

טביעת אצבע ומודול טביעת אצבע

1. עקרונות טביעת אצבע

זיהוי טביעת אצבע הוא אחד מאמצעי הזיהוי הביומטרי על פי תכונות מורפולוגיות הקשורות לצורת גופו של האדם, כמו זיהוי תווי פנים, זיהוי כף יד, זיהוי DNA וזיהוי קשתית או רשתית בעין. אמצעי נוסף הוא על פי ההתנהגות כמו זיהוי קצב הקלדתו של אדם, זיהוי אופן הליכתו של אדם וזיהוי קולו של אדם.

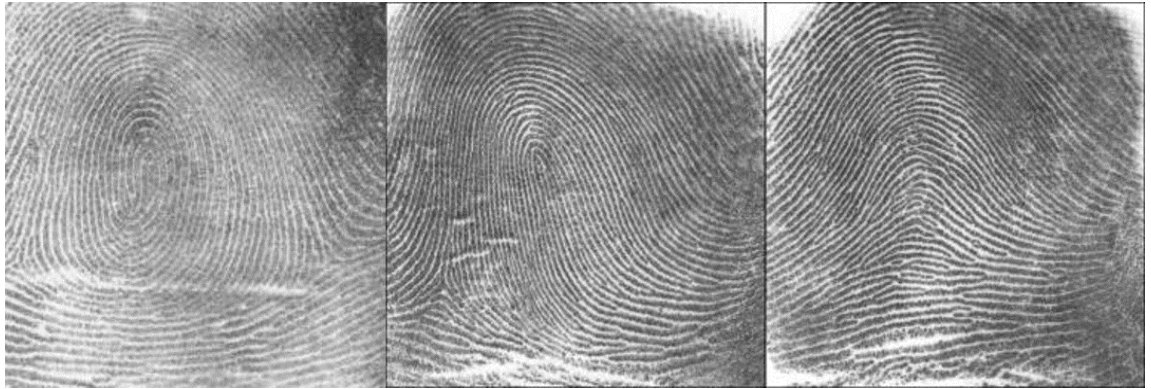
טביעות אצבע הן המקור האמין ביותר לזהות אישית. לא נמצאו שתי טביעות אצבעות זהות בין מיליוני השוואות. ההערכה היא כי הסיכוי של טביעת אצבע של אדם להיות זהה לאדם אחר היא 1:64 מיליארד. מכאן שאין שום סיכוי לטביעות אצבע זהות כי האוכלוסייה האנושית היא עדיין רק סביב 7 – 8 מיליארד. זיהוי טביעת אצבע משמשת היום בעסקאות רשמיות, תעסוקה, זיהוי פלילי, מחשבים, וסורק טביעות אצבע כ-ATM (בנקט), הגירה, שדות תעופה ועוד.

טביעת האצבעות הן הצורות הייחודיות שהאצבעות שלנו משאירות על חפצים שונים אחרי שהן נוגעות בהם. מה שגורם להטבעה הן הצורות המיוחדות של קפלי העור של אצבעות הידיים (וגם הרגליים) – שפועלים כמעין 'חותמת' ייחודית שמחתימה כל דבר שאנו נוגעים בו (ההחתמה נוצרת בעיקר מהזיעה). עד היום לא נמצאו שני אנשים בעלי אותה 'חותמת' כלומר עם אותן טביעות אצבעות בדיוק. אפילו לתאומים זהים – שהמטען הגנטי שלהם זהה לחלוטין יש טביעות אצבע שונות.

מדוע בכלל יש קפלי עור זעירים על האצבעות? מדוע הם שונים בין האנשים? ואיך בכלל הם נוצרים? ההשערות שהועלו הן שקפלי העור מהווים יתרון אבולוציוני בכך שהם מגדילים את כוח החיכוך בין היד לבין חפצים שונים, ובכך עוזרים לנו לאחוז בחפצים שונים בלי שהם יחליקו.

לדרכי היווצרותם של הקפלים שתי השערות: על פי השערה אחת הם נוצרים עקב התקפלות של עודפי העור בעת ההתפתחות העוברית. השערה אחרת אומרת שהם נוצרים עקב תבנית קצות העצבים שנוצרת בקצות האצבעות בעת ההתפתחות העוברית. בשני התהליכים יש רכיב אקראי (התקפלות או גדילת קצות עצבים) ומכאן התבנית האקראית הייחודית לכל אדם. מה שבטוח שהתבניות אכן נוצרות בזמן ההתפתחות העוברית, ובשבוע ה-16 להריון כבר יש לעובר תבניות ייחודיות של קפלי עור על כפות הרגליים והידיים, שילוו אותו בקביעות עד סוף חייו! טביעות האצבע שנטביע בגיל 10 זהות לאלו שנטביע בגיל 50 (פרט לגודל או צלקות שצוברים).

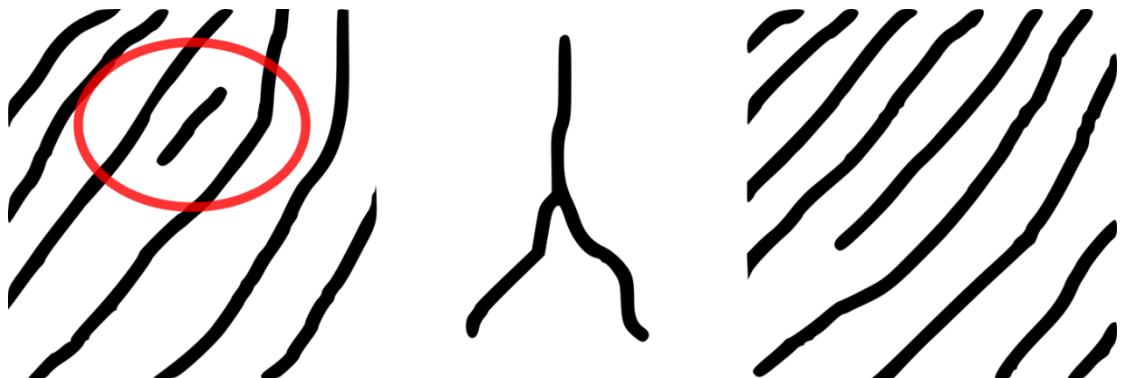
נהוג לסווג את התבניות של העור על האצבעות ל-3 משפחות עיקריות: קשת, לולאה ומערבולת. כפי שאפשר לראות באיור 1:



איור 1: מימין לשמאל: קשת, לולאה ומערבולת – שלוש התבניות העיקריות של טביעות אצבע

בתוך שלושה הסיווגים העיקריים האלו, מסווגים גם תת-סוגים. כאשר על כל אצבע יכולה להופיע התבנית הייחודית לה.

מה ששונה אצל כל אדם זה לא התבנית הגדולה, אלא פרטים כמו סיום של רכס על העור התפצלות של רכס, רכס קצרצר הנראה כנקודה ועוד. איור 2 מתאר את הנאמר:



איור 2: מימין לשמאל: סיום רכס, התפצלות רכס ורכס קצרצר – הפרטים הייחודיים בטביעת אצבע

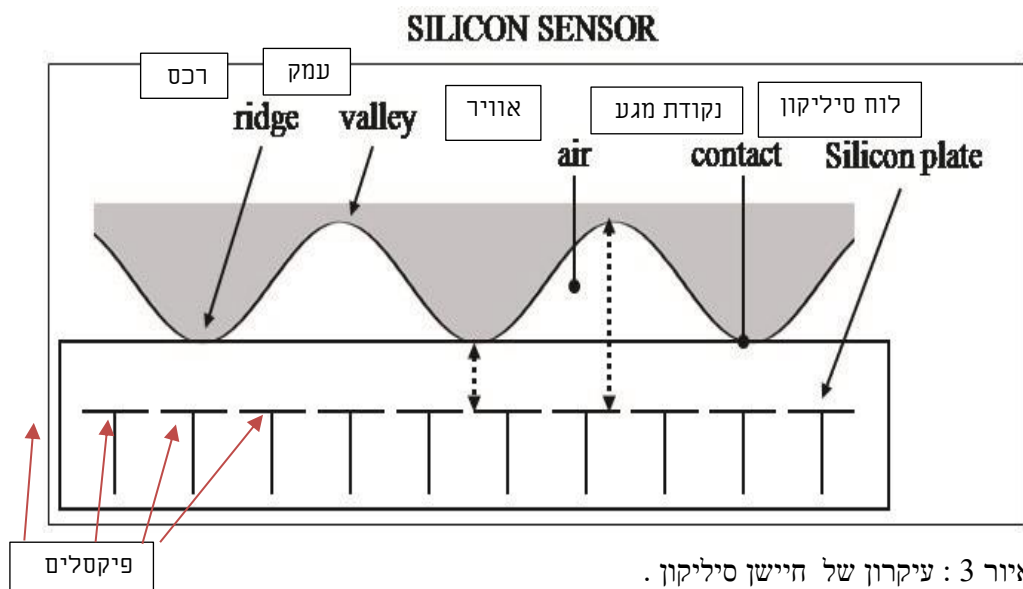
הפרטים שמפענחי טביעות האצבע מחפשים הן לא הלולאות והמערבולות. כפי שניתן לראות בתמונה לעיל, ישנם קווים רציפים מפותלים רבים הנקראים רכסים. הרכסים האלה בעיקר לא זהים. חלקם מקבל הסחות או הטיות או הפסקות פתאומיות. כמו כן ישנן נקודות קטנות או קווים קצרים או מרווחים ליד ההסתעפות של הרכסים. נקודות אלה נמצאות באופן אקראי במקומות שונים על טביעת האצבע.

2. מערכות סריקת טביעת אצבע

טביעת האצבע היא שיטת הזיהוי הידידותית ביותר למשתמש בין יישומים זיהוי ביומטרי. בדורות קודמים טביעת אצבע הייתה נלקחת על ידי לחיצת האצבע על כרית דיו והעברת האצבע עם הדיו על נייר. מטביעת האצבע ששורטטה על הנייר היו מעבדים את הנתונים הרצויים. היום לא משתמשים בשיטה זו וקיים מגוון רחב של סורקי טביעות אצבע. הם באים בגדלים שונים, צורות, וקיימים סוגי חיישנים שונים וטכנולוגיות שונות. מערכות זיהוי טביעות אצבע מסוימות יכולות לפעול ללא צורך בסיוע חיצוני כלשהו (מחשב למשל), כמו סורקי טביעות אצבע המשמשים לבקרת גישה או לנוכחות עובדים. הם מגיעים בדרך כלל עם היכולת להירשם, לאמת ולאחסן נתונים בנפרד. ניתן להעביר נתונים אלו למחשב, לטלפון חכם או למדיית אחסון.

קיימים סוגים שונים של חיישני טביעת אצבע. היותר נפוצים הם חיישן טביעת אצבע אופטי וחיישני טביעת אצבע של מצב מוצק או סיליקון. על חיישן אופטי נדבר בהרחבה בהמשך. נתחיל עם הסבר קצר על חיישן סיליקון.

איור 3 מתאר את עיקרון חיישן מצב סיליקון.



איור 3 : עיקרון של חיישן סיליקון.

חיישני מצב מוצק מורכבים ממערך של פיקסלים, כל פיקסל להיות חיישן עצמו. המשתמש מניח את האצבע על פני לוח הסיליקון, וישנן ארבע טכניקות המשמשות בדרך כלל כדי להמיר את המידע רכס/עמק לאות חשמלי:

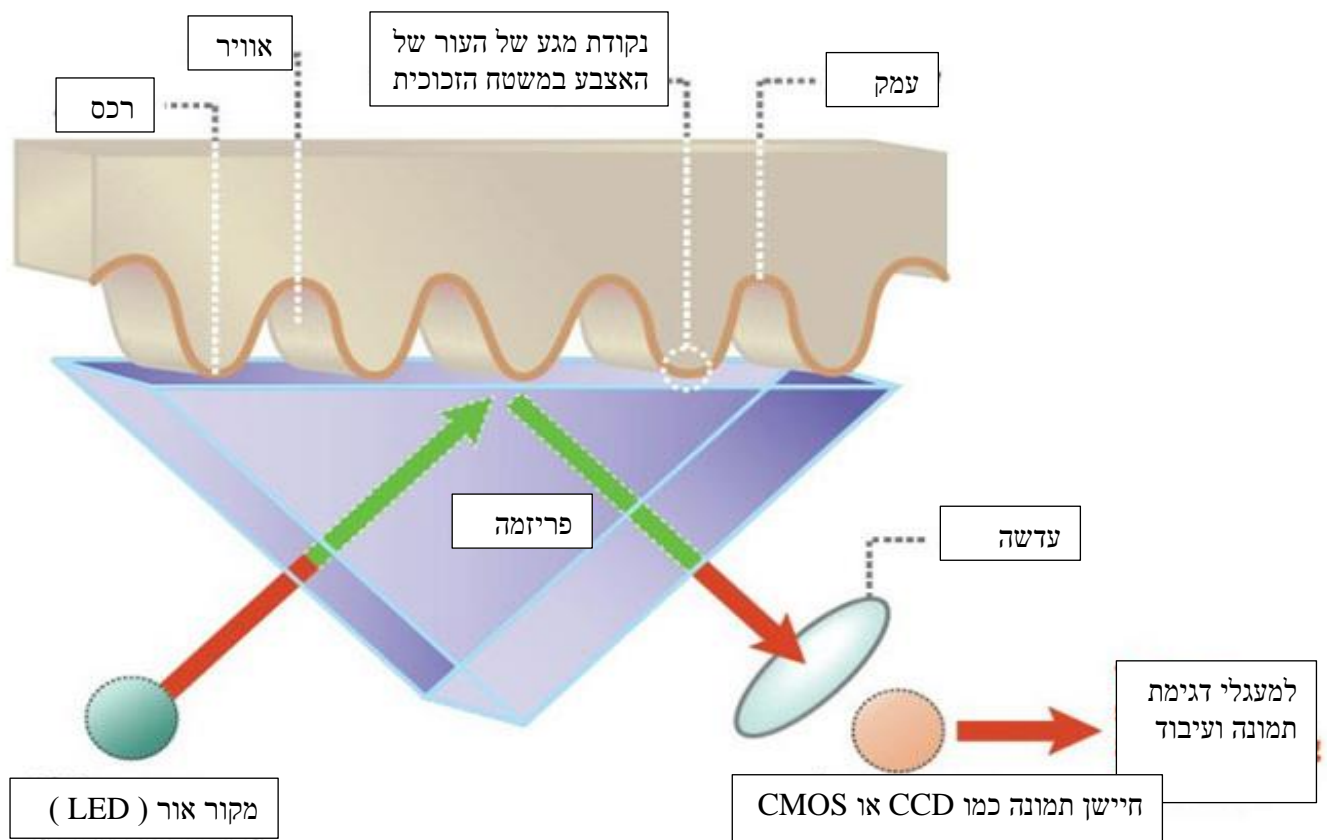
- קיבולי
- תרמי
- שדה חשמלי
- פיאזו אלקטרי (piezoelectric).

מאחר שחיישנים במצב מוצק אינם משתמשים ברכיבים אופטיים, גודלם קטן במידה ניכרת וניתן לשלב אותם בקלות בפרויקטים שגודל פיזי קטן הוא חשוב. מצד שני, חיישני הסיליקון יקרים, כך שאזור החישה של חיישנים מוצקים הוא בדרך כלל קטן ולכן גם טיב הזיהוי.

