



# **ДИПЛОМНА РАБОТА**

**НА ТЕМА:**

## ***ИНТЕЛИГЕНТНА СИСТЕМА ЗА КОНТРОЛ НА ДОСТЪПА С ИДЕНТИФИКАЦИЯ ЧРЕЗ СМАРТФОН***

**Дипломант:**

Дани Даниелов Димитров

**Ръководител:**

*инж. Венцислав Бойков Начев*

**Курсов номер:** 18107

**Специалност:** код 4810201 “Системно програмиране“

**Правец, 2023**

## Съдържание

Увод.....	3
Глава 1 Литературен обзор .....	4
Глава 2 Използвани технологии.....	7
Глава 3 Разработка и реализация .....	26
Заклучение .....	32
Използвани източници .....	33

## Увод

Тази дипломна работа има за цел създаването на врата с опция за отваряне чрез смартфон с помощта на развойна платка Arduino UNO. Вратата е изцяло създадена за домашна употреба, но може да се използва и в други среди. За реализиране на проекта ще бъдат използвани: приложение за телефон създадено с *MIT App Inventor* , Arduino IDE, Arduino UNO, Bluetooth модул, реле, електромагнитна брава.

Функциите на вратата са: при натискане на бутона за отваряне в мобилното приложение, бравата прибира резето и можеш да отвориш вратата, при натискане на бутона за затваряне, резето се връща и вратата не може да бъде отворена.

Когато влезеш в мобилното приложение те посреща страница за влизане, в която трябва да въведеш парола. При въвеждане на правилна парола те допуска до следващата страница, където се намират: бутон за отваряне на списък с наличните наоколо Bluetooth устройства и двата бутона за отваряне и затваряне. При въвеждане на грешна парола не те допуска до тази страница.

# Глава 1 Литературен обзор

Последните години дигитализирането и автоматизирането започна да навлиза и в строенето на домове. Така наречените Smart home системи, да можеш да контролираш, прахосмукачката, телевизора и други неща през смартфона си, е много иновативно.

Интелигентните системи за достъп предлагат много помощ за потребителите, като осигуряват по-удобен и сигурен начин за контрол на достъпа до сгради, помещения или други ограничени области. Някои от техните функции включват:

**Безопасност** - интелигентните системи за достъп могат да подобрят нивото на безопасност, като използват технологии като видео наблюдение, входни системи за контрол на достъпа и високотехнологични сензори. Тези системи могат да бъдат програмирани да позволяват достъп само на определени потребители или групи потребители, като се използват карти, биометрични идентификатори или други методи.

**Управление на достъпа** - интелигентните системи за достъп могат да осигурят по-добро управление на достъпа до помещения, сгради или други ограничени области. Те могат да бъдат програмирани да отварят или заключват вратите автоматично в определени часове, да създават допълнителни нива на достъп, като задаване на временни или персонализирани кодове за достъп, или да използват групово управление на достъпа.

**Удобство** - интелигентните системи за достъп предлагат по-удобен начин за контрол на достъпа до помещения или сгради. Потребителите могат да отключват вратите на дома си, като използват мобилните си устройства, като например телефони или таблети, което е особено полезно, когато не са в близост до ключа си.

**Ефективност** - интелигентните системи за достъп могат да осигурят по-ефективен начин за управление на достъпа до помещенията. Те могат да бъдат програмирани да осигурят автоматично разрешение за достъп на работници или гости, като се използват временни кодове за достъп или други методи. Това може да бъде особено полезно за организации или фирми, които имат много работници или гости.

Чрез тези системи можеш да спестиш много време и мъки, понеже нещата стават много по-бързо, пък и всичко може да се намира в джоба ти на едно устройство.

Винаги е било много дразнещо, когато тъкмо си легнеш и се сетиш, че не си заключил входната врата. С тази технология няма да има нужда да ставаш от кревата за да я заключиш, защото ще можеш да го направиш през телефона.

В ежедневието ни е досадно да носим ключове в нас, пък и доста често се губят, но вече може да се направи по много други различни начини, примерно: пръстов отпечатък, лицево разпознаване, идентификация с карта, идентификация със смартфон и т.н. Така се премахва нуждата от ключ и нещата стават много по-лесни.



