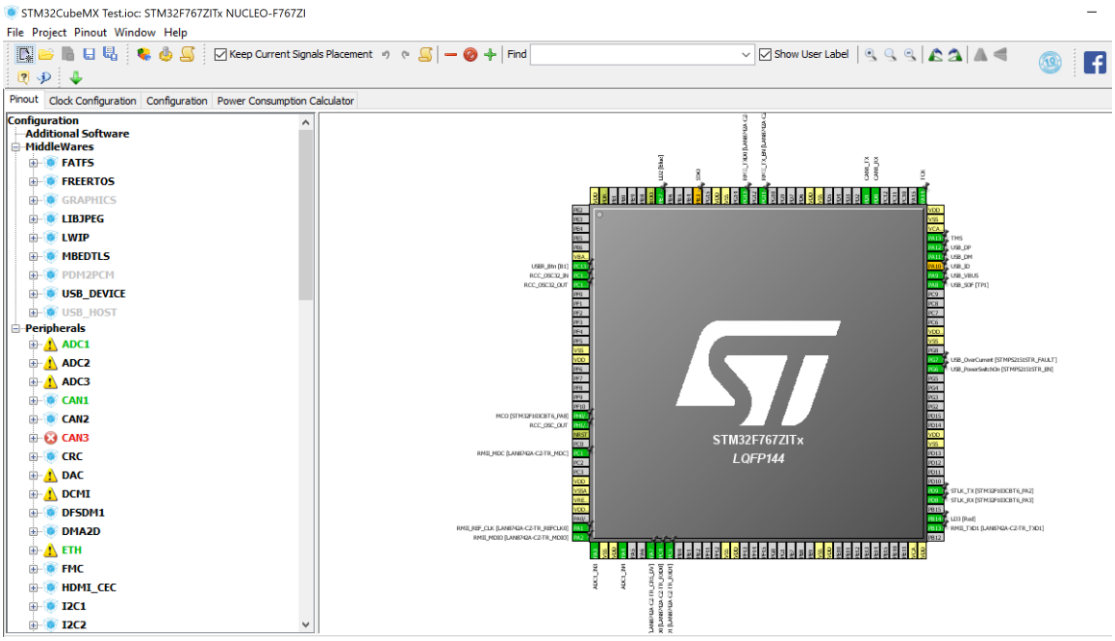


איור 14 - סכימת מערכת ADS

3.2 סביבת עבודה

- Hardware abstraction lyre – hal library – ספרייה מפורטת של STM32 שמאפשרת להשתמש בפונקציות מוכנות בכדי לגשת לחומרה.

- STM32CubeMX – סביבה המאפשרת להגדיר פרויקט Embedded בקלות ובמהירות. את ממשק המשתמש של התוכנה ניתן לראות באיור 14.

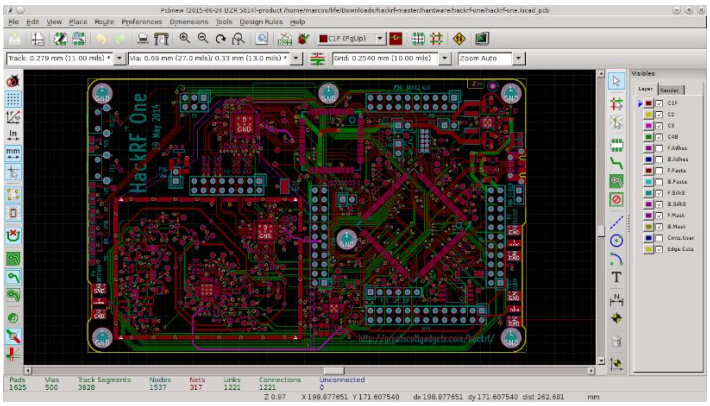


איור 15 - ממשק משתמש של STM32CubeMX

- FREERTOS – מערכת הפעלה מינימאלית מיועדת לבקרים.

- LwIP TCP/IP stack – middleware המאפשר בניית חבילות TCP/IP.

- Kicad EDA – תוכנה חופשית המשמשת לתכנון מעגלים מודפסים . צילום של ממשק המשתמש של המערכת ניתן לראות באיור 15.



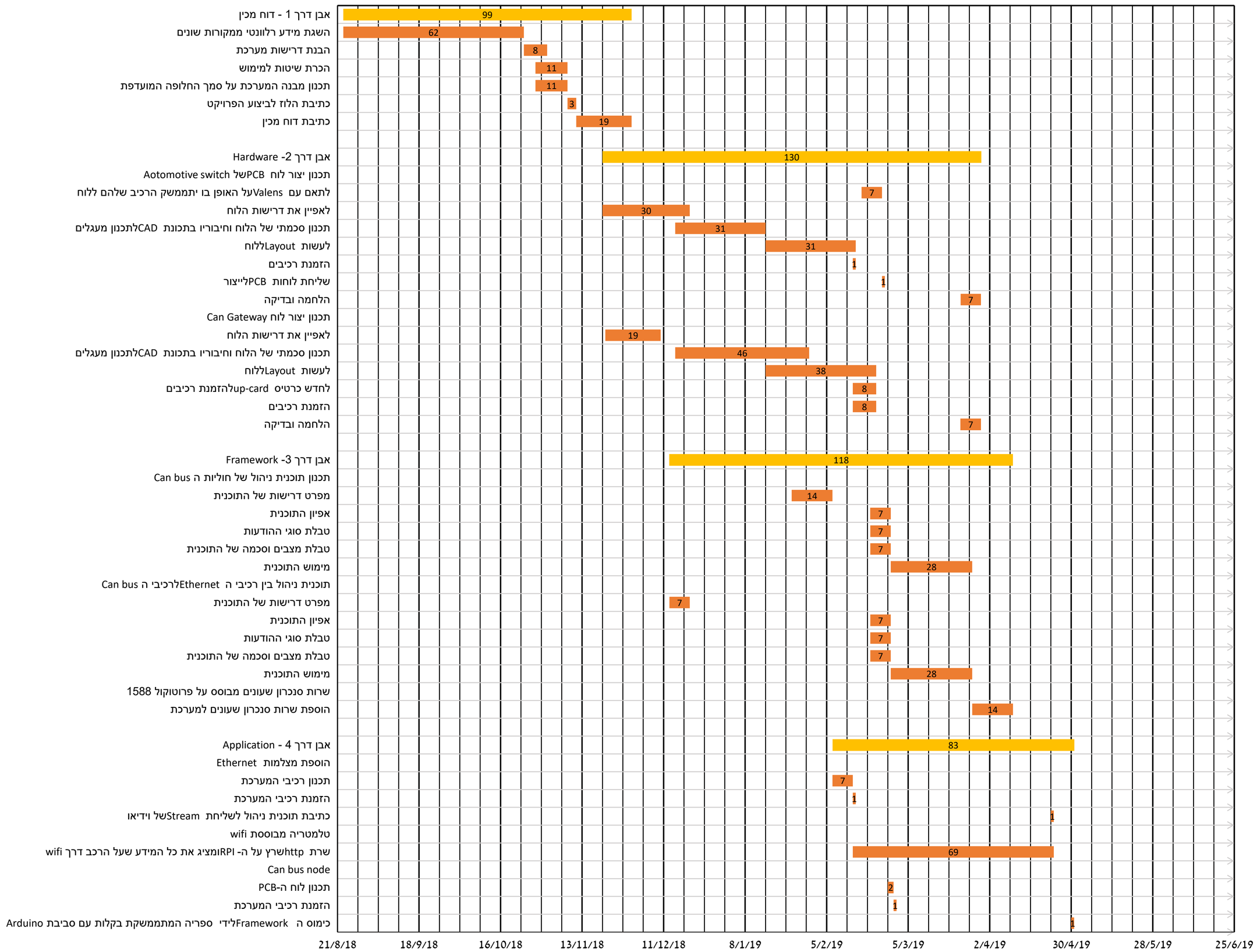
איור 16 - ממשק מערכת Kicad

4 שלבים לביצוע הפרויקט

End Date	Duration	Start Date	Tasks
30/11/18	99	23/8/18	אבן דרך 1 - דוח מכין
24/10/18	62	23/8/18	השגת מידע רלוונטי ממקורות שונים
1/11/18	8	24/10/18	הבנת דרישות מערכת
8/11/18	11	28/10/18	הכרת שיטות למימוש
8/11/18	11	28/10/18	תכנון מבנה המערכת על סמך החלופה המועדפת
11/11/18	3	8/11/18	כתיבת הלוז לביצוע הפרויקט
30/11/18	19	11/11/18	כתיבת דוח מכין
	0		
30/3/19	130	20/11/18	אבן דרך 2- Hardware
	0		תכנון יצור לוח PCB של Aotomotive switch
24/2/19	7	17/2/19	לתאם עם Valens על האופן בו יתממשק הרכיב שלהם ללוח
20/12/18	30	20/11/18	לאפיין את דרישות הלוח
15/1/19	31	15/12/18	תכנון סכמתי של הלוח וחיבוריו בתכונת CAD לתכנון מעגלים
15/2/19	31	15/1/19	לעשות Layout ללוח
15/2/19	1	14/2/19	הזמנת רכיבים
25/2/19	1	24/2/19	שליחת לוחות PCB לייצור
30/3/19	7	23/3/19	הלחמה ובדיקה
	0		תכנון יצור לוח Can Gateway
10/12/18	19	21/11/18	לאפיין את דרישות הלוח
30/1/19	46	15/12/18	תכנון סכמתי של הלוח וחיבוריו בתכונת CAD לתכנון מעגלים
22/2/19	38	15/1/19	לעשות Layout ללוח
22/2/19	8	14/2/19	לחדש כרטיס up-card להזמנת רכיבים
22/2/19	8	14/2/19	הזמנת רכיבים
30/3/19	7	23/3/19	הלחמה ובדיקה
	0		
10/4/19	118	13/12/18	אבן דרך 3- Framework
	0		תכנון תוכנית ניהול של חוליות ה Can bus
7/2/19	14	24/1/19	מפרט דרישות של התוכנית
27/2/19	7	20/2/19	אפיון התוכנית
27/2/19	7	20/2/19	טבלת סוגי ההודעות
27/2/19	7	20/2/19	טבלת מצבים וסכמה של התוכנית
27/3/19	28	27/2/19	מימוש התוכנית
	0		תוכנית ניהול בין רכיבי ה Ethernet לרכיבי ה Can bus
20/12/18	7	13/12/18	מפרט דרישות של התוכנית
27/2/19	7	20/2/19	אפיון התוכנית
27/2/19	7	20/2/19	טבלת סוגי ההודעות
27/2/19	7	20/2/19	טבלת מצבים וסכמה של התוכנית
27/3/19	28	27/2/19	מימוש התוכנית
	0		שרות סנכרון שעונים מבוסס על פרוטוקול 1588

הוספת שרות סנכרון שעונים למערכת	27/3/19	14	10/4/19
		0	
אבן דרך 4 - Application	7/2/19	83	1/5/19
הוספת מצלמות Ethernet		0	
תכנון רכיבי המערכת	7/2/19	7	14/2/19
הזמנת רכיבי המערכת	14/2/19	1	15/2/19
כתיבת תוכנית ניהול לשליחת Stream של וידיאו	23/4/19	1	24/4/19
טלמטריה מבוסס wifi		0	
שרת http שרץ על ה-RPI ומציג את כל המידע שעל הרכב דרך wifi	14/2/19	69	24/4/19
Can bus node		0	
תכנון לוח ה-PCB	26/2/19	2	28/2/19
הזמנת רכיבי המערכת	28/2/19	1	1/3/19
כימוס ה Framework לידי ספריה המתממשקת בקלות עם סביבת Arduino	30/4/19	1	1/5/19

תרשים גאנט


אבן דרך 1 - דוח מכין
99
השגת מידע רלוונטי ממקורות שונים
62
הבנת דרישות מערכת
8
הכרת שיטות למימוש
11
תכנון מבנה המערכת על סמך החלופה המועדפת
11
כתיבת הלוז לביצוע הפרויקט
3
כתיבת דוח מכין
19
אבן דרך 2- Hardware
130
תכנון יצור לוח PCBשל Aotomotive switch
לתאם עם Valensעל האופן בו יתממשק הרכיב שלהם ללוח
7
לאפיין את דרישות הלוח
30
תכנון סכמתי של הלוח וחיבוריו בתכונת CADלתכנון מעגלים
31
לעשות Layoutללוח
31
הזמנת רכיבים
1
שליחת לוחות PCBלייצור
1
הלחמה ובדיקה
7
תכנון יצור לוח Can Gateway
לאפיין את דרישות הלוח
19
תכנון סכמתי של הלוח וחיבוריו בתכונת CADלתכנון מעגלים
46
לעשות Layoutללוח
38
לחדש כרטיס up-cardלהזמנת רכיבים
8
הזמנת רכיבים
8
הלחמה ובדיקה
7
אבן דרך 3- Framework
118
תכנון תוכנית ניהול של חוליות ה Can bus
מפרט דרישות של התוכנית
14
אפיון התוכנית
7
טבלת סוגי ההודעות
7
טבלת מצבים וסכמה של התוכנית
7
מימוש התוכנית
28
תוכנית ניהול בין רכיבי ה Ethernetלרכיבי ה Can bus
מפרט דרישות של התוכנית
7
אפיון התוכנית
7
טבלת סוגי ההודעות
7
טבלת מצבים וסכמה של התוכנית
7
מימוש התוכנית
28
שרות סנכרון שעונים מבוסס על פרוטוקול 1588
הוספת שרות סנכרון שעונים למערכת
14
אבן דרך 4 - Application
83
הוספת מצלמות Ethernet
תכנון רכיבי המערכת
7
הזמנת רכיבי המערכת
1
כתיבת תוכנית ניהול לשליחת Streamשל וידיאו
1
טלמטריה מבוססת wifi
שרת http שרץ על ה- RPIומציג את כל המידע שעל הרכב דרך wifi
69
Can bus node
תכנון לוח ה-PCB
2
הזמנת רכיבי המערכת
1
כימוס ה Frameworkלידי ספריה המתממשקת בקלות עם סביבת Arduino
1
21/8/18
18/9/18
16/10/18
13/11/18
11/12/18
8/1/19
5/2/19
5/3/19
2/4/19
30/4/19
28/5/19
25/6/19

