

דוח מסכם לפרויקט בקורס אינטרנט של הדברים סמסטר אביב 2021  
ליאור דגו 034657528  
זאב ימפולסקי 312236037

CoolFridge – cool up you'r fridge!

## הגדרת הצורך היזמי

המוצר שלנו CoolFridge נוצר על מנת לתת מענה לחוסרים הקיימים בנישת המקררים הפשוטים, לתת ערך מוסף לכל משפחה שבבעלותה מקרר ואינה נמצאת רוב היום בבית. המקרה שבו מיהרנו בבוקר לעבודה ושכחנו לסגור את המקרר עד הסוף מוכר לכולנו, בערב כשאנחנו חוזרים לצערנו כל המקרר "מזיע" מבפנים והאוכל כבר לא קר וספק שחלקו מקולקל. המוצר שלנו נועד לספק פתרון בדיוק לבעיות ומקרים מהסוג הזה. הפתרון המוצע שלנו שבעזרתו ננסה לתת מענה לבעיות אלה הינו:

- מדידת הטמפרטורה והלחות בתוך המקרר, טמפרטורת הסביבה והצגתם על מסך LED קטן שיותקן על הדלת.
- התראה על כך שדלת המקרר פתוחה בצורה חזותית (מנורת LED אדומה) וקולית (buzzer קטן).
- התראות חכמות שנשלחות אל מייל הלקוח במקרים הבאים:
  - המקרר נשאר פתוח מעל 3 דקות.
  - הטמפרטורה הפנימית גבוהה מידי.
  - לא התקבלו מדידות מהמערכת במהלך 24 השעות האחרונות.
  - לחיצה על כפתור על גבי המערכת שיוזם שליחת מייל עם רשימת קניות.

המוצר שאנחנו מציעים הוא בעצם פתרון נוח, קטן וקל להתקנה על גבי כל מקרר. הוא נותן מענה למקרים רבים כמו הפסקת חשמל פתאומית במקרר, בעיה שגורמת לטמפרטורה גבוהה מידי במקרר וכמובן מקרים בהם הדלת נשארת פתוחה שלא בכוונה לזמן רב מידי.

## כפירוט הפתרון המוצע

בהמשך לתיאור הראשוני של המערכת בחלק הקודם כעת נפרט על סוגי ההתראות והמדידות שהמערכת שלנו מבצעת ומודדת על מנת לתת תמונה מלאה של הפתרון, פעילות המערכת והחיישנים השונים בתוכה.

### רכיבים

כאמור לעיל נשתמש בחיישנים ורכיבים חומרתיים שונים על מנת לבצע את המדידות הרלוונטיות ושליחת ההתראות המתאימות למדידות. הפרויקט שלנו היה מורכב מחלק חובה שכלל מדידת טמפרטורה במקרר, טמפרטורה מחוץ למקרר ומצב הדלת, בנוסף היה חלק בחירה שבו הוספנו מדידת לחות במקרר, התראה חזותית וקולית על מצב הדלת ומסך LED שהציג את המידע הנ"ל. כעת נפרט על חיישני ורכיבי המערכת:

### חלק חובה

- ESP32 Devkit v1 : רכיב זה הוא בעצם "המוח" של המערכת שלנו, זהו מיקרו-בקר אשר עליו ניתן "לצרוב" קוד ובכך לגרום לו לבצע את מה שנבקש. על הרכיב הזה צרבנו את הקוד שביצע את הדיגום למדידות השונות, שליחת אותות לרכיבים מקומיים כמו מנורת ה LED והמסך, וכמובן שליחת הנתונים לענן.
- TMP36 : חיישן טמפרטורה בסיסי שהשתמשנו בו על מנת למדוד את הטמפרטורה מחוץ למקרר. החיישן היה מותקן מחוץ למקרר, ה esp32 היה דוגם כל פרק זמן מסוים את הטמפרטורה וממצע אותה ואת המיצוע הזה היה שולח לענן - נרחיב על כך בהמשך.
- BME280 למדידת טמפרטורה : חיישן טמפרטורה יותר מדויק ובעל יכולות מדידה נוספות כמו לחות ולחץ ברומטרי. החיישן הותקן בתוך המקרר על מנת לבצע את המדידות, השתמשנו בחיישן זה על מנת למדוד את הטמפרטורה הפנימית במקרר מיצוע הנתונים ושליחתן לענן.
- לחצן 2 Pin PCB Momentary Push Button : לחצן שהיה מותקן בתוך המקרר, תפקידו של הלחצן היה לאפשר ללקוח לשלוח התראה מיידית למייל שלו כשנדרש לעשות קניות. הלחצן פשוט והיה מחובר ל board כך שה ESP יוכל לדגום ממנו, ברגע שהלחצן נלחץ לשניה ה ESP דגם לחיצה זאת ובהתאם שלח התראה לענן - נרחיב בהמשך על אופן הפעולה המדויק.
- Photo Resistor : חיישן אור שהיה מותקן בתוך המקרר ותפקידו היה למדוד את כמות האור, ברגע שהחיישן מזהה אור שהגיע מהמקרר ומחוצה לו התנגדותו של החיישן עולה, ה esp דגם את המתח בחיישן וברגע שהוא גבוה מסף מסוים, כלומר הדלת פתוחה, ה ESP שולח התראה לענן.

ברגע שהדלת נסגרת החיישן אינו חשוף לאור ובעקבות כך התנגדותו יורדת , ה ESP דוגם כי המתח נמוך , ובעקבות כך שולח התראה לענן.  
מלבד ההתראות לענן, בעת פתיחת הדלת וזיהוי שהתנגדות החיישן גבוהה ה ESP שולח אות buzzer ולמנורת ה LED האדומה כאינדיקציה חזותית שהדלת פתוחה – נרחיב השמך.

#### רכיבי בחירה:

- BME280 למדידת לחות: אנו משתמשים באותו חיישן BME280 מחלק החובה, אך כעת נמדוד בעזרתו את הלחות במקרר, החיישן מותקן בתוך המקרר ובכך מודד את הלחות בתוך המקרר. ה esp דוגם את הלחות מהחיישן כל פרק זמן, ממצע את המדידות האלה ולבסוף שולח את המדידה הממוצעת של הלחות לענן – נרחיב המשך.
- צג OLED מדגם SSD1306 : הצג מחובר ל esp32 , ותפקידו להציג את הנתונים הנשחים לענן ומצב הדלת , כלומר ה ESP שולח את הנתונים הבאים להדפסה על המסך: טמפרטורה פנימית , לחות , טמפרטורה חיצונית ומצב הדלת (1 פתוח 0 סגור).  
הנתונים הנ"ל מוצגים על המסך למשך 2 דקות , ופעם בשתי דקות הנתונים מתעדכנים והנתונים המעודכנים מודפסים על הצג. במידה ונפתחה או נסגרה הדלת, ה ESP מעדכן את כל הנתונים על גבי הצג באותו הרגע ואינו מעדכן שוב את ההדפסה למשך 2 דקות או עד אשר יש שוב שינוי במצב הדלת – נרחיב בהמשך על פונקציות ההדפסה.
- KY-006 Passive Buzzer : באזר פסיבי, מותקן בתוך המקרר, אופן פעולתו הוא שברגע שה esp מזהה שהדלת נפתחה ע"י דגימה של ה Photo resistor כפי שהוסבר קודם , ה esp שולח אות ל buzzer על מנת שישמיע צפצוף מתמשך כל עוד הדלת פתוחה . ברגע שהדלת נסגרת, ה esp שוב מזהה שינוי במתח ע"י דגימת photo resistor ומשנה את האות ל buzzer כך שלא ישמיע יותר צפצוף. מטרת ה buzzer הוא לתת אינדיקציה קולית למצב הדלת במקרים בהם אין למקרר צפצפה מובנת.
- Red LED Light – מנורת לד פשוטה בצבע אדום שמחוברת במקביל אל ה buzzer, מכיוון ששניהם חלק מאותו "פיצ'ר" , אופן הפעולה וההפעלה של המנורה זהה לאופן הפעולה של ה buzzer , כלומר ה ESP דוגם את photoresistor ובהתאם למדידת המתח מזהה האם הדלת פתוחה או לא , במידה והדלת פתוחה ה ESP שולח אות ל Buzzer לצפצף וחשמל למנורה על מנת שתאיר. במידה והדלת סגורה ה ESP מכבה את הצפצפה ומחליש את האות החשמלי למנורה על מנת שתכבה.

#### תדירות דגימות , החלקת הרעשים, הצגת המידע וההתראות ושליחת המידע לענן

##### תדירות הדגימה

תדירות הדגימה של ה esp היא אחת לשנייה, 1000 מילי-שניות, עבור כל החיישנים המחוברים , כלומר אחת לשנייה ה ESP דגם את החיישנים :

- BME280 עבור הטמפרטורה הפנימית במקרר והלחות במקרר.
- TMP36 עבור הטמפרטורה החיצונית בסביבת המקרר.
- PhotoResistor עבור התנגדותו של הנגד עבור מצב הדלת.
- PCB Pushbutton עבור התנגדותו של הכפתור והאם נסגר המעגל ברגע לחיצה על הכפתור.

#### החלקת הרעשים עבור שליחת הנתונים לענן אחת ל 5 דקות

תהליך החלקת הרעשים של מדידות מהחיישנים התבצע על ידי מיצוע המדידות במשך 5 דקות בין שליחת הנתונים לענן.

לפי דרישות הפרויקט המידע הנמדד ממוצע לטובת החלקת הרעשים ונשלח לענן אחת ל 5 דקות, במהלך חמשת הדקות האלה ה ESP מבצע את החלקת הרעשים בצורה ההבאה:

1. איפוס מונה זמן שסופר את כמות המילי-שניות שעברו מהשליחה האחרונה.
2. מדידה של הנתונים הנ"ל מהחיישנים המתאימים אחת לשנייה.
3. שמירת כל מדידה והוספתה לסכום רץ עבור על מדידה בנפרד.

4. שמירת כמות המדידות שבוצעו בחמשת הדקות , כלומר עבור כל מדידה נוסף 1 ל counter נפרד לכל מדידה שמבוצעת, נזכיר טמפרטורה פנימית וחיצונית ולחות , כולם אחת נדגים אחת לשניה.
5. בדיקה האם מונה הזמן הגיע ל 300,000 מילי-שניות , כלומר 5 דקות , במידה וכן:
  - a. חישוב ממוצע לכל מדידה בנפרד על ידי חלקות הסכום הנצבר עבורה בכמות המדידות שבוצעו עבורה .
  - b. שליחת הממוצע של כל מדידה לענן.

על ידי ביצוע ממוצע זה לכל אחת מהמדידות הנדרשות בפרויקט החלקנו את הרעשים , כלומר החלקנו את המדידות החריגות שבוצעו על ידי החיישנים.

