

המרכז האקדמי רופין
בית-הספר להנדסה
המחלקה להנדסת חשמל ומחשבים

ספר פרויקט גמר לשנת תשע"ט

נושא הפרויקט:

מערכת להצלת ילדים מטביעה

מבצעי הפרויקט:

054-4929292

נייד:

רן משיח

ran.mash@gmail.com

מייל:

054-7280910

נייד:

אלכסנדר גזל

sashagezel@gmail.com

מייל:

מנחה הפרויקט:

מר ברגמן אפרים

המרכז האקדמי רופין
בית-הספר להנדסה
המחלקה להנדסת חשמל ומחשבים

ספר פרויקט גמר לשנת תשע"ט

נושא הפרויקט:

מערכת להצלת ילדים מטביעה

מבצעי הפרויקט:

054-4929292	נייד:	רן משיח
ran.mash@gmail.com	מייל:	
054-7280910	נייד:	אלכסנדר גזל
sashagezel@gmail.com	מייל:	

מנחה הפרויקט:

מר ברגמן אפרים

הצהרה: העבודה המתוארת במסמך זה היא תוצאה של מחקר אישי שלי. כל טקסט או תוצאה שנלקחו והוכנסו לעבודה זו ממקורות אחרים מתועדים ככאלה. אני יודע/ת שאי עמידה בתנאים (של עבודה עצמית, וציטוט נאות של מקורות) היא עבירה על תקנון המשמעת של בית הספר העשויה לגרום צעדים משמעתיים בפני ועדת המשמעת.

.....תאריךחתימת הסטודנט
.....תאריךחתימת הסטודנט
.....תאריךחתימת המנחה

מקרי פגיעה ומוות כתוצאה מטביעה, בעיקר של תינוקות ושל ילדים, הפכו לשגרה עצובה בישראל ובעולם. טביעה הינה סיבת התמותה השנייה בשכיחותה בקרב ילדים. טביעות נגרמות בדרך כלל בהעדר השגחה, ואפילו לשניות בודדות. זה יכול לקרות בים או בבריכה.

בשל בעיה חשובה זו הוחלט לפתח מערכת המשלבת חומרה ותוכנה שתדע לזהות מקרה טביעה בקרב ילדים ותיתן מענה מהיר ונכון לצורך הצלת חיי הילד בעת טביעה. המערכת מיועדת לילדים בגילאים 0-8 שאינם יודעים לשחות ולא אמורים להיכנס למים ללא השגחת מבוגר.

המערכת אחראית לזיהוי הטביעה ולהפעלת המנגנונים השונים להצלת הילד במקרה הצורך. המערכת החשמלית כוללת מיקרו בקר, מנוע, מקור מתח, רכיב GSM וחיישנים שונים שינטרו את העומק בו נמצא הילד. כאשר הילד ימצא בעומק מסוים לפרק זמן שהוחלט מראש, המערכת החשמלית תפעיל מנגנונים שונים להצלת הילד.

מנגנון ראשון ודרישה עיקרית מהמערכת, היא היכולת להרים את ראשו של הילד מעל פני המים ולשמור עליו במצב זה. כאשר נענה לדרישה זו נמנע מגע של מים עם דרכי הנשימה שמוביל לחנק. מנגנון שני הוא שליחת הודעת חירום (SMS) למרכז בקרה, שם יתפעלו את האירוע, וישלחו הודעת SMS חזרה להורים. מנגנון זה עונה על הדרישה לקבלת תשומת לב של ההורים. בנוסף קיים גם כפתור לבדיקת תקינות המערכת. בתגובה ללחיצה תשלח הודעת SMS עם דוח תקינות מפורט.

בשלב היישום חוברו כל הרכיבים החשמליים למערכת אחת ונצרר קוד לארדואינו שמממש את הדרישות שהוגדרו. המערכת מזהה מגע עם מים באמצעות 2 פרובים כאשר הם מוכנסים למקור מים, בעקבות זאת המערכת נדלקת ומתחילה לעבד נתונים מהחיישן לחץ / עומק. באמצעות החיישן ניתן למדוד לחץ בדיוק של 0.01% ולחשב את הפרש העומק בו נמצא החיישן. המערכת מבצעת מדידה לחץ, ומתריעה על מצב טביעה ברגע שמתגלה עומק העולה על 10 ס"מ. ולאחר חציית הסף היא מפעילה טיימר של 2 שניות. אם במהלך הטיימר המערכת לא מזהה ירידה מהעומק שהוגדר, היא מפעילה מנוע DC שפותח את הווסת לשחרור ה- CO₂ מחסנית ובכך לנפח את בלוני הציפה. המתח הנדרש להפעלת המנוע הוא 6V ולכן נעשה שימוש בממיר מתח, המעלה את מתח סוללת ה- LiPo למתח הדרוש. לאחר ניפוח בלוני הציפה, המערכת שולחת הודעה חירום למרכז בקרה. שליחת הודעות ה- SMS מתבצעת באמצעות רכיב התקשורת הסלולרית. בנוסף לבדיקת תקינות יזומה ע"י לחיצה על כפתור, המערכת מבצעת בדיקת תקינות אוטומטית לרכיבי המערכת בעת הדלקתה. לאחר בדיקת תקינות יזומה או אוטומטית נשלחת הודעת תקינות למרכז בקרה ובנוסף נדלק לד בצבע המתאים לתוצאת הבדיקה. נפח בלוני הציפה חושב באמצעות חוק ארכימדס.

לאחר בדיקות רבות וניסויים ניתן לומר כי המערכת מבצעת את הדרישות שהוגדרו בשלב התכנון. המערכת יודעת לזהות טביעה באחוז דיוק גבוה מאוד. צריכת המערכת הכוללת היא כ- 4.5 mA במצב שינה, וכ- 15mA כאשר המערכת נמצאת במגע עם מים. מכיוון שבמוצר הסופי יש הגבלה על גודל הסוללה, הוחלט כי המערכת תהיה כבויה עד אשר היא תורכב על צוואר הילד. היא תדלק באופן אוטומטי ברגע נעילת סוגר השרשרת. במקרה של תקציב גדול יותר ניתן היה להגיע למוצר מוגמר שניתן יהיה להלביש על הילד.

מערכת להצלת ילדים מטביעה

תוכן עניינים

1.	הקדמה	6
2.	סקר ספרות	7
2.1	ארדואינו	7
2.2	UART	7
2.3	תקשורת I ² C	8
2.4	GSM	9
2.5	רמת אטימות IP	9
2.6	חוק ארכימדס	10
2.7	לחץ אטמוספרי	11
2.8	בלוני ציפה	11
2.9	נפח בלוני ציפה	12
2.10	משוואת הגז האידיאלי	13
2.11	פחמן דו חמצני (CO ₂)	14
3.	תכנון הנדסי	15
3.1	דרישות עיקריות	15
3.2	תיאור המערכת	17
3.3	מבנה הקוד (אלגוריתם)	26
3.4	אתגרים במימוש	30
4.	תוצאות והדגמות	32
4.1	המערכת מבצעת בדיקת תקינות כללית לאחר הדלקתה	32
4.2	הפעלת פסיקה לבדיקת תקינות המערכת לאחר לחיצה על הכפתור	33
4.3	זיהוי מגע עם מים	35
4.4	קבלה ועיבוד נתונים מחיישן העומק	36
4.5	זיהוי מצב טביעה	37
4.6	ניפוח בלוני ציפה כמנגנון ראשון להצלת התינוק / ילד	38
4.7	שליחת הודעות חירום כמנגנון שני להצלת התינוק / ילד	39
5.	מסקנות והמלצות	40
5.1	תוצאות והישגים	40
5.2	המלצות להמשך	40
6.	ביבליוגרפיה	41
7.	ABSTRACT	42
8.	נספחים	43
8.1	נספח 1 : קוד המערכת	43
8.2	נספח 2 : בדיקות המערכת (STD)	53
8.3	נספח 3 : ניהול סיכונים	56
8.4	נספח 4 : גאנט	59

רשימת תמונות ואיורים

7	תמונה 1 - חוט TX של UART אחד מתחבר לחוט RX של השני ולהפך.
7	תמונה 2 - UART המשדר בייט מידע.
8	תמונה 3 - תיאור של אפיק I ² C.
9	תמונה 4 - דרגות ההגנה שמספקות מעטפות.
12	תמונה 5 - טווח משקלים של תינוקות וילדים עד גיל 5.
12	תמונה 6 - טווח משקלים של תינוקות וילדים מגיל 5 עד 10.
14	תמונה 7 - תכונות הפחמן הדו חמצני.
14	תמונה 8 - מחסנית CO ₂ 16 גרם.
17	תמונה 9 - מבנה חיצוני.
18	תמונה 10 - ארכיטקטורה פנימית.
19	תמונה 11 - ARDUINO PRO MINI.
20	תמונה 12 - MS5803-14BA.
21	תמונה 13 - MINI GEARED DC MOTOR 6V.
22	תמונה 14 - בקר מנוע TB6612FNG DC.
22	תמונה 15 - SIM800L.
23	תמונה 16 - DC/DC BOOST CONVERTER + CHARGER.
23	תמונה 17 - סוללת 3.7V LIPO.
24	תמונה 18 - טעינה אלחוטית.
32	תמונה 19 - המערכת מבצעת בדיקת תקינות כללית לאחר הדלקתה - תקין.
33	תמונה 20 - קבלת הודעת SMS למרכז בקרה שהכל תקין.
33	תמונה 21 - המערכת מבצעת בדיקת תקינות כללית לאחר הדלקתה - לא תקין.
34	תמונה 22 - בדיקת תקינות המערכת לאחר לחיצה על הכפתור.
34	תמונה 23 - הודעת SMS לאחר לחיצה על הכפתור הבדיקות.
35	תמונה 24 - זיהוי מגע עם מים.
36	תמונה 25 - קבלה ועיבוד נתונים מחיישן העומק.
37	תמונה 26 - זיהוי מצב טביעה לאחר שנגמר הטיימר.
38	תמונה 27 - ניפוח בלוני ציפה כמנגנון ראשון להצלת התינוק / ילד.
39	תמונה 28 - שליחת הודעות חירום כמנגנון שני להצלת התינוק / ילד.
39	תמונה 29 - הודעות החירום המתקבלות.

רשימת טבלאות

13	טבלה 1 - גיל, משקל ממוצע ונפח הבלון הדרוש.
13	טבלה 2 - צפיפות לפי סוג החומר.
19	טבלה 3 - מאפייני המיקרו בקר מתוך דף המפרט.
20	טבלה 4 - מאפייני החיישן מתוך דף המפרט.
21	טבלה 5 - מאפייני המנוע מתוך דף המפרט.
22	טבלה 6 - מאפייני בקר המנוע מתוך דף המפרט.
22	טבלה 7 - מאפייני רכיב ה-GSM מתוך דף המפרט.
23	טבלה 8 - מאפייני רכיב הממיר מתח + מטען סוללות מתוך דף המפרט.

רשימת משוואות

10	משוואה 1 - כוח העילוי.
10	משוואה 2 - הכוח השקול על גוף.
13	משוואה 3 - משוואת הגזים האידאלי.
14	משוואה 4 - נפח הגז הנפלט ממחסנית 16 גרם.
30	משוואה 5 - חישוב שינוי בגובה.

1. הקדמה

טביעה מכונה "המוות השקט", היא מתרחשת במהירות ובדממה ובדרך כלל לא שומעים בכי או צעקות מצוקה. מקרי פגיעה ומוות כתוצאה מ"המוות השקט", בעיקר של תינוקות ושל ילדים, הפכו לשגרה עצובה בישראל ובעולם. טביעה הינה סיבת התמותה השנייה בשכיחותה בקרב ילדים¹. כשליש ממקרי הטביעה שהסתיימו במוות, היו של ילדים בגילי 0-4. מאז שנת 2008 מתו בישראל מעל 210 ילדים עקב טביעה².

מקובל לחשוב כי טביעות מתרחשות רק במים עמוקים, אך למעשה מרבית הטביעות נגרמות בלי קשר לעומק המים. גם בגובה עשרה סנטימטר ילד יכול לטבוע, אם פיו ואפו נמצאים מתחת לפני המים. אצל ילדים ולעיתים גם אצל מבוגרים, מגע של מים עם דרכי הנשימה מוביל להתכווצות של שרירי הלוע, דבר שבעצמו סוגר את דרכי הנשימה ומוביל לחנק.

הנזק העיקרי מטביעה הוא המחסור בחמצן - מצב המכונה היפוקסמיה. מצב זה גורם לנזק לכל רקמות הגוף, אולם רקמת המוח רגישה במיוחד למחסור בחמצן ולכן, הנזק המוחי הוא הגורם העיקרי לתחלואה ולתמותה כתוצאה מטביעה.

טביעות נגרמות בדרך כלל בהעדר השגחה, ואפילו לשניות בודדות. זה יכול לקרות בים, בבריכה העירונית, בבריכה הפרטית בבית, או במלון. נמצא כי תשעה מכל עשרה מקרי טביעה של ילדים התרחשו כשההורים היו בקרבת ילדיהם ולעיתים אף צמודים להם ממש.

למרות אמצעי הבטיחות הקיימים למניעת טביעות ומקרי מוות בקרב תינוקות וילדים, מקרים אלו נעשו שכיחים יותר מאשר בעבר. בשל בעיה חשובה זו הוחלט לפתח מערכת המשלבת חומרה ותוכנה שתדע לזהות מקרה טביעה בקרב ילדים ותיתן מענה מהיר ונכון לצורך הצלת חיי הילד בעת טביעה.

¹ משרד הבריאות - בטיחות ילדים במים ומניעת טביעה
² היפגעות ילדים בישראל: דו"ח 'בטרים' לאומה 2017

2. סקר ספרות

2.1. ארדואינו

אַרדואינו (מאיטלקית: Arduino) הוא מיקרו-בקר בעל מעגל מודפס יחיד, עם סביבת פיתוח משולבת (IDE) ברישיון קוד פתוח, אשר מטרתה ליצור סביבה נוחה וזולה לפיתוח פרויקטים המשלבים תוכנה עם רכיבי אלקטרוניקה.

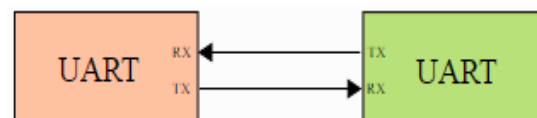
רכיב החומרה העיקרי על המעגל המודפס של הארדואינו הוא מיקרו בקר בן 8 סיביות מסדרת AVR של חברת ATMEGA, אשר מבצע את התוכנה ומכיל זיכרון הבזק (FlashMemory) לאחסונה. לארדואינו גרסאות חומרה שונות למטרות שונות. הוא מגיע בגדלים שונים, מתחי עבודה שונים וגם תדר שעון שונה.

לארדואינו יש את היכולת לתקשר עם רכיבים חיצוניים ומודולים באמצעות פרוטוקולים סטנדרטיים שהופכים את התקשורת לפשוטה ואמינה ויותר - פרוטוקולים I2C, SPI ו-UART.

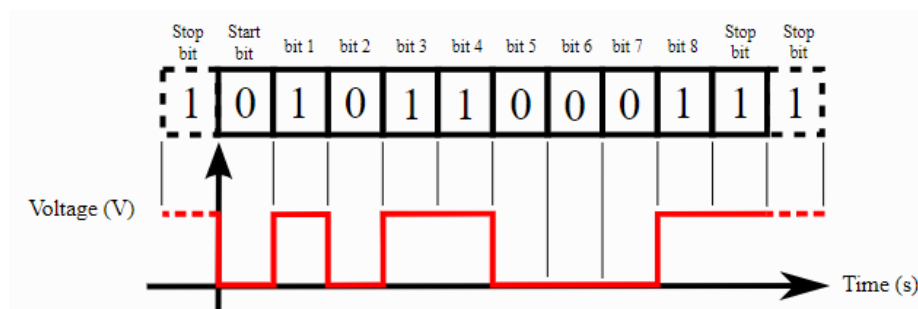
2.2. UART

UART (מבוטא "יו - ארט") הוא פרוטוקול תקשורת נפוץ לתקשורת טורית אסינכרונית. מקור השם באנגלית בראשי תיבות של Universal Asynchronous Receiver Transmitter, ובעברית "מקלט משדר אסינכרוני אוניברסלי" שהתייחס אל השבב אשר מימש את פרוטוקול התקשורת הטורית. מערכות רבות מממשות את הפרוטוקול כחלק ממעגל משולב או בתוך רכיב מתכנת.

הפרוטוקול מאפשר תקשורת טורית עם 2 חוטים בלבד (Tx ו-Rx) הפרוטוקול נותן שליטה על מספר פרמטרים כמו קצב העברת הנתונים, כמות סיביות המידע בכל, הוספת סיביות זוגיות ומשך סימבול הסיום.



תמונה 1 - חוט TX של UART אחד מתחבר לחוט RX של השני ולהפך.



תמונה 2 - UART המשדר בייט מידע

2.3. תקשורת I²C

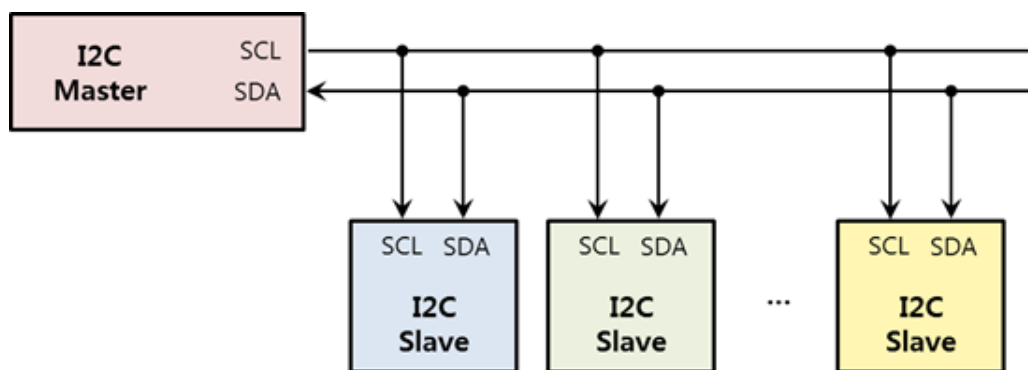
אפיק, או I²C-BUS (I²C - Circuit Integrated-Inter), ידוע גם בשם "ממשק שני-מוליכים" הוא ממשק תקשורת טורית (סריאל) המורכב משני מוליכים. הממשק נועד, במקורו, להעברת נתונים בין רכיבים במעגלים מוכללים שהותקנו במכשירי רדיו וטלוויזיה. כיום, השימוש העיקרי באפיק זה הוא כממשק תקשורת בין מיקרו-מעבד (או מיקרו-בקר) לבין רכיבי קצה. מהירויות העבודה האופייניות של הרכיב הן 100-3400 kbps.

