Candy Time – מערכת בקרה והתראה לאכילת מתוקים

גיל דוניץ 205689144 עומר פלדמן 308386846

שחף פור 205718620 נדב צברי 316462464

מס' קבוצה: 21

קבוצת תרגול: 14

**מטרת המערכת ותיאור משימת הבקרה**

מטרת המערכת:

צריכת מוצרים מתוקים יכולה להיות מסוכנת כאשר היא מתבצעת בשכיחות גבוהה ותורמת במיוחד להשמנת יתר כאשר מתבצעת בשעות מאוחרות של היום. מטרת מערכת Candy Time היא ליצור סביבת בקרה והתראה על צריכת מוצרים מתוקים על ידי דיירי הבית (שאוחסנו בכלי המיועד לכך) במסגרת זמן שהוגדרה מראש, זאת לטובת שמירה על בריאותם.

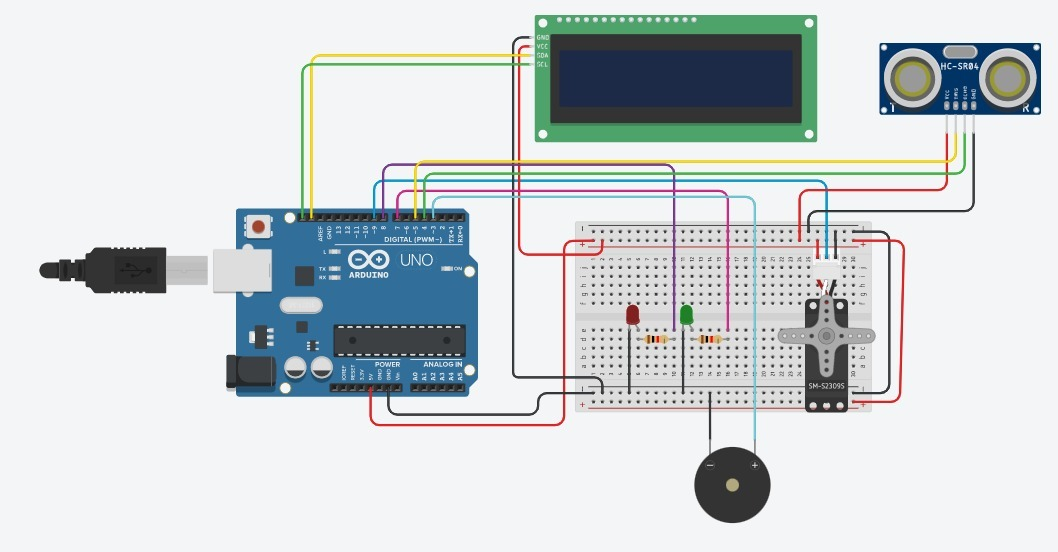
משימת הבקרה:

המערכת תאפשר גישה לצנצנת הממתקים בשעות מוגבלות במהלך היום ותבצע בקרה והתראה על כל ניסיון גישה לממתקים על ידי הרכיבים הבאים:

* חיישן אולטרה-סוני יזהה קירבה של היד לצנצנת הממתקים.
* צפצפה (Piezzo) המתריעה בעת זיהוי קירבה של החיישן. הצפצפה תופעל עד להתרחקות היד מהצנצנת.
* נורת לד אדומה אשר פועלת כאשר הגישה לצנצנת הממתקים חסומה.
* נורת לד ירוקה אשר פעולת כאשר הגישה מותרת.
* מסך LCD המציג את סטטוס הגישה לצנצנת הממתקים. כאשר הגישה לצנצנת חסומה, המסך ידפיס **No Candy Time**. כאשר הגישה לצנצנת מותרת, המסך ידפיס **Candy Time**.
* מנוע סרבו אשר יפעל בעת חישה של קירבה של יד לצנצנת. למנוע הסרבו תחובר יד פתוחה התדמה מכה על כף היד של מסיג הגבול בשעת הפעולה.

**תיאור התכן**

שרטוט המערכת:



טבלת רכיבים:

|  |  |  |  |
| --- | --- | --- | --- |
| מספר הפין | שם המשתנה | תצורה | הפעלה |
| 2 | Piezzo | OUTPUT | צפצפה להתראות המערכת:   1. אזעקה בעת קירבה של אובייקט בזמן אסור. 2. צפצוף עבור תחילה/ סוף זמן ממתקים. |
| 4 | DistanceSensor | OUTPUT | חיישן אולטרה-סוני לבדיקת המרחק של אובייקט מהצנצנת. |
| 5 | DistanceSensor | OUTPUT |
| 7 | greenLed | OUTPUT | נורת לד ירוקה – דולקת במהלך זמן הממתקים. |
| 8 | redLed | OUTPUT | נורת לד אדומה – דולקת במהלך זמן האסור לממתקים. |
| 6 | servoPin | OUTPUT | מופעל כל עוד חיישן המרחב מזהה קירבה של אובייקט בזמן האסור לממתקים. |