#### מצב הדלת ושליחת התראות מתאימות בהתאם למצב הדלת

כפי שצינו קודם חיישן האור נדגם אחת לשנייה עבור התנגדותו , במידה והתנגדותו גבוהה משמע החיישן קולט הרבה אור מה שאומר לנו שהדלת פתוחה , במידה והתנגדותו קטנה או אפסית אז החיישן אינו קולט אור משמע הדלת סגורה.

בגלל שמדובר ב"מצב הדלת" בחרנו להגדיר בקוד שתי משתנים בינארי שישמור את "מצב הדלת", בשניהם 1 עבור דלת פתוחה ו 0 עבור דלת סגורה.

- הראשון שומר את מצב הדלת הנוכחי , כלומר כל דגימה של ה photo resistor על ידי ה ESP מתקבלת מדידה מהחיישן , המדידה הזאת מתורגמת לערך בינארי שנשמר במשתנה "מצב הדלת הנוכחי".
  - השני שומר את מצב הדלת האחרון , כמו קודם 1 עבור פתוחה ו 0 עבורה סגורה. עדכון משתנה זה מתבצע רק כאשר "מצב הדלת הנוכחי" שונה ממצב הדלת האחרון , במקרה כזה נעדכן את "מצב הדלת האחרון" אותו ערך כמו "מצב הדלת הנוכחי".
- על ידי שילוב שתי משתנים אלה ""מצב דלת נוכחי" ו"מצב דלת אחרון", אנחנו יכולים להבדיל האם הדלת הייתה פתוחה ונסגרה , הייתה סגורה ונפתחה או הייתה פתוחה ונשארה פתוחה או הייתה סגורה ונשארה סגורה.

#### שליחת הנתונים לענן ברגע שינוי מצב הדלת:

1. דגימה של Photoresistor אחת לשנייה , ועדכון "מצב הדלת הנוכחי" בהתאם לגובה המתח הנמדד.
2. השוואה של "מצב הדלת הנוכחי" אל מול "מצב הדלת האחרון". במידה והם שונים , אזי היה שינוי במצב הדלת, כלומר או שהדלת הייתה סגורה ונפתחה או לחילופין , הייתה פתוחה ונסגרה.
3. במידה והיה שינוי במצב הדלת ה ESP שולח באותו הרגע את מצב הדלת החדש לענן , במידה ולא היה שינוי נחזור על 1,2 עד אשר יהיה שינוי כזה.

#### שליחת התראה במידה והדלת פתוחה במשך 3 דקות:

על מנת להוציא לפעול התראה זאת הוספנו משתנה אשר סופר את כמות הזמן ברזולוציות מילי-שניות את כמות הזמן שהדלת הייתה פתוחה בו, בנוסף כפי שהסברנו קודם הגדרנו בקוד שתי מצבי דלת "מצב דלת נוכחי" ו"מצב דלת אחרון" אנחנו נשתמש בשתי משתנים אלה על מנת לבדוק האם הדלת פתוחה בפרק זמן העולה על 3 דקות כך:

1. דגימה של ה Photoresistor אחת לשנייה , ועדכון "מצב הדלת הנוכחי" בהתאם לגובה ההתנגדות.
2. בדיקה האם "מצב הדלת הנוכחי" שונה מ"מצב הדלת האחרון" , במידה וכן כלומר המצבים שונים:
  - a. נאפס את משתנה שסופר את זמן שבו הדלת נשארה פתוחה , ונבצע שינוי בהדפסת המסך – נרחיב בהמשך.
3. נבדוק האם "מצב הדלת הנוכחי" זהה ל"מצב הדלת האחרון" והאם "מצב הדלת הנוכחי" הוא פתוח :
  - a. במידה וכן , נבדוק האם עברו 180,000 מילי-שניות(3 דקות) מרגע איפוס משתנה הזמן שלנו, נזכיר שבכל שינוי "מצב דלת" ממצב אחרון למצב נוכחי אנחנו מאפסים משתנה זה , לכן כאשר מצב הדלת לא משתנה אנחנו לא מאפסים משתנה זה.
    - i. במידה ועברו 180,000 מילי-שניות , כלומר הדלת פתוחה כבר 3 דקות ולכן נשלח התראה מתאימה לענן – נרחיב בהמשך על סוג ההתראה המדויק.
    - ii. נאפס את מונה הזמן הנ"ל לאחר שליחת ההתראה המיוחדת לענן.

### צפצוף ב buzzer ותאורת מנורת ה LED האדומה:

כפי שצינו קודם, הגדרנו שתי משתני "מצב דלת" נשתמש בהם על מנת לתת אינדיקציה קולית וויזואלית על מצב הדלת בצורה הבאה:

1. דגימת ה Photoresistor אחת לשנייה, ועדכון "מצב הדלת הנוכחי" במידה לגובה ההתנגדות.
2. בדיקה של "מצב הדלת הנוכחי"
  - a. במידה והדלת פתוחה "במצב הנוכחי" (ערך 1) נשמיע צפצוף מתמשך ב Buzzer ונדליק את מנורת ה LED האדומה, בנוסף נעדכן את התצוגה במסך – נפרט בהמשך.
  - b. במידה והדלת סגורה "במצב הנוכחי" (ערך 0) נכבה את הצפצוף ב Buzzer ונכבה את מנורת ה LED האדומה.

### הצגת הנתונים על גבי מסך ה SSD1306 :

נשתמש במשתנים שצינו קודם עבור הטמפרטורה הפנימית, הלחות, הטמפרטורה החיצונית, ו"מצב הדלת", בנוסף נגדיר משתנה זמן עבור המסך שסופר במילי-שניות מרגע עדכון האחרון של ההדפסה במסך. בתחילת הקוד נאפס את משתנה הזמן של המסך ל 0 מילי-שניות, נעדכן את הנתונים שמוצגים על המסך לפי השלבים הבאים:

1. נבדוק האם עברו 120,000 מילי-שניות (2 דקות) מרגע ההדפסה האחרונה על המסך, במידה וכן:
  - a. נמדוד את: טמפרטורה פנימית וחיצונית, לחות במקרר ומצב הדלת ונדפיס אותם על המסך.
  - b. לאחר ההדפסה נאפס את משתנה הזמן ל 0 מילי-שניות.
2. במידה והיה שינוי "במצב הדלת הנוכחי" כפי שתיארו קודם בחלק שליחת ההתראה לענן על מצב הדלת וההתראה ב buzzer ומנורת ה LED, נבצע את שלבים 1.a ו 1.b.
- a. במידה והדלת פתוחה במשך 3 דקות נעדכן את ההדפסה למסך רק כל שתי דקות.

### שליחת התראה לענן לפי מצב הכפתור:

נדגום את הכפתור כל שניה כפי שאנחנו עושים עבור שאר החיישנים, נשמור את מצב הכפתור לפי המדידה במשתנה חדש שנגדיר עבורו, "מצב נוכחי בכפתור". מצב הכפתור יכול להיות 0 או 1, כלומר במידה והכפתור היה לחוץ בעת הדגימה שלו על ידי ה ESP, ה ESP יקרא 0 ומצב זה ישמר ב"מצב נוכחי בכפתור". במידה ובמהלך הדגימה של הכפתור על ידי ה ESP הדגימה תחזיר ערך 1 נשמור ערך זה במשתנה "מצב נוכחי בכפתור", ערך 1 בעצם אומר שהכפתור לא היה לחוץ בזמן הדגימה.

תהליך שליחת ההתראה לענן:

1. דגימה של מצב הכפתור אחת לשנייה ושמירה מצב זה במשתנה "מצב נוכחי בכפתור"
2. במידה ו"מצב נוכחי של הכפתור" הוא אפס, כפי שהוסבר משמע הכפתור היה לחוץ ברגע הדגימה:
  - a. נשלח התראה מיוחדת על כך לענן, כפי שנאמר משמעות ההתראה היא שיש לבצע קניות.
3. במידה ומצב הכפתור הוא 1, משמע הכפתור לא היה לחוץ בעת הדגימה, לכן נמשיך ב 1 ו 2.

### שליחת נתונים לענן והגדרת ההתראות שישלחו למשתמש

המוצר שלנו מורכב מרכיב חומרתי ורכיב אינטרנטי. החלק החומרתי הוא ה ESP עצמו שאחראי על דגימת הנתונים, הצגתם על הצג המקומי, והשמעת התראה קולית וחזותית בעת פתיחת הדלת וכמובן שליחתם לענן. החלק המשלים של המערכת הוא הרכיב האינטרנטי שהוא בעצם המערכת של אתר AdaFruit שאליו אנחנו שולחים את כל ההתראות שצינו בחלק הקודם. באתר זה ניתן להגדיר Dashboard שנבחר על מנת שמציג את הנתונים התצורה שנבחר. בנוסף במערכת adaFruit ניתן להגדיר כללים שישלחו מיילים ללקוחות בהתאם לפיצ'רים שהחלטנו עליהם. הנתונים הנשלחים מה ESP מגיעים לאתר ונשמרים כרשומה נפרדת לכל התראה, רשומה כזאת נקראת Feed. אנחנו שולפים את ה Feed ומגדירים בעזרו את ה DashBaord והכללים שאחראים על ההתראות למייל.

## פירוט הנתונים ששלחנו מה ESP ל AdaFruit:

- כפי שצינו קודם הנתונים שנשלחים לענן נקראים Feeds , כעת נציג את ה Feeds השונים שנשמרים בענן:
  - טמפרטורה במקרר – שמה במערכת temp - מתקבלת פעם בחמש דקות כפי שתיארנו.
    - ערכי ה feed הם פשוט הטמפרטורה "המוחלקת" (ממוצעת) שנמדדה בתוך המקרר במהלך חמשת הדקות של המדידה.
  - טמפרטורה בחדר - שמה במערכת roomtemp – התקבל פעם בחדש דקות כפי שתיארנו.
    - ערכי ה feed הם פשוט הטמפרטורה "המוחלקת" (ממוצעת) שנמדדה מחוץ למקרר במהלך חמשת הדקות של המדידה.
  - לחות במקרר – humidity – התקבל פעם בחמש דקות כפי שתיארנו.
    - ערכי ה feed הם פשוט הלחות "המוחלקת" (ממוצעת) שנמדדה בתוך המקרר במשך חמשת הדקות.
  - מצב הנוכחי של הדלת – dooropen – התקבל בכל רגע שהיה שינוי במצב הדלת.
    - ערכי ה feed הם 0 או 1 בהתאם למצב הנוכחי של דלת המקרר.
  - דלת פתוחה למשך שלוש דקות – doorOpen3Min – התקבל רק כאשר הדלת פתוחה 3 דקות , במקרים בהם הדלת הייתה פתוחה למשך 9 דקות , התקבלו שלושה התראות במרווחי זמן של 3 דקות כל אחד.
    - ערך ה feed הוא רק 1 , כלומר במידה ודלת המקרר הייתה פתוחה 3 דקות ה ESP שלח את הערך 1 כדי להעיד שהדלת פתוחה 3 דקות.
  - כפתור לרשימת קניות – button – התקבל בכל פעם שהכפתור היה לחוץ למשך שניה ומעלה.
    - ערך ה feed הוא רק 1 , כלומר במידה והכפתור נלחץ ה ESP שלח את הערך 1 להעיד על כך שהכפתור נלחץ .