4.2 יומן פרויקט

	טבלה 3:	יומן פעילות הסטודנטים להגשמת המשימות לטובת הגשמת אבן הדרך				
	תאריך	שם הסטודנט המבצע	שם המשימה	**תאור הפעולה**	כמות שעות	הערות
1	13/9	גיא	מפגש עם חן			
2	13/9	גיא	עבודה בסדנה על המודל הפרוש			
3	16/9	גיא	מפגש עם Valens			
4	20/9	גיא	למידה על תכנון לוחות PCB			
5	27/9	גיא	עבודה על המודל הפרוש			
6	3/10	גיא	מפגש עם חן			
7	4/10	גיא	יום עבודה לפרויקט			
8	11/10	גיא	יום עבודה לפרויקט			
9	17/10	גיא	מפגש עם חן			
10	18/10	גיא	יום עבודה לפרויקט			
11	25/10	גיא	יום עבודה לפרויקט			
12	31/10	גיא	מפגש עם חן			
13	1/11	גיא	מפגש עם מוטי			
14	8/11	גיא	הכנת המודל הפרוש			
15	8/11	גיא	יום עבודה לפרויקט			
16	14/11	גיא	מפגש עם חן			
17	14/11	גיא	ישיבה על התוכנית הלוגית			
18	15/11	גיא	יום עבודה לפרויקט			
19	21/11	גיא	עבודה על שליחת הודעות CAN			
20	22/11	גיא	יום עבודה לפרויקט			
21	29/11	גיא	מפגש עם חן			
22	29/11	גיא	יום עבודה לפרויקט			
23	11/12	גיא	מפגש עם RTI			
24	12/12	גיא	מפגש עם חן			
25	24/12	גיא	עבודה בסדנה מיון וסידור חדר חשמל	סידור סביבת העבודה ומיון כל הרכיבים בחדר חשמל לטובת עדכון המלאי לקראת ביצוע הזמנה	19:00 - 1:00	
26	26/12	גיא	מפגש עם חן			

27	27/12	גיא	קריאת מאמרים בנושא מודלים מתמטיים במערכות real time	על מנת לתת נפח לפרויקט הגמר , אני בודק את האפשרות להוסיף שיטות לתת למערכת מודל מתמטי	8:00 - 9:00	
28	28/12	גיא	ביצוע layout ללוח pcb של ה switch		23:00 - 6:00	
29	29/12	גיא	ביצוע layout ללוח pcb של ה switch		22:00 - 4:00	
30	30/12	גיא	מפגש עם RTI			
31	13/1	גיא	מפגש עם גלעד מ Valnse לגבי אינטגרציה	מפגש עם גלעד מ Valnse לגבי אינטגרציה		
32	25/1	גיא	עבודה על לוחות ה PCB			
33	26/1	גיא	עבודה על לוחות ה PCB			
34	2/4	גיא	תדרוך על Layout אצל סתיו סטודנט שנה ד חשמל			
35	2/5	גיא	איפיון ה Framework יחד עם אבישי חבר קבוצה			
36	2/6	גיא	איפיון ה Framework יחד עם אבישי חבר קבוצה			

5 שלבים במימוש המערכת

בחלק הראשון של הפרויקט נבחנו שיטות התקשורת המקובלות ברכבים, התבצע איפיון של צורכי התקשורת ברכב מרוץ אוטונומי FSAE, והוצעה שיטה לארגון מחדש של שיטת התקשורת, השיטה נסתמכה על רכיב תקשורת יעודי שייגשר בין רשתות ה- can bus לבין רשתות ה Ethernet , החלק השני של הפרויקט עסק במימוש רכיב התקשורת היעודי והתמעתו ברכב.

חלק זה עסק בתהליך הכרוך בפיתוח חומרת תקשורת יעודית, החל מבחירת רכיבי הלוח, עריכת המעגל המודפס , ממשק מול ספקים ויצרני חומרה, הלחמת הלוח בידקתו ופיתוח תוכנה.

5.1 שיקולים לבחירת החומרה

בחירת החומרה נעשתה כך שמצד אחד נבחרו הרכיבים המתאימים ביותר עבור המערכת , ומצד שני נעשתה הערכה לגבי מגבלות יכולת הייצור שכן שיטת הייצור שנבחרה היא הלחמה בעזרת ציוד בסיסי.

על כן החולט שכל סט הצ'יפים יהיה ממשפחת ה-QFP (הצ'יפים המרובעים עם הרגליים), והרכיבים הפאסיבים יהיו בגודל 0603 (זאת אומרת 0.6 על 0.3 אינץ),

בכדי למנוע הרפתקאות מיותרות, על כל הרכיבים להיות זמינים במלאי של הספקית digikey , על מנת לפשט את תהליך הרכש.

בנוסף כל הרכיבים שנבחרו לפרויקט הם רכיבים שהיצרן מספק את כל המידע הנחוץ לשיבוץ הרכיב בלוח. תהליך האינטגרציה נסמך על המידע שסיפק היצרן בגליונות הנתונים. לכן, למידע זה ישנה חשיבות ממעלה ראשונה לסבירות הצלחת הפרויקט.

5.2 שלבים בתכנון לוח PCB

התוצר הסופי בתכנון לוחות PCB הוא הקבצי ה Gerber קבצים אלו מתארים לייצרן כיצד לייצר את לוחות ה PCB , את הקבצים הללו נתן להכין בעזרת תוכנות CAD יעודיות לתכנון מעגלים , מהלך העבדוה בתוכנות אלו הוא כדקלמן

1. שרטוט סכמתי של הלוח
2. מתן צורה פיזית שנקראת footprint לכול רכיב בסכמה
3. ייצור קובץ netlist ויבואו לסביבת ה layout מתוך השרטוט הסכמתי
4. בסביבת ה-layout מתבצע השרטוט הפיזי של הלוח
    a. ממקמים את footprint של הרכיבים על הלוח

b. מחברים את הרכיבים אחד לשני לפי התכנון הסכמתי באמצעות trace-ים.
c. בודקים שהלוח מקיים את תנאי ה Design rules
d. מייצרים קבצי Gerber על סמך שרטוט ה layout.

בפרויקט זה נעשה שימוש בתוכנת Kicad לעריכת המעגלים, נעשה שימוש בתוכנת Quacs studio לחישוב עובי הנתיבים הנחוץ ולתיאום אימפידאנס, הלחות יוצרו אצל היצרן הסיני JLPCB .

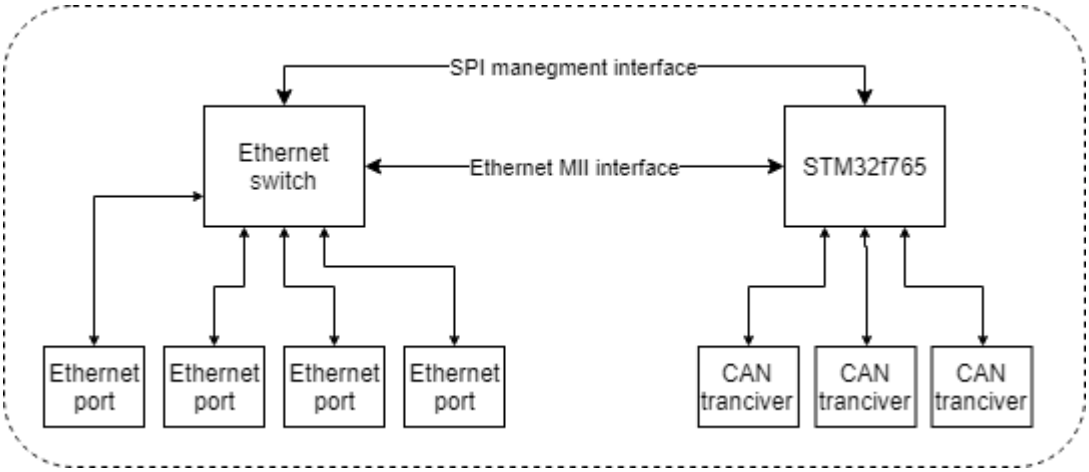
5.3 מעבר כללי על הלוח ה Gateway

בכדי לענות על דרישות המערכת שתוארו בשלב השיטה נבחרו שני הרכיבים הבאים לביצוע המשימה:

- KSZ8567 – אחראי לניהול ה-switch.
- stm32f76 – אחראי על ממשק ה-CAN-bus, ניהול ה-switch וניהול הלוגיקה של הלוח.

תהליך עריכת המעגל מורכב מהשלבים הבאים :

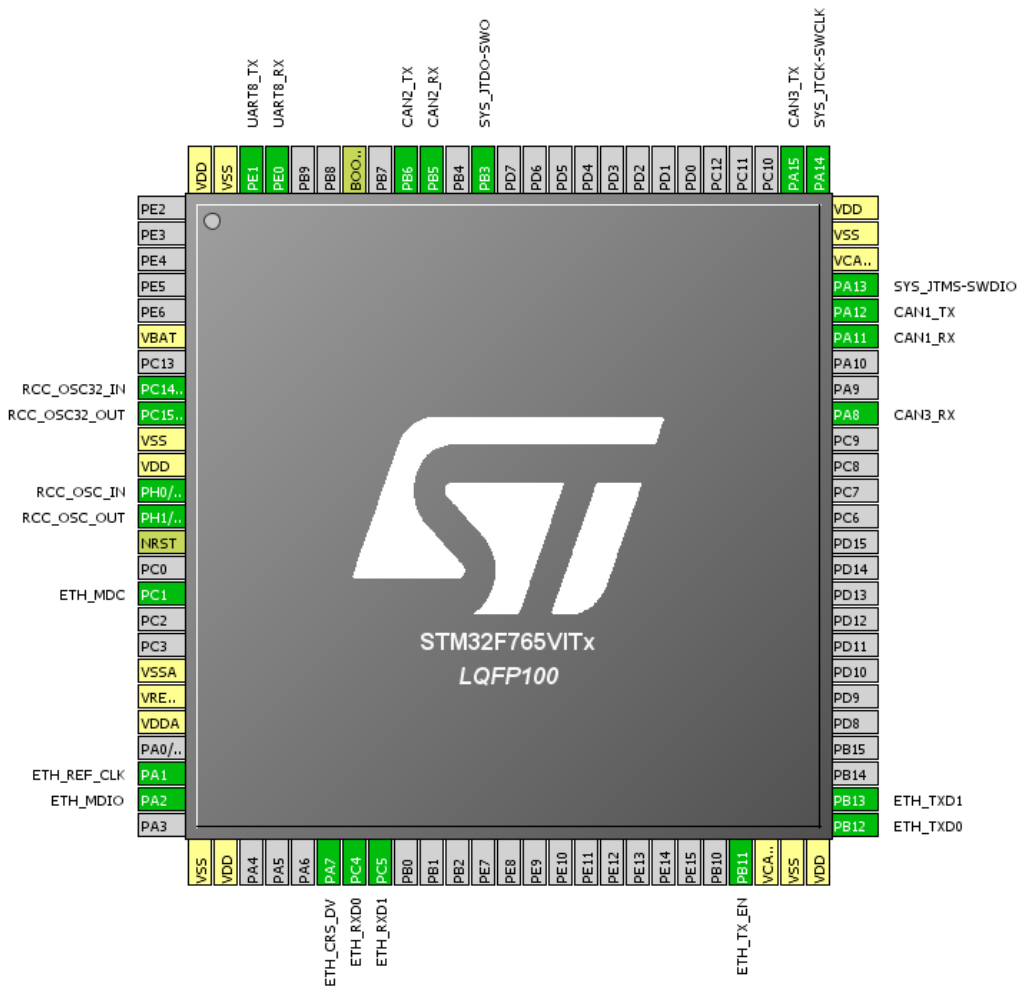
1. אספקת מקורות המתח הנחוצים לכל רכיב לפי מפרט הוראות היצרן
2. חיבור ממשקי התקשורת ותיאום האימפידאנס האופייני:
   I. Ethernet PHY בין ה-Ethernet switch לבין ה-RJ45
   II. חיבור MII בין ה-Ethernet switch לבין הבקר Stm32f765
   III. חיבור בין בקר ה-stm32f765 לבין ה-CAN transceiver
   IV. חיבור SPI המשמש לניהול ערכי ה-switch באמצעות ה-stm32f765



איור 17 - סכימת בלוקים של לוח ה gateway

5.4 רכיבי ה-IC בלוח

- stm32F765 – מיקרו בקר בארכיטקטורת ARM עם קצב עבודה מקסימלי של 216 MHz ובעל זכרון flash של 2MB. תומך במגוון רכיבים חומרתיים, כגון Ethernet controller ו-CAN-bus controller.