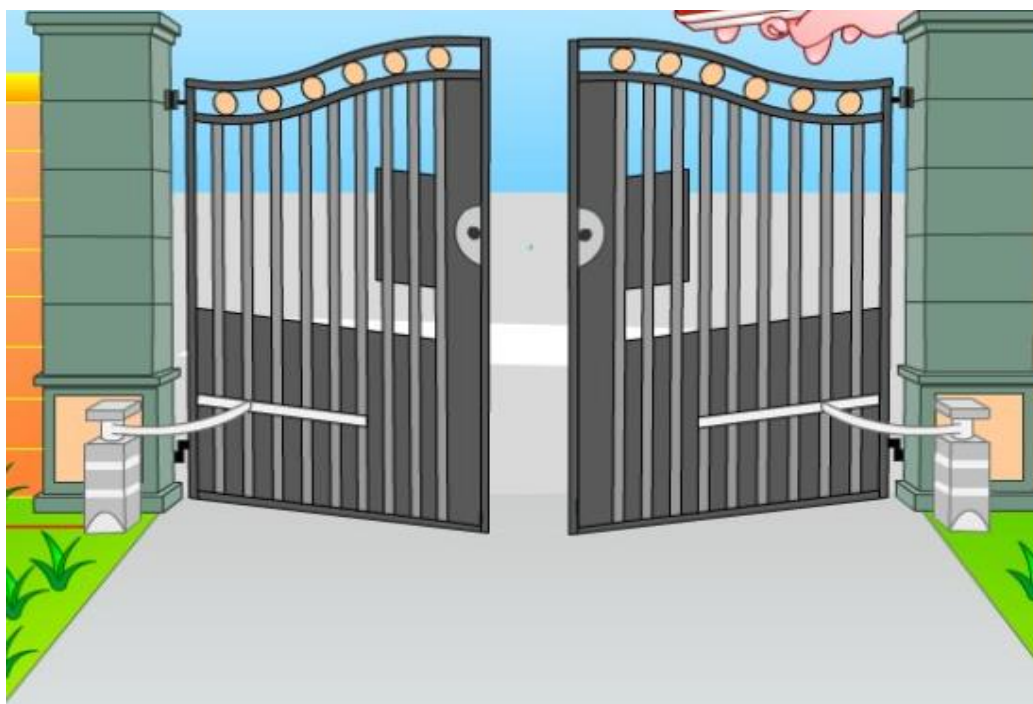
*Фиг. 1.1 Брава с пръстов отпечатък, опция за чип карта, код за достъп или wi-fi.*

Има много примери за врати за гаражи с електронно управление които работят по близък на този принцип начин. В повечето случай се прави с щора електро мотор и управлението се случва чрез дистанционно.



*Фиг. 1.2 Гаражна врата с дистанционно електронно управление.*

Има и порти за дворове с дистанционно управление, при които със натискането на копчето на дистанционното за отваряне се задействат два двигателя и отварят вратите. Когато се сложи порта с електро мотори отварянето на ръка се изключва, защото моторите имат голямо съпротивление и е много трудно да отвориш вратата, за тази цел се слага алтернативен вариант за отваряне, при изгубено дистанционно примерно, това се случва със жило което като издърпаш да се отпусне напрежението от моторите и да преминеш на ръчен режим.