## שימושים במערכת ה AdaFruit:

כעת נציג את השימושים שלנו במערכת ה AdaFruit:

- הצגת Dashboard למשתמש המציג את הנתונים בזמן אמת , נתונים שכוללים את : טמפרטורה במקרר , לחות במקרר וטמפרטורה מחוץ למקרר – כל אלה מוצגים בצורת גרף ומציגים את הנתונים מהמידות במהלך ה 24 שעות האחרונות. בנוסף מוצג מצב הדלת הנוכחי והיסטוריית פתיחת הדלתות של המקרר – נרחיב על ה Dashboard בהמשך.
- שליחת התראות למייל של המשתמש במקרים הבאים:
  - טמפרטורה במקרר גבוהה מ 15 מעלות.
  - דלת המקרר פתוחה במשך 3 דקות רצופות.
  - לחיצה על הכפתור שמורכב על המערכת – המייל ישלח על ידי מערכת adafruit.
  - התראה על כך שלא התקבלו מדידות מה ESP במהלך ה 24 שעות האחרונות (מדדי הטמפרטורה , מדדי הלחות ומצב הדלת).

## אופן שליחת ההתראות המותאמות למייל הלקוח:

כעת נציג כיצד הגדרנו עבור כל Feed את שליחת ההתראה למייל הלקוח . הסבר קצר לפני על סוגי ההתראות שהשתמשנו בהם:

1. Trigger + IFTTT Mail : על ידי הגדרת Trigger אנחנו מגדירים תנאי על FEED בהתאם לתנאים שהגדרנו , תנאים כמו גובה ערך ה Feed גבוה מסף מסוים. בשילוב עם הגדרת ה trigger אנחנו נגדיר שליחת מייל על ידי שירותי IFTTT של adafruit. ברגע שה Trigger מופעל נשלח מייל על ידי שירות ITFFF שהוגדר עבור אותו ה Trigger.
2. AdaFruit : Notification מאפשר להגדיר notification לכל feed שתפקידו לנתר האם לא התקבלו מדידות מה feed בפרק זמן מסוים , ובמידה ואכן לא התקבלו נתונים ל Feed בפרק זמן שהוגדר נשלח מייל על כך.

כעת נסביר כל Feed והתראה שהגדרנו:

- טמפרטורה במקרר גבוהה מידי – בעצם פה השתמשנו ב Trigger + IFTTT mail : הגדרנו Trigger שברגע שמתקבלת מדידה מה ESP אל תוך Feed הטמפרטורה במקרר , temp feed , שערכה גבוהה מ 15 שירותי ה IFTTT שולחים מייל על כך שהטמפרטורה במקרר גבוהה מידי.
- דלת מקרר פתוחה 3 דקות – פה גם השתמשנו ב Trigger + IFTTT mail : הגדרנו trigger שברגע שמתקבלת מדידה ל feed doorOpen3Min של 1 נשלח מייל על ידי שירותי IFTTT על כך שדלת המקרר פתוחה 3 דקות. נזכיר שבגלל שהתראה ל Feed זה מתקבלת רק כאשר ה ESP מדד שהדלת פתוחה 3 דקות , לכן אין "רעש" ומידע מיותר שמתקבל ל feed זה.
- רשימת קניות – השתמשנו ב Trigger + IFTTT mail : הגדרנו trigger כך שברגע שמתקבלת מדידה של 1 ב feed של button אז נשלח מייל על ידי שירותי IFTTT על כך שיש לעשות קניות , ונשלחת התראה עם רשימת הקניות הבסיסית שהגדרנו בגוף ההודעה שנשלחת על ידי ה IFTTT . נזכיר שכמו ב FEED של doorOpen3Min , דיווח מה ESP מתקבל רק ברגע שהכפתור נלחץ למשך יותר משנייה , לכן לא מגיע "רעש" ומידע מיותר ל feed.
- התראה על אי-קבלת נתונים – עבור feeds שמקבלים מידע אודות הטמפרטורה במקרר (temp) , הלחות במקרר (humidity) והטמפרטורה בחוץ (roomTemp) מוגדר Notification בתוך ה Feed עצמו שכאשר ה Feed אינו מקבל אף מדידה חדשה במהלך ה 24 שעות האחרונות נשלחת התראה למייל.

## אופן הרכבת המערכת , אתגרים שנתקלנו בהם ודוגמאות שימוש

בתחילת הפרויקט התחלנו בבניה אינטגרטיבית של המערכת שלנו , כלומר חיברנו כל רכיב בנפרד , כתבנו את הקוד שביצע את הדיגום והמיצוע עבור על מדידה , ובנוסף בדקנו כי הערכים המתקבלים הגיוניים. לאחר שבדקנו וייצרנו קוד עבור כל רכיב בנפרד התחלנו לאחד בין רכיבים שונים וחלקי הקוד שלהם על מנת לוודא שאין בעיה בעת חיבור של מספר חיישנים ורכיבים יחד. חשוב לציין שבכל שלב נתקלנו בקשיים כאלה ואחרים , בין אם זה בהטמעת הקוד ומציאת הפקודות המתאימות , כתיבת קוד באופן כללי , ובאופן החיבור של הרכיבים יחד. על כל קושי ופתרון לבעיות שנתקלנו בהם נרחיב בכל חלק בנפרד.

### חיישן BME280:

תחילה חיברנו רק את חיישן ה BME280 ל ESP על מנת לבדוק שמדידת הטמפרטורה והלחות שמתקבלות ממנו בטווח הערכים הסביר. ולאחר מכן התחלנו בכתיבת הקוד למיצוע הערכים כדי להחליק את הרעשים במדידות . לאחר הצלחה בהחלקת הרעשים הוספנו את בלוק הקוד שבודק האם עברו 5 דקות מאז שליחת הנתונים האחרונה, בשלב זה עדין לא הפעלנו את מודל ה WIFI של ה ESP לכן לא הייתה שליחת נתונים עדין.

### קשיים ופתרונות:

1. הקושי היחיד שחווינו היה באתחול החיישן בחלק של ה setup , בתחילה לא הצלחנו לגרום לחיישן לבצע מדידות. לאחר בדיקה גילינו כי אנינו מצליחים לאתחל אותו ומקבלים שגיאות כבר בשלב של ה setup בקו. הפתרון לבעיה היה שבפקודת האתחול של החיישן bme.begin() , יש להוסיף בסוגריים את הערך 0x76 שזוהי הכתובת של החיישן. לאחר שהוספנו את הכתובת בסוגריים החיישן התחיל לעבוד והתחלנו לקבל את המדידות.

### חיישן טמפרטורה TMP36:

לאחר שהצלחנו למדוד בצורה תקינה את הטמפרטורה בעזרת ה BME280 התחלנו בעבודה עם חיישן הטמפרטורה על מנת למדוד את הטמפרטורה בחוץ. בשלב זה כתבנו קוד שמבצע קריאה ישירה של גובה המתח מהחיישן שמשנתנה כתלות בטמפרטורה שהחיישן מודד. כתבנו את חלק הקוד שמבצע את המיצוע המדידות עבור החלקת הרעשים , בנוסף הוספנו את פקודת שליחת המידע הנמדד לענן לבלוק ששולח מידע פעם ב 5 דקות , חלק קוד שיצרנו כאשר עבדנו עם ה BME280.

## קשיים ופתרונות:

1. מדידת הערכים : בגלל שהחיישן משנה את התנגדותו בהתאם לטמפרטורה שהוא מודד נתקלנו בקושי בעת קריאת המתח מהחיישן. הבעיה העיקרת הייתה למצוא פונקציית המרה מהמתח שנמדד על ידי ה ESP לטמפרטורה. הפתרון היה פונקציה פשוטה שהוצגה במעבדה , השתמשנו בפונקציה הנ"ל עם שינוי קטן כדי לקבל ערכים הגיוניים בטווח ה 27-30 מעלות שנמדדו על ידי החיישן.

## חיישן ה photoresistor ומנורת ה LED האדומה:

בחלק זה החלטנו לחבר את חיישן ה Photoresistor ראשון. כמו חיישן הטמפרטורה , ה Photoresistor משנה את התנגדותו כתלות לחשיפה לאור . ה ESP קרא את המתח ישירות מהחיישן. לאחר מכן הוספנו את מנורת ה LED , והוספנו בקוד בלוק if else שבדק את גובה המתח החשמלי שעובר ב photoresistor , ובהתאם לכך מעדכן את משתנה "מצב הדלת הנוכחי" ו"מצב דלת אחרון" בהתאם. בנוסף הגדרנו בקוד שה ESP יעביר זרם חשמלי חזק או חלש דרך מנורת ה LED כתלות בכמות האור שהחיישן האור קולט.

## קשיים ופתרונות:

1. מדידת הערכים : כמו בחיישן tmp36 השתמשנו בקוד שהוצג בתרגול למדידת המתח החשמלי דרך הנגד , לכן לקח לנו זמן להצליח למדוד ולקבל ערכים הגיוניים.
2. טווח ערכים : שמנו לב שבתנאי תאורה שונים הערכים שחוזרים מפונקציית הדגימה משתנים משמעותית. בתנאי תאורה טובים הערכים יהיו "גבוהים" יחסית , ובתנאי תאורה לא טובים הערכים שחוזרים הם "נמוכים" יחסית. לכן החלטנו לשמור את הבלוק שאחראי על עדכון מצב הדלת "כחלק פתוח" עד אשר נכניס את הרכיב למקרר ונבדוק מה הם הערכים שמתקבלים תחת התאורה במקרר ובמטבח.
3. כתיבת הבלוק שאחראי על בדיקה האם הדלת נפתחה, נסגרה וכמה זמן היא פתוחה : חלק זה היה בעיקר תכנותי מכיוון שהינו צריכים למצוא דרך להבדיל בין דלת שהייתה פתוחה ונסגרה לבין דלת שהייתה סגורה ונפתחה, ובמידה והיא כבר פתוחה כמה זמן היא פתוחה. הפתרון שמצאנו הוא שימוש במשתנה נוסף ששומר את המצב האחרון של הדלת , שנשמר במשתנה "מצב אחרון של הדלת" . השתמשנו בשתי משתנים אלה אפשר לנו לבצע בדיקות של שינוי במצב הדלת ובדיקה של כמה זמן הדלת פתוחה במידה ובמצבה האחרון היה פתוח. בכך פיתחנו את הבלוק בקוד שאחראי על לשלוח התראה לענן ברגע פתיחה וסגירת הדלת , ושליחת התראה לענן במידה והדלת פתוחה כבר 3 דקות.

