



בית ספר תיכון ומכללה טכנולוגית
חיל האוויר ועמל 1 הולץ



כתובת : קיבוץ גלילות 150 תל-אביב

טלפון : 03-6835393

FAX : 03-6828543

סמל : 750044

פרוייקט גמר לתואר הנדסאי

התמחות : מערכות אלקטרונית.

הנושא : JukeBox – תיבת מוזיקה

שם המגישים: רוני דודקמן, אריה גלגור

שם המנחה : אריה פורת

מגמה : אלקטרוניקה מחשבים.

תאריך : מאי 2018 .

כתובת : קיבוץ גלויות 150 תל-אביב

טלפון : 03-6835393

פקס : 03-6828543

סמל : 570044

הצעת נושא לפרויקט
למילוי חלקי של הדרישות לקבלת
תואר הנדסאי במגמת חשמל - אלקטרוניקה
בהתמחות: מערכות אלקטרוניות

שם הנושא: תיבת נגינה - JUKEBOX

תואר: MA

שם המנחה: אריה פורת

בהתייחס לנאמר בחוברת "פרויקט ועבודת גמר במסלול על תיכוני
(כיתות י"ג, י"ד) במגמת אלקטרוניקה ומחשבים (תמוז התשנ"ד – יוני 1994)".

אופי עבודת הגמר:

- חקר הנדסי ע"פ הנאמר בפרק ד', סעיף 2.1 ☐
- בדיקת התכנות, ע"פ הנאמר בפרק ד', סעיף 2.2 ☐
- תכנון ע"פ הנאמר בפרק ד', סעיף 2.3 ☐
- תכנון ופיתוח, ע"פ הנאמר בפרק ד', סעיף 2.4 ☒
- תכנון, פיתוח ומימוש מערך בדיקה, ע"פ הנאמר בפרק ד', סעיף 2.5 ☐
- תכנון מערך החזקה, ע"פ הנאמר בפרק ד', סעיף 2.6 ☐

מקום הביצוע:

מכללה ☒

צה"ל ☐

תעשייה ☐

מוסד מחקר ☐

חתימת המנחה: _____

תאריך הגשת ההצעה: _____

חתימת המרכז
וחותמת המכללה: _____

שם מרכזת המגמה: נתי ארזי

הצהרת הסטודנט: לאחר שעיינתי בחוברת נוהלי ביצוע של עבודות גמר / פרויקטים
לטכנאים והנדסאים ובהצעה, ולאחר הסברי המנחה, הנני מאשר בזאת שההצעה על חלקיה
מובנת לי ומחייבת אותי.

חתימה: _____

תאריך: _____

תיאור הנושא: (פרויקט ל 2 סטודנטים)

תיבת הנגינה היא ה Jukebox שהיה בה פתח להכנסת מטבע ואז ניתן היה לבחור שיר מתוך אוסף שירים ולשמוע את השיר.

תיבת הנגינה שלנו תנגן שיר מתוך אוסף שירים המוצג בתצוגת TFT LCD. המשתמש יבחר על ידי נגיעה במסך את השיר הרצוי לו והוא יושמע בעזרת נגן MP3. בחירת שיר מתבצעת או על ידי הכנסת מטבע בערך רצוי (לדוגמא שקל) בעזרת מערכת בוררת מטבעות או בעזרת צ'יפ RFID שנקרא על ידי קורא RFID.

מפרט טכני:

1. מיקרו בקר בכרטיס ממשפחת הארדואינו .
2. בורר מטבעות Multi Coin Selector .
3. נגן MP3 המתחבר בתקשורת טורית.
4. קורא RFID .
5. תצוגת TFT LCD.

פירוט הדרישות מסטודנט א' (ראה תרשים המלבנים בהמשך)

1. לימוד מערכת נגן MP3, חיבור לארדואינו, הקלטת השירים הרצויים וקביעת תכנה מתאימה.
2. הכרת תצוגת TFT LCD, חיבור אל הארדואינו וקביעת תכנה מתאימה.
3. שילוב בין החומרה והתוכנה.
4. כתיבת ספר הפרויקט.

ביבליוגרפיה:

1. האתר של אריה פורת - www.arikporat.com
2. האתר של ארדואינו - <https://www.arduino.com>
3. דפי נתונים מהאינטרנט.

נספחים:

1. תרשים מלבני של המעגל

שם הסטודנט: רועי דודקמן

ת.ז. _____

החלטת הצוות המאשר: הנושא אושר לביצוע

שם וחתימת ראש הצוות המאשר

תאריך

תיאור הנושא: (פרויקט ל 2 סטודנטים)

תיבת הנגינה היא ה Jukebox שהיה בה פתח להכנסת מטבע ואז ניתן היה לבחור שיר מתוך אוסף שירים ולשמע את השיר.

תיבת הנגינה שלנו תנגן שיר מתוך אוסף שירים המוצג בתצוגת TFT LCD. המשתמש יבחר על ידי נגיעה במסך את השיר הרצוי לו והוא יושמע בעזרת נגן MP3. בחירת שיר מתבצעת או על ידי הכנסת מטבע בערך רצוי (לדוגמא שקל) בעזרת מערכת בוררת מטבעות או בעזרת צ'יפ RFID שנקרא על ידי קורא RFID.

מפרט טכני:

1. מיקרו בקר בכרטיס ממשפחת הארדואינו.
2. בורר מטבעות Multi Coin Selector.
3. נגן MP3 המתחבר בתקשורת טורית.
4. קורא RFID.
5. תצוגת TFT LCD.

פירוט הדרישות מסטודנט ב' (ראה תרשים המלבנים בהמשך)

1. לימוד מערכת בחירת מטבע, חיבורה לארדואינו וכתובת תכנה מתאימה.
2. לימוד מערכת ה RFID, חיבורה לארדואינו וכתובת התכנה המתאימה.
3. שילוב בין החומרה והתוכנה.
4. כתיבת ספר הפרויקט.

ביבליוגרפיה:

1. האתר של אריה פורת - www.arikporat.com
2. האתר של ארדואינו - <https://www.arduino.com>
3. דפי נתונים מהאינטרנט.

נספחים:

1. תרשים מלבני של המעגל

שם הסטודנט: אריה גלגור

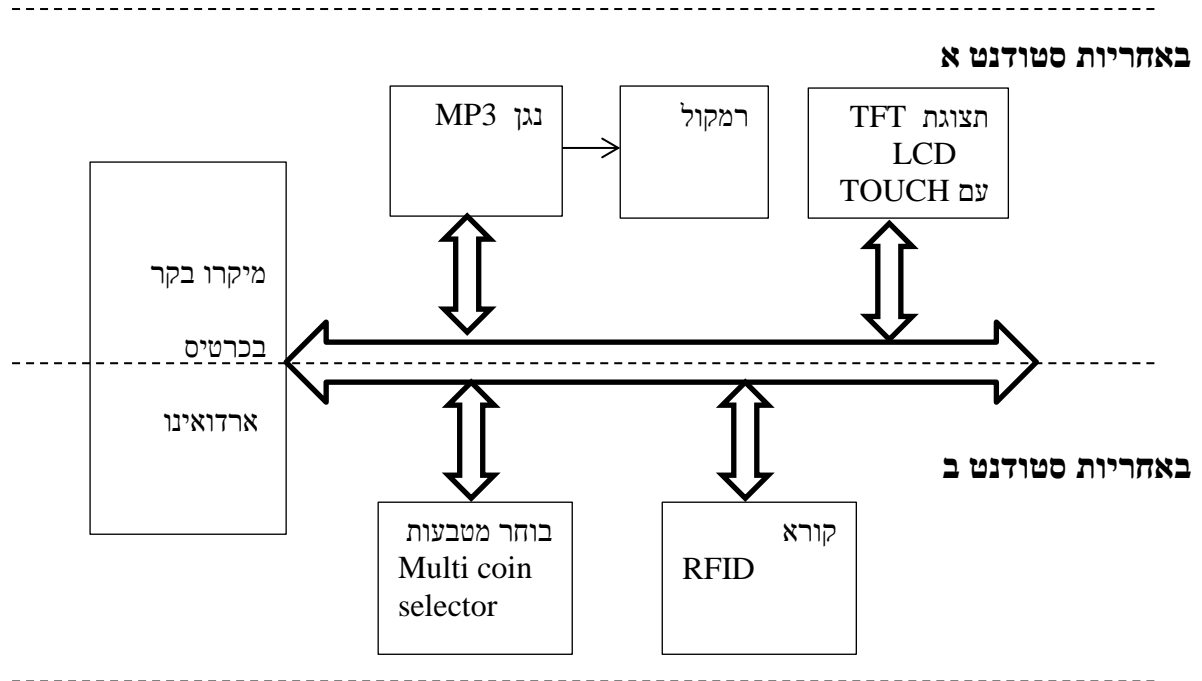
ת.ז. _____

החלטת הצוות המאשר: הנושא אושר לביצוע

שם וחתימת ראש הצוות המאשר

תאריך

תרשים מלבנים



הצהרת הסטודנט

אני: רועי דודקמן ת.ז. 206507709

החתום מטה, מצהיר בזאת שכל עבודת הפרויקט המוגשת בחוברת זו הינו פרי עבודתי בלבד, על בסיס הנחייתו של המנחה ותוך הסתמכות על מקורות הידע והמידע האחרים המצויינים בביליוגרפיה המובאת בסיום חוברת זו.

אני מודע לאחריות שהנני מקבל על עצמי ע"י חתימתי על הצהרה זו שכל הנאמר בה הינה אמת ורק אמת.

חתימת מגיש החוברת: _____.

אישור המנחה:

הריני לאשר הגשת החוברת להערכה.

הצהרת הסטודנט

אני: אריה גלגור ת.ז. 208723734

החתום מטה, מצהיר בזאת שכל עבודת הפרויקט המוגשת בחוברת זו הינו פרי עבודתי בלבד, על בסיס הנחייתו של המנחה ותוך הסתמכות על מקורות הידע והמידע האחרים המצויינים בביליוגרפיה המובאת בסיום חוברת זו.

אני מודע לאחריות שהנני מקבל על עצמי ע"י חתימתי על הצהרה זו שכל הנאמר בה הינה אמת ורק אמת.

חתימת מגיש החוברת: _____.

אישור המנחה:

הריני לאשר הגשת החוברת להערכה.

תקציר

תיבת הנגינה היא ה Jukebox שהיה בה פתח להכנסת מטבע ואז ניתן היה לבחור שיר מתוך אוסף שירים ולשמוע את השיר.

תיבת הנגינה שלנו תנגן שיר מתוך אוסף שירים המוצג בתצוגת TFT LCD. המשתמש יבחר על ידי נגיעה במסך את השיר הרצוי לו והוא יושמע בעזרת נגן MP3. אפשרות בחירת שיר מתבצעת או על ידי הכנסת מטבע בערך רצוי (לדוגמא שקל) בעזרת מערכת בוררת מטבעות או בעזרת צ'יפ RFID שנקרא על ידי קורא RFID.

מפרט טכני

1. מיקרו בקר Atmega 328 בכרטיס ארדואינו אנו.
2. בורר מטבעות Multi Coin Selector .
3. נגן MP3 המתחבר בתקשורת טורית.
4. קורא RFID .
5. תצוגת TFT LCD בגודל 3.5 אינץ'.

תוכן עניינים

2-5.....	הצעת הפרויקט
6-7.....	הצהרת הסטודנט
8.....	תקציר
9.....	מפרט טכני
10.....	תוכן עניינים
11.....	רשימת תרשימים/איורים

חומרה:

12.....	מבוא
13.....	תרשים סכימה מלבנית
14-16.....	הסבר תרשים סכימה מלבנית
17.....	סכימה חשמלית
18-21.....	הסבר סכימה חשמלית

תוכנה:

22.....	מבוא
23-34.....	תרשימי זרימה+הסברים
35-36.....	טבלת פונקציות
37-53.....	התוכנה בשפת C++

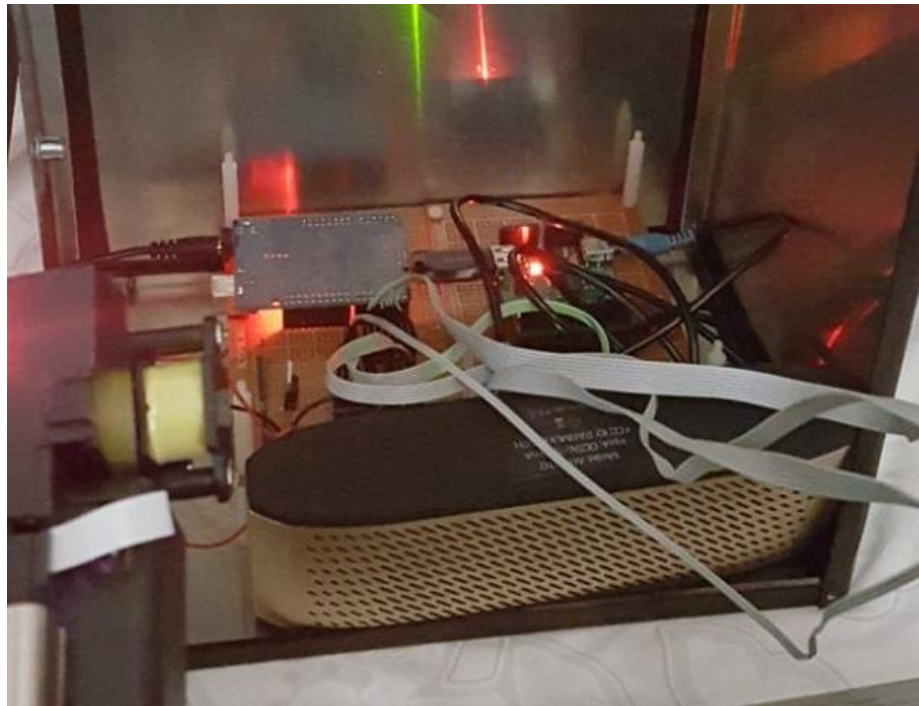
כלים:

54.....	תכנון מכני
55.....	כלי תכנון ממוחשבים
56.....	מדידות ותוצאות
57-59.....	הוראות הפעלה
60.....	איתור ליקויים אפשרי
61.....	סיכום ומסקנות
62.....	ביבליוגרפיה
63-69.....	נספחים

רשימת תרשימים/איורים

12.....	איור 1: מעגלי האלקטרוניקה בפרויקט.
13.....	איור 2: תרשים סכימה מלבנית.
14.....	איור 3: כרטיס ארדואינו אנו.
15.....	איור 4: תצוגת TFT של 3.5 עם מסך מגע.
15.....	איור 5: מכונת קבלת מטבעות Coin acceptor.
16.....	איור 6: קורא RFID.
16.....	איור 7: לוח נגן MP3.
17.....	איור 8: סכמה חשמלית.
18.....	איור 9: חיבור כרטיס הארדואינו אל תצוגת ה-TFT.
19.....	איור 10: חיבור מכונת קבלת המטבעות אל הארדואינו.
20.....	איור 11: חיבור קורא RFID אל ארדואינו.
21.....	איור 12: חיבור נגן MP3 אל הארדואינו.
23.....	תרשים זרימה מספר 1: תוכנת הארדואינו.
23.....	תרשים זרימה מספר 2: פונקציית האתחול.
25.....	תרשים זרימה מספר 3: הפונקציה הראשית LOOP.
26.....	תרשים זרימה מספר 4: פונקציית מסך התפריט.
28.....	תרשים זרימה מספר 5: פונקציית ה-TOUCH.
30.....	תרשים זרימה מספר 6: פונקציית ה-TrackScreen.
32.....	תרשים זרימה מספר 7: פונקציית ה-checkWorkStaten.
33.....	תרשים זרימה מספר 8: פונקציית ה-coinAcceptorn.
34.....	תרשים זרימה מספר 9: פונקציית ה-RFID.
54.....	איור 13: קופסת הפרויקט.
56.....	איור 14: מדידות על הסקופ.