איור 18 - תרשים של בקר stm32f765 והרגליים שבשימוש בלוח

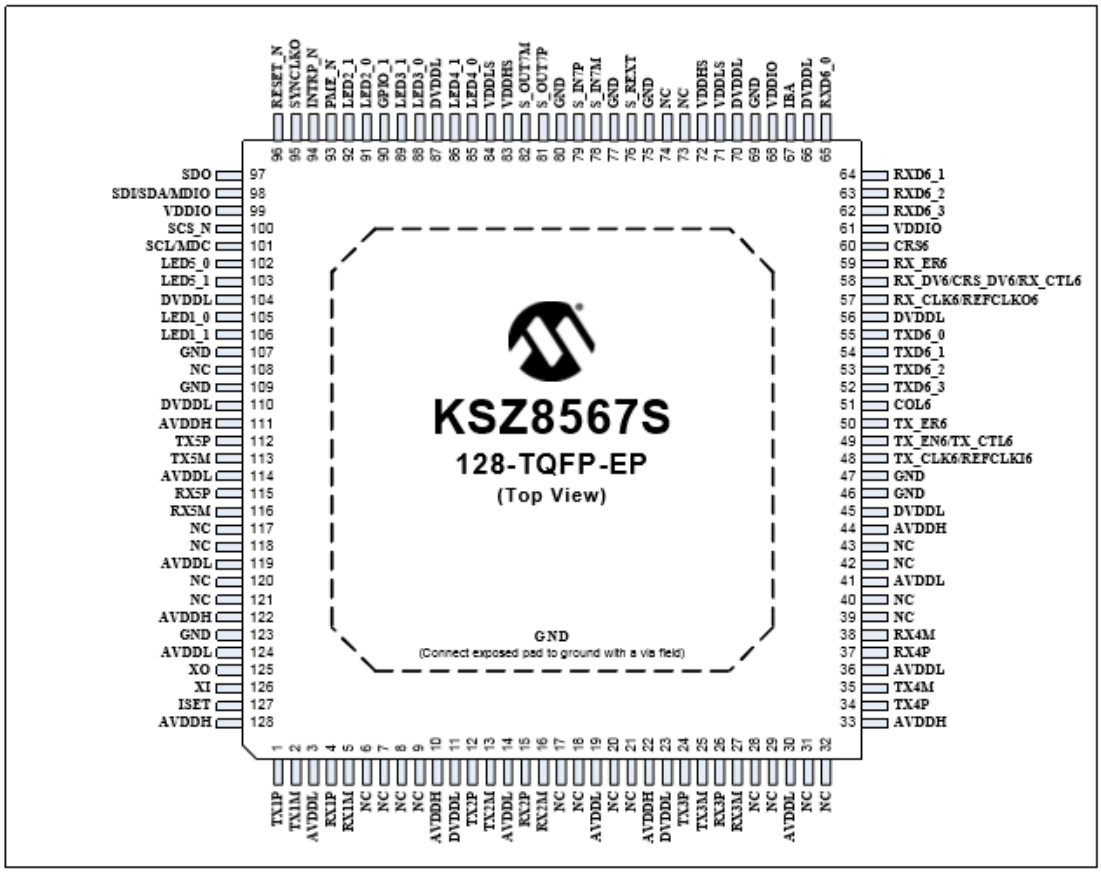
- Management Ethernet Switch KSZ8567STXI-TR –

AVB (Audio Video Bridging) switch בעל 7 כניסות Ethernet של 10/100 BaseT, עם ממשקי SGMII ו-RMII/MII. ב-[12] ניתן לראות את ה-datasheet של הרכיב. הרכיב נמצא מתאים משום ש-

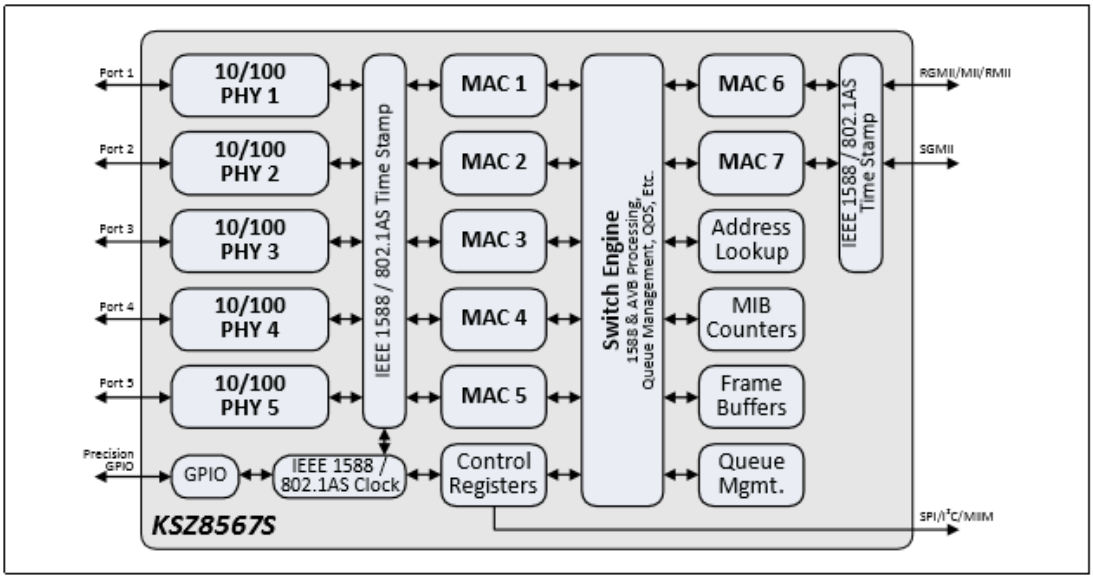
I. יש לו תמיכה בלפחות ארבעה ממשקי Ethernet.

II. יש לו ממשק SGMII אשר מאפשר לו לנתב מידע החוצה אל PHY ייעודי, למשל V6000 של חברת Valens, בתעבורת מידע העולה על 1 GB/SEC.

III. הרכיב תומך בטכנולוגיות ייעודיות להעברת זרמי נתונים וסנכרון שעונים ומשווק כפתרון טוב לתעבורת וידאו ברכבים עבור מערכות ADAS.

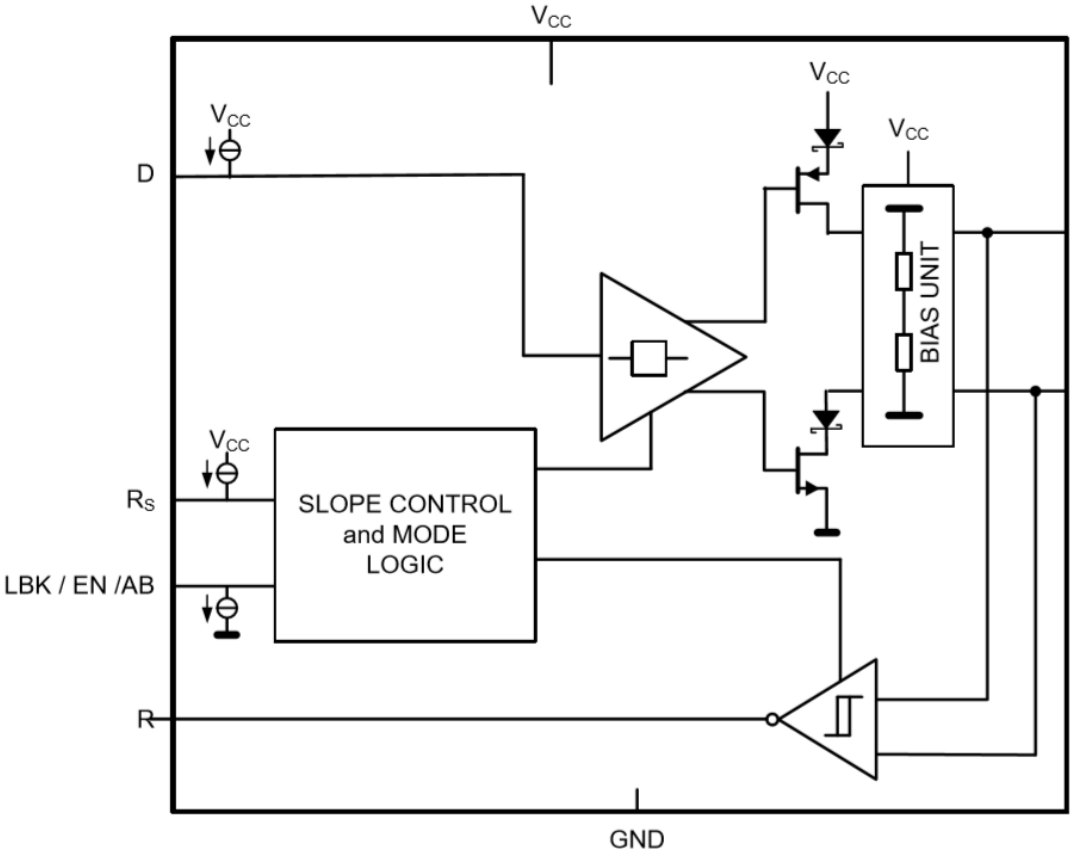


איור 19 - תרשים pinout של ה-Ethernet switch



איור 20 - תרשים בלוקים של המודולים מהם מורכב ה-Ethernet switch

- CAN-bus transceiver SN65HVD234 – רכיב מאפנן את אות ה-single ended הבינארי שנשלח מה-STM ל-differential הנשלח לערוץ ה-CAN-bus ולהפך. כמו כן הרכיב מבצע הגברה לאות. ב-[13] תוכלו למצוא את ה-datasheet של הרכיב.

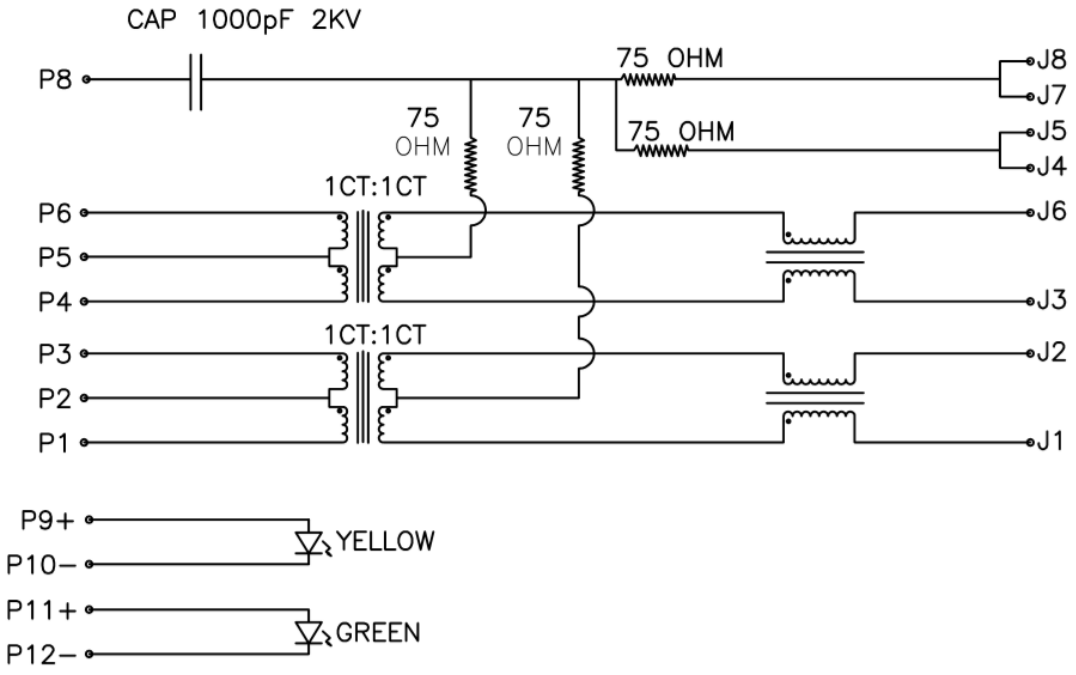


איור 21 - סכימה חשמלית של ה-CAN-bus transceiver

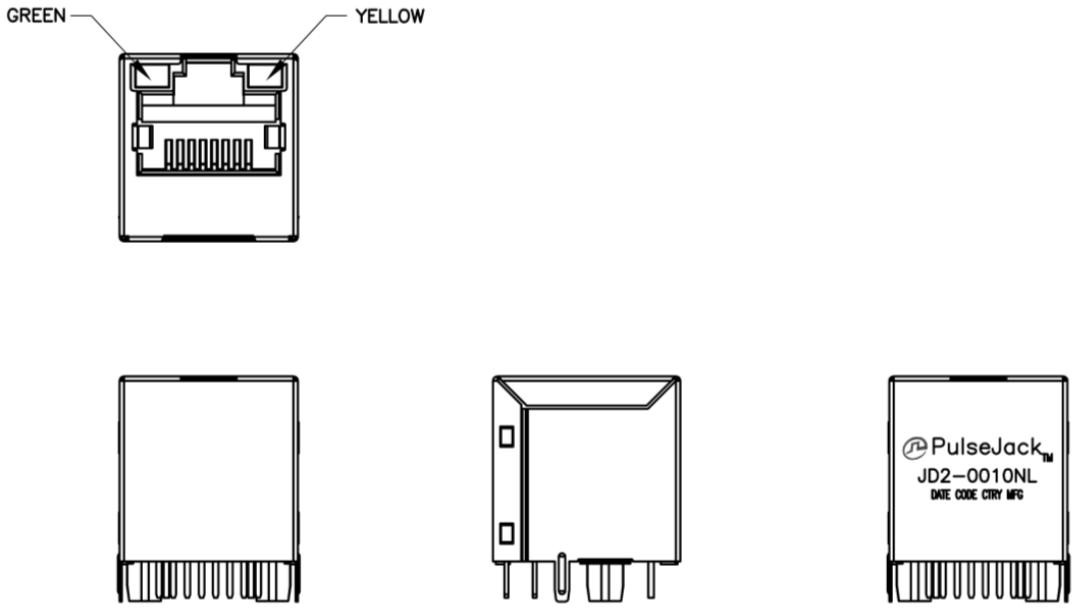
- RJ45 with integrated transformer – עבור בחירת המחבר וה-transformer לממשק ה-Ethernet הייתה התלבטות בין 2 גישות:
  - בגישה הראשונה נבחנה האפשרות לחפש אחר מחבר שיתן מענה היעמוד בקריטריונים המחימירים של תחרות FSAE הדורש עמידות בסטנדרט של-IP68
  - הגישה השנייה בחירה במחבר RJ45, באה בעקבות חשש שאם החיבור יהיה לא סטנדרטי חברי הקבוצה ימנעו מלהשתמש במערכת.

שתי הגישות נבחנו ונמצא כי העלות הכספית של הגישה הראשונה יקרה משמעותית, ולכן בשביל לא לסכן את הצלחת הפרויקט, נבחר ללכת עם הגישה השניה ולשים מחברים שאינם עונים לגמרי על דרישות המערכת, אך מתאימים לבחינת המערכת ולהתקנתה עבור ביצוע פעולות שאינן קריטיות לתפקוד הרכב.