## הוספת Buzzer למרכב ה photoresistor + Red LED

בחלק זה החלטנו להוסיף buzzer שמצפצף ברגע שהדלת פתוחה . את הפעלת וכיבוי ה buzzer ביצענו בתוך בלוק ה if else שבדק האם הדלת נפתחה או נסגרה ומפעיל את המנורה בהתאם.

## קשיים ופתרונות

1. הקושי העיקרי שנתקלנו בו בחלק זה הוא שלקח לנו הרבה מאוד זמן עד אשר מצאנו קוד שמתאים ומצליח להפעיל את ה buzzer שלנו. בעיה נוספת הייתה שגם קטעי הקוד שמצאנו השמיעו מנגינה ולא צפצוף כמו שרצינו, בנוסף חלקי הקוד הנ"ל היו די מסורבלים וארוכים וכללו הרבה דברים מיותרים. הפתרון היה לדלל מקטע הקוד שמצאנו והצליח לנגן מנגינה ב buzzer שלנו, את חלקי הקוד ששולחים את אות הצלילים ל buzzer עצמו. לבסוף הצלחנו לבודד שיש שתי פונקציות שאחראיות להשמעת הצלילים. אחת אחראית על גובה הצליל ב Hz , והשנייה אחראית על עוצמת הצליל ב dB . לבסוף לאחר בידוד פונקציות אלה הצלחנו לייצר את הצפצוף ולשלבו באותו בלוק קוד שבדק את מצב הדלת והפעיל את המנורה בהתאם.



## הוספת מסך ה-SSD1306:

בחלק זה נתקלנו בכמות הגבוהה ביותר של הקשיים שנתקלנו בו במהלך כתיבת הקוד. ברגע שהוספנו את המסך כבר היו לנו את פונקציות המדידה של המדידות שרצינו כמו: טמפרטורה בפנים ובחוץ, לחות במקרר ומצב הדלת. לכן בחלק זה רצינו לבצע הדפסה למסך של ערכים אלה ועדכון הערכים אחת לשתי דקות. לכן לאחר שפתרנו את הקשיים שנרחיב עליהם מטה, החלטנו לכתוב פונקציה הדפסה למסך, בפונקציה לקחנו מדידה מחיישן ה-BME280 עבור הטמפרטורה והלחות, מדידה TMP36 למדידת הטמפרטורה בחוץ, בנוסף בעת הפעלת ההדפסה נשלחה לפונקציה גם המשתנה של "מצב הנוכחי של הדלת". אופן ההדפסה למסך הוא:

אחת לשתי דקות הפעלנו את פונקציית ההדפסה בבולוק שבודק האם עברו 2 דקות מההדפסה האחרונה, הבולוק בנוי בצורה דומה לבולוק ששולח התראה פעם ב-5 דקות לענן. בתוך הבולוק פשוט קראנו לפונקציית ההדפסה למסך, במידה והיה שינוי במצב הדלת פונקציית ההדפסה למסך נקרא בבולוק שמעדכן את מצב הדלת לאחר שינויה ובנוסף אופס המשתנה שבודק שההדפסה האחרונה בוצעה אחת ל-2 דקות.

## קשיים ופתרונות:

1. מציאת תיעוד מתאים למסך - הבעיה העיקרית שנתקלנו בה עם המסך הוא שלא מצאנו בשום מקום באינטרנט מסך שתאם לדגם המסך שלנו. הדגם שאנחנו משתמשים בשמו הוא הדגם הכי קרוב למסך שקיים אצלינו. כל דגם מסך שמצאנו הכיל כמות חיבורים שונה מהחיבורים שיש לנו, ובנוסף לכך ברוב המסכים שמצאנו שמות החיבורים היו שונים, לבסוף הדגם שאנחנו משתמשים בו הכיל את כל החיבורים שהיו במסך שלנו ולכן בחרנו להשתמש בקוד שמצאנו עבורו.
2. אתחול המסך: בעיה שנבעה ממציאת דגם לא מדויק נבע מפקודות האתחול שנמצאו בקוד שמצאנו ברשת. פקודות אלה היו שונות במקצת מפקודות האתחול שאנחנו השתמשנו בהם, ולכן לקח לנו זמן מה עד אשר הצלחנו להפעיל את המסך בצורה תקינה.
3. חיבור ה-PIN ה-RST: בעיה שנבעה ממציאת דגם לא מדויק, לא היה הסבר מפורט לכל החיבורים שקיימים במסך שלנו ולכן תחילה לא חיברנו את חיבור ה-RST שקיים במסך שלנו, דבר זה גרם למסך לפעול לסירוגין. הפתרון הגיע מחברינו לקורס שגם השתמשנו במסך דומה, מצאנו שיש לחבר את ה-Pin ה-RST מהמסך ל-ESP גם, ברגע שחיברנו אותו והגדרנו את היציאה הנכונה ב-ESP אליה הוא היה מחובר הצלחנו להפעיל את המסך עם הדפסות הרצויות בצורה תקינה.
4. גודל ההדפסה: לקח לנו זמן מה להתאים בעזרת פונקציות ההדפסה את ההדפסה שרצינו למסך שלנו ולגרום לה לזוז כמו סרט נע על המסך, בסופו של הדבר הבעיה נפתרה די מהר פשוט על ידי ניסוי וטעיה בגדלים ובפונקציות הדפסה.
5. התקנת ספריות גרפיקה לסביבת ה-Arduino: נתקלנו בקושי שסביבת ה-Arduino לא הצליחה להוריד ולהתקין מסיבה מסוימת את ספריות ה-adafruit\_GFX ו-adafruit\_ssd1306. הבעיה כנראה נבעה מתקלה כלשהי בשרת שה-Arduino התחבר אליו על מנת להוריד את הספריות האלו. הפתרון שמצאנו היה למצוא מדריך שמסביר כיצד להוריד ולהתקין את הספריות בצורה ידנית.

## הוספת הכפתור

בחלק זה הוספנו את הכפתור, החיבור היה די ישיר מכיוון שמדובר בכפתור פשוט בשימוש רחב והיה לו תיעוד רב באינטרנט. לכן כתבנו לו את בלוק הקוד שקורא את המתח מהכפתור ומעדכן את הערך שנקרא במשתנה, שעליו מבוצעת בדיקה האם הכפתור נלחץ או לא, על ידי בדיקה האם ערך הנשמר הוא 0, במידה וכן נשלחת התראה לענן.

## קשיים ופתרונות:

1. הקושי היחיד שנתקלנו בו בעת הפעלת הכפתור הוא קצב הדגימה, מכיוון שהדגימה מהכפתור כמו משאר הרכיבים מתבצעת אחת לשנייה היה עלינו לוודא שלוחצים על הכפתור לטווח זמן של בערך שניה על מנת שה-ESP יספיק לדגום מהכפתור ברגע הלחיצה.

## חיבור יחד של כל החיישנים והרכיבים מלבד ה-WIFI

בשלב זה כבר הינו בשלבים סופיים של הרכב הפרויקט שלנו מכיוון שכעת ניסינו לאחד את החלקים יחדיו, כאמור קודם כבר הצלחנו לחבר ולממשק את חיישן האור מנורת ה-LED וה-BUZZER, כעת נתקלנו בקשיים שנפרט עליהם עכשיו.

## קשיים ופתרונות

1. בעיה בדגימה של חיישן האור וחיישן ה-TMP36 : מסיבה מסוימת הדגימה של חיישן הטמפרטורה לא עבדה טוב והחזירה ערכים ששונים לאפס ברגע שחיברנו גם את חיישן האור ודגמנו אותם אחד אחר השני.  
הפתרון שהשתמשנו בו הוא הוספת delay בין שורות הקוד שדוגמות את שתי הרכיבים השונים , מסיבה מסוימת הרכיב שנדגם שני נדגם עם ערך אפס תמיד , לאחר הוספתו של delay של 10 מילי-שניות בין שני השורות דגימה וזה פתר את הבעיה
2. מגבלת ערוץ I2C : ברגע שרצינו לחבר גם את המסך והחיישן BME280 שמנו לב ששניהם משתמשים באותם הפינים לכן חשדנו שלא נצליח לדגום ולקרוא את המדידות מהחיישן BME280 במקביל להדפסה למסך . הפתרון היה להתייעץ עם סגל הקורס ובדיקה במקורות אינטרנט האם הדבר אפשרי , בסופו של דבר שניהם מחוברים לאותם חיבורים ב ESP ושניהם מתקשרים על גבי אותו ערוץ I2C .
3. קושי פיזי : בגלל שכבר המערכת הייתה שלמה מבחינת החיישנים שאנחנו משתמשים בהם נהיה לנו "יער" של חוטי חשמל . הפתרון היה לשים לב ולעקוב בעדינות אחרי החיבורי חוטים על מנת לוודא שהכל מחובר בצורה תקינה.
4. פיצול ערוצי ה-3.3V והאדמה : מכיוון שהרבה רכיבים אצלנו השתמשו בחשמל מה ESP ונדרשו לחיבור אדמה , הוצאנו חוטי חשמל משתי Pins אלה מה ESP לתוך החיבור של ה- + וה- Breadboard , ולתוכם חיברנו את כל החיבורים שהיו צריכים חיבור 3.3V וחיבור אדמה.

## הפעלת מודל ה-WiFi:

בשלב זה הינו צריכים להפעיל את מודל ה-WiFi ביחד עם בלוקי הקוד שכותבים לענן ה-adafruit . בחלק זה פשוט הוספנו את הפקודה ששולחת את המדידות ל feeds המתאימים, בלוקים אלה הם בלוק של שליחת נתונים אחת ל 5 דקות , שליחת התראה על דלת פתוחה לאחר 3 דקות , ושליחת התראה בעת שינוי מצב הדלת.

## קשיים ופתרונות:

1. ערוצים : גילינו שכאשר הינו מוסיפים את מודל ה-WiFi לקוד ומאתחלים אותו בחלק ה- setup בקוד , ה-ESP הפסיק לקרוא ערכים מחיישנים מסוימים. בהתחלה בדקנו האם זה באמת מודל ה-WiFi פשוט על ידי אי-הפעלת מודל ה-WiFi ב- setup ובדיקה האם התקלה חוזרת על עצמה , ואכן כאשר אינו פעם החיישנים נמדדו בצורה תקינה , ורק כאשר הינו מדליקים את ה-WiFi ה-ESP לא היה מצליח לקרוא את הנתונים.  
הפתרון היה שגילינו לאחר בדיקה במקורות אינטרנט שבעת הפעלת מודל ה-WiFi חלק מהחיבורים ב-ESP מפסיקים לעבוד בתצורה פיזית ומתחילים לעבוד בתצורה לוגית וזאת לטובת פעולת רכיב ה-WiFi שהותקן ב-ESP , ואכן שמנו לב שבדיוק החיבורים שהיו רשומים כאנלוגיים והיו חיישנים שחיברנו אליהם הפסיקו לעבוד , לכן הפתרון היה להעביר את החיישנים האלה לpin שאינם הופכים דיגיטליים בעת הפעלת ה-WiFi ואכן התקלה נפתרה.
2. צפיפות : בעיה שנבעה מהבעיה שתיארנו קודם , בגלל ש"כיבנו" חלק מה pin על ידי הפעלת מודל ה-WiFi הינו צריכים להזיז את רוב הרכיבים והחיישנים לאותו טווח של pin , לכן רוב הרכיבים התרכזו ב-Pins הגבוהים במספרם מ-30 דבר שיצר צפיפות וקושי בחיבור ועבודה עם החוטים.