חומרה – מבוא



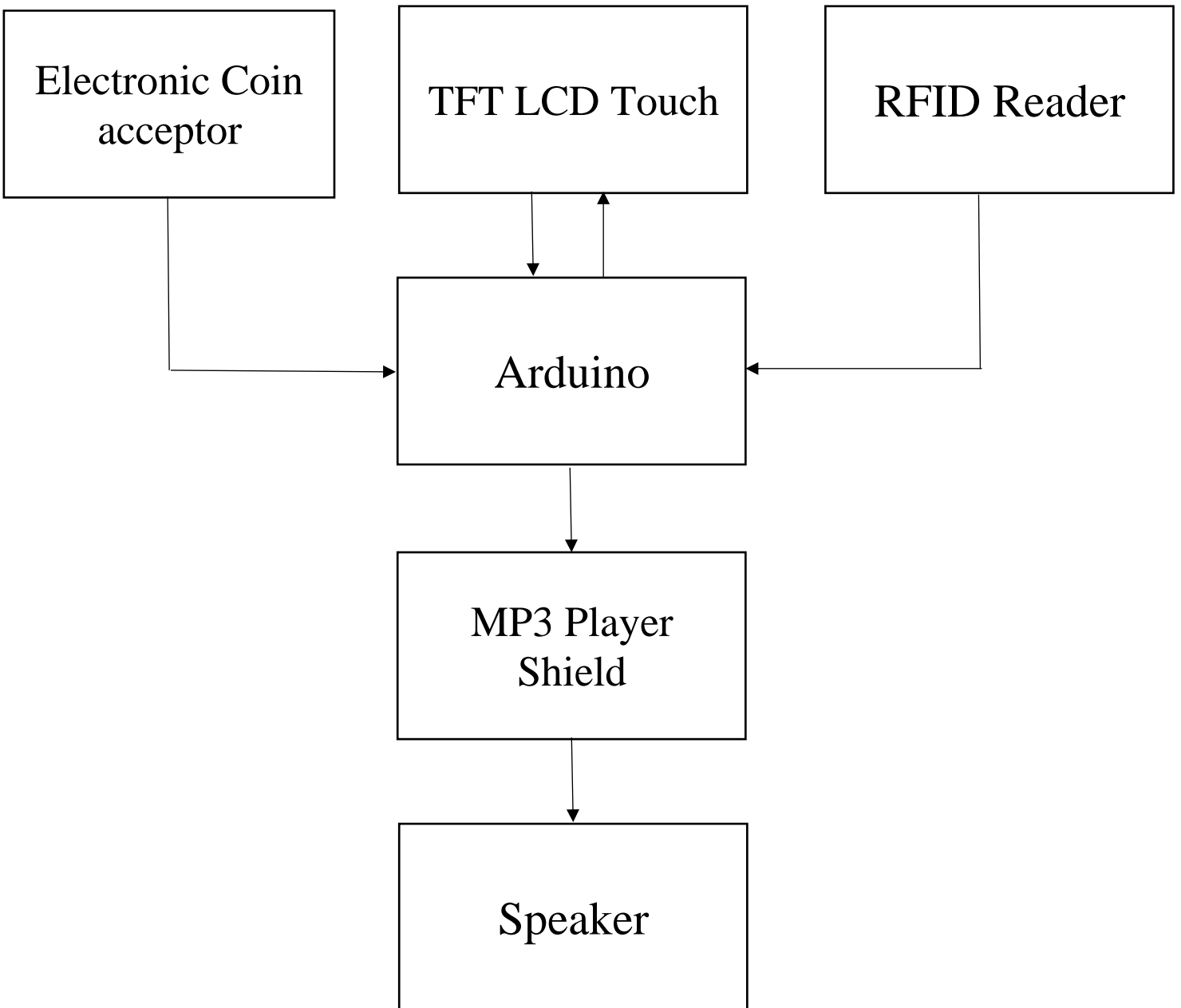
איור 1 : מעגלי האלקטרוניקה בפרויקט

הפרויקט שלנו מורכב משני רכיבי קלט, רכיב פלט, רכיב שימש גם כקלט וגם כפלט וארדואינו אשר ישמש כבקר על הכל.

רכיבי הקלט שנשתמש בהם יהיו Coin acceptor ו-RFID Reader. המשתמש יכניס מטבע לקולט המטבעות או שיעביר תג שיזוהה על ידי ה-RFID והם יכניסו אינפורמציה מתאימה לארדואינו. בנוסף לרכיבי הקלט יהיה גם מסך מגע TFT אשר ישמש כקלט מן המשתמש (מגע) ופלט כתגובה לפעולות המשתמש (מה שמופיע על המסך).

לאחר קבלת הקלט מהרכיבים המסך יגיב בהתאם להוראות מהארדואינו והארדואינו ישלח הוראה לנגן המוזיקה (MP3) לבצע את השיר הרצוי שנבחר על ידי המשתמש.

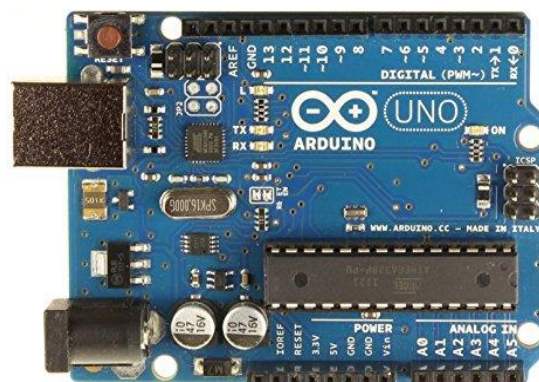
חומרה- תרשים סכימה מלבנית



איור 2 : סכמה מלבנית

הסבר תרשים סכימה מלבנית:

א. ארדואינו



איור 3 : כרטיס ארדואינו אונו

הארדואינו אונו הוא כרטיס האלקטרוניקה שהשתמשו בו יש את המיקרו בקר Atmega328 ששולט על כל התהליכים בפרויקט. את התכנית שפיתחנו העלנו לזיכרון ה FLASH שלו שהוא בגודל 32Kbytes (מתוכם 2Kbytes הם תוכנת המוניטור). בעזרת ההדקים הדיגיטאליים והאנלוגיים של הכרטיס התחברנו אל שאר המודולים בפרויקט.

כפי שניתן לראות בתרשים המלבני הארדואינו מחובר לכל רכיבי הקלט/פלט. ה RFID וה Coin acceptor מתחברים לארדואינו כרכיבי קלט אשר יודיעו לו על כל פעולה של המשתמש בכרטיס. בנוסף ניתן לראות שהמסך מחובר גם כקלט וגם כפלט משום שתפקידו הוא לקבל מידע מן המשתמש על ידי לחיצה (מסך מגע) וגם להציג למשתמש מידע על המסך. לארדואינו מחובר גם רכיב MP3 אשר יקבל מידע מהארדואינו על איזה מהשירים שבספרייה שמובנית בו לנגן והשיר הנבחר ינוגן דרך רמקול אשר מחובר אליו בכבל AUX.

ב. תצוגת TFT LCD



איור 4 : תצוגת ה TFT של 3.5" עם מסך מגע

תצוגת הTFT הינה תצוגת LCD צבעונית בעלת מסך מגע המאפשרת קליטת נתונים מהמשתמש תוך כדי לחיצה על האפשרויות המודפסות על המסך. המסך שלנו יציג את כמות הכסף שנכנס למכונה ויציג את אפשרויות בחירת השירים ככפתורים שניתן ללחוץ עליהם.

ג. מכונת קבלת מטבעות - COIN ACCEPTOR

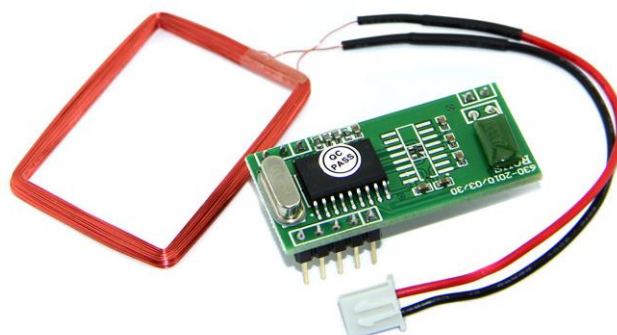


איור 5 : מכונת ברירת המטבעות CH 926

מכונת ברירת המטבעות יכולה לזהות עד 6 סוגי מטבעות ומגיבה לכל סוג בצורה שונה בהתאם

למה שהוגדר מראש (תכנות עצמי). בפרויקט שלנו תכנתנו את המכונה שתזהה רק את המטבעות:
1 שקל, 2 שקל ו5 שקל, כאשר המכונה מתוכנתת להגיב בכמות פולסים לפי ערך המטבע.

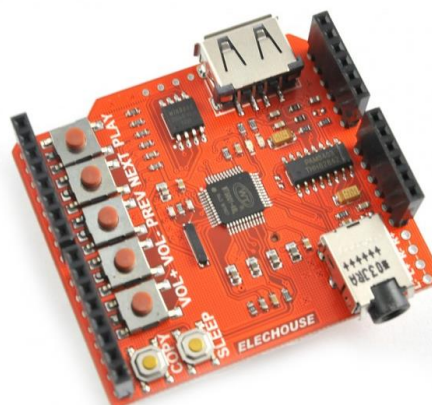
ד. קורא RFID - RFID READER



איור 6 : קורא RFID מסוג RDM630 (מימין) והציפים המתאימים לו.

טכנולוגית ה-RFID מאפשרת איתור פריטים וזיהוי חפצים בצורה אלחוטית באמצעות תדרי רדיו, ללא צורך בשדה ראייה או במגע ישיר מול החפץ אותו רוצים לזהות.
טכנולוגית RFID משתמשת בשדה אלקטרומגנטי הנוצר ע"י קורא קרבה (RFID reader) להעברת אנרגיה לתגי קרבה (RFID tags) בטווחים קצרים. בעזרת האנרגיה שנוצרת ע"י השדה, תגי הקרבה מסוגלים לפעול ללא צורך במקור מתח חיצוני ומשדרים לקורא קרבה את תוכן הזיכרון הצרוב בתוכם.

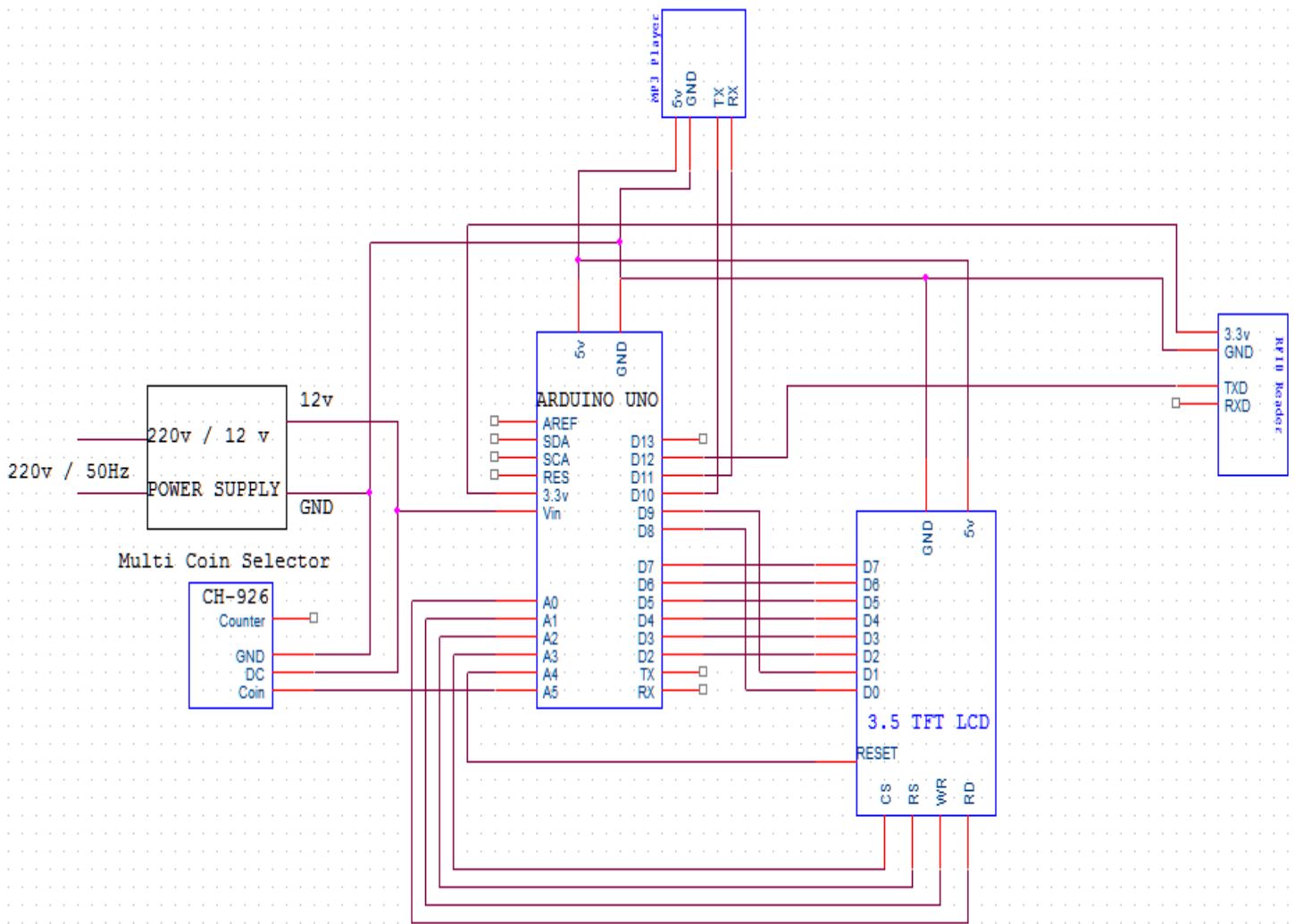
ה. נגן MP3 - Mp3 player Shield



איור 7 : לוח נגן MP3

נגן המוזיקה הינו רכיב המשמש להפעלת קבצי מוזיקה מסוג MP3 . את קבצי המוזיקה הנגן מקבל בעזרת זיכרון חיצוני UDISK או כרטיס SD. לרכיב יש יציאה לכבל AUX שמתחבר לרמקול חיצוני.

חומרה- סכימה חשמלית



איור 8 : סכמה חשמלית של הפרויקט

בסכמה שלפנינו ניתן לראות לאיזה הדקים כל חלק מתחבר בארדואינו:

TFT: הדקים D0-D7 של המסך מחוברים להדקים D2-D9 בארדואינו והרגליים

RD,WR,RS,CS,RESET מחוברים בהתאמה להדקים A0-A4 שבארדואינו.

RFID: הכרטיס מתחבר בתקשורת טורית רגילה אל הארדואינו. רגלי התקשורת הטורית RXD,TXD של

הכרטיס מחוברים להדקים D12,D13 בהתאמה ואנחנו משתמשים בתכנת softwareSerial

שמדמה קליטת נתונים טורית. לא מחברים את הדקי התקשורת הטורית של הקורא לרגלי התקשורת

של הארדואינו כי תהיה לנו בעיה ב UPLOAD מהמחשב לארדואינו.

MP3: רגלי התקשורת הטורית RX,TX מחוברים להדקי התקשורת הוירטואלית שבארדואינו

D10,D11 בהתאמה.