ב-[14] ניתן למצוא את ה-datasheet של הרכיב.



איור 22 - סכימה חשמלית של המעגל ב-RJ45



איור 23 - שרטוט מכאני של RJ45

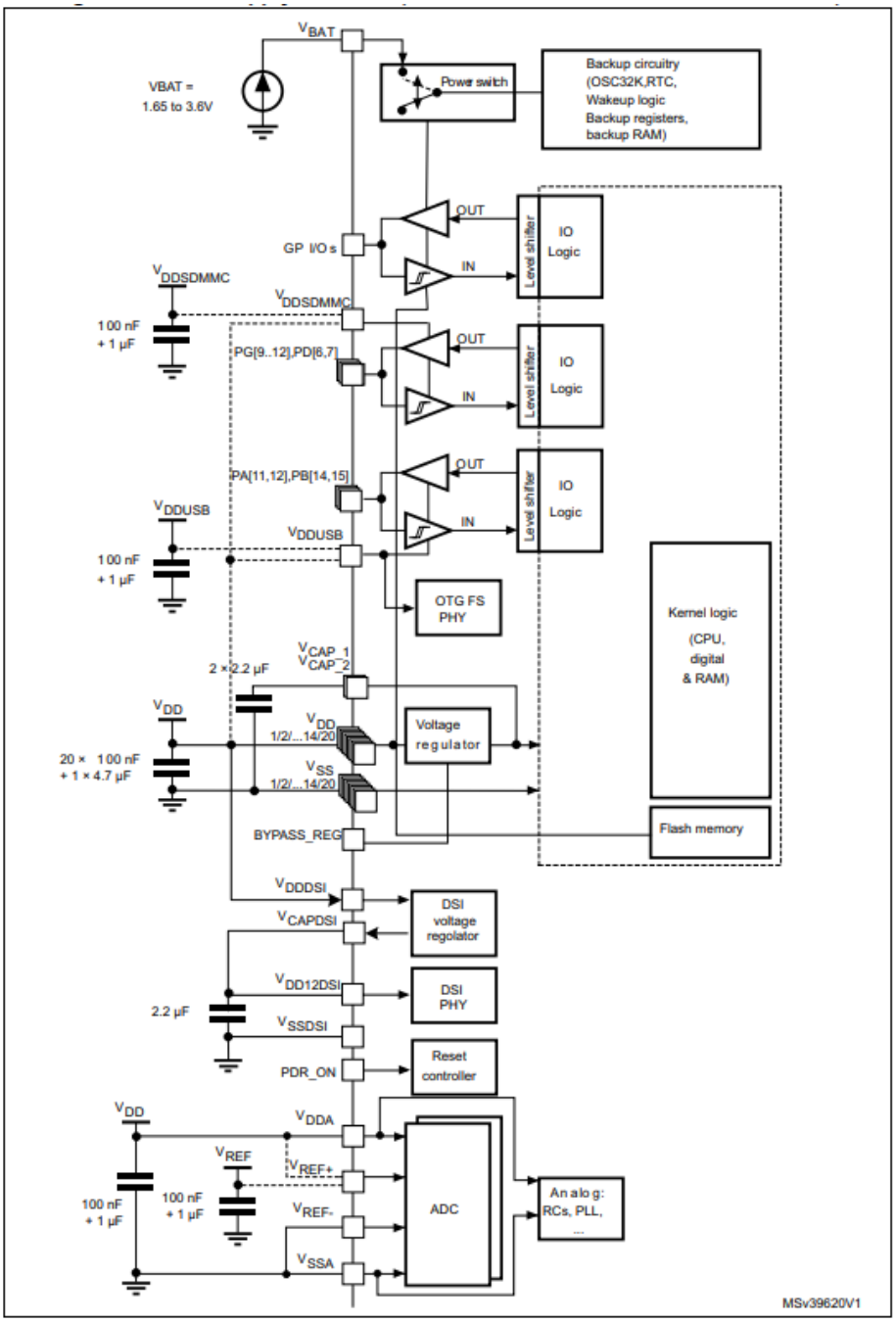
5.5 מעגלים מרכזיים בלוח

בחלק זה ישנה סקירה על העיקרי הסכימה וכיצד מתבצע התהליך של שיבוץ החומרה בעזרת המידע הנמצא ב-datasheet.

5.5.1 Stm32F765

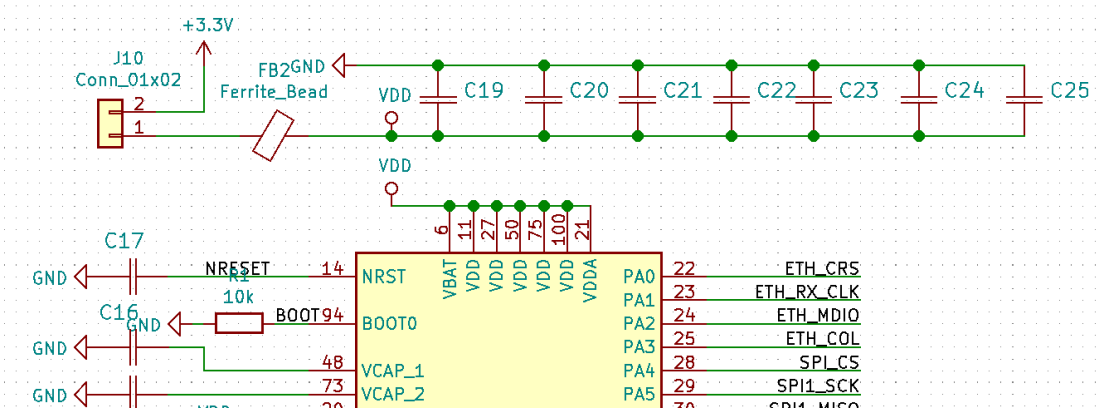
השלב הראשון בשיבוץ הרכיב הוא לספק לו את מקורות המתח הנחוצים באופן שבו ממליץ היצרן.

במסמך ה-datasheet [11] של הרכיב תחת סעיף power קיימת סכימה המתארת זאת.



איור 24 - סכימת application note לגבי המלצת היצרן לאספקת מתח ללוח

לפי הוראות היצרן, נשרטט בתוכנת ה-cad את המקורות הנחוצים.



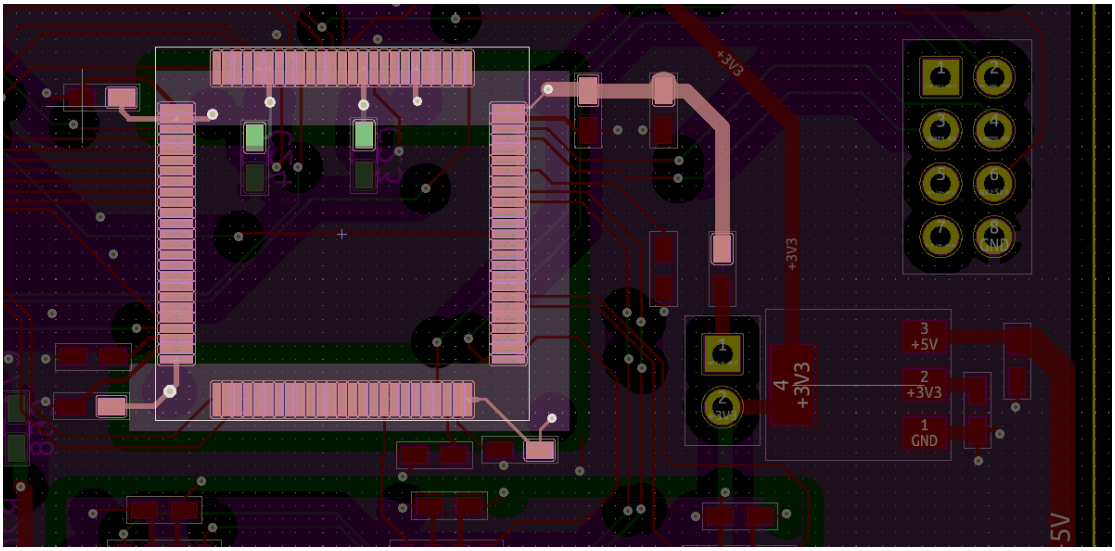
איור 25 - סכימה חשמלית של אספקת המתח ל-stm32 במעגל שלנו

בלוח הזה, מכיוון שמקור המתח 3.3 וולט משותף גם לבקר, גם ל-transceiver של ה-CAN-bus וגם ל-IOs של ה-Ethernet switch, נוסיף ferrite bead עבור כל רשת כדי לבצע הפרדה של המקור לשלושה תתי מקורות, ובכך להפחית את ההפרעות של רכיב אחד על השני.

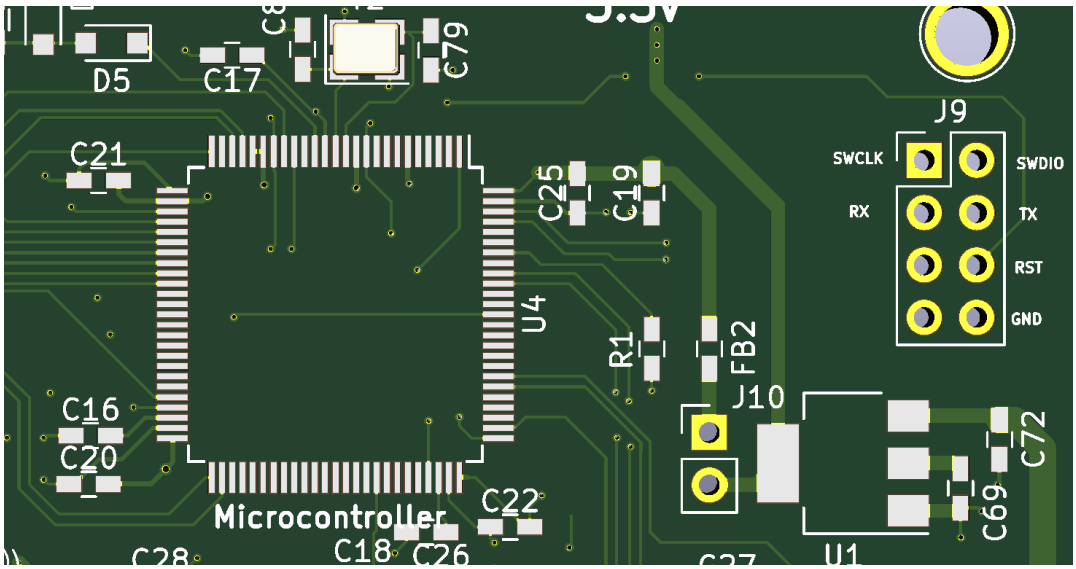
לגבי ה-layout, את קבלי ה-bypass ממקמים כמה שיותר קרוב לטרמינלים של הבקר.

השכבה הראשונה היא שכבת mixed power and signal, והשכבה השנייה היא שכבת gnd. השכבה השלישית היא שכבה שנועדה להעביר Power ואותות בעלי תעבורת מידע איטית. מכאן שמימוש הרגולטור שמספק 3.3 וולט נמצא על השכבה הראשונה יחד עם המיקרובקר. הקבלים נמצאים על השכבה הראשונה והרביעית באופן אקראי, ובשכבה השלישית עובר נתיב המעביר את רשת ה-VDD מתחת לבקר. נשים לב שחשוב מאוד להימנע מכל פגיעה ברציפות השכבה השנייה, שכן כל מודל חישוב האימפידנאסים שלנו הוא תחת ההנחה כי קיים GND מתחת לכל קו שמעביר אות. תהליך זהה נעשה גם עבור שאר הרכיבים.

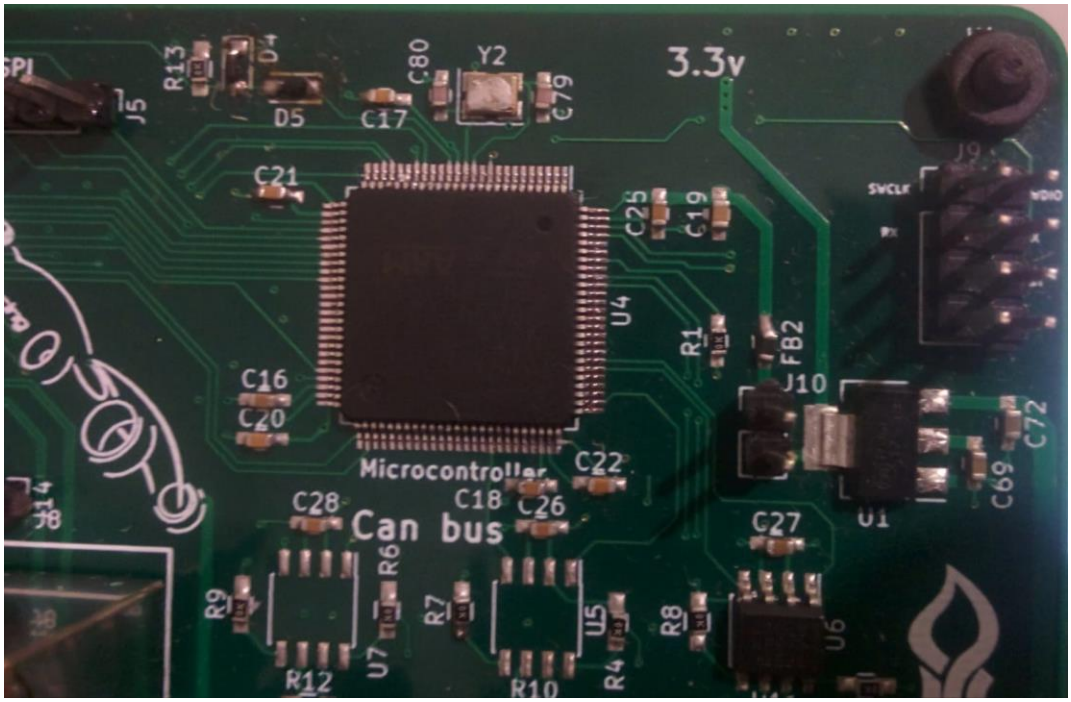
בשונה משאר הרכיבים אספקת המתחים ל-Ethernet switch הייתה משמעותית מורכבת יותר זאת משום של-switch יש עשרות טרמינלים הדורשים מתחים, 3 סוגי מתח שונים, 3.3, 2.5 ו-1.2 וולט, ו-6 רשתות מתח שונות. מצורפת תמונה להמחשה ראו איור31.