תקן (פרוטוקול) I²C מגדיר את נוהל התקשורת בין הרכיבים המחוברים לאפיק, לרבות תדר שידור המידע (data), תזמון התקשורת בין הרכיבים ומבנה יחידת המידע המשודרת. שני המוליכים באפיק הם דו-כיווניים ולכל אחד מהם תפקיד מוגדר:

- קו מידע (DataSerial-SDA) שנועד להעברת מידע בין רכיבים. כל אחד מהרכיבים המחוברים לאפיק זה יכול לשמש כמשדר או כמקלט.
- קו שעון (ClockSerial-SCL) שנועד לסנכרן את המידע העובר בין רכיבים.

יחידת המידע הבסיסית המשודרת מרכיב לרכיב היא מילה באורך של 8 סיביות. לכל אחד מהרכיבים המחוברים לאפיק משויכת כתובת הנקבעת על-ידי סוג הרכיב המחובר (מיקרו-בקר, חיישן, התקן זיכרון).

התקן מגדיר שני סוגי רכיבים שניתן לחבר לאפיק רכיבי MASTER ורכיבי SLAVE. רכיב MASTER מוגדר כרכיב שיוזם את התקשורת ומספק את אות השעון. הרכיב שאליו פונה ה- MASTER מוגדר כרכיב ה- SLAVE. בדרך כלל ה- MASTER הוא מיקרו-בקר (או מיקרו-מעבד), בעוד שה- SLAVE הוא רכיב קצה כדוגמת, חיישן, רכיב תצוגה, התקן זיכרון וממיר ספרתי לאנלוגי.



תמונה 3 - תיאור של אפיק I²C

2.4 GSM

GSM או Global System for Mobile Communications הוא תקן לרשתות תקשורת סלולרית. רשתות אלו הן הנפוצות ביותר בעולם. ב־2010 היו בעולם כחמישה מיליארד מנויי GSM.

GSM עושה שימוש בטכנולוגית גישה מרובה מבוססת זמן, Time Division Multiple Access. הקול מועבר באופן דיגיטלי ולכן רשתות GSM מקוטלגות כ-"דור שני". הרשת הבסיסית שמותאמת לשיחות קוליות בלבד הורחבה באמצעות שתי טכנולוגיות משלימות להעברת נתונים: GPRS ו-EDGE. טכנולוגיות אלו מכונות "דור 2.5" ו-"דור 2.75" בהתאמה.













GSM תומך בטווח רחב של תדרים. באופן כללי בטווח שבין 260 MHz ל-1900 MHz. GSM מגדיר לא רק את ממשק השיחה אלא את התשתית כולה, זאת כדי לאפשר תאימות בין יצרנים שונים וכן לאפשר ממסר בין תחנות ונדידה - קבלת שירות עבור המכשיר או כרטיס ה-SIM בין מדינות שונות.

2.5 רמת אטימות IP

רמת אטימות IP (גם דרגת IP) קיצור של International Protection Marking היא סיווג של מידת ההגנה והאטימות של מכשיר מסוים מפני חלקיקים (אבק) ומפני מים. הסיווג מורכב מצמד האותיות האנגליות IP ומשתי ספרות, המוגדרות על ידי תקן IEC 60529 של הנציבות הבינלאומית לאלקטרוטכניקה.

הספרה הראשונה מציינת הגנה נגד מגע מקרי או מגע רשלני בחלקים חיים או נעים בתוך המעטפת וגם חדירת גופים מוצקים לתוך המעטפת. הספרה שנייה מציינת את דרגת ההגנה מפני חדירת מים, דרך המעטפת.

IP (Ingress Protection) Ratings Guide

SOLIDS		WATER	
1	 Protected against a solid object greater than 50 mm such as a hand.	1	 Protected against vertically falling drops of water. Limited ingress permitted.
2	 Protected against a solid object greater than 12.5 mm such as a finger.	2	 Protected against vertically falling drops of water with enclosure tilted up to 15 degrees from the vertical. Limited ingress permitted.
3	 Protected against a solid object greater than 2.5 mm such as a screwdriver.	3	 Protected against sprays of water up to 60 degrees from the vertical. Limited ingress permitted for three minutes.
4	 Protected against a solid object greater than 1 mm such as a wire.	4	 Protected against water splashed from all directions. Limited ingress permitted.
5	 Dust Protected. Limited ingress of dust permitted. Will not interfere with operation of the equipment. Two to eight hours.	5	 Protected against jets of water. Limited ingress permitted.
6	 Dust tight. No ingress of dust. Two to eight hours.	6	 Water from heavy seas or water projected in powerful jets shall not enter the enclosure in harmful quantities.

תמונה 4 - דרגות ההגנה שמספקות מעטפות

לדוגמא:

IP68 מצוין הגנה מוחלטת מפני חדירת אבק ובנוסף הגנה בפני חדירת מים בעת ממושכת, בתנאים שפרט היצרן.

2.6. חוק ארכימדס

כאשר גוף נמצא בנוזל, הוא עשוי לצוף על פני הנוזל או לשקוע בתוכו. החוק מראה כי הגוף צף אם הצפיפות שלו (היחס בין מסתו לנפחו) קטנה מזו של הנוזל. אם הצפיפות של הגוף גדולה מצפיפות הנוזל, הגוף שוקע. אם הצפיפויות שוות, הגוף ירחף בנוזל, כלומר ישהה בגובה שבו נשים אותו.

קל להסביר את הממצאים הללו בעזרת תכונות הלחץ ההידרוסטטי. כאשר הגוף נמצא בתוך הנוזל פועל עליו לחץ הידורו סטטי מכל הצדדים. אולם מכיוון שהלחץ גדל עם העומק, הלחץ ההידרוסטטי על צידו התחתון של הגוף גדול יותר מאשר על צידו העליון. לכן פועל על הגוף כוח שקול בכיוון מעלה, המכונה כוח עילוי.

את גודלו של כוח העילוי נחשב כך: נניח שאת מקום הגוף הייתה תופסת כמות של נוזל, שנפחה וצורתה זהים לאלה של הגוף. הנוזל מסביב היה מפעיל עליה כוחות זהים לאלה שהוא מפעיל על הגוף. מכיוון שכמות הנוזל הייתה נמצאת במנוחה, אפשר להסיק כי השקול של הכוחות הללו שווה בגודלו למשקל של כמות הנוזל, שהוא כידוע כוח בגודל, mg שכיוונו כלפי מטה. מכאן שכוח העילוי על הגוף הוא, mg כאשר m היא המסה של כמות נוזל, שנפחה שווה לנפח הגוף. אם נפח הגוף הוא, V וצפיפות הנוזל היא ρ , נקבל $m = V\rho$ כוח העילוי נתון לכן על ידי:

$$F = V\rho g$$

משוואה 1 - כוח העילוי

כאשר V הוא נפח הגוף ו- ρ היא צפיפות הנוזל.

ארכימדס, שחי במאה השלישית לפנה"ס, גילה חוק זה לראשונה וניסח אותו כך:

"על גוף השקוע בנוזל פועל כוח עילוי השווה למשקל הנוזל שנדחה על ידי הגוף."

עלינו לזכור כי על הגוף פועל גם משקלו שלו, ולכן הכוח השקול הוא:

$$\sum F = V\rho g - V\rho'g = Vg - Vg(\rho - \rho')$$

משוואה 2 - הכוח השקול על גוף

ρ היא צפיפות הנוזל ו- ρ' היא צפיפות הגוף.

אם $\rho' < \rho$, כוח העילוי גובר על המשקל והכוח השקול מכוון כלפי מעלה. הגוף יצוף עד שחלקו יבצבץ מהמים, באופן שכוח העילוי על החלק השקוע יאזן את משקל הגוף כולו.

אם $\rho' > \rho$, המשקל גובר על כוח העילוי והגוף ישקע עד לקרקעית.

אם $\rho' = \rho$, כוח העילוי שווה למשקל והגוף ירחף.

2.7. לחץ אטמוספרי

לעיתים נקרא גם לחץ ברומטרי הוא הלחץ, שמפעיל האוויר באטמוספירה של כדור הארץ. גודלו של הלחץ האטמוספרי הממוצע בגובה פני הים הוא :

$$1_{atm} = 760_{mmHg} = 1.013_{bar} = 101,325 Pa$$

בקירוב, ניתן לומר, כי הלחץ האטמוספרי שקול ללחץ ההידרוסטטי, שמפעיל עמוד האוויר מעל כל נקודה בכדור הארץ. כך, מעל אזורי לחץ נמוך עמוד האוויר נמוך יותר מאשר באזורי לחץ גבוה. באופן דומה, ניתן להבין מדוע לחץ האוויר יורד, ככל שעולים גבוה יותר מעל פני הים.

הבדלים בלחץ האוויר נוצרים מהבדלים בקרינת השמש, בגובה ובקו הרוחב של האזור. המכשיר המודד לחץ אוויר נקרא ברומטר, ויחידות המידה בהן משתמשים למטרה זו הן מ"מ כספית ($mmHg$) או בר (bar) או אטמוספירה (ATM)

2.8. בלוני ציפה

רוב יצרני סירות הגומי או אביזרים מתנפחים מייצרים את סירותיהם מבד ניילון מצופה פוליאוריתן (TPU). שילוב חומרים זה חזק, אטום לאוויר, עמיד למים, עמיד בפני קרינה אולטרה סגולה, אינו נמתח הרבה ושומר על חוזקו גם בטמפרטורות נמוכות מאוד.

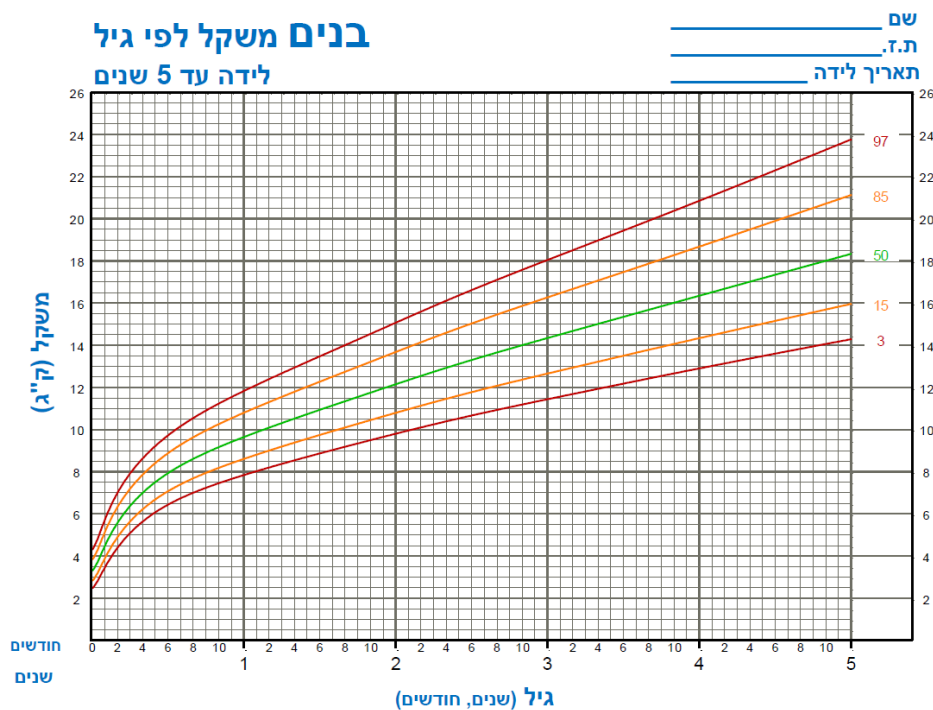
פוליאוריתן תרמו-פלסטי (TPU) הוא גרסה של PU הנקשרת לעצמה כשהיא מחוממת עם אקדח ברזל או אוויר חם. זה יוצר קשר חזק מאוד וקבוע, ומבטל את הצורך בתפירה או הדבקה.

אחת הדרכים לסיווג הבד היא על ידי (D) "denier", שהוא מדד למשקל החוט, כאשר denier אחד שווה למשקל של חוט משי בודד, ומספרים של denier גבוהים יותר מציינים חוט כבד יותר (ולכן עבה וחזק יותר). הקשר בין מספר denier למשקל וחוזק הבד אינו ליניארי.

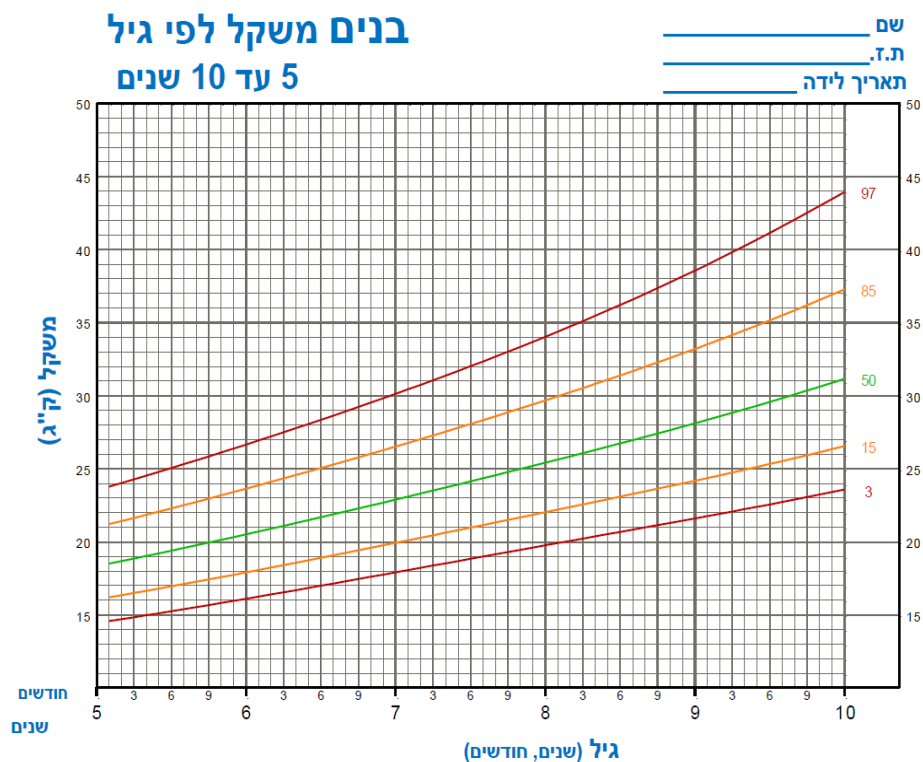
לאחר קריאה ממושכת באינטרנט מצאנו כי עובי הבד הנכון לשימוש הוא 210D כיוון שהוא ממש קל, מאפשר לקפל אותו, לא יקרע / ינוקב בקלות ומשתמשים בו רבות באביזרי ציפה למיניהם.

2.9. נפח בלוני ציפה

לחישוב הנפח הדרוש לניפוח בלוני הציפה (לפי חוק ארכימדס), נעשה שימוש בגרפים המתארים משקל לעומת גיל של תינוקות / ילדים. המשקל הממוצע העליון נלקח בכל קטגוריית גיל, מכיוון שזה המקרה הקיצון ביותר וייתן מענה לכלל המשקלים האפשריים.



תמונה 5 - טווח משקלים של תינוקות וילדים עד גיל 5



תמונה 6 - טווח משקלים של תינוקות וילדים מגיל 5 עד 10

³ הנתונים נלקחו ממשרד הבריאות העולמי WHO

מגרפים הנ"ל ניתן לבנות את הטבלה הבאה :

גיל (שנים)	משקל ממוצע (ק"ג, קיצון עליון)	נפח הבלון הדרוש (2 הבלונים ביחד)
1	11.8	0.52
2	15	0.72
3	18	0.86
4	20.8	0.99
5	24	1.15
6	26.8	1.28
7	30	1.43
8	34	1.62

טבלה 1 - גיל, משקל ממוצע ונפח הבלון הדרוש

לחישוב נפח הבלון הדרוש נעשה שימוש בנוסחת ארכימדס (משוואה 2) ובנתונים הבאים :

סוג חומר	צפיפות
מי בריכה	$1 \frac{g}{cm^3}$
מי ים	$1.03 \frac{g}{cm^3}$
גוף האדם	$0.985 \frac{g}{cm^3}$

טבלה 2 - צפיפות לפי סוג החומר

2.10. משוואת הגז האידיאלי

גז אידיאלי הינו גז אשר מורכב ממולקולות נקודתיות בעלי נפח זניח, שאינן מגיבות זו עם זו. הנחה זו הינה הכרחית לחישובים פשוטים של טמפרטורה, לחץ ונפח הגז. משוואת הגז האידיאלי מורכבת משלושה חוקים חשובים : חוק בוייל, חוק צ'ארלס וחוק הגז המשוכלב. שלושת החוקים הללו מגדירים את הקשר בין נפח, לחץ וטמפרטורה של הגז.