## הרכבת המערכת בתוך המעטפת:

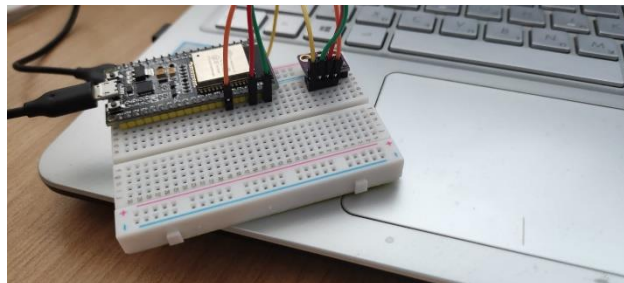
- כפי שהצגנו בכיתה בחרנו לשים את בתוך קופסת גלידה מכמה סיבות
1. הראשונה והעיקרית הייתה שבגלל כמות החוטים ועדינותם רצינו כלי כיבול שיגן על החיבורים שישבו בצורה עדינה בתוך ה-breadboard ולשמור על ESP , לכן רצינו פשוט כלי כיבול שיחזיק בתוכו בצורה כמה שיותר בטוחה.
  2. השנייה הייתה שה-breadboard שהשתמשנו בו לטובת חיבור ה-ESP ושאר החוטים שהובילו לשאר החיישנים והרכיבים היה די גדול מפאת כמות החוטים שנדרשו לכל חיישן, לכן הינו צריכים לבחור בכלי כיבול די גדול כדי להכיל הכל בפנים בצורה בטוחה ומרווחת.

בנוסף הינו צריכים לפצל את הרכיבים ל breadboard שונים מכיוון שחיישנים שונים היו צריכים לשבת במקומות שונים על מנת לבצע את המדידות שלהם

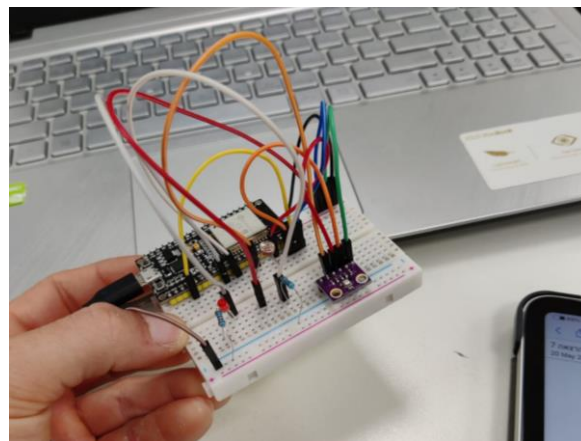
- חיישן photoresistor ומנורת LED – שני אלה רצינו לחבר על אותו breadboard ולמקם בתוך המקרר כמה שקרוב יותר , לכן חיברנו אותם על board שבהרכבו למכסה הקופסא שישבה בתוך המקרר
- חיישן BME280 – החיישן הזה נשאר על אותו ה Board של ה ESP והוא נמצא בתוך הקופסא שבתוך המקרר.
- חיישן TMP36 – חיישן זה ישב על Board משלו מחוץ למקרר , ה Board הודבק ממש מחוץ למקרר על הדופן החיצונית . בנוסף החיישן הוארך עם חוטים חשמל ארוכים מה Board עליו ישב ל board של ה ESP ומשמה לתוך ה ESP עצמו.
- Buzzer – את BUZZER רצינו למקם קרוב ל ESP לכן חיברנו אותו ללא board על גבי המכסה של הקופסא בה ישב ה esp וה board הראשי.
- מסך SSD1306 – המסך לא חובה עם Board אלא ישירות בחוטי חשמל ל Board הראשי והוא ישב על הקופסא שבתוכה מוקם ה Board הראשי וה ESP.

תמונות משלבי ההרכבה עד החיבור הסופי והצבה במקרר:

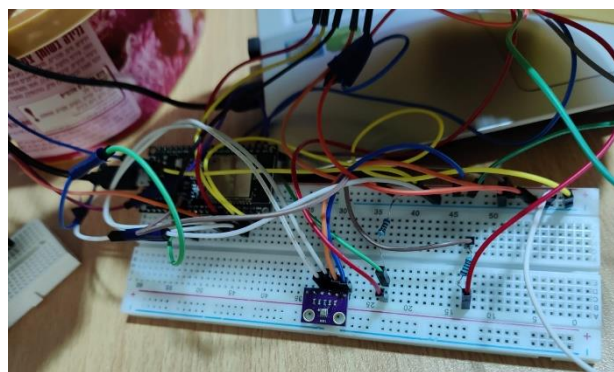
שלב ראשוני של חיבור ה BME280 בלבד

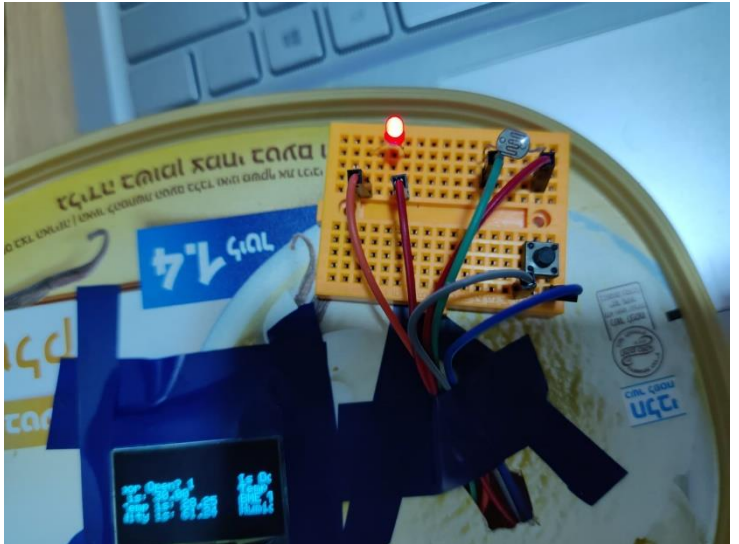
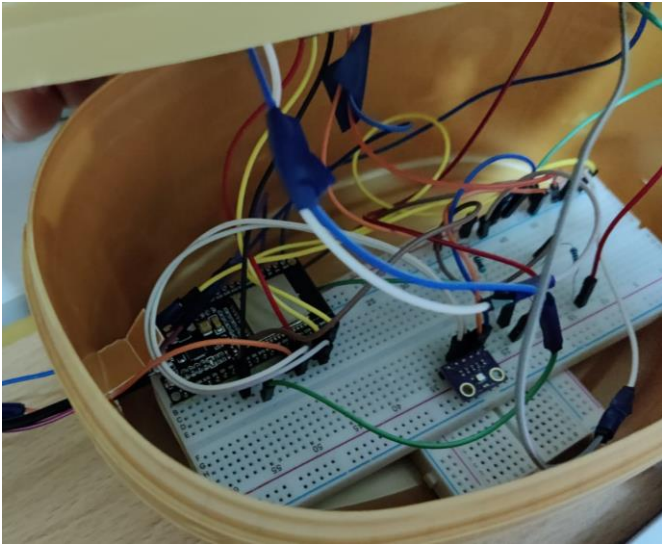


חיבור יחד של Photoresistor , BME280 , TMP36 , Red LED



חיבור כל הרכיבים יחדיו ופיצולם לBoard נפרדים









המערכת עצמה בתוך המקרר , לצערנו לא ניתן לצרף סרטון למסמך הסיום אך הסרטון קיים במצגת הסוף שהוגשה.

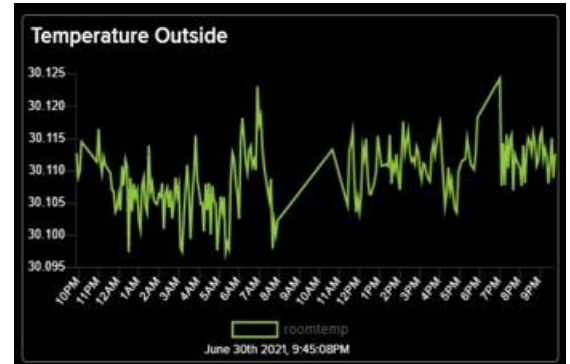
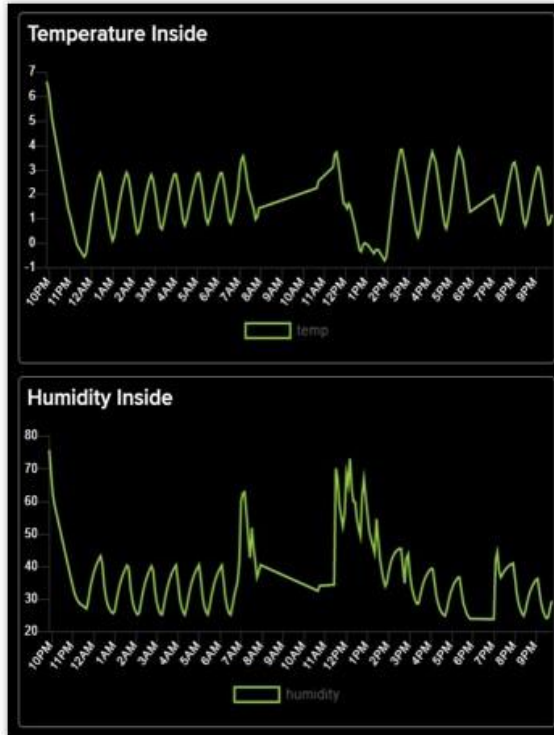


## תמונות של ה Dashboard וההתראות השונות

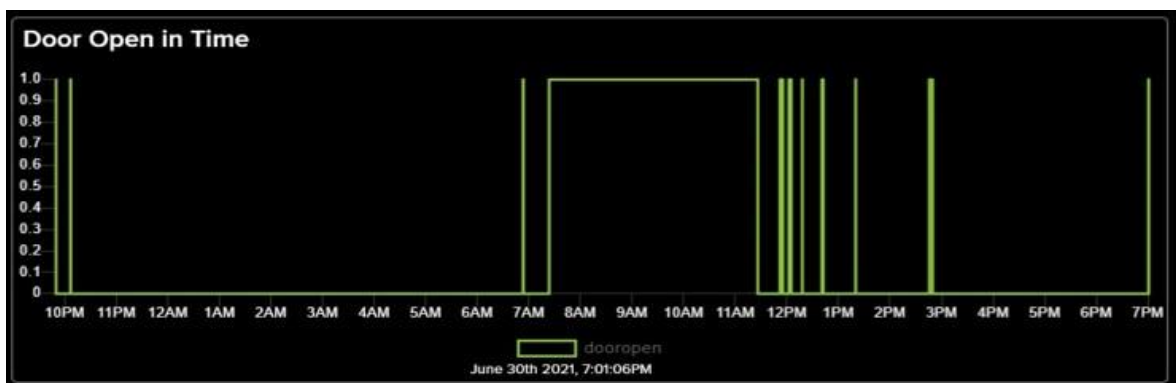
תמונה של ה DashBaord:

חשוב לציין שיש לנו רק צילום של תמונת מצב יחידה ולצערנו אין לנו סרטון שמשלב את עבודת המערכת בלייב והשינויים שקורים ב dashboard בלייב.

