**פרויקט גמר אלקטרוניקה – שעון לכבדי שמיעה**

מכללת אורט יובלי אריאל

עבודה / פרוייקט גמר – פרויקט גמר הנדסאי אלקטרוניקה

שם הפרויקט – שעון לכבדי שמיעה

מגיש: אלדד צמח

מנחה: דניאל גורוביץ'

מאי 2018

**מבוא**

הפרויקט הוא שעון לכבדי שמיעה אשר מעביר מידע לבעל השעון על ארועים שמתרחשים בבית. כגון, התינוק בוכה, צלצול בדלת. לדוגמא, כאשר התינוק בהשגחת מבוגר כבד שמיעה בוכה, בכדי שיוכל לדעת שהתינוק בוכה הוא צריך התראה מפני שהוא לא מסוגל לשמוע זאת. על מנת להתריע למבוגר על בכי, קיים חיישן קול בחדר של התינוק אשר קולט האם הוא בוכה או לא ומעביר את המידע לאינטרנט, והשעון שנמצא על היד של כבד השמיעה קולט את השינוי באינטרנט ומקבל את ההתראה בזמן אמת.

בפרויקט נעשה שימוש ברכיב ה - ESP8266 אשר מתחבר לאינטרנט של הבית ושולח או מקבל את המידע הרצוי. אותו מידע מתקבל על ידי חיישן הקול אשר בודק את רמת הקול הנמצאת סביבו ובמקרה של חריגה שולח אות מתאים לבקר.

מטרת הפרויקט היא להפוך את חיי כבד השמיעה לקלים יותר ונוחים יותר לתפקוד היומיומי. למשל, כבד שמיעה אשר רוצה לישון בלילה ברוגע כאשר ילדיו ישנים בחדרם יכול לעשות זאת בעזרת שעון זה אשר נמצא על ידו, ועל ידי החיישן המותקן בחדר הילדים, כך הוא יוכל לקבל את התרעה על הבכי גם אם הוא לא שומע אותם.

הפרויקט פונה אל קהל כבדי השמיעה אשר מתקשים לתפקד ביומיום בגלל הבעיה שלהם לשמוע את הדברים השונים המתרחשים בבית.

הבעיה הקיימת היא שלכבדי השמיעה יש פתרון למשך היום עם מכשירי שמיעה או הגברת השמיעה, לעומת זאת בלילה כאשר הם הולכים לישון, הם מורידים את המכשיר ולכן לא מודעים לדברים הקורים סביבם. משום כך הפרויקט שלי מהווה פתרון לבעיה זו בכך שהוא יכול לשמש את כבדי השמיעה גם בלילה כאשר מותקן בשעון רטט אשר יעיר אותם במקרה ויש רעש בבית, ואין להם דרך לדעת זאת בלי ההתראה של השעון.

היתרונות של השעון שלי הם בנוחות ויעילות השימוש. כמו כן, ניתן להוזיל את עלות השעון ביצור המוני, וכך להנגיש לכל כבד שמיעה את הפתרון האולטימטיבי.  
בנוסף השעון מציג את השעה, ומאפשר למשתמש לראות את השעה המדוייקת, בלי הצורך בכיוון השעון. בנוסף, כאמור, השעון נותן מענה גם בלילה כאשר כבדי השמיעה לא מחוברים למכשירם, ומאפשר להם להיות מודעים ולהגיב לאירועים שונים, דבר שלא קיים היום.

**תודות**

קודם כל תודה לדניאל המנחה שלנו, שהיה שם כמו סופרמן, עבר בין הכיתה לעזור לכולם, ואנחנו כיתה קשה. אני אישית רוצה להודות לו על התמיכה הרבה והעזרה שנתן לי במהלך כל הפרויקט, שתמיד האמין בי ורצה רק שהתקדם יותר.

תודה לדור שעזר לי בספר פרויקט ותמך בי לאורך כל הדרך.

תודה לחברי הכיתה שלי שהיו לי לחברים טובים ועל העזרה שנתנו אחד לשני לאורך כל השנים.

לבסוף תודה לאמא ואבא שהביאו אותי עד הלום, שבזכותם זכיתי להגיע למקום שאני נמצא היום ולהצליח במה שאני עושה, תודה שתמכתם בי כל החיים ושנתתם לי הכל.

**הצהרת הסטודנט**

**אני \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ת"ז \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**החתום מטה, מצהיר בזאת שכל פרויקט הגמר המוגש בספר זה, הינו פרי עבודתי בלבד, על בסיס**

**הנחייתו**

**של המנחה ותוך הסתמכות על מקורות הידע והמידע האחרים המצויים בביבליוגרפיה המובאת**

**בסיום ספר**

**זה.**

**אני מודע לאחריות שהנני מקבל על עצמי ע"י חתימתי על הצהרה זאת,**

**שכל הנאמר בה הינה אמת ורק אמת.**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**חתימת הסטודנט**

**תוכן עניינים**

|  |  |
| --- | --- |
| **מבוא** | **2** |
| **תודות** | **3** |
| **סכמת מלבנים והסבר** | **6-9** |
| **סכמה חשמלית** | **10-11** |
| **רכיבי הפרויקט** | **12** |
| **Arduino** | **13-18** |
| **צג – Oled** | **19-23** |
| **חיישן המיקרופון** | **24-25** |
| **הסבר הרטט** | **26** |
| **רכיב האינטרנט ESP8266** | **27-30** |
| **תוכנה – קוד הפרויקט** | **31-47** |
| **Datasheet – רטט** | **48-51** |
| **Datasheet – ESP8266** | **52-55** |
| **Datasheet – Oled** | **56-57** |
| **DataSheet – מיקרופון** | **58-59** |
| **ביבליוגרפיה** | **60** |

**סכמת מלבנים - שעון**

**כפתור**

**אתר האינטרנט**

**מיקרו מנוע רטט**

Oled

**Arduino Lilypad**

**ESP8266**

**הסבר סכמת מלבנים – שעון:**

1. ESP8266 רכיב לחיבור האינטרנט, משמש כדי לקלוט מידע מהאינטרנט לגבי הרעש בבית/ בחדר המבוקש.  
   תפקידו לעדכן את הארדואינו כאשר יש אירוע בבית.  
   בנוסף תפקידו להתריע לאתר כאשר המשתמש מקבל את ההודעה, וזאת על ידי לחיצה על הכפתור, אשר שולח התראה לארדואינו, ובכך הארדואינו מחזיר תשובה לאתר דרך רכיב האינטרנט.
2. מיקרו מנוע רטט, משמש כדי להתריע למשתמש על אירוע שקורה בבית, גורם למשתמש לתצומת הלב כאשר השעון שעל היד שלו רוטט.  
   הרטט מפסיק ברגע שהכפתור נלחץ, ומתחיל כאשר הארדואינו מקבל מידע על אירוע.
3. Oled צג, משמש כדי להעביר את המידע אל המשתמש.  
   הOled הוא מסך קטן אשר מראה למשתמש היכן מתרחש האירוע.  
   בנוסף הוא מראה למשתמש את השעה בזמן שלא קורה אירוע.
4. אתר האינטרנט מקבל את המקום של האירוע בבית, מאחסן אותו במסמך טקסט, ונותן גישה לרכיב אינטרנט לגשת את המסמך כדי לקחת את הנתונים הרצויים.  
   בנוסף הוא מאפשר לרכיב אינטרנט מידע על השעה של היום בשעת אמת ללא צורך בכיוון השעה.