*Фиг. 1.3 Врата с електро мотори, която се управлява през дистанционно*

## Глава 2 Използвани технологии

### 1. *Arduino*

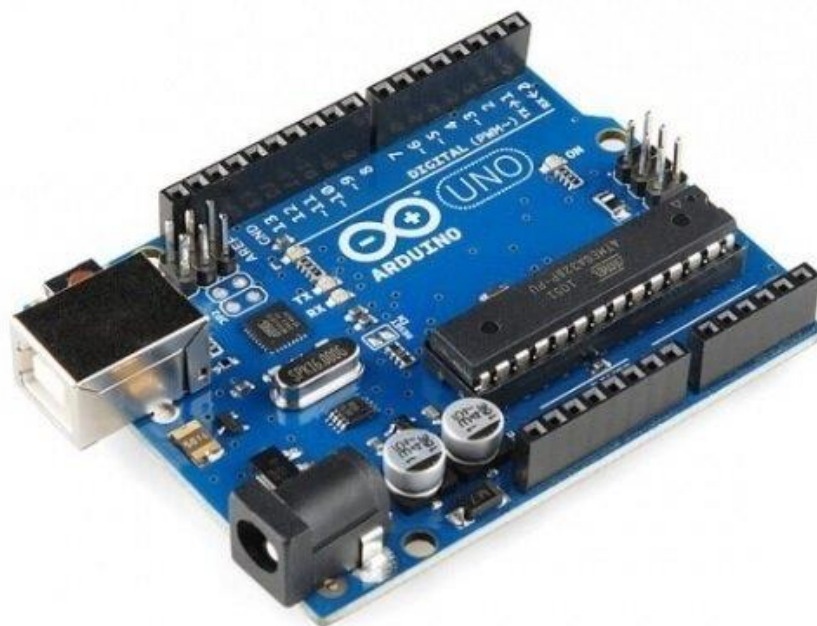
**Arduino** е проект с отворен код и едноименна компания, произлязла от него, както и свързаното с него потребителско общество. Основната цел е проектиране и производство на електронна платформа с лесен за ползване свободен хардуер софтуер, позволяващи постигането на интерактивност за неспециалисти.



фиг. 2.1 Фирмено лого на *Arduino*

**Arduino** се състои от 8-битов *Atmel AVR* микроконтролер с допълващи се компоненти, които улесняват програмирането и включването в други вериги. Важен аспект на **Arduino** платформата е наличието на стандартни конектори, които позволяват на потребителите да свързват CPU платката към голям набор от различни, взаимнозаменяеми модули, наречени разширения. Някои комуникират с **Arduino** директно, посредством различни конектори. Благодарение на I2C шина, няколко разширения могат да бъдат прикачени и използвани паралелно.

Повечето **Arduino** платформи функционират с 5V работно напрежение и разполагат с 14 цифрови входно-изходни порта, 6 аналогови входа, 16 MHz кварцов резонатор (въпреки че някои проекти работят с 8 MHz), четири светодиода, USB конектор, захранващ куплунг, бутон за рестартиране и ICSP конектор. Свързването с компютър става посредством USB кабел USB-A/USB-B.



Фиг. 2.2 Развойна платка Arduino UNO

- **Arduino UNO** е микроконтролерна развойна платка с ATmega328P AVR микроконтролер. Има 14 цифрови входно-изходни (I/O) порта, 6 аналогови входа, 16 MHz кварцов резонатор, четири светодиода (един потребителски, свързан на 13-и цифров I/O порт и три, които индикират работата на платката: ON, Tx и Rx), USB конектор, захранващ куплунг, бутон за рестартиране и ICSP конектор. Шест от цифровите I/O порта могат да се използват като PWM (ШИМ) изходи. Свързването с компютър се осъществява чрез USB кабел USB A – USB B. Uno може да се захранва през USB порта на компютъра или от външен източник, като превключването между различните начини за захранване е автоматично. Външният източник на захранване може да е DC адаптер 7-12V или батерия.

**Arduino UNO** има следните параметри:

- Микроконтролер: ATmega328P
- Работно напрежение: 5V
- Входно напрежение (препоръчително): 7-12V
- Входно напрежение (гранично): 6-20V
- Цифрови пинове: 14 (от които 6 могат да се използват за PWM изходи)
- Аналогови пинове: 6
- Ток на всеки цифров/аналогов пин: 20 mA
- Флаш памет: 32 KB (ATmega328P, от които 0,5 KB са заделени за bootloader)
- SRAM: 2 KB (ATmega328P)
- EEPROM: 1 KB (ATmega328P)
- Тактова честота: 16 MHz