חוק בוייל מגדיר את הקשר בין לחץ לנפח. ככל שהנפח גדל, הלחץ קטן משום שההסתברות של מולקולות הגז להתנגש בדפנות הכלי יורד. חוק צ'ארלס מגדיר את הקשר בין טמפרטורה לנפח. ככל שהטמפרטורה עולה, גם הנפח עולה, משום שמולקולות הגז מקבלות אנרגיה חום המתורגמת לאנרגיה קינטית הגורמת להן לנוע מהר יותר, והלחץ גדל. כאשר הלחץ גדל, הוא מאפשר את הגדלת הנפח על חשבון הלחץ. חוק הגז המשוכלב משלב בין שני החוקים הנ"ל ומראה את יחסי הגומלין בין טמפרטורה, לחץ ונפח. וחוק הגזים האידיאליים מכניס את החוקים הללו למשוואה :

$$V = \frac{nRT}{P}$$

משוואה 3 - משוואת הגזים האידיאלי

כאשר P - לחץ, T - טמפרטורה, V - נפח, R - קבוע הגזים $(0.082 \frac{L \times atm}{K \times mol})$, n - מספר המולים

2.11. פחמן דו חמצני (CO₂)

פחמן דו חמצני, נקרא גם דו-תחמוצת הפחמן, הוא גז (בטמפרטורת החדר) המהווה תרכובת של פחמן וחמצן. כל מולקולה שלו מורכבת מאטום פחמן אחד ושני אטומי חמצן, הקשורים אליו בקשר קוולנטי כפול. זהו גז חסר צבע אשר מצב הצבירה שלו קרוי קרח יבש, בזכות תכונת ההמראה המאפיינת אותו, כלומר מעבר ישיר ממצב מוצק למצב גז. בעל מסה מולרית של 44.0095 גרם/מול⁵. שכיחותו באטמוספירה נמוכה - רק 0.035 אחוזים. פחמן דו חמצני הוא התחמוצת המצויה ביותר בעולם.

לצורך מילוי בלוני הציפה, נדרש גז שניתן לאחסן ולטפל בו בקלות. כמו כן, הוא חייב להיות בטוח לשימוש ולא נפיץ. לאחר סקירת ספרות בחרנו בפחמן דו חמצני כגז שימלא את בלוני הציפה מכיוון שצפיפותו היא מאוד נמוכה ביחס לגזים האחרים. בנוסף זה גז שמאוכסן במחסנית במצב נוזל, ככה שבעת שחרורו הוא תופס הרבה יותר נפח כגז.

פחמן דו-חמצני	
	
פחמן דו-חמצני	
	
פחמן דו-חמצני	שם סיסטמטי
פחמן דו-חמצני	שמות נוספים
קרח יבש (מוצק)	נוסחה כימית
CO ₂	מסה מולרית
44.0095 גרם/מול	מראה
אינו נראה (חסר צבע)	מספר CAS
124-38-9	צפיפות
0.001977 (גז) 1.562 (מוצק)	מצב צבירה
גרם/סמ"ק	מסיסות
גז (נפלט בתהליך הנשימה של בעלי החיים)	טמפרטורת המראה
g/L 1.45	K 194.65
C° -78.5	חומציות
pK _a 10.33 / 6.35	

תמונה 7 - תכונות הפחמן הדו חמצני

מחסניות ה- CO₂ משמשות למגוון רחב של שימושים כגון: נשקי איירסופט / פיינטבול, ניפוח בלוני ציפה בעלי נפח משתנה וכל המשקאות המוגזים כגון סודה-סטרס.



תמונה 8 - מחסנית CO₂ 16 גרם

לחישוב כמות הגז המשתחררת בעת פתיחת המחסנית משתמשים במשוואת הגז האידיאלי. לדוגמה עבור מחסנית CO₂ בגודל 16 גרם נקבל:

$$V = \frac{nRT}{P} = \frac{16}{44.0095} \times 0.082 \times 295.4}{1.06351} = 8.2[L]$$

משוואה 4 - נפח הגז הנפלט ממחסנית 16 גרם

כאשר T הוא הטמפר' הממוצעת השנתית בים ו- P הוא הלחץ בעומק 0.5 מטר.

⁵ מול (Mole) הוא יחידת מידה סטנדרטית המגדירה כמות חומר על פי מספר חלקיקים קבוע.

3.1. דרישות עיקריות

דרישה עיקרית מהמערכת, היא היכולת להרים את ראשו של הילד מעל פני המים ולשמור עליו במצב זה. כאשר יינתן מענה לדרישה זו יימנע מגע של מים עם דרכי הנשימה שמוביל לחנק.

דרישה זו נובעת בגלל שראשם של תינוקות וילדים עד גיל 5 כבד יותר משאר חלקי הגוף. זאת הסיבה שילדים מתהפכים ונותרים עם הראש בתוך המים, מבלי יכולת עצמאית להתהפך. אם פיו ואפו של הילד נמצאים מתחת לפני המים לזמן של שתי דקות הוא עלול לאבד את הכרתו, ולאחר מכן קיים סיכון של נזק מוחי בלתי הפיך ועד למוות.

דרישה עיקרית נוספת מהמערכת היא שליחת הודעת SMS להורים על ידי מרכז בקרה, לקבלת תשומת הלב שלהם. דרישה זו נובעת מכיוון שהסיבה העיקרית לטביעת תינוקות וילדים היא העדר השגחה או נתק קצר בהשגחה. יש גורמים רבים המסיחים את דעת ההורים ופוגעים במידת השגחתם על ילדיהם, כמו קריאה, שיחה עם אנשים וכמובן גם שימוש בטלפונים ניידים.

3.1.1. דרישות ספציפיות

דרישות חיצוניות - המערכת כולה צריכה להיות קלה ונוחה לשימוש מכיוון שמדובר בפריט לביש שמיועד לשבת על צוואר הילד. מכך עולה הדרישה להשתמש בחומרים קלים שלא יכבידו על צוואר הילד. בנוסף החומרים צריכים להיות עמידים למים וכמובן לשימוש הילדים והורים. בנוסף המערכת מיועדת להיות במגע עם מים בעת שימוש, באזורי בריכה או ים, לכן קיימת דרישה שהמערכת החשמלית תהיה במארז עמיד למים (IP67). מארז זה יתן גם מענה לתנאי מזג אוויר קיצוניים אשר עלולים להשפיע על החומרה ולפגוע בה.

דרישות פנימיות - המערכת החשמלית חייבת להיות כמה שיותר אמינה ולפעול עם כמה שפחות תקלות. כלומר צריך לבחור רכיבים חשמליים שהוכיחו את אמינותם ודיוקם בפרויקטים אחרים ובשוק הכללי. רכיבי המערכת החשמלית צריכים להיות כמה שיותר קטנים וקלים כדי שהמוצר כולו יהיה נוח ולא כבד מידי לילד.

בנוסף המערכת החשמלית תקבל מתח ע"י סוללה (LiPo 3.7V) קטנה ולכן הרכיבים בה, יהיו רכיבים בעלי צריכת אנרגיה נמוכה ככל הניתן. הדרישה היא שהמערכת תוכל לעבוד ללא טעינה כשבוע ימים כאשר היא דלוקה (מצב שינה).

דרישות פונקציונליות (תוכנה) -

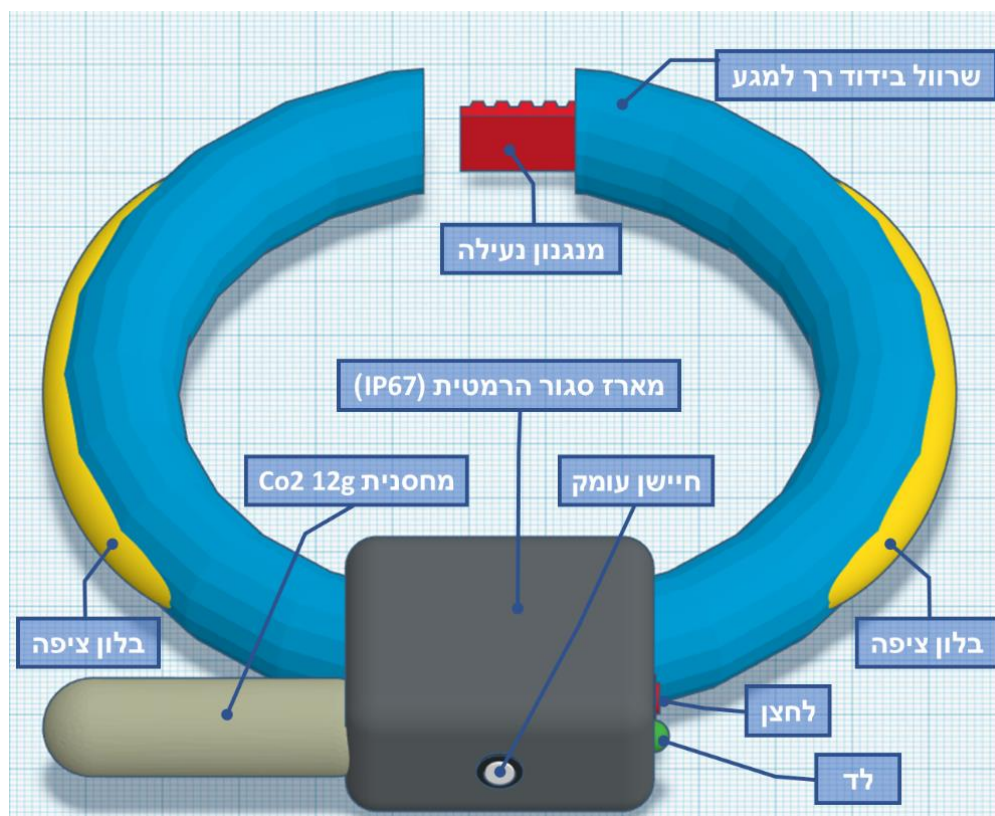
- בדיקת תקינות המערכת באופן אוטומטי בעת הדלקת המערכת. במטרה לבדוק את תקינות הרכיבים, התקשורת ביניהם ואת אחוז הסוללה. להגיב בהתאם לנדרש לאחר הבדיקות.
- מצב שינה: בעת הדלקת המערכת לאחר בדיקת התקינות, היא תכנס למצב שינה. ובכך צריכת אנרגיה נמוכה ע"י זמן שינה ארוך ככל האפשר.
- קבלה ועיבוד נתוני החיישנים: תקשורת רציפה ואמינה בין המיקרו-בקר לחיישן המים והעומק ובנוסף לרכיב GSM.
- זיהוי עומק סף: כתיבת אלגוריתם לזיהוי עומק שיוגדר מראש ובדיקה אם הערך שהתקבל אחרי ניתוח של תוצאות החיישנים על ידי הארדוואינו חוצה את הסף.
- הפעלת טיימר לזמן מוגדר מראש שיבדוק אם הילד לא יוצא מהעומק סף במסגרת הזמן הזה.
- אם זוהתה טביעה, יופעל המנוע לסיבוב הווסת במטרה לשחרר את הגז מהמחסנית ובכך למלא את בלוני הציפה.
- שליחת הודעת SMS למרכז הבקרה על מקרה הטביעה, על מנת שיוכלו לפעול במהירות לשליחת הגורמים הרלוונטיים (הורים, כוחות הצלה) למיקום האירוע.
- בדיקת תקינות המערכת באופן יזום על ידי לחיצה על כפתור בדיקת תקינות. במטרה לבדוק את תקינות הרכיבים, התקשורת ביניהם ואת אחוז הסוללה. להגיב כנדרש לאחר הבדיקות.
- מבחינת עיצוב תוכנה ומבחינת קוד צריכה להיות בנויה וכתובה בצורה ג'נרית ומודולרית, כך שתאפשר מרחב ומערך שיפורים עתידיים.

3.2. תיאור המערכת

3.2.1. מבנה חיצוני

המוצר הסופי יהיה בנויה בצורת שרשרת אשר יולבש על צוואר הילד. משני צידי השרשרת יהיו בלוני ציפה מקופלים לתוך השרשרת עצמה. לשרשרת יחובר מארז עמיד למים (IP67) ובתוכה תהיה מערכת חשמלית.

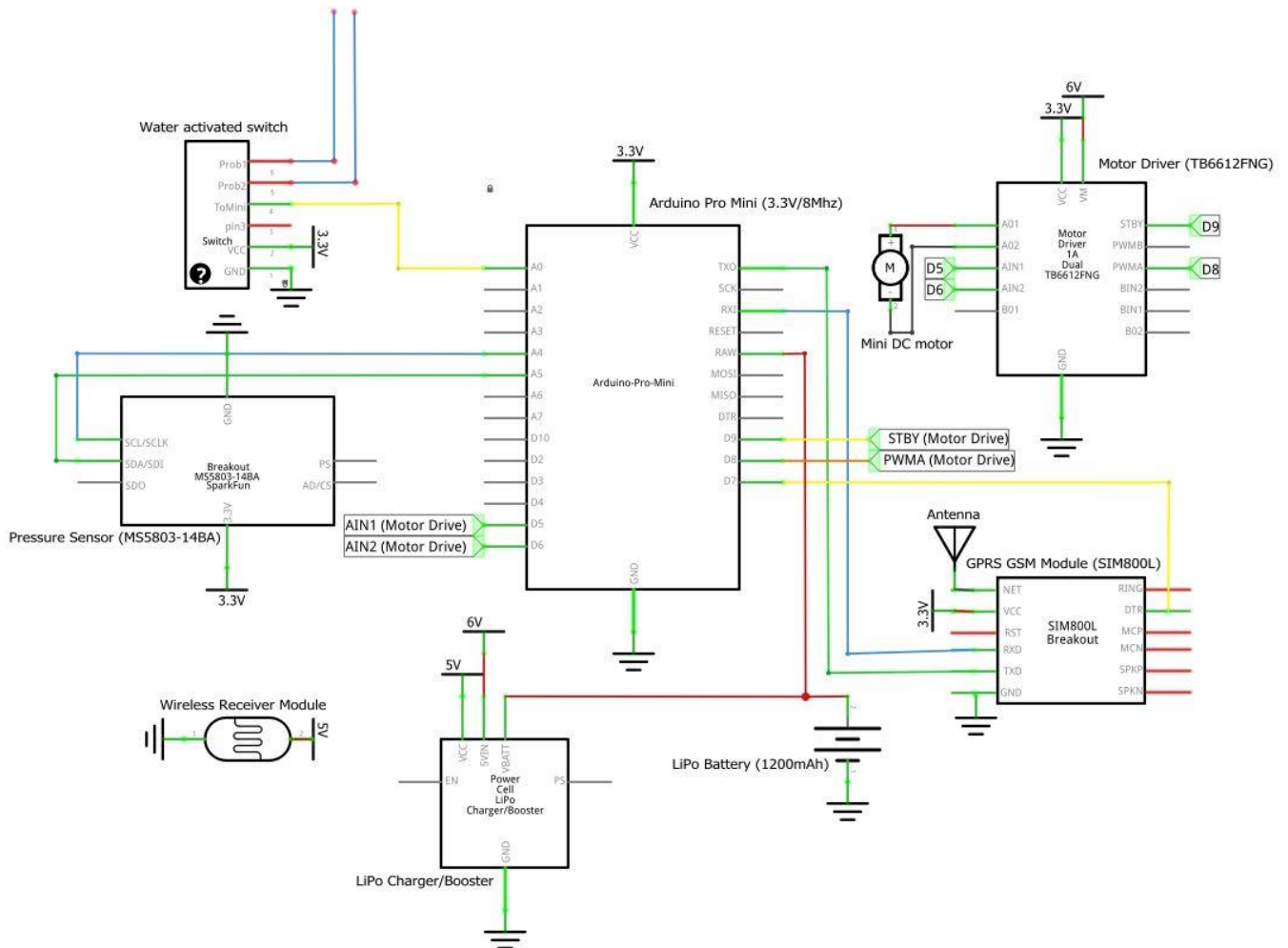
- המערכת החשמלית תדלק כאשר מנגנון הנעילה נסגר, ככה יימנע מצב שבו ההורים שוכחים להדליק את המערכת.
- כפי שצוין בחלק סקר הספרות (2.9) בלוני הציפה יהיו מבד עמיד מצופה פוליאוריתן (TPU). בלוני הציפה יהיו בנפח 1 ליטר בכל צד, לפי שצוין בחלק סקר הספרות (2.9)
- כפתור ההדלקה וכפתור בדיקת התקינות יהיו כפתורים עמידים למגע עם מים.
- השרשרת עצמה תהיה מורכבת משרוול בידוד דק, קל ורך למגע. זהו חומר לא דליק, לא רעיל ובעל ספיגת מים מזערית. חומר זה משמש לבידוד צינורות נחושת במערכות קירור וחימום.
- כפי שצוין בחלק סקר הספרות (2.6) ו-(2.8) נעשה שימוש במחסנית CO_2 בגודל 16 גרם במטרה לנפח את 2 בלוני הציפה.



תמונה 9 - מבנה חיצוני

3.2.2. מבנה פנימי

המערכת משלבת חומרה שנבחרה בקפידה כדי לבצע בצורה טובה ואמינה את המטרות והדרישות שהוגדרו. המערכת החשמלית תהיה אחראית לזיהוי הטביעה ולהפעלת המנגנונים השונים להצלת הילד במקרה הצורך.



תמונה 10 - ארכיטקטורה פנימית

* מלבד זאת נמצאים במעגל גם לחצן הדלקה/כיבוי ולחצן בדיקת תקינות, נגדים ו- LED.

3.2.3. פירוט הרכיבים החשמליים במערכת החשמלית

מיקרו-בקר

כל הרכיבים החשמליים יהיו מחוברים ליחידת מיקרו-בקר. בפרויקט זה הוחלט לעבוד עם ארדואינו כי הוא כלי פיתוח ופלטפורמה בקוד פתוח שכולל סביבת פיתוח קלה ונוחה למשתמש.

בפרויקט נעשה שימוש ברכיבים כמה שיותר קטנים, בעלי משקל נמוך ובעלי צריכה אנרגיה נמוכה. כלומר קיים צורך להשתמש במיקרו-בקר שיהיה קטן, קומפקטי וחסכוני באנרגיה.

אופציה עיקרית שעונה על צרכי הפרויקט הן מבחנית מקום וכן מבחנית חסכון באנרגיה היא הארדואינו פרו-מיני (Arduino Pro Mini) [או חיקויים דומים במבנה וביכולות].

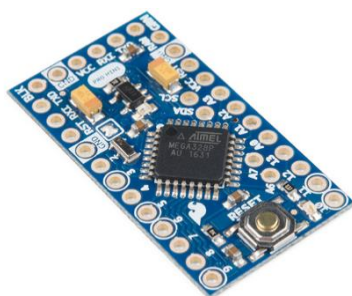
הוא כולל את כל החיבורים הדרושים לנו לחיבור שאר הרכיבים החשמליים.