קיימים סורקי טביעת אצבע זעירים זמינים בשוק, אשר לא ניתן לבצע איתם כל פעולה מבלי להיות מחובר למחשב או לטלפון החכם. במחשב או בטלפון החכם יש את היישום המיועד המעבד את הנתונים שצולמו על ידי הסורק. לסורק עצמו אין יכולת עיבוד. מחיר של סורקי טביעת אצבע יכולים להשתנות מאוד בהתאם לגורמים כמו המותג, איטום, אישורים, סוג של חיישן וכו'. סורק טביעת אצבע קטן יכול לעלות עשרות דולרים בודדים וסורק מתוחכם לעשר אצבעות עם יכולת זיהוי אצבע יכול לעלות עד \$2500.

3. עיקרון סריקה אופטית

חישה אופטית היא השיטה המוקדמת ביותר המשמשת ללכידת תכונות טביעת אצבע. חיישני טביעת אצבע אופטיים מבצעים צילום של הפרק האחרון של האצבע. התצורה הבסיסית של לכידת תמונה טביעת אצבע על ידי חיישן אופטי הוא פחות או יותר כמו מצלמת צילום דיגיטלי. התמונה המתקבלת עוברת עיבוד כדי להפוך אותה למעשית עבור אלגוריתם זיהוי טביעת אצבע. ככל שהרזולוציה של התמונה גבוהה יותר, תגדל רמת האבטחה. באיור 4 רואים את העיקרון של חיישן טביעת אצבע אופטי.



איור 4 : עיקרון של חיישן טביעת אצבע אופטי

חיישנים אופטיים מורכבים ממקור אור פנימי, בדרך כלל LED, כי תצרוכת ההספק שלה נמוכה (זרם נמוך). הליך מקרינה על משטח פריזמה מזכוכית שהיא כעין מנסרה וקרני האור המוחזרים מרוכזים על ידי עדשות

ומגיעות לחיישן תמונה כמו CCD או חיישן CMOS אחר ומועברים למעגלי דגימת תמונה ועיבוד. העיקרון הוא קריאת האותות המוחזרים כמו סריקת תמונה של שורות משמאל לימין כאשר הרכסים והעמקים הופכים במעגל העיבוד לאפסים ואחדים. האותות שהגיעו למעגל דגימת התמונה עוברים עיבוד של הרכסים ונקודות עניין חשובות כמו סיום רכס, התפצלות רכס ורכס קצרצר ובנוסף עוברים גם עיבוד כיוון (האם כיוון הרכס נוטה ימינה או שמאלה) ורק אז נשמרת התמונה (IMAGE) של טביעת האצבע.

4. מודול טביעת אצבע אופטי

מספר מודולים של טביעת אצבע עם מחירים (נכון לנובמבר 2019) נראים באיור 5: ניתן למצא באינטרנט חיישנים החל מ 30 ש"ח והלאה.



איור 5: מודול טביעת אצבע FPM10A עם קונקטור בצד (המחיר 23.46 ש"ח).

חיישן טביעת האצבע שנסביר כאן הוא מודול נפוץ ביותר לקנייה באינטרנט ונקרא FPM10A, אך רוב התוכנות והחיבורים תואמים למודולים R305 ו-ZFM20 ועוד.

קיימים מודולים רבים ושונים עם קונקטורים שונים ויש להיזהר מאוד בחיבור.

המודול מבצע הרשמה של טביעת אצבע, עיבוד תמונה, טביעת אצבע התאמת חיפוש ואחסון תבנית. ההתאמה יכולה להיות 1:1 או N:1. הוא מתחבר בתקשורת טורית UART אל מיקרו בקר. קצב השידור המוגדר כברירת מחדל הוא בדרך כלל 57600 bps, למרות שהמודול יכול לתמוך בקצב מ-9600 עד 115200 ביטים בשנייה.

5. מאפיינים עיקריים

כדי להבין את המאפיינים של זיהוי ביומטרי נסביר מספר מושגים בסיסיים .

5.א התאמה 1:1 ו 1:N

התאמה 1:N אומרת למצא את טביעת האצבע שנמצאת כרגע בחיישן מתוך N טביעות האצבע במערכת. התאמה של 1:1 אומרת האם טביעת האצבע שנמצאת כרגע בחיישן זהה לטביעת אצבע מסוימת. כלומר האם טביעת האצבע מתאימה לטביעת אצבע מספר 20 ששמורה במערכת.

5.ב

FAR - False Acceptance Rate

שיעור הקבלה הכוזב הוא מדד הסבירות שמערכת האבטחה הביומטרית תקבל באופן שגוי ניסיון גישה של משתמש לא מורשה. בדרך כלל, הכוונה ליחס של מספר הקבלה השקרית המחולקת למספר ניסיונות הזיהוי.

5.ג

FRR - False Recognition Rate - קצב זיהוי כוזב

קצב הזיהוי הכוזב הוא מדד הסבירות שמערכת האבטחה הביומטרית תדחה באופן שגוי את ניסיון הגישה של משתמש מורשה. FRR של המערכת בדרך כלל מצוין כיחס של מספר הזהות המזויפת מחולק על ידי מספר ניסיונות הזיהוי.

7.5 זיהוי או אימות

מערכת זיהוי ביומטרית יכולה לפעול בשני מצבים שונים: **זיהוי** (identification) או **אימות** (verification).

זיהוי הוא תהליך של ניסיון לגלות את זהותו של אדם על-ידי בחינת תבנית ביומטרית המחושבת מהתכונות הביומטריות של האדם. במקרה הזיהוי, המערכת מאומנת (השלב נקרא גם רישום או הרשמה) עם תבניות של מספר אנשים. עבור כל אחד מהאנשים, מחושבת **תבנית ביומטרית**. בשלב הזיהוי מבצעים התאמה בין התבנית הרצויה כנגד כל תבנית שנלקחה בשלב האימון ונמצאת בספריית המערכת ונותנים **ציון** (או מרחק או משקולות) המתארים את הדמיון בין התבניות. המערכת בוחרת את התבנית עם **התבנית הביומטרית** הדומה ביותר. כדי למנוע בעיית התחזות על ידי כך שאיש שאיננו קיים בספריית המערכת יזוהה עם תבנית ביומטרית שנמצאת בספריית המערכת הדמיון חייב לחרוג מרמה מסוימת. אם לא מגיעים לרמה זו, התבנית נדחת. במקרה **האימות**, התבנית הביומטרית המסוימת משווה לתבנית רצויה הנמצאת בספריית המערכת הבודדת של האדם. בדומה לזיהוי, בודקים אם הדמיון בין 2 התבניות עובר רמת סף.

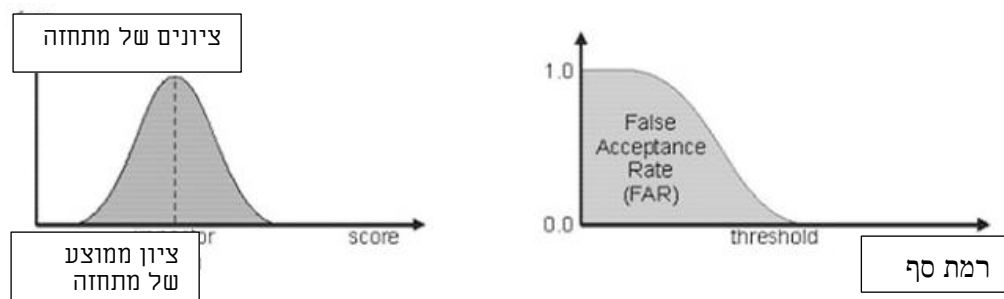
5.ה רמת סף : קבלה שגויה/זיהוי שקר (False Acceptance / false recognition)

אנו משתמשים בציונים (המכונים גם משקולות) כדי לבטא את הדמיון בין תבנית לתבנית ביומטרית. ככל שהציון גבוה יותר, כך הדמיון ביניהם גבוה יותר. לדוגמה הגישה למקום בטחוני ניתנת לאדם רק אם התוצאה של התבנית הביומטרית שלו (זיהוי) או האדם שהתבנית אומתה נגד (אימות) גבוהה מסף מסוים. באופן תאורטי, הציון של אדם שקיים בספריה צריך להיות גבוה מאשר ציוני מתחזים. קיים סף יחיד, שמפריד בין שתי הקבוצות של הציונים בין לקוחות ומתחזים.

באופן מעשי תבנית מתחזה יכולה לייצר ציון גבוה יותר מזה של הלקוחות בספריה. מכאן שאפילו אם בחרנו רמת סף מסוימת יכולים עדיין לקבל טעויות. לדוגמה: ניתן לבחור רמת סף גבוהה ואז באמת לא נקבל מתחזים אבל גם לא נקבל תבניות הנמצאות בספריה. במצב הפוך, ניתן לקבוע רמת סף מאד נמוכה כך שנקבל את התבנית הרצויה שבספריה אבל בטעות ניתן לקבל גם תבנית של מתחזה. אם קובעים רמת סף באמצע יכול לקרות שנקבל גם זיהוי נכון אבל גם זיהוי מתחזה נקבל כנכון.

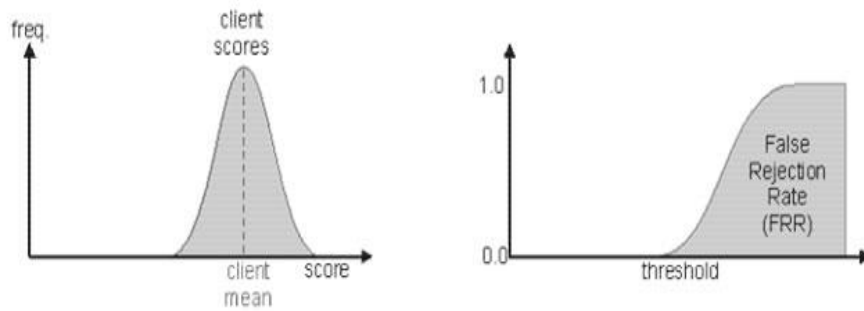
האיורים שבהמשך יסייעו בהשגת הבנה טובה יותר. נניח שיש לנו מערכת ביומטרית הבודקת כמות גדולה של נתונים המורכבים מתבניות אמיתיות וגם תבניות מתחזים. נגדיר את מושג ה **FAR**: היחס בין התבניות הכוזבות שהתקבלו כחוקיות לכלל התבניות הכוזבות נקרא **FAR** (False Acceptance Rate) – קצב קבלת טעויות או שיעור קבלת שווא. הערך שלו אחד אם כל תבניות המתחזה מתקבלות כחוקיות ואפס אם אף אחת מתבניות המתחזה לא מתקבלת. אנחנו נרצה שהיחס יהיה 0, כלומר 0 תבניות מתחזות שהתקבלו כחוקיות.

נסתכל על תבנית של מתחזה שבצד שמאל באיור 6. הציונים יהיו מפוזרים סביב ציון ממוצע ונראה התפלגות גאוס נורמלית התלויה ברמת הסף. באיור הימני לקחנו את הגרף השמאלי ושמנו אותו עבור רמות סף שונות. רואים שעבור רמת סף של 0 ורמות סף נמוכות ה **FAR** הוא 1 וזה אומר שכל התבניות של המתחזה מתקבלות כחוקיות. ככל שנעלה את רמת הסף כמות קבלת מתחזים תלך ותקטן עד שברמת סף מסוימת ואחריה לא יתקבל אף מתחזה כחוקי. ה **FAR** הוא 0.



איור 6: משמאל - ציונים של תבנית מתחזה מימין גרף FAR עבור רמות סף שונות.

עכשיו נעבור ונעבוד הפוך. נבדוק שגיאות עבור תבניות אמיתיות (שנמצאות בספריה). נייער באיור 7 .

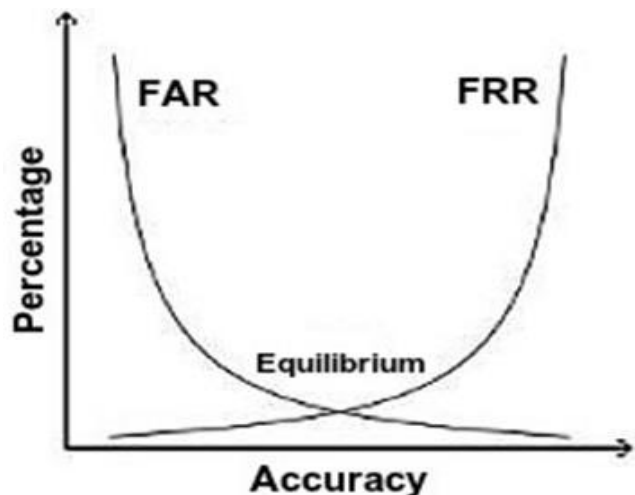


איור 7 : משמאל - ציונים של תבנית לקוח שבספריה, מימין גרף FRR עבור רמות סף שונות .

נגדיר מושג הנקרא False Rejection Rate - FRR או False Recognition Rate שמבטא את היחס בין כמות הלקוחות החוקיים שנדחתה בטעות לחלק בכמות הלקוחות הכללית . כמוכן שנרצה שהיחס יהיה 1 , כלומר אף לקוח חוקי לא יידחה.

נסתכל על תבנית של לקוח חוקי שבצד שמאל באיור. הציונים יהיו מפוזרים סביב ציון ממוצע ונראה התפלגות גאוס נורמלית התלויה ברמת הסף. באיור הימני לקחנו את הגרף השמאלי ושמנו אותו עבור רמות סף שונות. רואים שמרמת סף 0 ועד רמת סף מסוימת לא נדחה תבנית של מתחזה (כלומר כל תבנית מתחזה לא תידחה ותחשב חוקית). מרמת סף זו וימינה נתחיל לדחות חלק מתבניות מתחזים עד שברמת סף מסוימת כל התבניות של התחזים יידחו.

באיור 8 נראים גרף FRR ו ARR ביחד. הציר האופקי הוא הדיוק והאנכי הוא אחוזים.



איור 8 : גרף FAR ו FRR

בחירת נקודת הסף מאוד בעייתית. קיימת בגרף נקודה - equilibrium – שיווי משקל - שבה כמות המתחזים המתקבלת כחוקית שווה לכמות הלקוחות שנדחים.

5.7 השוואה בין מערכות ביומטריות

יצרני המערכות צריכים לציין גם את ה FAR וגם את ה FRR . אם הם מציינים רק את אחד מהם ניתן להניח שהמושג השני הוא גרוע . אבל גם כאשר היצרן מציין את שניהם – האם זה מספיק ? בהסברים שהבאנו הראינו שה FAR וה FRR תלויים ברמת הסף שנקבעה. מנסים לקבוע פרוטוקול – תקן – סטנדרט - של מערכות ביומטריות (אולי בימים אלו כבר יש ??) ואז כל יצרן ירשום תוצאות העומדות בסטנדרט ונוכל לבצע השוואה נכונה. גם עם יהיה תקן נוכל להשוות מערכות ביומטריות מאותו סוג (לדוגמה : טביעת אצבע לא תוכל להיבדק מול זיהוי פנים).

5.8 מאפייני מודול זיהוי טביעת האצבע

באזור 9 נראה טבלה של מאפיינים כלליים . עבור כל מודול יש לקרוא את המאפיינים שלו בדף הנתונים של היצרן.