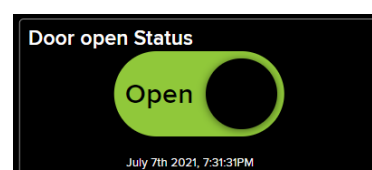
- גרפים הטמפרטורות והלחות, הגרפים משקלים את המדידות שהתקבלו מה ESP לאורך 24 השעות האחרונות:



- גרף היסטוריית פתיחת המקרר במהלך 24 השעות האחרונות, כל פתיחה מעלה את גובהה הגרף ל 1 בשעה שבה נפתח, וסגירה מורידה את גובהה הגרף ל 0 בעת סגירתו.



- Toggle – מתאר את מצב דלת המקרר ברגע הנוכחי, אם הדלת סגורה, כלומר המדידה האחרונה שהתקלה היא שהמקרר סגור יוצג Closed בצבע אדום, במידה והתקל מידע שהדלת פתוחה, ה toggle ישתנה לצבע ירוק ויציג Open עד הקבלת המידע החדש שיתריע כי המקרר נסגר שוב.



תמונות של ההתראות:

- התראה על רשימת הקניות:

## Grocery Shopping! Inbox x



zeevyam@gmail.com

to me ▼

Button Was Pressed , Please Remember To Buy: Eggs, Milk, Vegetables, Water, Fruits and Bread.

- התראת דלת פתוחה יותר מ 3 דקות:

## Fridge Door Open! Inbox x



zeevyam@gmail.com

to me ▼

The Fridge Door Is Open for 3 Minutes now! Please Close It before any of the Groceries goes Bad

- התראות על כך שלא התקבלו נתונים במהלך ה 24 שעות האחרונות

<input type="checkbox"/> ☆ <span>Inbox x</span> Adafruit IO	Your dooropen feed has been marked offline - Your dooropen feed has been marked as offline. The last value received wa...	Jul 2
<input checked="" type="checkbox"/> ☆ <span>Inbox x</span> Adafruit IO	Your temp feed has been marked offline - Your temp feed has been marked as offline. The last value received w...	
<input type="checkbox"/> ☆ <span>Inbox x</span> Adafruit IO	Your humidity feed has been marked offline - Your humidity feed has been marked as offline. The last value received was: ...	Jul 2
<input type="checkbox"/> ☆ <span>Inbox x</span> Adafruit IO	Your doorOpen3Min feed has been marked offline - Your doorOpen3Min feed has been marked as offline. The last value r...	Jul 2

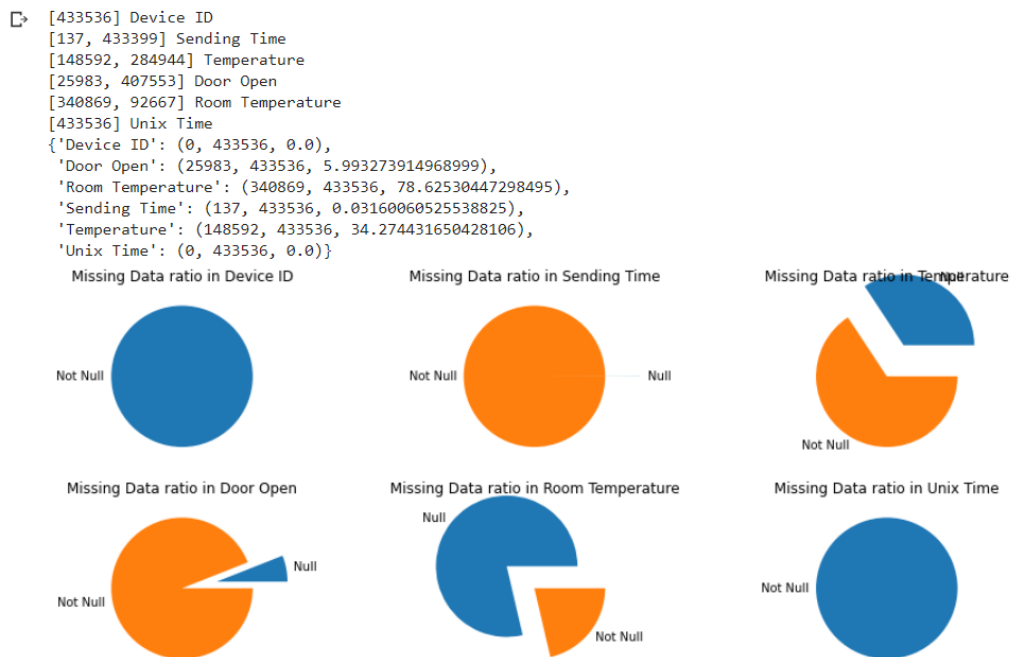
## דיון ומסקנות

- במהלך העבודה נחשפנו לעולם תוכן ומידע חדש , העולם של המיקרו-בקרים , בנוסף במהלך העבודה ההרכבה והפעלת המערכת הסופית למדנו והסקנו מסקנות שלא ראינו במהלך העבודה ונדון בהם פה.
1. תוצאות המדידות שעלו לענן : שמנו לב לתבניות עבודה של המקרר , למשל אפילו אם נסתכל על גרף הטמפרטורה והלחות הפנימית במקרר נשים לב כי יש תמיד עליות וירידות הן בלחות והן בטמפרטורה ושניהם בדרך כלל הולכות יד ביד , מסקנתנו הוא שכנראה משיכלו חיסכון באנרגיה המקרר אינו תמיד עובד , אלא המקרר מקרר לפרק זמן מסוים עד אשר מגיע לטמפרטורה ולחות מסוימת ומפסיק לקרר לפרק זמן מסוים על מנת לחסוך באנרגיה , ניתן לשים לב לזה בעליות וירידות בגרפים שציינו , צורתם מזכירה מסור מסוג כלשהו.
  2. החיישנים יצרו תקלות : כפי שניתן לראות בגרף היסטורית פתיחת המקרר ב 24 שעות האחרונות , יש חלק ארוך שבו הגרף בגובהה 1 בין שעה 7 וחצי בבוקר עד 11 וחצי בבוקר , מה שמרמז שהדלת הייתה פתוחה 3 שעות , דבר שלא קרה ומעיד על טעות בקריאותיו של החיישן . דבר שיכול היה להתרחש מעודף לחות שהצטבר על המגעים וביניהם וגרם לקריאה לא מדויקת של החיישן.
  3. בחירת המיקום של הרכיבים : כפי שניתן לראות הצבנו את המסך עם המידע על גבי הקופסא שישבה בתוך המקרר , רק לאחר חיבור המערכת הבנו כי הדבר חסר היגיון , לכן למדנו שיש לתכנן גם את חלק הפריסה של המערכת ולהתחשב בפונקציונליות של כל רכיב ומכך לגזור היכן הכי כדאי להציב אותו.

## ניתוח נתונים משנה קודמת

בניתוח הנתונים משנה שעברה בחרנו להתרכז בניתוח המידע ברמת העל ולא התרכזות במקרר ספציפי , רצינו לקבל תמונה יחסית "מקיפה" לתפקוד הממוצע של המקררים ולא להתרכז בקבוצה קטנה של מקררים שהם בעלי ביצועים טובים או גרועים. הסיבה היא שאנחנו יותר מעוניינים בנתונים סטטיסטים על "האוכלוסיה" ולא על פריטים בודדים מהאוכלוסיה. את הניתוח נתונים ביצענו ב Python בסביבת Google Collab.

תחילה בדקנו בכלל אילו עמודות קיבלו עם ה DS , ולאחר מכן בדקנו עבור כל עמודה כמו עמודות הגיעו עם ערך ריק ומה אחוז תאים הריקים מתוך כל התאים באותה עמודה. כל תרשים pie מציין עבור כל עמודה את החלק היחסי של התאים הריקים אל מול התאים בעלי המידע.

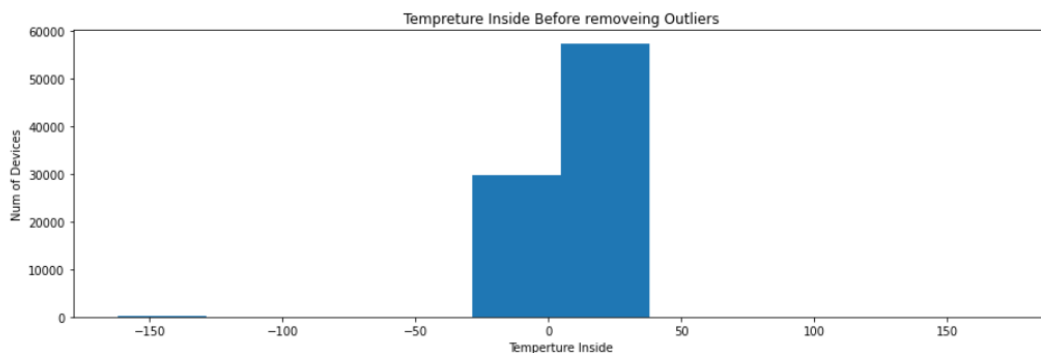


שמנו לב שבעמודות Device ID אין תאים ריקים כמצופה , בנוסף שמנו לב שיש שתי עמודות שמבטאות את אותו הדבר Unix Time | Sending Time , כאשר בעמודה Sending Time יש עמודות ריקות .