Name	Arduino Pro Mini
Microcontroller	ATmega328
Operating Voltage	3.3 V
Input Voltage (recommended)	3.3-12 V
Digital I/O Pins	14
PWM Channels	6
Analog Input Channels	8
DC Current per I/O Pin	40 mA
Flash Memory	32 KB
SRAM	2 KB
Clock Speed	8 MHz
Power Consumption	ACT - 5 mA, PDS - 0.9 mA
Length	33 mm
Width	18 mm
Weight	2 g

טבלה 3 - מאפייני המיקרו בקר מתוך דף המפרט

* ACT - Active Mode

* PDS - Power-Down Sleep with Watchdog Timer enabled



תמונה 11 - Arduino Pro Mini

חיישן לחץ / עומק (Altimeter Pressure Sensor)

חיישן העומק צריך לקיים כמה דרישות:

- תחום עבודה (Pressure range) - בפרויקט זה לא דרושה עמידה בלחצים גבוהים (עומק רב) מכיוון שמטרת הפרויקט הינה הצלת ילדים מטביעה בבריכה או בים. כלומר תחום העבודה הדרוש הוא עד ± 5 מטר. אין לנו צורך בחיישן שיועד לעבוד בעומק רב.
- דיוק (Accuracy) - מכיוון שהתחום עבודה הוא קטן, הדיוק והרזולוציה חשובים לזיהוי נכון של העומק, כלומר על החיישן להיות בעל דיוק ורזולוציה גבוהה ככל הניתן. בחיישנים טובים ניתן להגיע לדיוק של מ"מ בודדים.
- המרה מהירה - מכיוון שמדובר על חיי אדם, כל שניה חשובה. מהירות הזיהוי של ילד שטובע היא הכרחית. לכן קיימת דרישה שהחיישן ידע להמיר במהירות את מתח היציאה האנלוגי הלא מקוזז מחיישן הלחץ הפיאו-התנגדותי לערך דיגיטלי.
- חיישן אבסולוטי (Absolute) - חיישנים מסוג זה בעלי מקדמים פנימיים מכוילים במפעל, כלומר ניתן להשתמש בהם ללא חיישן נוסף מחוץ למים לקבל נתוני ייחוס.
- תקשורת מהירה ונוחה לשימוש שמתאימה למיקרו בקר כגון הארדואינו פרו מיני.

לאחר בדיקה מעמיקה ברחבי האינטרנט נמצא כי קיימת משפחת חיישנים בשם MS5803 של חברת TE⁶ שעונה על צרכי הפרויקט. ההבדל בין החיישנים במשפחה הוא הטווח לחץ שבוא ניתן להשתמש בחיישן. קיים חיישן עם טווח לחץ של עד 1 bar או עד 5 bar וכך הלאה.

בגלל בעיה של זמינות נרכש את החיישן שטווח העבודה שלו הוא עד 14 bar.

ולכן החיישן איתו מתבצעת העבודה הוא [MS5803-14BA](#)

FEATURES:

Integrated pressure sensor	6.2 x 6.4 mm
Pressure range	0 - 14 bar
Resolution	1 / 0.6 / 0.4 / 0.3 / 0.2 mbar
Accuracy	± 20 mbar
ADC	24 Bit
Output Interface	I ² C and SPI interface
Clock line	32.768 kHz
Operating Temperature	-40 – 85 °C
Supply Voltage (V)	1.8 – 3.6
Power consumption	Operating current < 1uA Standby current < 0.15uA

טבלה 4 - מאפייני החיישן מתוך דף המפרט



תמונה 12 - MS5803-14BA

⁶ Tyco Electronics - TE Connectivity Ltd. (TEL: NYSE), לשעבר Tyco Electronics

מנוע זרם ישר (DC Motor)

הדרישה העיקרית מהמנוע בפרויקט היא שהמנוע יהיה בעל כוח זוויתי (מומנט - torque) מספיק חזק בשביל לסובב את השסתום שמיועד לסגירת מעבר או לוויסות הזרימה של הגז.

כמו כן דרישה חשובה נוספת המשותפת לכל רכיבי המערכת היא הגודל הרכיב. לכן המנוע יהיה קטן בממדיו.

צריכת אנרגיה של המנוע פחות חשובה לנו מכיוון שהמנוע יפעל רק במקרה חירום. רק היה חשוב לוודא שהסוללה תוכל לתת לו את הזרם הדרוש להתנעה.

בעקבות דרישות אלה בחרו לעבוד עם מנוע זרם ישר (DC Motor) קטן בעל מערכת תמסורת פנימית של גלגלי שיניים לאחר שנבדקה אופציה נוספת (מנוע סרוו). מנוע הסרוו שעמד בדרישות הגודל לא היה בעל כוח זוויתי מספיק לפתיחת השסתום.

מאפיינים:



תמונה 13 - Mini geared DC motor 6V

22 rpm	מהירות סיבוב
6V	מתח עבודה
40 mA	צריכת זרם ללא עומס
350 mA	צריכת זרם עם עומס
670 mA	זרם תקיעה
6.5 kg-cm	כוח סיבוב (מומנט)
10 גרם	משקל

טבלה 5 - מאפייני המנוע מתוך דף המפרט

בקר מנוע זרם ישר

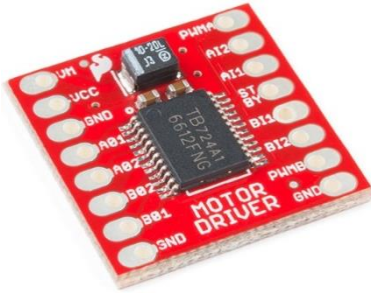
גם בבחירת רכיב זה הייתה דרישה של גודל רכיב קטן ככל האפשר. בנוסף הרכיב צריך להיות בעל צריכת אנרגיה נמוכה מאוד במצב שינה, כי במצב זה הוא יהיה כל הזמן, למעט מקרה טביעה.

לכן הרכיב שנבחר לצורך שליטה בפעולת המנוע הוא רכיב TB6612FNG של חברת Toshiba.

רכיב זה דוחף זרם לשני מנועים שיכול לספק זרם קבוע של עד 1.2A לכל ערוץ, או זרם רגעי של 3.2A. הוא כולל הגנות פנימיות מפני מתח נמוך, זרם גבוה או התחממות של הרכיב.

שתי כניסות IN1 ו-IN2 משמשות לבחירת כיוון הסיבוב בלימה או עצירה של המנוע.

מהירות הסיבוב נשלטת ע"י אות PWM בתדר של עד 100 KHz. אפשר לשלוט על כל ערוץ בנפרד. כדי שהרכיב יעבוד, צריך לעלות את כניסת ה-Standby ל-1 לוגי.



תמונה 14 - בקר מנוע TB6612FNG DC

מתח עבודה	5.5 - 2.7V
מתח כניסה	עד 15V
זרם רגעי מקסימלי	3.2A
צריכה זרם בעבודה	40 mA
צריכה זרם בשינה	1μA
תקשורת	PWM
אורך	20.3 מ"מ
רוחב	20.3 מ"מ

טבלה 6 - מאפייני בקר המנוע מתוך דף המפרט

מתאם תקשורת סלולרית (GPRS GSM Module)

הדרישה העיקרית מהרכיב, היא היכולת לשלוח הודעת חירום והודעה לבדיקת תקינות המערכת למרכז בקרה. תחום התדירים איתו עובד הרכיב צריך להתאים לתחום התדירים הנפוץ בישראל. אין צורך לקבל או לבצע שיחות בפרויקט זה.

כמו כן דרישה חשובה נוספת המשותפת לכל רכיבי המערכת היא הגודל. לכן הרכיב יהיה קטן בממדיו ולא יתפוס שטח גדול בתוך המארז האטום. בנוסף הרכיב צריך להיות בעל צריכת הספק נמוכה לשמירת הסוללה.

הרכיב העונה על צרכי הפרויקט ואיתו מתבצעת העבודה הוא SIM800L:

Features:

Quad-band 850/900/1800/1900MHz
Connect onto any global GSM network with any 2G SIM
Make and receive voice calls using a headset
Send and receive SMS messages
Send and receive GPRS data (TCP/IP, HTTP, etc.)
Lead out buzzer and vibrational motor control port
AT command interface with "auto baud" detection
Supply Voltage: 3.3 - 4 V
Current Consumption:
Stand-by: 15 mA
Sleep mode: 1 mA
SMS send: 60 mA

טבלה 7 - מאפייני רכיב ה-gsm מתוך דף המפרט



תמונה 15 - sim800l

ממיר מתח + מטען סוללות (DC/DC Boost Converter + Charger)

בשוק קיימים מעגלים שיוזעים להמיר מתח מ-3.7V (מתח הסוללה) למתח עבודה של 6V ובנוסף לשמש בתור מטען סוללות. היתרון של מעגלים אלה בפרויקט הוא כמובן החיסכון במקום. אין צורך ב-2 מעגלים נפרדים להגברת המתח ולהטענת הסוללה.

הרכיב העונה על צרכי הפרויקט ואיתו מתבצעת העבודה הוא DD05CVSA, ברכיב זה יש את בקר טעינה TP4056 IC וממיר מתח.

Charge voltage	DC 4.5V-8V
Charging current	0-1A
(Boost in) Discharge current	0-2A
(Boost in) Discharge quiescent current	450μA
(Boost out) Output current	0-1A (6V)

טבלה 8 - מאפייני רכיב ה ממיר מתח + מטען סוללות - מתוך דף המפרט



תמונה 16 - DC/DC Boost Converter + Charger

מקור מתח – סוללת ליתיום-פולימר (Lithium ion polymer - LiPo)

הסוללה היא מתקן לייצור חשמל בזרם ישר. הסוללה מורכבת מ"תא חשמלי" אחד או יותר. ה"תא החשמלי" הנפוץ כיום הוא תא אלקטרוכימי שבו החשמל מופק כתוצאה מתהליך כימי. "סוללה נטענת" משלבת למעשה תא אלקטרוכימי להפקת חשמל, ותא אלקטרוליטי ל"השבת" אנרגיה לסוללה באמצעות שינויים כימיים, לצורך המשך הפקת החשמל.

ישנם סוגים שונים של סוללות, אך בפרויקט זה העבודה מתבצעת עם סוללת ליתיום-פולימר (LiPo), מכיוון שהן סוללות קטנות ודקות במיוחד המיועדות למכשירים חשמליים קטנים. הן שכיחות בעיקר במכשירים אלקטרוניים ניידים, הודות לצפיפות האנרגיה הגבוהה שלהן, מספר מחזורי פריקה/טעינה גבוה ואיבוד נמוך של הקיבול במהלך המחזורים.

מחישוב משוער של צריכת הזרם של כל רכיבי המערכת, התקבל כי יש לעשות שימוש בסוללה של לפחות 1200mAh על מנת להפעיל את כל המערכת לתקופה של שבוע במצב שינה.



תמונה 17 - סוללת 3.7V LIPO

כמו כן סוללה בגודל כזה היא מספיק קטנה בשביל להיכנס למארז האטום למים.

רכיב לטעינה אלחוטית (Wireless Receiver Module)

המערכת החשמלית יהיו בתוך מארז עמיד למים (IP67) שיאפשר כניסה למים ללא חשש של הריסת המכשיר. כדי למנוע פתיחה וסגירה של המארז כל פעם בשביל להטעין את הסוללה, תתווסף למערכת אופציה לטעינה אלחוטית.

להוספת אופציה זאת נעשה שימוש במקלט טעינה אלחוטית אשר כולל סליל נחושת ובקר טעינה.



תמונה 18 - טעינה אלחוטית

3.2.4. כניסות / יציאות במערכת החשמלית

קלטיים במערכת החשמלית:

- קלט רכיב הזיהוי של מגע עם מים : אות מתח אנלוגי.
- קלט חיישן העומק / לחץ : אות מתח אנלוגי, יקבל מתח מהמיקרו בקר.
- קלט מתאם תקשורת סלולרית : GSM, יקבל מתח ישירות מהסוללה.
- קלט מנוע DC : יקבל מתח מהבקר מנוע.
- קלט ממיר מתח + מטען סוללות : אות מתח אנלוגי.
- קלט המיקרו בקר : אות אנלוגי מרכיב זיהוי המים ומלחצן הבדיקה.
תקשורת טורית סינכרונית (I^2C) מהחיישן עומק / לחץ.
תקשורת טורית אסינכרונית (UART) למתאם תקשורת
סלולרית.
יקבל מתח ישירות מהסוללה.

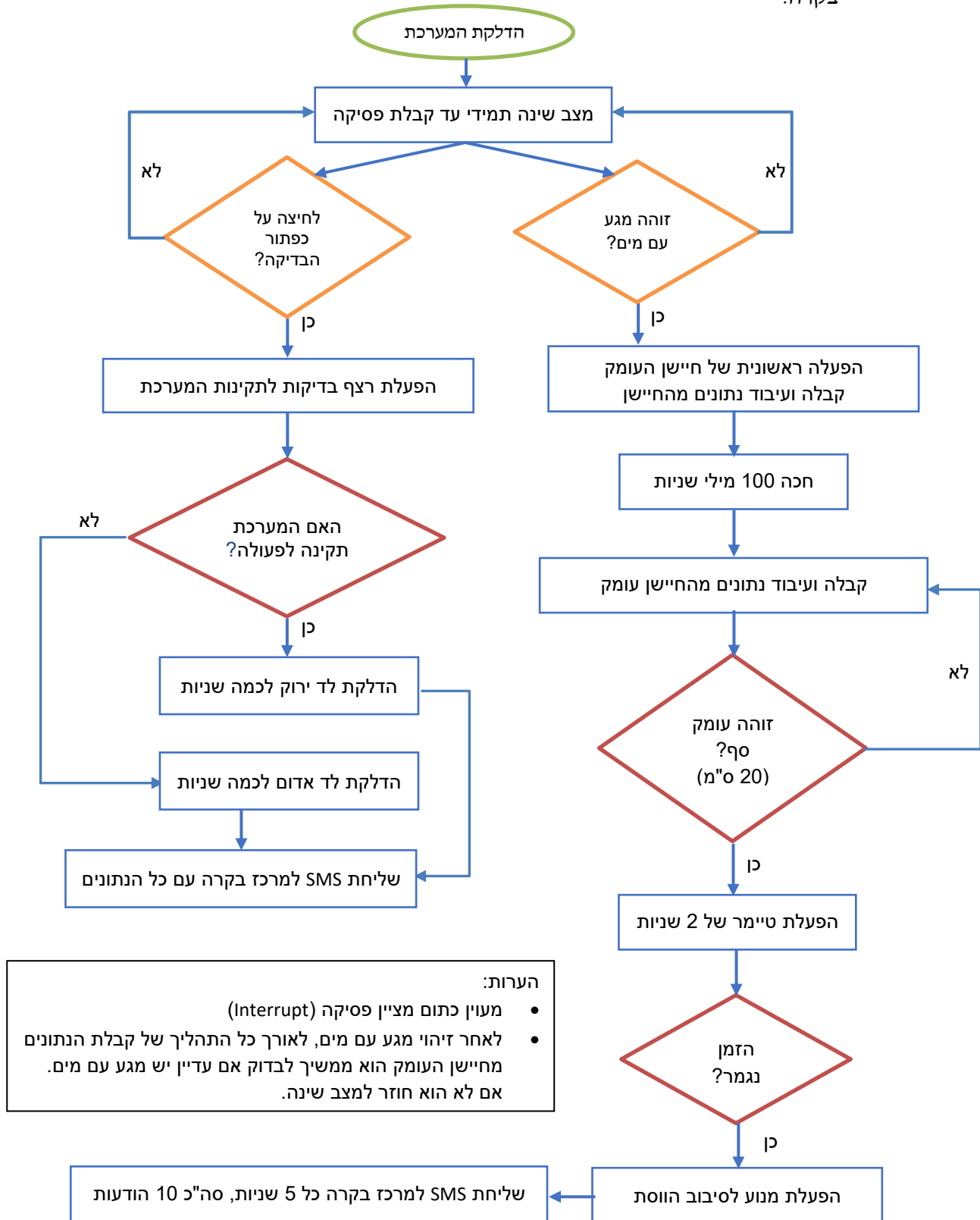
פלטים במערכת החשמלית:

- פלט רכיב הזיהוי של מגע עם מים : אות מתח אנלוגי שיועבר למיקרו בקר.
- פלט חיישן העומק / לחץ : תקשורת טורית סינכרונית (I^2C) בין החיישן למיקרו בקר.
- פלט מתאם תקשורת סלולרית : תקשורת טורית אסינכרונית (UART) הודעת SMS למרכז בקרה.
- פלט מנוע DC : סיבוב ציר המנוע - פתיחת ווסט ה-Co2.
- פלט ממיר מתח + מטען סוללות : אות מתח אנלוגי מוגבר.
- פלט המיקרו בקר : אות מתח דיגיטלי לבקר מנוע.
תקשורת טורית אסינכרונית (UART) למתאם תקשורת
סלולרית.
תקשורת טורית סינכרונית (I^2C) לחיישן עומק / לחץ.

3.3. מבנה הקוד (אלגוריתם)

המערכת משלבת תוכנה שנכתבה כדי לבצע את המטרות והדרישות שהוגדרו. החלק התכנותי במערכת, יהיה בקוד שכתוב בשפת ++C בסביבת הפיתוח Arduino Software IDE הצרוב על מיקרו-בקר.

האלגוריתם - שרשרת של פעולות ופקודות אשר לפיהן יעבדו רכיבי המערכת החשמלית. בקוד זה יהיה עיבוד האותות מהחיישנים, זיהוי טביעה ושליחת הודעת חירום למרכז בקרה.



3.3.1. פירוט האלגוריתם

הקוד המלא נמצא בנספח מספר 1.

האלגוריתם בנוי מ- 5 ספריות עזר:

- **SoftwareSerial** - מאפשר תקשורת טורית על כניסות דיגיטליות אחרות של הארדואינו, עבור רכיב SIM800L (ספריה מובנית).
- **Wire** - תקשורת עם I²C, עבור חיישן הלחץ MS5803-14BA (ספריה מובנית).
- **LowPower** - ספריית לניהול מצב השינה של המיקרו בקר.
- **SparkFun_MS5803_I2C** - מאפשר לדבר בקלות עם חיישן הלחץ MS5803-14BA.
- **Battery** - מאפשר פיקוח על צריכת הסוללה.