Power	DC 3.6V-6.0V	Interface	UART(TTL logical level)/ USB 1.1
Working current	Typical: 100mA Peak: 150mA	Matching Mode	1:1 and 1:N
Baud rate	(9600*N)bps, N=1~12 (default N=6)	Character file size	256 bytes
Image acquiring time	<0.5s	Template size	512 bytes
Storage capacity	256	Security level	5 (1, 2, 3, 4, 5(highest))
FAR	<0.001%	FRR	<0.1%
Average searching time	< 1s (1:1000)	Window dimension	18mm*22mm
Working environment	Temp: -10°C - +40°C	Storage environment	Temp: -40°C - +85°C
	RH: 40%-85%		RH: <85%
Outline Dimention	Split type	Module: 32*23*7mm Sensor:56*20*21.5mm	
	Integral type	54.5*20.6*23.8mm	

איור 9 – טבלת מאפיינים

Power DC 3.6V-6.0V - ספק כוח

Interface UART(TTL logical level)/ USB 1.1 ממשיק

Working current Typical: <120mA Peak: <130mA זרם עבודה

Matching Mode 1:1 and 1:N (הסבר בהמשך) אופן התאמה

Baud rate	(9600*N)bps, N=1 -> 12 (default N=6)	קצב תקשורת
Character file size	256 bytes	גודל קובץ התווים
Image acquiring time	<1s	זמן קבלת התמונה
Template size	512 bytes	גודל תבנית
Storage capacity	162/ 930	קיבולת אחסון (כמה טביעות אצבע)
Security level	5 (1, 2, 3, 4, 5(highest))	רמת אבטחה
FAR	<0.001% False Acceptance Rate	(הסבר בהמשך) קצב הקבלה הכוזב -
FRR	<1% False Recognition Rate	(הסבר בהמשך) קצב זיהוי כוזב -
Average searching time	< 1s (1:160)	זמן חיפוש ממוצע (1 מ 160 טביעות)
Window dimension	14mm*18mm	גודל החלון
Working environment	Temp: -20 -> +50 °C RH:40% - 85%	טמפרטורת סביבת עבודה לחות יחסית
Storage Environment	Temp: -40 -> 85 °C RH < 90%	טמפרטורת סביבת אחסנה לחות יחסית
Outline Dimention	מידות היצרניות	
Split type Module:	42*25*8.5mm (install dimension: 31.5*19mm)	כשמגיע בנפרד
Sensor:	56*20*21.5mm	החיישן
Integral type	56*20*21.5mm	

6. עקרון העבודה עם מודול חיישן טביעת אצבע

למערכת סורק טביעות אצבע יש שני שלבים של פעולה בסיסיים :

א. **הרשמה** - **enrollment** - קבלת תמונה של האצבע הנקראת גם צילום או לכידת - capture - התמונה ויצירת קובץ תמונה הנקרא IMAGE.

ב. **התאמה** - **matching** - לקבוע אם תבנית הרכסים והעמקים בתמונה זו תואמת לתבנית של רכסים ועמקים בתמונות שנסרקו מראש. ישנן 2 אפשרויות התאמה : 1. **התאמה 1:N** האומרת למצא את טביעת האצבע שנמצאת כרגע בחיישן מתוך N טביעות האצבע במערכת. 2. **התאמה 1:1** אומרת האם טביעת האצבע שנמצאת כרגע בחיישן זהה לטביעת אצבע מסוימת. כלומר האם טביעת האצבע שבחיישן מתאימה לטביעת אצבע מספר 50 ששמורה במערכת.

בשלב ההרשמה **enrollment**, המשתמש צריך לשים אצבע בחלון החיישן פעמיים. המערכת מעבדת את 2 התמונות שהתקבלו ותיצור תבנית של האצבע המבוססת על התוצאות ושומרת אותה בספרייה של המערכת.

בשלב ההתאמה המשתמש שם את האצבע בחלון חיישן טביעת האצבע האופטי והמערכת יוצרת תבנית של האצבע ותשווה אותה לתבניות הנמצאות בספרייה של המערכת. בהתאמה של 1:1 תתבצע התאמה עם תבנית מסוימת שנרצה מתוך הספרייה. בהתאמה 1:N תתבצע התאמה עם כל התבניות שבספרייה. התשובה החוזרת ל 2 ההתאמות היא הצלחה או כשלון.

א.6 תקשורת טורית עם המודול

ישנם מודולים המתחברים בתקשורת טורית ויש כאלו שיש להם אפשרות חיבור עם USB. באיור 8 נראה טבלה שבה מתוארים הדקי התקשורת הטורית של המודול.

Pin Number	Name	Type	Function Description
1	Vin	in	Power input
2	GND	—	Signal ground. Connected to power ground (color: black)
3	TD	in	Data output. TTL logical level
4	RD	out	Data input. TTL logical level

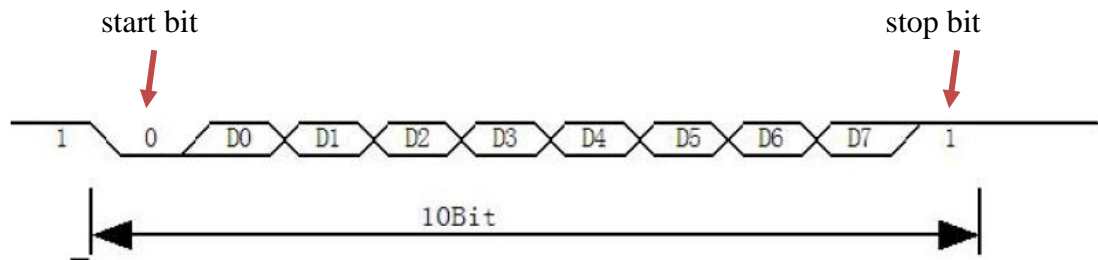
איור 8 : הדקי התקשורת הטורית של המודול.

במשק טורי המודול מתחבר עם מיקרו בקר העובדים עם מתח ספק של 3.3 וולט או 5 וולט. הדק TD של המודול מתחבר אל RXD של המיקרו בקר והדק RD של המודול עם הדק TXD של המיקרו. אם המחשב עובד ב RS232 יש לבצע המרת מתחים (לדוגמה עם רכיב MAX232).

1.ב.6 איך נראה ביית של תקשורת טורית ?

אופן העבודה הוא תקשורת טורית אסינכרונית וברירת המחדל של קצב התקשורת הוא 57600 ביטים בשנייה. המשתמש יכול לקבוע קצב רצוי מ 9600 ועד 115200 ביטים בשנייה. הפורמט של העברת הנתונים הוא של 10 ביטים ל ביית. מתחילים עם ביט התחלה (start bit), אחר כך 8 ביטים של נתון (מה LSB אל ה MSB) והביט ה 10 האחרון הוא ביט סיום (stop bit). אין ביט בדיקה (כמו לדוגמה ביט זוגיות parity). בפרק 8 נתאר כיצד נראה פרוטוקול התקשורת.

איור 9 מתאר את הנאמר:



איור 9 : פורמט של ביית

2.ב.6 זמן ה RESET

בהפעלת ספק, לוקח כ 500mS לאתחול המודול. במשך זמן זה המודול לא מקבל פקודות מהמיקרו בקר.

3.ב.6 פרמטרים חשמליים

באיור 10 רואים את הנתונים החשמליים של המודול:

1. Power supply

Item	Parameter			Unit	Note
	Min	Typ	Max		
Power Voltage (Vin)	3.6		6.0	V	Normal working value.
Maximum Voltage (Vin _{max})	□0.3		7.0	V	Exceeding the Maximum rating may cause permant harm to the Module.
Operation Current (I _{cc})	90	100	110	mA	
Peak Current (I _{peak})			150	mA	

2. TXD □ output, TTL logic level □

Item	Condition	Parameter			Unit	Note
		Min	Typ	Max		
V _{OL}	I _{OL} = □4mA			0.4	V	Logic 0
V _{OH}	I _{OH} = 4mA	2.4		3.3	V	Logic 1

3. RXD □ input, TTL logic level □

Item	Condition	Parameter			Unit	Note
		Min	Typ	Max		
V _{IL}				0.6	V	Loige 0
V _{IH}		2.4			V	Logic 1
I _{IH}	V _{IH} = 5V		1		mA	
	V _{IH} = 3.3V		30		uA	
V _{I_{max}}		□0.3		5.5	V	Maximum input voltage

איור 10 : נתונים חשמליים

באיור 10 רואים 3 טבלאות: 1. מתח ספק הכוח 2. הדק השידור TXD 3. הדק הקליטה RXD

7. מרכיבי המערכת

למערכת יש הרבה מרכיבים כדי לענות על דרישות צרכנים שונות. המודול נשלט על ידי מיקרו בקר כדוגמת AS608 של חברת Synochip

7.א מחשב או notepad

המודול מאפשר לשמור במחשב המשתמש 512 בתים (16 דפים * 32 בתים). המשתמש יכול לגשת לדף על ידי הוראות PS_WriteNotepad או PS_ReadNotepad. יש לשים לב שכאשר כותבים דף מהמחשב אל המודול הוא מוחק את הדף האורגינלי עם כל 32 הבתים שקיימים שם!

7.ב חוצץ

ב RAM של המודול קיים **חוצץ תמונה - Image buffer** ו 2 חוצצים של 512 בתים של **חוצצי קובץ תווי - Character file**. המשתמש יכול לקרוא או לכתוב לכל אחד מהחוצצים בעזרת פקודות. בזמן כיבוי ספק הכוח התוכן של החוצצים נמחק.

7.ג. חוצץ התמונה - Image Buffer

חוצץ התמונה משמש לשמירת התבנית והוא בפורמט של 288*256 פיקסלים. בתקשורת UART (טורית) שהיא איטית, מגבירים את הקצב על ידי סוג של דחיסת נתונים ומעבירים רק את 4 הביטים הגבוהים של הפיקסל (כלומר את 16 דרגות האפור - grey degree). מ 2 פיקסלים סמוכים של אותה השורה ייצרו ביית לפני ההעברה. כשמעלים אותם למחשב אז 16 דרגות האפור יורחבו ל 256 דרגות אפור, כלומר פורמט BMP (כמו מפת ביטים Bit Map). כאשר מעבירים את הנתונים עם USB התמונה היא של 8 ביט לפיקסל כלומר 256 דרגות אפור.

7.ד חוצץ קובץ תווי

ישנם 2 חוצצים CharBuffer1, CharBuffer2 וניתן להשתמש בהם לשמור גם את הקובץ התווי וגם את קובץ התבנית (template file).

7.ה ספריית טביעת האצבע

המערכת שומרת בזיכרון ה FLASH מקום לספרייה של תבניות של טביעת אצבע. התוכן לא נמחק גם בהפסקת החשמל. המערכת תשמור את טביעת האצבע האחרונה באופן אוטומטי. התבניות נכנסות לספרייה בזיכרון ה FLASH היא טורית כלומר, תבניות טביעות האצבע מסודרות בספרייה בצורה טורית 0, 1, 2 ...N. למשתמש יש גישה לספרייה על ידי מספר התבנית.

8. פרוטוקול התקשורת

הפרוטוקול מגדיר את פורמט העברת נתונים בין מחשב או מיקרו בקר למודול טביעת האצבע. הפורמט מתאים לתקשורת טורית UART וגם לתקשורת USB. כאשר מעבירים או קולטים פקודה או נתון או תוצאה אז החבילה package נראית באיור הבא:

בדיקת סכום תוכן	החבילה : הוראה/נתון/פרמטר אורך	חבילה מזהה	חבילה כתובת	כתובת	Header
Checksum	Package content (Instruction/data/Parameter)	Package length	Package identifier	Adder	Header

איור 11 : פורמט חבילת הנתונים

HEADER - כותרת - ערך קבוע של 2 בתים 0xEF01 כאשר הביית הגבוה נשלח ראשון.

ADDER - כתובת - 4 בתים. ברירת המחדל היא 0xFFFFFFFF וניתן לשנות אותו בעזרת פקודה. הביית הגבוה נשלח תחילה ואם יש שגיאה בכתובת המודול לא יחזיר תשובה.

Package identifier - מזהה חבילה - ביית שמראה מה סוג החבילה כמתואר באיור 12 :

Package identifier	PID	1 byte	
		01H	Command packet;
		02H	Data packet; Data packet shall not appear alone in executing process, must follow command packet or acknowledge packet.
		07H	Acknowledge packet;
		08H	End of Data packet.

איור 12 : מזהה החבילה

01H - חבילה של פקודה **02H** - חבילה של נתון (החבילה מופיעה אחרי חבילת פקודה או חבילת אישור). **07H** - חבילת אישור. **08H** - סוף חבילת נתונים.

LENGTH - אורך החבילה - 2 בתים שמציינים את הסכום של כמות הבתים בחבילת הפקודה וחבילת הנתונים פלוס 2 בתים של ה Checksum. אורך מקסימלי הוא 256 כאשר הביית הגבוה מועבר תחילה.

Package content - תוכן החבילה - יכול להיות פקודות, נתונים, פרמטרים של פקודות, אישור תוצאות וכו'. ערך תווי של טביעת אצבע ותבניות נחשבות כנתונים.

Checksum - בדיקת סכום - נקרא גם סיכום ביקורת שהוא מנגנון אלגוריתמי לבדיקת תקינות ההעברה של קובצי נתונים. כאן זה 2 בתים שהם הסכום החשבוני של מזהה החבילה, אורך החבילה וכל תוכן החבילה. אם יש גלישות (הסכום עובר את הערך שיכולים להכיל 2 בתים) לא מתייחסים אליהן. הביית הגבוה מועבר בתחילה.

ACKNOWLEDGE - פקודות נשלחות מהמחשב למודול והמודול מגיב באישור.

כאשר המודול מקבל פקודה ממחשב/מיקרו הוא ידווח על סטטוס ביצוע הפקודה ועל התוצאות בעזרת חבילת אישור. לחבילת אישור יש פרמטרים ויכולה להיות אחריה גם חבילת נתונים. למחשב/מיקרו אין אפשרות לאמת את הסטטוס או הפקודה או תוצאות הביצוע שנקלטו אלא רק דרך חבילת האישור של המודול. חבילת האישור כוללת 1 ביית של אישור קוד ואולי גם פרמטר מוזר.

8.A הקודים של האישור :

- א. 00H - הפקודה בוצעה או OK (הכל בסדר).
- ב. 01H - שגיאה בקבלת חבילת נתונים.
- ג. 02H - אין אצבע על החיישן.
- ד. 03H - כשלון בהרשמה האצבע.
- ה. 06H - כשלון ביצירת קובץ תווי כתוצאה מתמונת טביעת אצבע מבולגנת מידי.
- ו. 07H – כשלון ליצור קובץ תווי כתוצאה מחוסר נקודות איפיון או תמונת טביעת אצבע קטנה מידי.
- ז. 08H – אין תאימות טביעת אצבע.
- ח. 09H - כשלון למצא אצבע תואמת.
- ט. 0AH - כשלון באיחוד הקבצים התווים.
- י. 0BH - הכתובת של pageID הוא מחוץ לספריית טביעת האצבע.
- יא. 0CH – טעות בקריאת התבנית מהספרייה או התבנית לא נמצאת.
- יב. 0DH – טעות בהטענת התבנית.
- יג. 0EH – המודול איננו יכול לקבל את חבילות הנתונים הבאות.
- יד. 0FH – שגיאה בטעינת תמונה.
- טו. 10H – כשלון במחיקת תבנית.
- טז. 11H – כשלון במחיקת ספרית האצבע.
- יז. 13H – סיסמה שגויה.
- יח. 15H – כשלון ביצירת תבנית בגלל חוסר תמונה ראשוני חוקית.
- יט. 18H – שגיאה בכתיבה ל FLASH.
- כ. 19H – שגיאה לא מוגדרת.
- כא. 1AH – מספר רגיסטר לא חוקי.
- כב. 1BH – רגיסטר תצורה לא נכון.
- כג. 1CH – מספר דף מחשב שגוי.
- כד. 1DH – כשלון בהפעלת פורט התקשורת.
- כה. כל השאר הם רזרבה של המערכת.

9. מערכת ההוראות של המודול

כל הפקודות / נתונים מועברות לפי פורמט חבילה שבפרק הקודם.

9.א הוראת עיבוד טביעת אצבע

3 הפקודות בסעיפים 1.א.9 עד 3.א.9 משמשות בביצוע הרשמה של טביעת אצבע.