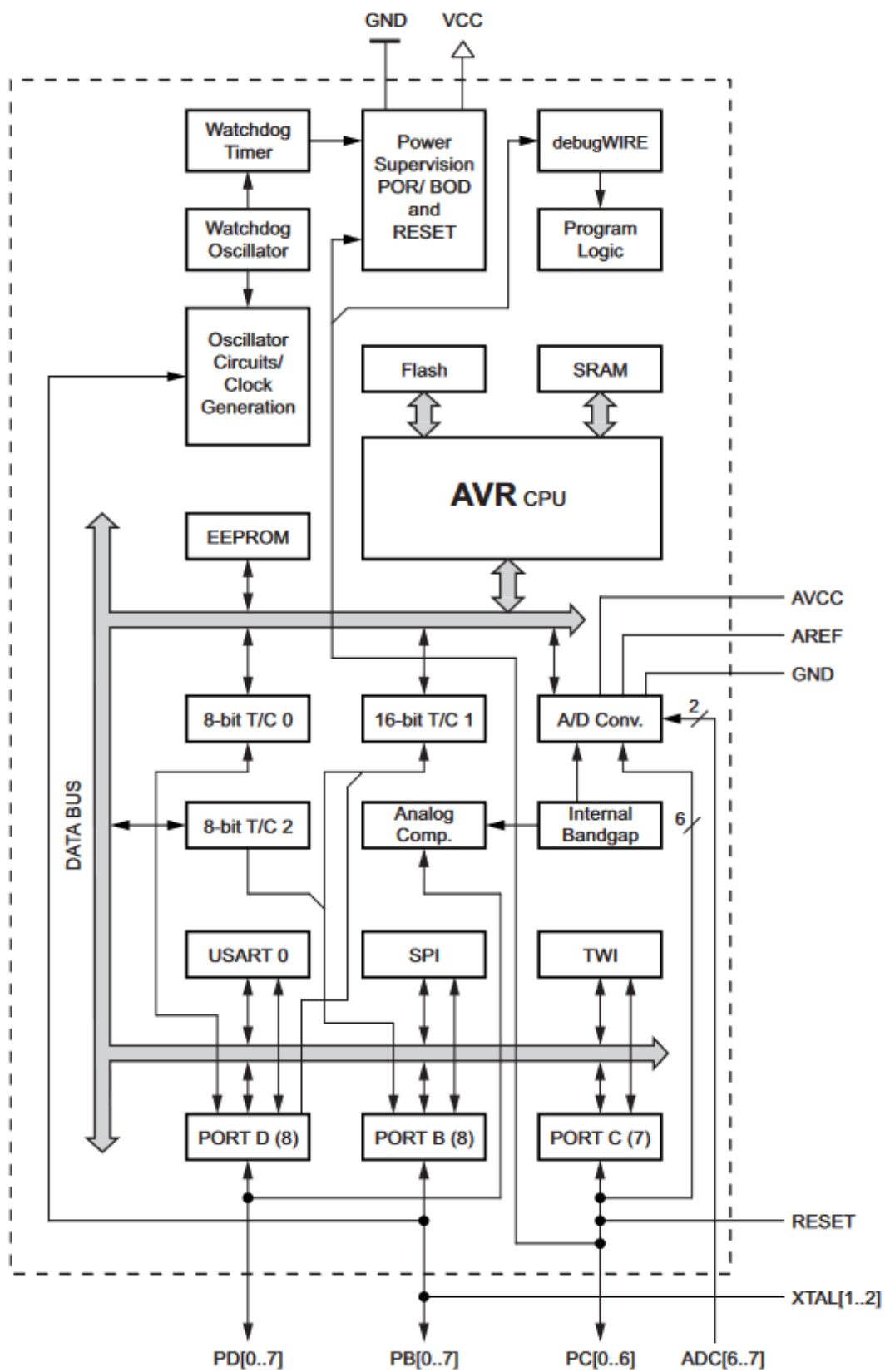
### ○ *ATmega328P*

ATmega328P е микроконтролер на Atmel (който сега е част от Microchip Technology), който се използва в много проекти на Arduino.