האלגוריתם בנוי מ- 4 פונקציות עיקריות, 2 פונקציות פסיקה ושאר הפונקציות משמשות כפונקציות עזר.

הפונקציות העיקריות הן:

- **void setup()**

פונקציה זו נקראת כאשר המיקרו בקר מתחיל להריץ את הקוד. אנו משתמשים בפונקציה זו להתחיל משתנים, מצבי כניסות / יציאות, להתחיל להשתמש בספריות, הכנסה למצב שינה וכדומה. בנוסף פונקציה זו עושה בדיקת תקינות לרכיבי המערכת באופן אוטומטי. לאחר הבדיקה שולחת הודעה עם פרטי הבדיקה למרכז בקרה.

קלט: אין

פלט: אין

- **void loop()**

הפונקציה עושה בדיוק את מה ששמה מרמז, לולאה אינסופית. היא מאפשרת / משביתה פסיקות, מפעילה מצב השינה, מפעילה מנגנוני זיהוי טביעה ומפעילה רצף בדיקת המערכת.

קלט: אין

פלט: אין

- **void In_water()**

פונקציה זו נקראת על ידי loop() לאחר סיום פונקציית הפסיקה המתאימה. מודד לחץ ראשוני ולאחר פרק זמן מסוים מודד שוב ואז שולח את הנתונים לפונקציה Check_depth() לחישוב הפרש הגובה. אם הפרש הגובה הוא מתחת לסף (לזמן מוגדר מראש), הפונקציה Drowning_detected() מופעלת. אחרת, הפונקציה מסתיימת וחוזרת לפונקציית הלולאה loop().

קלט : אין

פלט : אין

- **void Check_device()**

פונקציה זו נקראת על ידי loop() לאחר סיום פונקציית הפסיקה המתאימה (זוהתה לחיצה על כפתור הבדיקה). בודקת את כל רכיבי המערכת ולאחר מכן שולחת הודעת SMS מתאימה למרכז בקרה. אם כל הרכיבים תקינים ואחוז הסוללה הוא בטווח המתאים נדלק לד ירוק, אחרת אדום.

קלט : אין

פלט : אין

פונקציות הפסיקה הן:

- **void Water_activated_switch()**

פונקציה זו נקראת כאשר יש עליה בפין מספר 2 (פונקציית פסיקה). היא מעבירה את הדגל InWater לנכון, במטרה לסמן שהמערכת מזהה מגע עם מים.

קלט : אין

פלט : אין

- **void Button_activated_switch()**

פונקציה זו נקראת כאשר יש ירידה בפין מספר 3 (פונקציית פסיקה). היא מעבירה את הדגל Check_dev לנכון, במטרה לסמן שהמערכת מבצעת בדיקה לרכיבי המערכת.

קלט : אין

פלט : אין

- **double Check_depth(double pressure_baseline())**

פונקציה זו נקראת על ידי In_water() כאשר נדרש לחשב את דלתא הגובה. מקבלת מדידת לחץ ראשונית ולאחר פרק זמן מסוים, מודדת שוב ומחשבת את ההפרש בגובה.

קלט : pressure_baseline (Po) בתור רפרנס.

פלט : altitude_delta

- **void Drowning_detected()**

פונקציה זו נקראת על ידי In_water() כאשר זוהה מקרה טביעה. הפונקציה מתזמנת שתי פונקציות שונות למניעת טביעה. הפונקציה הראשונה היא של פתיחת השסתום לשחרור ה - Co2. פונקציה שנייה לשליחת הודעת חירום למרכז בקרה.

קלט : אין

פלט : אין

- **void Toggle_valve()**

פונקציה זו נקראת על ידי Drowning_detected() כאשר זוהה מקרה טביעה. אם הדגל שגוי (השסתום סגור) הפעל את בקר המנוע והפעל את המנוע לפתיחת השסתום ולהיפך.

קלט : אין

פלט : אין

- **void Sending_sms(String message)**

פונקציה זו נקראת על ידי Drowning_detected() כאשר זוהה מקרה טביעה או על ידי Check_device() כאשר זוהתה לחיצה על כפתור. היא מעירה את ה - SIM800L ושולחת הודעת SMS (לפי המחרוזת שהתקבלה ולפי המצב במצב בו אנו נמצאים) למרכז בקרה. לאחר שליחת הודעות, חוזר לישון.

קלט : message מחרוזת

פלט : אין

- **void Check_device()**

פונקציה זו נקראת על ידי loop() כאשר זוהתה מקרה של לחיצה על כפתור בדיקת הרכיבים. בודק את כל רכיבי המערכת ואז שולח SMS מתאים למרכז בקרה. אם כל הרכיבים עובדים ותקינים, נורת לד ירוקה נדלקת אחרת נורד לד אדומה נדלקת.

קלט: אין

פלט: אין

- **String readSerial()**

הפונקציה מחזירה את הפלט של רכיב ה - GSM במטרה לעקוב אחרי תוצאות הפקודות שנשלחו לרכיב.

קלט: אין

פלט: הפלט של רכיב ה - GSM

3.4. אתגרים במימוש

במהלך העבודה על הפרויקט נתקלנו במספר אתגרים ובעיות שדרשו התייחסות במיוחדת על מנת להגיע למצב בו המערכת עובדת באופן מלא ומדויק. להלן האתגרים איתם התמודדנו ודרכי הפתרון:

(1) זיהוי מגע עם מים בצורה מדויקת ואמינה

על מנת שזיהוי מגע עם מים יהיה יעיל ומהיר נדרש למצוא פתרון ולשלב אותו במערכת. לאחר בדיקות וניסויים של כלל האופציות הוחלט להשתמש בשני פרובים לזיהוי מגע עם מים. כאשר המעגל נסגר (בעזרת מוליכות המים), זה ייחשב כמגע עם מים.

(2) קבלה ועיבוד הנתונים מחיישן הלחץ / עומק

כאשר חיישן הלחץ מעביר נתונים למיקרו בקר הוא עושה את זה בעזרת תקשורת טורית I^2C . בהתחלה נדרש ללמוד איך ליצור את הקשר הזה בין החיישן למיקרו בקר. לאחר שהחיישן העביר נתונים, נדרש לנתח את המידע ולדעת להשתמש בו לטובת חישוב שינוי הגובה. לאחר קריאה מרובה וחיפוש באינטרנט נמצאה נוסחה שמאפשרת למצוא את השינוי בגובה בעזרת ההפרש בין שני מדידות לחץ. לחץ ראשוני (לחץ רפרנס) ולחץ נוסף.

$$\text{altitude} = 44330 * \left(1 - \left(\frac{p}{p_0} \right)^{\frac{1}{5.255}} \right)$$

משוואה 5 - חישוב שינוי בגובה

(3) הפעלת רכיב ה- SIM800L בצורה רציפה וללא ניתוקים מהרשת

רכיב ה- SIM800L הוא רכיב לתקשורת סלולרית המשתמש בתקנים GSM ו- GPRS (דור 2 ו-2.5 בהתאמה). אחת הבעיות עם הרכיב הזה היא שכל הזמן הרכיב מאבד את הרשת הסלולרית. בנוסף לקח הרבה זמן עד שהוא מתחבר לרשת. לאחר בדיקות רבות של קודים שונים לשימוש עם הרכיב, החלפה של האנטנה לאנטנה יותר גדולה ואיכותית לא נמצאה הסיבה לבעיה. העבודה נעשתה המפרט הטכני של הרכיב וסופק לו המתח והזרם הדרושים בשבילו.

בסופו של דבר, לאחר בדיקה מקיפה, התגלה כי לרכיב פיק צריכת זרם של עד 2A בעת חיבור לרשת הסלולרית. הסוללה שחוברה למערכת התאימה למשימה, הבעיה שהחוטים שמחוברים לסוללה דקים מידי, בעלי התנגדות גדולה. ולכן במטרה לפתור את הבעיה החוטים הוחלפו ובסוף חוטי המתח בין הסוללה לרכיב קוצרו. כתוצאה מהשינויים הללו, הרכיב מתחבר לרשת בלי בעיה ועובד בצורה רציפה וללא ניתוקים מהרשת.

(4) הפעלת רכיב ה- SIM800L בצורה חסכונית

רכיב ה- SIM800L הוא רכיב לתקשורת סלולרית קטן וקומפקטי לצרכי הפרויקט. הבעיה שהוא הרכיב הכי זולל אנרגיה שקיים במערכת. במצב רגיל, כאשר מחובר לרשת הוא צורך כ- 13mA. מכיוון שאחת הדרישות בפרויקט שהרכיבים יהיו קטנים כל הניתן במטרה שיכנסו למארז קטן, הסוללה תהיה קטנה, כלומר תהיה בעלת קיבול נמוך יחסית. ולכן נדרש דרך להוריד את צריכת האנרגיה של הרכיב.

לאחר בדיקות וניסויים של קודים שונים הצלחנו הוכנס את הרכיב למצב שינה, מה שהוריד את צריכת האנרגיה שלו ל- 1mA.

4. תוצאות והדגמות

בתחילה מימוש הפרויקט נבדק כל רכיב בנפרד ונכתב לו קוד מתאים שיבצע את הדרוש ממנו לפי הדרישות שהוגדרו. לאחר שכל הרכיבים עבדו בנפרד כמצופה מהם, כל הקודים השונים חוברו לקוד אחד מרוכז שירוצ על המיקרו בקר.

כל רכיבי המערכת החשמלית חוברו על מטריצת בדיקות במטרה לבדוק את המערכת כולה. הקוד שנכתב נצרב על המיקרו בקר והתבצעה בדיקה של המערכת בכלל המצבים האפשריים.

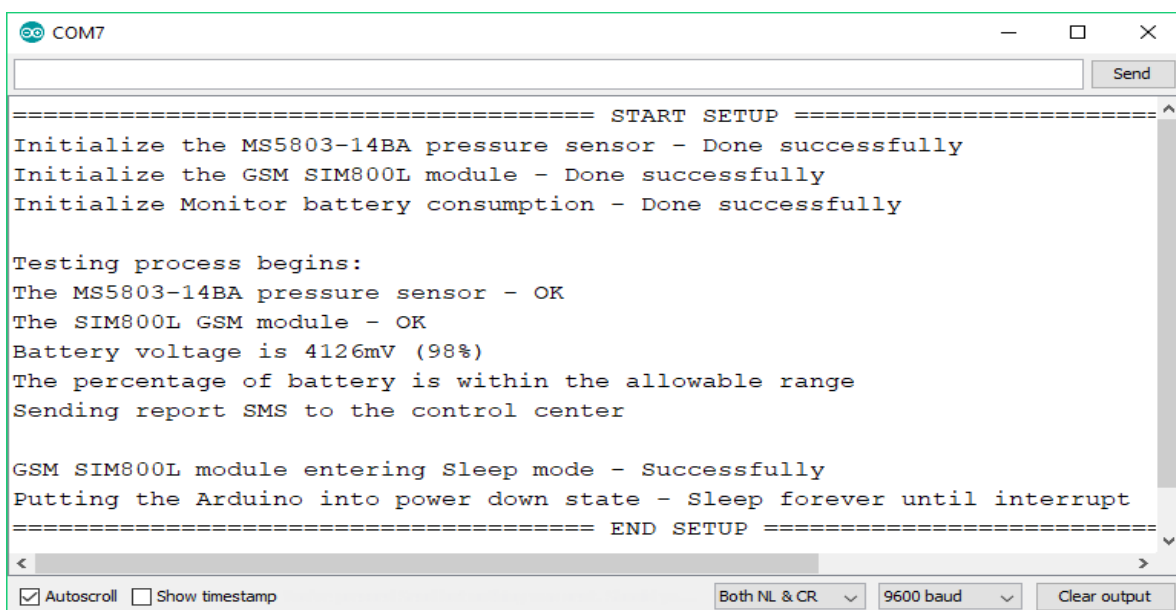
לאחר הרצות ובדיקות של המערכת בשלמותה התקבלו תוצאות שתואמות לדרישות שהגדרנו.

4.1. המערכת מבצעת בדיקת תקינות כללית לאחר הדלקתה

אחת הדרישות שהוגדרו היא לבצע בדיקת מערכת אוטומטית כללית לווידוי תקינות המערכת לאחר הלבשת השרשרת על הילד.

כאשר מדליקים את המערכת היא מבצעת סבב בדיקות לרכיבי המערכת. אם הכל תקין היא שולחת הודעה למרכז בקרה עם פרטי הבדיקות ומדליקה לד ירוקה המציין כי המערכת תקינה. אם יש בעיה היא מדווחת למרכז בקרה ומדליקה לד בצבע אדום המציין כי קיימת בעיה במערכת.

כפי שניתן לראות בתמונות הבאות המערכת מגיבה כמצופה ולפי הדרישה שהגדרנו.



```
===== START SETUP =====
Initialize the MS5803-14BA pressure sensor - Done successfully
Initialize the GSM SIM800L module - Done successfully
Initialize Monitor battery consumption - Done successfully

Testing process begins:
The MS5803-14BA pressure sensor - OK
The SIM800L GSM module - OK
Battery voltage is 4126mV (98%)
The percentage of battery is within the allowable range
Sending report SMS to the control center

GSM SIM800L module entering Sleep mode - Successfully
Putting the Arduino into power down state - Sleep forever until interrupt
===== END SETUP =====
```

תמונה 19 - המערכת מבצעת בדיקת תקינות כללית לאחר הדלקתה
המערכת תקינה

יום רביעי, 7 באוגוסט 2019

21:44 The device is working properly, Battery Level: 98%



תמונה 20 - קבלת הודעת SMS למרכז בקרה שהכל תקין

```
COM7
===== START SETUP =====
Initialize the MS5803-14BA pressure sensor - Done successfully
Initialize the GSM SIM800L module - Done successfully
Initialize Monitor battery consumption - Done successfully

Testing process begins:
The MS5803-14BA pressure sensor - NOT OK
The SIM800L GSM module - OK
Battery voltage is 4126mV (98%)
The percentage of battery is within the allowable range
Sending report SMS to the control center

GSM SIM800L module entering Sleep mode - Successfully
Putting the Arduino into power down state - Sleep forever until interrupt
===== END SETUP =====
```

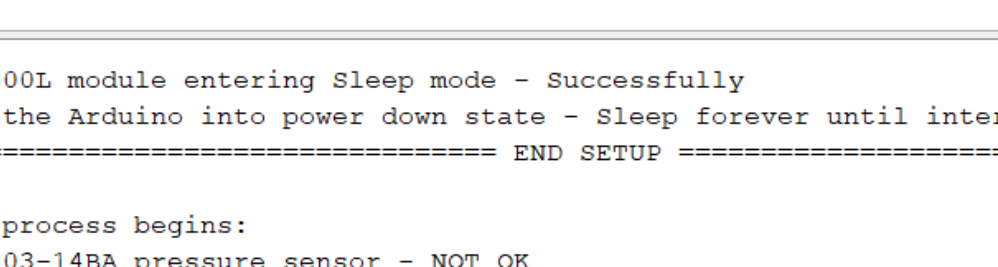
תמונה 21 - המערכת מבצעת בדיקת תקינות כללית לאחר הדלקתה
המערכת לא תקינה כי החיישן לחץ מנותק

4.2. הפעלת פסיקה לבדיקת תקינות המערכת לאחר לחיצה על הכפתור

אחת הדרישות שהוגדרו היא האפשרות לבצע בדיקת מערכת כללית יזומה לווידוי תקינות המערכת. לאחר הדלקת המערכת ובדיקת תקינות אוטומטית היא נכנסת למצב שינה. אם קיים רצון לבצע בדיקת תקינות באופן יזום יש צורך ללחוץ על כפתור בדיקת התקינות. לחיצה על הכפתור מפעילה פסיקה שמעירה את המערכת.

לאחר לחיצה על הכפתור המערכת מבצעת סבב בדיקות לרכיבי המערכת. אם הכל תקין היא שולחת הודעה למרכז בקרה עם פרטי הבדיקות ומדליקה לד ירוקה המציין כי המערכת תקינה. אם יש בעיה היא מדווחת למרכז בקרה ומדליקה לד בצבע אדום המציין כי קיימת בעיה במערכת.

כפי שניתן לראות בתמונות הבאות המערכת מגיבה כמצופה ולפי הדרישה שהגדרנו.



The screenshot shows the Arduino IDE serial monitor window. The title bar indicates the connection is to COM7. The text in the monitor is as follows:

```
GSM SIM800L module entering Sleep mode - Successfully
Putting the Arduino into power down state - Sleep forever until interrupt
===== END SETUP =====

Testing process begins:
The MS5803-14BA pressure sensor - NOT OK
The SIM800L GSM module - OK
Battery voltage is 4126mV (98%)
The percentage of battery is within the allowable range
Sending report SMS to the control center

Putting the Arduino into power down state - Sleep forever until interrupt
```

At the bottom of the window, there are controls for the serial monitor: ☒ Autoscroll, ☐ Show timestamp, a dropdown menu set to 'Both NL & CR', a dropdown menu set to '9600 baud', and a 'Clear output' button.