1.א.9 - איסוף תמונת אצבע - Collect finger image - GenImg (או getImage)

תאור : לגלות שיש אצבע בחלון המודול ולאחסן את תמונת האצבע בחוצץ התמונה ImageBuffer. מחזירים קוד אישור הצלחה אם הפקודה הצליחה. אם אין אצבע בחלון המודול אז קוד האישור יהיה "לא מזהה אצבע" - can't detect finger .

פרמטר כניסה : אין

פרמטר מוחזר : קוד אישור (1 בייט) .

קוד ההוראה : 01H .

פורמט החבילה נראה באיור 13 :

2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
Header	Module address	Package identifier	Package length	Instruction code	Checksum
0xEF01	Xxxx	01H	0003H	01H	05H

איור 13 : פורמט החבילה של איסוף תמונת אצבע

2 בתים ראשונים הם של הכותרת והם עם הערך הקבוע : 0xEF01

4 בתים הבאים גם הם קבועים : 0xFFFFFFFF

מזהה החבילה הוא של 1 בייט וערכו 01H שמציין שמדובר בפקודה.

אורך החבילה הוא של 2 בתים וערכו 3 (החיבור של כמות הבתים של קוד ההוראה (1 בייט) ועוד ה Checksum (2 בתים)).

ה Checksum הוא 5 והוא הסכום של מזהה החבילה – Package identifier (01H) ועוד אורך החבילה

- Package length (0003H) ועוד קוד ההוראה - Instruction code (01H) , כלומר :

. 01H+0003H+01H= 0005H

האישור שיגיע מהמודול נראה באיור 14 :

2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
Header	Module address	Package identifier	Package length	Confirmation code	Checksum
0xEF01	Xxxx	07H	0003H	xxH	Sum

איור 14 : האישור ממודול טביעת האצבע

2 בתים ראשונים הם של הכותרת והם עם הערך הקבוע : 0xEF01

4 בתים הבאים גם הם קבועים : 0xFFFFFFFF

מזהה החבילה הוא של 1 ביית וערכו 07H שמציין שמדובר בחבילת אישור.
 אורך הפקודה הוא של 3 בתים וערכו 3 (החיבור של כמות הבתים של קוד האישור (1 ביית) ועוד ה
 Checksum (2 בתים).

ה Checksum הוא הסכום של מזהה החבילה – Package identifier (07H) ועוד אורך החבילה -
 Package length (0003H) ועוד קוד האישור - Confirmation code לפי המספרים הבאים:

00H - הצלחה באיסוף נתוני האצבע .

01H - שגיאה בקבלת החבילה.

02H - לא מזהה אצבע בחלון המודול.

03H - כשלון באיסוף נתוני האצבע.

לדוגמה : אם קוד האישור הוא 00H (הצלחה באיסוף נתוני האצבע) אז :

$$\text{Checksum}=07\text{H}+0003\text{H}+00\text{H}=000\text{AH}$$

2.9.2 יצירת קובץ תווי מהתמונה - Img2Tz (או image2TZ)

תאור : ליצור בחוצץ התמונה - ImageBuffer - קובץ תווי מהתמונה של האצבע ולאחסן אותה בחוצץ

תווי 1 - CharBuffer1 או בחוצץ תווי 2 - CharBuffer2 .

פרמטר כניסה : מספר החוצץ התווי - BufferID (1 או 2).

פרמטר מוחזר : קוד אישור (ביית אחד).

קוד הפקודה : 02H .

פורמט החבילה נראה באיור 15 :

2 bytes	4bytes	1 byte	2 bytes	1 byte	1 byte	2 bytes
Header	Module address	Package identifier	Package length	Instruction code	Buffer number	Checksum
0xEF01	xxxx	01H	04H	02H	BufferID	sum

איור 15 : פורמט חבילה של יצירת קובץ תמונה

החבילה נראית כמו חבילת הפקודה הקודמת חוץ מקוד ההוראה - Instruction code - שהוא 02H

שאומר שהפקודה היא של חבילת נתונים .

מספר החוצץ - Buffer number – רושמים 1 או 2 . מספר שונה מ 1 או 2 - נחשב כ 2 .

האישור שיגיע מהמודול נראה באיור 16 :

2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
Header	Module address	Package identifier	Package length	Confirmation code	Checksum
0xEF01	xxxx	07H	03H	XxH	sum

איור 16 : פורמט האישור

- פורמט האישור דומה לזה של איור 14 . קוד האישור Confirmation code יכול להיות :
- 00H : יצירת קובץ תווי הסתיימה.
 - 01H : שגיאה בקליטת החבילה.
 - 06H : כשלון ביצירת קובץ תווי כתוצאה מתמונת טביעת אצבע מבולגנת מידי.
 - 07H : כשלון ליצור קובץ תווי כתוצאה מחוסר נקודות איפיון או תמונת טביעת אצבע קטנה מידי.
 - 15H : כשלון ביצירת תבנית בגלל חוסר תבנית ראשוני חוקית.

3.א.9 שמירת תבנית Store

- תיאור :** לשמור תבנית של חוצץ 1 או חוצץ 2 במקום המיועד בזיכרון ה FLASH .
- פרמטר כניסה :** מספר החוצץ (1 או 2), מזהה הדף PageID , המיקום ב FLASH , 2 בתים כשהביית הגבוה בהתחלה והנמוך אחריו.
- פרמטר מוחזר :** קוד אישור (ביית אחד).
- קוד הפקודה :** 06H .
- פורמט חבילת הפקודה נראה באיור 17 :

2 bytes	4bytes	1 byte	2 bytes	1 byte	1 byte	2 bytes	2 bytes
Header	Module address	Package identifier	Package length	Instruction code	buffer number	Location number	Checksum
0xEF01	xxxx	01H	06H	06H	BufferID	PageID	sum

- איור 17 : פורמט פקודת שמירת תבנית
- 7 הבתים הראשונים מורכבים מ 2 בתים של הכותרת , 4 בתים של 0xFFFFFFFF ובית של מזהה פקודה שהוא 01H שאומר שמדובר בפקודה.
- אורך החבילה Package length הוא 06H בתים שהם הסכום של קוד הפקודה (ביית אחד) ועוד מספר החוצץ (ביית אחד) ועוד 2 בתים של מספר המיקום – Location number ועוד 2 בתים של בדיקת סכום - Cchecksum .
- קוד הפקודה - Instruction code - הוא 06H שאומר שמירת תבנית.
- מספר החוצץ - Buffer number - בית אחד שהוא 01H או 02H . ערך אחר ייחשב כ 02H .
- מספר הדף – PageID – 2 בתים שהם למעשה מספר התבנית (מספר טביעת האצבע).
- בדיקת סכום - Checksum - 2 בתים שהם סכום מזהה החבילה + אורך החבילה + קוד הפקודה + מספר החוצץ + מספר הדף.

האישור שיגיע מהמודול נראה באיור 18 :

2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
Header	Module address	Package identifier	Package length	Confirmation code	Checksum
0xEF01	Xxxx	07H	03H	xxH	sum

איור 18 : חבילת האישור לפקודת שמירת תבנית.

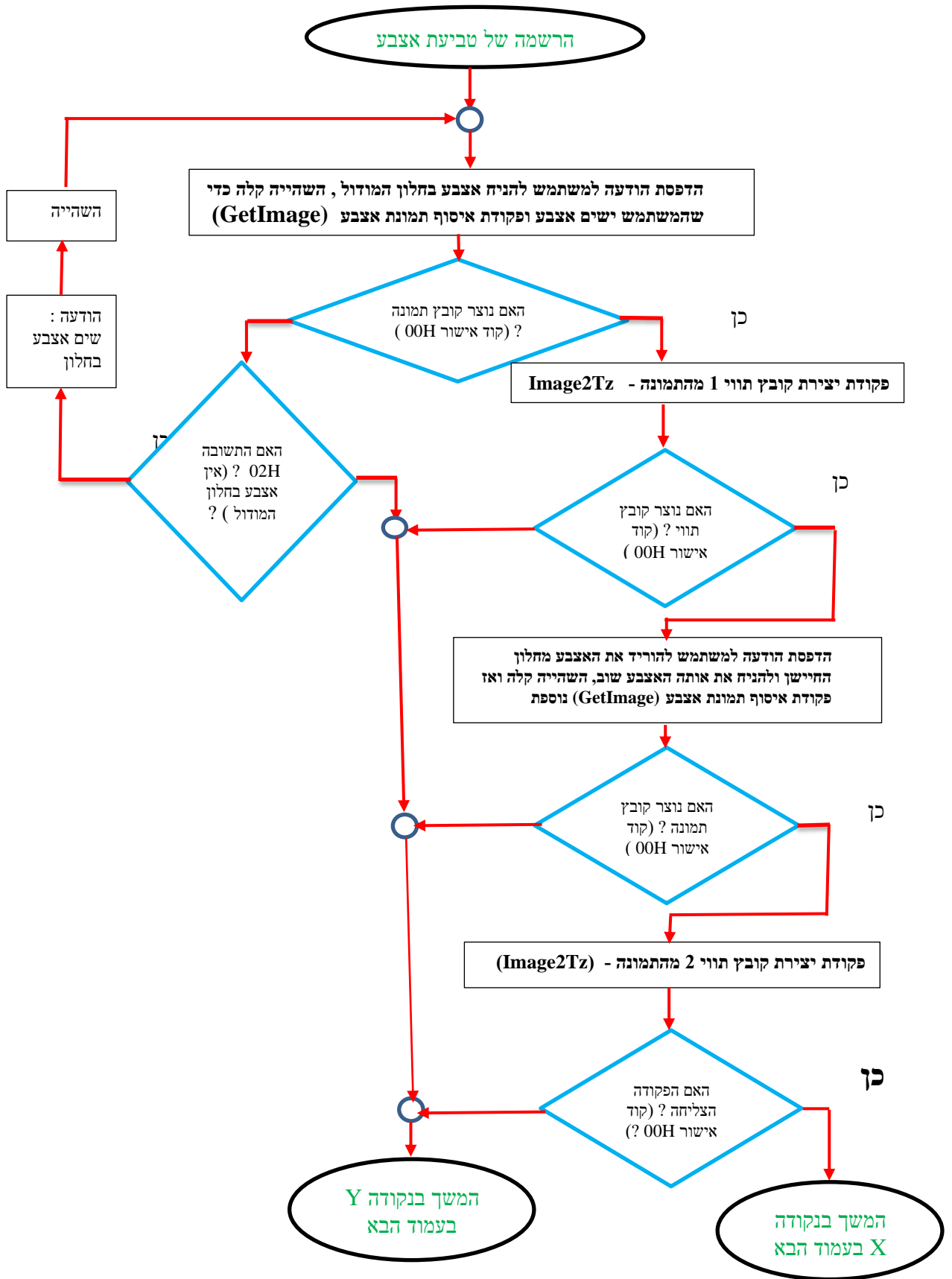
- 7 בתים ראשונים הם כמו האישורים בחבילות אישור קודמות.
אורך החבילה Package length של 2 בתים וערכו 0003H .
קוד האישור Confirmation code של בית אחד וערכו יכול להיות :
00h – שמירה הצליחה.
01H - שגיאה בקבלת החבילה.
0BH - כתובת מזהה הדף PageID הוא מעבר לספריית האצבע.
18H - שגיאה בכתיבה ל FLASH .

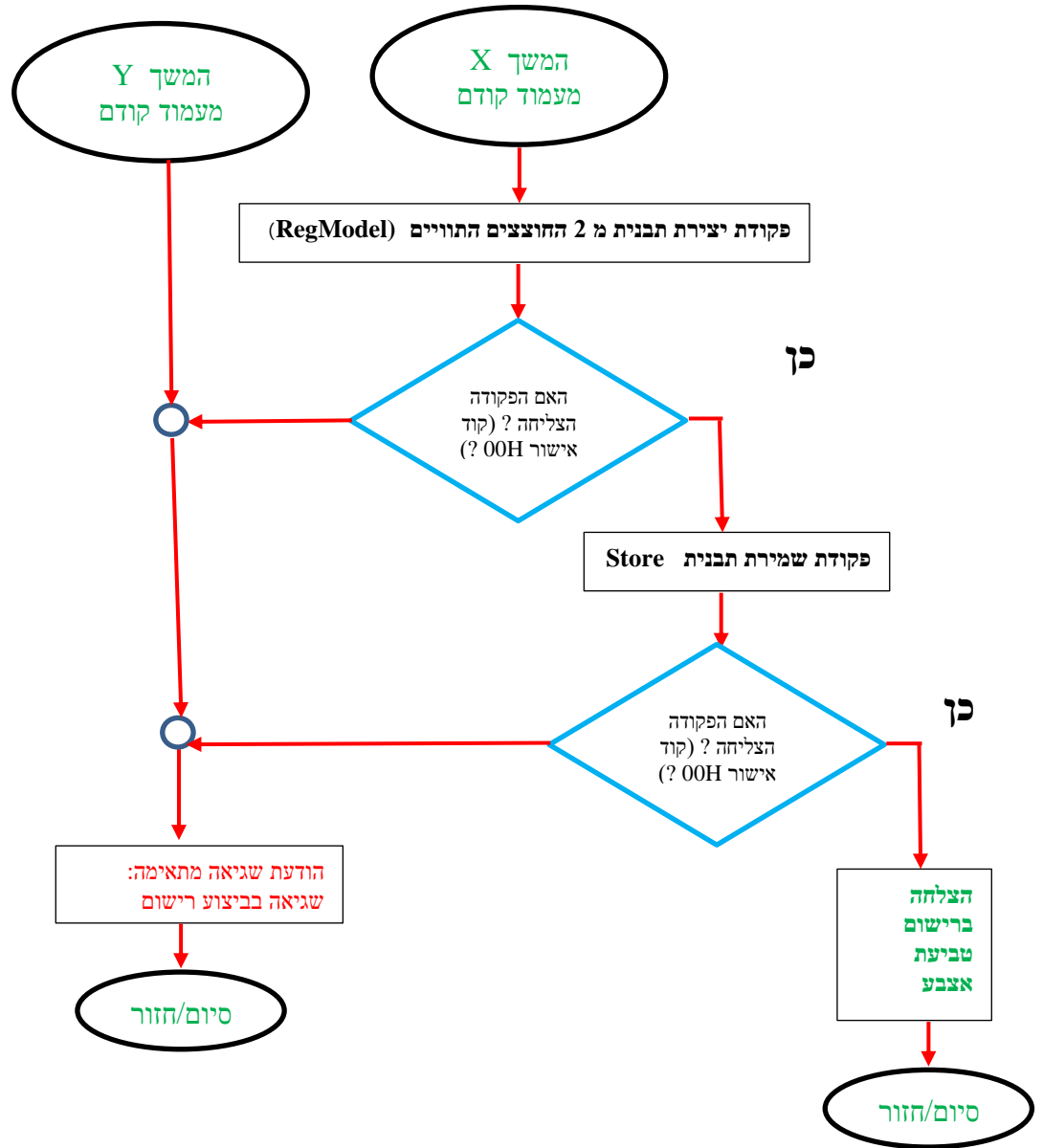
10. א. הרשמה - Enrollment

כאשר רוצים להכניס טביעות אצבע של אנשים מסוימים שיישמרו בספרייה של הרכיב יש לבצע פעולת הרשמה הנקראת Enrollment. בפעולת ההרשמה מתבצעות הפעולות הבאות :

- א. המשתמש מניח את האצבע בחלון המודול.
- ב. כאשר האצבע מונחת בחלון המיקרו שולח למודול פקודת getImage (חלק מהתוכנות קוראים לפונקציה GenImg - Generate Image) ליצור תמונה מהאצבע המונחת בחלון. בסיום יצירת התמונה המודול מחזיר אישור 00H או כשלון.
- ג. במידה וקיבלנו אישור מתבצעת פקודה Img2Tz - ליצור קובץ תווי מהתמונה שנוצרה. המודול מקודד את קובץ התמונה בעזרת אלגוריתם כדי לקבל נקודות אבחנה מיוחדות ויוצר קובץ תווי הנשמר בחוצץ 1 - CharBuffer1. במידה והתקבל אישור מהמודול מוציאים הודעה למשתמש לסלק את האצבע מחלון המודול.
- ד. המשתמש מסלק את היד מחלון המודול ואז מקבל הודעה לשים את אותה האצבע פעם נוספת. המשתמש מניח את אותה האצבע פעם נוספת בחלון המודול.
- ה. כאשר האצבע מונחת בחלון המיקרו שולח פקודת GenImg (Generate Image) פעם נוספת ליצור תמונה נוספת מהאצבע המונחת בחלון.
- ו. עם קבלת אישור מהמודול שנוצר קובץ תמונה, המיקרו שולח פקודה Img2Tz - ליצור קובץ תווי שני מהתמונה שנוצרה. נוצר קובץ תווי הנשמר בחוצץ 2 - CharBuffer2.
- ז. עם קבלת האישור ליצירת הקובץ התווי השני המיקרו שולח פקודת RegModel ליצירת תבנית מ 2 החוצצים התוויים .
- ח. עם קבלת אישור מהמודול שנוצר משני קבצי התו קובץ תבנית אחד המיקרו שולח פקודת STORE לשמור את קובץ התבנית בזיכרון המודול.