Ето пълните му параметри:

- Архитектура: AVR
- Работно напрежение: 1,8V - 5,5V
- Централен процесор: 8-битов AVR
- Честота на тактовия генератор: до 20 MHz при 5V захранване, до 10 MHz при 3V захранване
- Вградена памет:
  - Флаш памет: 32 KB
  - SRAM: 2 KB
  - EEPROM: 1 KB
- Комуникационни интерфейси:
  - 1 x UART
  - 1 x SPI
  - 1 x I2C/TWI
- Аналогови канали за измерване: 8
- Разделителна способност на АЦП: 10 бита
- PWM канали: 6
- Брой цифрови входни/изходни пинове: 23
- Разход на ток:
  - Работен режим: 1.8 mA при 1 MHz, 5V
  - Спящ режим: 100 nA
- Размер на корпуса: 28-пинов DIP, 32-пинов TQFP

Това са някои от основните параметри на ATmega328P. Микроконтролерът е често използван в много приложения, поради гъвкавостта и високата му производителност.



Фиг. 2.4 ATmega328P блокова схема.

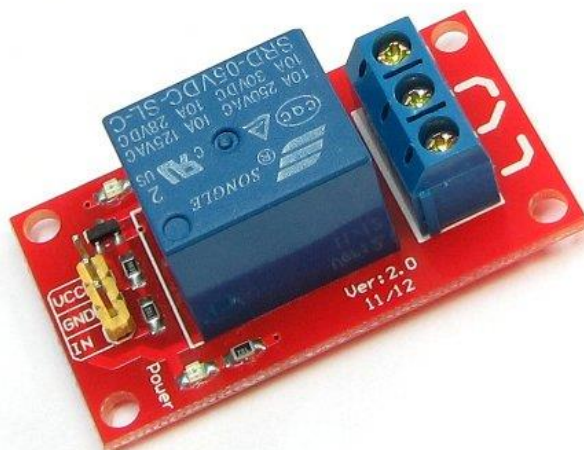
На *Arduino* може да се правят и много други различни проекти, в зависимостта от креативността ти и въображението. Примерно може да си създадеш количка която да бъде контролирана от телефона



*Фиг. 2.5 Bluetooth управляема количка с Arduino*

## ***2. Реле***

Релето е електромеханичен превключвател. Съществуват различни категоризации на релетата. По разпространените са според управляемата величина: токови релета, напреженови релета, реле за мощност, реле за налягане (пневматичен пресостат), реле за контрол на изолацията и др.; по принципа на работа: електромагнитни релета, магнитоелектрически релета, електродинамични релета, индукционни релета. Съществува и отделен клас полупроводникови апарати, които при изменение на определени физични параметри на средата, превключват електрически вериги наричани също релета.



Фиг. 2.6 Реле модул

Релето се състои от две части - обвивка и контакт. Обвивката е отворен електрически кръг, чието основно напрежение е ток, а контакта е механичен или електронен компонент, който се прехвърля от обвивката към друг електрически кръг.

Релетата се контролират от контролния сигнал, който може да бъде електрически или механичен.

### ***3. Език за програмиране на Ардуино – базиран на C++***

C++ е език за програмиране от високо ниво, който е разширение на първичния език “C”. Той е обединява обектно-ориентираното програмиране и други допълнителни неща с функциите на езика C. C++ е език, който има възможности за програмиране на операционни системи, големи програмни системи, с високо ниво на функционалност, представяне и ефикасност.



*Фиг. 2.7 C++ лого*

C++ е езикът който се използва в софтуера на ARDUINO IDE (за което ще говорим малко по-късно), но там е доста улеснено и е направено с готови функции за директна употреба.