Coin Acceptor: למכונת המטבעות יש הדק Coin שממנו נשלחים הפולסים אל הארדואינו להדק

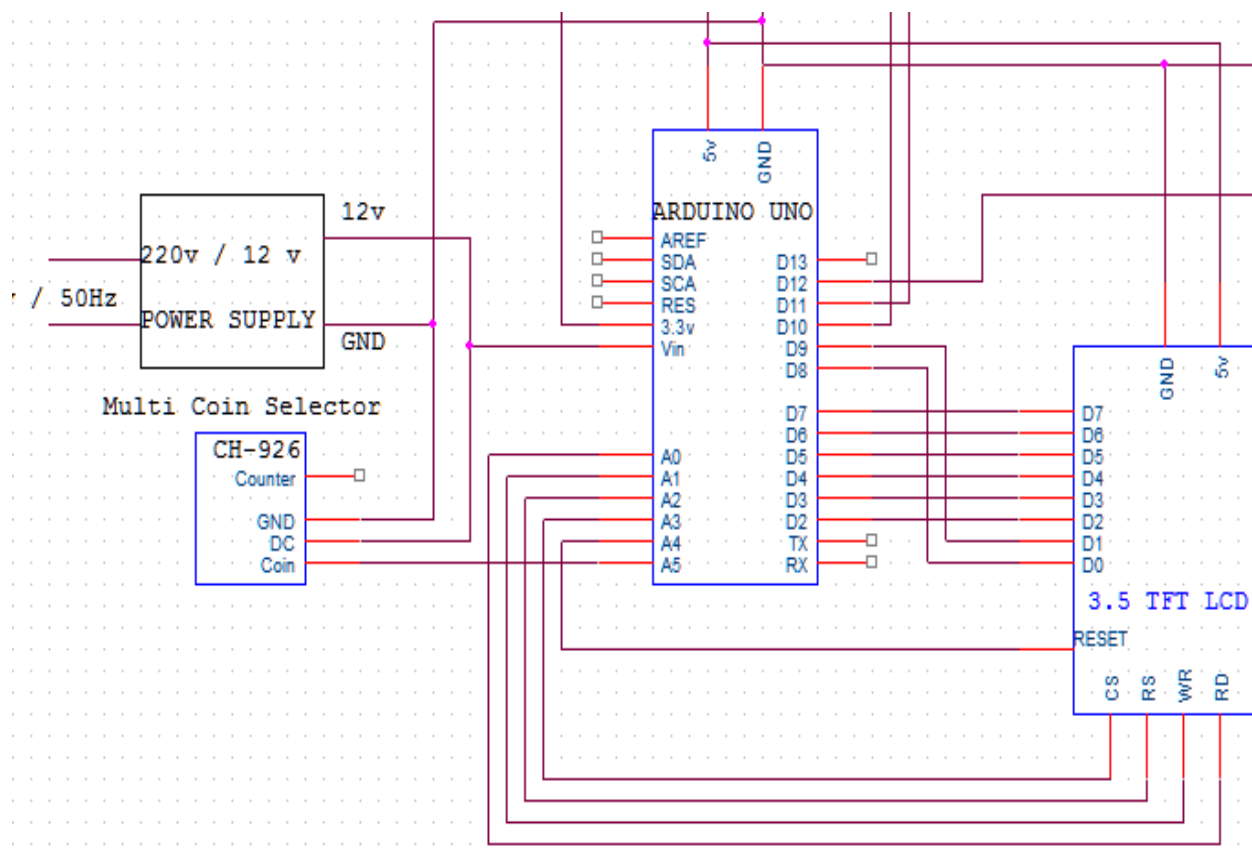
A5. המכונה צריכה מתח ספק של 12 וולט ולכן חיברנו אותה להדק ה-Vin של הארדואינו שמקבל

12 וולט ובכרטיס הארדואינו יש מייצב ל 5 וולט.

הסבר סכימה חשמלית

א. ארדואינו אונו

לארדואינו אונו יש 14 הדקים דיגיטאליים (D0-D13) המחוברים לרכיבים שמעבירים מידע דיגיטלי ביניהם. בנוסף להדקים הדיגיטאליים ישנם גם 6 הדקים אנאלוגיים (A0-A5) שמשמשים לכניסות אל ממיר מאנאלוגי לדיגיטאלי. ניתן להשתמש בהן גם כהדקים דיגיטאליים.



איור 9 : חיבור כרטיס הארדואינו אל תצוגת ה TFT

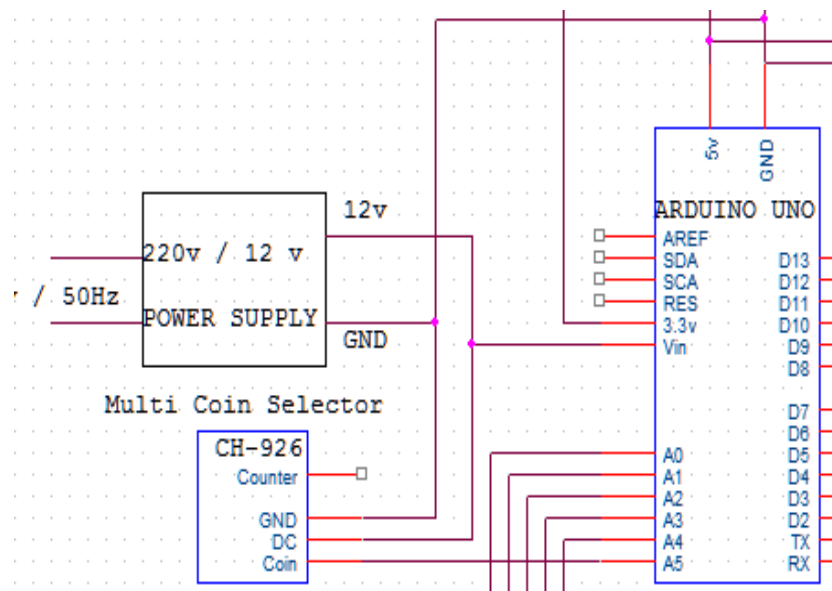
כמו שניתן לראות באיור, לבקר הארדואינו יש הדקי אדמה והדקי מתח שיהיו להזנה לכל הרכיבים המחוברים. בשרטוט רואים את שהתצוגה מחוברת אל ה 5 וולט והאדמה של הארדואינו. כמו כן רואים את ההדקים של הארדואינו המחוברים אל הדקי הנתונים D0 עד D7 של התצוגה ואת רגלי ההדקים A0 עד A5 של הארדואינו המתחברים אל רגלי הבקרה המתאימים בתצוגה.

ב.תצוגת TFT LCD

ניעזר באיור של חיבור הארדואינו והתצוגה שדיברנו עליו בסעיף הקודם.
לתצוגת ה TFT יש 8 הדקים דיגיטליים (D0-D7) המתחברים בהתאמה להדקים הדיגיטליים של הארדואינו (D2-D9) – כדאי לשים לב שהדקים D0 ו D1 של התצוגה מחוברים בהתאמה ל D8 ו D9. בנוסף לכך, לתצוגה יש 5 הדקים אנלוגיים (RESET,CS,RS,WR,RD) המתחברים להדקים האנלוגיים של הארדואינו (A0-A4).
הצג עובד עם מתח הזנה של 5 וולט ולכן הוא מחובר להדק ה - 5 וולט של הארדואינו.

ג. מכונת מטבעות

למכונת המטבעות יש שלושה הדקים: אדמה, מתח הזנה של 12 וולט והדק A5 (הדק אנלוגי שמשמש אצלנו כהדק דיגיטלי), הנותן מידע על סוג המטבע שנכנס. את ההדק מחברים להדק A5 שבארדואינו ואת האדמה מחברים לאדמה של הארדואינו.
מאחר ומכונת המטבעות דורשת מתח של 12 וולט יש לחבר את הדק המתח להדק ה-Vin של הארדואינו, על מנת שהארדואינו יוכל לספק מתח כה גבוה הוא מחובר לספק של 12 וולט המותאם לכניסת הבקרה.



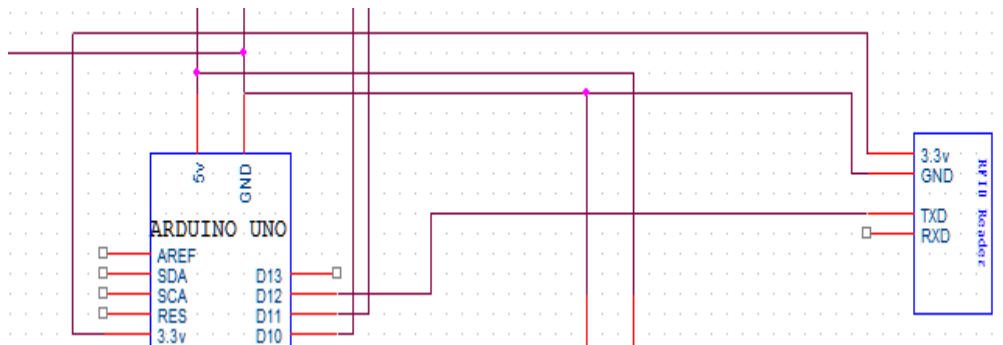
איור 10 : חיבור מכונת קבלת המטבעות אל הארדואינו

את הכרטיס מכיילים לפני ההפעלה בעזרת הוראות יצרן. אצלנו כיוונו את המערכת להוצאת פולס אחד עבור הכנסת מטבע של 1 שקל, 2 פולסים להכנסת מטבע של 2 שקלים ו 5 פולסים בהכנסת מטבע של 5 שקלים.

ד. RFID

לרכיב ה-RFID 4 הדקים: 2 למתח ואדמה ו-2 לתקשורת הטורית. על מנת ליצור תקשורת טורית עם הארדואינו יש לחבר את הדקי התקשורת להדקים הדיגיטליים של הארדואינו, מאחר ובפרויקט שלנו אין צורך בשליחת מידע ל RFID אלא רק קבלת מידע ממנו אנו נשתמש רק ברגל ה-TXD שתחובר להדק D12 בארדואינו. התקשורת לכרטיס היא תקשורת טורית רגילה ואנחנו נשתמש בתכנת ה softwareSerial שמאפשרת שידור/קליטה טורית לא בעזרת הדקי TXD RXD של הארדואינו. מידע בנושא תוכנת softwareSerial נמצא באתר :

<http://www.arikporat.com/arduino1.htm>

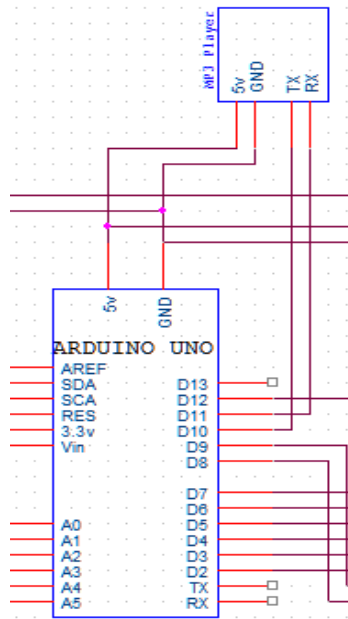


איור 11 : חיבור קורא RFID אל ארדואינו

כרטיס הקורא עובד עם מתח של 3.3V ולכן חיברנו אותו להדק ה 3.3 וולט של הארדואינו.

ה. Mp3 Player

לנגן המוזיקה 4 הדקים: 2 למתח ואדמה ו-2 לתקשורת טורית עם הארדואינו. בפרויקט שלנו נצטרך גם לשלוח מידע לנגן וגם ולקרוא ממנו, לכן יש צורך בהצלבה מלאה בין רגלי השידור והקריאה. על מנת לעשות זאת מחברים את הרגליים להדקים הדיגיטליים המאפשרים תקשורת טורית שהם: D10-D11 (ישנם הדקים שלא ניתן להפעיל אותם עם תוכנת ה softwareSerial).



איור 12 : חיבור נגן MP3 אל הארדואינו.

הכרטיס מקבל את ה 5 וולט והאדמה של האדואינו.

את קבצי הקול המושמעים ניתן לקבל מ Disk On Key או בעזרת כרטיס SD . אצלנו קבצי הקול נמצאים על דיסק און קי . בעזרת תוכנה קבענו שהקבצים הרצויים יילקחו מהדיסק און קי ולא מה SD .

מבוא לתוכנה

התוכנה נכתבה בשפת ++C, מאחר שזוהי השפה שבה הארדואינו עובד. סביבת הפיתוח הייתה תוכנה של הארדואינו, היודעת לתקשר בין המחשב לבין כרטיס הארדואינו באמצעות USB. כמו כן, סביבת הפיתוח מאפשרת את כתיבת התכנית עם אפשרויות העתקה הדבקה גזירה וכו', מבצעת קומפילציה של התוכנית, שמירה של התוכנית והעלאה של התוכנית אל כרטיס הארדואינו.

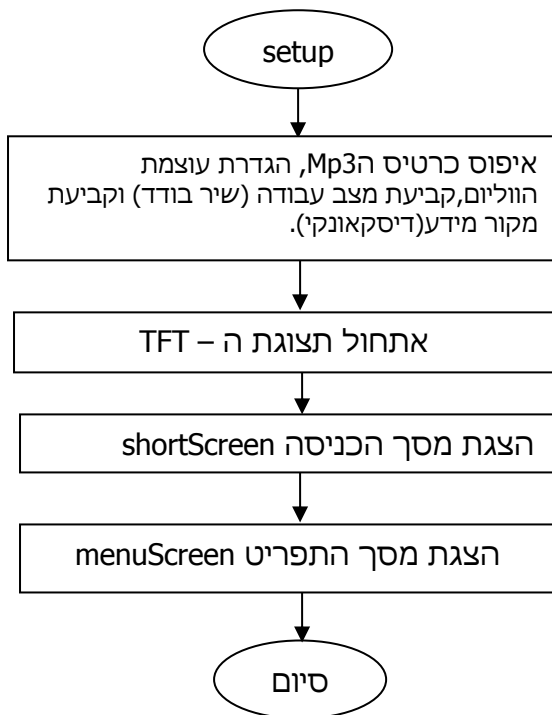
בעזרת כבל ה-USB המתחבר בין המחשב וכרטיס הארדואינו, הרצנו את התוכנית ובדקנו האם הוא מבצע את הדבר הרצוי. במידה ולא שינינו את התוכנית, שוב תרגמנו וצרבנו וכך עד שהתוכנית עבדה לשביעות רצוננו.

היות ויש לנו מספר מודולים של חומרה שונים, פיתוח התוכנית התבצע בצורה הבאה: קריאה לפונקציה כלשהי המטפלת באחד המודולים ואז כתיבת הפונקציה עצמה. לאחר מכן בוצעה הרצה של התוכנית ובדקנו האם הפונקציה מבצעת את מה שאנחנו רוצים ובמקרה הצורך, שינינו / הוספנו עד שהפונקציה עבדה כראוי. שוב קראנו לפונקציה נוספת בתוכנית הראשית (המתאימה למודול חומרה נוסף) וכתבנו את הפונקציה ובדקנו. כך עשינו עם כל הפונקציות הנוספות.

תוכנת הארדואינו ניתנת להורדה באתר של החברה בכתובת www.arduino.com. התוכנה היא בחינם והיא OPEN SOURCE, כלומר ניתן להשתמש בה לפי כללים בינלאומיים הניתנים במסגרת חוזה הנקרא GNU.

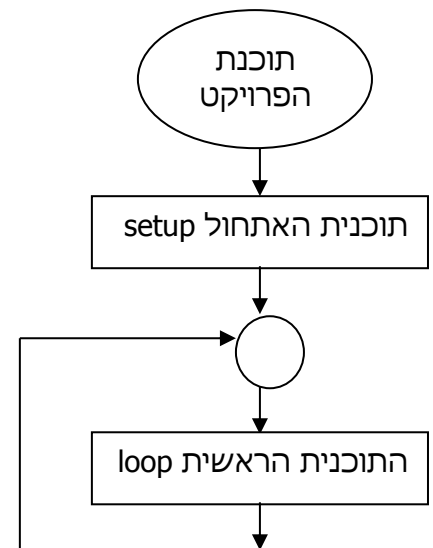
א. תוכנת הארדואינו ופונקציית האתחול :

א.2) פונקציית האתחול:



תרשים זרימה מספר 2 :
פונקציית האתחול.

א.1) תוכנת הארדואינו :



תרשים זרימה מספר 1 :
תוכנת הארדואינו.

הסבר תרשים זרימה

כפי שניתן לראות בתרשים הזרימה הימני ישנן בארדואינו 2 פונקציות:

(א1)

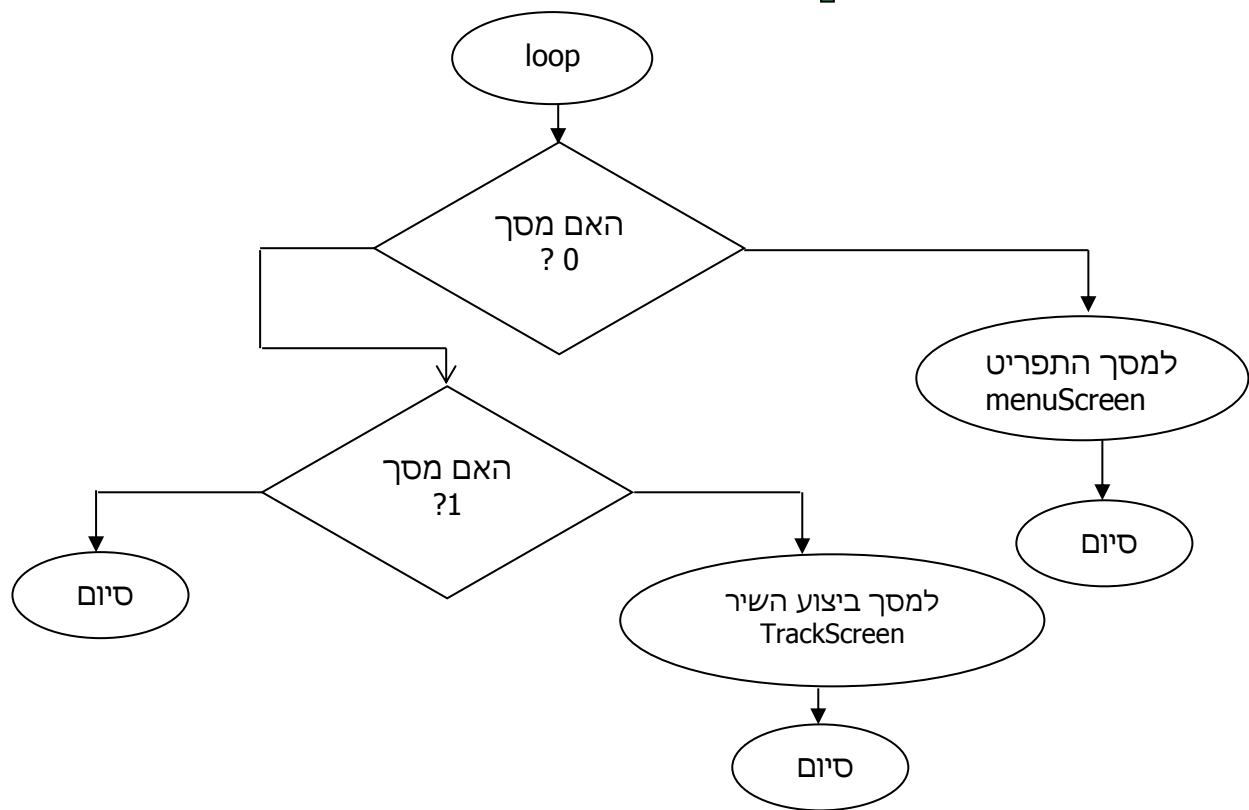
פונקציית setup - פונקציית האתחול מתבצעת פעם אחת בלבד בהפעלת המערכת.

