



ПРОЕКТ „ПАРКТРОНИК“

Финален проект за модул 8: ВСОС към НП „Обучение за ИТ Кариера“

Александър Димитров Димитров

Никола Александров Хаджиев

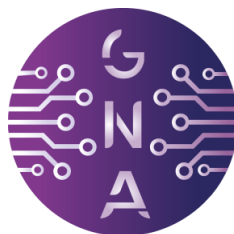
Георги Антонов Трапов

</BETTER DO IT ALL AT ONCE>



СЪДЪРЖАНИЕ

Увод.....	3
1. Цели.....	3
2. Начини на използване	3
3. Съществуващи решения	3
• Парктроник за автомобили	3
4. Нашето решение.....	4
• Главни Компоненти.....	4
• Полза от компонентите.....	4
Описание на цялостната работа на проекта.....	5
1. Кратко описание на проекта.....	5
2. Принцип на работа	5
3. Схема и описание на свързване.....	8
Инструкции за работа с проекта.....	14
Заклучение	15
Източници.....	16



Увод

1. Цели

- Целта на този проект е създаване на устройство, с помощта на ардуино микроконтролер, чрез което да може да се измерва разстояние до даден обект и да се сигнализира потребителя чрез звуков и светлинен сигнал на какво разстояние е той.

2. Начини на използване

- Нашият проект може да се използва посредством връзка с блутут към компютър или телефон. За целта трябва да използвате нашите приложения, създадени за връзка с тези устройства

3. Съществуващи решения

- Парктроник за автомобили



Фиг. 1



Фиг. 2



4. Нашето решение

● Главни Компоненти

В нашия проект сме използвали: Ардуино Уно микроконтролер, бредборд, блутут модул HC-05, светодиод, бърз за звукова сигнализация и ултразвуков датчик за разстояние

● Полза от компонентите

● Ардуино Уно

За микроконтролер в нашата вградена система ползваме Ардуино Уно, благодарение на който нашата система функционира пълноценно.

● Блутут Модул

Използваме го, за да изпращаме информация за разстоянието до всяко блутут поддържащо устройство с предназначена програма.

● Ултразвуков сензор за разстояние

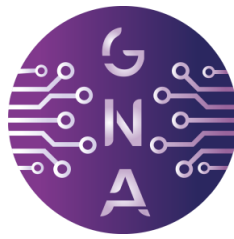
Неговото предназначение е да даде информация за разстоянието до определен предмет, посредством ултразвукови вълни.

● Светодиод

Предназначението му е да сигнализира чрез светлинна сигнализация в 3 степени

● Бърз

Вторичен индикатор за разстояние чрез звуков сигнал в три степени на честота на индикацията



Описание на цялостната работа на проекта

1. Кратко описание на проекта

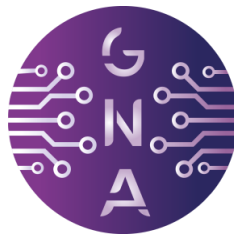
Идеята на проекта е проста, полезна и приложима в реалния свят, а именно да наподобява парктроник на автомобил. В предложението от нас вариант не е задължително да се използва на автомобил, а може да се използва навсякъде, където имате нужда, стига да имате компютър или батерия, за да дадете заряд на ардуиното.

2. Принцип на работа

За да можем да осъществим тази идея са ни необходими няколко главни компонента.

Таблица 1.

Име на компонента	Количество
Arduino Uno R3	1
Ultrasonic Distance Sensor	1
Buzzer MB12A12	1
LED лампичка с два цвята – зелено; червено	1
HC-05 блутут модул	1
HC-SR04 ултразвуков датчик за разстояние	1
Jumper(джъмпер) кабели	20
Резистор 150 Ом	1
Резистор 1,2 kОм	1



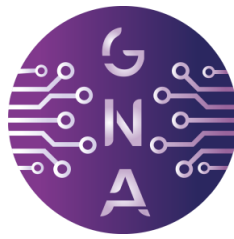
Първият е най-важният – сензор за разстояние. Този, който сме избрали за проекта е стандартен и често използван . Работи на лесен за разбиране принцип, като изпраща звукови вълни и чака отговор или така нареченото „Ехо“. Изминалото време от изпращането на сигнал до получаване на ехо се връща под формата на милисекунди.