**סכמת מלבנים – מקלט האירוע**

**כפתור**

**חיישן הקול**

**אתר האינטרנט**

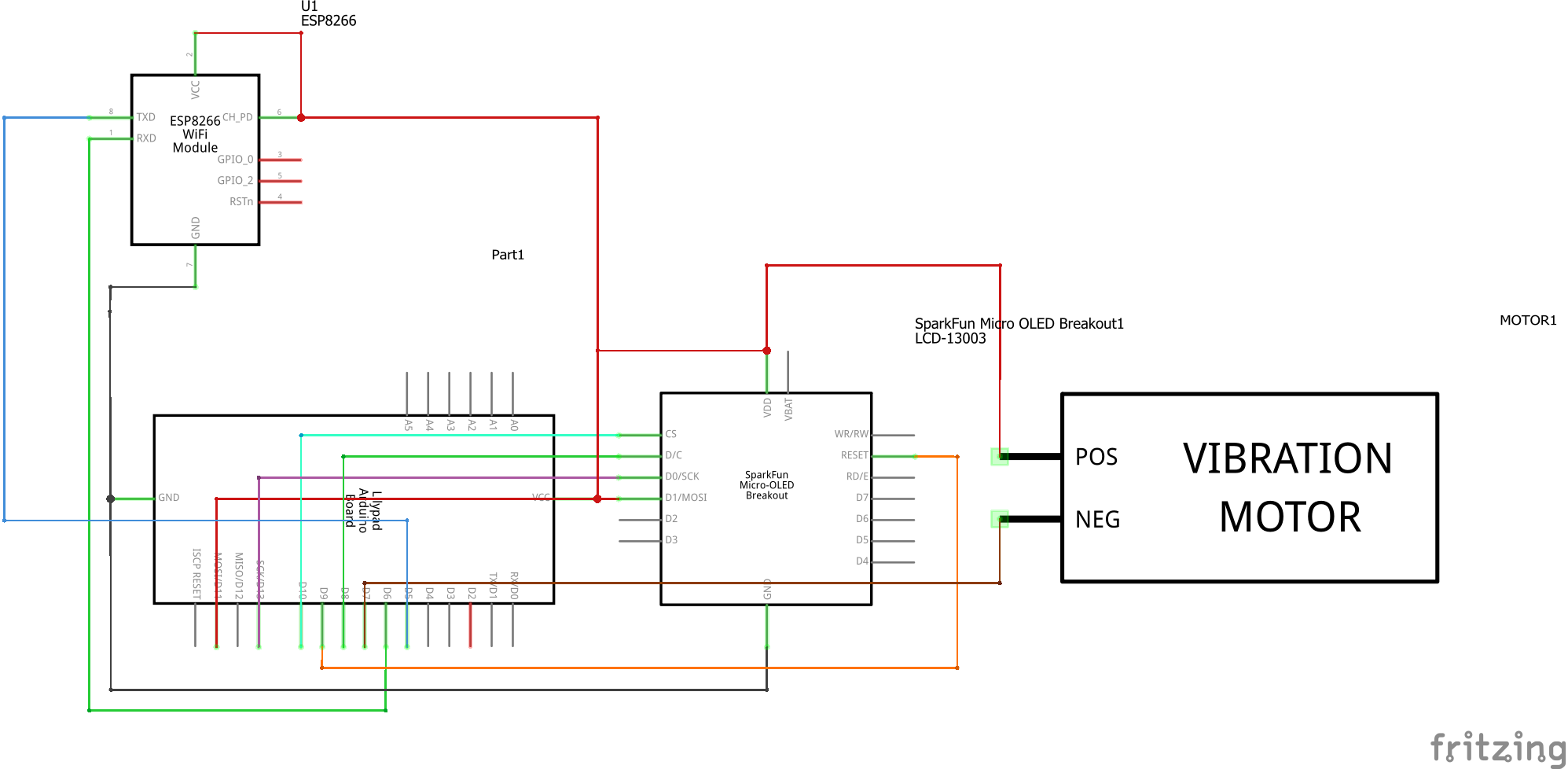
**Esp8266**

**Arduino Uno**

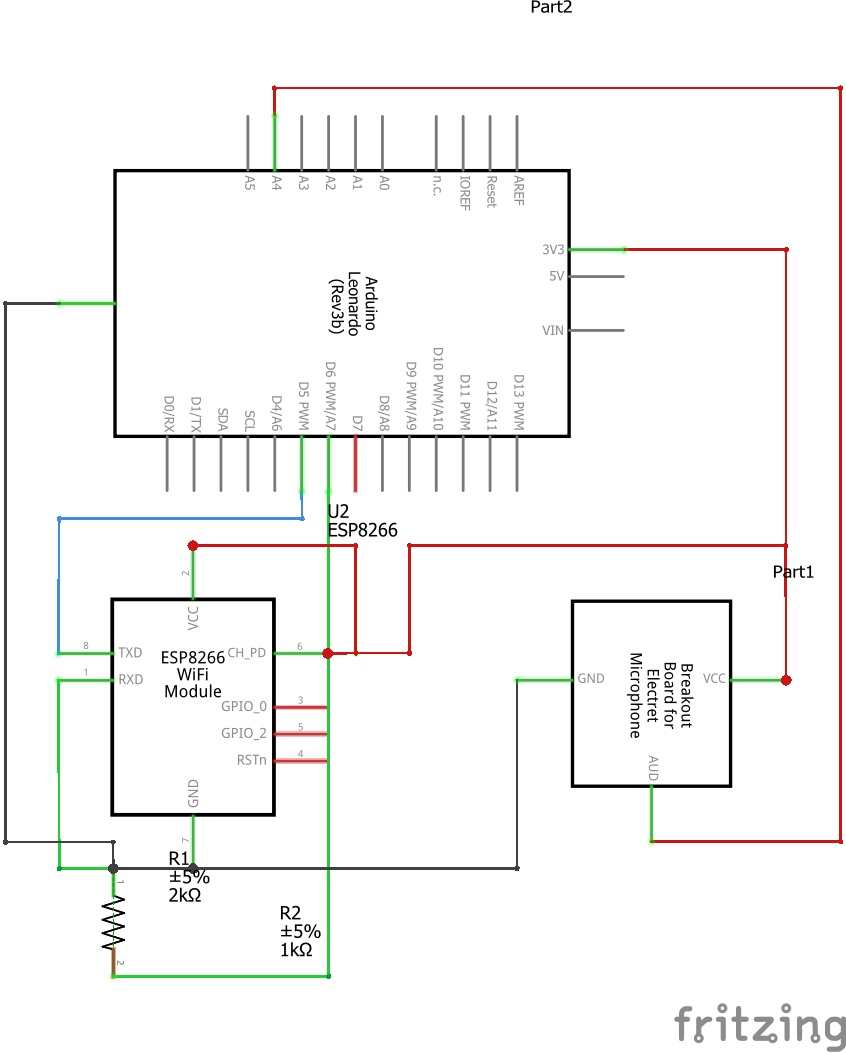
**הסבר סכמת מלבנים – מקלט האירוע**

1. הכפתור והחיישן קול משמשים כדי להעביר מידע אל הארדואינו ברגע שקורה אירוע.  
   החיישן קול שולח התראה על אירוע כאשר הרעש סביבו גדל יותר ממה שתוכנן מראש.  
   הכפתור שולח התראה על אירוע כאשר יש לחיצה עליו, שבעצם מתריע על אדם שמצלצל בפעמון הבית.
2. ארדואינו משמש כדי לקבל את המידע מהכפתור או מהחיישן קול, מציב תנאים כגון עוצמת הקול הניתנת מהחיישן קול ומעביר את המידע אל רכיב האינטרנט יחד עם פעולות מסודרות, כגון לאן לשלוח את המידע ומה לעשות איתו.
3. רכיב האינטרנט ESP8266 משמש כדי להעביר את המידע הודות האירוע אל האתר האינטרנט.  
   אתר האינטרנט מקבל את המידע ומאחסן את המידע שהתקבל אליו בתוך מסמך טקסט.

**סכמה חשמלית – שעון**



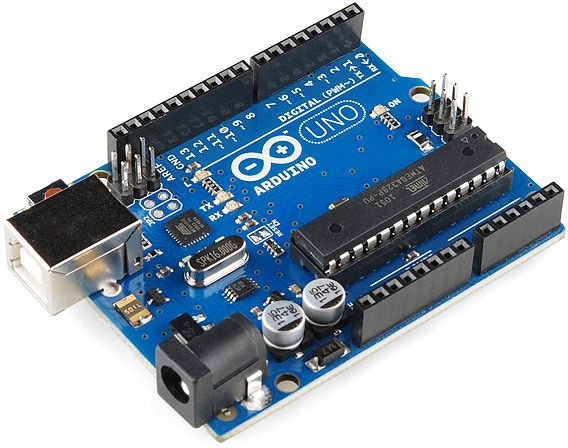
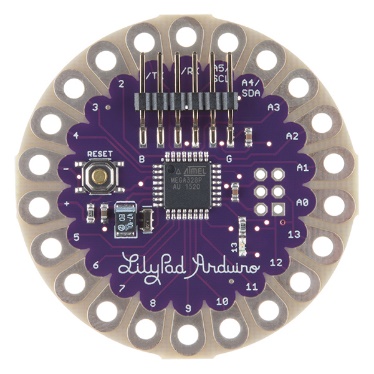
**סכמה חשמלית – מקלט האירוע**



**רכיבי הפרויקט**

* ארדואינו אונו
* ארדואינו Lilypad
* Oled breakout by Sparkfun
* Microphone Sensor AVR
* ESP8266
* Vibration micro motor
* Mini innovative single key

ארחיב על רכיבי הפרויקט, על אופן פעילותם, על תפקידם בפרויקט, ויתרונותם.

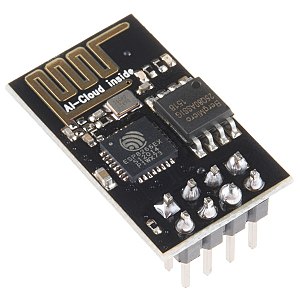


**Microphone Sensor AVR**

**Lilypad Arduino**

**Arduino uno**





**Mini innovative single key**

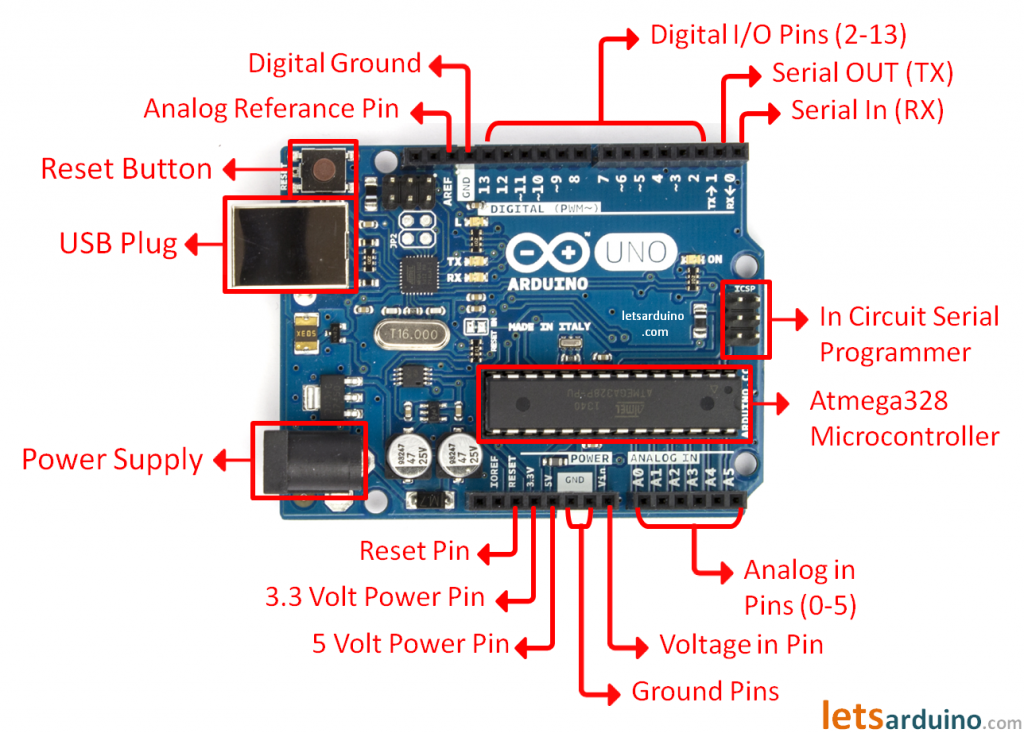
**Oled breakout**

**ESP8266**

**בקר – Arduino Uno**

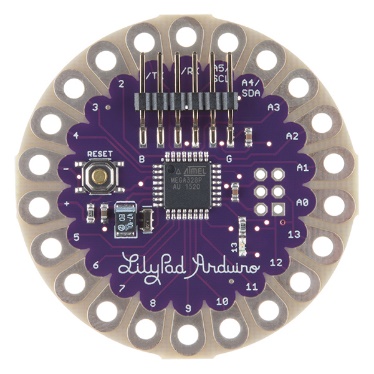
כרטיס הבקרה ארדואינו הוא רכיב אלקטרוני הניתן לתכנות שצורה קלה ונוחה לשימוש.

מבנה המיקרו בקר Arduino Uno:



* 14 הדקים דיגיטליים המשמשים ככניסות/ יציאות, כאשר 6 מתוכם הם הדקי PWM (מסומנים בגל על גבי הלוח) , הדק Rx והדק נוסף שלTx .
* 6 הדקים אנאלוגים
* 2 הדקי אדמה
* יציאת מתח – 3.3v , 5v
* רגל Reset- משמשת לאיפוס הבקר שהוא עובד איתו באותו הזמן שהוא עושה איפוס לעצמו (יכול לשמש לעבודה עם בקר נוסף).
* כפתור לאיפוס – Reset
* חיבור של USB, אשר מספק לו מתח של 5 וולט וגם מאפשר העברת נתונים.
* חיבור לסוללה ואו חשמל כזה או אחר.

**בקר – Arduino Lilypad**

מבנה המיקרו בקר Arduino Lilypad:

* 14 הדקים דיגיטליים המשמשים ככניסות / יציאות, כאשר 6 מתוכם הם הדקי PWM , רגל 0 משמשת כהדק RX ורגל 1 משמשת כהדק TX
* 6 הדקים אנאלוגים
* יציאת מתח
* הדק אדמה
* כפתור RESET לאיפוס

**תוכנה:**

המיקרו בקר פועל על ידי תכנות בשפה איחודית משלו אשר משלבת בין שפת C לשפת C++ והשימוש בו הוא חינמי. על מנת להקל על משתמשים חדשים קיימות תוכניות דוגמה רבות להתנסות.

מבנה התוכנית מתבקש לממש לפחות שתי פונקציות:

* פונקציית אתחול, שתתבצע כאשר הארדואינו מקבל חשמל.
* פונקציית לולאה, אשר מתבצעת מיד אחרי פונקציית האתחול, בלולאה אין סופית.

יתרון של לוח הפיתוח ארדואינו הוא שיש פורום גדול עם קהל גדול אשר נותן מענה אחד לשני בבעיות אשר אנשים נתקלים בהם בזמן העבודה עם המיקרו בקר.

דוגמה לקטע קוד בסיסי בארדואינו:

}Void setup ()  
pinMode (LED\_BUILTIN, OUTPUT); //  
}

הגדרת יציאה מספר 13 כפלט

Void loop () {  
digitalWrite (LED\_ BUILTIN, HIGH); //  
delay(1000); //  
digitalWrite (LED\_ BUILTIN, LOW); //  
delay(1000); //  
}

המתנה של שנייה

המתנה של שנייה

כיבוי הנורית על ידי הפסקת מתח ליציאה

הדלקת הנורית על ידי אספקת מתח ליציאה

תוכנית זאת גורמת להבהוב של נורית LED, המחוברת להדק מספר 13.

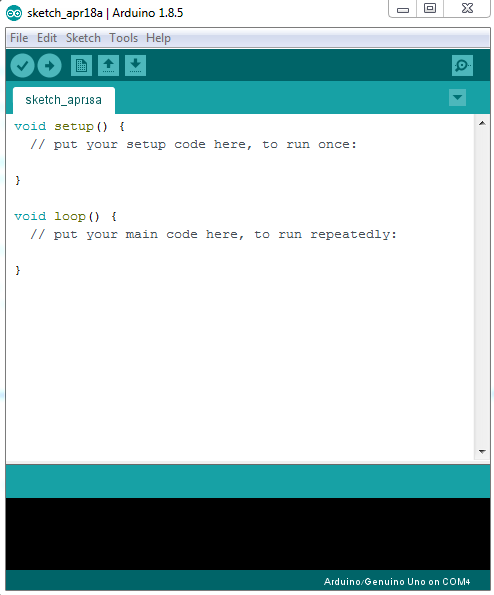
קיימים המון סוגים לארדואינו שאפשר לעבוד איתם, כולם משתמשים באותו הקוד אך בצורות שונות לנוחות של השימוש של כל אחד.

הלוחות המשמשים לפרויקט זה הם Arduino Uno ו Arduino Lilypad.

**דרכי שימוש בארדואינו:**

המיקרו בקר מגיע עם תוכנה איחודית אשר אפשר בקלות להוריד דרך האתר שלהם.

בפתיחת התוכנה, התוכנה תבנה את הבסיס לכתיבת הקוד למיקרו בקר. הכוללת את ה SETUP, מה שיריץ המיקרו בקר בעלייה שלו ויקרה רק פעם אחת. ואת ה LOOP שיפעל עם המיקרו בקר במחזוריות, פעם אחרי פעם ללא הפסקה.