תרשים זרימה לביצוע הרשמה של אצבע בספריית האצבע.

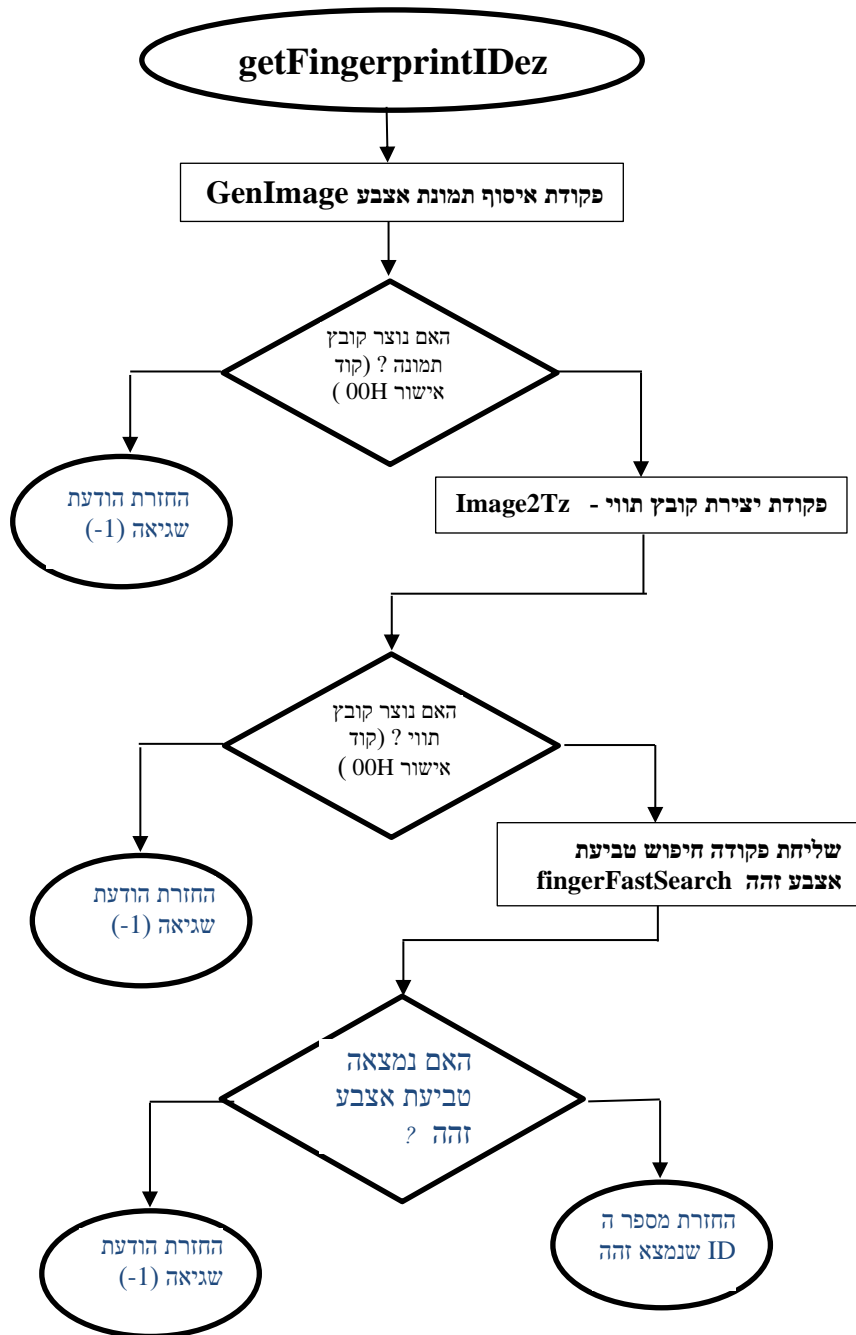




איור 19א : תרשים זרימה של הרשמה – Enrollment של טביעת אצבע.

הסבר על התרשים ניתן לקרא בסעיף 10א שהוסבר מקודם .

11.ב תרשים זרימה של פונקציית זיהוי טביעת אצבע `getFingerprintIDez`



איור 19 ב: תרשים זרימה של זיהוי ה id (מספר) של טביעת אצבע.

הסבר :

- שולחים פקודה ליצירת קובץ תמונה
- אם נוצר קובץ תמונה שולחים פקודה ליצירת קובץ תווי.
- אם הפקודה הצליחה שולחים פקודת חיפוש במאגר טביעות האצבע של המודול.
- אם נמצאה טביעת אצבע זהה מוחזר מספרה הסידורי של טביעת האצבע.
- בכל בעיה באחד השלבים מחזירים קוד שגיאה של מינוס אחד. יש אפשרות לזהות מה הייתה השגיאה שנוצרה. המודול מחזיר קוד שמסביר מה הבעיה. (סעיף 8א).

12. פרמטרים של תצורת המערכת (קונפיגורציה)

למשתמש יש מספר פרמטרים שעליהם הוא שולט כדי להקל עליו בשימוש במודול . ההוראות הבסיסיות הן SetSysPara (קבע את הפרמטרים של המערכת) ו ReadSysPara (קרא את הפרמטרים של המערכת). שתי הפקודות משתמשות במספר הפרמטר כפרמטר (לכל פרמטר יש מספר . לדוגמה שינוי קצב התקשורת הוא פרמטר מספר 4 וכו'...נשמע מסובך...?.. נבין בהמשך). כאשר שולחים פקודה לשינוי פרמטר המודול מגיב על ידי שליחת הפרמטר האורגינלי של התצורה ואז מבצע את שינוי הפרמטר וכותב את התצורה החדשה ב FLASH. בהפעלה הבאה המערכת תפעל עם הפרמטרים החדשים.

11.א בקרה על קצב התקשורת - Baud rate control (פרמטר מספר 4)

הפרמטר שולט על קצב התקשורת הטורית של ה UART של המודול. ערך הפרמטר הוא מספר שלם בין 1 ל 12 . הקצב שנקבע הוא ערך המספר כפול 9600 ביטים בשנייה . לדוגמא אם שלחנו את הערך 2 אז קבענו את קצב התקשורת ל $2 * 9600 = 19200$ ביטים בשנייה.

11.ב רמת האבטחה - Security Level - (פרמטר מספר 5)

הפרמטר שולט על רמת הסף של ההתאמה של חיפוש והתאמה של טביעת האצבע. רמת האבטחה מחולקת ל 5 רמות מ 1 עד 5 . ברמה 1 ה FAR הוא הגבוה ביותר וה FRR הוא הנמוך ביותר (ראה סעיפים 5 ו 5ג) . ברמה 5 FAR הוא הנמוך ביותר ו FRR הוא הגבוה ביותר.

11.ג נתון אורך (גודל) החבילה - Data package length (פרמטר מספר 6)

הפרמטר קובע את האורך המקסימאלי של חבילת ההעברה כשעובדים עם מחשב. הערכים שלו הם 0 , 1 , 2 , 3 המתאימים ל 32 בתים, 64 בתים, 128 בתים ו 256 בתים .

לפירוט בנושא פקודות למודול ניתן להיעזר באתר :

<https://www.rhydolabz.com/documents/finger-print-module.pdf>

אנחנו נתעכב על פקודות עיבוד טביעת אצבע.

12. פקודות עיבוד טביעת אצבע

נתאר כאן בהרחבה מספר קטן של הפקודות שמשמשות להרשמה של טביעת אצבע ואימות טביעת אצבע. בסיום פרק 12 יש טבלה מסכמת של כל הפקודות ב 2 אופנים : א. לפי פונקציות(תפקיד) ב. לפי סדר כרונולוגי של הפקודות.

12.א איסוף תמונת אצבע - GenImg (הפונקציה גם נקראת getImage)

תיאור : קוד הפקודה 01H . הפקודה בודקת האם יש אצבע בחלון החיישן ואם כן היא שומרת את התמונה של האצבע בחוצץ התבנית imageBuffer . מחזירה קוד אישור אם הייתה הצלחה. אם אין אצבע בחלון החיישן תחזיר קוד 02H - "can't detect finger" שאומר שאין אצבע בחיישן.
 הפונקציה איננה מקבלת פרמטרים . הפונקציה מחזירה קוד אישור .
 פורמט הפקודה נראה כך :

2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
Header	Module address	Package identifier	Package length	Instruction code	Checksum
0xEF01	Xxxx	01H	03H	01H	05H

פורמט חבילת האישור :

Header	Module address	Package identifier	Package length	Confirmation code	Checksum
0xEF01	Xxxx	07H	03H	xxH	Sum

00H - אישור הצלחה. נוצר ב imageBuffer איסוף של תמונת טביעת אצבע בחוצץ התמונה.
 01H – שגיאה בקליטת חבילת הפקודה (החבילה לא חוקית).
 02H – אין אצבע בחיישן.
 03H – כשולן באיסוף טביעת אצבע.

12.ב יצירת קובץ תווי מקובץ תמונה Img2Tz

תיאור : פקודה ליצירת קובץ תווי (אפשר לקרוא לו גם קובץ מאפיין) מתמונת האצבע האורגנית שב imageBuffer ושמירה של הקובץ ב CharBuffer1 או CharBuffer2 . קוד הפקודה 02H . מקובץ התמונה נלקחים מאפינים ייחודיים לטביעת אצבע כמו סיום רכס, התפצלות רכס, רכס קצר ועוד .
 פרמטר כניסה : מספר החוצץ התווי (1 או 2) . הפונקציה מחזירה קוד אישור .
 פורמט הפקודה :

2 bytes	4bytes	1 byte	2 bytes	1 byte	1 byte	2 bytes
Header	Module address	Package identifier	Package length	Instruction code	Buffer number	Checksum
0xEF01	xxxx	01H	04H	02H	BufferID	sum

BufferID הוא מספר 1 או 2 . מספר גדול מ 2 נחשב כ 2 .

פורמט חבילת האישור :

2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
Header	Module address	Package identifier	Package length	Confirmation code	Checksum
0xEF01	xxxx	07H	03H	XxH	sum

00H - נוצר קובץ תווי.

01H - שגיאה בקליטת חבילת הפקודה.

06H – כשולן ביצירת קובץ תווי כתוצאה מחוסר סדר בתמונת האצבע.

07H – כשולן ביצירת קובץ תווי כתוצאה מחוסר נקודות בתמונה או תמונת אצבע קטנה במיוחד.

15H - כשולן ביצירת קובץ תווי כתוצאה מחוסר תמונה ראשית תקפה.

12.g יצירת קובץ תבנית RegModel

תיאור : יצירת קובץ תבנית אחד מתוך שני קבצי התווים שב CharBuffer1 ו CharBuffer2 . קובץ התבנית נשמר בחזרה בשני החוצצים CharBuffer1 ו CharBuffer2 . קוד הפקודה 05H . הפונקציה איננה מקבלת פרמטרים ומחזירה ביית אחד של קוד אישור. פורמט חבילת הפקודה :

2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
Header	Module address	Package identifier	Package length	Instruction code	Checksum
0xEF01	xxxx	01H	03H	05H	09H

פורמט חבילת האישור :

2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
Header	Module address	Package identifier	Package length	Confirmation code	Checksum
0xEF01	xxxx	07H	03H	xxH	sum

00H – נוצר קובץ תבנית.

01H – שגיאה בקליטת חבילת הפקודה.

0aH – כשולן בשילוב 2 קבצי התווים. 2 הקבצים לא שייכים לאותה האצבע.

12.d שמירת קובץ התבנית - Store template

תיאור : שמירת התבנית של החוצץ המסוים (CharBuffer1 או CharBuffer2) במיקום רצוי בספרייה בזיכרון ה FLASH . קוד הפקודה 06H . הפונקציה מקבלת את מספר החוצץ BufferID ואת ה PageID שהוא המיקום בזיכרון ה FLASH שמורכב משני בתים (הביית הגבוה תחילה). הפונקציה מחזירה קוד אישור. פורמט חבילת הפקודה :

2 bytes	4bytes	1 byte	2 bytes	1 byte	1 byte	2 bytes	2 bytes
Header	Module address	Package identifier	Package length	Instruction code	buffer number	Location number	Checksum
0xEF01	xxxx	01H	06H	06H	BufferID	PageID	sum

BufferID הוא 1 או 2 ובו קובעים האם לשמור את CharBuffer1 או CharBuffer2 (מספר גדול מ 2 נחשב כ 2).
פורמט חבילת האישור :

2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
Header	Module address	Package identifier	Package length	Confirmation code	Checksum
0xEF01	Xxxx	07H	03H	xxH	sum

00H - שמירה הצליחה.

01H – שגיאה בקליטת חבילת הפקודה.

0bH – כתובת ה PageID מעבר לתחום ספריית האצבע.

18H – שגיאה בכתיבה לזיכרון ה FLASH .

12. טעינה של קובץ תבנית מזיכרון ה FLASH LoadChar

תיאור : טעינה של תבנית ממיקום רצוי בזיכרון PageID לחוצץ התווי CharBuffer1 או CharBuffer2 .
קוד הפקודה 07H .

הפונקציה מקבלת את מספר החוצץ BufferID (חוצץ 1 או 2) ואת ה pageID (מיקום התבנית בזיכרון ה FLASH (2 בתים, הביית הגבוה תחילה). הפונקציה מחזירה קוד אישור .
פורמט חבילת הפקודה:

2 bytes	4bytes	1 byte	2 bytes	1 byte	1 byte	2 bytes	2 bytes
Header	Module address	Package identifier	Package length	Instruction code	buffer number	Page number	Checksum
0xEF01	xxxx	01H	06H	07H	BufferID	PageID	sum

BufferID הוא 1 או 2 ובו קובעים האם לשמור את CharBuffer1 או CharBuffer2 (מספר גדול מ 2 נחשב כ 2).

פורמט חבילת האישור:

2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
Header	Module address	Package identifier	Package length	Confirmation code	Checksum
0xEF01	xxxx	07H	03H	XxH	sum

00H – טעינה הצליחה.

01H - שגיאה בקליטת חבילת הפקודה.