Пример: ако има предмет на 10см от сензора и скоростта на звука е 340 m/s или преобразувано 0.034 cm/ μ s, звуковата вълна ще се върне за 294 μ s. Резултатът който получаваме е двоен, защото това време включва преминаването на вълната и в 2-те посоки. За да премахнем излишното време разделяме на две. В крайна сметка формулата излиза така: Дистанцията = Времето * 0,034 / 2.

Светодиодът за светлинна сигнализация свети в 2 цвята – червено и зелено. Ние използваме средно положение, в което се смесват двата цвята, за да получим жълто при нужда. Когато предметът е на критично близко разстояние, светодиода свети в червено и примигва с висока честота. При средно разстояние свети в жълто и мига със средна честота, а когато дистанцията е нормална свети в зелено и не примигва.

Бъзерът е нашата вторична сигнализация посредством звуци с различни честоти. Той се включва в зависимост от данните получени от сензора за разстояние. Когато обектът е критично близко звуковата сигнализация е най-интензивна. При средно разстояние издава звукова сигнализация с по-малък интензитет, а при нормално не прави нищо.

Блутут модула е една от най-интересните части в проекта. Той позволява на вградената система да се свързва с други устройства безжично, посредством блутут. Чрез този модул изпращаме данните получени от сензора и прочетени от ардуиното към свързаните устройства и те биват визуализирани благодарение на написаните от нас програми, които ни помагат да четем нужната информация.

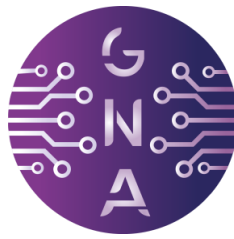


Основният принцип на работа се базира на прочитането на разстоянието до даден обект, независимо неговата големина и вид. Чрез ултразвуковия сензор може да се изчисли определено разстояние от обекта към самата вградена система. Формулата, която използваме е описана по рано. Блутут модула ни служи за сложна, но интересна комуникация между платката и устройство поддържащо блутут. Чрез този модул и програмите, написани от нашия екип, платката успява да предаде нужната информация на телефона или компютъра под следния формат:

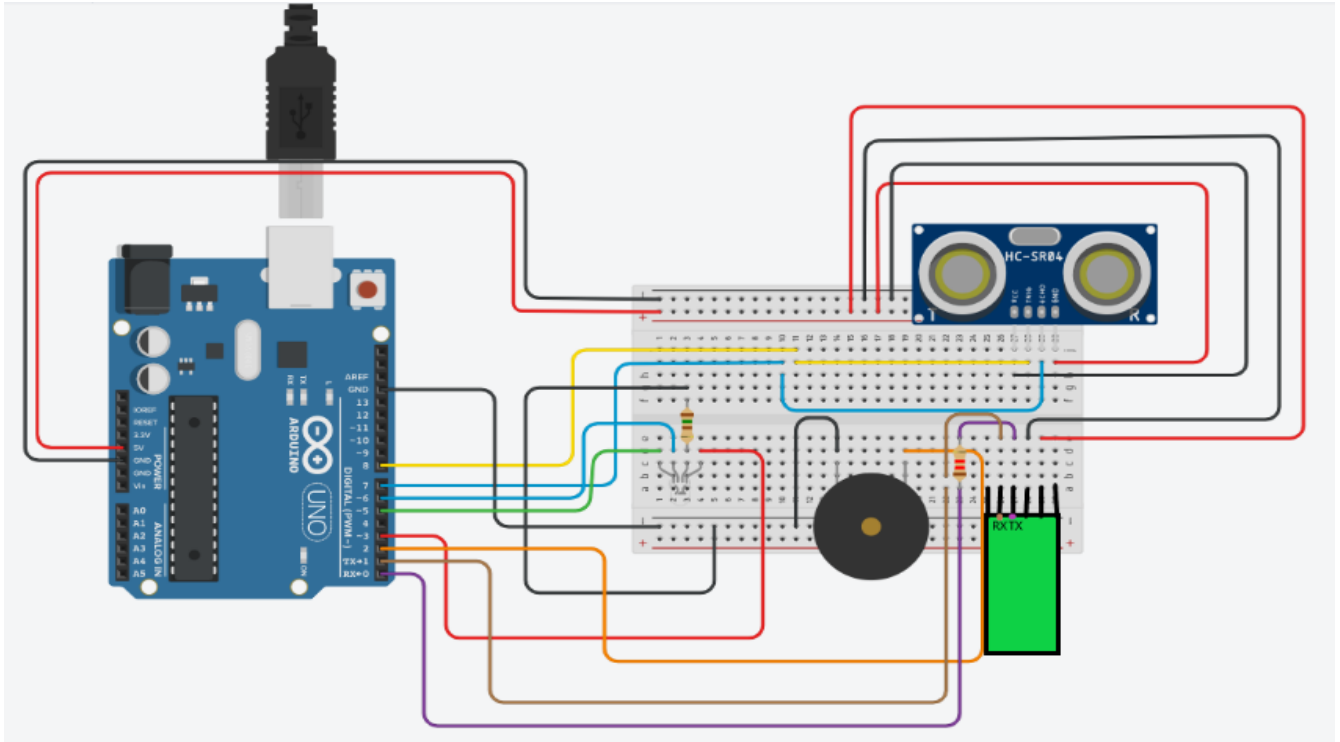
„Distance: {distance} cm”

Така ние смятаме, че е най-лесно за разбиране от потребителя, който използва нашия Парктроник.

Фиг. 3



3. Схема и описание на свързване.

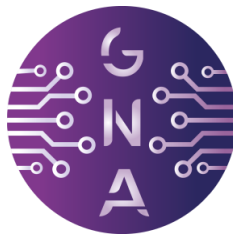


Фиг. 4

Представената схема е виртуална реализация на нашия проект на Tinkercad, за по улеснено обяснение на връзките във вградената система. Първоначално проектът беше направен там, но решихме да го пресъздадем и в реалния живот.

Setup:

В началото на програмата сме инициализирали пиновете 7, 8, 5, 3, 2. За да не заема никаква памет от оперативната памет на Ардуиното, сме използвали `#define` за тяхната инициализация, тъй като паметта с която разполагаме е сравнително малка. След това сме създали две променливи – **duration** (за изчисляване на разстоянието от обекта до платка чрез „Ехо вълни“) и **state** (за обмен на данни между платката и компютър/ телефон чрез блугут модула). `Setup`-а е метод, който се извиква един път. В него използваме `pinMode`, като за Лед лампичката казваме, че тя е `OUTPUT`, тоест че през нея като премине ток, тя ще е изходяща. В нашия случай това е пин 5 и 3, като им казваме, че те са изходящи пинове. Понеже те са означени с тилда, това значи ,че на тези пинове можем да се възползваме от ШИМ(PWM). Пин номер 7 и 8 се използват от нашия ултразвуков сензор, като `trigPin(7)` означаваме като `OUTPUT`, защото се връзва към `TRIG` частта на сензора, през която излиза самия звук, който в послевъздействие се



отбъсква от обекта и се връща към ECHO частта на сензора. Именно за това, свързаният към тази част пин номер 8 е дефиниран като INPUT, защото получава върналите се вълни. Serial.begin(9600) се използва за да може върнатата информация да се запише по подразбиране на Екрана, вграден в Arduino IDE-то, но в нашия случай- на телефон или компютър.

```
#define trigPin 7
#define echoPin 8

#define greenLedPin 5
#define redLedPin 3

#define buzzPin 2

// defines variables

long duration;
int state = 0;

void setup() {
  pinMode(greenLedPin, OUTPUT);
  pinMode(redLedPin, OUTPUT);

  pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
  pinMode(echoPin, INPUT); // Sets the echoPin as an Input

  Serial.begin(9600); // Starts the serial communication 9600
}
```