תמונה 22 - בדיקת תקינות המערכת לאחר לחיצה על הכפתור המערכת לא תקינה כי החיישן עומק מנותק

22:06 The device NOT working properly,
Battery Level: 98%

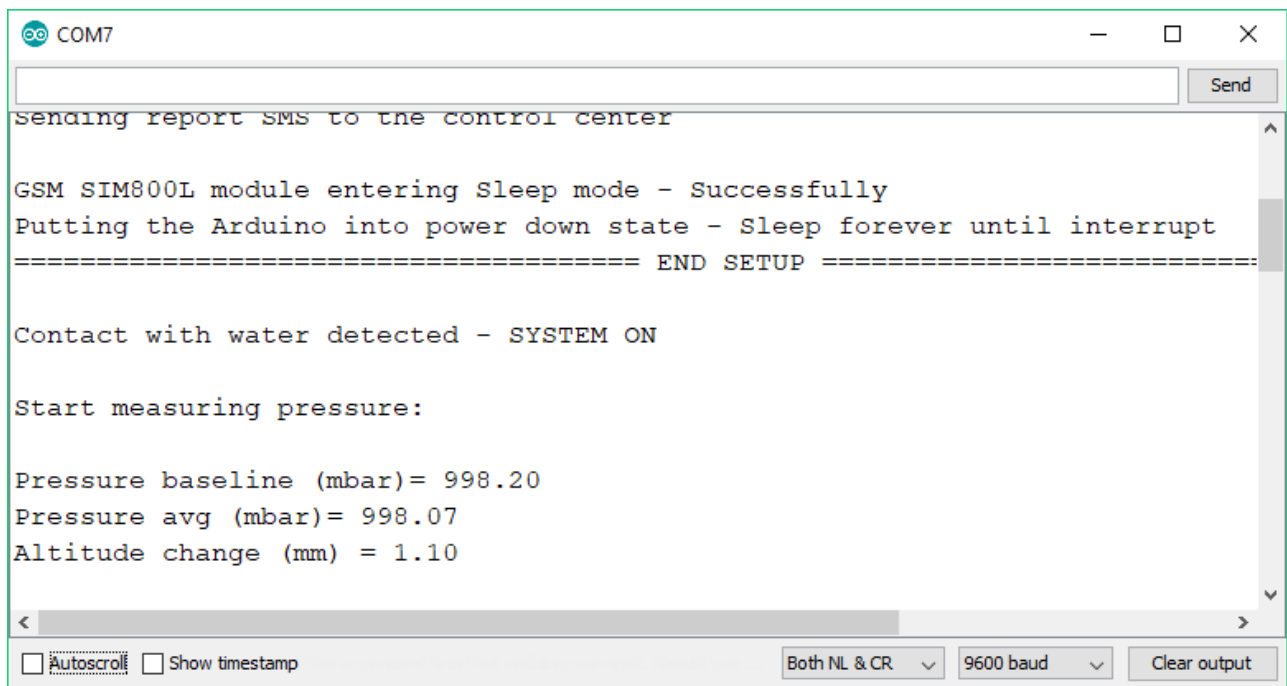
תמונה 23 - הודעת SMS לאחר לחיצה על הכפתור המערכת לא תקינה כי החיישן עומק מנותק

4.3. זיהוי מגע עם מים

אחת הדרישות שהוגדרו היא לזהות מגע עם מים במטרה להעיר את המערכת משינה ולהתחיל למדוד את העומק בו נמצא התינוק / ילד. באופן זה המערכת נמצאת במצב שינה וצורכת פחות אנרגיה.

כאשר הפרובים מוכנסים למקור מים, המערכת מתעוררת, וכאשר הם מוצאים ממקור המים המערכת חוזרת למצב שינה.

כפי שניתן לראות בתמונה הבאה המערכת מגיבה כמצופה ולפי הדרישה שהגדרנו.



```
sending report SMS to the control center

GSM SIM800L module entering Sleep mode - Successfully
Putting the Arduino into power down state - Sleep forever until interrupt
===== END SETUP =====

Contact with water detected - SYSTEM ON

Start measuring pressure:

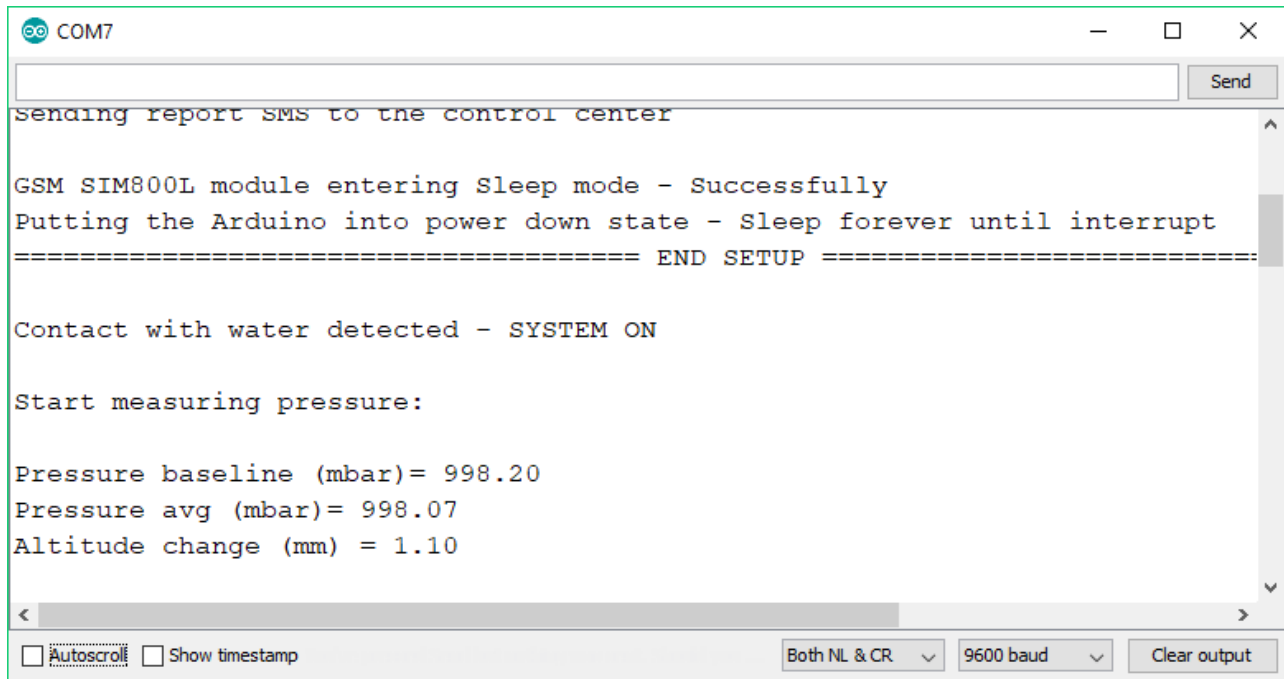
Pressure baseline (mbar)= 998.20
Pressure avg (mbar)= 998.07
Altitude change (mm) = 1.10
```

תמונה 24 - זיהוי מגע עם מים

4.4. קבלה ועיבוד נתונים מחיישן העומק

אחת הדרישות שהוגדרו היא לזהות את העומק בו נמצא התינוק / ילד כאשר הוא במקור מים. לשם כך המערכת משתמשת בחיישן לחץ למדידת הלחץ בכל רגע נתון. לאחר מדידה היא מחשבת את העומק בו התינוק / ילד נמצא באמצעות משוואה מספר 3.

כפי שניתן לראות בתמונות הבאות המערכת מגיבה כמצופה ולפי הדרישה שהגדרנו.



```
sending report SMS to the control center

GSM SIM800L module entering Sleep mode - Successfully
Putting the Arduino into power down state - Sleep forever until interrupt
===== END SETUP =====

Contact with water detected - SYSTEM ON

Start measuring pressure:

Pressure baseline (mbar)= 998.20
Pressure avg (mbar)= 998.07
Altitude change (mm) = 1.10
```

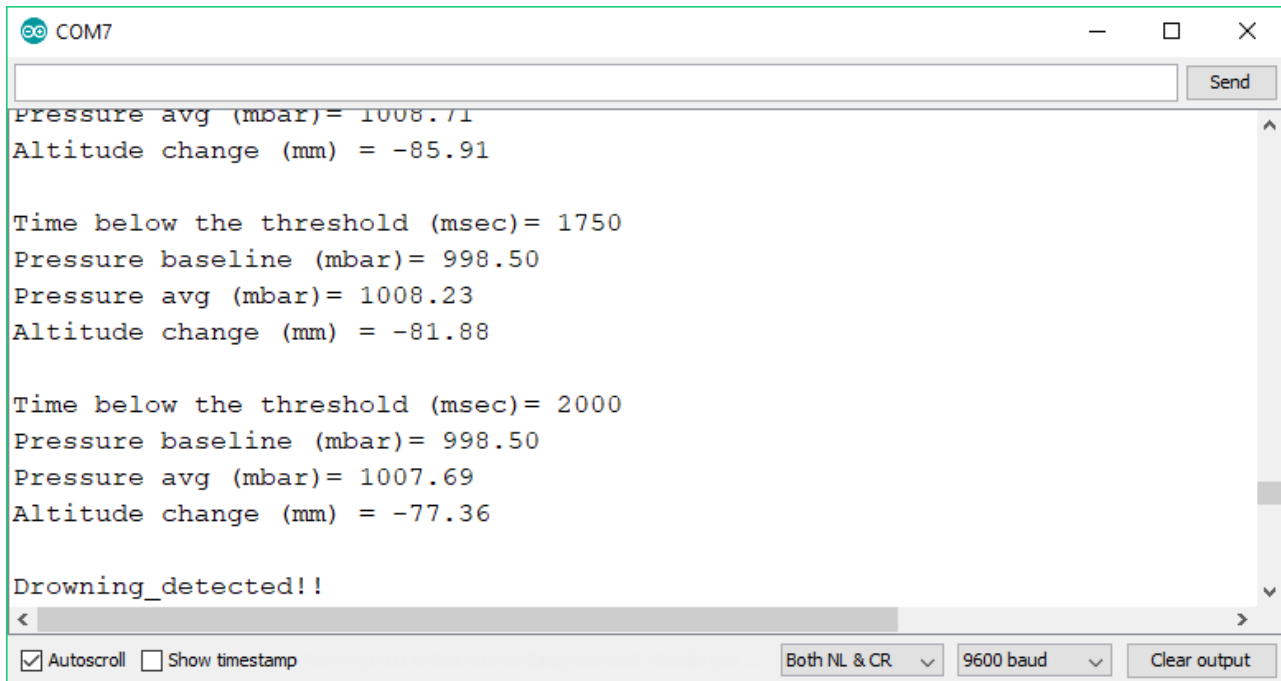
תמונה 25 - קבלה ועיבוד נתונים מחיישן העומק

4.5. זיהוי מצב טביעה

אחת הדרישות שהוגדרו היא זיהוי עומק סף: כתיבת אלגוריתם לזיהוי עומק שיוגדר מראש ובדיקה אם הערך שהתקבל אחרי ניתוח של תוצאות החיישנים על ידי הארדווינו חוצה את הסף.

בנוסף הפעלת טיימר לזמן מוגדר מראש שיבדוק אם הילד לא יוצא מהעומק סף במסגרת הזמן הזה.

כפי שניתן לראות בתמונות הבאות המערכת מגיבה כמצופה ולפי הדרישה שהגדרנו.



```
COM7
Send
Pressure avg (mbar)= 1008.71
Altitude change (mm) = -85.91

Time below the threshold (msec)= 1750
Pressure baseline (mbar)= 998.50
Pressure avg (mbar)= 1008.23
Altitude change (mm) = -81.88

Time below the threshold (msec)= 2000
Pressure baseline (mbar)= 998.50
Pressure avg (mbar)= 1007.69
Altitude change (mm) = -77.36

Drowning_detected!!
```

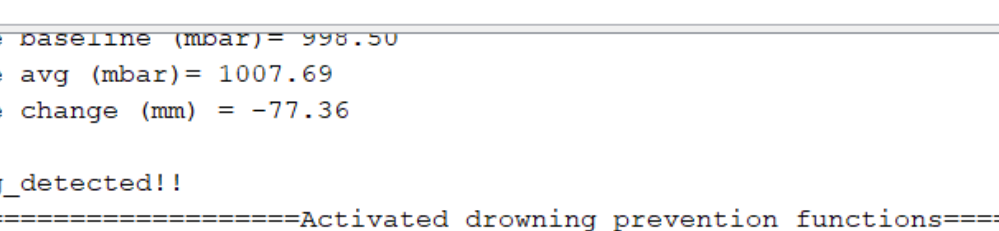
Autoscroll ☒ Show timestamp Both NL & CR 9600 baud Clear output

תמונה 26 - זיהוי מצב טביעה לאחר שנגמר הטיימר

4.6. ניפוח בלוני ציפה כמנגנון ראשון להצלת התינוק / ילד

אחת הדרישות שהוגדרו היא שכאשר זוהתה טביעה, יופעל המנוע לסיבוב הווסת במטרה לשחרר את הגז מהמחסנית ובכך למלא את בלוני הציפה.

כפי שניתן לראות בתמונות הבאות המערכת מגיבה כמצופה ולפי הדרישה שהגדרנו.



The screenshot shows a serial terminal window with the title 'COM7'. The window contains the following text:

```
Pressure baseline (mbar)= 998.50
Pressure avg (mbar)= 1007.69
Altitude change (mm) = -77.36

Drowning_detected!!

=====Activated drowning prevention functions=====

Turn valve ON

Sending emergency SMS to the control center number

Turn valve OFF

=====Finish drowning prevention functions=====
```

At the bottom of the window, there are three checkboxes: 'Autoscroll' (checked), 'Show timestamp' (unchecked), and 'Both NL & CR' (selected). To the right of these checkboxes are two dropdown menus: '9600 baud' and 'Clear output'.

תמונה 27 - ניפוח בלוני ציפה כמנגנון ראשון להצלת התינוק / ילד

5. מסקנות והמלצות

כפי שצוין בפרק ההקדמה מטרת הפרויקט היא לפתח מערכת המשלבת חומרה ותוכנה שתדע לזהות מקרה טביעה בקרב ילדים ותיתן מענה מהיר ונכון לצורך הצלת חיי הילד בעת טביעה.

המערכת החשמלית שתוכננה ונבנתה במהלך הפרויקט, עונה על הדרישות שהוגדרו. במקרה של תקציב יותר גדול ניתן היה להגיע למוצר מוגמר כמו שתוכנן. המוצר הסופי, כמו שניתן לראות בסעיף 3.2.1 (מבנה חיצוני) המוצר הסופי יהיה בנויה בצורת שרשרת אשר יולבש על צוואר הילד. משני צידי השרשרת יהיו בלוני ציפה מקופלים לתוך השרשרת עצמה. לשרשרת יחובר מארז עמיד למים ובתוכה תהיה מערכת חשמלית.

5.1. תוצאות והישגים

- (1) המערכת החשמלית שתוכננה ונבנתה במהלך הפרויקט, עונה על הדרישות שהוגדרו.
- (2) הקוד שנכתב עומד בדרישות שהוגדרו.
- (3) הקוד של המערכת מתועד היטב, על מנת לאפשר עדכונים ושיפורים עתידיים.
- (4) על אף כל האתגרים שתוארו בפרק 3 המערכת עומדת בכל הדרישות שעלו בשלב התחלתי של הפרויקט.

5.2. המלצות להמשך

- בניית אב-טיפוס של המוצר הסופי ובדיקתו בבריכה ובים.
- תכנון מעגל מודפס שיכיל את כל רכיבי המערכת החשמלית ובכך ליצור מוצר קטן יותר.
- הדפסה בתלת מימד את המארז למערכת החשמלית, לקבלת מראה אלגנטי יותר ולקבלת מערכת נוחה יותר לשימוש על צוואר הילד.
- הכנסת רכיב נוסף למערכת - רמקול קטן שיצפץ במקרה של טביעה, לקבל תשומת הלב של הסובבים.

6. ביבליוגרפיה

1. ספר "מדריך ארדואינו למתחילים חלק א"
URL: <https://hackstore.co.il/books/arduino-for-beginners-1/>
2. ספר "מדריך ארדואינו למתחילים חלק ב"
URL: <https://hackstore.co.il/books/arduino-for-beginners-2/>
3. Arduino Pro Mini 328 - 3.3V/8MHz
URL: <https://www.sparkfun.com/products/11114>
4. ATmega328P datasheet
URL: <https://www.sparkfun.com/datasheets/Components/SMD/ATMega328.pdf>
5. Pressure Sensor Breakout - MS5803-14BA
URL: <https://www.sparkfun.com/products/12909>
6. MS5803-14BA datasheet
URL: https://cdn.sparkfun.com/datasheets/Sensors/Weather/ms5803_14ba.pdf
7. LiPo Charger/Booster
URL: <https://www.sparkfun.com/products/14411>
8. GPRS GSM Module (SIM800L)
URL: https://filipeflop.com/files/download/Datasheet_SIM800L.pdf
9. Micro Metal Gearmotors
URL: <https://www.pololu.com/pololu-micro-metal-garmotors>
10. Motor Driver - TB6612FNG
URL: <http://www.sparkfun.com/datasheets/Robotics/TB6612FNG>
11. Lithium Ion Polymer battery
URL: <https://learn.adafruit.com/li-ion-and-lipoly-batteries>
12. 12-gram CO2 cartridge
URL: https://www.spudfiles.com/wiki/12_gram_CO2_cartridge
13. משרד הבריאות - בטיחות ילדים במים ומניעת טביעה
URL: <https://www.health.gov.il/KidsAndMatures/safety/DrownPrevent>
14. היפגעות ילדים בישראל: דו"ח 'בטרם' לאומה 2017
URL: http://www.beterem.org/Beterem_ChildInjury2017.pdf
15. מידע על סוגי בדים לבלויני ציפה
URL: <https://www.diypackraft.com/packraft-details/fabrics>
16. רמת אטימות IP - מה זה רמת IP לציד חשמלי, באתר המוסד לבטיחות ולגיהות
URL: http://oldsite.osh.org.il/d_1527_ip.pdf
17. Human Body Density and Fat
URL: <https://doi.org/10.1093/ajcn/20.4.305>
The American Journal of Clinical Nutrition, Volume 20, Issue 4, April 1967, Pages 305–310
Published: 01 April 1967
18. מדריך למורה - לביצוע עבודות גמר : פרויקט במגמת הנדסת אלקטרוניקה ומחשבים
URL: <https://school.kotar.cet.ac.il/KotarApp/BookID=98064720#51>
מקום ההוצאה : תל אביב, הוצאה : מטח : המרכז לטכנולוגיה חינוכית. שנת ההוצאה : תשע"ב - 2012
19. יסודות הפיזיקה א : מכניקה, אלקטרוסטיקה, זרם ישר - יחידות 1-2
URL: <https://kotar.cet.ac.il/nBookID=93001430>
מקום ההוצאה : רעננה, הוצאה : האוניברסיטה הפתוחה. שנת ההוצאה : תשנ"ט - 1999

Injuries and deaths from drowning, especially of infants and children, have become a sad routine in Israel and around the world. Drowning is the second cause of mortality in children. Imprints are usually caused by the absence of supervision, even by a few seconds. It can happen at sea or at the pool.