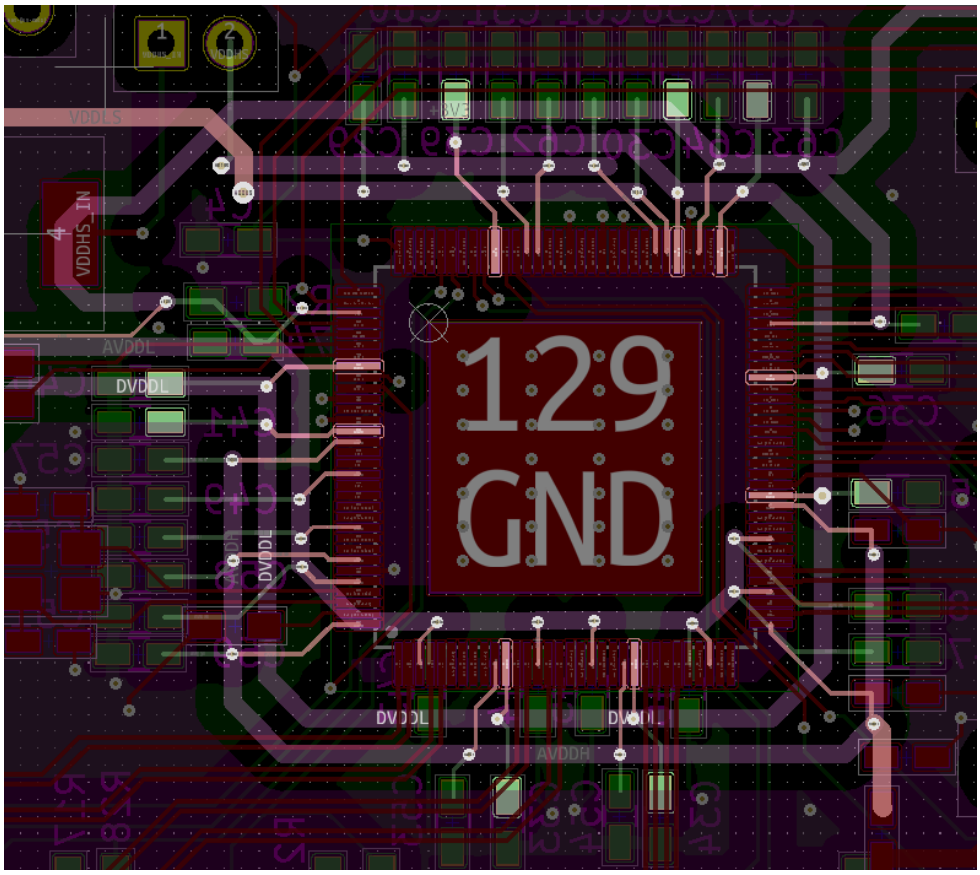
איור 26 - מימוש המעגל ב-layout



איור 27 - הדמיה של מימוש המעגל



איור 28 - תמונה של המעגל בפועל



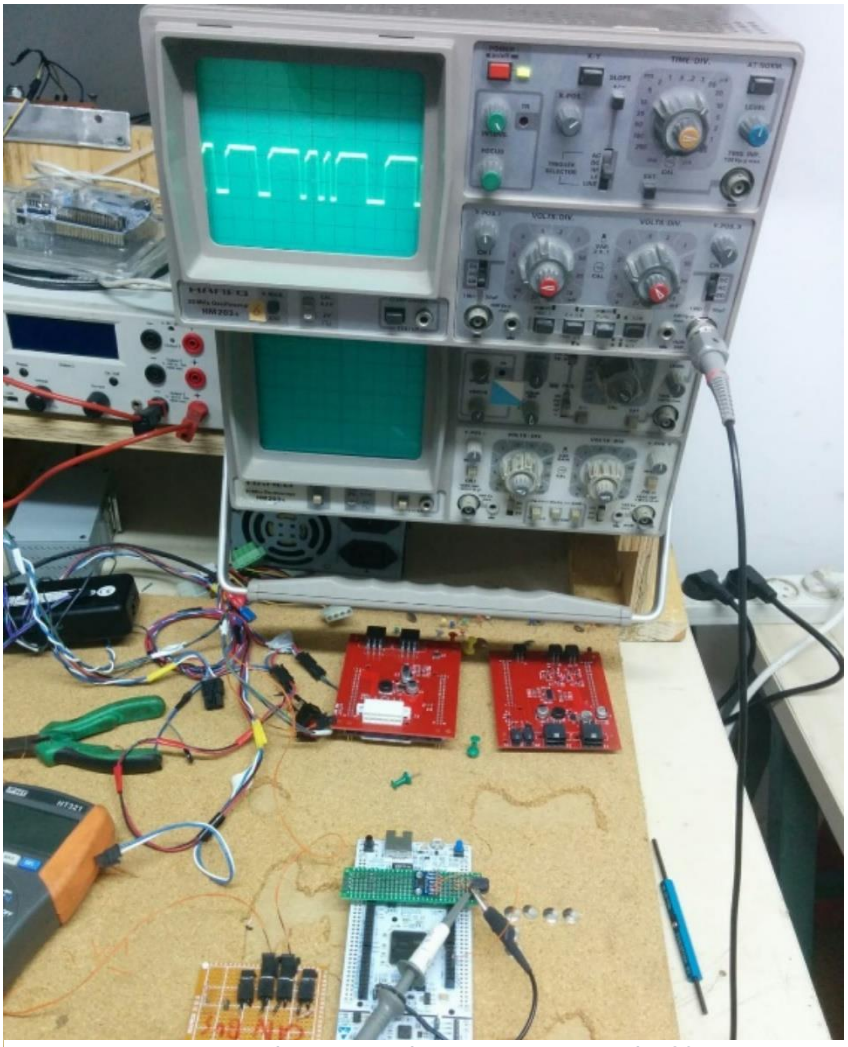
איור29 - אספקת המתח לרכיב ה-ethernet switch

CAN-bus 5.5.2

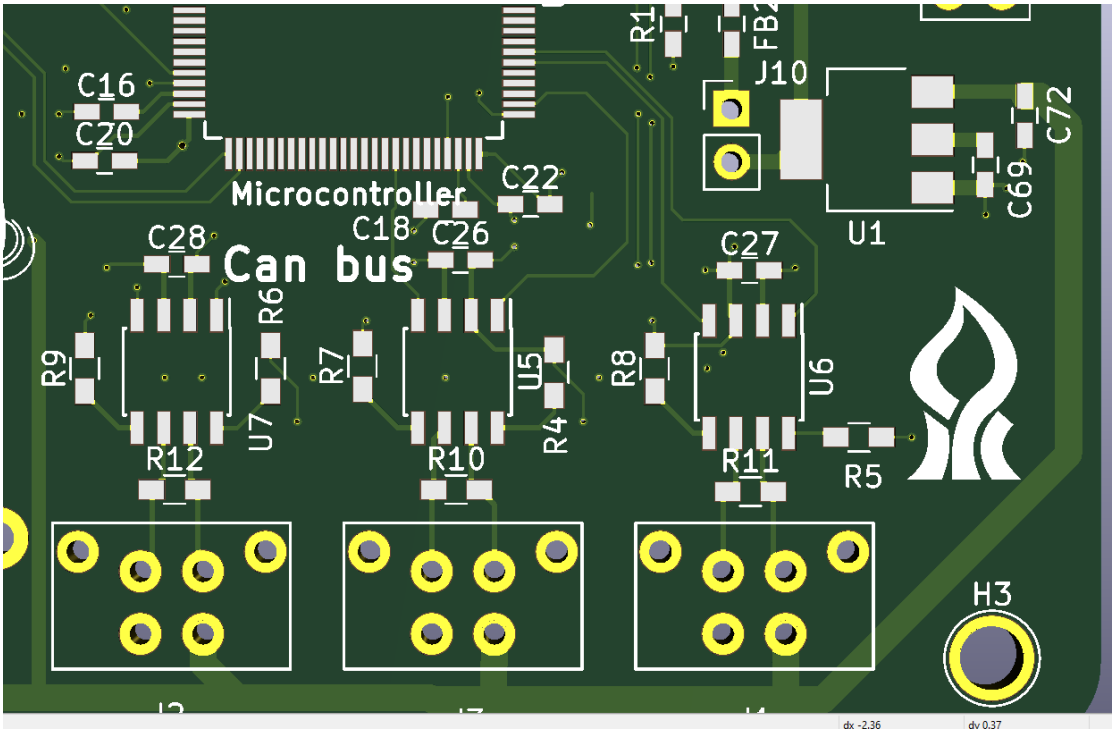
את התפקוד של ה-CAN transceiver יחד עם ה-STM בדקנו בשלב ה-prototyping, לפיו ניסינו לבחון כל רכיב בנפרד. בשלב זה השתמשנו בלוח פיתוח מסוג nucleao 144 אשר מדמה את ה-STM שלנו, והשתמשנו במודול transceiver של CAN-bus שרכשנו מ-ebay.

בשלב זה שידרנו הודעות CAN-bus, רשמנו driver ל-STM ובחנו את האותות שבין ה--CAN busל-STM. לאחר מכן, שיבצנו את רכיב ה-can transceiver בלוח.

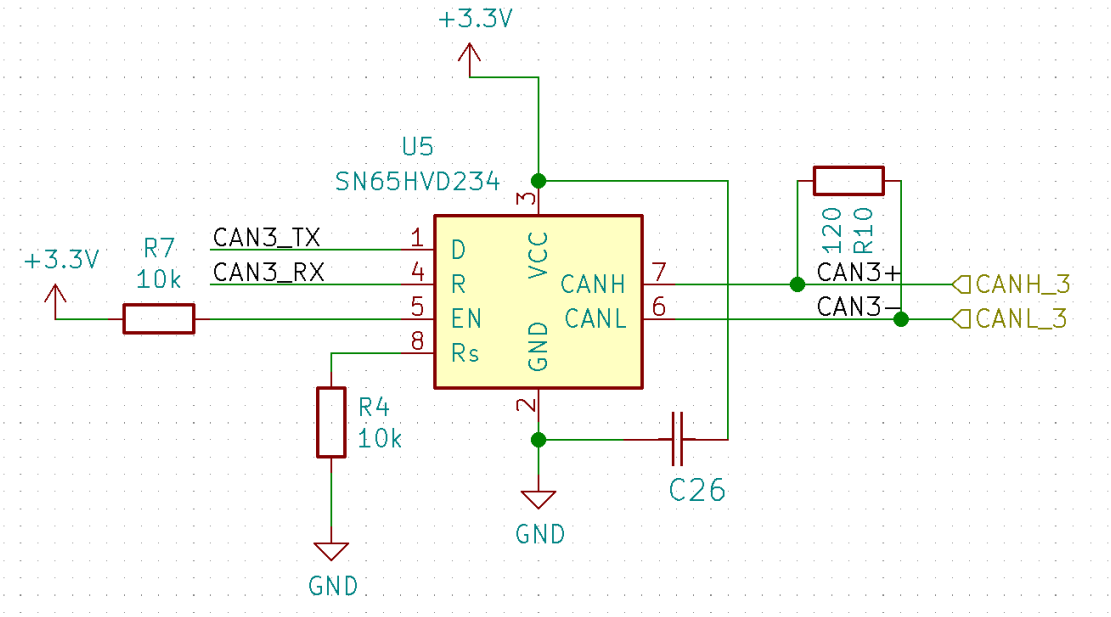
ברכיב זה חשוב להקפיד שהאורך הנתיב של ה-D ושל ה-R יהיה דומה, זאת בשל גישת שיטת הערוץ של ה-CAN-bus, לפיה כל אות שנשלח ממתין לקבלת האות שהתקבל. במקרה וההפרש בין האורכים שונה מסנטימטר, מנגנון ה-can controller אינו יכול לתפקד כראוי. בנוסף, בין ה- CAN high ל-CAN low יש לשים נגד טרמינציה של 120 אוהם.