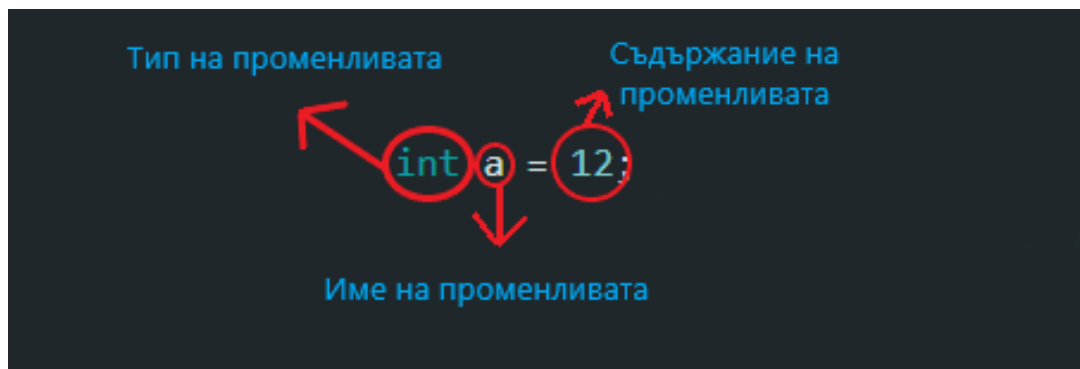
***Какво е променлива?***

```
int a = 12;  
char letter = 'a';  
double b = 3.14;  
bool condition = false;
```

*Фиг. 2.8 Различни променливи*

Основна градивна единица на езиците за програмиране са променливите. Известни от математиката, променливите служат за временно съхранение на данни в дадена програма.

### *Как се декларира променлива?*



Фиг. 2.9 Частите на една променлива и точно деклариране

- **Integer data type**

```
int a = 12;
```

Фиг. 2.10 Променлива от тип *Integer*, която запазва цели числа

Първият тип променливи са цели числа, те както и в математиката могат да бъдат използвани в различни функции и задачи. Те се декларират с ключовата дума *int*.

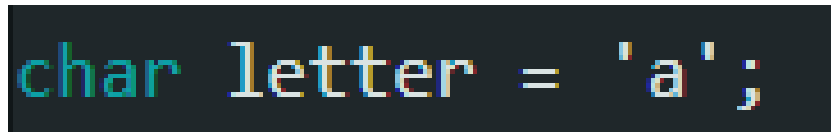
- **Floating-Point Types**

```
double b = 3.14;
```

Фиг. 2.11 Променлива от тип *double*, която запазва десетични числа.

Числа с плаваща запетая или десетични числа се декларираат с ключовата дума *double* или *float*. В този тип данни се запазват десетични числа.

- **Character Types**



```
char letter = 'a';
```

Фиг. 2.12 Променлива от тип *char*, която запазва символ

В този тип данни се запазва цифра или буква под формата на текст и не може да е повече от един символ.

- **Управляващи конструкции:**

C++ поддържа обичайните логически условия от математиката:

- Less than: `a < b`
- Less than or equal to: `a <= b`
- Greater than: `a > b`
- Greater than or equal to: `a >= b`
- Equal to: `a == b`
- Not Equal to: `a != b`

Фиг. 2.13 Видовете сравнения

Тези условия биват главно използвани в така наречените *if* изрази. Тези изрази се използват когато искаш да поставиш условие за това дали част от кода да бъде изпълнена, пример за такъв е : Ако бутон е натиснат бравата се отваря, обаче ако не е, не се отваря.



## ***4. Bluetooth***

Понеже нашата система ще се управлява през телефон се налага да се направи връзка между телефона и ардуиното. За тази цел ще използваме Bluetooth модул с който можем да свържем телефона и ардуиното, чрез Bluetooth технологията.