פונקציית ה loop - מתבצעת כלולאה אין סופית.

(א2) בפונקציית האתחול מתבצעים הפעולות הבאות:

1. תחילה מאפסים את כרטיס הMp3 ולאחר מכן מגדירים את עוצמת הווליום ואיך הוא יעבוד: במבחינת מצב עבודה (ינגן שיר ויעצור) ומבחינת חיבור של מקור המידע (דיסקאונקי).
2. מאתחלים את תצוגת ה TFT
3. הצגה של מסך הכניסה ל3 שניות שבו כתוב את שם הפרויקט ואת המגישים.
4. הצגה של מסך התפריט שבו כתוב הוראות למשתמש ושלושה כפתורים עבור שלושה שירים.

ב. הפונקציה הראשית LOOP :

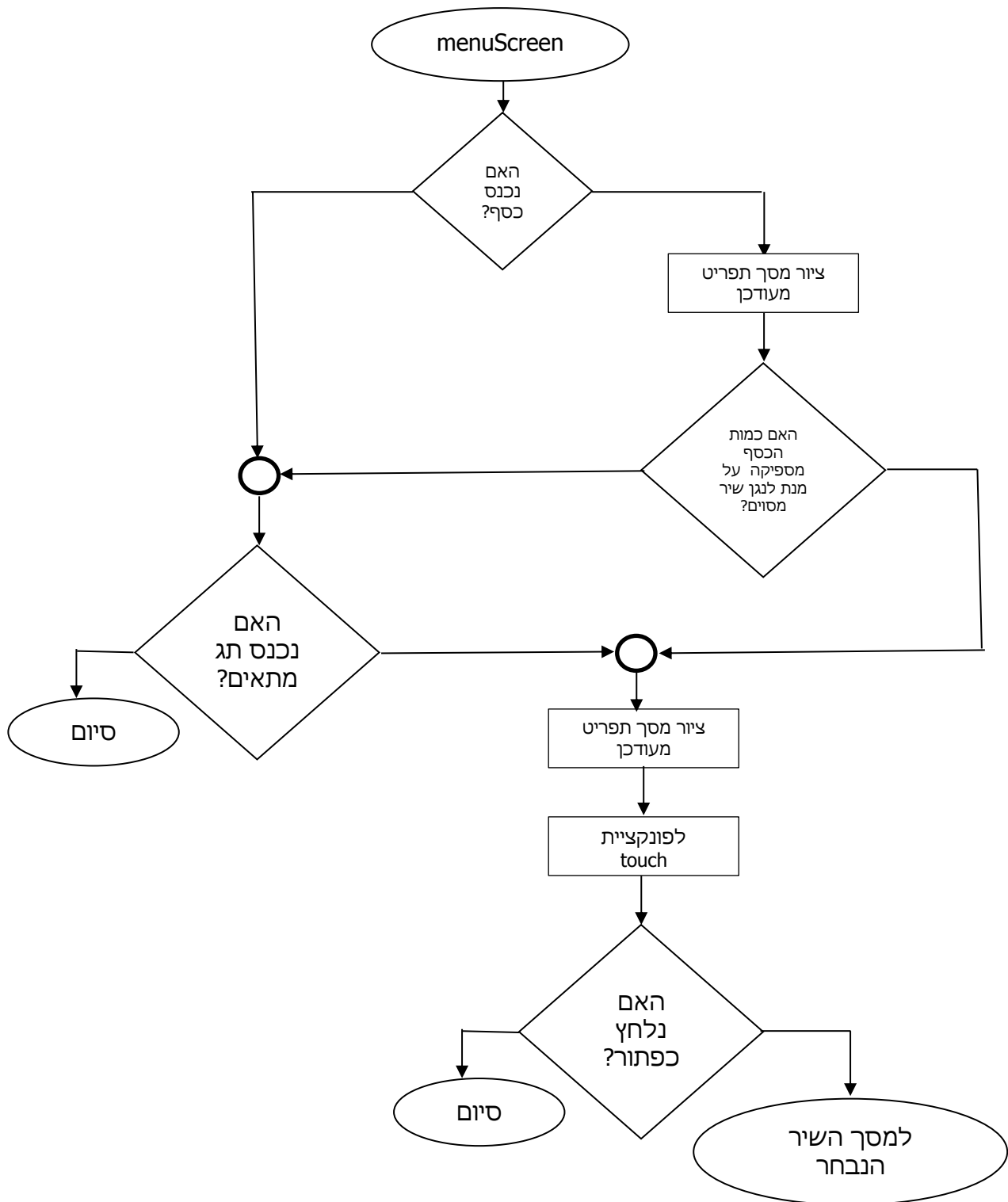


תרשים זרימה מספר 3 : הפונקציה הראשית LOOP.

הסבר תרשים זרימה

בפונקציית ה-loop המתבצעת בלולאה אין סופית בודקים מה הערך במשתנה screen. זהו משתנה המכיל 1 מתוך 2 ערכים: 0 ו-1. אם הערך הוא 0 עוברים למסך התפריט. אם הערך 1 עוברים למסך ביצוע השיר. בכל מסך אנחנו מציירים את הנתונים הרלוונטיים למסך המסוים וממתינים שהמשתמש יבצע פעולה מתאימה. אם המשתמש נמצא במסך התפריט יש באפשרותו להגיע למסך שיר מסוים שתלוי בכסף שהכניס והבחירה של המשתמש. כאשר המשתמש נמצא במסך השיר מתנגן השיר ברקע ויש למשתמש אפשרות לחכות עד שהשיר ייגמר ואז הוא יוחזר אוטומטית למסך התפריט או ללחוץ על הכפתור BACK שיחזיר אותו למסך התפריט ויפסיק את השיר.

ג. פונקציית מסך התפריט :



תרשים זרימה מספר 4 : פונקציית מסך התפריט.

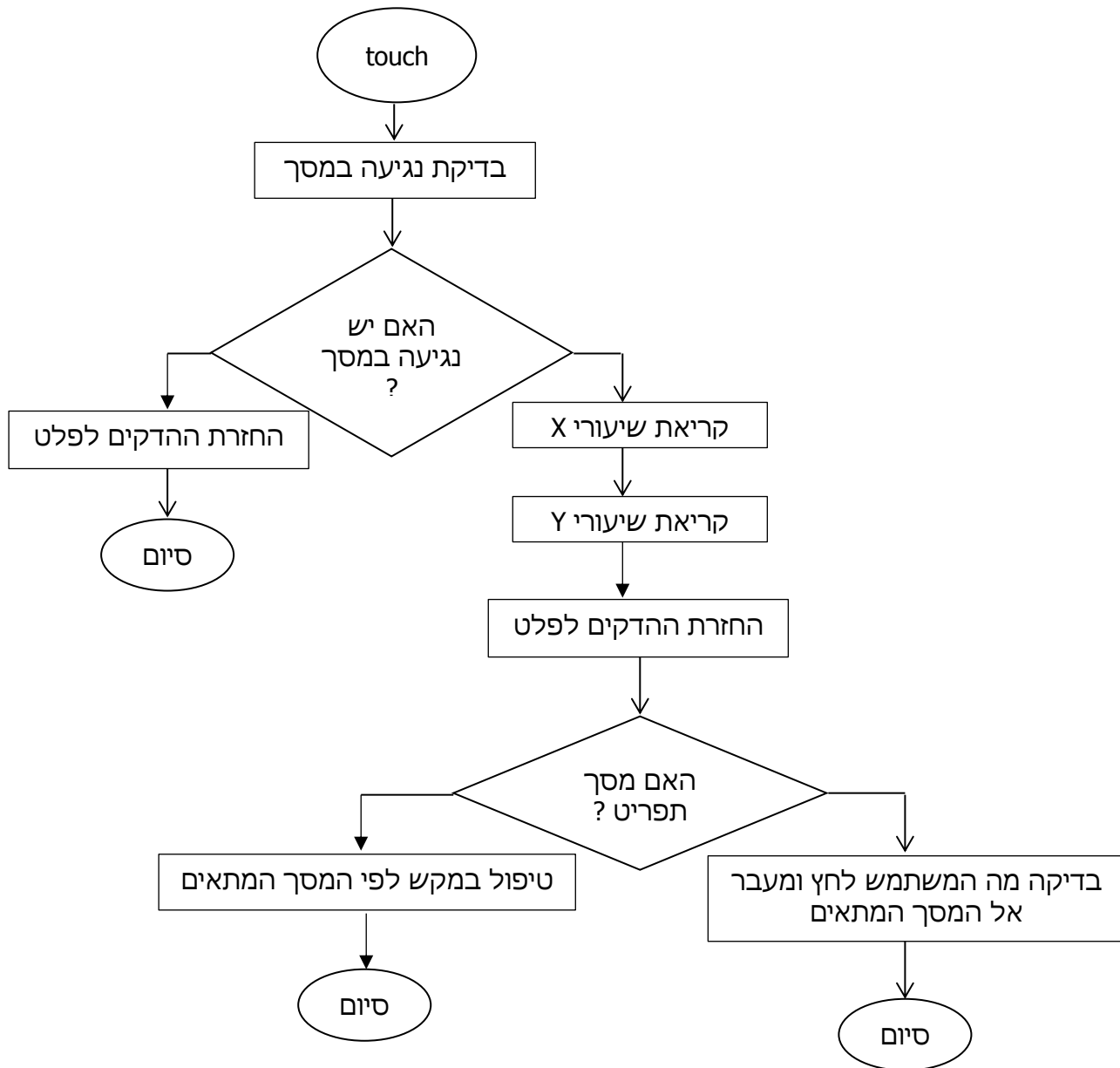
הסבר תרשים זרימה

במסך התפריט מופיע הסבר למשתמש איזה מטבעות המכונה יכולה לקלוט, את כמות הכסף שנכנס למכונה ושלושה כפתורים "נעולים" לכל השירים.

כאשר המשתמש מכניס מטבעות למכונה המסך מתעדכן ומציג מחדש את כמות המטבעות שנכנסו למכונה. ברגע שנכנס סכום כסף שעונה על דרישות הכפתורים המסך מתעדכן ומשנה את צבע הכפתור מאדום לירוק ועם שינוי זה מתאפשר מגע (אפשרות לעבור למסך הנתון) באותו מקום. במידה ורכיב ה-RFID קולט תג מתאים (שהוגדר מראש) המסך מתעדכן וכותב במקום של סכום המטבעות שבמכונה "free access", עם שינוי זה נפתחת גישה לכל הכפתורים.

לאחר שנלחץ אחד הכפתורים עוברים למסך השיר הנבחר.

ד. פונקציית ה- touch :

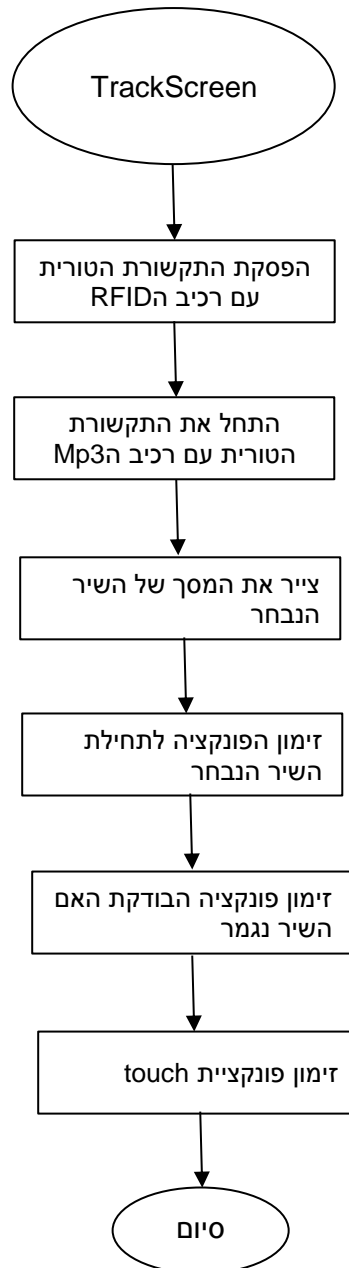


תרשים זרימה מספר 5 : פונקציית ה- touch.

הסבר תרשים זרימה

בודקים האם הייתה נגיעה במסך. את הבדיקה עושים בעזרת בדיקה של לחץ בציר Z . אם כן קוראים את שיעורי הנקודות X ו Y . מחזירים את הדקי המסך למצב פלט כי הדקי המגע משמשים גם כהדקי בקרה של התצוגה. בהתאם לנקודת הנגיעה מבצעים את הפעולה המתאימה.

ה. פונקציית ה-TrackScreen :

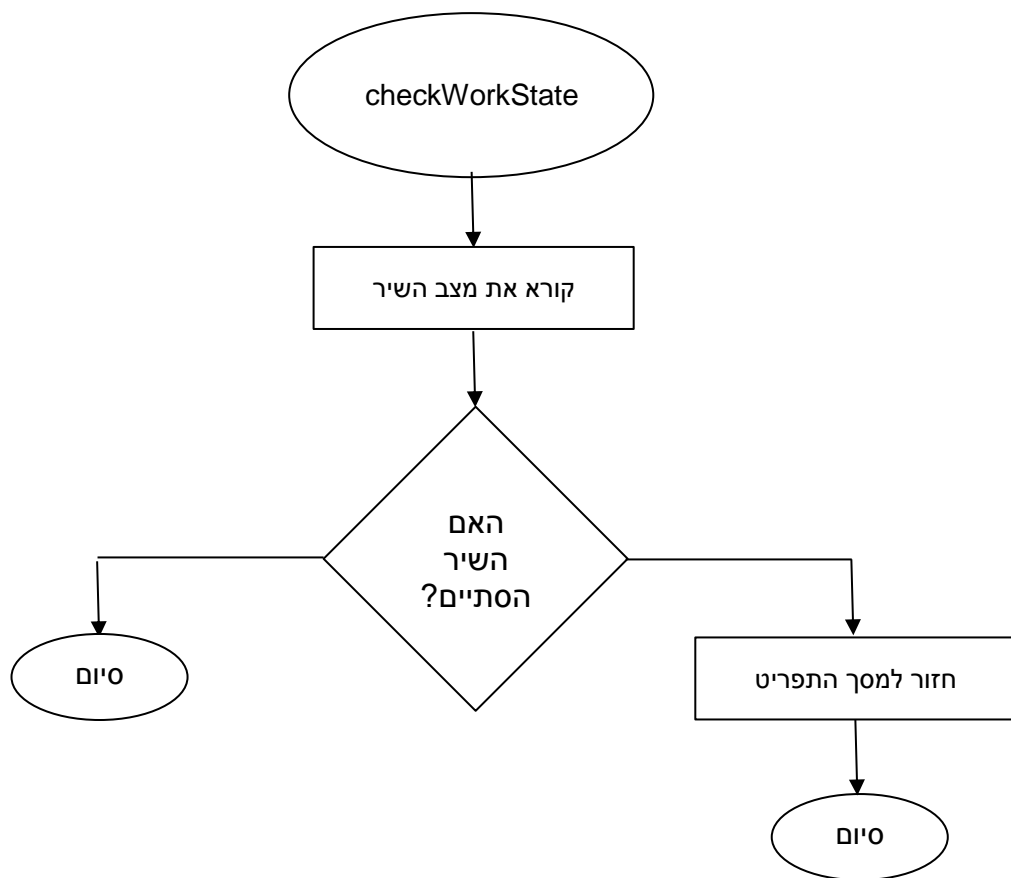


תרשים זרימה מספר 6 : פונקציית ה-TrackScreen.