לאחר ניתוח זה ביצענו את הפעולות הבאות

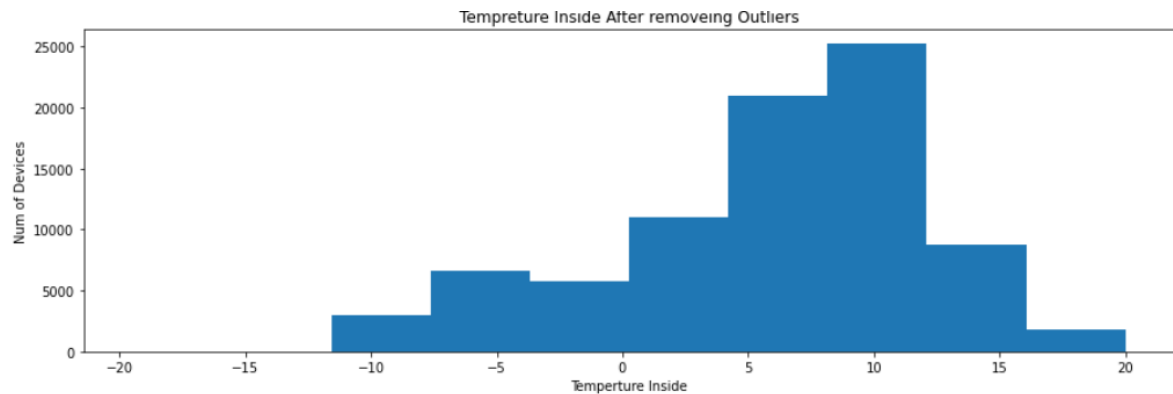
1. מחקנו את עמודות Sending Time
2. הוספנו שני עמודות Regular Time | Time Of Day מתוך הזמן שיש ב Unix Time
3. מחקנו את השורות בהם יש תא ריק של Temperature או Room Temperature

כעת בחרנו להתרכז בניתוח הטמפרטורה הפנימית במקרר , לכן יצרנו היסטוגרמה של העמודה של הטמפרטורה בתוך המקרר:





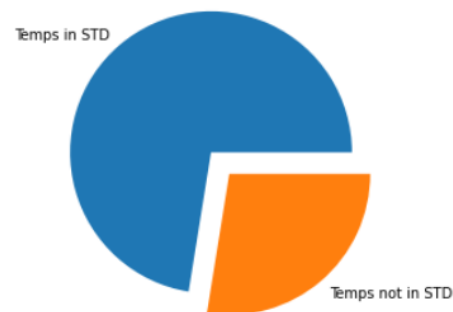
שמנו לב שיש תצפיות חריגות , כגון מדידות ב -150 . בחרנו לסנן ערכים שאינם בטווח ערכים [-20,20] טווח רחב כזה מכיוון שלא ידענו האם החיישן הוצב במקרר או במקפיא , לכן לאחר סינון זה יצרנו היסטוגרמה נוספת.



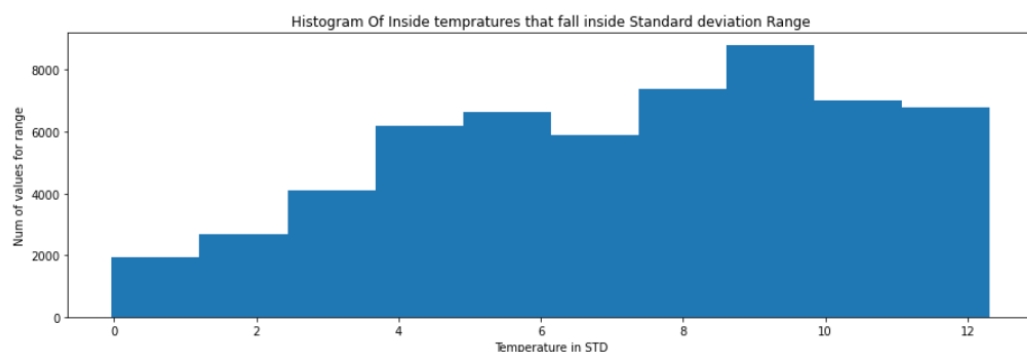
כעת ההיסטוגרמה הגיונית , הערכים נמצאים בין -10 מעלות ועד 20 מעלות , כאשר רוב התצפיות הן באזור ה 5-10 מעלות.

בחרנו לבדוק יותר לעומק את הטמפרטורות שנמדדו במקרר ולכן ביצענו עוד חישובים סטטיסטים על הטמפרטורות כמו : טמפרטורה מקסימלית , מינימליות , ממוצע , שונות וסטיית תקן של המדידות. הצגנו בגרף את יחסי הטמפרטורות בתוך סטיית התקן הראשונה אל מול כאלה שלו וקיבלנו

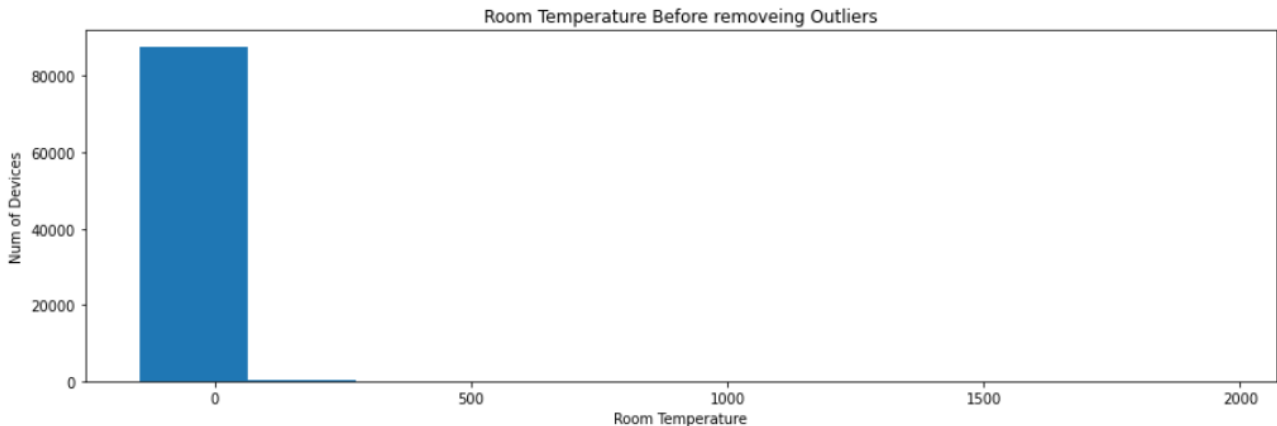
Max Inside Temperature is 20.0 And Minimum Inside Temperature is -19.459999  
Mean temp is 6.139548715371983  
Inside Temperture variance 38.17551085275693  
Inside Temperature Standard deviation 6.17863341304183  
Percentage of Inside Temps inside fridge that fall inside STD range is 72.45547555892384 %  
Inside Temperatures That fall inside Standard deviation range



נשים לב שבערך 72 אחוז מהתצפיות הן בטווח סטיית התקן הראשונה . בנוסף נשים לב שממוצע הטמפרטורות הוא בערך 6.2 מעלות , מה שתואם לטמפרטורה בממוצעת שהינו מצפים לראות במקרר , בין 2-8 מעלות. כעת נציג את היסטוגרמת המדידות של הטמפרטורה במקרר רק של התצפיות שנכנסות לטווח סטיית התקן הראשונה:



כעת נתרכז בניתוח המדידות של הטמפרטורה בחדר , room Temperature , לכן כמו קודם יצרנו היסטוגרמה של העמודה של Room Temperature

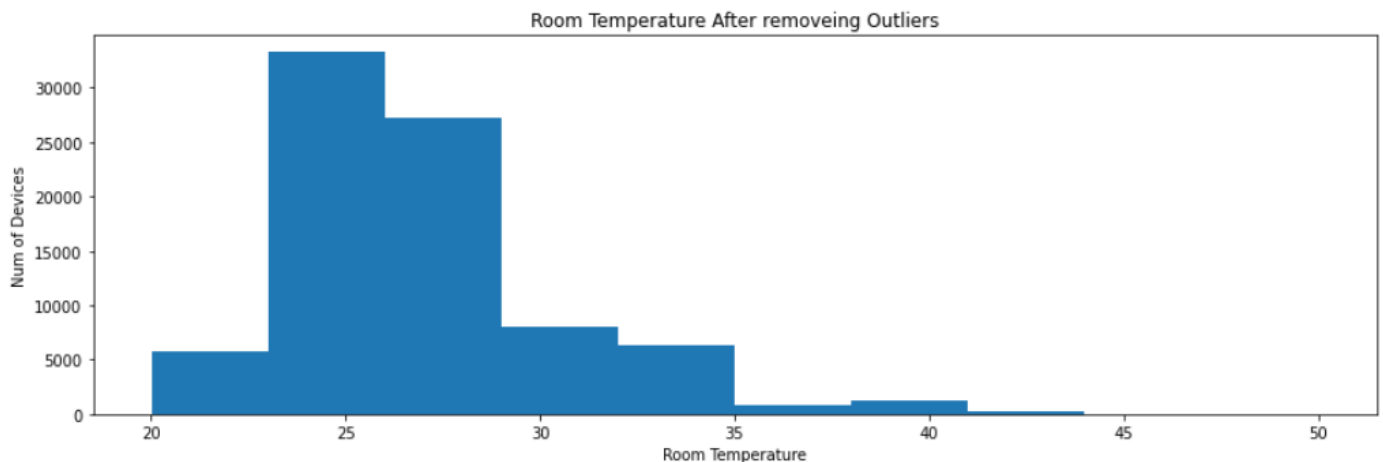


כמו קודם שמנו לב שיש מדידות חריגות שגבוהות מ 100 מעלות ונמוכות מאפס מעלות .  
בחרנו לבדוק את טווח התאריכים בהם בוצעו המדידות כדי לשער טווח מדידות טמפרטורה תקין , וגילינו שהמדידות בוצעו כולן בחודש יוני של שנת 2020

```
[27] show_dates_ranges(data)
```

First date recorded is 2020-06-04 05:02:22 and last date recorded is 2020-06-23 11:00:35

בהתחשב במידע זה על התאריכים והעובדה שאנחנו חיים בישראל ולא בקנדה בחרנו לסנן את המדידות ולהשאיר רק מדידות בטווח [20,50]



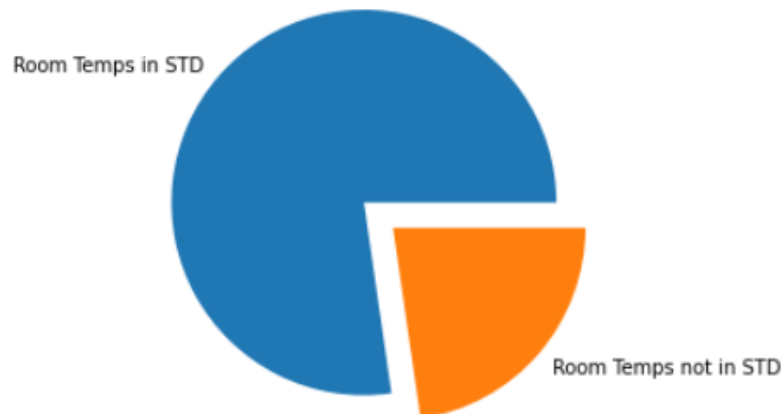
כעת נוכל להבחין שההיסטוגרמה הטמפרטורות בחדר הרבה יותר מתאימות לטמפרטורות בישראל בחודש יוני שכבר די חם , רוב הטמפרטורות הן בטווח של 22 מעלות עד 28 מעלות, ויש זנב ימני להיסטוגרמה – זנב זה של טמפרטורות שגבוהות מ 30 מעלות , משהו שפחות אופייני לחודש יוני , לכן יש מצב שעדין יש הטיה סטטיסטית במדידות שנובעת ממדידה שגויה.