איור 30 - לוח פיתוח nucleo 144 אליו מחובר מודול CAN transceiver



איור 31 - הדמיה של מעגל ה-CAN-bus על הלוח

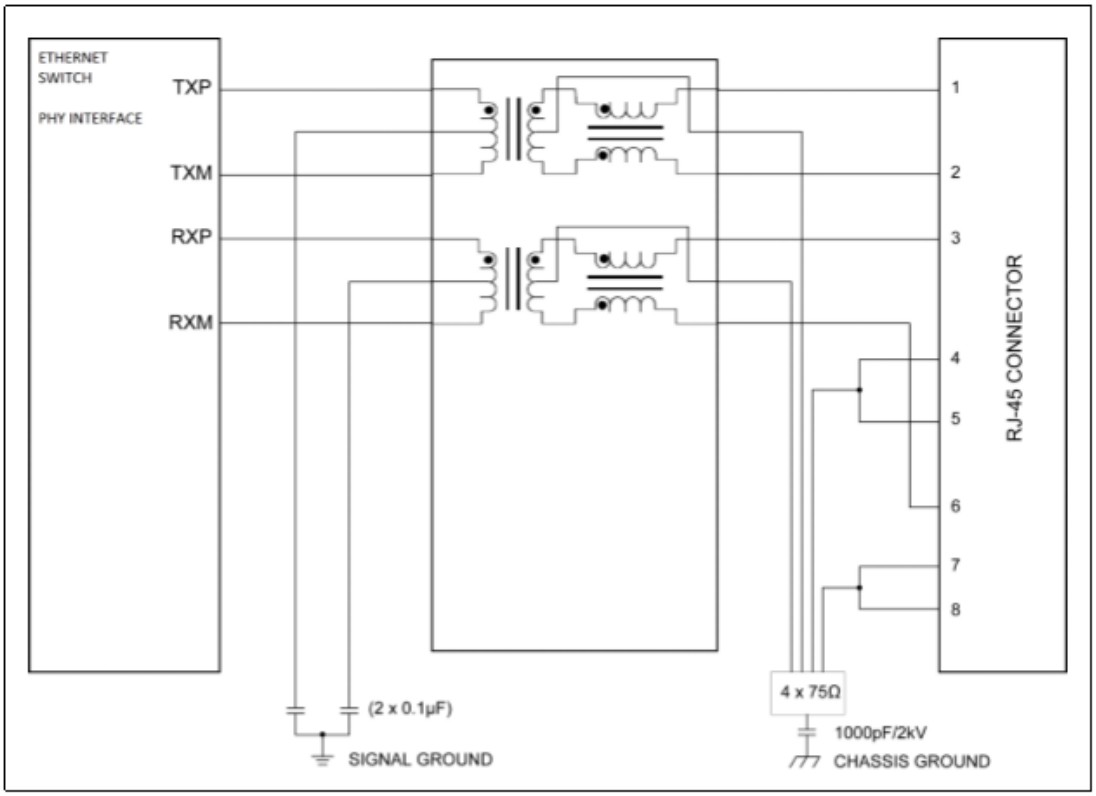


איור 32 - שרטוט חשמלי של מעגל ה-CAN-bus

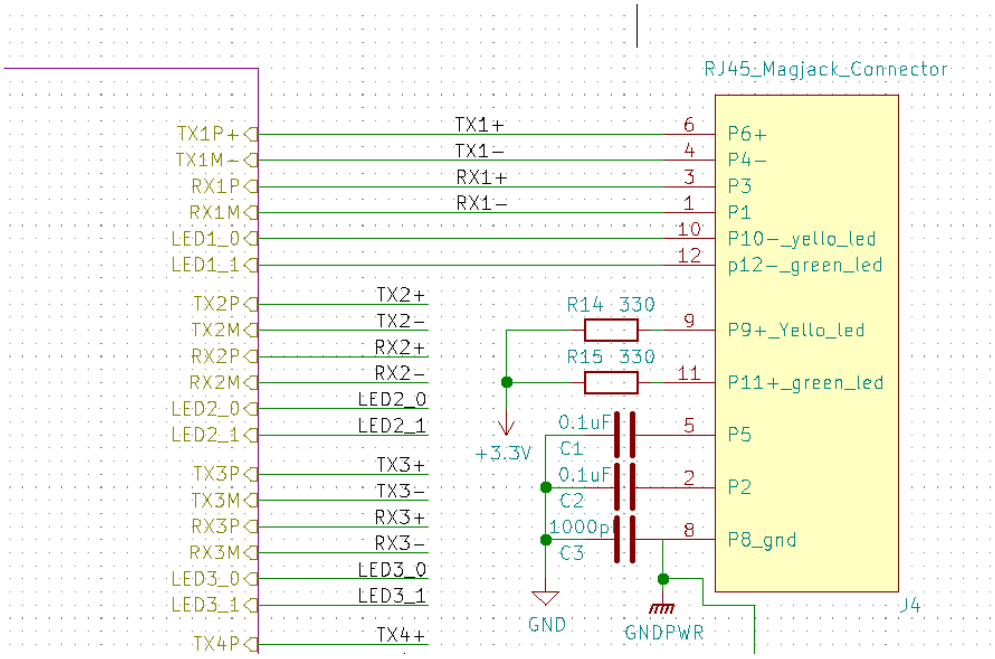
Ethernet 5.5.3

עבור חיבור Ethernet PHY בין ה-ethernet switch למחבר RJ45 קיים הסבר קצר המסביר על הקווים הכלליים של חיבור כזה ב-datasheet של ה-Ethernet switch. ההסבר מתאר כיצד יש לחבר כראוי לפי הגדרת הסטנדרט של ה-Ethernet.

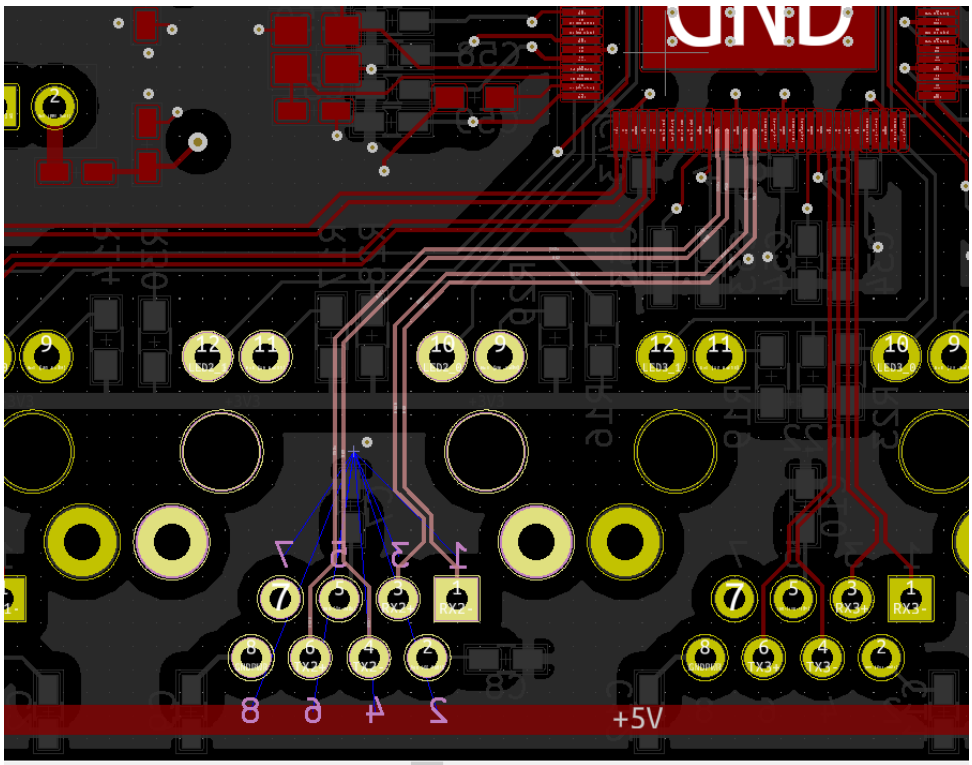
הטווח של ה-Ethernet מורכב משני קווים, כאשר כל קו מורכב משני גידים מסובסבים (TP) כך שהאימפידאנס האופייני של טווח זה הוא 100 אוהם. אולם, נגדי הטרמינציה במעגל זה נמצאים הן בתוך המחבר RJ45, והן בתוך רכיב ה-Ethernet switch.



איור 33 – המלצת היצרן לחיבור transformer אל ה-PHY של ה-Ethernet swtich



איור 34 - סכימה חשמלית של חיבור ה-RJ45 ל-PHY



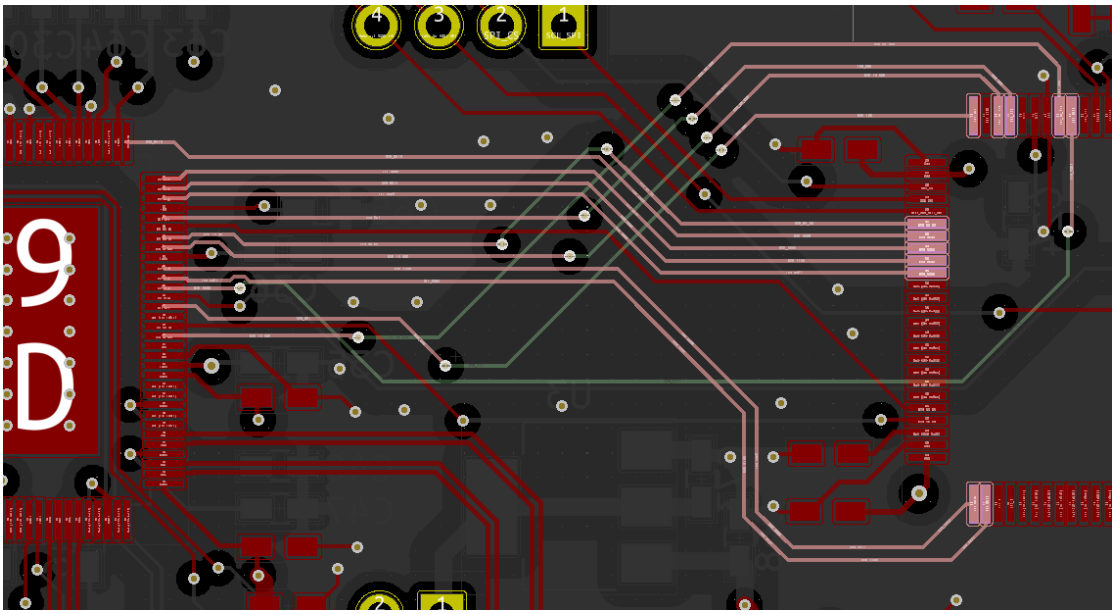
איור 35 - חיבור ה-RJ45 ל-Ethernet switch

דגשים לגבי ה-layout: האימפדאנס האופייני של ה-trace-ים צריך להיות גם הוא 100 אוהם עבור כל זוג differential, וההתנגדות ל-trace קטנה מ-50 אוהם.

5.5.4 חיבור MII

בלוח זה בחרנו להשתמש בממשק שנחשב מיושן מעט וכבר כמעט ולא משתמשים בו, זאת משום שבכדי להעביר מידע במהירות של 100mbps נעשה שימוש של 4 קווים single ended לכל כיוון. לכן, אנחנו מקבלים שסך כל קווי ה-data בממשק הם 8, לא כולל עוד 2 קווים להעברת אות השעון. ומכאן שקיבלנו ממשק מסורבל עם קצב העברה בינוני.

כיום מקובל להשתמש בוריאציה המכווצת שלו RMII או SMII, כאשר ב-RMII עוברת אותה כמות של מידע על 2 קווי data לכל כיוון, וב-SMII על קו אחד לכל כיוון. הסיבה שבחרתי להשתמש בממשק ה-MII היא מכיוון שרציתי להגדיל כמה שיותר את סיכויי ההצלחה של הלוח כבר בניסיונות הראשונים. למרות שמהירות השעון של ממשק זה היא 25MHz עדיין נעשתה השתדלות למדל ולשערך עד כמה שאפשר את האות העובר בממשק זה.



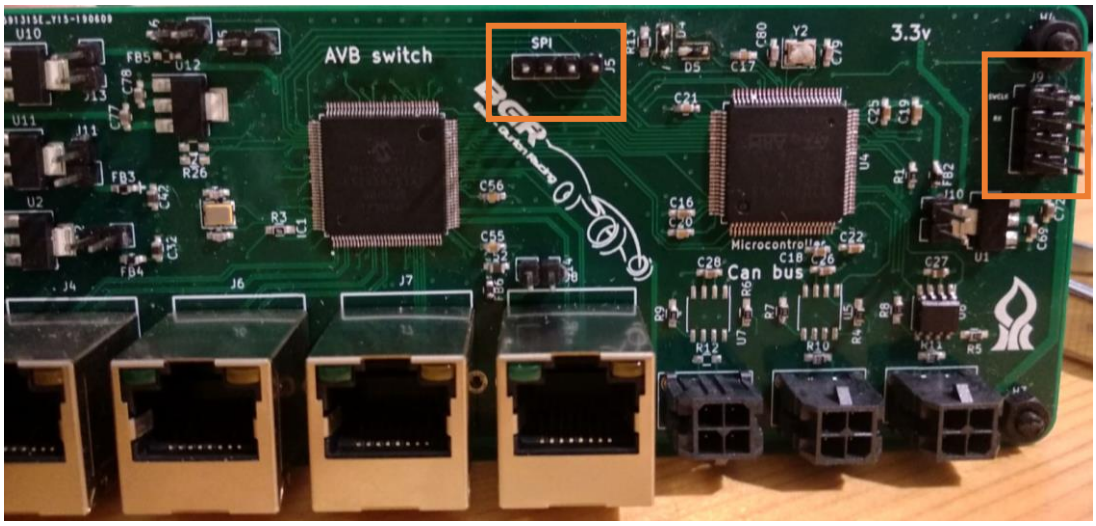
איור 36 - תרשים layout של חיבור MII

5.5.5 הוספת קריסטל קוורץ

ה-CAN-bus הוא פרוטוקול סינכרוני וכדי לממש את הפרוטוקול סטיית השעון שלו צריכה לעמוד בטווח מסויים, ולכן עדיף שמקור מחולל התדר של רכיב ה-CAN controller יהיה מקריסטל חיצוני ולא מהאוסילטור הפנימי של בקר ה STM32. לגבי ה-Ethernet switch, בכדי לממש את פרוטוקל ה-MII יש צורך בהוספת קריסטל חיצוני בתדירות של 25MHz. בשני המקרים בעת הוספת קריסטל קוורץ יש להוסיף קבלי תהודה

5.5.6 ממשקי קינפוג ודיבאג

ה-STM תומך בממשק לצריבת תוכנה ודיבאגר SWD. הממשק קולט 2 קווים: שעון ו-data. כמו כן, ממשק הקינפוג של ה-Ethernet switch הוא באמצעות SPI. בלוח זה הוצאתי header-ים הן עבור קווי הדיבאגר והן עבור קווי ה-SPI, כדי שיהיה נוח לחבר את הדיבאגר ולחבר logic analyzer אל קו ה-SPI.



איור 37 - ממשק הדיבאגר וה-SPI

5.5.7 מקורות מתח

הלוח מקבל 3 מקורות מתח: 3.3, 2.5 ו-1.2 וולט, כאשר ה-Ethernet switch צורך חצי אמפר ב-1.2 וולט, דהיינו ההספק שלו הוא 0.6 watt.