Loop:

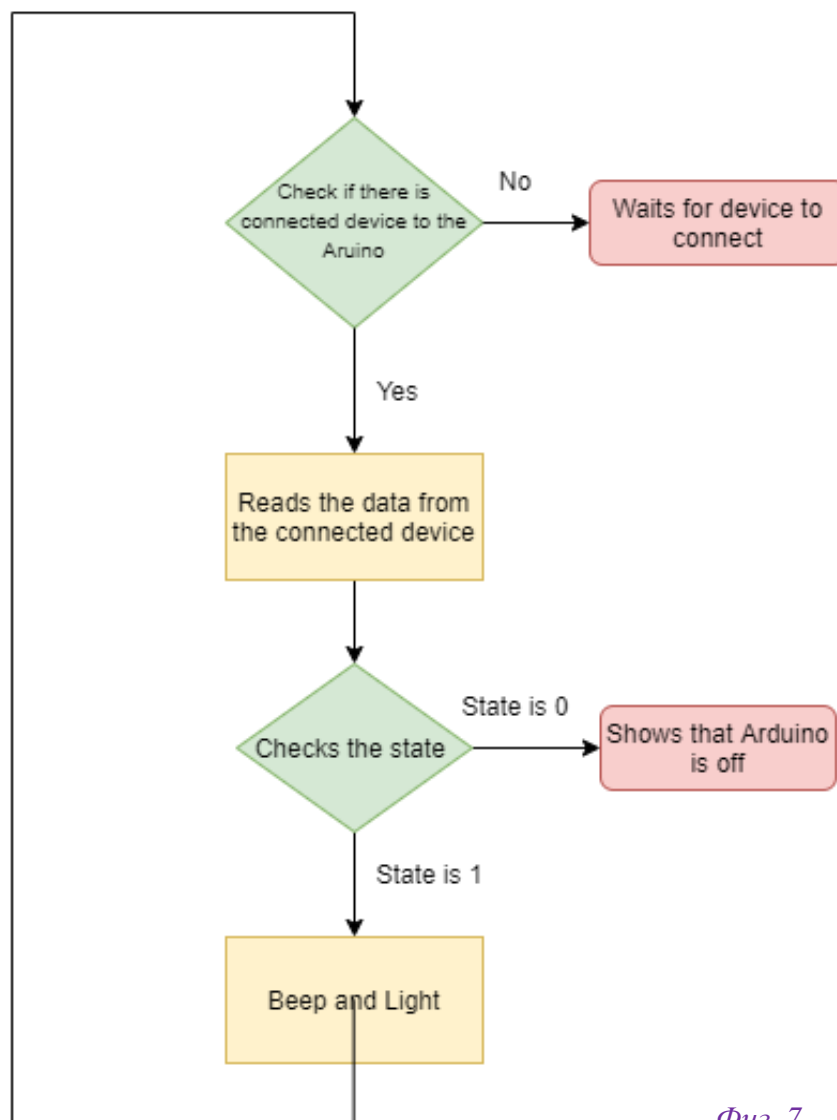
Фиг. 5 и 6

```
void loop() {
  if(Serial.available() > 0) // Checks whether data is coming from the serial port
  {
    state = Serial.read(); // Reads the data from the serial port
  }

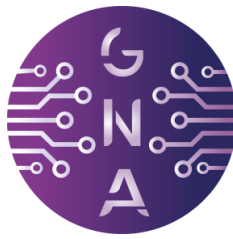
  if (state == '1'){
    BeepAndLight();
  }
}
```