0BH - כתובת ה PageID מעבר לתחום ספריית האצבע.

0cH – שגיאה בקריאת התבנית מהספרייה או התבנית איננה תקפה (אין תבנית כזו או הנתונים המוצגים אינם תקפים).

1.12 – בדיקת התאמה בין 2 תבניות אצבע

To carry out precise matching of two finger templates Match

תיאור : ביצוע התאמה מדויקת בין 2 התבניות שב CharBuffer1 ו CharBuffer2 ולתת תוצאות להתאמה .
קוד הפקודה 03H .

הפונקציה איננה מקבלת פרמטרים ומחזירה תוצאות התאמה .
פורמט חבילת הפקודה :

2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes
Header	Module address	Package identifier	Package length	Instruction code	Checksum
0xEF01	Xxxx	01H	03H	03H	07H

פורמט חבילת האישור :

2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes	2 bytes
Header	Module address	Package identifier	Package length	Confirmation code	Matching score	Checksum
0xEF01	Xxxx	07H	05H	XxH	XxH	sum

00H – 2 התבניות תואמות.

01H - שגיאה בקליטת חבילת הפקודה.

08H - 2 התבניות אינן תואמות .

הפקודה איננה משפיעה על תוכן החוצצים.

1.12 – חיפוש בספריית האצבע - Search - To search finger library

תיאור : לחפש בכל ספריית האצבע תבנית תואמת את זו שב CharBuffer1 או CharBuffer2 . אם נמצאת תבנית כזו יוחזר ה PageID . קוד הפקודה 04H .
הפונקציה מקבלת את הפרמטרים הבאים : מספר החוצץ BufferID , כתובת התחלת החיפוש Start Page , מספרי החיפוש (PageNum) .

הפונקציה מחזירה קוד אישור שהוא PageID (מיקום התבנית התואמת).

פורמט חבילת הפקודה:

2 bytes	4bytes	1 byte	2 bytes	1 byte	1 byte	2 bytes	2 bytes	2 bytes
Header	Module address	Package identifier	Package length	Instruction code	buffer number	Parameter	Parameter	Checksum
0xEF01	xxxx	01H	08H	04H	BufferID	StartPage	PageNum	sum

מספר החוצץ BufferID הוא 1 או 2 . ערך גבוה יותר ייחשב כ 2 .

פורמט חבילת האישור :

2 bytes	4bytes	1 byte	2 bytes	1 byte	2 bytes	2 bytes	2 bytes
Header	Module address	Package identifier	Package length	Confirmation code	页码	得分	Checksum
0xEF01	xxxx	07H	7	xxH	PageID	MatchScore	sum

00H – נמצאה אצבע תואמת.

01H - שגיאה בקליטת חבילת הפקודה.

09H – אין התאמה בספרייה (גם ה PageID וגם ה MatchScore הם 0).

ההוראה איננה משפיעה על תוכן החוצצים.

12. סיכום פקודות

1.12. לפי פונקציה

type	num	code	description	Type	num	Code	description
פ ק ו ד ו ת מ ע ר ת	1	13H	To verify password	ע י ו ד ט ו ת א ע	13	08H	to upload template
	2	12H	To set password		14	09H	To download template
	3	15H	To set device address		15	06H	To store template;
	4	0EH	To set system Parameter		16	07H	to read/load template
	5	17H	Port control		17	0CH	to delete tempates
	6	0FH	To read system Parameter		18	0DH	to empty the library
	7	1DH	To read finger template numbers		19	03H	Carry out precise matching of two templates;
ע י ב ו ד ט ו ת א ע	8	01H	Collect finger image	א ח ר ו ת	20	04H	Search the finger library
	9	0AH	To upload image		21	14H	to get random code
	10	0BH	To download image		22	18H	to write note pad
	11	02H	To generate character file from image		23	19H	To read note pad
	12	05H	To combine character files and generate template				

טבלת סיכום פקודות לפי התפקיד

בעמוד הבא מוצגת טבלת הפקודות לפי סדר כרונולוגי של הקוד.

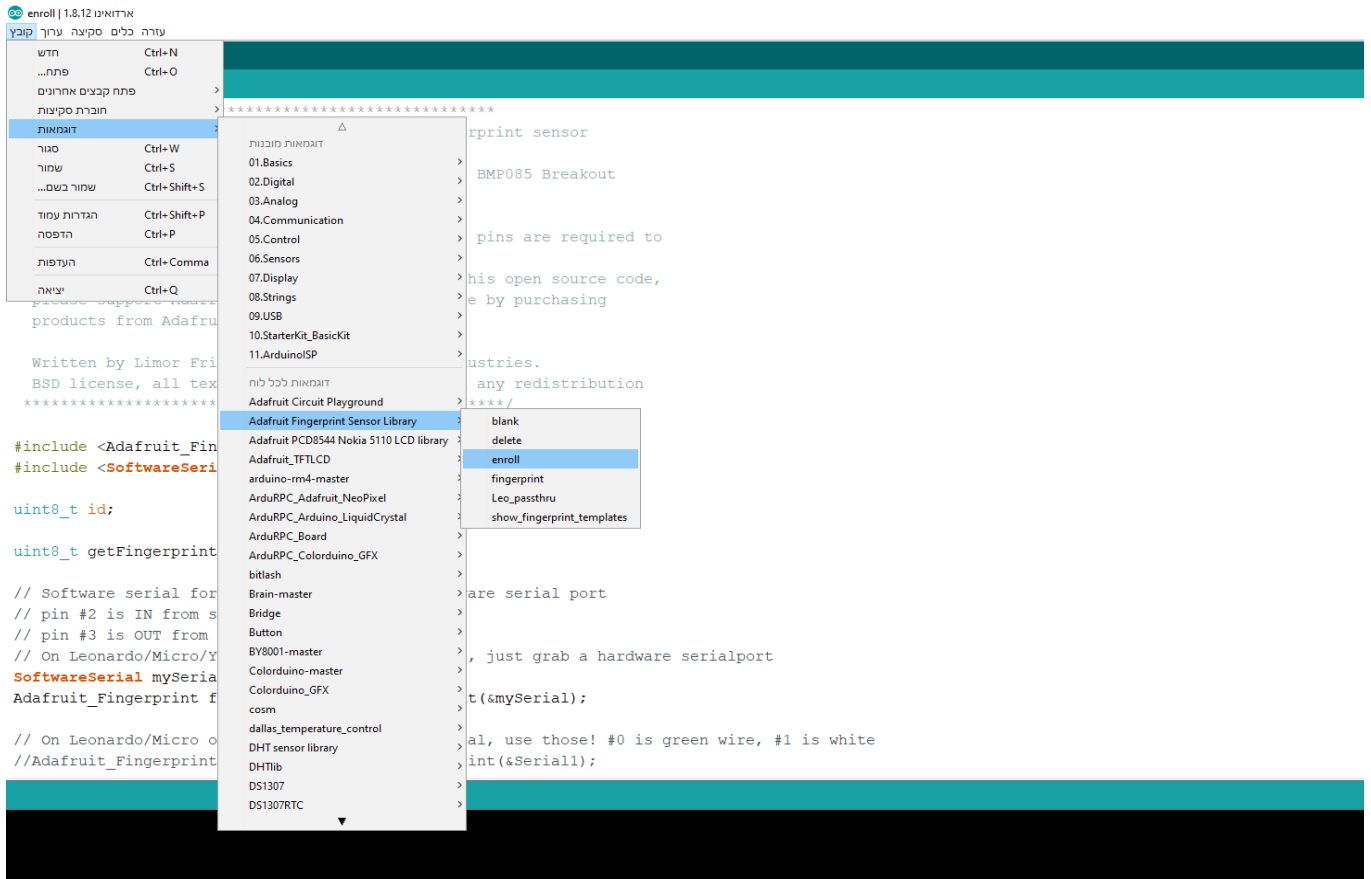
2.ת.12 טבלת הפקודות לפי סדר כרונולוגי של הקוד

code	identifier	Description	Code	Identifier	Description
01H	GenImg	Collect finger image	0DH	Empty	to empty the library
02H	Img2Tz	To generate character file from image	0EH	SetSysPara	To set system Parameter
03H	Match	Carry out precise matching of two templates;	0FH	ReadSysPara	To read system Parameter
04H	Serach	Search the finger library	12H	SetPwd	To set password
05H	RegModel	To combine character files and generate template	13H	VfyPwd	To verify password
06H	Store	To store template;	14H	GetRandomCode	to get random code
07H	LoadChar	to read/load template	15H	SetAdder	To set device address
08H	UpChar	to upload template	17H	Control	Port control
09H	DownChr	to download template	18H	WriteNotepad	to write note pad
0AH	UpImage	To upload image	19H	ReadNotepad	To read note pad
0BH	DownImage	To download image	1BH	HiSpeedSearch	Search the library fastly
0CH	DeletChar	to delete tempates	1DH	TempleteNum	To read finger template numbers

טבלת סיכום הפקודות לפי סדר כרונולוגי

13: תוכנה

13.א נסביר 2 תוכנות של חברת Adafruit. האחת נקראת הרשמה enroll והשנייה fingerprint המזהה את מספר טביעת האצבע שהכנסנו בהרשמה. לשם כך יש להכניס לספריית ה libraries של הארדואינו את הספרייה Adafruit-Fingerprint-Sensor-Library-master (ניתן למצוא ב google ולהוריד את קובץ ה zip לספרייה). שתי התוכניות נמצאות בספרייה וניתן להגיע אליהן בעזרת האיור הבא :



איור 20 : כיצד להפעיל את תוכנית ההרשמה enroll .

בתוכנת הארדואינו מקליקים עם העכבר על קובץ ובחלון שנפתח עוברים עם העכבר לדוגמאות ובחלון הנוסף שנפתח עוברים ל Adafruit Fingerprint Sensor Library ובחלון הנוסף שנפתח לוחצים עם העכבר על enroll ומקבלים את התוכנית הבאה :

13.ב תוכנית הרשמה enroll .

בתחילת העבודה עם מודול טביעת אצבע יש לבצע הרשמה, כלומר להכניס דוגמאות של טביעות אצבע שרוצים שהמערכת תכיר ותזהה . לשם כך מפעילים את תוכנת ה enroll . כאמור התוכנה נכתבה על ידי

Limor Fried/Ladyada for Adafruit Industries

התוכנה מתוכננת לעבוד עם ארדואינו אונו והיות ויש לו רק תקשורת טורית אחת משתמשים ב SoftwareSerial
שהיא תוכנה של תקשורת טורית. קישור להסבר התוכנה :

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

אם מחברים ארדואינו מגה שלו 4 תקשורות טוריות ניתן להשתמש בפקודות שידור וקליטה לתקשורת טורית שלא בתוכנה.

את הדק ה TXD של המודול מחברים להדק 2 של הארדואינו שימש כהדק קליטה ואת הדק ה RXD של המודול נחבר להדק 3 בארדואינו שימש כהדק השידור.

1.2.13

נצרף את שורות ההתחלה של קוד ההרשמה ופונקציית ה setup. נסביר כל קטע קוד.

```
/******
```

This is an example sketch for our optical Fingerprint sensor

Designed specifically to work with the Adafruit BMP085 Breakout

----> <http://www.adafruit.com/products/751>

These displays use TTL Serial to communicate, 2 pins are required to interface

Adafruit invests time and resources providing this open source code, please support Adafruit and open-source hardware by purchasing products from Adafruit!

Written by Limor Fried/Ladyada for Adafruit Industries.

BSD license, all text above must be included in any redistribution

```
*****/
```

```
#include <Adafruit_Fingerprint.h>
```

```
#include <SoftwareSerial.h>
```

```
uint8_t id;
```

```
// pin #2 is IN from sensor (GREEN wire)
```

```
// pin #3 is OUT from arduino (WHITE wire)
```

```
// On Leonardo/Micro/Yun, use pins 8 & 9. On Mega, just grab a hardware serialport
```

```
SoftwareSerial mySerial(2, 3);
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);

void setup()
{
  Serial.begin(9600);
  Serial.println("Adafruit Fingerprint sensor enrollment"); // הדפסה למסך הטורי
  finger.begin(57600);
  if (finger.verifyPassword())
  {
    Serial.println("Found fingerprint sensor!");
  } else {
    Serial.println("Did not find fingerprint sensor :(");
    while (1);
  }
}
```

8.1.ב.13

בהתחלה מנחים את הקומפיילר לכלול את קובץ הכותרת עבור ספריית טביעות האצבע ואת הקובץ *SoftwareSerial*. לאחר מכן מגדירים אובייקט *SoftwareSerial* ששמו *mySerial* עם פרמטרים שהם ההדקים 2 ו 3 (RX ו-TX בהתאמה). לאחר מכן מגדירים אובייקט טביעת אצבע בשם *finger* והפרמטר של הבנאי שלו הוא הכתובת של היציאה *SoftwareSerial* שבשורה מעליו שתהיה בשימוש על-ידי המודול.

8.1.ב.13 ב הפונקציה *setup()*

מאתחלת את פורט התקשורת הטורית למסך שישימש להצגת הודעות וגם לאיתור באגים, במידת הצורך. אובייקט האצבע מאותחל עם קצב שידור של 57600 שזוהי ברירת המחדל של המודול. לאחר מכן מתחיל פרוטוקול של שידור מהמיקרו למודול ותגובה של המודול חזרה לפי הפרוטוקול שהסברנו בסעיפים קודמים. מזמנים את המתודה *verifyPassword()* של האובייקט. אם קיים מודול טביעת אצבע היא מחזירה TRUE וההודעה "נמצא חיישן טביעת אצבע" מודפסת למסך הטורי המציינת הצלחה והמודול מוכן לקבלת פקודות. אם לא חזרה תשובה מהמודול המתודה מחזירה 0 ומודפסת ההודעה השנייה "לא נמצא חיישן טביעת אצבע" והתוכנית נתקעת בלולאה אין סופית.

2.ב.13 הפונקציה readnumber

זוהי פונקציה שקולטת מהמשתמש את מספר טביעת האצבע שרוצים לשמור בזיכרון המודול. קוראים לפונקציה מתוכנית ה- loop(). הפונקציה לא מקבלת ערכים והיא מחזירה את מספר טביעת האצבע שנקרא לו מזהה. המשתמש מכניס את המזהה בעזרת המסך הטורי. להלן הקוד :

```
uint8_t readnumber(void) {
    uint8_t num = 0;
    boolean validnum = false;
    while (1) {
        while (! Serial.available());
        char c = Serial.read();
        if (isdigit(c)) {
            num *= 10;
            num += c - '0';
            validnum = true;
        } else if (validnum) {
            return num;
        }
    }
}
```

בהתחלה מגדירים משתנה לא מסומן בגודל 8 ביטים ששמו num והוא מאוחלל ב 0. הוא יכיל את מספר המזהה שהמשתמש הקיש.

לאחר מכן נמצאים בלולאה שממתינה שהמשתמש יקיש על מקש כלשהו במסך התקשורת הטורית. בלולאה בודקים האם המשתמש הכניס נתון כלשהו ויש נתון מוכן בפורט הטורי בעזרת הפקודה :

```
while (! Serial.available());
```

אם המשתמש הקיש נתון קוראים אותו מהערוץ הטורי למשתנה התווי ששמו c. בודקים אם הוכנסה ספרות (ולא תווים) בעזרת הפונקציה isdigit(). אם כן היא הופכת את הספרות שהגיעו בקוד אסקי למספר (מכל ספרה באסקי היא מחסרת '0') ומחזירה את הערך המספרי שנמצא ב num לשורה שקראה לפונקציה.

3.ב.13 הפונקציה loop :

הפונקציה מתחילה בהדפסה הודעה למסך הטורי האומרת : "מוכן להרשמה של אצבע. הדפס את מספר המזהה – ID שאתה רוצה לשמור אצבע זו ... " .