בסיום כתיבת תוכן הקוד הרצוי, יש צורך בבדיקת שגיאות על ידי לחיצה על הכפתור הבא:



לאחר שהקוד נמצא ללא תקלות, יש לעלות את הקוד אל המיקרו בקר על ידי לחיצה על הכפתור הבא:



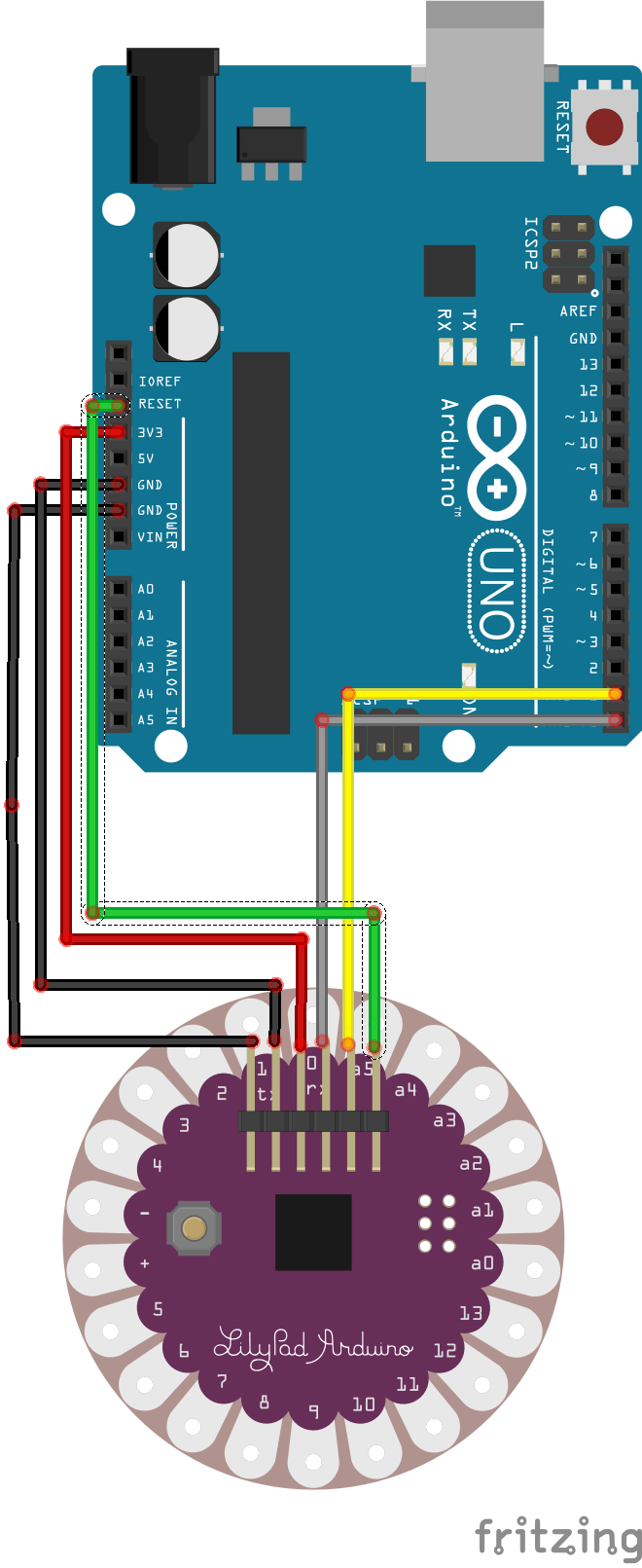
בסיום ההרצה ניתן יהיה לראות את כיצד השפיע תוכן הקוד על הבקר.

**העברת נתונים למיקרו בקר – Arduino Lilypad**

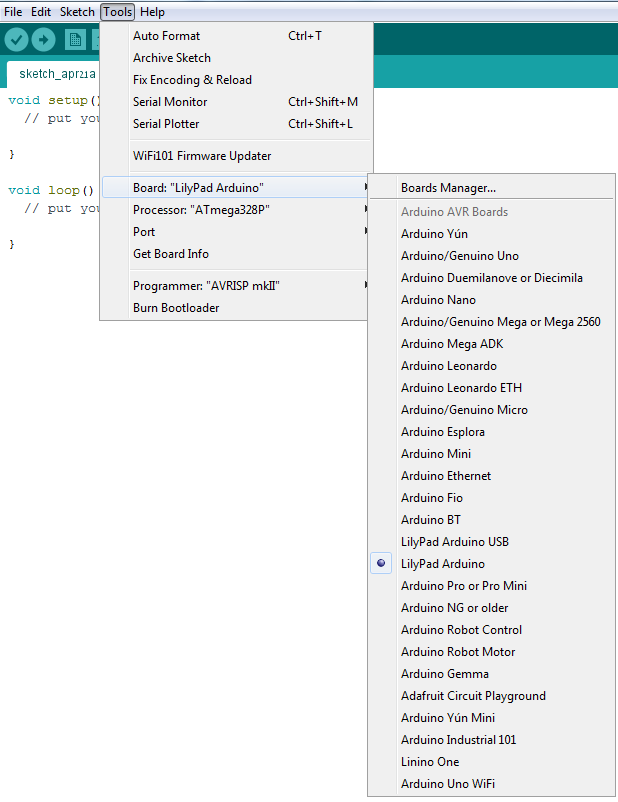
למיקרו בקר Arduino Lilypad אין חיבור USB למחשב, אשר נותן את האופציה לעלות לו את המידע, לכן על מנת לעלות מידע למיקרו בקר הנל, יש צורך בשימוש במיקרו בקר Arduino Uno.

בנוסף לרגליים של המיקרו בקר Arduino Lilypad ישנם 5 יציאות שעוד לא דיברתי עליהם והשימוש שלהם הוא העלת מידע אל המיקרו בקר.

יש לחבר את החיבורים הללו אל הArduino Uno בצורה הבאה:



כדי לעלות את התוכנית יש לוודה שעל ה Arduino Uno אין תוכנית כבדה משום שאחרת הוא לא יוכל להעביר את התוכנית שאנו רוצים להעביר למיקרו בקר. רצוי לפני החיבור לעלות אל ה Arduino uno תוכנית בסיסית שלא לוקחת כמעט מקום. לאחר החיבור וכתיבת התוכנית, צריך לשנות בהגדרות את העלאה אל Arduino Lilypad כמו בדוגמא הבאה:



**צג – Oled**

צג Oled הוא רכיב תאורה אשר מצייר כחול על גבי שחור את מה שנרצה, זאת על ידי פיצול המסך על ידי פיקסלים, וצביעתם לפי הצורך שלנו בהם.

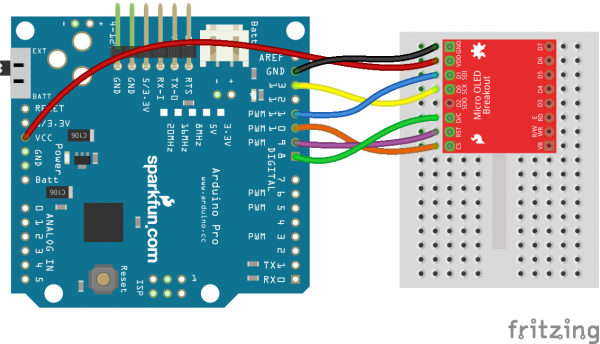
היתרונות של Oled לעומת LCD:

* כמות פיקסלים גדולה ושימושית עבור הפרויקט.
* מאפשר שימוש להצגת ציור, צג להודעה ואפילו לשניהם ביחד.
* מאפשר שימוש רחב בבחירת הפיקסלים אותם בוחרים להאיר, ובכך מגדיל את מספר אפשרויות התצוגה על גבי המסך.

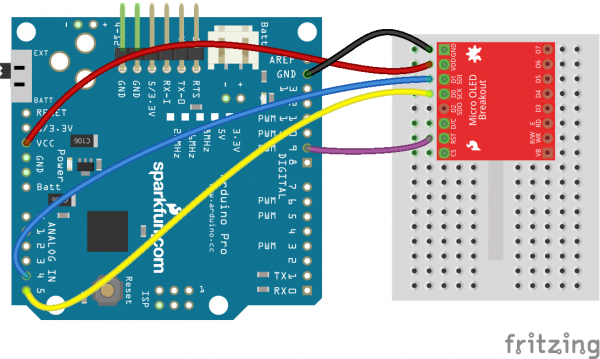
לOled יש 2 שיטות עיקריות לשימוש בארדואינו:

* I2C
* SPI

חיבור SPI:



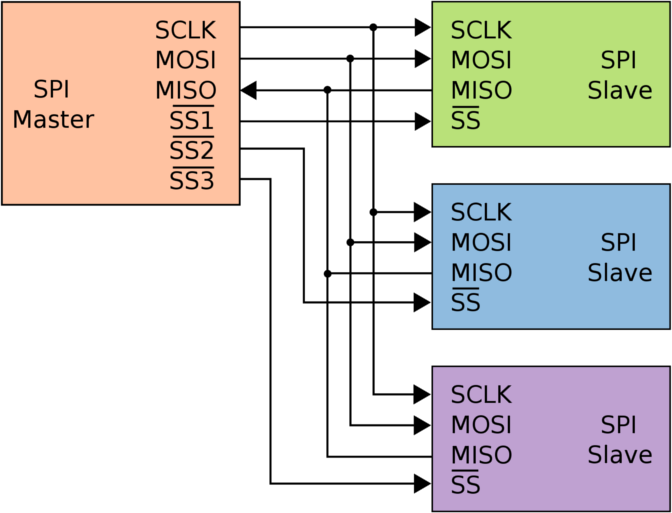
חיבור I2C:



SPI , I2C הם פרוטוקולים של תקשורת טורית אך יש בניהם הבדלים שונים.

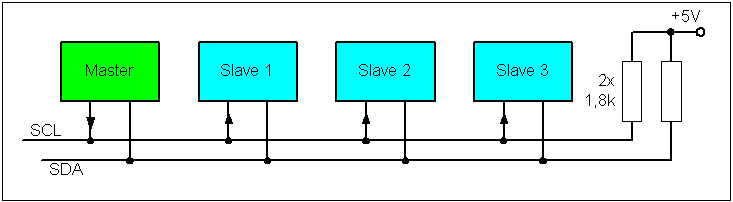
* SPI הוא בדרך כלל מהיר יותר.
* SPI נחשב פחות מתוחכם אבל עם זאת יותר פשוט להפעלה.
* I2C דורש יותר חוטים מ SPI.
* I2C יכול להיות יותר מסובך להפעלה מבחינת תכנות, אבל מבחינם חומרה הוא צורך רק 2 חוטים לתקשורת.

**חיבור SPI**



ה SPI צורך 4 חוטים, שמתוכם 1 מהחוטים הוא לפעמים יכול להיות כמה אבל אם אותו השימוש והוא הSS, כפי שאפשר לראות באיור יש 3 חיבורי SS.  
SCLK – קו שעון תזמון משותף לכל הרכיבים, כדי שהמידע יעבור בסנכרון בניהם הוא משתמש בקו שעון.  
MOSI – מידע שיוצא מה MASTER אל ה Slave, אפשר להבין שבפרויקט ה Master הוא הארדואינו והSlave זה ה Oled.  
MISO – מידע שיוצא מה Slave אל ה MASTER. במקרה של הפרויקט השימוש של הרגל הזאת הוא לא שימושי, מכיוון שהOLED מקבל מידע אך הוא לא מחזיר מידע.  
SS – Slave Select – בוחר מבין כל הרכיבים המחוברים את הSlave אליו הוא רוצה לתקשר באותו הרגע או ממי הוא רוצה לקבל מידע.

**חיבור I2C**



אפשר לראות כבר מהאיור את השינוי בין כמוך החוטים הדרוש עבור I2C ועבור SPI.  
הארדואינו, ה Master מתחיל שליחת נתונים וכל שאר הרכיבים ה Slaves מקשיבים לנתונים, לכל אחד מהם יש מספר זיהוי איחודי משלו (כתובת) שמבדיל אותו משאר הרכיבים. אם התקשורת מופנית לכתובת של הרכיב הוא מגיב לפי הקוד, ובדרך כלל שולח נתונים חזרה לארדואינו.  
ה I2C מחובר בעזרת 2 חוטים:

SCL – קו שעון שאחראי לתזמון של העברת הנתונים.

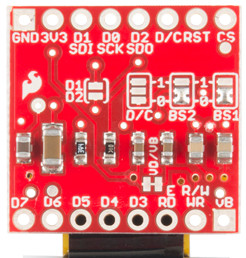
SDA – אחראי על העברת הנתונים, רוב הזמן משמש בתור העברה מהארדואינו לרכיבים, אך הוא משמש גם כדי לאשר העברה בינו לבין הרכיבים שהוא מתקשר איתם, על ידי משוב מהרכיב שאליו רצה שיתקבל המידע.

**סדר האירועים:**

* הארדואינו שולח ביט להתחלת התקשורת.
* הארדואינו שולח כתובת של 7 ביט שמופנית לאחד הרכיבים.
* הארדואינו שולח קריאה או כתיבה כדי להגדיר את הפעולה שהוא רוצה.
* הרכיב שולח ביט חזרה שקיבל את ההודעה.
* הארדואינו כותב או קורא מהרכיב על ידי שליחת בייט אחד אחרי השני, הרכיב מודיע שקיבל את הבייט אחרי כל שליחה.
* לסיום הארדואינו שולח ביט לסגירת התקשורת.

**Breakout Board**

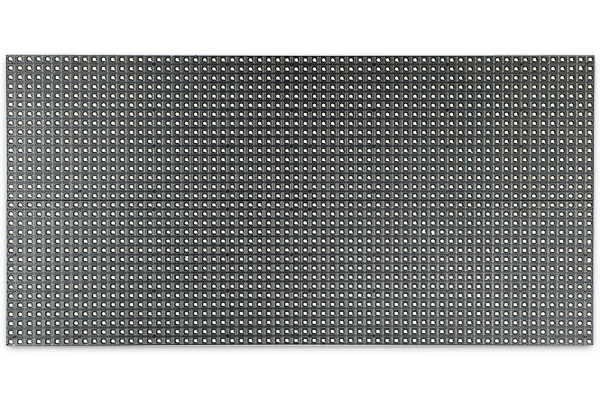
Breakout Board הוא החלק המחבר בין הארדואינו לבין הצג. החלק זה מקבל נתונים מהארדואינו, ולפי הנתונים שהתקבלו הוא צובע את הפיקסלים הרצויים.