הסבר תרשים זרימה

כשקוראים לפונקציה TrackScreen נפסקת התקשורת הטורית עם ה RFID ומתחילה התקשורת עם ה Mp3, זאת מאחר ולא ניתן ליצור תקשורת טורית עם שני רכיבים באותו הזמן. לאחר מכן מציירים מסך חדש של השיר הרצוי ומזמנים פונקציה לתחילת השיר שנבחר. לבסוף מזמנים פונקציה שבודקת מה מצב השיר (מתי הוא נגמר) ואת הפונקציה המאפשרת מגע כי קיים כפתור המאפשר חזרה למסך התפריט.

ו. פונקציית ה- `checkWorkState` :

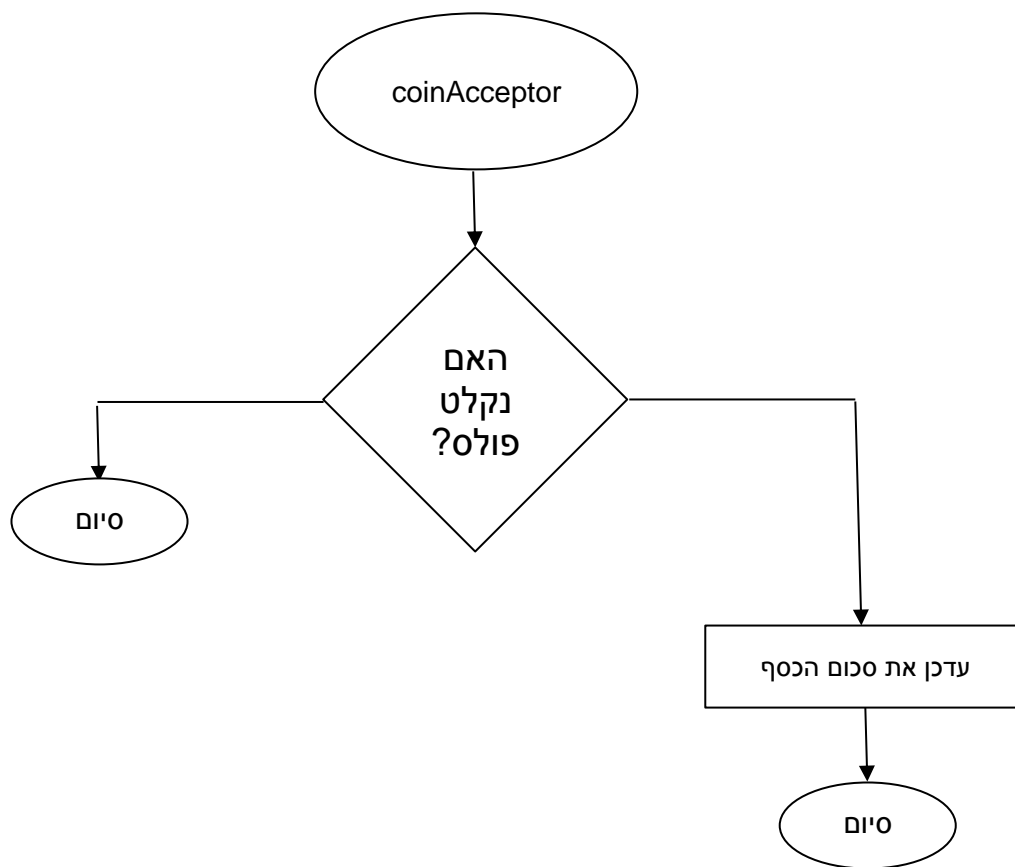


תרשים זרימה מספר 7 : פונקציית ה- `checkWorkState`.

הסבר תרשים זרימה

כאשר קוראים לפונקציה `checkWorkState` הפונקציה שואלת את רכיב ה-Mp3 מה מצב השיר. לאחר מכן הפונקציה שואלת האם השיר הסתיים (כאשר השיר מסתיים ה-Mp3 מחזיר נתון מסוים), במידה והשיר אכן הסתיים אז חוזרים למסך התפריט.

ז. פונקציית ה- coinAcceptor :

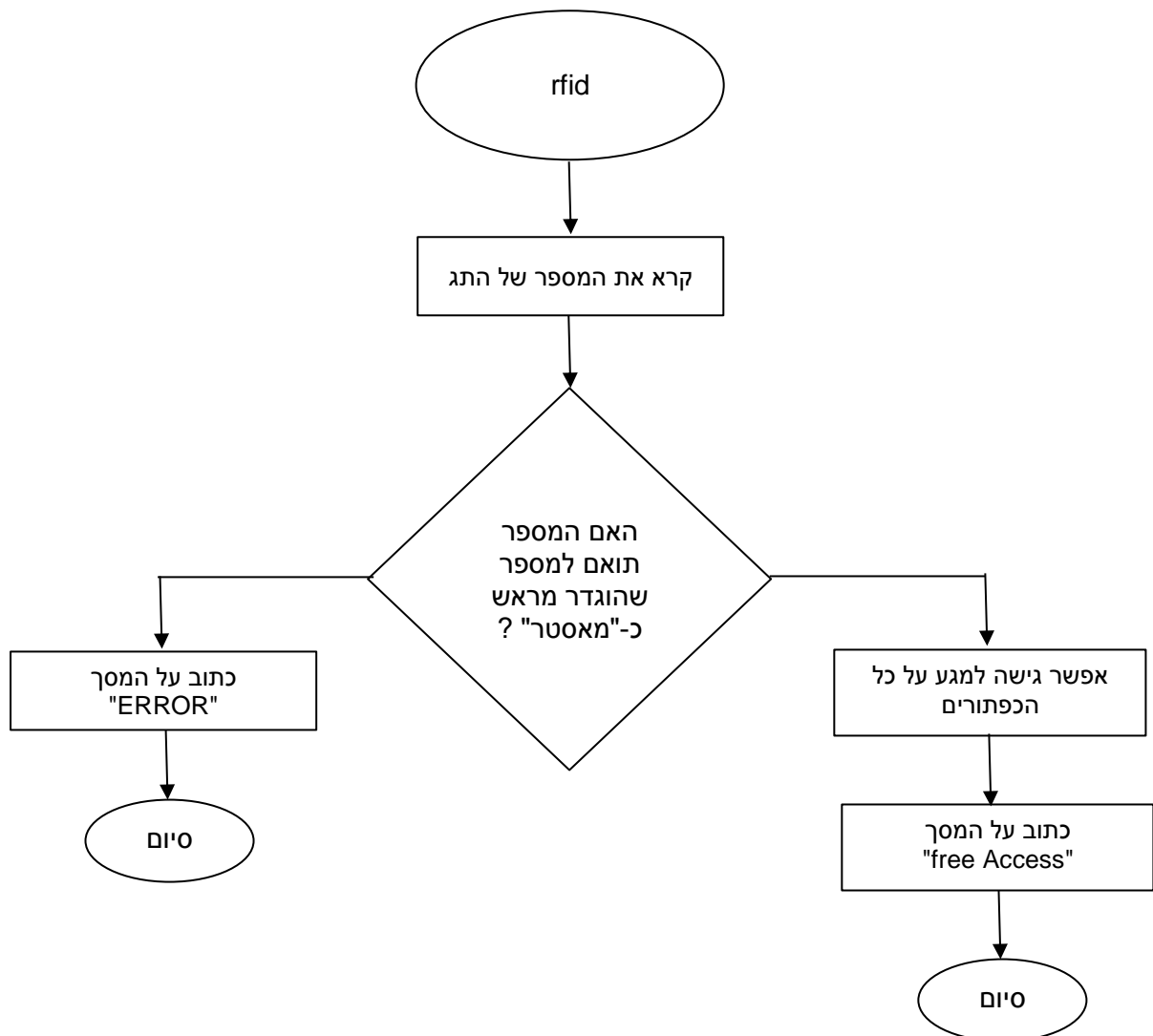


תרשים זרימה מספר 8 : פונקציית ה- coinAcceptor.

הסבר תרשים זרימה

כאשר קוראים לפונקציה היא בודקת האם הגיע פולס בהדק A5 שבארדואינו, במידה והגיע הפולס הפונקציה קוראת את הפולס כשקל אחד ומחשבת את סך הפולסים שנקראו במידה והגיעו עוד פולסים לאחר מכן (לדוגמא, 5 שקלים ישלחו 5 פולסים). לאחר שהפונקציה מחשבת את הסכום של הכסף שנכנס ניתן להדפיס את הסכום על המסך.

ח. פונקציית ה- rfid :



תרשים זרימה מספר 9 : פונקציית ה- rfid.

הסבר תרשים זרימה

כאשר מזמנים את הפונקציה rfid הפונקציה קוראת את המספר שעל התג שקרוב לאנטנה, הפונקציה משווה את המספר שנקלט למספר שהוגדר מראש כמספר "מאסטר" (המספר הנכון), במידה והמספר אינו תואם יופיע על המסך "ERROR" ובמידה והמספר תואם יופיע על המסך "free access" שמאפשר לבחור כל שיר ללא כסף (מאפשר מגע לכפתורים).

טבלת פונקציות

שם הפונקציה	מה הפונקציה מקבלת	מה הפונקציה מחזירה	תפקיד הפונקציה
1	loop		הפונקציה הראשית שמתבצעת בלולאה אין סופית.
2	setup		פונקציה המתבצעת פעם אחת בתחילת התכנית. כאן קובעים תפקידי הדקים (קלט או פלט) ואתחול רכיבי חומרה.
3	pinMode	מספר הדק והאם הוא קלט או פלט	פונקציה של הארדואינו שמקבלת את מספר ההדק והאם הוא קלט או פלט ומגדירה הדק זה בהתאמה.
4	digitalWrite	מספר הדק והאם להוציא לו 0 או 1	פונקציה של הארדואינו המוציאה להדק המסוים את הערך 0 או 1.
5	Serial.print	מספר הבתים שנכתבו למסך	פונקציה של הארדואינו שמדפיסה על המסך טקסט אסקי (ASCII).
6	Serial.println	מספר הבתים שנכתבו למסך	פונקציה של הארדואינו שמדפיסה נתונים ליציאה הטורית כטקסט ASCII.
7	Serial.begin		פונקציה של הארדואינו שמגדירה את קצב התקשורת בסיביות לשנייה.
8	delay	מספר מטיפוס long	פונקציה של הארדואינו המבצעת השעייה במילי שניות לפי המספר שנשלח אליה.
9	menuScreen		פונקציה המטפלת במסך התפריט, ובה ניתן לבחור בין המסכים השונים.
10	shortScreen		פונקציה המטפלת במסך הקצר שמופיע כשמפעילים את הפרויקט.
11	touch		פונקציה הבודקת האם הייתה נגיעה במסך.
12	tft.setTextSize		פונקציה הקובעת את גודל הטקסט שמודפס למסך.
13	tft.fillRect		פונקציה הצובעת את המסך שמודפס.
14	tft.setTextColor		פונקציה הקובעת את צבע הטקסט שמודפס למסך.

15	tft.drawRoundRect	מקבלת 6 ערכים: הנקודות x0, y0 של הפינה השמאלית העליונה, רוחב, גובה, רדיוס של הפינות המעוגלות, וצבע רצוי.	פונקציה המציירת מלבן עם קצוות מעוגלים שמודפס למסך.
16	command		פונקציה השולחת ערכים לכרטיס ה-Mp3 שאומרים לו איך להתנהג.
17	playPause		פונקציה המגדירה את הערכים הנדרשים להפעלה והפסקה של שיר.
18	stop		פונקציה המגדירה את הערכים הנדרשים להפסקה מוחלטת של השיר.
19	playTrack		פונקציה המגדירה את הערכים הנדרשים לקביעת מקור המידע (דיסקאונקי).
20	SetVolume		פונקציה המגדירה את הערכים הנדרשים לשינוי הווליום.
21	SetMode		פונקציה המגדירה את הערכים הנדרשים לבחירת דרך העבודה (ביצוע של השיר והפסקתו בסיומו).
22	checkWorkState		פונקציה המגדירה את הערכים הנדרשים לבדיקת מצב השיר. אם השיר מסתיים הפונקציה מחזירה למסך התפריט.
23	StartRFID		פונקציה המפעילה תקשורת טורית עם ה-RFID על-9600Hz.
24	StartMP3		פונקציה המפעילה תקשורת טורית עם ה-Mp3 על-9600Hz.
25	TrackScreen		פונקציה המעבירה למסך השיר הנבחר והפעלת השיר.
26	drawHebrewTav		פונקציה לציור תו בעברית.
27	drawHebrewString		פונקציה לציור מחרוזת של תווים בעברית.
28	coinAcceptor		פונקציה לבדיקה האם כסף נכנס ומה סכומו.
29	rfid		פונקציה לבדיקה האם התג שנקלט הוא התג המתאים.

C++ בתוכנה בשפת

```
#include <MCUFRIEND_kbv.h>
MCUFRIEND_kbv tft;

#define YP A1
#define YM 7
#define XM A2
#define XP 6
#define rxPin 12
#define txPin 13
#include <SoftwareSerial.h>
SoftwareSerial MP3(10,11); // Rx , Tx
SoftwareSerial RFID(12, 13); // RX, TX
-----*/COIN ACCEPTOR/*-----
const int coinSelector = A5; // הדק אנלוגי שבו נכנס האות מבורר המטבעות
const int signalCostFactor = 1; // כל פולס הוא שקל אחד ;

int signalValue = 0; // משתנה לשמירת הערך מבורר המטבעות
int state; // מצב נוכחי ;
int lastState = 0; // מצב אחרון ;
int balance = 0; // מצב החשבון ;
int coinValue = 0; // ערך המטבע הנוכחי ;
bool newCoinInserted; // מראה שהכניסו מטבע
int update = 1; // מראה שיש עדכון ;
long updateDebounceTime = 0; // הפעם האחרונה ששלחנו עדכון ;
long updateDebounceDelay = 500; // עדכון לאחר 500 מילי לאחר ניתור המגעות
-----*/MP3/*-----
unsigned char volume=15; // מראה את עצמת השמע ;
unsigned char mp3Buffer[10]; // חוצץ עבור נגן השמע
unsigned char i;
byte track; // מראה איזה שיר מתנגן
-----*/TFT/*-----
byte tav; // התו שנלחץ על ידי המשתמש
byte randomTav; // התו שקיבלנו על ידי הפונקציה random
```

```

long currentTime המערכת הפעלת מרגע עברו שנייה // ;
long oldTime הזמן הקודם // ;
int screen=0 מראה באיזה מסך נמצאים // ;
int x,y מסך, קריאת של ערכים ראשוניים // ;
unsigned char change מראה שיש לבצע החלפת מסכים // ;
bool first מראה שפעם ראשונה במסך // ;
bool debug=0;
-----*/RFID/*-----
byte masterRfid1[14]=
{0x2,0x31,0x45,0x30,0x30,0x42,0x35,0x36,0x30,0x46,0x30,0x33,0x42,0x3};
byte acceptid=0;
byte masterRfid2[14]=
{0x2,0x31,0x45,0x30,0x30,0x42,0x35,0x42,0x31,0x32,0x45,0x33,0x34,0x3};
byte rfidArray[14]
bool newRfid;

*****//
// נותנים שמות עם משמעות לצבעים
#define BLACK 0x0000
#define BLUE 0x001F
#define RED 0xF800
#define GREEN 0x07E0
#define CYAN 0x07FF
#define MAGENTA 0xF81F
#define YELLOW 0xFFE0
#define WHITE 0xFFFF

// הודעות שונות המוצגות בתצוגה
char str1;{"תיבת מוזיקה"}=[]
char str2;{"שקל"}=[]
char str4;{"הפרויקט של רועי ואריה"}=[]

// טבלת הגופנים בעברית
char hebrewTavs[28]=[8][