לאחר מכן מזמנים את הפונקציה readnumber שהוסברה בסעיף 2.ב.13. הפונקציה החזירה את מספר המזהה הרצוי והוא נכנס למשתנה id.

אחר כך הפונקציה מדפיסה למסך: "רושמת מזהה: " ואת מספר המזהה.

לאחר מכן מזמנים את הפונקציה getFingerprintEnroll שרשומה מיד אחרי פונקציית ה loop. אם הפונקציה מחזירה ערך שאיננו 0 זה מראה שההרשמה הצליחה ואז חוזרים לתחילת פונקציית ה loop להרשמה של אצבע חדשה. אם הייתה בעיה כלשהי חוזרים לפונקציה getFingerprintEnroll ומנסים להצליח לרשום טביעת אצבע.

```
void loop()                // run over and over again
{
  Serial.println("Ready to enroll a fingerprint! Please Type in the ID # you want to save this finger as...");
  id = readnumber();
  Serial.print("Enrolling ID #");
  Serial.println(id);
  while (! getFingerprintEnroll() );
}
```

4.ב.13 הפונקציה getFingerprintEnroll :

כפי שכתבנו בסעיפים קודמים הפונקציה מבקשת מהמשתמש להקיש את המזהה – מספר טביעת האצבע, לוקחת דגימת אצבע ויוצרת קובץ תמונה ושומרת אותו ה buffer1. לוקחת פעם שנייה דגימה של אותה האצבע, יוצרת קובץ תמונה ושמה אותו ב buffer2. משתי הדגימות יש ליצור קובץ תווי ולשמור אותו בזיכרון המודול לפי מספר הדגימה – מזהה שהמשתמש הקיש. נרשום את הפונקציה עם הסברים בעברית ליד כל שורה.

```
uint8_t getFingerprintEnroll() // הפונקציה לא מקבלת ערכים ומחזירה ערך שלם לא מסומן של 8 ביט
{
  int p = -1;                // הגדרת משתנה מטיפוס שלם בשם p ומאתחלים אותו בערך מינוס אחד.
  Serial.print("Waiting for valid finger to enroll as #"); // הדפסה למסך ממתנינים להרשמה של אצבע מספר
  Serial.println(id); // מספר המזהה שהכנסנו בפונקציה readnumber בסעיף 2.ב.13
  while (p != FINGERPRINT_OK)
  // מבצעים את הלולאה עד ש p לא שווה 0 ( FINGERPRINT_OK ) הוא קבוע המוגדר כ 0. כל בעיה
  // תחזיר ערך השונה מ 0 ואז יוצאים מהלולאה
  {
    p = finger.getImage(); // מזמנים את הפונקציה שיוודעת לומר האם יש אצבע בחלון המודול
```

```
// אם יש אצבע בחלון ואם כן יוצרת קובץ תמונה . היא מחזירה ל p ערך . הערך נבדק בשורה הבאה
switch (p)
{
case FINGERPRINT_OK: // הערך המוחזר הוא 0 המראה שנוצר קובץ תמונה
    Serial.println("Image taken");
    break;
case FINGERPRINT_NOFINGER: // הערך הוא 2 שמראה שאין אצבע בחיישן
    Serial.println(".");
    break;
case FINGERPRINT_PACKETRECEIVEERR: // הערך 1 : שגיאה בקליטת חבילת התשובה
    Serial.println("Communication error");
    break;
case FINGERPRINT_IMAGEFAIL: // הערך 3 מראה שיש שגיאה ביצירת קובץ תמונה
    Serial.println("Imaging error");
    break;
default: // כל שאר הערכים של השגיאות
    Serial.println("Unknown error");
    break;
}
}

// OK success!
// לכאן מגיעים אם אין יש אצבע בחלון החיישן ואין שגיאה ונוצר קובץ תמונה
p = finger.image2Tz(1); // p ל נכנס ל
switch (p) { // בודקים מה הוחזר מהפונקציה
case FINGERPRINT_OK: // הערך המוחזר הוא 0 המראה שנוצר קובץ תווי
    Serial.println("Image converted");
    break;
case FINGERPRINT_IMAGEMESS: // מוחזר הערך 6 שאומר שהקובץ התווי לא ברור
    Serial.println("Image too messy");
    return p;
case FINGERPRINT_PACKETRECEIVEERR: // הערך 1 : שגיאה בקליטת חבילת התשובה
    Serial.println("Communication error");
    return p;
```

```
case FINGERPRINT_FEATUREFAIL:
    // הערך 7 מראה שלא נוצר קובץ תווי כי אין נקודות איפיון או קובץ תמונה לא מספיק
    Serial.println("Could not find fingerprint features");
    return p;
case FINGERPRINT_INVALIDIMAGE: // הערך 15H מראה שגיאה ביצירת קובץ תווי בגלל בעיה בקובץ התמונה
    Serial.println("Could not find fingerprint features"); // לא יכול למצא מאפייני טביעת אצבע
    return p;
default: // כל האפשרויות תשובה האחרות
    Serial.println("Unknown error");
    return p;
}
// ***** לכאן הגענו לאחר שנלקחה טביעת אצבע ראשונה ויש לבצע אותה פעולה פעם שנייה *****
Serial.println("Remove finger"); // הדפסה למסך הטורי לסלק את האצבע מחלון המודול
delay(2000);
p = 0;
while (p != FINGERPRINT_NOFINGER) { // ממתנים עד שהשתמש מסלק את האצבע מחלון המודול.
    p = finger.getImage();
}
Serial.print("ID "); Serial.println(id); // מדפיסים את מספר המזהה במסך הטורי פעם נוספת.
p = -1;
Serial.println("Place same finger again"); // הדפסה למסך לשים את אותה האצבע בחלון המודול.
/*
מתחיל תהליך דומה של לקיחת טביעת אצבע פעם שנייה. אין צורך להסביר את התהליך שוב.
*/
while (p != FINGERPRINT_OK) {
    p = finger.getImage();
    switch (p) {
    case FINGERPRINT_OK:
        Serial.println("Image taken");
        break;
    case FINGERPRINT_NOFINGER:
        Serial.print(".");
        break;
    case FINGERPRINT_PACKETRECEIVEERR:
```

```
Serial.println("Communication error");
break;
case FINGERPRINT_IMAGEFAIL:
Serial.println("Imaging error");
break;
default:
Serial.println("Unknown error");
break;
}
}

// OK success!

p = finger.image2Tz(2);
switch (p) {
case FINGERPRINT_OK:
Serial.println("Image converted");
break;
case FINGERPRINT_IMAGEMESS:
Serial.println("Image too messy");
return p;
case FINGERPRINT_PACKETRECIIEVEERR:
Serial.println("Communication error");
return p;
case FINGERPRINT_FEATUREFAIL:
Serial.println("Could not find fingerprint features");
return p;
case FINGERPRINT_INVALIDIMAGE:
Serial.println("Could not find fingerprint features");
return p;
default:
Serial.println("Unknown error");
return p;
}
```

```
/**
 * עד כאן לקחנו דגימות של אותה אצבע פעמיים. כעת ניצור קובץ תבנית אחד משילוב שני קבצי התו
 // OK converted!
Serial.print("Creating model for #"); Serial.println(id);

p = finger.createModel(); // פונקציה היוצרת מודל – קובץ טביעת אצבע – משני קבצי התווים
if (p == FINGERPRINT_OK) { // אם מוחזר הערך 0 זה מראה שנוצר קובץ אחד משני קבצי התו
  Serial.println("Prints matched!"); // הדפסה : טביעות האצבע תואמות.
} else if (p == FINGERPRINT_PACKETRECEIVEERR) { // הערך 1 : שגיאה בקליטת חבילת התשובה
  Serial.println("Communication error");
  return p;
}
else if (p == FINGERPRINT_ENROLLMISMATCH) { // הערך 0AH שאומר ש 2 קבצי התו לא תואמים
  Serial.println("Fingerprints did not match");
  return p;
} else { // עבור כל השגיאות האחרות שהתקבלו
  Serial.println("Unknown error");
  return p;
}
}

/** עכשיו נשמור את קובץ התבנית בזיכרון ה FLASH של המודול
Serial.print("ID "); Serial.println(id);
p = finger.storeModel(id); // זימון פונקציה לשמירת הקובץ . שולחים לה את מספר המזהה
if (p == FINGERPRINT_OK) { // אם הערך שחזר הוא 0 , כלומר פעולת השמירה הצליחה
  Serial.println("Stored!"); // הדפסה למסך הטורי : "נשמר"
}
else if (p == FINGERPRINT_PACKETRECEIVEERR) { // הערך 1 : שגיאה בקליטת חבילת התשובה
  Serial.println("Communication error");
  return p;
} else if (p == FINGERPRINT_BADLOCATION) {
  /*
  הערך 0bH מראה שלא ניתן לשמור את טביעת האצבע בזיכרון בכתובת של מספר המזהה, כלומר עברנו
  את הכתובת האחרונה בזיכרון ה FLASH .
  */
}
```

```
*/  
Serial.println("Could not store in that location");  
return p;  
}  
else if (p == FINGERPRINT_FLASHERR) {  
/*
```

הערך 18H מראה שגיאה בכתיבה לזיכרון ה FLASH

```
*/  
Serial.println("Error writing to flash");  
return p;  
} else {  
Serial.println("Unknown error");  
return p;  
}  
}
```

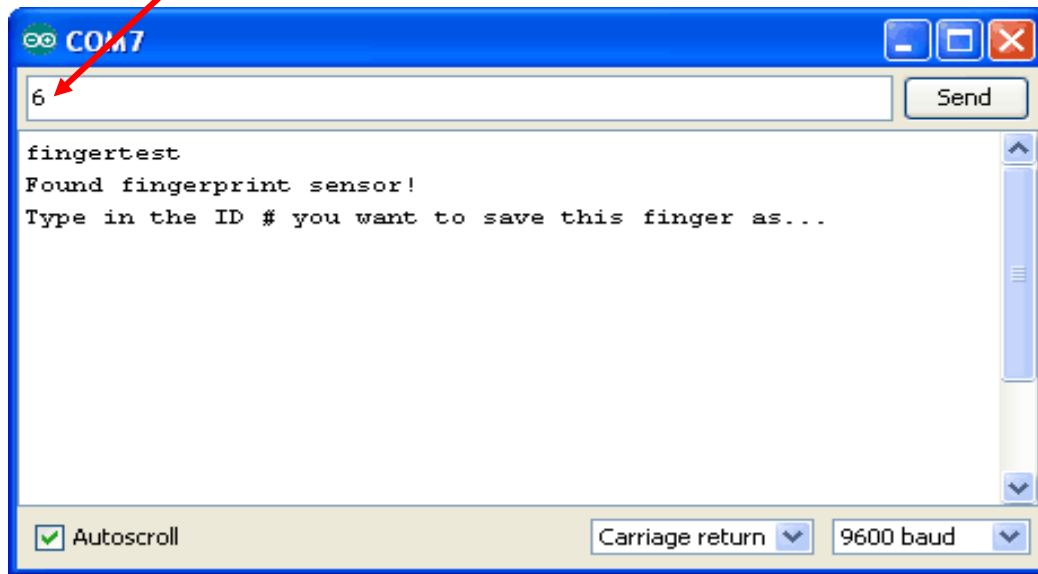
6.2.13

לאחר העלאת הקוד, פתח את הצג הטורי בקצב השידור 9600 ועקוב אחרי ההוראות במסך הטורי.

בעת פתיחת הצג הטורי, נקבל את ההודעות הבאות :

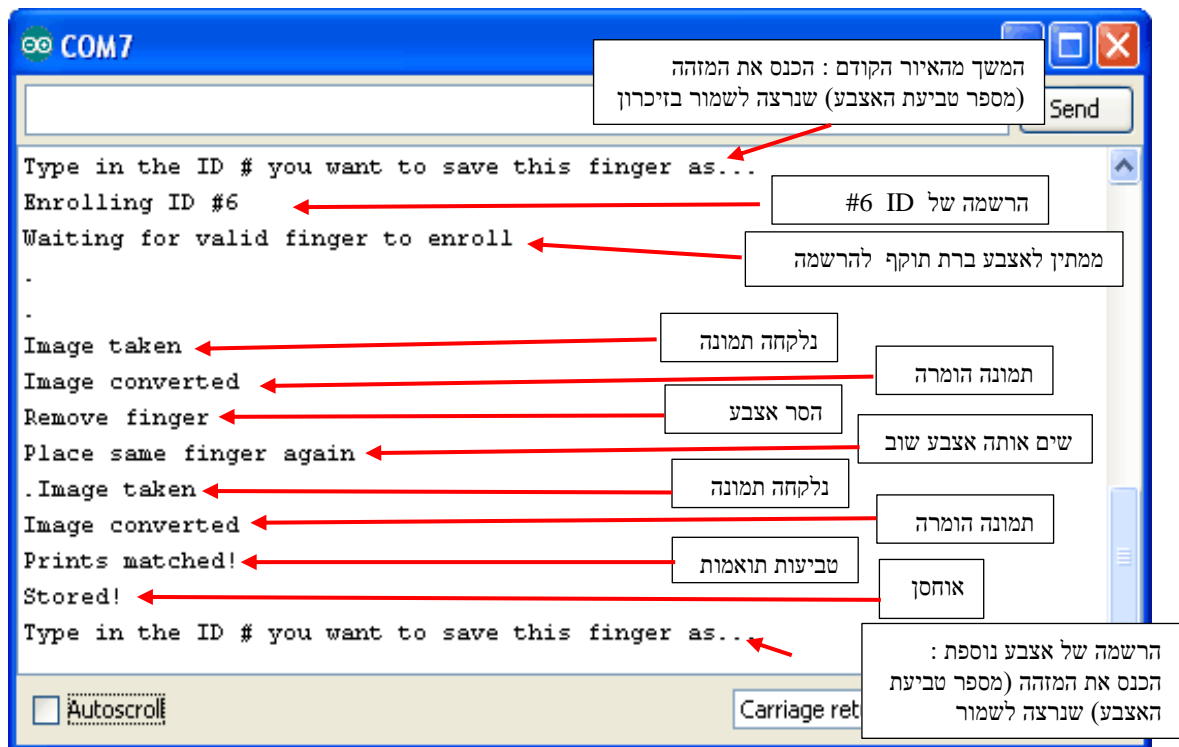
- `fingerstest` – בדיקת אצבע.
- במידה וחיברנו את מודול החיישן נכון נקבל את ההודעה : `Found fingerprint sensor` – נמצא חיישן טביעת אצבע !
- הכנס את המזהה (מספר טביעת האצבע) שנרצה לשמור בזיכרון מודול טביעת האצבע.
- את המספר נרשום בחלון כמו באיור הבא (בדוגמה שלנו המזהה הוא 6) . מביאים בעזרת העכבר את הסמן לחלון מקלידים את המספר הרצוי ולוחצים על `send` (בצד ימין במסך)