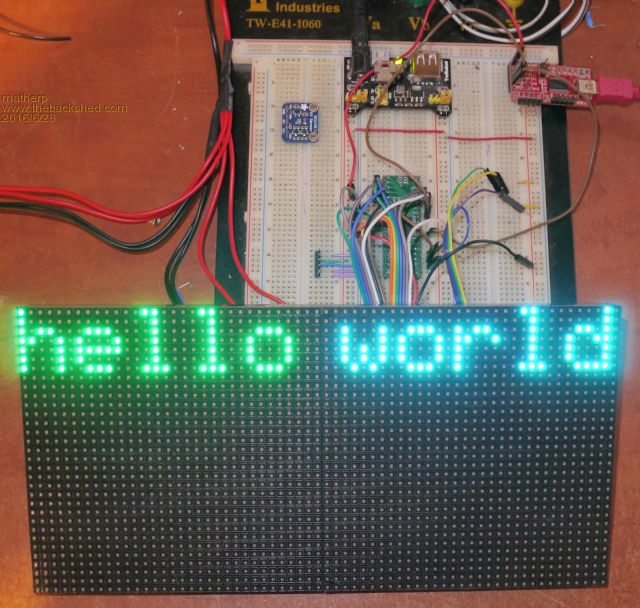
* GND – רגל האדמה.
* 3V3 – חיבור ה3 וולט שמספקים לצג.
* D1 SDI – Serial Data In – המידע המתקבל מהארדואינו.
* D0 SCK – הוא הclock עבור השיטות של ה I2C וה- SPI.
* D2 SDO – רגל לשימוש רק בשיטת SPI, הוא משמש למידע מהצג אל הארדואינו.
* D/C – זה רגל דיגיטלית שתפקידה הוא לשלוח הודעה לצג האם הפקודה שמגיע כרגע היא פקודה או מידע לעלות על הצג. (לדוגמא האם למחוק את כל המידע שמוצג במסך או האם להוסיף למסך עוד חלק צבועה).
* CS – זה עבור SPI – משמש ב CHIP SELECT – מחליט מתי הצג מחזיר תשובה ומתי הוא מקבל, עובד בנמוך. עובד בשיטת Master and Slave.

**דרך הפעולה של הOled**

הOled - מחולק לפיקסלים, בפרויקט נעשה שימוש בצג בעל 64 על 32 פיקסלים, כמו שמופיע בדוגמא הבאה:

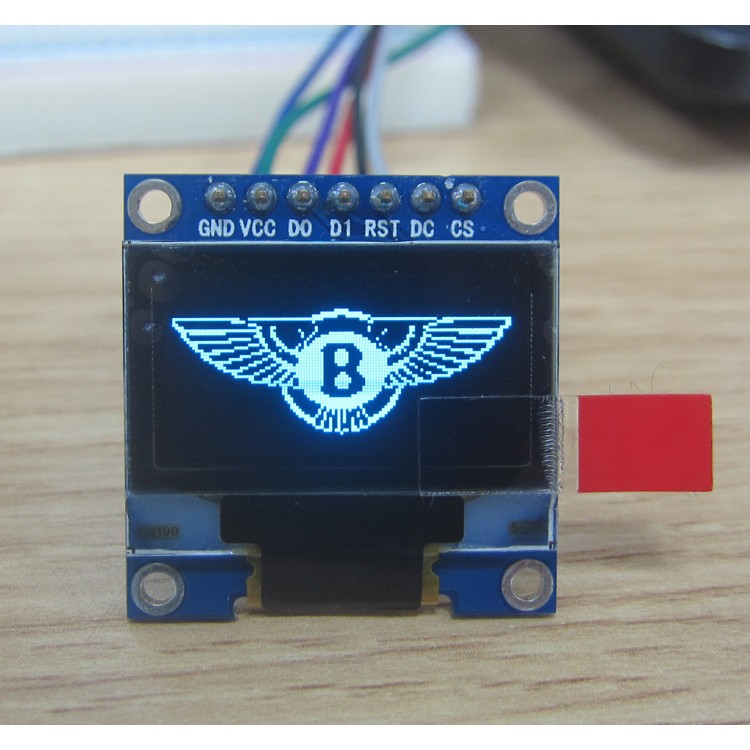


דוגמא להצגת המילים hello world:



באיור ניתן לראות כי הפיקסלים המוארים במסך מייצרים את המילים אותן בחר המשתמש לכתוב, במקרה זה יצרו את המילים hello world. כל מסך בעולם, ממסך טלויזיה, ועד מסך מולטימדיה ברדיו עובד באותה הצורה, הוא מחולק לכמות פיקסלים, והפיקסלים נצבעים לפי הצורך בתצוגה של המשתמש. ככל שיש יותר פיקסלים הצג יהיה יותר אמין.

בנוסף אפשר ליצור לוגו כמו בדוגמא הבאה:



**חיישן הקול – Microphone Sensor AVR**

חיישן הקול Microphone sensor avr, הוא חיישן המשנה את התנגודותו עקב הקול הנכנס אליו בכניסתו, המקרופון. ככל שהקול גבוה יותר התנגדות החיישן קטנה יותר, והמתח שהוא מוציא גדול יותר.

יתרונות של רכיב זה:

* קל לשימוש, מחברים לו מתח והוא מוציא מתח לפי הקול הנכנס אליו.
* זול, אפשר לקנות אותו במחיר של פחות מדולר.
* קל לחיבור, יש לו 4 רגלים בסך הכל, 2 רגלים של יציאה אחת בצורה אנלוגית ואחת דיגיטלית והבחירה היא שלנו במה להשתמש.

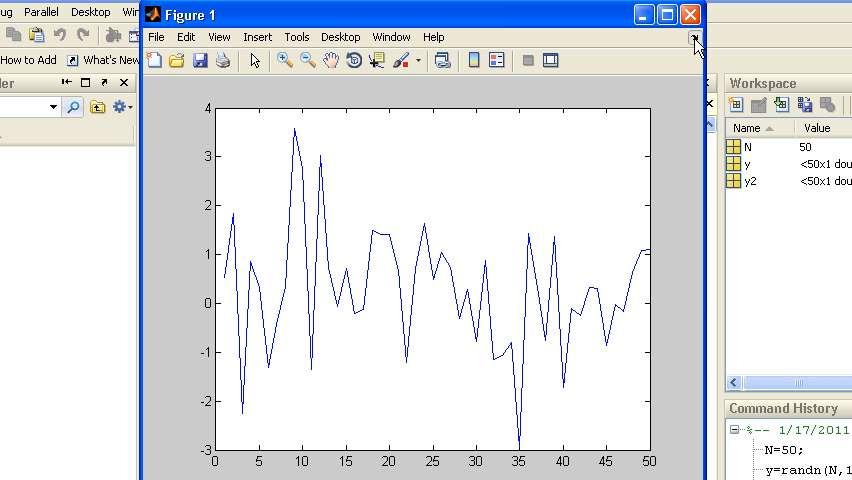
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
**חיבורי החיישן**

לחיישן יש 4 רגליים בלבד.

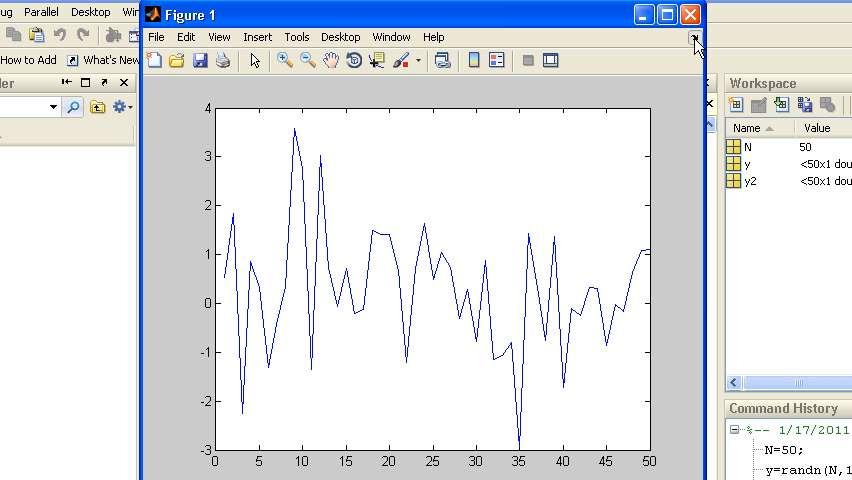
* AO – רגל יציאה אנלוגי, שולח מידע כפונקציה של המתח הנכנס לחיישן ולהתנגדות הקיימת. בדרך כלל בין 0-124.
* G – רגל אדמה.
* + - רגל כניסת מתח, רצוי 3.3 וולט.
* DO – רגל יציאה דיגיטלית, שולח מתח לפי ההתנגדות והמתח היוצא. בדרך כלל משמש בתצורה של HIGH או LOW. הכוונה שאו שהוא עובד או שהוא לא לפי המיקרופון, אך בגלל שתמיד יש רעש בבית אי אפשר להשתמש בו.

**אופן הפעולה**

החיישן שולח ביציאה אנלוגית מתח המשתנה לפי הרעש המגיעה. זה נראה כך:



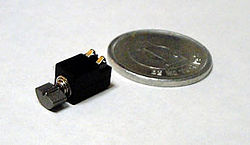
אנו מסתכלים על המידע שמגיע מהחיישן, ומזהים שהקול שאנו רוצים להשתמש בו הוא כאשר המתח גדול מ2 וחצי וולט. לכן אנו מציבים לו תנאי שכאשר הוא עובר את הקלוט הרצוי , במקרה שלנו 2 וחצי וולט. הוא יעשה מה שנרצה. ממש כך:



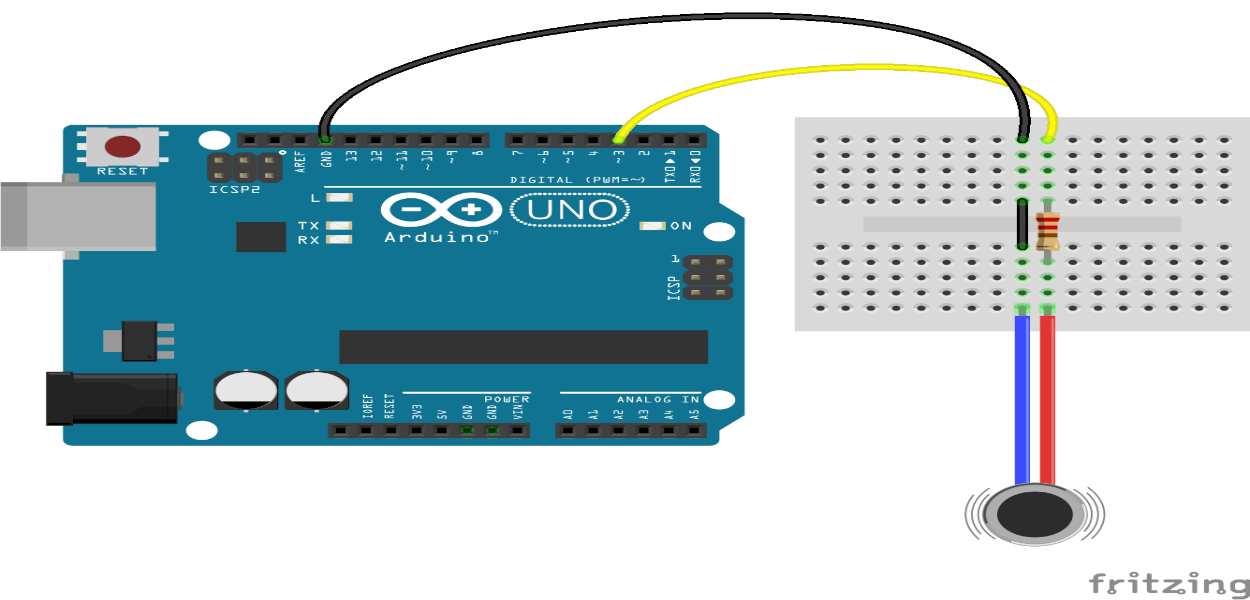
אפשר לראות שהוא חוצה את הקו האדום פעמיים. לכן הפעולה שנרצה, תתבצע פעמיים.

**מנוע מיקרו רטט - Vibration micro motor**

מנוע מיקרו הרטט הוא רכיב המשמש כדי להתריע למשתמש השעון על אירועים המתרחשים בבית, בעזרת משיכת צומת ליבו בעזרת רטט בידו.  
הרטט בנוי עם 2 רגליים, רגל מתח ורגל אדמה. כאשר מחברים את הרטט למקור מתח, בתוך הרכיב נמצא מנוע שלצירו מחוברת טבעת אובלית, שכאשר הציר מסתובב ויחד איתו מסובב את הטבעת נוצר תנועות זעירות היוצרות רטט במכשיר.