```

```

{0x11,0x11,0x9,0x0d,0x12,0x11,0x11,0},{0x1e,2,2,2,2,0x1f,0},{0x1f,1,1,3,9,0x11,0x
11,0},{0x1f,2,2,2,2,2,0}ד א ב ג // ,
{0x1f,1,1,1,0x11,0x11,0x11,0},
{0x0e,2,2,2,2,2,0},{0x1f,4,4,8,8,8,0},{0x1f,0x11,0x11,0x11,0x11,0x11,0x11,0} // ,
ח ז ח
{0x11,0x13,0x15,0x11,0x11,0x11,0x1f,0},{0x0e,2,2,2,0,0,0,0},{0x1f,1,1,1,1,1,1,1},{0x
1f,1,1,1,1,1,0x1f,0} ט י כ // ,
{0x10,0x1f,1,1,2,4,8,0},{0x1f,0x11,0x11,0x11,0x11,0x11,0x1f,0},{0x10,0x13,0x0d,0x9
,0x11,0x11,0x17,0},{0x0e,2,2,2,2,2,2,2} ל מ ן // ,
{3,1,1,1,1,1,0x1f,0},{0x1f,9,9,9,9,0x9,0x0f},{9,9,9,9,5,5,0x1f,0},{0x1f,0x11,0x11,0x0d,
1,1,1,0},{0x1f,9,9,0x0d,1,1,0x1f,0} נ ס ע פ ף // ,
{0x11,0x11,0x0a,4,2,1,0,0},{0x11,0x11,0x0a,4,2,1,0x1f,0},{0x1f,1,1,9,9,9,9,8},{0x1f,1,
1,1,1,1,1,0} ק ר צ ץ // ,
{0x15,0x15,0x15,0x15,0x15,0x15,0x1f,0},{0x0f,9,9,9,9,9,0x19,0}ש ת // ,
{0,0,0,0,0,0,0,0} // עבור התו רווח
};

```

*****//

```
void setup()
```

```
{
```

```
-----*/ Mp3/*-----
```

```
delay(1000); // המתנה לאיפוס של כרטיס ה mp3
```

```
setVolumeעוצמת השמע;()
```

```
setMode;()
```

```
/** קביעת מצב העבודה **/
```

```
mp3Buffer[0] = 0x7Eהתחלת הפקודה // ;
```

```
mp3Buffer[1] = 0x03התחלה של התווים + האורך ללא 2 התווים של התחלה // ;
```

```
(וסיום)
```

```
mp3Buffer[2] = 0xA9קביעת מצב עבודה // ;
```

```
mp3Buffer[3] = 0x00שיר בודד // ;
```

```
mp3Buffer[4] = 0x7Eסיום הפקודה // ;
```

```
Command;()
```

```

מספר (מיקום 3 ו 4) *UDISK/ **/ השמעה מ
mp3Buffer[0] = 0x7E // התחלת הפקודה ;
mp3Buffer[1] = 0x04 // אורך המחרוזת(כמות התווים +האורך ללא 2 התווים של התחלה ;
(וסיום
mp3Buffer[2] = 0xA2 // נגן שיר מ UDISK;
mp3Buffer[3] = 0x00 // החלק הגבוה של מספר השיר ;
mp3Buffer[4] = 0x00 // החלק הנמוך של מספר השיר ;
mp3Buffer[5] = 0x7E ;
Command;()

```

```

*****/TFT/*****
tft.reset;()
tft.begin(0x9481) // עבור המסך של 3.2 אינץ ;
tft.setRotation;(1)
tft.fillScreen(BLACK);
first=1;
shortScreen;()
delay;(3000)
menuScreen // מסך בחירה בין תצוגות שונות ;()
}

```

```

*****//
void loop()
{
switch (screen)
{
case 0:
menuScreen;()
break;
case 1:
TrackScreen;()
break;
}
}

```


/****** פונקציות של אם פי 3 *****/

//MP3 פונקציה המוציאה בצורה טורית את המחרוזת לכרטיס ה

void Command()

```
{
    unsigned i;
    for(i=0; i<mp3Buffer[1]+2; i++)
    {
        MP3.write(mp3Buffer[i]);
    }
}
```

void playPause()

```
{
    mp3Buffer[0] = 0x7E // התחלת הפקודה ;
    mp3Buffer[1] = 0x02 // אורך המחרוזת(כמות התווים +האורך ללא 2 התווים של התחלה ;
    (סיום)
    mp3Buffer[2] = 0xA3 // פקודת השמעה או עצור ;
    mp3Buffer[5] = 0x7E // סיום הפקודה ;
    Command // זימון פונקציה המוציאה בצורה טורית את המחרוזת לכרטיס ה MP3();
}
```

// הפסקה - עצירה של השיר

void Stop()

```
{
    mp3Buffer[0] = 0x7E // התחלת הפקודה ;
    mp3Buffer[1] = 0x02 // אורך המחרוזת(כמות התווים +האורך ללא 2 התווים של התחלה ;
    (סיום)
    mp3Buffer[2] = 0xA4 ;
    mp3Buffer[3] = 0x7E // סיום הפקודה ;
    Command // זימון פונקציה המוציאה בצורה טורית את המחרוזת לכרטיס ה MP3();
}
```

// הפעלה של שיר רצוי שמספרו במיקום 3 ו 4

void playTrack(unsigned char trackNumber)

```
{
```

```

mp3Buffer[0] = 0x7E // התחלת הפקודה ;
mp3Buffer[1] = 0x04 // אורך המחרוזת(כמות התווים +האורך ללא 2 התווים של התחלה ;
וסיום)
mp3Buffer[2] = 0xA2 // שימוש בדיסקאונקי ;
mp3Buffer[3] = 0x00 // הביית הגבוה של מספר השיר ;
mp3Buffer[4] = trackNumber // הביית הנמוך של מספר השיר ;
mp3Buffer[5] = 0x7E // סיום הפקודה ;
MP3(); // זימון פונקציה המוציאה בצורה טורית את המחרוזת לכרטיס ה Command
}

```

```

/* קביעת עצמת השמיעה */
void setVolume()
{
mp3Buffer[0] = 0x7E // התחלת הפקודה ;
mp3Buffer[1] = 0x03 // אורך המחרוזת(כמות התווים +האורך ללא 2 התווים של התחלה ;
וסיום)
mp3Buffer[2] = 0xA7 // פקודת קביעת עצמת שמע ;
mp3Buffer[3] = volume // רמת שמע 15 מתוך 31 אפשריות ;
mp3Buffer[4] = 0x7E // סיום הפקודה ;
MP3(); // זימון פונקציה המוציאה בצורה טורית את המחרוזת לכרטיס ה Command
}

```

```

// קביעת מצב העבודה : 00 לנגן שיר אחד בלבד
// 01 חזור על השיר 02 חזור על הכל 03 נגן אקראי
void setMode()
{
mp3Buffer[0] = 0x7E // התחלת הפקודה ;
mp3Buffer[1] = 0x03 // אורך המחרוזת(כמות התווים +האורך ללא 2 התווים של התחלה ;
וסיום)
mp3Buffer[2] = 0xA9 // קביעת מצב עבודה ;
mp3Buffer[3] = 0x00 // שיר בודד ;
mp3Buffer[4] = 0x7E // סיום הפקודה ;
MP3(); // זימון פונקציה המוציאה בצורה טורית את המחרוזת לכרטיס ה Command
}

```

```

    // בדיקת מצב העבודה
void checkWorkState()
{
    int i=0,myBuffer,[10]
    // for(i=0;i<10;i++)
    mp3Buffer[0] = 0x7E // התחלת הפקודה ;
    mp3Buffer[1] = 0x02 // אורך המחרוזת(כמות התווים +האורך ללא 2 התווים של התחלה ;
    (וסיום
    mp3Buffer[2] = 0xC2 // פקודת קריאת מצב עבודה ;
    mp3Buffer[3] = 0x7E // סיום הפקודה ;
    Command MP3 // זימון פונקציה המוציאה בצורה טורית את המחרוזת לכרטיס ה
    i=0;
    myBuffer[1]==0;
    while(MP3.available()>0)
        myBuffer[i++]=MP3.read();
    if(myBuffer[0]==0xc2 && myBuffer[1]==2)
    {
        screen=0;
        first=1;
    }

    for(i=0;i<10;i++)
    {
        Serial.print(myBuffer[i],HEX);
        Serial.print(" ")
    }
    Serial.println();

}

***** //RFID*****

void StartRFID(){
    Serial.begin(9600)
    Serial.println("Serial Ready");
}

```

```

    RFID.begin;(9600)
    Serial.println("RFID Ready");
}
void StartMP3(){
    Serial.begin;(9600)
    MP3.begin;(9600)
}
*****//

// מציג רועי ואריה במסך פתיחה
void shortScreen()
{
    if(first)
    {
        first=0;
        tft.fillScreen(BLACK);
        drawHebrewString(50,120,str4,GREEN,3);
        tft.setCursor;(60,200)
        tft.setTextSize;(5)
        tft.setTextColor(RED);
        tft.println("JukeBox v2.0");
        first=1;
    }
}

// מסך תפריט לבחירה על התצוגה הרצויה
void menuScreen ()
{
    if(first)
    {
        first=0;
        // עצירת השיר המתנגן
        Stop ;()
        MP3.end;()
        StartRFID;()
        // balance=0;

```

```

    acceptid=0;
// המסך הראשון - מסך המתכננים והמנחה
    tft.fillScreen(BLACK);
    tft.setCursor;(20 ,10)
    tft.setTextSize;(3)
    tft.setTextColor(CYAN);
    tft.print("insert a coin");
    tft.setCursor;(10,60)
    tft.setTextSize;(3)
    tft.setTextColor(CYAN);
    tft.println("works with:\n\n 1,2,5");

    tft.drawRoundRect(5,175,290,140,10,WHITE); //show how much money inserted
    tft.setCursor;(20,180)
    tft.setTextColor(MAGENTA);
    tft.setTextSize;(3)
    tft.print("coins inserted:");

    tft.setCursor;(110,230)
    tft.setTextColor(YELLOW);
    tft.setTextSize;(7)
    tft.print(balance);
----// מלבן עבור לחיצה על צורות ----
    if(balance<1)
        tft.drawRoundRect(340,28,135,70,10,RED);
    else
        tft.drawRoundRect(340,28,135,70,10,GREEN) ;
    tft.setCursor;(355,45)
    tft.setTextColor(WHITE);
    tft.setTextSize;(3)
    tft.print("Track1");
    tft.setTextSize;(2)
    tft.setCursor;(390,80)
    tft.setTextColor(YELLOW);
    tft.print("1")

```

```

drawHebrewString(350,80,str2,YELLOW,2)שקל// ;

if(balance<3)
    tft.drawRoundRect(340,128,135,70,10,RED);
else
    tft.drawRoundRect(340,128,135,70,10,GREEN);
tft.setCursor;(355,145)
tft.setTextColor(WHITE);
tft.setTextSize;(3)
tft.println("Track2");
tft.setTextSize;(2)
tft.setCursor;(390,180)
tft.setTextColor(YELLOW);
tft.print;("3")
drawHebrewString(350,180,str2,YELLOW,2)שקל// ;

if(balance<8)
    tft.drawRoundRect(340,228,135,70,10,RED);
else
    tft.drawRoundRect(340,228,135,70,10,GREEN);
tft.setCursor;(355,245)
tft.setTextColor(WHITE);
tft.setTextSize;(3)
tft.println("Track3");
tft.setTextSize;(2)
tft.setCursor;(390,280)
tft.setTextColor(YELLOW);
tft.print;("8")
drawHebrewString(350,280,str2,YELLOW,2)שקל// ;
}
coinAcceptor;()
if(newCoinInserted)
{
    newCoinInserted=0;
    tft.fillRect(15,230,260,50,BLACK);

```

```

    tft.setCursor;(110,230)
    tft.setTextColor(YELLOW);
    tft.setTextSize;(7)
    tft.print(balance);
}
    rfid // פונקציה לקריאה מה rfid ;()
    touch// ;()
}
-----// פונקציה לקליטת שעורי לחיצה במסך -----
void touch()
{
    int x,y;

    // קריאת שעורי איקס
    pinMode(XM,INPUT);
    pinMode(XP,INPUT);
    pinMode(YP,OUTPUT);
    pinMode(YM,OUTPUT);
    digitalWrite(YM,LOW);
    digitalWrite(YP,HIGH);
    x=analogRead(XM);

    // קריאת שעורי וואיי
    pinMode(XM,OUTPUT);
    pinMode(XP,OUTPUT);
    pinMode(YM,INPUT);
    pinMode(YP,INPUT);
    // digitalWrite(YM,0);
    digitalWrite(XM,0);
    digitalWrite(XP,1);
    y=analogRead(YP);

    pinMode(YM,OUTPUT);
    pinMode(YP,OUTPUT);
    if(debug)

```

```

{
Serial.print("x = ");
Serial.print(x);
Serial.print(" y = ");
Serial.println(y);

}

if(screen==0) מסך התפריט הראשי //
{
if(x>=190 && x <=310 && y>=90 && y<= 300 && (balance>=1 || acceptid==1)) //
לבחירת שיר מספר 1
{
screen=1 ;
first=1;
track=11 שיר מספר 11 // ;
if(acceptid==0)
balance=balance-1;
}
else if(x>=430 && x <=550 && y>= 100 && y<= 300 && (balance>=3 ||
acceptid==1))2 לשיר מספר 2//
{
screen=1 ;
first=1;
track=22 שיר מספר 22 // ;
if(acceptid==0)
balance=balance-3;
}
else if(x>=660 && x <=800 && y>95&& y<= 300 && (balance>=8|| acceptid==1))
3 לשיר מספר 3//
{
screen=1 ;
first=1;
track=33 שיר מספר 33 // ;
if(acceptid==0)
balance=balance-8;
}

```



```

}
}

else if(screen==1) שיר(1) במסך המנגן שיר(1)
{
    if(x>=760 && x<=850 && y>= 120 && y<= 270) חזור // האם לחצנו על חזור
    {
        screen=0;
        first=1;
    }

}

}
}

```

```

// פונקציה להשמעת השירים
void TrackScreen()
{
    int answerFromMp3;
    if(first)
    {
        first=0;
        RFID.end();
        StartMP3();
        tft.setTextSize(3);
        tft.fillScreen(BLACK);
        tft.setCursor(110,130)
        tft.setTextColor(YELLOW);
        tft.print("playing track ");
        tft.print(track) // ; לרשום את מספר השיר
        playTrack(track);
        tft.drawRoundRect(360,275,100,40,10,WHITE);
        tft.setCursor(285 ,380)
        tft.print("BACK");
    }
}

```

```

checkWorkState? בדיקת מצב עבודה - האם השיר הסתיים // ;()
touch;()
}

// פונקציה לציור תו. היא מקבלת תו בעברית ומציירת אותו במסך. היא מקבלת את ראשית
הקואורדינטות שהתו יתחיל
// נקודות איקס ו וואיי. היא מקבלת את התו בעברית את צבע התו ואת גודל הגופן
void drawHebrewTav(int x, int y, byte *tav, int color, int tavSize)
{
    int x0=x, y0=y, i, j, sizeX, sizeY;
    byte check;
    for (i=0; i<8; i++, tav++)
        for (sizeY=0; sizeY<tavSize; sizeY++, y0++)
            for (x0=x, check=16, j=0; j<5; x0=x0+tavSize, check=check>>1, j++)
                if (*tav & check)
                    for (sizeX=0; sizeX<tavSize; sizeX++)
                        tft.drawPixel(x0+sizeX, y0, color);
}

void drawHebrewString(int x, int y, byte *ptr, int color, int tavSize)
{
    char englishTav;
    byte *ptr1=ptr;
    while (*ptr++);
    ptr=ptr-2;
    for (int x0=x; ptr>ptr1; ptr--, x0=x0+tavSize*5+tavSize)
    {
        if (*ptr>0xb0)
            ptr--;
        if (*ptr > 0x80 )
            drawHebrewTav(x0, y, hebrewTavs[*ptr-0x90], color, tavSize);
        else if (*ptr==' ') // האם התו הוא רווח
            drawHebrewTav(x0, y, hebrewTavs[27], color, tavSize); // שיציג רווח
        else
        {
            englishTav=*ptr;

```

```

tft.setCursor(x0, y);
tft.setTextColor(color) ;
tft.setTextSize(tavSize);
tft.print(englishTav);
x0=x0+tavSize*5+tavSize;
}
}
}