Loop метода е в основата на вградените системи. Чрез него ние даваме „живот“ на нашата платка. В нашия случай в loop-а имаме 2 if-a, като първия проверява дали имаме свързано към нашия блутут модул устройство. Ако имаме, то чрез state променливата четем от входа на някое навързано устройство input-a на потребителя, тоест какво ще натисне потребителя в нашите създадени програми. В случай че нямаме навързано устройство програмата продължава да работи, но няма да изписва никаква информация за разстоянието до обекта, тъй като няма къде да се изпише. Когато работи, сложената от нас лампичка в зависимост от разстоянието, започва да свети във вариращи цветове. В зависимост от разстоянието на обекта, бъзерът издава звуков сигнал с различни честоти. Ако в state променливата е записано числото 1, тоест потребителят е натиснал бутона „Connect“ в десктоп програмата или „Turn Parktronic ON“ на андроид програмата, платката ще започне да изпраща информация за желаното разстояние чрез навързания блутут. В случая се използва void метода BeepAndLight();, който е описан по-долу в документа.



Фиг. 7



Метода BeepAndLight:

```
void BeepAndLight() {  
    state = 1;  
    digitalWrite(redLedPin, LOW);  
    digitalWrite(greenLedPin, LOW);  
  
    printDistance();  
  
    if (calculateDistance() > 30) {  
        SetLedGreenAndBeep();  
    }  
    if (calculateDistance() >= 15 && calculateDistance() <= 30) {  
        SetLedYellowAndBeep();  
    }  
    if (calculateDistance() < 15) {  
        SetLedRedAndBeep();  
    }  
  
    delay(200);  
}
```

Фиг. 8

Чрез този метод се изписва информация на потребителя използващ нашите програми. Към променливата state присвояваме единица, за да може потребителят да продължи да получава информация за разстоянието на обекта, докато не натисне друг бутон от програмата, който ще промени state променливата в зависимост от предназначението му. След това използваме digitalWrite метода на двата пина за ЛЕД Светодиода, за да създадем ефекта за мигане. Всеки път като е светнат и програмата се върне в началото на цикъла, като премине през този метод ,диодът загасва и по късно пак се пуска. PrintDistance() метода също е void и се използва, за да изпише на потребителя съответните сантиметри между обекта и платката. След това имаме три If кейса, като в тях използваме методите calculateDistance(), за да изчислим разстоянието чрез използвана от нас формула. Методите SetLedGreenAndBeep(), SetLedYellowAndBeep(), SetLedRedAndBeep() се използват в зависимост от разстоянието, за да индикират положението на обекта, тоест в нашия случай, ако обектът се разполага на повече от 30 см от платката Диода свети в зелено и бъзера не издава звук- ако обекта се разполага между 15 и 30 см Диодът свети в жълто(смесица между червено и зелено) и бъзерът започва интензивно да издава звук- ако обектът се разполага на по-малко от 15 см от платката Диода свети в червено и бъзерът издава звук много по начесто. Накрая слагаме delay() от 200 милисекунди, за да може да бъде видим процесът на работа за човешкото око. Изброените методи са описани подробно по надолу в документа.



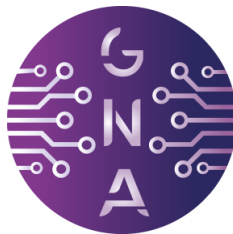
Методу printDistance() u calculateDistance():

```
void printDistance() {  
    Serial.print("Distance: ");  
    Serial.print(calculateDistance());  
    Serial.println(" cm");  
}  
  
int calculateDistance() {  
    // Clears the trigPin  
    digitalWrite(trigPin, LOW);  
    delayMicroseconds(2);  
    // Sets the trigPin on HIGH state for 10 micro seconds  
    digitalWrite(trigPin, HIGH);  
    delayMicroseconds(10);  
    digitalWrite(trigPin, LOW);  
    // Reads the echoPin, returns the sound wave travel time in microseconds  
    duration = pulseIn(echoPin, HIGH);  
  
    // Prints the distance on the Serial Monitor  
    //Serial.print("Distance: ");  
    //Serial.println(distance);  
    return duration * 0.034 / 2;  
}
```

Фиг. 9

Методът printDistance() се използва за показване на информация на потребителя. В случая използваме Serial.print и Serial.println, като чрез тях информацията се записва на екрана на телефон или компютър.