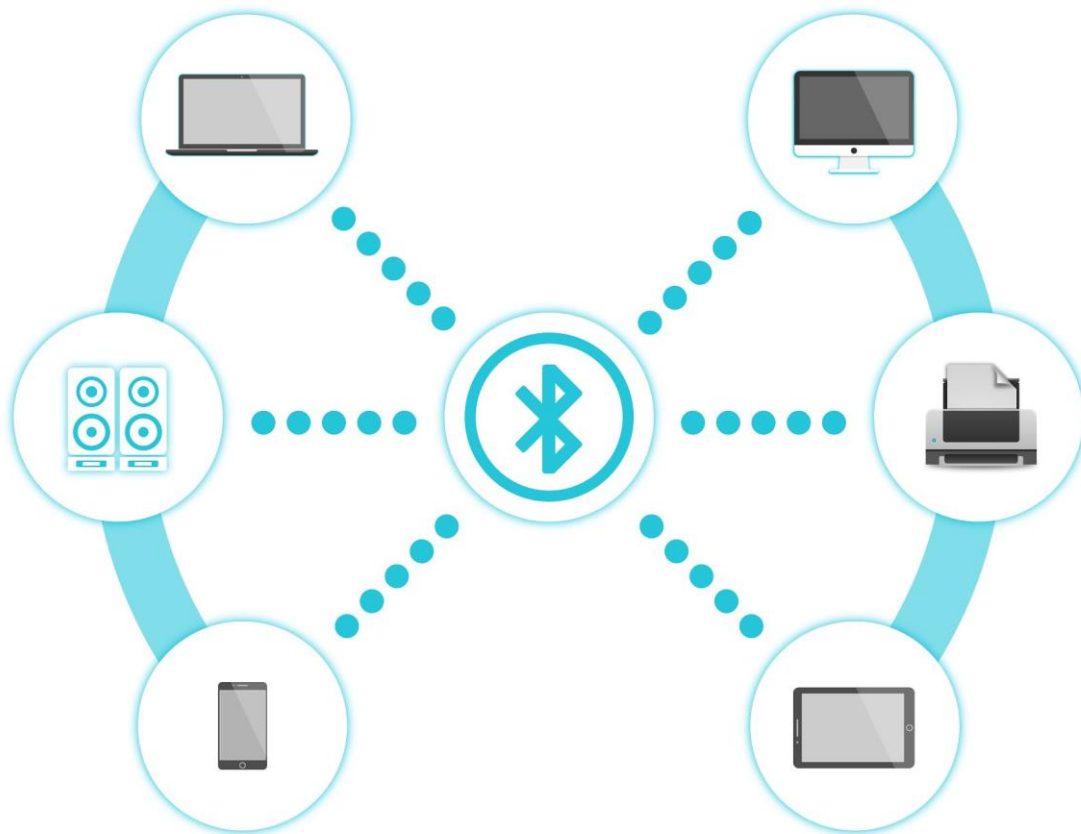


*Фиг. 2.14 Логото на bluetooth*

### ***Какво е Bluetooth?***

Bluetooth е безжична технология за обмен на данни между устройства, която позволява на потребителите да свързват устройства като смартфони, таблети, лаптопи, слушалки, колонки и други, без да изисква кабели или интернет връзка. Bluetooth технологията е разработена през 1994 година от една кройка компании включително Ericsson, Intel, Nokia и Toshiba и е наречена така заради името на датския крал Харалд Блютанд, който е известен с това, че е обединил многото племена в Дания през 10 век.

Bluetooth използва радиовълни в честотния диапазон 2,4 GHz, който е безлицензионен и може да бъде използван в много държави по света. Технологията позволява на две или повече устройства да се свържат към кратък обхват (от около 10 до 100 метра, в зависимост от класа на устройствата) и да обменят данни между тях, като същевременно могат да комуникират с други устройства. Комуникацията между устройствата може да бъде различна в зависимост от приложението - като пример може да се посочи предаване на гласови данни при използване на слушалки, пренос на данни между две лаптопи или свързване на телефон с кола чрез Bluetooth.



*Фиг. 2.15 Връзка на устройства чрез Bluetooth*

Bluetooth технологията се използва широко в множество устройства за комуникация и развлечения, като високоговорители, слушалки, микрофони, автомобилни системи, медицински устройства, устройства за домашно забавление и други. Също така, Bluetooth е съвместима с почти всички операционни системи на пазара, включително Windows, macOS, Linux, Android и iOS.