לכן ננסה לנתח עוד יותר את המידע של הטמפרטורה בחדר בטווח של [20,50] , לכן ביצענו ניתוח של המדידות הנ"ל כמו מקודם : טמפרטורה מקסימלית , מינימלית , ממוצע , שונות וסטיית תקן של המדידות.

```
Max Room Temperature is 50.0 And Minimum Room Temperature is 20.027374
Mean Room temp is 26.800560081950234
Room Temperature Variance is 12.974661855226262
Room Temperature Standard Deviation is 3.602035793162842
Percentage of Room Temp reading inside Standard Deviation range is 77.28685107995453
```

נעלה את הטמפרטורות בטווח סטיית התקן הראשונה וכאלה שמחוץ לה על גרף pie

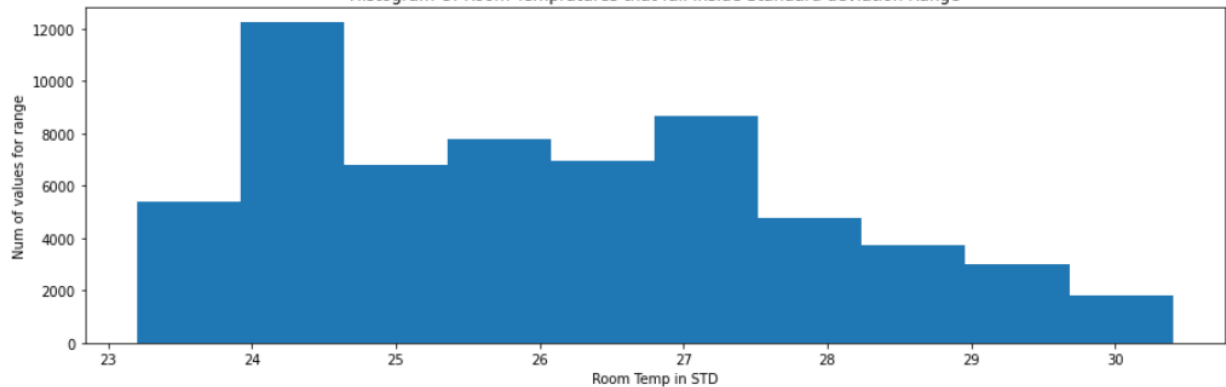
Room Temperatures That fall inside Standard deviation range



בהתאם לגרף ולתוצאות הכתובות מעלה, נוכל לראות שבעמם הטמפרטורה הממוצעת בחדר היתה 26 מעלות, במבט ראשון די הגיוני לטווח הזמן הזה בישראל, ובנוסף בערך 77% מהמידות היו בתוך טווח הראשון של סטיית התקן.

כעת ננסה להעלות על היסטוגרמה רק את המדידות בתוך טווח סטיית התקן הראשונה:

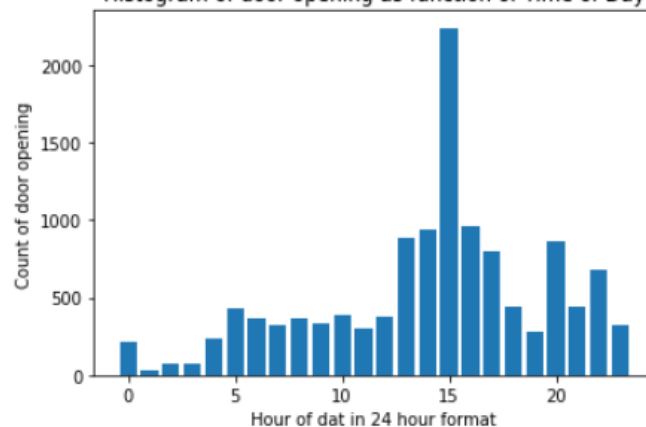
Histogram Of Room Temperatures that fall inside Standard deviation Range



נוכל לשים לב שיש קפיצה משמעותית באזור ה 24 מעלות, דבר שמאוד הגיוני לטמפרטורות בחודש יוני. נשים לב שרוב הטמפרטורות שנמדדו מתפזרות בטווח 22-27.5 מעלות, שוב דבר שמאוד הגיוני לחודש יוני בארץ, מה שיכול לרמז שהמדידות היו די מדויקות עבור הטמפרטורה החדר.

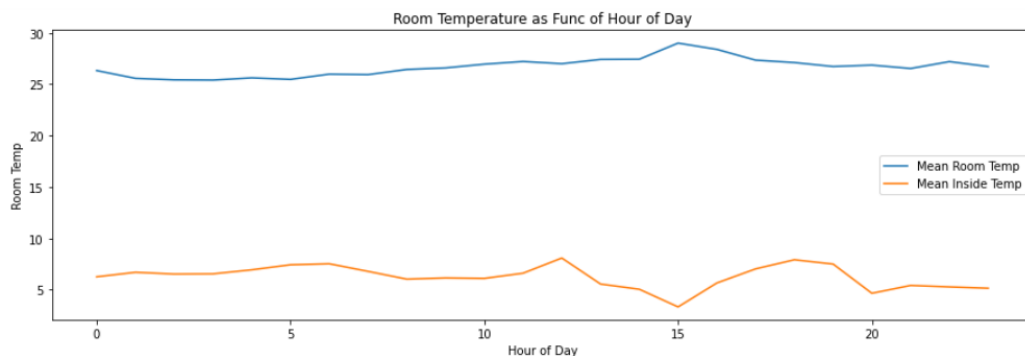
הדבר הבא שבחרנו להתרכז בו הוא כמות פתיחות הדלת לאורך שעות היום, לכן אספנו את כל המדידות וספרנו עבור כל שעה ביממה את כמות הפתיחות דלת שהתקבלו בה, העלנו על היסטוגרמה וקיבלנו

Histogram of door opening as function of Time of Day



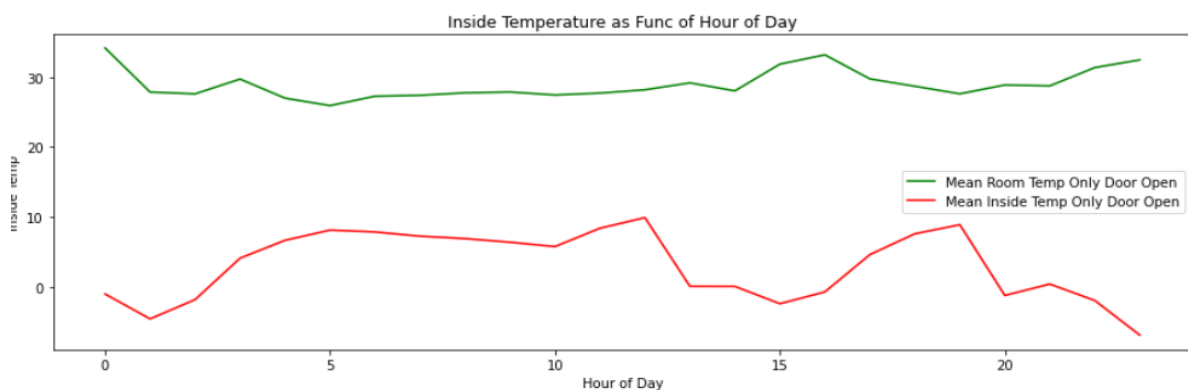
השעות בגרף בציר ה X הן מ 0 עד 24 , כאשר כל ספרה מייצגת שעה בפורט 24hours. נשים לב שיש לנו קפיצה מאוד גדולה של יותר מ 2000 פתיחות סביב השעה 15 , כנראה דבר זה נובע מטעות במדידה שנכנסה ל DS ולא שמנו לב אליה , אך אם נתעלם מכמות הפתיחות החריגה סביב השעה 15 בצהריים. נשים לב שיש לנו עלייה קטנה סביב השעה 5-6 , סביב השעות 12-16 וסביב השעה 20 , דברים אלה בסהכ הגיוניים כי אפשר להניח שאלה שעות של הארוחות הגדולות ביום , בבוקר סביב השעה 5-6 ארוחת בוקר וקפה , בצהריים סביבות השעה 14-16 ארוחת צהריים , ובערב סביב השעה 20 כשחוזרים מהלימודים לעבודה לארוחת ערב. נתונים אלה סך הכל עושים שכל ושוב בהסתייגות מהמדידה החריגה סביב השעה 15.

דבר אחרון שרצינו לבדוק הוא אם יש שינוי בטמפרטורה בתוך המקרר בקורלציה עם הטמפרטורה בחוץ , לכן העלנו על גרף את הטמפרטורה הפנימית והחיצונית הממוצעת לאורך שעות היום , כמו קודם בטווח השעות בין 0-24 , הגרף שקיבלנו:



ניתן לראות שאין כל כך קורלציה , אולי מלבד סביב השעה 12-15 בה ניתן להבין שהגרף העליון שמייצג את טמפרטורת החדר עולה בשעות אלה , דבר שהגיוני בהתאם לסביבה , ודווקא גרף הטמפרטורה במקרר בשעות האלה יורד בטמפרטורה הממוצעת. דבר זה יכול לנבוע אולי מכך שהחיישנים המובנים במקרר חשים שיש עליה בטמפרטורת הסביבה ולכן על מנת שזה לא ישפיע על המוצרים בפנים הוא מתחיל לקרר את החלל הפנימי.

בגלל שלא מצאנו קורלציה מובהקת בין הטמפרטורה הממוצעת מחוץ ובתוך המקרר במהלך שעות היום , בחרנו לבדוק האם ניתן למצוא קשר בין שעות שבהן היו פתיחות דלת רבות , לטמפרטורה הפנימית הממוצעת של המקרר אל מול הטמפרטורה החיצונית הממוצעת , העלנו על גרף כמו מקודם וקיבלנו:



הגרף שקיבלנו מייצג את הטמפרטורה הממוצעת בחוץ ובפנים מתוך השורות שבהן היו פתיחות דלת בחלון השעות המתאים , ואנחנו מקבלים שהטמפרטורה החיצונית לא משתנה כל כך כפונקציה של פתיחות הדלת. לעומת זאת נשים לב שדווקא הטמפרטורה הממוצעת בתוך המקרר גבוהה יחסית בשעות 3-12 , יורדת בין 12-14 ועולה שוב בין 16-19 ויורדת שוב בין 19-24 , דבר זה לא כל כך היוני ואינו מתיישב עם המסקנות מהיסטוגרמת פתיחות הדלת שהצגנו , לכן כנראה שאין קורלציה בין פתיחות הדלת של המקרר לטמפרטורה הפנימית בניגוד למה שהינו מצפים לראות.