Because of this important problem, it was decided to develop a system that integrates hardware and software that will recognize a drowning case in children and provide a quick and correct response to save the child's life while drowning. The system is for children aged 0-8 who cannot swim and should not enter the water without adult supervision.

The system is responsible for identifying the drowning and operating the various mechanisms to save the child if necessary. The electrical system includes a microcontroller, motor, power source, GSM component and various sensors that will monitor the depth of the child in the water. When the child is at a certain depth for a predetermined amount of time, the electrical system will operate different mechanisms to save the child.

A first mechanism and the main requirement of the system is the ability to raise the child's head above the water and keep it in this position. Responding to this requirement avoids contact with the airways leading to suffocation. A second mechanism is to send an emergency message (SMS) to a control center, where the event will operate, sending an SMS message back to the parents. This mechanism meets the requirement for parental attention. There is also a button to check the system's integrity. In response to a click, you will send an SMS with a detailed health report.

At the implementation stage, all electrical components were connected to one system and a code was written to Arduino that fulfills the defined requirements. The system detects contact with water using 2 probes when they are inserted into a water source, after which the system turns on and starts to process data from the pressure / depth sensor. With the sensor you can measure a pressure of exactly 0.01% and calculate the depth difference where the sensor is located. The system performs pressure measurements, and alerts about any drowning condition as soon as the depth exceeds 10 cm. After crossing the threshold, it activates a 2-second timer.

The voltage required to run the motor is 6V and therefore a voltage converter is used, which increases the voltage of the LiPo battery to the required voltage. After inflating the buoyancy balloons, the system sends an emergency message to a control center. The SMS is sent via the cellular communication component. In addition to the initiated integrity check by pressing a button, the system performs an automatic integrity check for the system components when it is turned on. After an initiated or automatic integrity check, the data is sent to a control center and the LED is lit in the color corresponding to the test result.

After many tests and experiments it can be said that the system fulfills the requirements defined in the design phase. The system is capable of recognizing drowning at a very high accuracy percentage. The total system consumption is about 4.5 mA in sleep mode, and about 15mA when the system is in contact with water. Because the final product has a battery size limit, it was decided that the system must be turned off until it is mounted on the child's neck. It will automatically turn on as soon as the chain bracket is locked. In the case of a larger budget, a finished product could be reached that could be dressed for the child.

8.1. נספח 1: קוד המערכת

הקוד הבא הוא קוד בדיקה (Debugging) המכיל הרבה Serial.print במטרה לבדוק את הקוד ובנוסף להצגה בשלב ההגנה.

```
// Version 0.9 - Debugging

// Include Libraries :
//-----//
#include <SoftwareSerial.h> // Allow serial communication on other
digital pins of the Arduino(built-in librarie) - for SIM800L
#include <Wire.h> // Communicate with I2C (built-in
librarie) - for MS5803-14BA pressure sensor

#include <LowPower.h> // Power management library
(https://github.com/rockscream/Low-Power)
#include <SparkFun_MS5803_I2C.h> // Allows easily talk to the MS5803-14BA
pressure sensor (http://librarymanager/All#SparkFun MS5803-14BA)
#include <Battery.h> // Allows monitor battery consumption
(https://github.com/rlogiacco/BatterySense)

SoftwareSerial SIM800L_Serial(11, 10); // TX and RX pins (respectively) to
communicate with SIM800L module
MS5803 pressure_sensor(ADDRESS_HIGH); // Available addresses (selected by
jumper on board) - ADDRESS_HIGH = 0x76 (default) OR ADDRESS_LOW = 0x77
Battery batt = Battery(3000, 4200, A0); // minVoltage, maxVoltage and
sensePin
//-----//

// Define Pins :
//-----//
#define InWater_Prob 2 // The pin number for the Prob (Only pin 2,3
USABLE FOR INTERRUPTS)
#define Button 3 // The pin number for the Push button (Only pin
2,3 USABLE FOR INTERRUPTS)
#define RedledPin 8 // The pin number for the Red LED
#define GreenledPin 9 // The pin number for the Green LED

// Pinout for TB6612FNG
#define PWMA 6 // PWM input that controls the speed
#define AIN1 5 // One of the two inputs that determines the direction.
#define STBY 7 // Allows the H-bridges to work when high
#define AIN2 4 // One of the two inputs that determines the direction.
//-----//

// Global Variables :
//-----//
volatile boolean InWater = false; // Flag for symbolizes the contact
with water (volatile because associated with interrupt)
volatile boolean Check_dev = false; // Flag that indicates test status
or not (volatile because associated with interrupt)
boolean Arduino_state = false; // Flag for the Arduino status
(stand-by = true, sleep = false)
boolean Valve_state = false; // Flag for the valve status (open
= true, close = false)
boolean Gsm_sleepMode = false; // Flag for the SIM800L status
(stand-by = false, sleep = true)
//-----//
```

```

// Function Declaration :
//-----//
void Water_activated_switch();
void Button_activated_switch();
void In_water();
double Check_depth(double pressure_baseline);
void Drowning_detected();
void Toggle_valve();
void Sending_sms(String message);
void Check_device();
//-----//

/* --- setup() -----
   This function is called when a sketch starts. We use it to initialize
   variables, pin modes, start using libraries, etc.
   The function will only run once, after each powerup or reset of the
   Arduino board.
   Input: None
   Output: None
*/
void setup() {

    pinMode(InWater_Prob, INPUT);    // Initialize the Prob pin as an input
                                     // for water contact
    pinMode(Button, INPUT_PULLUP);   // Initialize the button pin as an
                                     // input with pull up resistor

    // PinMode for common Anode RG
    pinMode(RedledPin, OUTPUT);      // initialize the Red LED pin as an
                                     // output
    pinMode(GreenledPin, OUTPUT);    // initialize the Green LED pin as an
                                     // output

    digitalWrite(RedledPin, 255);    // set initial LED state
    digitalWrite(GreenledPin, 255);  // set initial LED state

    // PinMode for TB6612FNG
    pinMode(STBY, OUTPUT);           // PWM input that controls the speed
    pinMode(AIN1, OUTPUT);           // One of the two inputs that
                                     // determines the direction
    pinMode(AIN2, OUTPUT);           // Allows the H-bridges to work when
                                     // high
    pinMode(PWMA, OUTPUT);           // One of the two inputs that
                                     // determines the direction

    analogReference(INTERNAL);        // Configures the reference voltage
                                     // used for analog input volts

    // initialize serial communication at 9600 bits per second:
    Serial.begin(9600);

    Serial.println(F("===== START SETUP ====="));

    // Start I2C object for the MS5803-14BA pressure sensor
    Wire.begin();

    Serial.print("Initialize the MS5803-14BA pressure sensor - ");
    pressure_sensor.reset();         // Reset device I2C
    if (!pressure_sensor.begin()) {   // Initialize library for subsequent
                                     // pressure measurements
        Serial.println("Done successfully");
    }
    else {

```

```

    Serial.println("Unsuccessful");
}

Serial.print("Initialize the GSM SIM800L module - ");
SIM800L_Serial.begin(9600);    // Initialize Serial Monitor for the
                                SIM800L

delay(50);

SIM800L_Serial.println("AT");
if ((readSerial().indexOf("OK")) != -1 ) {
    Serial.println("Done successfully");
}
else {
    Serial.println("Unsuccessful");
}

Serial.println("Initialize Monitor battery consumption - Done
successfully\n");
batt.begin(1100, 5.9651, &sigmoidal);    // Initialize monitor battery
                                           consumption

Check_dev = true;                        // Indicates that test mode enable.
Check_device();
Check_dev = false;                      // After finish the Check_device, the flag
                                           returns to its initial value

Serial.print("\nGSM SIM800L module entering Sleep mode - ");
SIM800L_Serial.println("AT+CSCLK=2");    // Entering Sleep mode
if ((readSerial().indexOf("OK")) != -1 ) {
    Serial.println("Successfully");
    Gsm_sleepMode = true;                // Indicates that the SIM800L
                                           is in sleep mode
}
else {
    Serial.println("Unsuccessful");
}

attachInterrupt(digitalPinToInterrupt(2), Water_activated_switch,
RISING);    // When pin 2 rising change, exit sleep mode and execute
            Water_activated_switch function
attachInterrupt(digitalPinToInterrupt(3), Button_activated_switch,
FALLING);    // At the push of a button, exit sleep mode and execute
            Button_activated_switch function

Serial.println(F("Putting the Arduino into power down state - Sleep
forever until interrupt"));
Serial.println(F("=====END SETUP =====\n"));
delay(10000);
LowPower.powerDown(SLEEP_FOREVER, ADC_OFF, BOD_OFF); // Putting
microcontroller into power down state - Sleep forever until interrupt
}
//-----//

/* --- loop() -----
The function does precisely what its name suggests, and loops
consecutively.
Enable / disable interrupt, enable sleep mode, and enable drowning
detection mechanisms or Check device mechanisms.
Input: None
Output: None
*/
void loop() {

    if (InWater) {

```

```

    detachInterrupt(digitalPinToInterrupt(2)); // Remove interrupt to stop
it from keep firing
    detachInterrupt(digitalPinToInterrupt(3)); // Remove interrupt to stop
it from keep firing

    In_water(); // Function is activated after exiting
sleep mode when water contact is detected
    InWater = false; // After leaving the water, the flag
returns to its initial value
    Arduino_state = true; // Indicates that the Arduino is in
stand-by mode
    attachInterrupt(digitalPinToInterrupt(2), Water_activated_switch,
RISING); // When pin 2 rising change, exit sleep mode and execute
Water_activated_switch function
    delay(100);
}

    if (Check_dev) {
        detachInterrupt(digitalPinToInterrupt(3)); // Remove interrupt to
stop it from keep firing

        Check_device(); // Function is activated after exiting
sleep mode when push of a button is detected
        Check_dev = false; // After finish the Check_device, the
flag returns to its initial value
        Arduino_state = true; // Indicates that the Arduino is in
stand-by mode
        attachInterrupt(digitalPinToInterrupt(3), Button_activated_switch,
FALLING); // At the push of a button, exit sleep mode and execute
Button_activated_switch function
        delay(100);
    }

    if (Arduino_state) {
        Serial.println(F("\nPutting the Arduino into power down state - Sleep
forever until interrupt"));
        Arduino_state = false; // After finish the Check_device or In_water,
the flag returns to its initial value (sleep)
        delay (100);
        LowPower.powerDown(SLEEP_FOREVER, ADC_OFF, BOD_OFF); // Putting
microcontroller into power down state - Sleep forever until interrupt
    }
}
//-----//

/* --- Water_activated_switch -----
This function is called when pin 2 rising change (interrupt function).
Switches the flag to water contact mode.
Input: None
Output: None
*/
void Water_activated_switch() {
    InWater = true; // Indicates that water
contact detected.
    Check_dev = false; // After finish the
Check_device, the flag returns to its initial value
}
//-----//

/* --- Button_activated_switch() -----
This function is called when pin 3 falling change (push of a button)
(interrupt function).
Switches the flag to Check device mode.
*/

```

```

void Button_activated_switch() {
    Check_dev = true; // Indicates that push of a
    button detected.
}
//-----//

/* --- In_water() -----
    This function is called by loop() after the interrupt function ended.
    Initial pressure gauge, after a certain amount of time, measures again
    and then sends the data
    to the Check_depth() function to calculate the altitude difference.
    If the height difference is below the threshold (for a predefined time)
    then the Drowning_detected() function is activated.
    Otherwise, the function ends and returns to the loop() function.
    Input: None
    Output: None
*/
void In_water() {

    const float depth_threshold = -70.0; // The threshold depth from which a
    drowning is defined (Negative as we go down deep)

    // Create variables to store results from the pressure sensor
    double altitude_delta, pressure_baseline;

    int timer_count = 0; // Used to measure time below the threshold

    Serial.println("Contact with water detected - SYSTEM ON\n");
    Serial.println("Start measuring pressure:\n");

    pressure_baseline = pressure_sensor.getPressure(ADC_4096); // Read
    initial pressure measurement in mbar for reference (Po)

    while (digitalRead(2) == HIGH) { // As long as there is contact with
    water, calculate altitude delta
        altitude_delta = Check_depth(pressure_baseline);
        while (altitude_delta <= depth_threshold && digitalRead(2) ==
HIGH) { // As long as below the depth threshold, turn on timer
            if (timer_count >= 2000) { // 8 * 250 msec = 2 sec = 2000
msec
                Drowning_detected(); // Identified a drowning case
                break;
            }
            Serial.print("Time below the threshold (msec)= ");
            delay(250);
            timer_count = timer_count + 250;
            Serial.println(timer_count);
            altitude_delta = Check_depth(pressure_baseline);
        }
        timer_count = 0;
        delay(500);
    }
    Serial.println("Contact with water NOT detected - SYSTEM OFF");
}
//-----//

/* --- Check_depth() -----
    This function is called by In_water() when we need to calculate the
    altitude delta.
    Initial pressure gauge, after a certain amount of time, measures again
    and calculate the altitude difference.
    Input: pressure_baseline (Po) for reference.
    Output: altitude_delta.
*/
double Check_depth(double pressure_baseline) {

```

```

// Create variables to store results from the pressure sensor
double pressure_abs, pressure_avg = 0.0, altitude_delta = 0.0;

// Perform a number of pressure measurements
// and average calculation for more accurate pressure
for (int i = 0; i < 10; i++)
{
    // Read pressure from the sensor in mbar.
    pressure_abs = pressure_sensor.getPressure(ADC_4096);
    delay(100); // 100 msec delay
    pressure_avg = pressure_avg + pressure_abs;
}
pressure_avg = pressure_avg / 10.0;

Serial.print("Pressure baseline (mbar)= ");
Serial.println(pressure_baseline);
Serial.print("Pressure avg (mbar)= ");
Serial.println(pressure_avg);

/* Taking our baseline pressure at the beginning we can find an
approximate
change in altitude based on the differences in pressure.
Given a pressure measurement P (mbar) and the pressure at a baseline P0
(mbar),
return altitude (millimeters) above baseline.*/

altitude_delta = (44330.0 * (1 - pow(pressure_avg / pressure_baseline, 1
/ 5.255)));
Serial.print("Altitude change (mm) = ");
Serial.println(altitude_delta);
Serial.println("");
return altitude_delta;
}
//-----//

/* --- Drowning_detected() -----
This function is called by In_water() when we have identified a drowning
case.
The function scheduling two different drowning prevention functions.
The first function of opening the valve is to release the Co2.
Second function for sending an emergency message to a control center.
Input: None
Output: None
*/
void Drowning_detected() {

    Serial.println(F("Drowning_detected!!"));
    Serial.println(F("=====Activated drowning prevention
functions====="));
    Toggle_valve(); // Toggle the valve using
the DC motor
    Sending_sms(String("Drowning detected!!")); // Sending an emergency
message to a control center with the SIM800L
    Toggle_valve(); // Toggle the valve using
the DC motor
    Serial.println(F("=====Finish drowning prevention
functions=====\\n"));
    Serial.println(F("Continue to measuring pressure:\\n"));
}

```



```

}
//-----//

/* --- Toggle_valve() -----
This function is called by Drowning_detected() when we have identified a
drowning case.
If the flag is false (the valve is closed) start the engine controller
and turn the engine to open the valve and vice versa
Input: None
Output: None
*/
void Toggle_valve() {
    if (!Valve_state) { // If the valve close turn valve on
        Serial.println("\nTurn valve ON\n");

        // Turn valve on
        digitalWrite(STBY, HIGH);
        digitalWrite(AIN1, LOW);
        digitalWrite(AIN2, HIGH);
        analogWrite(PWMA, 255);
        delay(5000);

        Valve_state = true;    // Indicates that valve in open

        // Stop the motor
        digitalWrite(AIN1, LOW);
        digitalWrite(AIN2, LOW);
        analogWrite(PWMA, 0);
        delay(2000);
    }
    else {
        Serial.println("\nTurn valve OFF\n");

        // Turn valve off
        digitalWrite(AIN1, HIGH);
        digitalWrite(AIN2, LOW);
        analogWrite(PWMA, 255);
        delay(5000);

        Valve_state = false;    // Indicates that valve in close

        // Stop the motor
        digitalWrite(STBY, LOW);
        digitalWrite(AIN1, LOW);
        digitalWrite(AIN2, LOW);
        analogWrite(PWMA, 0);
    }
}
//-----//

/* --- Sending_sms() -----
This function is called by Drowning_detected() when we have identified a
drowning case
or by Check_device() when we have identified push of a button.
Wakes up the SIM800L and sends SMS (depending on the state we in) to a
control center.
After sending messages, go back to sleep.
Input: String message
Output: None
*/
void Sending_sms(String message) {

    const String number = "547280910";    // Control Center Emergency Number

```

```

    SIM800L_Serial.println("AT");           // A dummy message to wake the
component
    delay(150);
    Gsm_sleepMode = false;                 // Indicates that the SIM800L is in
stand-by mode

    if (Valve_state) {
        Serial.println("Sending emergency SMS to the control center number");

        for (int i = 0; i < 3; i++) {
            SIM800L_Serial.println("AT+CMGF=1");           // Set the
Mode as Text Mode
            delay(150);
            SIM800L_Serial.println("AT+CMGS=\"+972\" + number + \"\"); // To send
a message to a control center
            delay(150);
            SIM800L_Serial.print(message);                 // Content
of the SMS message
            delay(150);
            SIM800L_Serial.write((byte)0x1A);               // End of
message character 0x1A : Equivalent to Ctrl+z
            delay(50);
            SIM800L_Serial.println();
            delay(3000);
        }
    }
    else if (Check_dev) {
        Serial.println("Sending report SMS to the control center");

        SIM800L_Serial.println("AT+CMGF=1");           // Set the
Mode as Text Mode
        delay(150);
        SIM800L_Serial.println("AT+CMGS=\"+972\" + number + \"\"); // To send a
message to a control center
        delay(150);
        SIM800L_Serial.print(message);                 // Content
of the SMS message
        delay(150);
        SIM800L_Serial.write((byte)0x1A);           // End of message character 0x1A :
Equivalent to Ctrl+z
        delay(50);
        SIM800L_Serial.println();
        delay(3000);
    }

    if (Gsm_sleepMode) {
        Serial.print("GSM SIM800L module entering Sleep mode - ");
        SIM800L_Serial.println("AT+CSCLK=2");           // Entering Sleep mode
        if ((readSerial().indexOf("OK")) != -1 ) {
            Serial.println("Successfully");
            Gsm_sleepMode = true;                         // Indicates that the SIM800L
is in sleep mode
        }
        else {
            Serial.println("Unsuccessful");
        }
    }
}
//-----//

/* --- Check_device() -----
This function is called by loop() when we have identified a push button
case.
Checks all system components and then sends an appropriate SMS to a
control center.