כאן נרשום את מספר המזהה



איור 21.א : מסך הרשמה enroll

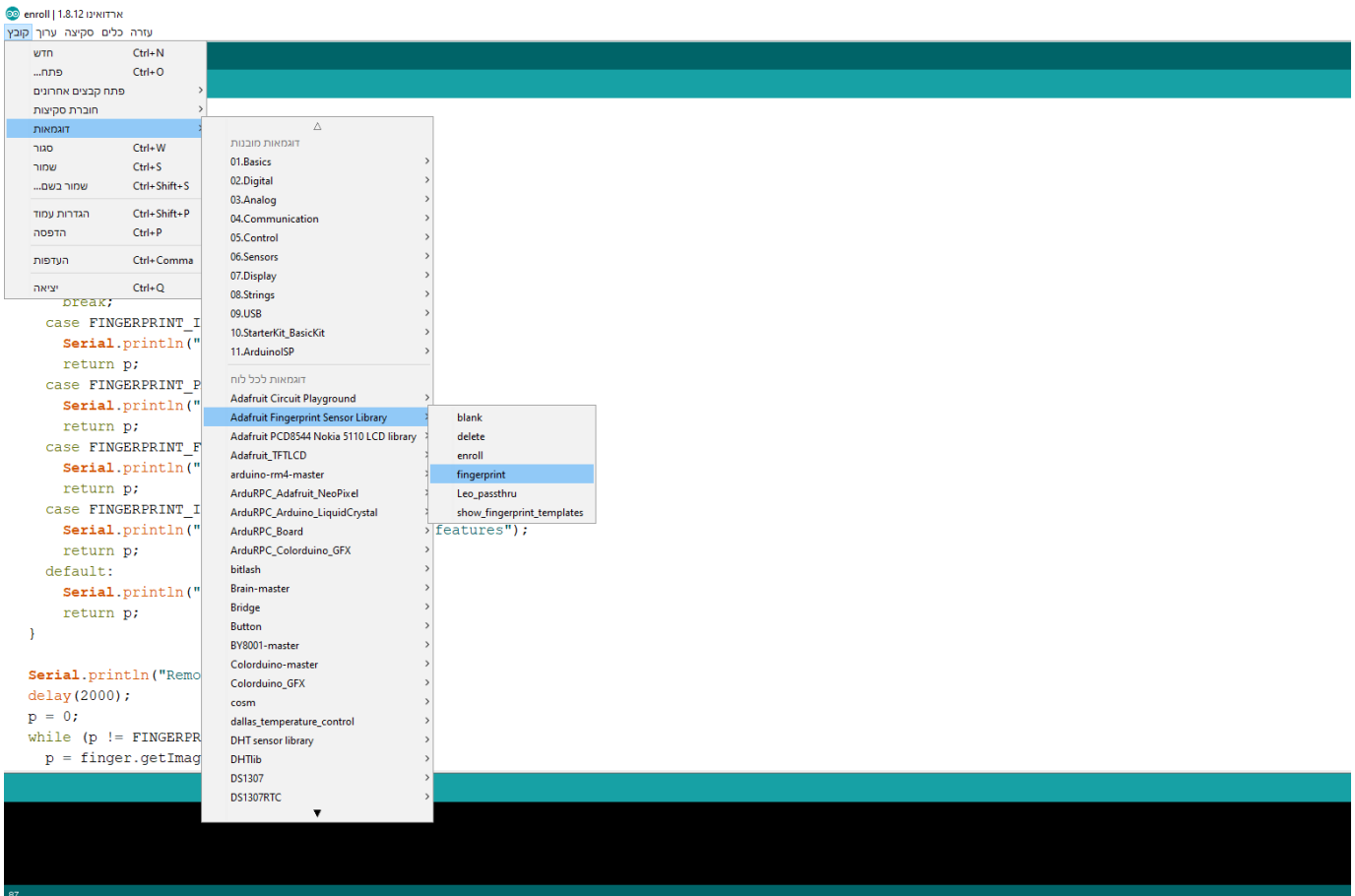
לאחר מכן עוברים להמשך ההודעות שנראות באיור הבא:



איור 21.ב : המשך מסך רישום

8.ב.13 תוכנית זיהוי טביעת האצבע fingerprint

תפקיד התוכנית לזהות טביעת אצבע הקיימת במאגר הזיכרון של המודול ולהדפיס למסך הטורי את המזהה (ID) של טביעת האצבע.
 (ניעזר באיור הבא כדי לקבל את תוכנית זיהוי טביעת האצבע הנקראת fingerprint .



איור 22 : הדרך להגיע לפונקציית טביעת אצבע fingerprint

כאשר נלחץ על תוכנית fingerprint נקבל את התוכנית הבאה שגם היא נכתבה על ידי Limor Fried/Ladyada מחברת Adafruit :

9.ב.13 הסבר התוכנית

#include <Adafruit_Fingerprint.h> // הנחייה לקומפיילר לכלול קובץ כותר של טביעת אצבע

#include <SoftwareSerial.h> // הנחייה לקומפיילר לכלול את קובץ הכותר של התוכנה של תקשורת טורית

```
int getFingerprintIDez(); // אין צורך בשורה זו. השורה מזהירה על פונקציה שנפגוש בהמשך
```

```
// pin #2 is IN from sensor (GREEN wire) Adafruit של BMP085 הצבע של החוט בחיישן  
// pin #3 is OUT from arduino (WHITE wire) Adafruit של BMP085 הצבע של החוט בחיישן  
SoftwareSerial mySerial(2, 3); // הגדרת אובייקט תוכנה של תקשורת טורית והדקים שאליו מתחבר המודול  
// mySerial הגדרת אובייקט של טביעת אצבע המקבל את הכתובת של האובייקט  
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);
```

```
/*
```

```
void setup()
```

```
{
```

```
Serial.begin(9600); // אתחול התקשורת הטורית עם המסך לקצב 9600 ביטים בשנייה
```

```
Serial.println("Adafruit finger detect test"); // הדפסה למסך: "בדיקת גילוי אצבע של Adafruit"
```

```
// קביעת קצב התקשורת הטורית עם המודול
```

```
finger.begin(57600);
```

```
/*
```

```
בשורות הבאות מזמנים את המתודה verifyPassword() של האובייקט. אם קיים מודול טביעת אצבע היא מחזירה TRUE וההודעה "נמצא חיישן טביעת אצבע" מודפסת למסך הטורי המציינת הצלחה והמודול מוכן לקבלת פקודות.
```

```
אם לא חזרה תשובה מהמודול המתודה מחזירה 0 ומודפסת ההודעה השנייה "לא נמצא חיישן טביעת אצבע" והתוכנית נתקעת בלולאה אין סופית.
```

```
*/
```

```
if (finger.verifyPassword()) {
```

```
Serial.println("Found fingerprint sensor!");
```

```
} else {
```

```
Serial.println("Did not find fingerprint sensor :(");
```

```
while (1);
```

```
}
```

```
Serial.println("Waiting for valid finger...");
```

```
}
```

```

/***** loop פונקציית ה *****/
void loop() // run over and over again
{
  getFingerprintIDez(); // (נמצאת הרבה שורות למטה עם כוכביות מודגשות)
  delay(50); //don't need to run this at full speed. השהייה קלה.
}

uint8_t getFingerprintID() // הגדרה של פונקציה שמחזירה ערך שלם חיובי בן 8 ביט ולא מקבלת נתונים
{
  uint8_t p = finger.getImage();
  // זימון פונקציה שיוצרת קובץ תמונה. אם נוצר קובץ מוחזר 0. אם יש בעיה מוחזר ערך שמציין את סוג השגיאה
  // את ההסברים החל משורה זו ועד השורה המסומנת בכוכביות למטה ניתן למצא בתוכנית ההרשמה
  switch (p) {
    case FINGERPRINT_OK:
      Serial.println("Image taken");
      break;
    case FINGERPRINT_NOFINGER:
      Serial.println("No finger detected");
      return p;
    case FINGERPRINT_PACKETRECEIVEERR:
      Serial.println("Communication error");
      return p;
    case FINGERPRINT_IMAGEFAIL:
      Serial.println("Imaging error");
      return p;
    default:
      Serial.println("Unknown error");
      return p;
  }

  // OK success!

  p = finger.image2Tz();
  switch (p) {

```

```
case FINGERPRINT_OK:
    Serial.println("Image converted");
    break;
case FINGERPRINT_IMAGEMESS:
    Serial.println("Image too messy");
    return p;
case FINGERPRINT_PACKETRECEIVEERR:
    Serial.println("Communication error");
    return p;
case FINGERPRINT_FEATUREFAIL:
    Serial.println("Could not find fingerprint features");
    return p;
case FINGERPRINT_INVALIDIMAGE:
    Serial.println("Could not find fingerprint features");
    return p;
default:
    Serial.println("Unknown error");
    return p;
}

/*****/

// OK converted!
p = finger.fingerFastSearch();
// זימון פונקציה המחפשת טביעת אצבע תואמת
// אם לא נמצאה התאמה או כל שגיאה אחרת יש פקודת חזרה לשורה delay בפונקציית ה loop
if (p == FINGERPRINT_OK) { // 0 הערך
    Serial.println("Found a print match!");
} else if (p == FINGERPRINT_PACKETRECEIVEERR) { // טעות בקליטת החבילה
    Serial.println("Communication error");
    return p;
} else if (p == FINGERPRINT_NOTFOUND) { // לא נמצאה התאמה
    Serial.println("Did not find a match");
    return p;
} else {
    Serial.println("Unknown error");
}
```

```

return p;
}
// found a match! אם הגענו לכאן אז נמצאה התאמה ומדפיסים למסך שנמצא מזהה ואת המספר שלו
/*
כמו כן מודפס למסך גם מידת הביטחון של ההתאמה שמציין את כמות המאפיינים התואמים בין הקובץ במודול
והקובץ של האצבע שבחלון. ככל שהמספר גדול יותר ההתאמה טובה יותר.
*/
Serial.print("Found ID #"); Serial.print(finger.fingerID);
Serial.print(" with confidence of "); Serial.println(finger.confidence);
}
/*****/

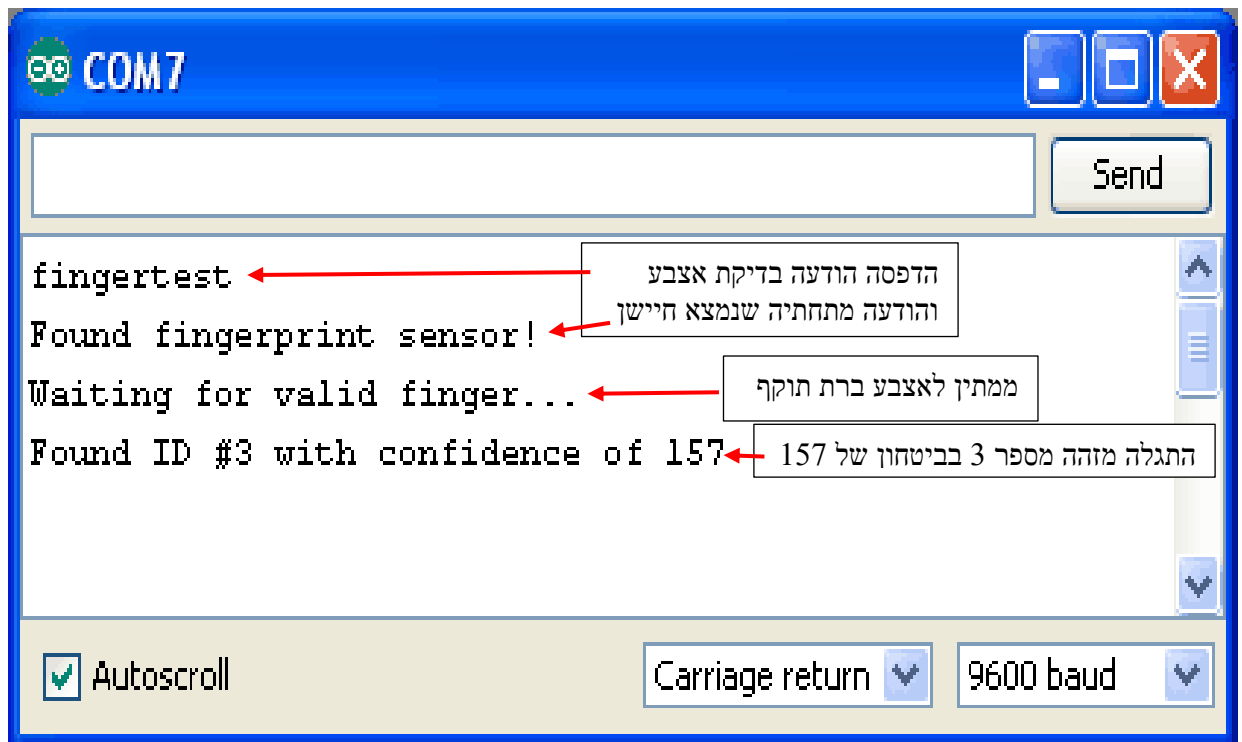
// returns -1 if failed, otherwise returns ID # המזהה או את מספר המזהה
int getFingerprintIDez() { // לא מסומן
/*
זימון הפונקציה finger.getImage ששייכת לספריית טביעת האצבע והערך שחוזר ממנה נכנס למשתנה p .
הפונקציה בודקת האם יש אצבע בגלאי ואם כן היא יוצרת קובץ תמונה. אם נוצר קובץ מוחזר 0 וממשיכים
לשורה הבאה. אם מוחזר ערך שונה מ 0 מחזירים מינוס 1 לשורה שקראה לפונקציה ( כמה עמודים למעלה).
uint8_t p = finger.getImage();
if (p != FINGERPRINT_OK) return -1;
// אם נוצר קובץ תמונה ...
p = finger.image2Tz();
/*
זימון פונקציה הממירה את קובץ התמונה לקובץ תווי. הפונקציה תחזיר 0 אם הצליחה ליצור קובץ תווי או
ערך שונה מ 0 המציין את מספר השגיאה ואז מחזירים מינוס 1 ל לשורה שקראה לפונקציה ( כמה עמודים למעלה).
if (p != FINGERPRINT_OK) return -1;

פונקציה למציאת טביעת אצבע תואמת בספרייה של המודול.
p = finger.fingerFastSearch(); //
if (p != FINGERPRINT_OK) return -1;
// found a match! נמצאה התאמה ואז מדפיסים את מספר המזהה ואת רמת הביטחון
Serial.print("Found ID #"); Serial.print(finger.fingerID);
Serial.print(" with confidence of "); Serial.println(finger.confidence);
return finger.fingerID;
}

```

10.ב.13 הרצת התוכנית

הפעלת התוכנית ופתיחת המסך הטורי תיתן לנו את האיור הבא :



איור 23 : המסך הטורי המתקבל בהרצת התוכנית

ולסיום :

כאמור ישנם סוגים שונים של מודולים של טביעת אצבע וקיימת סכנה שחיבור לא נכון של ההדקים יביא לגרימת נזק בלתי הפיך למודול. ניתן מספר איורים למודולים שונים .

באחריות המשתמש לוודא מהו המודול בו הוא משתמש וחיבור ההדקים המדויק שלו.

1. Introduction to Biometric Recognition Technologies and Applications
Dr. Marios Savvides
2. INTERFACING THE FPM10A WITH AN ARDUINO (by Brian Ejike)
3. ANSI-INCITS 378, Fingerprint Minutiae Format for Data Interchange. American National Standard (2004)
4. Alonso-Fernandez, F., Fierrez, J., Ortega-Garcia, J., Gonzalez-Rodriguez, J., Fronthaler, H., Kollreider, K., Bigun, J.: A comparative study of fingerprint image quality estimation methods. IEEE Trans. on Information Forensics and Security 2(4), 734–743 (2007)
5. Alonso-Fernandez, F., Fierrez, J., Ortega-Garcia, J., Fronthaler, H., Kollreider, K., Bigun, J.: Combining Multiple Matchers for fingerprint verification: a case study in BioSecure Network of Excellence. Annals of Telecommunications, Multimodal Biometrics, Eds. B. Dorizzi and C. Garcia-Mateo, Vol. 62, (2007)
6. <http://www.adafruit.com/datasheets/ZFM%20user%20manualV15.pdf>.
7. <https://www.rhydolabz.com/documents/finger-print-module.pdf>