**חיבור הרטט**

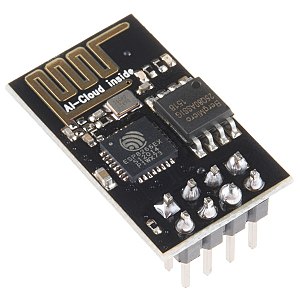


לרטט 2 רגליים:  
רגל אדומה- רגל המתח.  
ורגל כחולה – רגל האדמה.  
אפשר לראות החיבור רגל המתח מתחבר אל רגל 3, ועל מנת להפעיל את הרטט יש להוציא מתח מרגל 3, כדי שלא יעבוד הרטט רגל 3 לא תוציא מתח.

**רכיב האינטרנט – ESP8266**

רכיב האינטרנט ESP8266 הוא רכיב המגשר בין הפרויקט לאינטרנט, הוא הרכיב המתחבר לרשת האינטרנט הביתי, ומקבל או שולח נתונים לאתר האינטרנט, על פי הצורך.  
לרכיב ESP8266 יש 3 מצבי עבודה, אותם אפשר להגדיר דרך העבודה עם הארדואינו בקלות.  
מצב 1 – HOST – זהו נקרא גם צד שרת, הוא יוצר את האתר ומאחסן אותו על עצמו, בנוסף כתובת האתר נקבעת על פי הכתובת של האינטרנט אליו הוא מחובר. השימוש העיקרי במצב הזה הוא כאשר אין אתר ואחסון כדי לאחסן את הנתונים שהמשתמש רוצה לאחסן, ולכן המצב הזה יותר אתר מאוחסן לגמרי ללא תשלום על כתובת בינלאומית ועל אחסון חיצוני. החסרונות שלו הם שכאשר הוא פותח את עצמו לעולם בצורה כזאת, יש אפשרות לעולם לגשת לכל הרשת אליו הוא משדר ממנה. לכן השימוש בו הוא בסיכון ויש להשתמש בהמון אמצעי הגנה בזמן השימוש בו. בנוסף רוב הרשתות לא פתוחות לעולם והן חוסמות גישה אליהן לכן הגישה של העולם לאתר שהוא פותח לא תמיד מתאפשרת.  
מצב 2 – CLIENT – זהו נקרא גם צד לקוח, הוא פונה לאתר בנוי כבר, לדוגמא האתר שהפרויקט יושב עליו, ושולף ממנו מידע או שולח אליו מידע על פי הרצון של המשתמש. הוא לא יכול לשנות דברים חיוניים באתר משום שהוא לא זה שפתח את האתר. אך הוא יכול לקבל מידע ולשלוח לפי הצורך שלו והאתר יכול לשנות את עצמו על פי הנתונים המתקבלים ממנו. השיטה הזאת היא מומלצת ביותר בשימוש כמו בפרויקט הזה, מכיוון שרצון הוא להיות מחוברים לאינטרנט לאתר עם כתובת בינלאומית, פתוחה לעולם, ולקבל מידע בנוגע למה שקורה בבית בכל רגע נתון, ללא שינויים באתר בכלל. האתר משמש בתור מעבר בין הקולט לשעון ובכך הוא משמש בצורה הטובה ביותר.

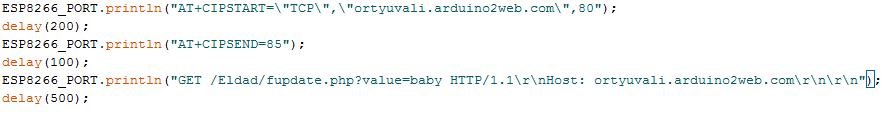
מצב 3 – Host and Client – המצב המשולב כאשר האתר יכול לשלב בין אירוח לבין שימוש. הכוונה היא שהאתר יכול להפתח על ידי הרכיב. אבל בו זמנית יש אפשרות לקבל נתונים מאתרים אחרים ולעדכן את האתר שבפתח הרכיב על פי המידע שהתקבל. השיטה הזאת היא לוקחת המון זכרון והשימוש בה הוא רק כאשר זה נחוץ ביותר. לא מומלצת וחשוב לזכור שכמות המידע שיכול לאחסן הרכיב הוא מוגבל.



**השימוש ברכיב האינטרנט - ESP8266**

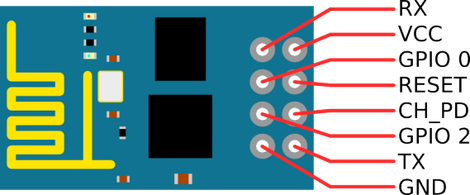
יש המון שיטות לקודד את הרכיב, אך בפרויקט זה נעשה על ידי שימוש בספרייה קיימת בארדואינו, SoftwareSerial. הספרייה הזאת מאפשרת לנו להשתמש בפקודות AT Commands על מנת להשתמש ברכיב.  
על מנת להגדיר את המצב של הרכיב ולהשתמש בו, יש צורך להגדיר תחילה את השימוש ברכיב, איזה מצב אנו רוצים להשתמש בו מבין ה3 מצבים, וזאת על ידי השימוש בפקודה הבאה:

AT+CWMODE=?  
יש להוסיף מספר אחרי השווה במקום הסימן שאלה, המספר הוא המצב שישמש כדי שהרכיב ידע איך לפעול. מספרי המצבים נמצאים בעמוד הקודם.  
שלב הבא הוא התחברות לרשת האינטרנט המקומית על ידי שם הרשת, וסיסמת ההתחברות אליה.  
AT+CWJAP=ssid,pwd  
יש לשנות את המשתנים לפי הצורך, ssid – שם הרשת המקומית, pwd – סיסמת הרשת.  
כרגע הרכיב מוכן לשימוש ומחובר לרשת המקומית. הינה דוגמא לשימוש ברכיב בצורת CLIENT.



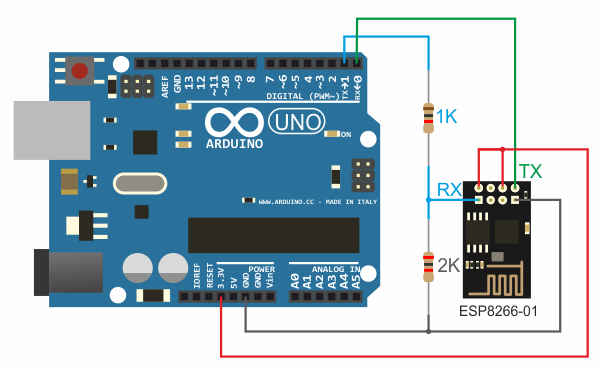
בדוגמא אפשר לראות שהרכיב שולח מידע אל האתר, המידע שהוא שולח זה baby.

**רגלי הרכיב אינטרנט – ESP8266**



* VCC – רגל המתח, 3.3v
* GND – רגל האדמה
* RX – רגל לקבלת המידע – DATA Reciever pin
* TX – רגל לשליחת המידע – DATA Transfer pin
* CH\_PD – Chip power down pin – משמש כדי להפעיל את החיישן כאשר הוא מקבל מתח הוא עובד, וכאשר לא הוא לא עובד. יש לחבר אותו יחד עם הVCC למתח.
* RESET – רגל איפוס הרכיב, על מנת שהוא יתאפס יש לחבר אותו לאדמה, משום שהוא עובד בנמוך. כדי שלא יעבוד יש צורך לא לחבר אותו לשום מקום. משמש רק כאשר הוא ה HOST ובפרויקט הזה אין בו שימוש.
* GPIO 0, Gpio 2, משמשים כדי לקבל מידע בצורה דיגיטלית, לדוגמא HIGH או LOW משמש בעיקר לשאלות של כן או לא, השימוש בו הוא רק כאשר הוא הHOST ובפרויקט הזה אין בו שימוש.

חיבור בין הרכיב לארדואינו



אדום – רגל הVCC , ורגל ה CH\_PD, מחוברים אל המתח 3.3v שבארדואינו.

שחור – רגל האדמה מחוברת אל האדמה של הארדואינו.

ירוק – TX – רגל לשליחת המידע מחובר לרגל 0 של הארדואינו, רגל 0 בארדואינו זה רגל קבלת המידע של הארדואינו.

כחול – RX – רגל לקבלת מידע, מחובר לנגדים כדי שהמתח המתקבל מהארדואינו 5v ישתנה בעזרת חלוקת מתח ויתקבל 3.3v לרגל קבלת המידע. נוספת הרגל מחוברת בצד אחד אל האדמה, ובצידו השני לרגל 1 של הארדואינו, שזאת הרגל לשליחת המידע של הארדואינו.

**תוכנה**

**קוד מקלט האירוע**

#include <SoftwareSerial.h>

#define ESP8266\_BAUD 9600

#define SS\_RX 5

#define SS\_TX 6

SoftwareSerial ESP8266\_PORT(SS\_RX, SS\_TX);

const String ESP8266\_APNAME = "Eldad";

const String ESP8266\_APPSWD = "12345678";

String ESP8266\_CMD;

int CheckButton = 7;

int voicepin = A4;

int analogval = 0;

void setup()

{

pinMode(voicepin, INPUT);

pinMode(CheckButton, INPUT);

Serial.begin(9600);

ESP8266\_PORT.begin(ESP8266\_BAUD);

delay(1000);

ESP8266\_Init();

}

void ESP8266\_Init()

{

String WWWResponse;

Serial.println("Connecting to WiFi (" + ESP8266\_APNAME + ", " + ESP8266\_APPSWD + "), please wait...");

ESP8266\_Send("AT+CWMODE=1", 100, false);

WWWResponse = ESP8266\_Send("AT+CWJAP=\"" + ESP8266\_APNAME + "\",\"" + ESP8266\_APPSWD + "\"", 5000, false);

if (find\_text("CONNECTED", WWWResponse) == 0)

{

Serial.println("Error: can't connect to AP.");

while (1);

}

else

{

Serial.println("Ready!");

while (Serial.available())

Serial.read();

}

}

void loop() {

if (ESP8266\_PORT.available()) {

int inByte = ESP8266\_PORT.read();

Serial.write(inByte);

}

ESP8266\_Connect();

}

void ESP8266\_Connect()

{

analogval = analogRead(voicepin);

Serial.println(analogval);

if (analogval > 68) {

sendata("baby");

}

else if(digitalRead(CheckButton) == HIGH){

sendata("door");

}

}

void sendata(String problam){

String ESP8266\_CMD, WWWResponse;

Serial.println(String(problam));

ESP8266\_Send("AT+CIPSTART=\"TCP\",\"ortyuvali.arduino2web.com\",80", 1000, false);

ESP8266\_CMD="GET /Eldad/fupdate.php?value=" + String(problam) + " HTTP/1.1\r\nHost: ortyuvali.arduino2web.com\r\n\r\n";

ESP8266\_Send("AT+CIPSEND="+String(ESP8266\_CMD.length()), 1000, false);

ESP8266\_Send(ESP8266\_CMD, 1000, false);

}

String ESP8266\_Send(String command, const int timeout, boolean debug)

{

String response = "";

ESP8266\_PORT.println(command);

long int time = millis();

while ( (time + timeout) > millis())

{

while (ESP8266\_PORT.available())

{

char c = ESP8266\_PORT.read();

response += c;

}

}

if (debug)

{

Serial.print(response);

}

return response;

}

int find\_text(String needle, String haystack) {

int foundpos = 0;

for (int i = 0; i <= haystack.length() - needle.length(); i++) {

if (haystack.substring(i, needle.length() + i) == needle) {

foundpos = i;

}

}

return foundpos;