```

***** פונקציה לטיפול בבורר המטבעות *****//

```

void coinAcceptor()
{
  signalValue = analogRead(coinSelector) // קרא ערך אנלוגי מכניסת בורר המטבעות ;
  if (signalValue > 1000)
  {
    newCoinInserted=1;
    state = 1; // State is 1 as we're high

  }
  else
  {
    state = 0;
    // Should we send a balance update
    if (update == 0)
    {
      if ((millis() - updateDebounceTime) > updateDebounceDelay)
      {
        Serial.print("Coin Value: ");
        Serial.println(coinValue); // WARNING: The coin value will be wrong if coins are
inserted within the updateDebounceDelay, adjust the delay and test
        Serial.print("Balance: ");
        Serial.println(balance); // This should be the most accurate as we should get
the same ammount of pulses even if multiple coins get inserted at once
        coinValue = 0; // Reset current coin value
        update = 1; // Make sure we don't run again, till next coin

```

```

        first=1לצייר שוב את המסך // ;
    }
}
}
if (state != lastState)
{
    // Process new signal
    if (state == 1)
    {
        Serial.println("state 1");
        balance = balance + signalCostFactor; // Update balance
        coinValue = coinValue + signalCostFactor; // Update coin value
        updateDebounceTime = millis(); // Update last time we processed a signal
        update = 0; // Time to send a update now?

    }
    else
    {
        Serial.println("updated");
    }
    lastState = state; // Update last state
}
delay;(1)
}

-----*/ RFID/*-----
void rfid()
{
    int i=0;
    while(RFID.available())
    {
        for (i=0; (i<14);i++)
        {
            Serial.println("reading");
        }
    }
}

```

```

    rfidArray[i]= RFID.read;()
    Serial.print(rfidArray[i],HEX);
}
    Serial.println;()
}
}
    if(i==14)? תווים 14 נקלטו //
{
    // השוואה בין קוד מסטר והקוד שנקלט
    for (i=0;i<14&&masterRfid1[i]==rfidArray[i];i++) ;
    if(i==14) הקוד הנקלט הוא מסטר ומאפשר גישה חופשית //
    {
        acceptid=1;
        tft.fillRect(15,230,260,50,BLACK);
        tft.setCursor;(15,240)
        tft.setTextColor(GREEN);
        tft.setTextSize;(4)
        tft.print("free access");
        tft.drawRoundRect(340,28,135,70,10,GREEN);
        tft.drawRoundRect(340,128,135,70,10,GREEN);
        tft.drawRoundRect(340,228,135,70,10,GREEN);
        RFID.flush;()
    }
    else זהו צ'יפ לא מוכר //
    {
        tft.fillRect(15,230,260,50,BLACK);
        tft.setCursor;(80,240)
        tft.setTextColor(RED);
        tft.setTextSize;(4)
        tft.print("ERROR");
        RFID.flush ;()
    }
}
}
}

```

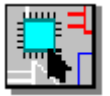
תכנון מכני

בנינו את הקופסא מאלומיניום והשתמשנו בטפט עץ:



איור 13: קופסת הפרויקט
מידות הקופסא:
רוחב: 27.5 ס"מ.
אורך: 35 ס"מ.

תוכנת OrCAD Lite 17.2 :



זוהי התוכנה שבה השתמשנו לשרטוט המעגל החשמלי של הפרויקט, את התוכנה למדנו במסגרת לימודי המכללה.

תוכנת Arduino 1.8.5 :



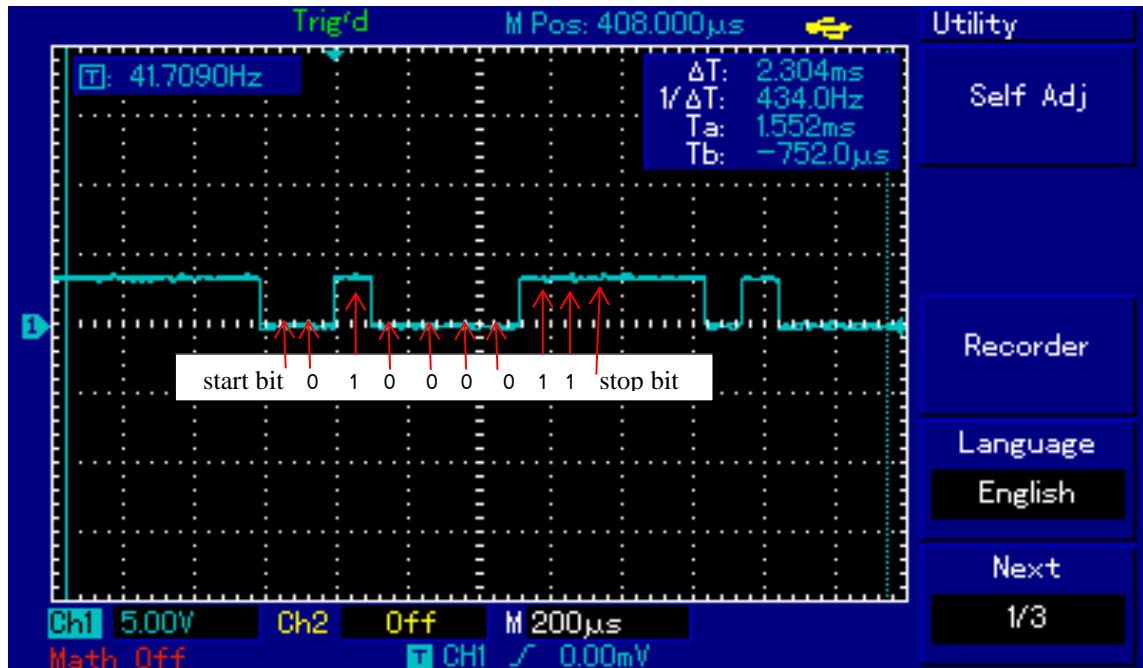
זוהי התוכנה שבה השתמשנו בפיתוח הפרויקט שלנו, את שימוש התוכנה למדנו במסגרת לימודי המכללה.

תוכנת Word 2013 :



את ספר הפרויקט כתבנו בעזרת מעבד תמלילים זה של חברת מיקרוסופט.

MP3 player RX



איור 14: מידות על הסקופ

מדדנו באמצעות סקופ את רגל ה RX של ה MP3 בזמן שניגנו שיר.
לפי דפי הנתונים של הרכיב, אמור לצאת מרגל ה RX הנתון C2H.

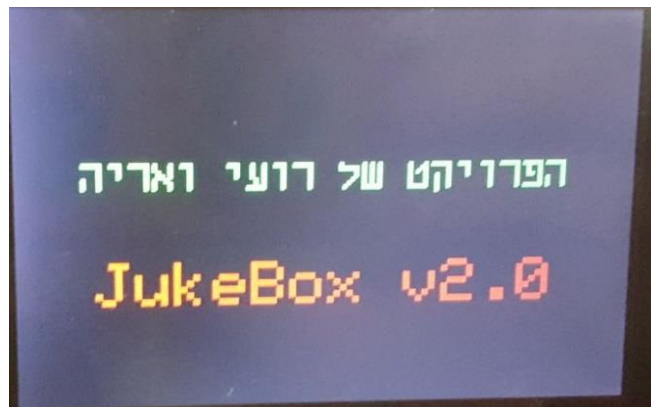
Check Work State			
START	LENGTH	COMMAND	END
0x7E	0x02	0xC2	0x7E

Return value: 0xC2

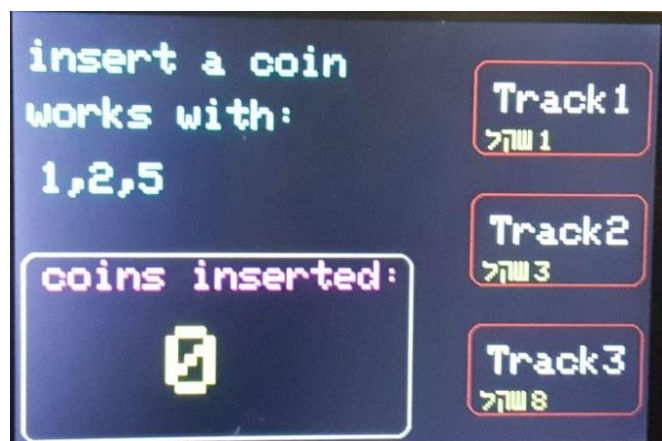
ניתן לראות שאכן המספר שמופיע על המסך הוא "11000010". הנתון נכנס מביט ה LSB אל ה .MSB.

הוראות הפעלה

כאשר מחברים את הערכה לחשמל, על תצוגת ה-LCD מופיע בראשונה מסך קצר :

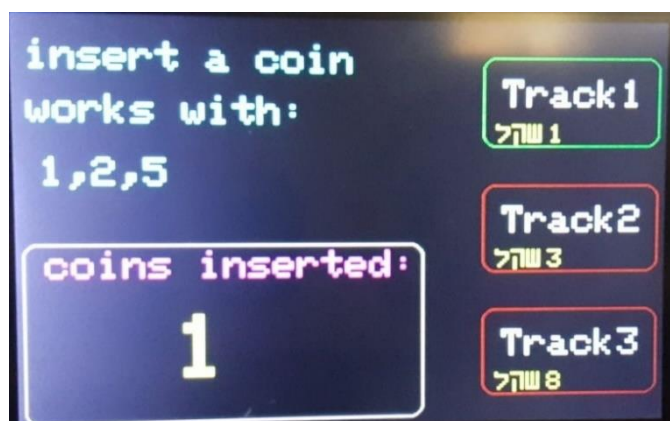


לאחר סיום הופעת המסך הקצר המשתמש מגיע למסך הבית:

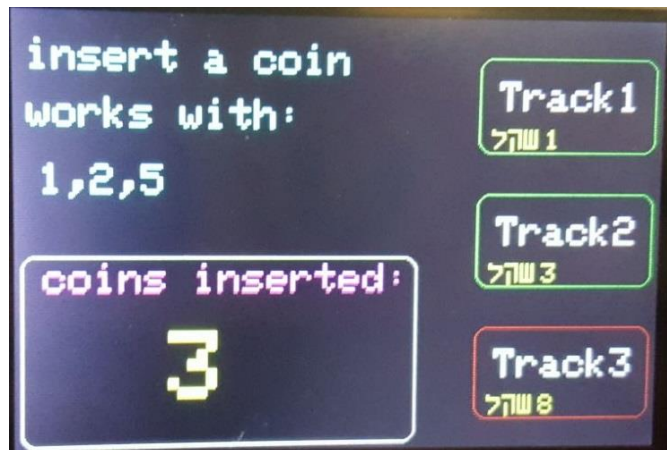


במסך הבית נמסר למשתמש שעליו להכניס כסף על מנת להפעיל שירים, לכן לאחר שהוכנס מספיק כסף נפתחות אפשרויות חדשות.

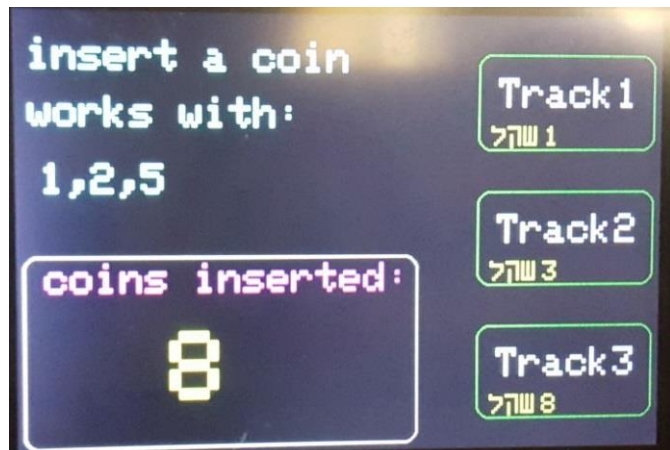
לאחר שהוכנס שקל אחד:



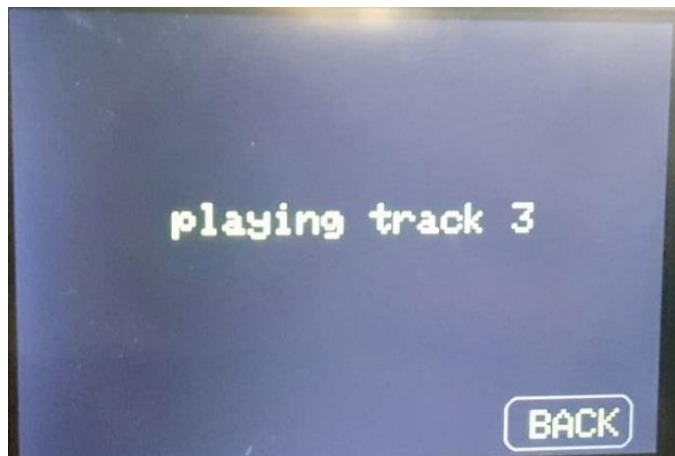
לאחר שהוכנסו עוד שני שקלים (סך הכול 3):



לאחר שהוכנסו עוד חמישה שקלים (סך הכול 8):

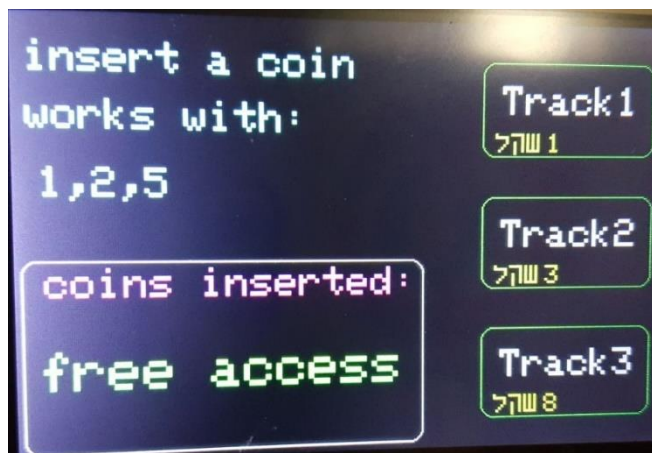


כעת לאחר שהוכנס מספיק כסף נניח שהמשתמש בוחר בשיר מספר 3 שעולה שמונה שקלים, המשתמש יועבר למסך השיר שברקע יתנגן השיר שבחר:

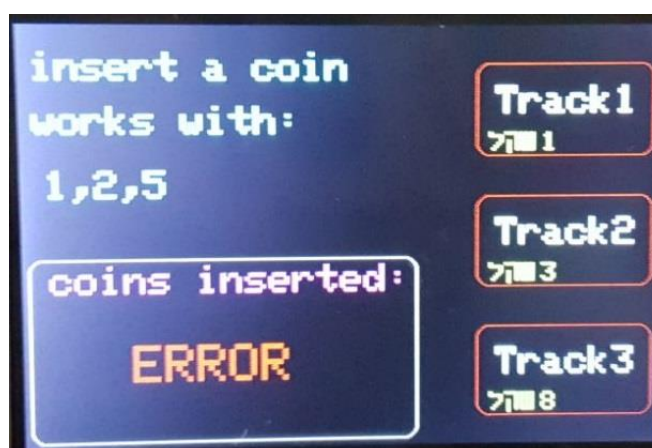


המשתמש יוכל לחזור אחורה למסך הבית ולהפסיק את השיר או לחכות שהשיר ייגמר.

במידה ולמשתמש יש תג RFID הוא יוכל להעביר אותו במכונה וכתוצאה מכך לקבל גישה חופשית לכל השירים בחינם ויופיע המסך הבא:



אך במידה ולמשתמש יש תג RFID שגוי תופיע הודעת ERROR שאומרת שהתג לא נכון:



יש לציין שהעברת תג לא נכון לא מבטלת את תג נכון/כסף שנכנס ולכן אם יש גישה לאחד הכפתורים העברת תג לא נכון לא תשנה כלום.

איתור ליקויים אפשרי

מאחר וזוהי הפעם הראשונה שאנו עובדים לבד על תוכנה בסדר גודל של פרויקט לשניים, היינו צריכים ללמוד לחשוב קדימה ולתכנן איך נבנה את התוכנה, למדנו איך לסדר את התוכנה במבחינה אסתטית ומבחינה לוגית (TOP DOWN) שתעבוד ולמדנו איך לבדוק בעיות בתוכנה ולתקן אותם (debug).

תקשורת סיראלית - מאחר ובפרויקט שלנו יש שני רכיבים שמשתמשים בתקשורת סיראלית (MP3,RFID) נתקלנו בבעיה שהם לא יכולים לפעול באותו הזמן, לכן היינו צריכים להפעיל ולסגור את התקשורת של הרכיבים בהתאם לזמן השימוש שלהם.