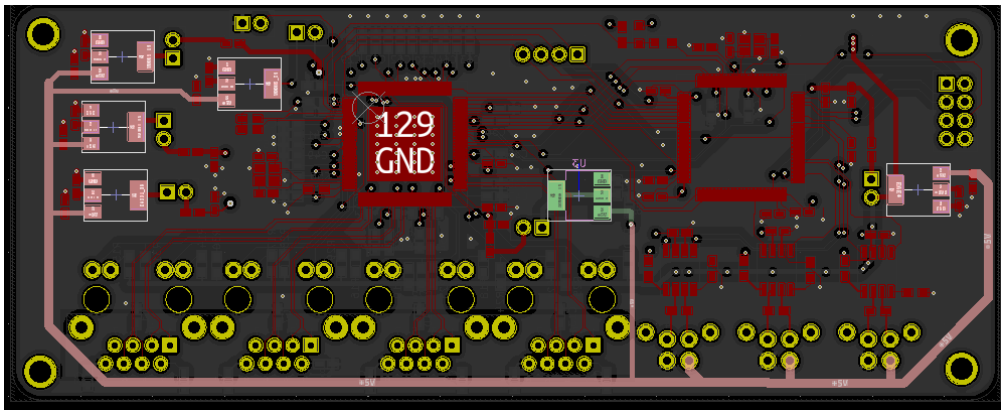
במהלך הפרויקט תכננתי וייצרתי 2 איטרציות של הלוח הזה:

- באיטרציה הראשונה הייתה לי טעות ב-footprint של הרגולטורים, ולכן הלוח סופק מספק כוח, כאשר בעת ההדלקה, שלא במודע, הדלקתי בכל פעם את מקור המתח הגבוה יותר ולאחר מכן את מקור המתח הנמוך. הלוח עבד היטב, לא התחמם ולא צרך הספק העולה על הטווח התקין.
- באיטרציה השנייה, תיקנתי את ה-footprint-ים ואז גיליתי את הבאים:

1. הרגולטור U2 מתחמם מאוד. התחממות זו מחממת את כל האזור בלוח והאזור הזה מחמם את ה-Ethernet switch, ולכן הלוח באיטרציה השנייה לא יכול לעבוד לאורך זמן בשל בעיית פינוי חום.

   בנוסף, מתבצע בזבוז של אנרגיה ברגולטור U2, לפיו מתבצעת הורדה מ-5 וולט ל-1.2 וולט בזרם של 0.3 אמפר, מה שגורם להספק חום של 1.14 watt.
2. פרוצדורת הדלקת המערכת צריכה להתבצע מהמתח הגבוה אל המתח הנמוך ביותר. בלוח שלי כל המתחים מוזנים ממקור משותף של 5 וולט.

3. בלוח הראשון סבלנו מבעיה של אדמה צפה בחלק מאיזורי הלוח. בכדי לפתור את הבעיה, הגדלתי את ה-clearance, הסרתי מהשכבה העליונה את המילוי של ה-GND. הדבר פתר את בעיית האדמה הצפה אך יצר בעיה של נידוף חום, שכן ה- Ethernet switch נוטה להתחמם יותר, זאת משום שה-GND פחות רציף והחום לא מצליח להתפשט לשכבה העליונה והתחתונה, ולכן החום נאגר במרכז הלוח.



איור 38 - מקור המתח בלוח

5.6 הלחמה ובדיקת הלוחות

כדי לפשט את תהליך ההלחמה, ולקבל גימור יותר יפה ללוח בחרנו להשתמש בשיטת הלחמה אחרת, משחת הלחמה, לוח stencil ואוויר חם. תוכנת ה-kicad מאפשרת לתכנן גם שבלונה המיועדת לתהליך ההלחמה. בשיטה זה ממקמים את השבלונה מעל הלוח, כאשר בשבלונה מחוררים כל הפדים המיועדים להלחמה. יש למקם אותה מעל הלוח כך שחורי השבלונה יהיו מעל הפדים המתאימים בלוח, מורחים את משחת ההלחמה, לאחר מכן מניחים את הרכיבים בעדינות עם פינצטה ולבסוף, בעזרת אקדח אוויר חם, מחממים את הלוח ל-300 מעלות ומשחת ההלחמה מומסת ותופסת את הרכיבים במקום.

השיטה נמצאה מוצלחת מאוד עבור הרכיבים הפאסיבים בלוח, הכבלים, הנגדים ועבור רכיבי IC, אך עבור ה-stm32 ועבור ה-Ethernet switch השיטה הייתה גסה מידי ויצרה גשר אחד גדול בין כל רגלי הצ'יפ. ולכן מומלץ להלחים את הצ'יפ בסוף באופן ידני. בעת ההלחמה מומלץ מאוד להשתמש במיקרוסקופ. אני השתמשתי במיקרוסקופ דיגיטאלי פשוט שנמצא מתאים מאוד.
דגש להלחמה בשיטה זו:

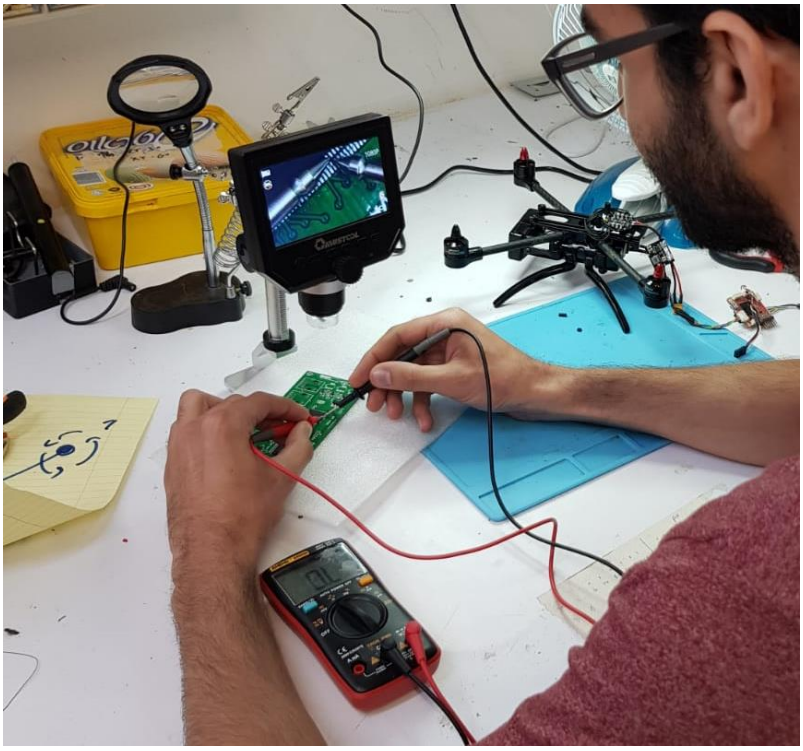
מהרגע שמורחים את המשחת ההלחמה יש לסיים את שלב הנחת הרכיבים במשך זמן שלא עולה על 15 דקות שכן לאחר מכן המשחה מאבדת משמעותית את היכולת שלה להנמס ואז יש להפעיל יותר חום, מה שגורם לשריפת הלוח ומזיק לרכיבים.

5.6.1 תהליך בדיקת הלוח

שלב בדיקת הלוח היה ללא ספק השלב הכי מורט עצבים בתהליך: לוקח הרבה זמן, אנרגיות וכסף להגיע לרגע הזה שמחברים את הלוח לחשמל. בשלב הזה הלוח מאוד עדין, ישנן טעויות, הן בתכנון והן בהלחמה, והמשחק הוא למצוא אותן במלאכת חקירה עדינה, מבלי להזיק ללוח.

בלוח שלי יצרתי מחברי jumper אשר מאפשרים ניתוק מלא של מקורות החשמל מה-stm32 וה-Ethernet switch, כך בשלב הראשון פתחתי את כל החיבורים.תהליך הבדיקה מתבצע באופן הבא:

- בשלב הראשון טרם מכניסים חשמל למערכת מגדירים איזה מנגנון בוחנים עכשיו בלוח. אני התחלתי מבדיקת ספקי הכוח. לפני החיבור לחשמל עברתי על כל הצמתים שאותם אני בודק ווידאתי שאין קצרים ושהלוח תואם את התכנון. את בדיקת הקצרים עשיתי בעזרת voltmeter. לאחר מכן בדקתי את ההתנגדויות לראות שאין שום מדד חריג, אשר יכול להגרם מתקלה בייצור הלוח או מכשל בהלחמה. לאחר שבדקתי את כל אלו יש לעבור לחלק השני בניסוי.
- בשלב השני מספקים ללוח מתח. אספקת המתח חייבת להיעשות רק על ידי ספק כוח ועל ידי הגבלת זרם, זה מאוד מאוד חשוב. הגבלת הזרם מאפשרת לטעויות להשאר טעויות ולא להפוך לאסון שברגע אחד מבטל את כל עבודת ההלחמה שנעשתה ויש להלחים לוח חדש. בעת הספקת המתח מהספק יש להגדיר את מתח והזרם הדרושים כך שלא ייחרגו מהגדרות המקסימום שרשומות במסמך ה-Datasheet של היצרן. לאחר מכן יש להעריך כמה המעגל הנוכחי צורך, זאת בכדי לזהות בשלב זה אם קיימות זליגות זרם.
אספקת המתח היא שלב עדין . ברגע זה יש לשים לב ללוח, לעקוב בעיניים ולהריח שאין עשן, לגעת בידיים ברכיבים ולאמוד את הטמפרטורה שלהם, לבחון את התנהגות ספק הכוח ולהיות דרוכים לכבות אותו במקרה של קצר. אם שום אסון לא קרה, אפשר להמשיך ולבחון האם המעגל מבצע את מה שהוא אמור לעשות.
בעזרת voltmeter יש למדוד את המתח בצמתים ולראות שהוא תואם את המדדים.
לאחר מכן בעזרת אוסילוסקופ יש לבחון את טיב האותות כדי לקבל סדר גודל של טיב תכנון הלוח והרכיבים.
- באותו האופן ממשיכים את הבדיקה עבור ה-stm32 וה-Ethernet switch. אני בדקתי כל רכיב לחוד בעזרת ניתוק של האחד מהשני בעזרת ה-headers. בנוסף, חשוב מאוד לבדוק קודם שאין שום קצרים בין ה-VCC ל-GND. לאחר אספקת המתח, יש לחבר את ה-debugger ובעזרת התכונה STM32 ST-LINK Utility לבדוק האם ה-debugger מצליח לתקשר עם ה-stm32.
עבור ה-switch הבדיקה שלי הייתה האם הוא מצליח לנתב הודעות Ethernet.



איור 39 - תהליך בדיקת קצרים בין הרגליים של stm32f765 בעזרת מיקרוסקופ ו-וולטמטר

5.7 אלגוריתמיקה ותוכנה

ענף האלגוריתמיקה והתוכנה אינו תלוי בתהליך שיבוץ החומרה, ולאורך כל הפרויקט לא המתנתי עד לסיום שלב החומרה, אלא העמדתי סביבת עבודה מקבילה בה השתמשתי בלוח פיתוח וחיברתי אליו את הרכיבים הנחוצים.

חברת stm32 מספקת את כלי הפיתוח הבאים :

1. Stm32Cube – כלי גראפי המאפשר לארגן את סביבת העבודה ולחבר בקלות את מחלקות ה-HAL (hardware abstraction layer) אל סביבת הפרויקט בקלות. כמו כן מאפשר להתחיל פרויקט עם main ו-interrupt handler.
2. IDE Stm32workbanch – יעודי לעבודה עם בקרי stm32 ומבוסס על eclipse.

על אף שתי הכלים שמאוד מקלים על העבודה, תכנות הבקר דורש הבנה מלאה של ארכיטקטורת הרכיבים עליהם עובדים, ולכן בשלב הראשון יש לקרוא ב-reference manual על הרכיב אותו רוצים להפעיל, ולהבין את שיטת העבודה שלו טרם ניגשים לכתיבת התוכנית. עיקר העבודה עסקה במימוש פונקציונאליות ה-CAN-bus במערכת ועבודה עם ה-Ethernet.

בפרויקט תחילה עבדנו בשליחת הודעות raw Ethernet. את ה-header של ה-IP וה-UDP מילאתי

לבד, אולם לאחר מכן עברתי לשימוש ב-middleware שנקרא lwip, אשר נותן סביבה עשירה יותר תמיכה ב-DHCP, הודעות ARP ו-ICMP ומדמה מין תהליך של פתיחת socket.

ארכיטקטורת התוכנה הייתה main loop אשר בו התנהלו הפעולות הפחות רגישות בזמן, וב-interrupt handler התרחשו הפעולות היותר רגישות.

5.8 אינטגרציה ברכב – הטמעת המערכת ברכב

בכדי להצדיק את רלוונטיות פרויקט הגמר בחרתי לשלב את השיטות והרעיונות שנהגו בשלב התחקיר בתוך פרויקט הרכב של בן גוריון.

זאת השנה הרביעית שבה אני חבר בקבוצת הרכב. מלבד משימת פיתוח תשתית התקשורת במסגרת פרויקט הגמר, יש לי גם אחריות על התקנה ומימוש מערכות ברכב אשר הכרחיות לנסיעת הרכב והשתתפותו בתחרות.

המערכת עליה אני אחראי, יחד עם חבר צוות נוסף, היא מכלול מחשב הרכב ומערכת ה- drive by wire. משימתנו היא לכתוב את תוכנית הנסיעה כך שתעמוד בקריטריונים של תקנון התחרות ולהיות אחראים על הממשק של המערכת עם מערכות המתח הגבוה, האינוורטרים ודגימת ערכי דוושות התאוצה והבלם בקדמת הרכב.