**תיאור הבדיקות שבוצעו**

בחרנו לבצע מבחן סטטיסטי שבודק האם אכן חיישן הsupersonic אמין ומציג ערך מספיק קרוב למרחק במציאות. בחרנו להשתמש במבחן סטטיסטי t מזווג מאחר וקבוצת הבדיקה שלנו זהה (אותו החיישן) השערת 0H -המרחק שהחיישן מציג אינו תקין, והשערת 1 – Hהמרחק שהחיישן מציג תקין (תואם למרחק במציאות). ביצענו 15 מדידות כאשר ערך השגיאה הינו הפרש התוחלות שעבורו אנחנו מחשבים את הסטטיסטי לפי המבחן לפי רמת מובהקות 5%.

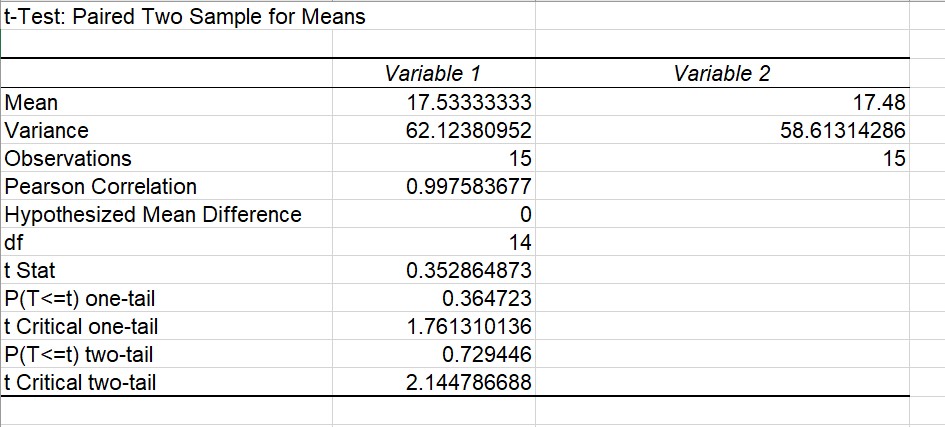
**תיאור תוצאות הבדיקות**

טבלת הנתונים שעליה ביצענו את המבחן הסטטיסטי:

טבלה 1 מדידות שגיאת חיישן מרחק supersonic

|  |  |  |  |
| --- | --- | --- | --- |
| מספר בדיקה | מרחק שהחיישן מציג | מרחק במציאות | שגיאה אבסולטית |
| 1 | 15 ס"מ | 14.7 ס"מ | 0.3 ס"מ |
| 2 | 12 ס"מ | 12 ס"מ | 0 |
| 3 | 14 ס"מ | 14 ס"מ | 0 |
| 4 | 16 ס"מ | 16 ס"מ | 0 |
| 5 | 11 ס"מ | 10 ס"מ | 1 ס"מ |
| 6 | 9 ס"מ | 10 ס"מ | 1 ס"מ |
| 7 | 30 ס"מ | 29 ס"מ | 1 ס"מ |
| 8 | 23 ס"מ | ס"מ 23 | 0 |
| 9 | 25 ס"מ | 25 ס"מ | 0 |
| 10 | 19 ס"מ | 20 ס"מ | 1 ס"מ |
| 11 | 21 ס"מ | 21 ס"מ | 0 |
| 12 | 27 ס"מ | 26.5 ס"מ | 0.5 ס"מ |
| 13 | 5 ס"מ | 5 ס"מ | 0 |
| 14 | 8 ס"מ | 8.5 ס"מ | 0.5 ס"מ |
| 15 | 28 ס"מ | 27.5 ס"מ | 0.5 ס"מ |

פלט ערכי התוצאות שקבלנו עבור המבחן הסטטיסטי שביצענו בEXCEL



ניתן לראות בנתוני הפלט שערך הסטטיסטי יצא קטן יותר מהערך הקריטי. נרצה לדחות את השערת ה-0 עבור ר"מ 5% משמע, המרחק שהחיישן מציג נמצא תקין.

**מסקנות מהעבודה**

המסקנה המרכזית מהעבודה שלמדנו היא שלא מספיק לרשום קוד איכותי עבור הבקר והרכיבים, חשוב לבצע בדיקות וצריבה לאורך הוספת הרכיבים. כמו כן חשוב לדעת למקם בצורה נכונה על הלוח החיבורים אחרת לא כל הרכיבים יקבלו מתח ויפעלו כראוי.