}

קוד שעון

**#include <SoftwareSerial.h>**

**#include <Wire.h>**

**#include <SPI.h>**

**#include <SFE\_MicroOLED.h>**

**#include <avr/pgmspace.h>**

**#define PIN\_RESET 9**

**#define PIN\_DC 8**

**#define PIN\_CS 10**

**#define DC\_JUMPER 0**

**MicroOLED oled(PIN\_RESET, PIN\_DC, PIN\_CS);**

**#define ESP8266\_BAUD 9600**

**#define SS\_RX 5**

**#define SS\_TX 6**

**SoftwareSerial ESP8266\_PORT(SS\_RX, SS\_TX);**

**#include "TimerOne.h"**

**#define ONBOARD\_LED 12**

**#define button 2**

**#define led 13**

**#define bazzer 7**

**const String ESP8266\_APNAME = "Eldad";**

**const String ESP8266\_APPSWD = "12345678";**

**String ESP8266\_CMD;**

**int CheckButton;**

**int hour=0;**

**int minute=0;**

**int second=0;**

**int b=0;**

**void setup()**

**{**

**pinMode(led,OUTPUT);**

**pinMode(button,INPUT);**

**pinMode(bazzer,OUTPUT);**

**pinMode(ONBOARD\_LED, OUTPUT);**

**digitalWrite(bazzer,HIGH);**

**ISR\_Init(10000000);**

**Serial.begin(9600);**

**ESP8266\_PORT.begin(ESP8266\_BAUD);**

**delay(1000);**

**ESP8266\_PORT.println("AT+CIOBAUD=9600");**

**ESP8266\_PORT.end();**

**delay(1000);**

**ESP8266\_PORT.begin(9600);**

**delay(1000);**

**ESP8266\_Init();**

**oled.begin();**

**oled.clear(ALL);**

**oled.display();**

**}**

**void ESP8266\_Init()**

**{**

**String WWWResponse;**

**Serial.print(F("Connecting to WiFi ("));**

**ESP8266\_Send("AT+CWMODE=1", 100, false);**

**WWWResponse=ESP8266\_Send("AT+CWJAP=\"" + ESP8266\_APNAME + "\",\"" + ESP8266\_APPSWD + "\"", 5000, false);**

**if(find\_text("CONNECTED", WWWResponse) == 0)**

**{**

**Serial.println(F("Error: can't connect to AP."));**

**Timer1.stop();**

**digitalWrite(ONBOARD\_LED, 0);**

**while(1);**

**}**

**else**

**{**

**Serial.println(F("Ready!"));**

**while(Serial.available())**

**Serial.read();**

**}**

**}**

**void loop() {**

**if (ESP8266\_PORT.available()) {**

**int inByte = ESP8266\_PORT.read();**

**Serial.write(inByte);**

**}**

**ESP8266\_Connect();**

**}**

**void ESP8266\_Connect()**

**{**

**String ESP8266\_CMD, WWWResponse;**

**if(b==0){**

**String hour1;**

**String minute1;**

**String second1;**

**ESP8266\_Send("AT+CIPSTART=\"TCP\",\"ortyuvali.arduino2web.com\",80", 1000, false);**

**ESP8266\_CMD="GET /Eldad/watch/time.txt HTTP/1.1\r\nHost: ortyuvali.arduino2web.com\r\n\r\n";**

**ESP8266\_Send("AT+CIPSEND="+String(ESP8266\_CMD.length()), 1000, false);**

**WWWResponse=ESP8266\_Send(ESP8266\_CMD, 1000, true);**

**int idPos;**

**for (int i = 0; i <= WWWResponse.length() - 4; i++) {**

**if (WWWResponse.substring(i,4+i) == "hour") {**

**idPos = i;**

**}**

**}**

**if(WWWResponse.substring(idPos, idPos+4) == "hour")**

**{**

**hour1 = WWWResponse.substring(idPos+5,idPos+7);**

**minute1 = WWWResponse.substring(idPos+8,idPos+10);**

**second1 = WWWResponse.substring(idPos+11,idPos+13);**

**b=5;**

**ISR\_Init(1000000);**

**while(Serial.available())**

**Serial.read();**

**}**

**hour = hour1.toInt();**

**minute = minute1.toInt();**

**second = second1.toInt();**

**}**

**if (b==5){**

**ESP8266\_Send("AT+CIPSTART=\"TCP\",\"ortyuvali.arduino2web.com\",80", 1000, false);**

**ESP8266\_CMD="GET /Eldad/value.txt HTTP/1.1\r\nHost: ortyuvali.arduino2web.com\r\n\r\n";**

**ESP8266\_Send("AT+CIPSEND="+String(ESP8266\_CMD.length()), 1000, false);**

**WWWResponse=ESP8266\_Send(ESP8266\_CMD, 1000, true);**

**int idPos=0;**

**for (int i = 0; i <= WWWResponse.length() - 4; i++) {**

**if (WWWResponse.substring(i,4+i) == "baby") {**

**idPos = i;**

**}**

**else if (WWWResponse.substring(i,4+i) == "door") {**

**idPos = i;**

**}**

**}**

**oled.begin();**

**oled.clear(ALL);**

**oled.setFontType(0);**

**if(WWWResponse.substring(idPos, idPos+4) == "baby")**

**{**

**digitalWrite(led,HIGH);**

**digitalWrite(bazzer,LOW);**

**CheckButton = digitalRead(button);**

**while(CheckButton<1){**

**CheckButton = digitalRead(button);**

**oled.clear(PAGE);**

**oled.setCursor(LCDWIDTH/5, LCDHEIGHT/2);**

**oled.print("Baby CRY");**

**oled.line((LCDWIDTH/5)-5, (LCDHEIGHT/2)+10, LCDWIDTH-3, (LCDHEIGHT/2)+10);**