Методът calculateDistance() е int, тъй като връща стойност. В началото на метода по същия начин както и за светодиода сме използвали digitalWrite() като сме оставили triggerPin на LOW, за да го изчиства и да може интензивно да пуска "Ехо" вълна при всяко завъртане в цикъла. След това го пускаме чрез HIGH, като от дясната част на сензора се пускат вълни в продължение на 10 милисекунди. Но тези вълни все някога трябва да спрат – именно затова след 10 милилисекунди слагаме пина на LOW. Използваме променливата duration с метода pulseIn, като чрез този метод и думата HIGH казваме на echoPin да се включи и когато го направи, лявата част на сензора да приеме отблъзнатата от обекта вълна, която е пусната. Накрая тази стойност се връща като се умножава по 0.034, тъй като това е времето за което импулсът изминава 1 сантиметър. Делим всичко на 2, защото искаме да вземем само тази част, която се е върнала, а не целия импулс.



Допълнителни методи:

```
void SetLedGreenAndBeep() {  
    buzOnGreen();  
}  
  
void SetLedRedAndBeep() {  
    buzOnRed();  
}  
  
void SetLedYellowAndBeep() {  
    buzOnYellow();  
}  
  
void buzOnGreen() {  
    //tone(buzzPin, 255);  
    //delay(100);  
    noTone(buzzPin);  
    analogWrite(greenLedPin, 255);  
    delay(900);  
}  
  
void buzOnYellow() {  
    tone(buzzPin, 255);  
    delay(100);  
    noTone(buzzPin);  
    analogWrite(redLedPin, 150);  
    analogWrite(greenLedPin, 255);  
    delay(600);  
}  
  
void buzOnRed() {  
    tone(buzzPin, 255);  
    delay(100);  
    noTone(buzzPin);  
    analogWrite(redLedPin, 255);  
    delay(300);  
}
```

Фиг. 10

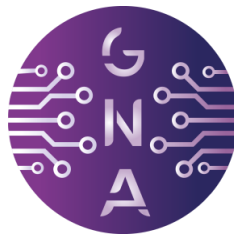
Трите метода `SetLedGreenAndBeep()`, `SetLedRedAndBeep()`, `SetYellowAndBeep()` се използват за по-голяма подреденост на кода, като в себе си извикват други три метода, които са съществено по-важни.

Методите `buzOnGreen()`, `buzOnYellow()`, `buzOnRed()`:

-`buzOnGreen()` – Използваме този метод, за да накараме Лед Диода да светне в зелено с максимална честота на импулса, използван в ШИМ, тъй като използваме 5 пин, който е означен с тилда и може да приеме стойности от 0 до 255. В този метод бързърът не издава звук, точно заради това използваме `noTone`.

-`buzOnYellow()` – Използваме този метод за да накараме Лед Диодът да светне в жълто, като по същата причина като при `buzOnGreen()` използваме ШИМ за пиновете и смесваме цветовете за да получим жълт цвят. Тук използваме `tone()` метода, за да накараме бързърът да започне да издава звук, изчакаме 100 милисекунди, за да се издаде звук и след това го спираме с `noTone()`. Накрая изчакаме 600 милисекунди и ако обектът е все още на това разстояние, методът продължава да се изпълнява. Това важи и за трите метода.

-`buzOnRed()` – Идентично като миналите два метода, този метод работи по същия начин, но тук светва само червената част на Светодиода и изчакаването накрая е по малко, тъй като колкото по-малко е изчакаването, толкова по бързо свети лампата и толкова по бързо бързърът издава звук, което придава ефект на аларма.