```

```

    If all components are correct, a green LED turns on, otherwise red.
    Input: None
    Output: None
*/
void Check_device() {
    Serial.println("Testing process begins:");

    boolean fault = false;           // Flag that indicates if the device
    is working properly (true = Not good)
    boolean MS5803_state = false;     // Flag that indicates if the MS5803-
    14BA pressure sensor is working properly (true = OK, false = NOT ok)
    String message = "";

    int batt_volt = batt.voltage();
    int batt_percent = batt.level();
    float temperature_c = pressure_sensor.getTemperature(CELSIUS,
    ADC_256); // Temperature measurement to check the sensor pressure

    if (temperature_c >= -20 && temperature_c <= 40) {
        Serial.println("The MS5803-14BA pressure sensor - OK");
        MS5803_state = true;
    }
    else {
        Serial.println("The MS5803-14BA pressure sensor - NOT OK");
        MS5803_state = false;
        fault = true;
    }

    SIM800L_Serial.println("AT"); // A dummy message to wake the component
    delay(200);

    SIM800L_Serial.println("AT");
    if ((readSerial().indexOf("OK")) != -1 ) {
        Serial.println("The SIM800L GSM module - OK");
    }
    else {
        Serial.println("The SIM800L module - NOT OK");
        fault = true;
    }

    Serial.print("Battery voltage is ");
    Serial.print(batt_volt);
    Serial.print("mV ");
    Serial.print(batt_percent);
    Serial.println("%");

    if (batt_volt >= 3300) {
        Serial.println("The percentage of battery is within the allowable
range");
    }
    else {
        Serial.println("The percentage of battery is NOT within the allowable
range");
        fault = true;
    }

    if (fault) { // One of the components is not
working or the battery is low
        message = message + "The device NOT working properly, Battery Level: "
+ String(batt_percent) + "%";
        digitalWrite(RedledPin, 0); // Low battery -> Red led ON
        digitalWrite(GreenledPin, 255); // Low battery -> Green led OFF
    }
    else { //All components work and the battery
is good

```

```

        message = message + "The device is working properly, Battery Level: " +
String(batt_percent) + "%";
        digitalWrite(RedledPin, 255);      // OK battery -> Red Led OFF
        digitalWrite(GreenledPin, 0);      // OK battery -> Green led ON
    }

    delay(5000);

    Sending_sms(message);                    // Sending report message to a
control center with the SIM800L

    digitalWrite(RedledPin, 255);           // Turns off the Red LED
    digitalWrite(GreenledPin, 255);        // Turns off the Green LED
}
//-----

/* --- readSerial() -----
The function returns the output of the SIM800L.
Input: None
Output: SIM800L_Serial.readString - The output of the SIM800L.
*/
String readSerial() {
    int _timeout = 0;
    while (!SIM800L_Serial.available() && _timeout < 1000 )    {
        delay(13);
        _timeout++;
    }
    if (SIM800L_Serial.available()) {
        return SIM800L_Serial.readString();
    }
}

```

8.2. נספח 2: בדיקות המערכת (STD)

8.2.1. בדיקות הרכיבים החשמליים בנפרד

מס	הבדיקה	פעולה לביצוע	תוצאה צפויה	תוצאה	הערות
1	בדיקת תקשורת טורית בין המחשב למיקרו בקר (USB)	כתיבת קוד לבדיקת התקשורת. ניסיון צריכת למיקרו בקר בעזרת סביבת העבודה (IDE) של ארדואינו.	במסך ה serial monitor של IDE נראה שינויים בהתאם לקוד.	עובד כנדרש, קיימת תקשורת תקינה בין המחשב לבקר. הצריבות מתבצעות בהצלחה.	
2	בדיקת צריכת האנרגיה של המיקרו בקר	מדידת צריכת הזרם של המיקרו בקר בשני מצבים : במצב רגיל (הכל מופעל) במצב שינה עמוק	בהתאם למפרט הטכני של המיקרו בקר.	ACT - 5 mA PDS - 1 mA	ניתן להוריד עוד את צריכה, ע"י הסרת הלדים (0.2mA)
3	בדיקת תקשורת טורית סינכרונית (UART) בין המיקרו בקר לבקר תקשורת סולארית	כתיבת קוד וצריבתו במיקרו בקר לשליחת SMS לטלפון שלנו.	קבלת ההודעה בטלפון.	עובד כנדרש, התקבלה הודעת SMS עם התוכן המבוקש למספר שהוגדר.	
4	בדיקת תקשורת טורית סינכרונית בין המיקרו בקר לחיישן עומק	כתיבת קוד וצריבתו במיקרו בקר למדידת העומק בו נמצא החיישן והצגתו במסך ה - serial monitor.	הצגת העומק בו נמצא החיישן במסך ה- serial monitor הקריאה תהיה תואמת לעומק בו נמצא החיישן.	עובד כנדרש, קורא באופן תקין את העומק בו נמצא החיישן.	
5	בדיקת התאמת פעילות המנוע למפרט הטכני שלו	חיבור המנוע לווסת ולמקור מתח לבדיקת יכולתו המכאנית לסיבוב הווסת.	בהתאם למפרט הטכני של המנוע.	עובד כנדרש, המנוע מצליח לפתוח ולסגור את הווסת.	
6	בדיקת התאמת פעילות המנוע למפרט הטכני שלו	מדידת צריכת הזרם של המנוע בשני מצבים : עם עומס (הווסת מחובר) בלי עומס	בהתאם למפרט הטכני של המנוע.	No Load - 42 mA Load - 330 mA Stall - 670 mA	
7	בדיקת ממיר המתח + מטען סוללה בהתאם למפרט הטכני שלו.	בדיקת יכולת הרכיב להגן על הסוללה במצבים הבאים : Overcharge, Discharge and Short circuit	בהתאם למפרט הטכני של ממיר המתח + מטען סוללה	עובד כנדרש, קיימת הגנה על הסוללה.	

מס	הבדיקה	פעולה לביצוע	תוצאה צפויה	תוצאה	הערות
8	בדיקת ממיר המתח + מטען סוללה בהתאם למפרט הטכני שלו.	בדיקת ממיר המתח וצריכת האנרגיה שלו	בהתאם למפרט הטכני של ממיר המתח + מטען סוללה	No Load - 462μA	
9	בדיקת טעינה אלחוטית כמקור מתח למטען הסוללה	לבדוק שרכיב הטעינה האלחוטית מספק מספיק זרם לטעינת הסוללה.	מטען הסוללה יקבל את הזרם הדרוש להטענת הסוללה. מתחילה להיטען לפי הצפי.	עובד כנדרש, בהנחת הרכיב על המשטח טעינה הסוללה נטענת	
10	בדיקת כשירותו המכאנית של הווסת	חיבור הווסת למחסנית CO2 ובדיקת דליפות.	הווסת ינקב את המחסנית ולא יאפשר בריחת גז עד סיבוב הווסת.	עובד כנדרש, אין בריחת גז	
11	בדיקת יכולות הרכיב לזיהוי מגע עם מים	הכנסת הפרובים למים. צריבת הקוד המתאים למיקרו בקר	הרכיב יזהה מגע עם מים.	עובד כנדרש, בעת הכנסת הפרופים למים, נדלק ה-LED	
11	בדיקת אטימות המים של המארז	הכנסת המארז למקור מים עם צמר גפן	המארז עמיד למים כפי שמצוין במפרט שלו	המארז אטום למים	

8.2.2. בדיקות המערכת בשלמותה

מס	הבדיקה	פעולה לביצוע	תוצאה צפויה	תוצאה	הערות
1	בדיקת צריכת כלל המערכת במצב שינה (מחוץ למקור מים)	מדידת צריכת כלל המערכת במצב שינה	תואם את החישובים שבוצעו.	קצת יותר גבוה מהמחושב Sleep - 4.5 mA	השינוי נובע מהוספת כמה רכיבים שלא נכללו בחישובים
2	הפעלה יזומה של בדיקת תקינות המערכת בעזרת כפתור ייעודי	לחיצה על כפתור המשמש לביצוע בדיקת תקינות של המערכת.	לאחר לחיצת הכפתור נקבל הודעת SMS ובה מידע על אחוז הסוללה ואם המערכת תקינה.	המערכת מתעוררת ממצב שינה ומבצעת סבב בדיקות לרכיבי המערכת. לאחר הבדיקה שולחת הודעה ומדליקה לד בהתאם לתוצאות הבדיקה.	
3	האם המערכת נדלקת כתוצאה מהימצאות במקור מים.	הכנסת הפרובים למקור מים.	המערכת תזהה מגע עם מים ותעורר ממצב שינה. בשלב זה תתחיל למדוד לחצים.	עובד כנדרש	
4	בדיקת צריכת אנרגיה של כלל המערכת בכניסה למקור מים	מדידת צריכת כלל המערכת בעת הכנסת הפרובים למים	תואם את החישובים שביצענו.	קצת יותר גבוה מהמחושב 15 mA	השינוי נובע מהוספת כמה רכיבים שלא נכללו בחישובים
5	האם המערכת מצליחה לזהות מקרה טביעה לפי מה שהוגדר	הכנסת המערכת למים מתחת לעומק הסף המוגדר	המערכת תזהה כי נחצה עומק הסף ותפעיל טיימר. לאחר סיום הטיימר תפעיל מנגנונים שונים להצלת הילד	עובד כנדרש	
6	פתיחת הווסט לשחרור גז ה - CO2 וסגירתו	הכנסת המערכת למים מתחת לעומק הסף המוגדר	סיבוב הווסט ושחרור הגז ובסוף סגירה של הווסט	עובד כנדרש	
7	שליחת SMS בעת זיהוי טביעה	שליחת SMS לפלאפון מוגדר שיתריע על מקרה הטביעה	קבלת הודעת SMS	עובד כנדרש	

8.3.1. הקדמה

ניהול הסיכונים בפרויקט יתבצע באופן שוטף לאורך כל הפרויקט. באופן זה ניתן לעקוב אחר מצב הסיכונים העדכני בפרויקט.

ניהול הסיכונים משמש אמצעי חשוב וקריטי שמטרתו להפחית סיכונים קיימים ואפשריים במהלך הפרויקט. ניהול הסיכונים מאפשר זיהוי מוקדם של תקלות פוטנציאליות אפשריות, מתן פתרונות בזמן אמת ומניעת נזקים צפויים.

בניהול סיכונים נבצע זיהוי סיכונים עיקריים בתהליך העבודה, ניתוח הסתברותם להופיע והשפעתם על התהליך. עם זיהוי והערכת הסיכונים נערכת חשיבה מחודשת על מנת להקטין את הסיכונים למינימום האפשרי.

8.3.2. מחזור ניהול הסיכונים בפרויקט

ניהול סיכונים הינו תהליך מובנה במסגרת מוגדרת של עבודה שוטפת בפרויקט. מעורבים בו הסטודנטים שעובדים על הפרויקט וכן גם המנחה.

תהליך ניהול הסיכונים כולל:

- זיהוי והערכת סיכונים
- קביעת פעולות מונעות ושיטות ופתרונות להורדת הסיכון
- פיקוח ובקרה על אופן הטיפול בסיכונים וחוזר חלילה.

8.3.3. זיהוי והערכת סיכונים

זיהוי הסיכונים יעשה בשיתוף הסטודנטים והמנחה של הפרויקט. כל אחד יעלה את הסיכונים שלדעתו עלולים להופיע במהלך העבודה על הפרויקט.

על מנת לקבל חיווי כמותי של חומרת הסיכונים מחושבת החומרה של כל סיכון בהתאם לנוסחה הבאה:

$$\text{חומרת הסיכון} = \text{הסתברות הסיכון לקרות} \times \text{נזק}$$

כלומר, חומרת הסיכון היא המכפלה של נזק בעקבות הסיכון בהסתברות שהסיכון אכן יקרה.

מדדי נזק - נמדד בין הערכים 1-4

כש-0 הוא ללא נזק ואילו 4 הוא הנזק החמור ביותר.

מדדי ההסתברות - ההסתברות נמדדת בין הערכים 1-4

כש-0 היא ההסתברות הנמוכה מאוד ביותר ואילו 4 היא ההסתברות הגבוהה ביותר.

המכפלות בין מדדי הנזק ומדדי ההסתברות עבור כל סיכון מוצגות בצבעי רמזור - אדום, צהוב, ירוק כשהצבע האדום מסמן חומרת סיכון גבוה והצבע הירוק מסמן חומרת סיכון נמוך.

טבלה מס' 1 - קביעת רמת / עוצמת הסיכון					
הסתברות		1 - זניח	2 - נמוכה	3 - בינונית	4 - גבוהה
נזק	4 - חמור	4	8	12	16
	3 - בינוני	3	6	9	12
	2 - מועט	2	4	6	8
	1 - קל	1	2	3	4

טבלה מס' 2 - קריטריונים לקביעת הסתברות / סבירות להתרחשות הסיכון	
הסתברות / סבירות לסיכון	הסבר
4 - גבוהה	עלול להתרחש בכל עת.
3 - בינונית	קיימת סבירות גבוהה לסיכון. יכול לקרות כמה פעמים.
2 - נמוכה	עלול להתרחש לעיתים רחוקות.
1 - זניחה	עלול להתרחש לעיתים רחוקות אך כנראה לא תתרחש אף פעם.

טבלה מס' 3 - קריטריונים לקביעת חומרת הסיכון	
נזק	הסבר
4 - חמור	קיימת השפעה על הלו"ז ופגיעה ביעדי הפרויקט. לא תתאפשר עמידה בביצועים הנדרשים למרות שננקטו הפעילויות הנדרשות.
3 - בינוני	תגרם הפחתת ביצועים ברמה בינונית במידה והנושא לא יטופל. נדרשת פעילות נוספת על מנת לעמוד בלו"ז.
2 - מועט	קיימת השפעה מועטה על ביצוע תהליך העבודה. קיימת אפשרות של ביצוע שינויים מזעריים בתכנון ובתהליך העבודה למניעת פגיעה בלו"ז.
1 - קל	ההשפעה על תהליך העבודה תהיה נמוכה מאוד או בכלל לא ומכאן שלא תהיה פגיעה בפרויקט.

8.3.4. קביעת פעולות מונעות, שיטות ופתרונות להורדת הסיכון

לאחר שהסיכון זוהה ובוצעה הערכה שלו מבחינת הסתברות ונזק, יש לקבוע פתרונות בתהליך העבודה שימנעו נזקים במידה והסיכון יתממש. הפתרונות צריכים להקיף תהליכים קרובים ביותר לסיכוי שבהם הסיכון יתרחש.

כדאי להקטין הסיכון. ננסה לבצע כמה פעולות :

1. הקטנת ההסתברות שהסיכון יקרה.
2. הקטנת השפעתו של הסיכון על הפרויקט.
3. ביצוע שתי הפעולות יחד.

כל הפתרונות יהיו מתועדים וימצאו במעקב כך שניתן יהיה לבדוק את יעילותם והגנתם מפני התרחשות הסיכון.

8.3.5. טבלת ניהול סיכונים:

#	תיאור סיכון	תחום	משמעות	הסתברות 1-4	נזק 1-4	חומרה 0-16	רמת חומרה	התמודדות / הקטנת הסיכון	שארית סיכון
1	חוסר התאמה בין רכיבים לבין נתוני המפרט שלהם	טכני	חיפוש חלופה - רכיב אחר	1	4	4	בינונית	*בחירת רכיבים אמינים בעלי ביקורות טובות ושהוכיחו את עצמם בשוק *שימוש ברכיב תוך כדי חיפוש חלופה *קנייה של כמה רכיבים מאותו סוג (ספיירים)	3
2	תקלות רכיבים בזמן בדיקות		חיפוש חלופה - רכיב אחר	3	2	6	בינונית	*בחירת רכיבים אמינים בעלי ביקורות טובות ושהוכיחו את עצמם בשוק *חידוד והבנת ערכי סף של רכיבים שונים, כמו מתח וזרם מקסימלי *קנייה של כמה רכיבים מאותו סוג (ספיירים)	4
3	עיכובים בפיתוח התוכנה		למידת הנושא בצורה יותר מעמיקה	2	4	8	בינונית	*התייעצות עם בעל מקצוע - מומחה *חיפוש מידע בפורומים שונים	6
4	אי עמידת המערכת בדרישות האפיון		*בדיקת הגורמים לבעיה *תיקון התכנון *בדיקות נוספות	3	4	12	גבוהה	*בדיקה מקיפה של הרכיבים לפני הקנייה, במבט על כלל המערכת *נגדיר טווחים מדויקים באלגוריתם הזיהוי על מנת לעמוד באחוזי השגיאה הדרושים	8
5	מחלה של אחד הסטודנטים	ניהולי	אי עמידה בלוח"ז	2	1	2	נמוכה	*חפיפה ושיתוף ידע בין הסטודנטים *תיעוד מפורט לשימור ידע	1
6	מילואים		אי עמידה בלוח"ז	2	3	6	בינונית	*שינוי לוח הזמנים ברגע קבלת הצו לצמצום העיכוב. *לנסות לבטל צווי מילואים	2
7	עזיבת כוח אדם בפרויקט (סטודנט או מנחה)		אי עמידה בלוח"ז	1	4	4	בינונית	*חפיפה ושיתוף ידע בין הסטודנטים *תיעוד מפורט לשימור ידע	4