**oled.line((LCDWIDTH/5)-5, (LCDHEIGHT/2)-5, LCDWIDTH-3, (LCDHEIGHT/2)-5);**

**oled.line(LCDWIDTH-3, (LCDHEIGHT/2)+10, LCDWIDTH-3, (LCDHEIGHT/2)-5);**

**oled.line((LCDWIDTH/5)-5, (LCDHEIGHT/2)+10, (LCDWIDTH/5)-5, (LCDHEIGHT/2)-5);**

**oled.display();**

**}**

**sendata();**

**while(Serial.available())**

**Serial.read();**

**}**

**else if(WWWResponse.substring(idPos, idPos+4) == "door")**

**{**

**digitalWrite(led,HIGH);**

**digitalWrite(bazzer,LOW);**

**CheckButton = digitalRead(button);**

**while(CheckButton<1){**

**CheckButton = digitalRead(button);**

**oled.clear(PAGE);**

**oled.setCursor(LCDWIDTH/5, LCDHEIGHT/2);**

**oled.print(" Door");**

**oled.line((LCDWIDTH/5)-5, (LCDHEIGHT/2)+10, LCDWIDTH-3, (LCDHEIGHT/2)+10);**

**oled.line((LCDWIDTH/5)-5, (LCDHEIGHT/2)-5, LCDWIDTH-3, (LCDHEIGHT/2)-5);**

**oled.line(LCDWIDTH-3, (LCDHEIGHT/2)+10, LCDWIDTH-3, (LCDHEIGHT/2)-5);**

**oled.line((LCDWIDTH/5)-5, (LCDHEIGHT/2)+10, (LCDWIDTH/5)-5, (LCDHEIGHT/2)-5);**

**oled.display();**

**}**

**sendata();**

**while(Serial.available())**

**Serial.read();**

**}**

**else**

**digitalWrite(led,LOW);**

**digitalWrite(bazzer,HIGH);**

**}**

**}**

**void sendata(){**

**Serial.println(F("Sending off"));**

**ESP8266\_PORT.println(F("AT+CIPSTART=\"TCP\",\"ortyuvali.arduino2web.com\",80"));**

**delay(250);**

**ESP8266\_PORT.println(F("AT+CIPSEND=83"));**

**delay(100);**

**ESP8266\_PORT.println(F("GET /Eldad/fupdate.php?value=off HTTP/1.1\r\nHost: ortyuvali.arduino2web.com\r\n\r\n"));**

**delay(500);**

**}**

**String ESP8266\_Send(String command, const int timeout, boolean debug)**

**{**

**String response = "";**

**ESP8266\_PORT.println(command);**

**long int time = millis();**

**while( (time+timeout) > millis())**

**{**

**while(ESP8266\_PORT.available())**

**{**

**char c = ESP8266\_PORT.read();**

**response+=c;**

**}**

**}**

**if(debug)**

**{**

**Serial.print(response);**

**}**

**return response;**

**}**

**int find\_text(String needle, String haystack) {**

**int foundpos = 0;**

**for (int i = 0; i <= haystack.length() - needle.length(); i++) {**

**if (haystack.substring(i,needle.length()+i) == needle) {**

**foundpos = i;**

**}**

**}**

**return foundpos;**

**}**

**void ISR\_Init(unsigned long ISR\_INTERVAL)**

**{**

**Timer1.initialize(ISR\_INTERVAL);**

**Timer1.attachInterrupt(ISR\_callback);**

**}**

**void ISR\_callback()**

**{**

**if (b==5){**

**oled.begin();**

**oled.clear(ALL);**

**oled.setFontType(0);**

**if (hour<12 && hour!=11){**

**if (minute<60 && minute!=59){**

**if (second<60 && second!=59){**

**oled.clear(PAGE);**

**oled.setCursor(LCDWIDTH/5, LCDHEIGHT/2);**

**oled.print(hour); oled.print(":");**

**oled.print(minute); oled.print(":");**

**oled.print(second);**

**oled.line((LCDWIDTH/5)-5, (LCDHEIGHT/2)+10, LCDWIDTH-3, (LCDHEIGHT/2)+10);**

**oled.line((LCDWIDTH/5)-5, (LCDHEIGHT/2)-5, LCDWIDTH-3, (LCDHEIGHT/2)-5);**

**oled.line(LCDWIDTH-3, (LCDHEIGHT/2)+10, LCDWIDTH-3, (LCDHEIGHT/2)-5);**

**oled.line((LCDWIDTH/5)-5, (LCDHEIGHT/2)+10, (LCDWIDTH/5)-5, (LCDHEIGHT/2)-5);**

**oled.display();**

**second++;**

**}**

**else if(second == 59){**

**oled.clear(PAGE);**

**oled.setCursor(LCDWIDTH/5, LCDHEIGHT/2);**

**oled.print(hour); oled.print(":");**

**oled.print(minute); oled.print(":");**

**oled.print(second);**

**oled.line((LCDWIDTH/5)-5, (LCDHEIGHT/2)+10, LCDWIDTH-3, (LCDHEIGHT/2)+10);**

**oled.line((LCDWIDTH/5)-5, (LCDHEIGHT/2)-5, LCDWIDTH-3, (LCDHEIGHT/2)-5);**

**oled.line(LCDWIDTH-3, (LCDHEIGHT/2)+10, LCDWIDTH-3, (LCDHEIGHT/2)-5);**

**oled.line((LCDWIDTH/5)-5, (LCDHEIGHT/2)+10, (LCDWIDTH/5)-5, (LCDHEIGHT/2)-5);**

**oled.display();**

**second=00;**

**minute++;**

**}**

**}**

**else if(minute == 59){**

**if (second<60 && second!