ממליצים להמשך לבצע בדיקות על כל רכיב בנפרד בבקר הפיזי במהלך העבודה יכול ליעל את התהליך ולחסוך זמן רב בחיפוש התקלות.

**נספחים**

**קובץ readme**

* יש להגדיר את הזמן בו יש גישה לממתקים (CandyTimeDuration) ואת הזמן בו אין גישה לממתקים (NoCandyTimeDuration).
* יש לחבר את כל הרכיבים ללוח הלבן במקום המתאים: את הlcd, נורות הled, הservo, הpiezzo והאולטרה-סוניק.
* בזמן בו הגישה לממתקים מותרת, על גבי צג הlcd תוצג ההודעה ״Candy time!!!״ ונורת הled הירוקה תדלק. בזמן הזה, חיישן האולטרה-סוניק יהיה כבוי וכן הservo והpiezzo.
* בזמן בו הגישה לממתקים אסורה, על צג הlcd תופיע ההודעה ״No candy time!!!״ ונורת הled האדומה תדלק. בנוסף, במידה וחישן האולטרה-סוניק מזהה שמישהו מנסה להגיע לממתקים, המערכת תגיב בכך ש:

1. על צג הLCD תופיע ההערה ״Dont touch!!!".
2. הpiezzo יופעל וישמע ממנו צפצוף.
3. הservo יסתובב (וידמה כף יד שמרחיקה את הניסיון להגיע לממתקים.

נספח תיעוד הקוד

#include <LiquidCrystal\_I2C.h>

#include <Servo.h>

Servo servo;

#include <Wire.h>

// initialize the library with the numbers of the interface pins

LiquidCrystal\_I2C lcd\_1(0x27, 16, 2);

#define pingPin 4

#define trigPin 5

#define redLed 8

#define greenLed 7

#define PiezoPin 2

//Input

int no\_candy\_time\_duration = 10; // define no candy time duration

int candy\_time\_duration = 4; // define candy time duration

//Help variables

unsigned long previousMillis = 0;

bool isCandyTime = false;

long duration, cm, distance;

int interval;

int interval2;

void setup() {

interval = 1000\*no\_candy\_time\_duration;

interval2 = 1000\*candy\_time\_duration;

Serial.begin(9600);

servo.attach(6);

lcd\_1.init();

lcd\_1.backlight();

pinMode(pingPin, INPUT);

pinMode(trigPin, OUTPUT);

pinMode(redLed, OUTPUT);

pinMode(greenLed, OUTPUT);

pinMode(PiezoPin, OUTPUT);

digitalWrite(PiezoPin, HIGH);

}

void loop() {

unsigned long currentMillis = millis();

Serial.println(isCandyTime);

if((unsigned long)(currentMillis - previousMillis) < interval && isCandyTime == false){

digitalWrite(redLed, HIGH);

distance = checkDistance();

if(distance <= 30){

printScreen("Dont Touch!! ");

digitalWrite(PiezoPin, LOW);

startServo();

}

else{

servo.write(90);

digitalWrite(PiezoPin, HIGH);

printScreen("No Candy Time!");

}

}

else if ((unsigned long)(currentMillis - previousMillis) >= interval && isCandyTime == false) {

candyTime();

}

else if((unsigned long)(currentMillis - previousMillis) >= (interval+interval2) && isCandyTime == true) {

digitalWrite(greenLed, LOW);

previousMillis = currentMillis;

isCandyTime = false;

printScreen("No Candy Time!");

}

}

void candyTime(){

servo.write(90);

digitalWrite(PiezoPin, HIGH);

digitalWrite(redLed, LOW);

digitalWrite(greenLed, HIGH);

isCandyTime = true;

printScreen("Candy Time! ");

}

void printScreen(String print){

lcd\_1.setCursor(0, 0);

lcd\_1.print(print);

}

double checkDistance(){

// The PING))) is triggered by a HIGH pulse of 2 or more microseconds.

// Give a short LOW pulse beforehand to ensure a clean HIGH pulse:

digitalWrite(trigPin, LOW);

delayMicroseconds(2);

// Sets the trigPin HIGH (ACTIVE) for 10 microseconds

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

// Reads the pingPin, returns the sound wave travel time in microseconds

duration = pulseIn(pingPin, HIGH);

// convert the time into a distance

// Calculating the distance

distance = duration \* 0.034 / 2;

// Speed of sound wave divided by 2 (go and back)

return distance;

}

void startServo(){

Serial.println((String)distance);

servo.write(180);

}