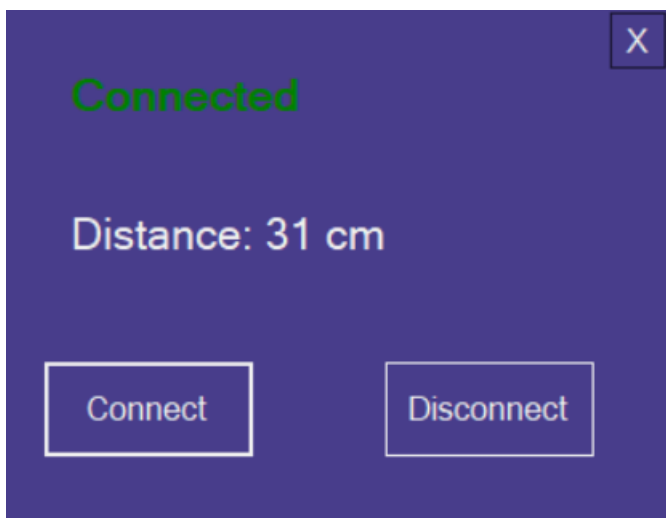
Инструкции за работа с проекта

1. Ардуиното се включва към компютър посредством USB кабел или се включва към батерия
2. За да може да се свържете с блутут модула трябва да инсталирате някое от изготвените от нас приложения за съответната платформа.
3. За да може да изкарвате нужната информация трябва да натиснете бутона Connect
4. След като изпише Connected, вече сте готови за работа!

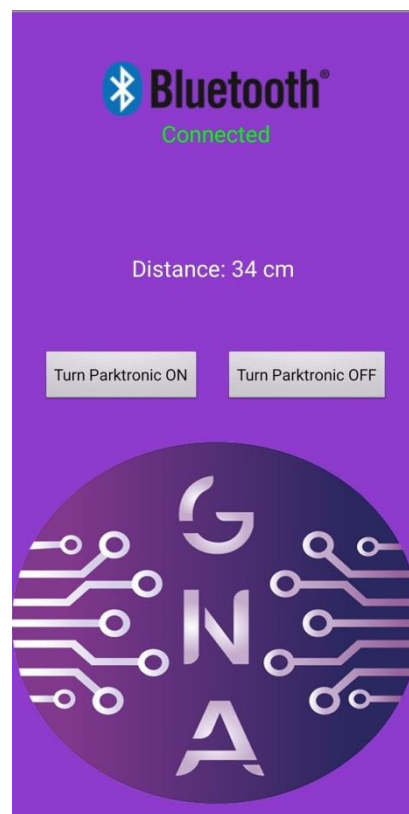
*Програмата за телефон може да се изтегли от този линк:

<http://ai2.appinventor.mit.edu/#5858491692875776>

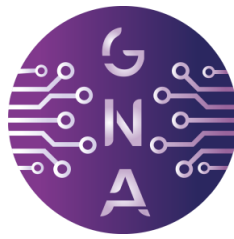
*Програмата за компютър се намира в папка Parktronic Reciever



Фиг. 11



Фиг. 12



Заклучение

Проектът е реализиран, както като симулация на платформата Тинкъркат, така и като физическа реализация. Той изпълнява своето предназначение, а именно, да известява за обект на близко разстояние. Благодарение на Bluetooth модула и нашите разработени приложения, можете в реално време да следите на какво разстояние е даденият обект.

В бъдеще проектът може да се осъвършенства, било то като хардуер или като оптимизация на кода. При смяна на сензора с по-точен и прецизен, може да бъде пригоден за реална среда и използван в реални условия, което го прави по-полезен за потребителите.



Източници

Проект в тинкъркат: <https://www.tinkercad.com/things/7JfhwXpI5hH>

Информация за свързване с блутут модул:

https://howtomechatronics.com/tutorials/arduino/arduino-and-hc-05-bluetooth-module-tutorial/?fbclid=IwAR1_1eZSfjZQitZgRC7yd5r7N2dthwKPhL3r-LBg9OgSeHAuaP8YpkHWO4E

Идеи за кода и за приложенията са ползвани от: <https://stackoverflow.com>