=59){**

**oled.clear(PAGE);**

**oled.setCursor(LCDWIDTH/5, LCDHEIGHT/2);**

**oled.print(hour); oled.print(":");**

**oled.print(minute); oled.print(":");**

**oled.print(second);**

**oled.line((LCDWIDTH/5)-5, (LCDHEIGHT/2)+10, LCDWIDTH-3, (LCDHEIGHT/2)+10);**

**oled.line((LCDWIDTH/5)-5, (LCDHEIGHT/2)-5, LCDWIDTH-3, (LCDHEIGHT/2)-5);**

**oled.line(LCDWIDTH-3, (LCDHEIGHT/2)+10, LCDWIDTH-3, (LCDHEIGHT/2)-5);**

**oled.line((LCDWIDTH/5)-5, (LCDHEIGHT/2)+10, (LCDWIDTH/5)-5, (LCDHEIGHT/2)-5);**

**oled.display();**

**second++;**

**}**

**else if(second == 59){**

**oled.clear(PAGE);**

**oled.setCursor(LCDWIDTH/5, LCDHEIGHT/2);**

**oled.print(hour); oled.print(":");**

**oled.print(minute); oled.print(":");**

**oled.print(second);**

**oled.line((LCDWIDTH/5)-5, (LCDHEIGHT/2)+10, LCDWIDTH-3, (LCDHEIGHT/2)+10);**

**oled.line((LCDWIDTH/5)-5, (LCDHEIGHT/2)-5, LCDWIDTH-3, (LCDHEIGHT/2)-5);**

**oled.line(LCDWIDTH-3, (LCDHEIGHT/2)+10, LCDWIDTH-3, (LCDHEIGHT/2)-5);**

**oled.line((LCDWIDTH/5)-5, (LCDHEIGHT/2)+10, (LCDWIDTH/5)-5, (LCDHEIGHT/2)-5);**

**oled.display();**

**second=00;**

**minute=00;**

**hour++;**

**}**

**}**

**}**

**else if (hour==11){**

**if (minute<60 && minute!=59){**

**if (second<60 && second!=59){**

**oled.clear(PAGE);**

**oled.setCursor(LCDWIDTH/5, LCDHEIGHT/2);**

**oled.print(hour); oled.print(":");**

**oled.print(minute); oled.print(":");**

**oled.print(second);**

**oled.line((LCDWIDTH/5)-5, (LCDHEIGHT/2)+10, LCDWIDTH-3, (LCDHEIGHT/2)+10);**

**oled.line((LCDWIDTH/5)-5, (LCDHEIGHT/2)-5, LCDWIDTH-3, (LCDHEIGHT/2)-5);**

**oled.line(LCDWIDTH-3, (LCDHEIGHT/2)+10, LCDWIDTH-3, (LCDHEIGHT/2)-5);**

**oled.line((LCDWIDTH/5)-5, (LCDHEIGHT/2)+10, (LCDWIDTH/5)-5, (LCDHEIGHT/2)-5);**

**oled.display();**

**second++;**

**}**

**else if(second == 59){**

**oled.clear(PAGE);**

**oled.setCursor(LCDWIDTH/5, LCDHEIGHT/2);**

**oled.print(hour); oled.print(":");**

**oled.print(minute); oled.print(":");**

**oled.print(second);**

**oled.line((LCDWIDTH/5)-5, (LCDHEIGHT/2)+10, LCDWIDTH-3, (LCDHEIGHT/2)+10);**

**oled.line((LCDWIDTH/5)-5, (LCDHEIGHT/2)-5, LCDWIDTH-3, (LCDHEIGHT/2)-5);**

**oled.line(LCDWIDTH-3, (LCDHEIGHT/2)+10, LCDWIDTH-3, (LCDHEIGHT/2)-5);**

**oled.line((LCDWIDTH/5)-5, (LCDHEIGHT/2)+10, (LCDWIDTH/5)-5, (LCDHEIGHT/2)-5);**

**oled.display();**

**second=00;**

**minute++;**

**}**

**}**

**else if(minute == 59){**

**if (second<60 && second!=59){**

**oled.clear(PAGE);**

**oled.setCursor(LCDWIDTH/5, LCDHEIGHT/2);**

**oled.print(hour); oled.print(":");**

**oled.print(minute); oled.print(":");**

**oled.print(second);**

**oled.line((LCDWIDTH/5)-5, (LCDHEIGHT/2)+10, LCDWIDTH-3, (LCDHEIGHT/2)+10);**

**oled.line((LCDWIDTH/5)-5, (LCDHEIGHT/2)-5, LCDWIDTH-3, (LCDHEIGHT/2)-5);**

**oled.line(LCDWIDTH-3, (LCDHEIGHT/2)+10, LCDWIDTH-3, (LCDHEIGHT/2)-5);**

**oled.line((LCDWIDTH/5)-5, (LCDHEIGHT/2)+10, (LCDWIDTH/5)-5, (LCDHEIGHT/2)-5);**

**oled.display();**

**second++;**

**}**

**else if(second == 59){**

**oled.clear(PAGE);**

**oled.setCursor(LCDWIDTH/5, LCDHEIGHT/2);**

**oled.print(hour); oled.print(":");**

**oled.print(minute); oled.print(":");**

**oled.print(second);**

**oled.line((LCDWIDTH/5)-5, (LCDHEIGHT/2)+10, LCDWIDTH-3, (LCDHEIGHT/2)+10);**

**oled.line((LCDWIDTH/5)-5, (LCDHEIGHT/2)-5, LCDWIDTH-3, (LCDHEIGHT/2)-5);**

**oled.line(LCDWIDTH-3, (LCDHEIGHT/2)+10, LCDWIDTH-3, (LCDHEIGHT/2)-5);**

**oled.line((LCDWIDTH/5)-5, (LCDHEIGHT/2)+10, (LCDWIDTH/5)-5, (LCDHEIGHT/2)-5);**

**oled.display();**

**second=00;**

**minute=00;**

**hour=00;**

**}**

**}**

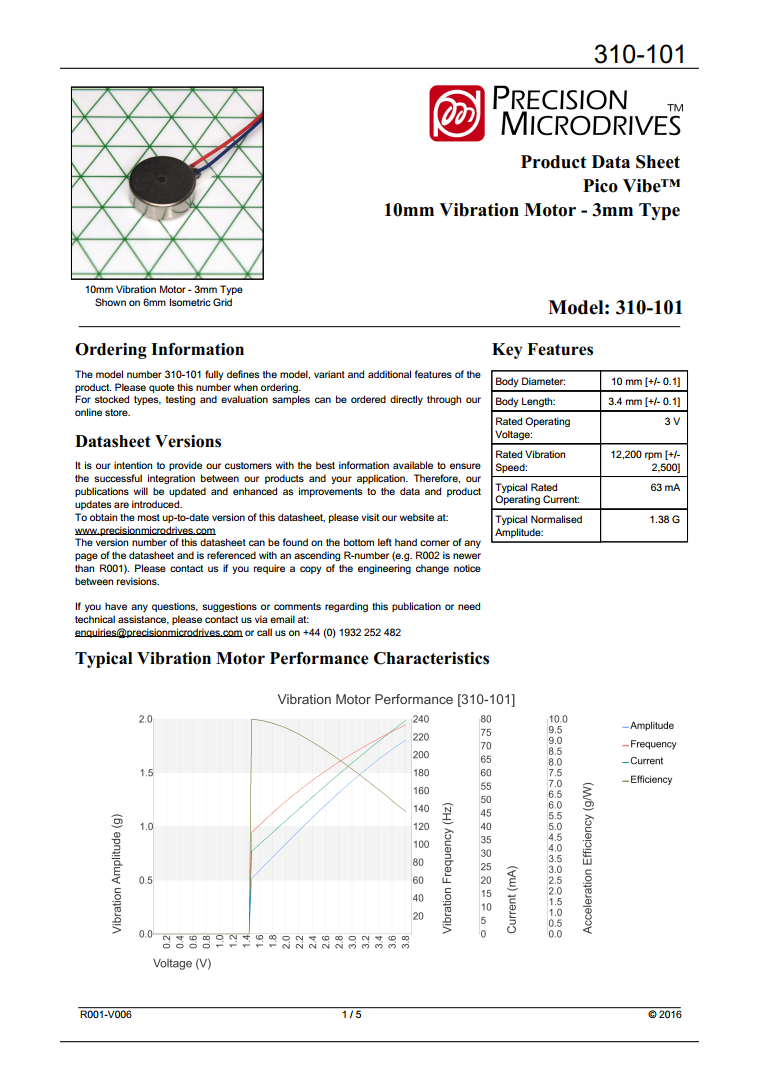
**}**

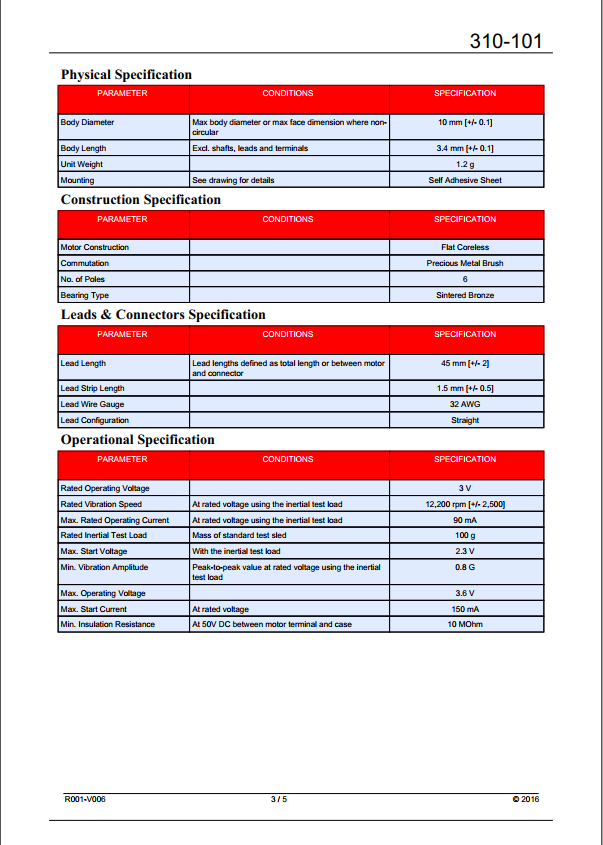
**}**

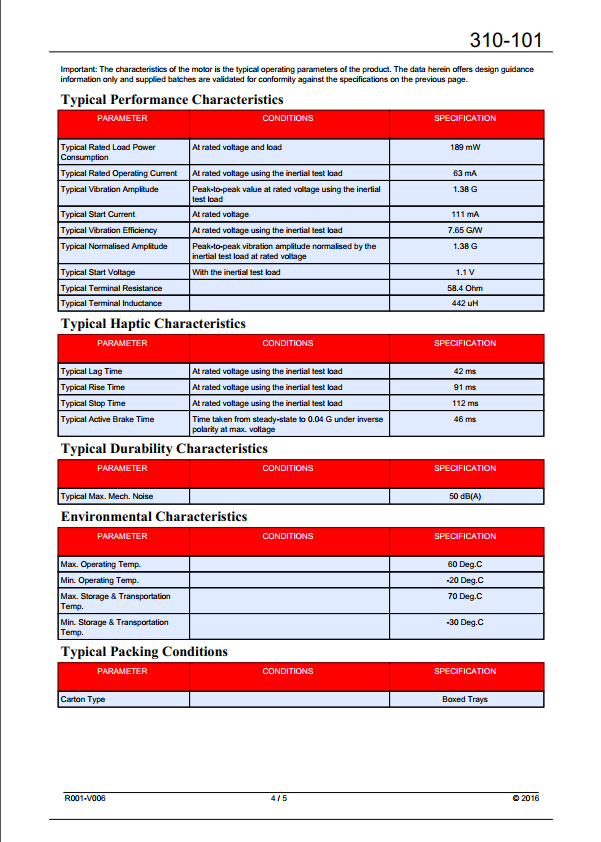
**}**

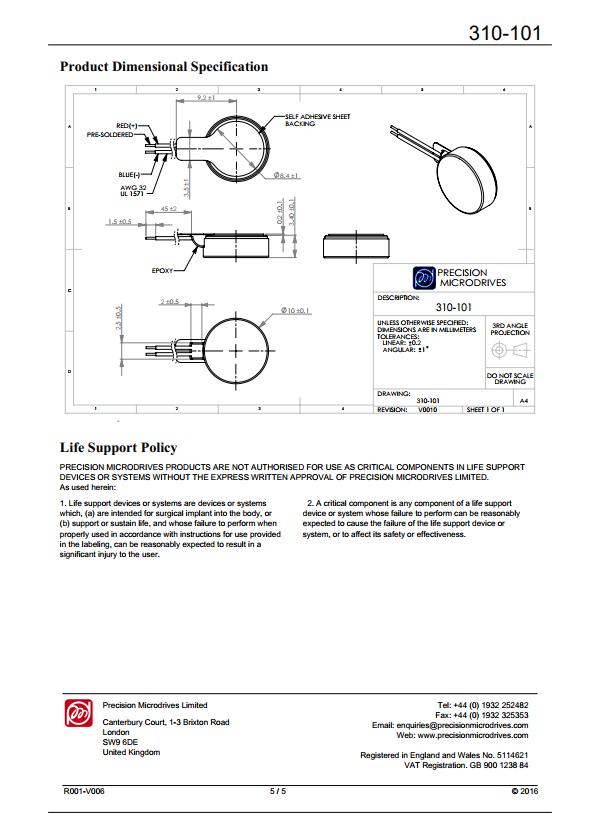
**Datasheet**

**Vibration micro motor**

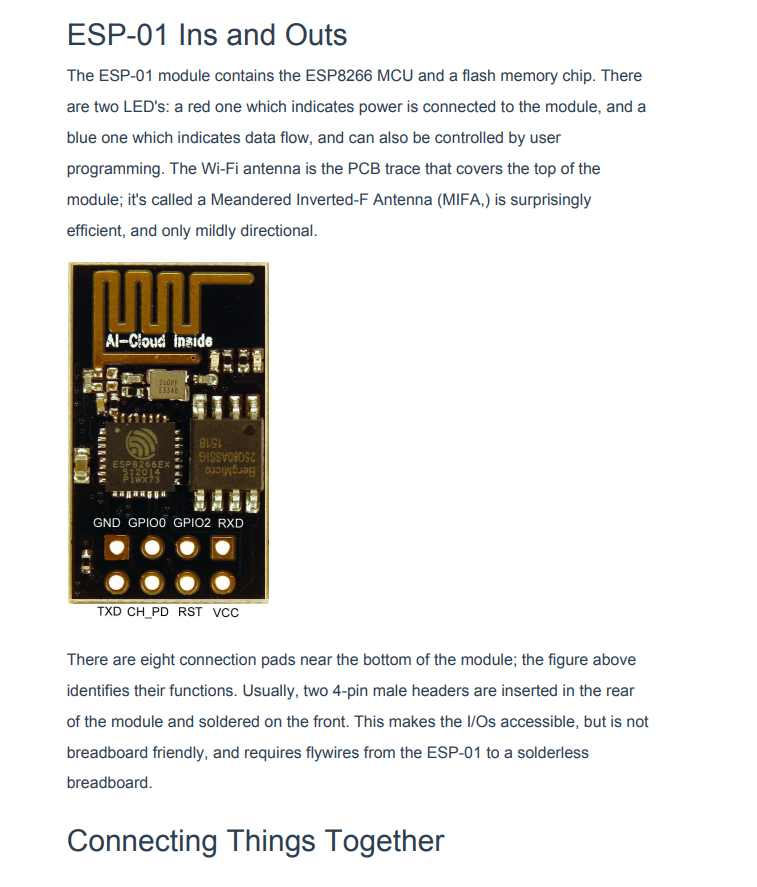


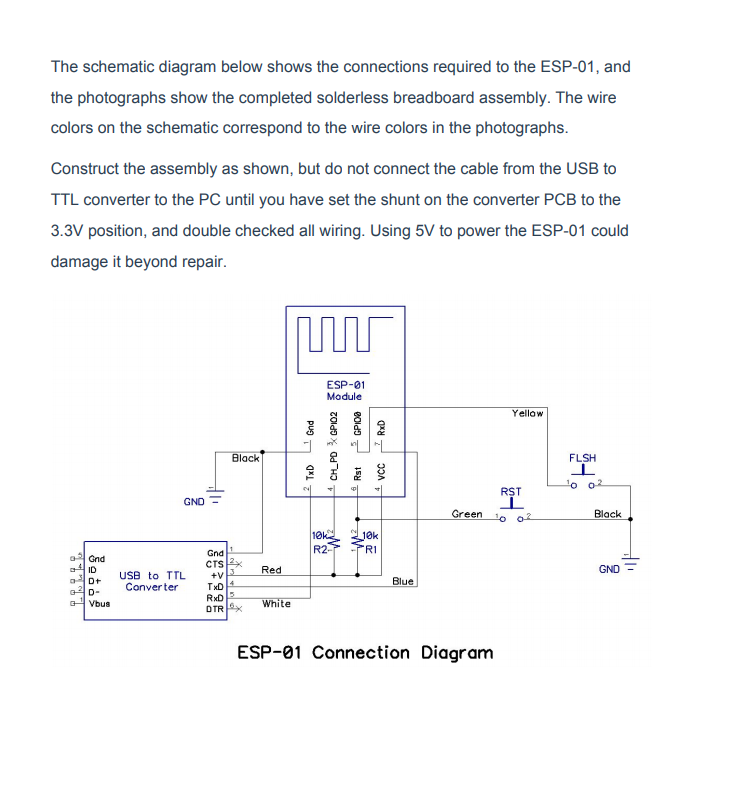


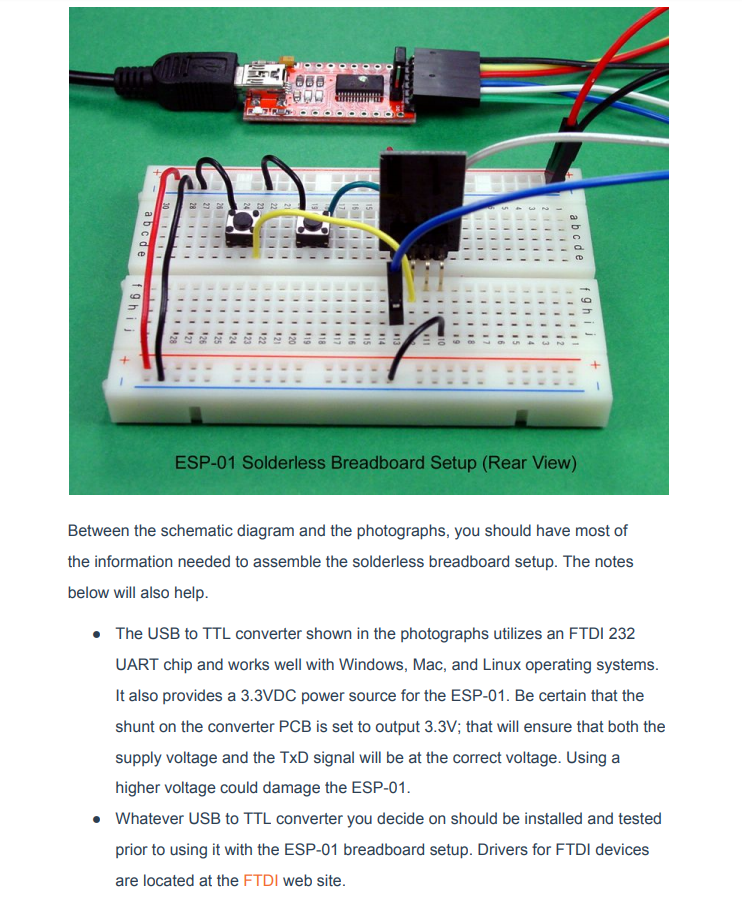


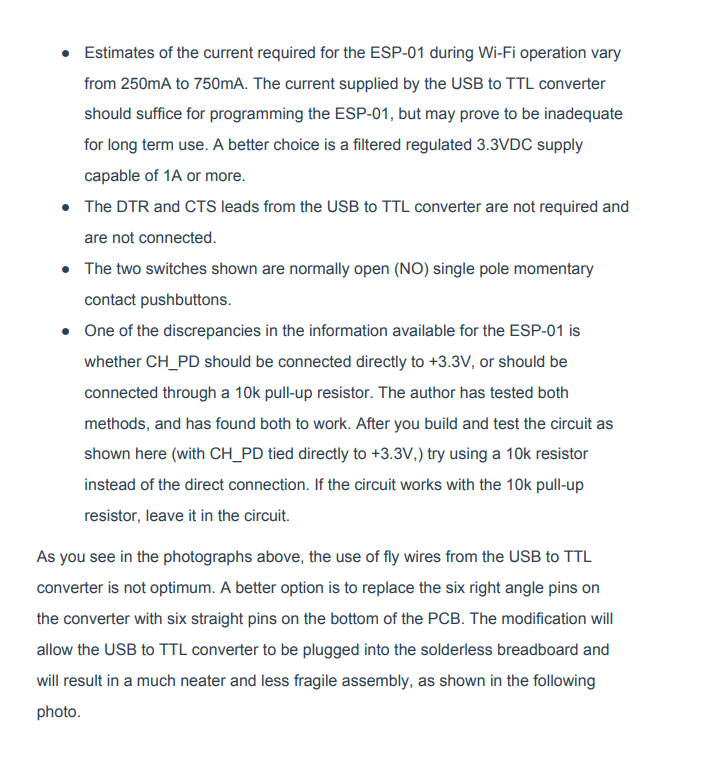


**ESP8266**

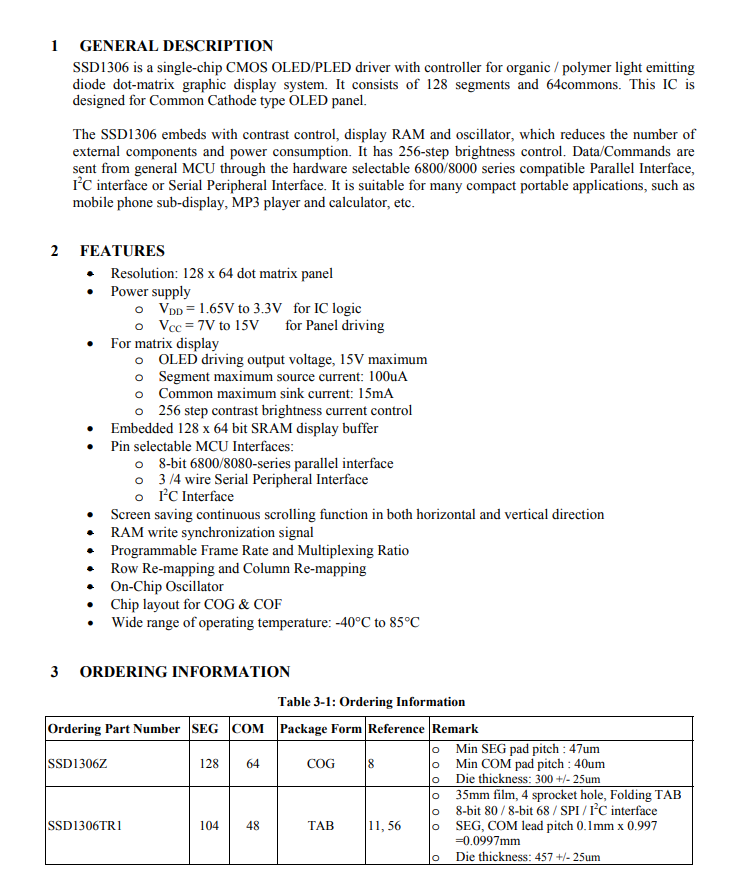


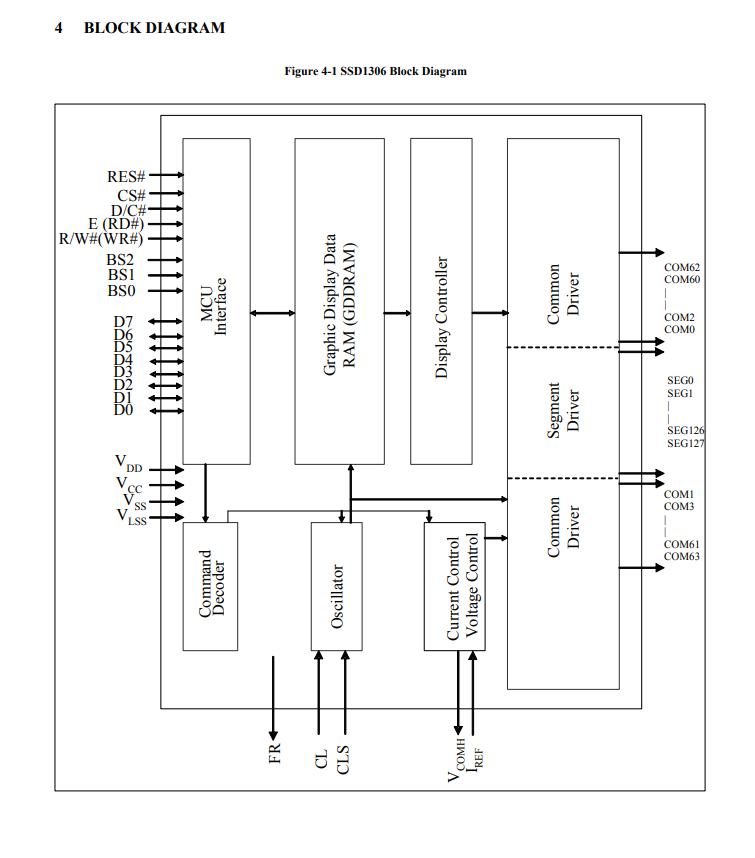




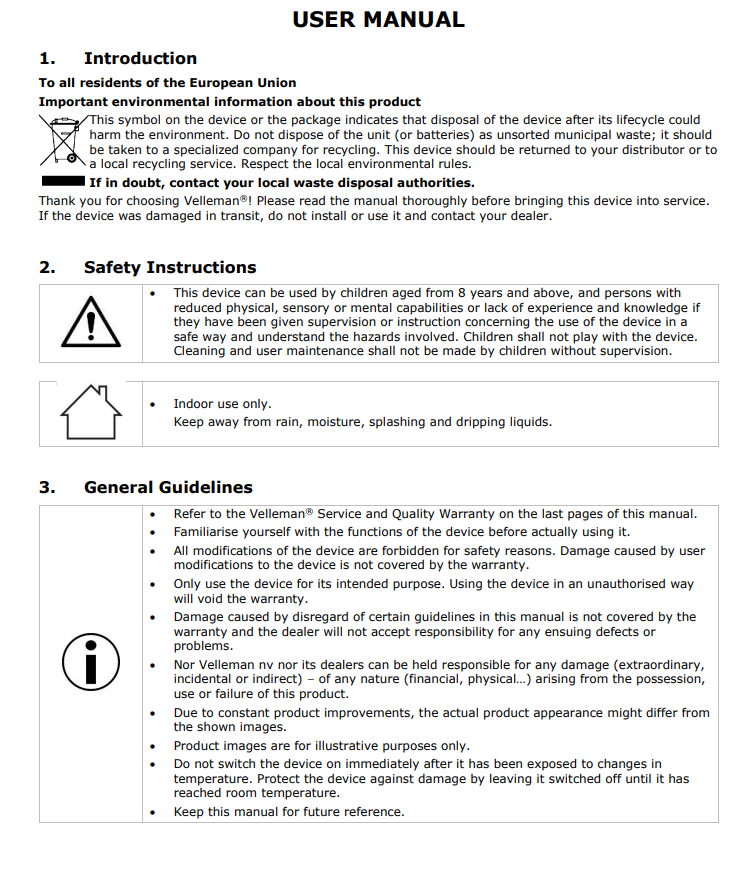


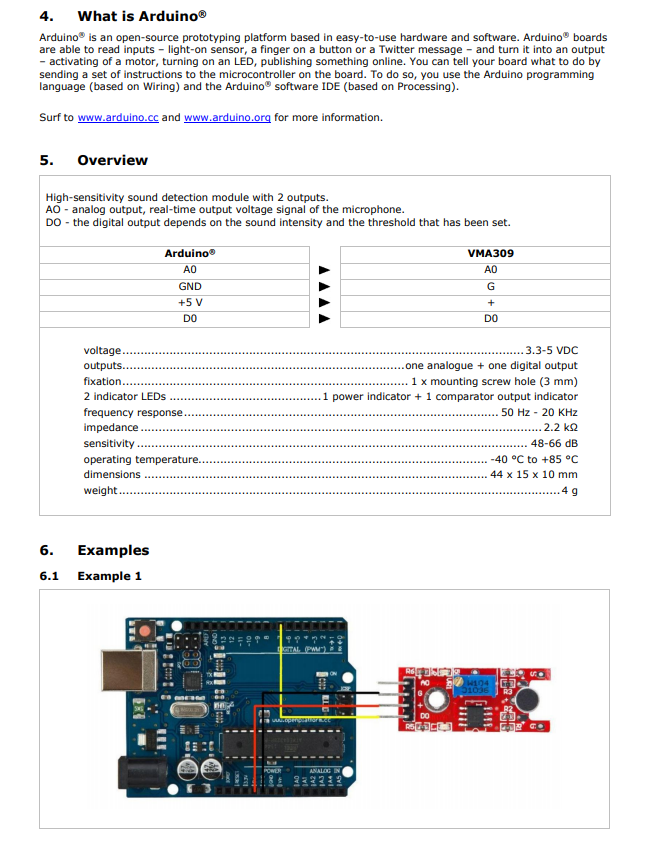
**Oled Breakout**





**Microphone Sensor AVR**





**ביבליוגרפיה**

Oled Breakout - <https://www.sparkfun.com/products/13003>

ESP8266 - https://www.electronicshub.org/esp8266-at-commands/