ציור על גבי המסך - על מנת לצייר על המסך במקומות הנכונים יש לבדוק את שיעורי ה-x,y של ארבעת הפינות של המסך וכך בעזרת ניסוי ותהייה למקם כל דבר במקומו.

MP3 - על מנת להשתמש בנגן כפי שאנחנו רוצים יש "ללמד" אותו כיצד לעבוד, ניתן למצוא באינטרנט הסברים וטבלאות שמכילות את המידע שיש למסור לרכיב על מנת לבצע כל פקודה, לדוגמא: שהנגן ידע שהוא לוקח את המידע מדיסקאונקי ולא מכרטיס SD.

מכונת המטבעות - למכונה יש אפשרות לתכנות עצמי כדי שיהיה אפשר להגדיר לה מהו מטבע נכון ואיך להגיב בהתאם, יחד עם מכונת המטבעות מגיע דף הוראות הפעלה שמסביר כיצד לעשות זאת.

סיכום ומסקנות

בסיום כיתה יג' נאמר שעלינו להתחיל לחשוב על רעיונות לפרויקט גמר. אנחנו כבר ידענו שאנחנו רוצים לעבוד יחד ולכן היינו צריכים לבחור פרויקט גדול שיספיק לשניים. כאשר הציעו לנו רעיונות לפרויקטים שכבר בוצעו בעבר לא אהבנו את הרעיונות משום שרצינו לעשות משהו משותף ששנינו אוהבים ומסכימים עליו ובנוסף גם רצינו לעשות משהו שיהיה יחודי לנו שנוכל להתחיל מאפס ולסלול את דרכנו לבד.

מאחר ושנינו אוהבים מוזיקה חשבנו על לבנות מכונת מוזיקה כמו ה-JukeBox שהיה קיים פעם, רק שבשונה מהמכונה הישנה החלטנו לשפר אותה עם טכנולוגיה עכשווית, אהבנו מאוד את הרעיון של מכונת המטבעות לכן הוספנו אותה לשאר הרכיבים כמו מסך מגע, MP3 ו-RFID. ידענו שעומדת בפנינו תוכנה ארוכה לכתוב לבד אך במקום לחשוש, דווקא ציפינו לכך משום שאנחנו אוהבים להתעסק עם תוכנה.

מאחר ואהבנו כל כך את הפרויקט ורצינו להתחיל אותו כבר, התחלנו אותו מוקדם בכך שקראנו הסברים על הרכיבים והזמנו אותם מראש.

ההזמנה המוקדמת של הרכיבים פעלה לטובתנו משום שחצי מהרכיבים שהזמנו הגיעו שבורים/פגומים:

מסך ה-TFT – הגיע שבור ולכן לא פעל והיינו צריכים להזמין עוד אחד.

ה-RFID שהזמנו הגיע לא עובד ולכן הזמנו עוד שניים ליתר בטחון.

הארדואינו שהשתמשנו בו קיבל מתח חיצוני של 12V וה 5 וולט שהוא מיצר עם המייצב שלו לא הצליח להפעיל את המודולים שמסביבו. מצאנו ארדואינו עם מייצב חזק יותר ללא צורך למייצב מתח חיצוני.

לבסוף הצלחנו לסיים את הפרויקט מוקדם בזכות ההכנות המוקדמות שלנו.

למדנו שעם תכנון נכון של זמנים ועבודה קשה ניתן לסיים פרויקטים גדולים.

הבחירה של הפרויקט נתנה לנו מוטיבציה להמשיך לעבוד כדי להגיע לתוצאה הסופית הנדרשת.

ביבליוגרפיה

האתר של ארדואינו: <https://www.arduino.cc>
האתר של אריה פורת: <http://www.arikporat.com>

דפי נתונים של MP3 :

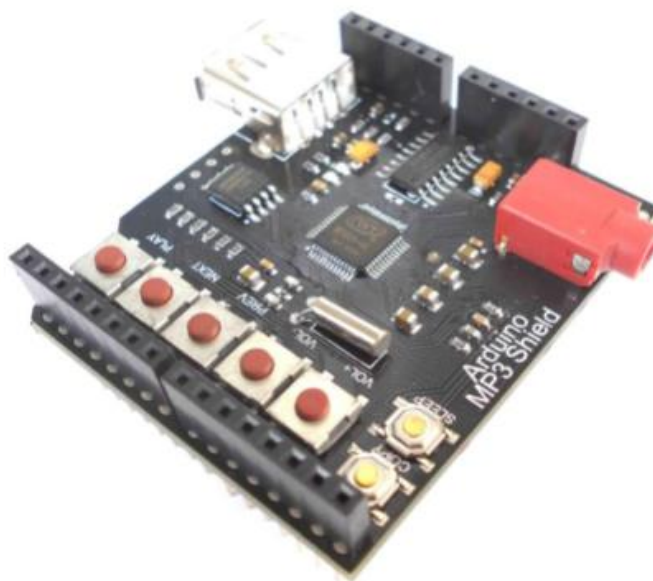
<https://www.elechouse.com/elechouse/images/product/Arduino%20MP3%20Shield/Arduino%20MP3%20Shield%20User%20Guide.pdf>

דפי נתונים coin acceptor:

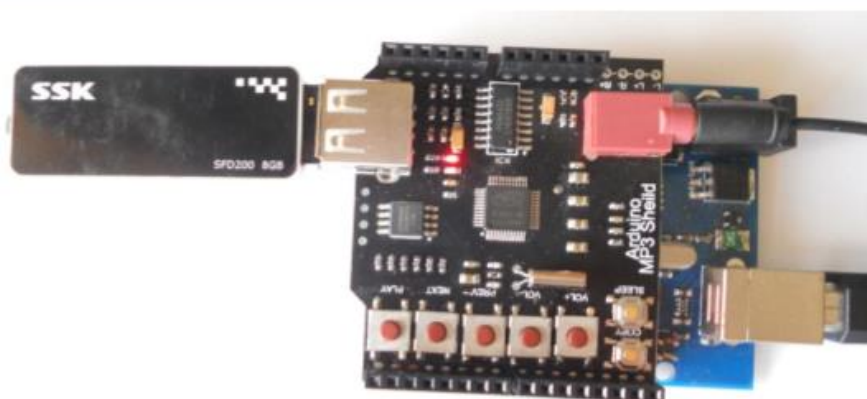
<http://blog.deconinck.info/post/2017/02/25/CH923/CH926/CH928-coin-acceptor-features-and-caveats>

כדי לחסוך בכמות הדפים נכניס כאן רק את דפי הנתונים של מודול ה MP3 ומכונת המטבעות. את שאר הרכיבים ניתן למצוא בקלות באינטרנט.

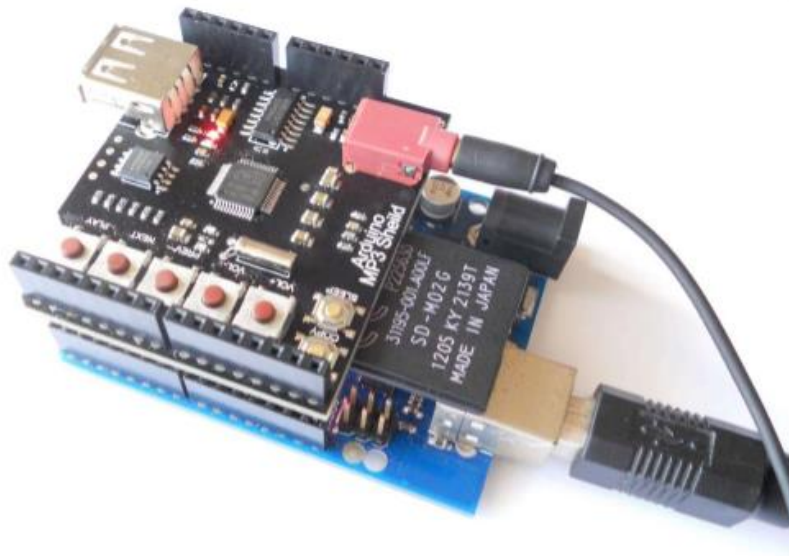
Arduino MP3 Shield User Guide



Instruction



Arduino MP3 Shield is a high quality MP3 module, supporting standard MP3 and WAV format audio files. It also supports three kinds of storage devices, including an onboard SPI-Flash, external SD Card and U-Disk.



The Arduino MP3 Shield has an onboard 3W Class-D stereo audio amplifier which could drive 4ohm load such as loudspeaker.

The Arduino MP3 Shield has seven buttons and a UART interface, that is to say user has two ways to control the module flexibly. The UART interface is connected to the Arduino hardware UART by default. Since most Arduino boards have only one UART interface, this module's UART interface can be redefined to Arduino D7 (TX) and D8 (RX) through the jumpers on board. With SoftSerial lib, Arduino could also send data to this module.

We tested with 8G SDHC card and 16G U-disk. It works in perfect compatibility. We haven't tested larger SD and U-disk yet.

Feature

- Plug in and play for Arduino
- Control by Serial UART interface or keys onboard.
- Directly drive 32ohm headset.
- On-board Audio Amplifier to drive 3W/4Ω external loudspeaker.
- 32 levels adjustable volume.
- Support FAT16/FAT32 file system.
- Support up to SD/SDHC card and U-Disk.
- On board 64MBit SPI-FLASH.
- Support copying audio files from SD card or U-Disk to SPI-FLASH.
- Supporting Sleep Mode.
- Support 16K~320Kbps MP3 files and 8K~44.1KHz sampling rate WAV files (SPI-FLASH only supports MP3 format)

Parameter

- Power Supply: 4.5V~5.5V, typically 5V.
- Interface: UART (5V TTL) and 7 buttons.
- Size: 47mm x 53mm.

Command Format

START	LENGTH	COMMAND	PARAMETER	END
0x7E				0x7E

NOTE: LENGTH value is the total count of areas including LENGTH, COMMAND and PARAMETER.

Writing Command

Writing command means commands which tell Arduino MP3 Shield to do something. After each command executed correctly, the command code will be returned.

Play SD Card File

Run this command to play selected SD card file.

START	LENGTH	COMMAND	TRACKS HIGH BYTE	TRACKS LOW BYTE	END
0x7E	0x04	0xA0	XX	XX	0x7E

TRACKS HIGH BYTE and TRACKS LOW BYTE are HEX numbers. They refer to the music file which you want to play. This is not the file name, but the file's order number. We have explained the file order above. For example, if you want to play the second music, TRACKS HIGH BYTE is 00, and TRACKS LOW BYTE is 02.

Play SPI FLASH File

Run this command to play selected SPI FLASH file.

START	LENGTH	COMMAND	TRACKS HIGH BYTE	TRACKS LOW BYTE	END
0x7E	0x04	0xA1	XX	XX	0x7E

TRACKS HIGH BYTE and TRACKS LOW BYTE are the same with 0xA0 command.

Play U-Disk File

Run this command to play selected U-Disk file.

START	LENGTH	COMMAND	TRACKS HIGH BYTE	TRACKS LOW BYTE	END
0x7E	0x04	0xA2	XX	XX	0x7E

TRACKS HIGH BYTE and TRACKS LOW BYTE are the same with 0xA0 command.

Pause OR Play

Run this command to switch between pause and play.

START	LENGTH	COMMAND	END
0x7E	0x02	0xA3	0x7E

Stop Playing

Run this command to stop playing.

START	LENGTH	COMMAND	END
0x7E	0x02	0xA4	0x7E

Next Music

Run this command to play next music.

START	LENGTH	COMMAND	END
0x7E	0x02	0xA5	0x7E

Previous Music

Run this command to play previous music.

START	LENGTH	COMMAND	END
0x7E	0x02	0xA6	0x7E

Set Volume

Run this command to set volume.

START	LENGTH	COMMAND	VOLUME LEVEL	END
0x7E	0x02	0xA7	0x00~0x1F	0x7E

The volume value is between 0x00~0x1f.

Set Play Mode

Run this command to set play mode.

START	LENGTH	COMMAND	PARAMETER	END
0x7E	0x02	0xA9	0x00 Single Play 0x01 Repeat single music 0x02 Repeat all 0x03 Play random	0x7E

Reading Command

Reading command means commands which check the state of Arduino MP3 Shield.

Check Volume

START	LENGTH	COMMAND	END
0x7E	0x02	0xC1	0x7E

Return:

COMMAND	VOLUME
0xC1	0x00~0x1F

Check Work State

START	LENGTH	COMMAND	END
0x7E	0x02	0xC2	0x7E

Return value:

COMMAND	RETURN VALUE
0xC2	0x01 Playing 0x02 Stop 0x03 Pause

Check Total Number of SPI FLASH Files

START	LENGTH	COMMAND	END
0x7E	0x02	0xC3	0x7E

Return value:

COMMAND	RETURN VALUE
0xC3	File numbers(2Bytes)

Check Total Number of SD Card Files

START	LENGTH	COMMAND	END
0x7E	0x02	0xC4	0x7E

Return value:

COMMAND	RETURN VALUE
0xC4	File numbers(2Bytes)

Check Total Number of U-Disk Files

START	LENGTH	COMMAND	END
0x7E	0x02	0xC5	0x7E

Return value:

COMMAND	RETURN VALUE
0xC5	File numbers(2Bytes)

Check Current Playing File

START	LENGTH	COMMAND	END
0x7E	0x02	0xC6	0x7E

Return value:

COMMAND	TRACKS HIGH BYTE	TRACKS LOW BYTE
0xC3	XX	XX

TRACKS HIGH BYTE and TRACKS LOW BYTE are the same with 0xA0 command. Please refer to [Command 0xA0](#)

Manual of CH-926

CH-926 is a multi coin selector, can accept up to 6 kinds of different coins at the same time. This type of coin selector is widely used in Vending machine, Arcade Game, Message chair, and other self-management system. CH-926 is mainly based on material, weight and size to identify coins. We use the most up to date algorithm to design software. Therefore, CH-926 is very stable and accurate even when environment changes such as temperature, and humidity etc... In order to increase the accuracy, we suggest different version of coins use different channel to set up.

Specifications

Coin diameter : 15mm~32mm

Coin thickness : 1.2mm~3.8mm

Working voltage : DC +12V \pm 10%

Working current : 65mA \pm 5%

Signal output : pulse

Atmospheric pressure : 86Kpa—106Kpa

Working humidity : \leq 95%

Speed : \leq 0.6s

Accuracy rate of identification : 99.5%

Features

- a. Capable of accepting all worldwide Coins and Tokens.
- b. Intelligent CPU software control, and high accuracy .
- c. Self-programming without PC.
- d. Accept 1~6 different kinds of coins at the same time.
- e. Free to set up pulses' output.
- f. Prevent not only electric shock but also electromagnetic interference.
- g. Automatic self-test for problems.

The Process of Setup for Parameters

1. Press the "Add" and "minus" buttons at the same time for about three seconds, then the letter "A" will appear from the LED display.
2. Press the "setup" button once, and the letter "E" will appear. Next, use the buttons to choose how many kinds of coins you would like to use; then press the "setup" button again to finish.
3. The letter "H" will appear after pressing the button. Use the "Add" and "minus" buttons to choose how many samples you would like to insert later. Next press the "setup" button again to finish.
4. The letter "P" will appear after pressing the button. Use the "Add" and "minus" buttons to choose the amount of output's signals/pulses you want. The quantity limited is 50 times. Next, press the "setup" button to finish.
5. The letter "F" will appear after pressing the button. Use the "Add" and "minus" buttons to choose accuracy. The value is from 1~30, and 1 is the most accurate. Normally, 5~10 will be fine. Next, press the "setup" button to finish.
6. So far, you have successfully set up the first coin. please repeat all above procedures until you have set up all the coins. The letter "A" will appear again after all above procedures are finished.
7. Press the "setup" button, and the letter "E" will appear. Finally, turn off and turn on the power. The setup will be stored.

You can start sampling after the setup is finished. Please choose at least 20 coins. The sampling process will affect the accuracy of coin selector.

Sampling

1. Press the "setup" button, then letter "A" will appear from the LED display.
2. Press the "setup" button again, then letter "A1" will appear. Next, start to insert sample coins. The LED display will show how many coins you insert. The letter "A1" will appear again after finished.
3. Press the "setup" button again, then the letter "A2" will appear. Next, Start to sample 2nd coin, and repeat No. 1 and No. 2 until all the coins are set up .
4. After finished the sampling, press the "setup" button. The letter "A" will appear, then turn off and turn on the power. Now you can start to use it.

Connections' manual