על אף שמשימתי בפרויקט הרכב שונה מפרויקט הגמר, הצלחתי למצוא הרבה מוטיבים דומים ולאורך העבודה שילבתי בין השניים. הידע שצברתי בפרויקט הרכב העצים את פרויקט הגמר וזירז תהליכים, מצד שני הכלים החדשים שרכשתי בשיבוץ חומרה יעודית, אותם רכשתי במסגרת פרויקט הגמר, אפשרו לי להגיע לרמת גימור גבוהה יותר ולשדרוג הפתרון שהיה קיים ברכב.

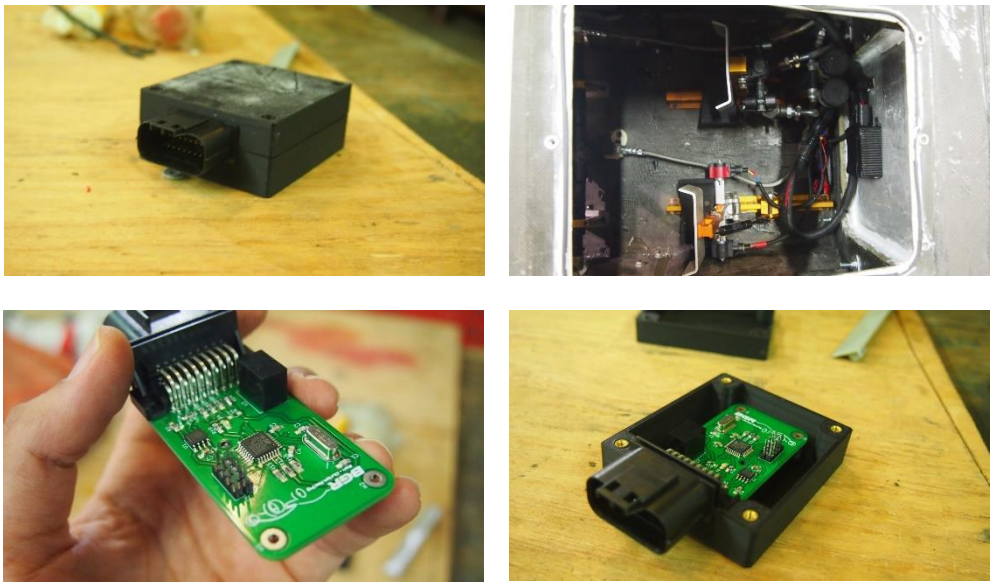
5.8.1 מערכות התקשורת ברכב

ברכב פזורות מספר מערכות באחריות צוות תקשורת:

1. מחשב רכב – תפקיד מחשב הרכב הוא לספק פקודת נסיעה בכל 5 אלפיות השנייה. את החיווי עבור ערך פקודת הנסיעה הרצויה מקבל מחשב הרכב באמצעות הודעת CAN-bus המגיעה מבקר הדוושות, ואת פקודת הנסיעה שהנפיק משדר אל האינוורטרים (בקרי המנועים) באמצעות הודעת Ethernet. בנוסף, אחראי מחשב הרכב על ניהול מכונת המצבים של הרכב: האם הרכב במצב ניטרלי, מצב נסיעה או במצב תקלה. כמו כן, מחשב הרכב גם מנהל תעבורת הודעות עם בקרי המנועים ומנטר את המצב שלהם ותקינותם. בנוסף, מחשב הרכב אחראי על סנכון רשת ה-CAN-bus ע"י שליחת הודעת סנכרון 0x80, מה שמקובל בפרוטוקול ניהול CAN open.

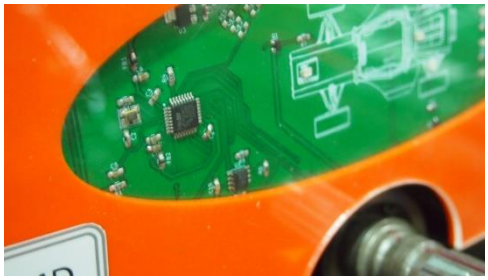
איור 40 - קופסת תקשורת ובתוכה מחשב הרכב

2. בקר דוושות – בפרויקט זה בקר הדוושות עבר מתיחת פנים ונוצר מחדש כשהוא פי 4 יותר קטן מגירסתו הקודמת, ותוכנן מראש לעמוד בסביבת הרכב. תפקיד בקר הדוושות הוא לדגום את החיישנים הממוקמים על הדוושות, שהם חיישן לחץ שמן וחיישן זווית גז, לבצע המרה אנלוגית לדיגיטלית ולשלוח את המידע הספרתי אל רשת ה-CAN-bus . בקר הדוושות מורכב ממעבד STM32 ו-CAN transceiver באופן שנגזר מפיתוח לוח ה-gateway בפרויקט הגמר.



איור 41 - בקר דוושות

3. דשבורד – תפקיד הדשבורד לתת חיווי לנהג. בדשבורד שלנו ניתן לקבל חיווי על תקינות מערכות הנסיעה, ביניהן תקינות מחשב הרכב, בקר דוושות ואינוורטרים. כמו כן, יש 2 צגי 7 segment שעליהם ניתן להציג מדדים רצויים. בתוך הדשבורד משובץ מעבד STM32 ורכיב can transceiver, מה שמאפשר את החופש לתכנת את הדשבורד ולהציג כל מידע שעובר על רשת ה-CAN-bus.



איור 42 - דשבורד

4. טלמטריה – את פתרון לטלמטריה ברכב מימשנו באמצעות מוצרי מדף. השתמשנו בלוח raspberry pi ובמתאם wifi usb. את ה-RPI חיברנו לרשת ה-Ethernet ברכב והתקנו עליו תוכנה שהופכת אותו לנקודת hotspot, ובכך הצלחנו להתחבר לרכב באופן אלחוטי. מאחר וה-RPI הוא מכונת לינוקס ניתן לממש עליו מגוון יישומים.

5.8.2 פיתוח יכולות לפני התקנתן על הרכב

חלון הזמן בפרויקט הרכב בו עובדים על הרכב עצמו הוא קטן מאוד. זאת משום שרק לקראת סוף השנה מגיעים כל המכלולים והרכיבים, ורוב השנה הרכב מפורק ואינו במצב נסיעה. צוות התקשורת, בשונה מצוותים אחרים, אינו עסוק בתכנון וייצור המוצר, אלא עסוק בפיתוח מתמשך. ולכן, צריך לעבוד ולתכנת על הרכב. הפתרון שמצאנו, שבעזרתו ניתן לפתח מצד אחד אך מצד שני ללא הרכב, הוא בעזרת מודל פרוש המדמה את מערכת התקשורת ברכב,

המודל הפרוש מאפשר לפתח את כל הפונקציונליות הנחוצה לתחרות, אולם המערכת לא נבדקה בסביבה הייעודית, שהיא סביבה של רכב חשמלי נוסע, ולכן קיים פער בין מה שעובד על המודל הפרוש לבין מה שיעבוד בפועל .

5.8.3 עבודה על הרכב והכנת הרכב לנסיעה

הרכב הוא פרויקט משותף למגוון מחלקות וספקטרום המערכות המותקנות על הרכב נע בין מערכות מכניות, מערכות חשמליות של מתח גבוה ומערכות בקרה. טרם נסיעה ברכב יש להכין אותו בסדר הבא:

- בשלב הראשון עובר הרכב כיול מכאני. בודקים ומכיילים את מערכות התמסורת, המתלים והשלדה, בודקים את מערכות מכלול הגלגל וכו'.
- לאחר מכן בודקים את מערכות המתח הנמוך ותקינותן.
- בשלב הבא בודקים את מערכות המתח הגבוה.
- ורק לאחר שכל אלה התרחשו ניתן לבדוק את מערכות הבקרה ולהפעיל מנועים (השלב שבו צוות התקשורת עובד על הרכב). אנחנו הצוות האחרון בשרשרת שמכין את הרכב לקראת נסיעה.

תהליך הכנת הרכב לנסיעה אורך מספר שעות, מתחילים את העבודה בסדנת הרכב בשעה ב-18:00 (אחרי שהחום נשבר), ובדרך כלל צוות התקשורת מתחיל לעבוד לקראת שעות הבוקר המוקדמות. לכן, אם נרצה לבצע עדכון גרסה ולדרוש זמן עבודה על הרכב יהיה זה לפנות בוקר.

איור 43 - עבודה והכנת הרכב לקראת נסיעה – מתחילים בערב, מסיימים בבוקר

5.9 תחרות

פעילות הקבוצה מסתכמת בכל שנה בשליחת הרכב להתחרות בתחרויות ה-FSAE שנערכות באירופה. התחרות היא תחרות הנדסית לפיה בוחנים את הרכב לפי הבאים:

- תחילה בודקים אם בניית הרכב תואמת את התקנון.
- לאחר מכן, נערך אירוע ה-design. שם הרכב נבחן ומקבל דירוג על סמך טיב התכנון שלו.
- לבסוף, מגיע השלב הדינאמי בו הרכב עולה על המסלול ונבחן על הביצועים שלו.

התחרות מאפשרת לנו לקבל פרספקטיבה על התוצרים שמייצרות קבוצות אחרות בעולם, ומאפשר לנו ללמוד לשתף מידע, וליישר קו עם הסטנדרטים של שאר האוניברסטאות העולם .

לפני שנתיים, טסתי להונגריה להשתתף בתחרות FSAE, ושם למדתי כיצד קבוצות אחרות מממשות את מערכות התקשורת, מערכת מחשב הרכב ומערכות בקרה שונות. התרשמתי מאוד מהסטנדרטים ומרמות הגימור אליהן מגיעות הקבוצות הללו. השנה, המערכת שצוות התקשורת התקין על הרכב לא נפלה מרמות הגימור והסטנדרטים של אף קבוצה. היכולת ליצור לבד את החומרה ואת התוכנה ולהתאימם לרכב באופן לא מתפשר, הציגה מערכת בשלה וסופית. לאורך התחרות המערכת עבדה באמינות ובכל פעם שהעלו את המתח ברכב, המערכת עבדה.

במהלך התחרות הסתובבתי בין הקבוצות השונות וניהלתי שיח עם חברי הקבוצות השונות, בעיקר עם אלה המקבילים אליי בתפקידם. מהר שמחתי לגלות שבהמלך 4 שנות פעילותי בקבוצה הצלחנו לצמצם את מרבית הפערים שהיו לנו לפני שנתיים בתחום התקשורת, ובמקום להיות מקור שלומד ושואב ידע הפכנו להיות מקור שמייעץ ונותן ידע לקבוצות אחרות בתחום התקשורת.

אנחנו בשלים לקחת את פרויקט הרכב לשלב הבא ולהתחיל להיכנס לתחרות הרכב האוטונומי לצד האוניברסיטאות המובילות בעולם. **סטודנטים מהמחלקה להנדסת מערכות תקשורת הינם חברי צוות יקרים בפרויקט שכזה וביכולתם לתת מענה לשלל אתגרים הנדסיים** ויכולים ללמוד המון בדרך.

במהלך אירוע ה-design שופטי התחרות התעכבו ארוכות לשוחח איתי על פתרונות התקשורת ברכב. על הרעיון של לגשר בין רשת ה-CAN-bus לרשת ה-Ethernet ועל היישומים שניתן לעשות עם גישור זה. הצגתי להם את לוח ה-gateway והם התרשמו מאוד מרכיב שונה זה בנוף התחרות, ואפילו קיבלתי מחמאה על המחשבה והיצירתיות יוצאת הדופן.

את התחרות סיימנו בקבלת פרס driving the future בזכות החדשנות שהצגנו, הערבות ההדדית והדינמיקה הקבוצתית.

איור 44 - רגע ההודעה על זכייה בפרס driving the future

stratasys
ARAN
AMICELL
Altair
HyperWorks
Arihav
Hoosier
PRONAT
matrix
POLYMER G
55
CAR 155
BGRe - Ben Gurion
Racing Electric
Ben-Gurion University of the Negev
Formula SAE Italy,
Formula Electric Italy,
& Formula Driverless

6 מקורות

[1] Kim, Seung-Han, et al. "A gateway system for an automotive system: LIN, CAN, and FlexRay." *Industrial Informatics, 2008. INDIN 2008. 6th IEEE International Conference on*. IEEE, 2008.

[2] https://www.lifewire.com/what-is-drive-by-wire-534825

[3] Isermann, Rolf, Ralf Schwarz, and Stefan Stolzl. "Fault-tolerant drive-by-wire systems." *IEEE Control Systems* 22.5 (2002): 64-81.

[4] Pawar, Vikas Tukaram T. *Active Torque Vectoring for All Wheel Drive FSAE Electric Car*. Diss. 2016.

[5] Alderisi, Giuliana, Giancarlo Iannizzotto, and Lucia Lo Bello. "Towards IEEE 802.1 Ethernet AVB for advanced driver assistance systems: A preliminary assessment." *Emerging Technologies & Factory Automation (ETFA), 2012 IEEE 17th Conference on*. IEEE, 2012.

[6] IEEE Standards Association. "Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks." *IEEE, Mar*. Vol. 30. 2011.

[7] https://support.ixiacom.com/sites/default/files/resources/whitepaper/ixia-automotive-ethernet-primer-whitepaper_1.pdf

[8] Meyer, Philipp, et al. "Extending IEEE 802.1 AVB with time-triggered scheduling: A simulation study of the coexistence of synchronous and asynchronous traffic." *Vehicular Networking Conference (VNC), 2013 IEEE*. IEEE, 2013.

[9] https://www.valens.com/automotive-solutions ,Valens – HdbaseT Automotive Seminar

[10] /https://atul.fyi/post/2017/01/03/how-self-driving-cars-work

[11] https://www.mouser.com/ds/2/389/stm32f765bg-956218.pdf

[12] http://www.mouser.com/datasheet/2/268/00002391A-1108116.pdf

[13] www.ti.com/lit/ds/symlink/sn65hvd234.pdf

[14] https://productfinder.pulseeng.com/doc_type/WEB301/doc_num/JD2-0010NL-01/doc_part/JD2-0010NL.pdf