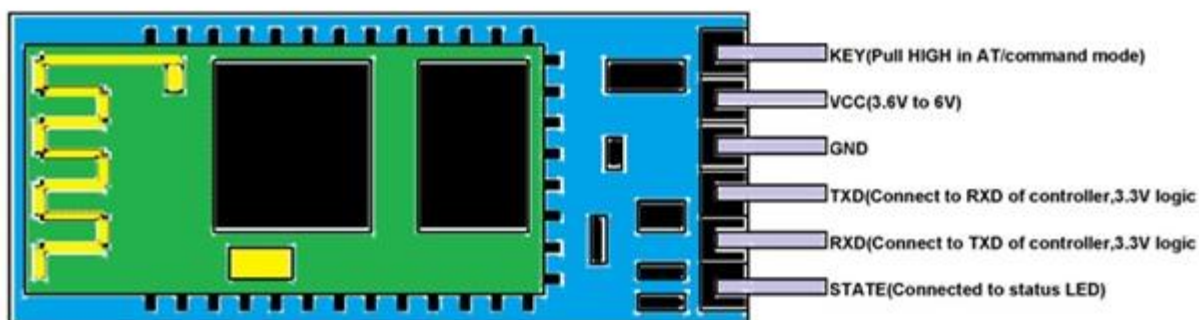
## ***5. HC-06 Bluetooth module***

HC-06 е блутут модул създаден за осъществяване на безжична връзка на къси разстояния до 100 м между два микроконтролера. Серийният блутут модул работи чрез Bluetooth 2.0 комуникационен протокол и може да се държи само като подчинено устройство. Скоростта на комуникация може да достигне до 2.1Mb/s. Серийната комуникация и захранването се осъществяват чрез цифтовата рейка разположена на модулната платка.



*Фиг. 2.16 HC-06 модул*

Пиновете имат следното разположение:

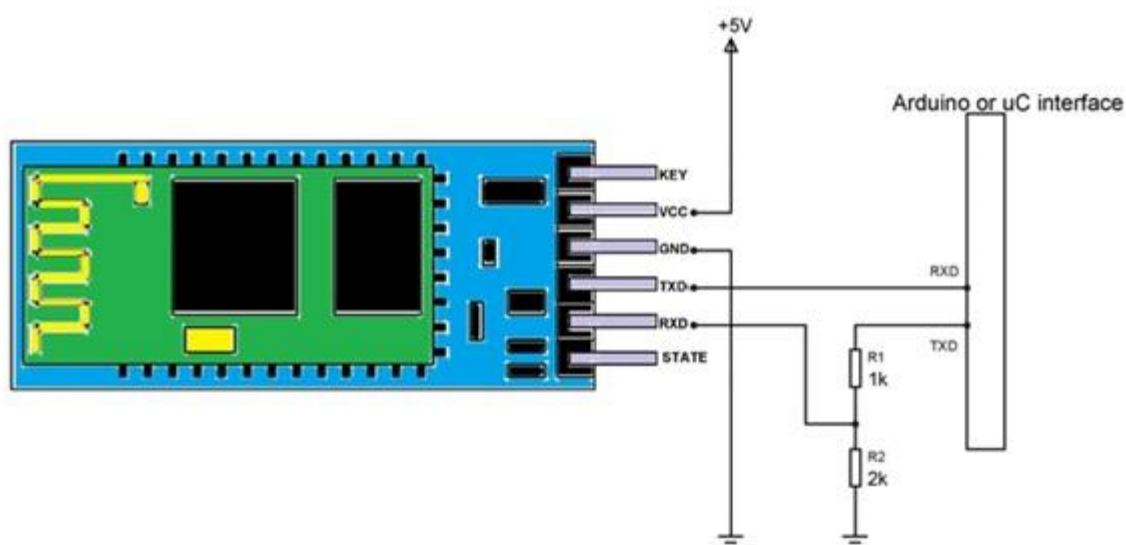


Фиг. 2.17 Пиновете на HC-06 и тяхното предназначение

И следната функционалност:

Пин	Наименование	Функция
1	Key	Високо ниво – режим на приемане на АТ команди. Ниско ниво или висящ - нормална работа на Bluetooth модул
2	VCC	5V захранващо напрежение
3	GND	Свързване към маса
4	TXD	Пин за изпращане на данни(3.3V)
5	RXD	Пин за получаване на данни(3.3V)
6	STATE	Свързан на светодиодиод, оказва състоянието на модула

Използване на модула:

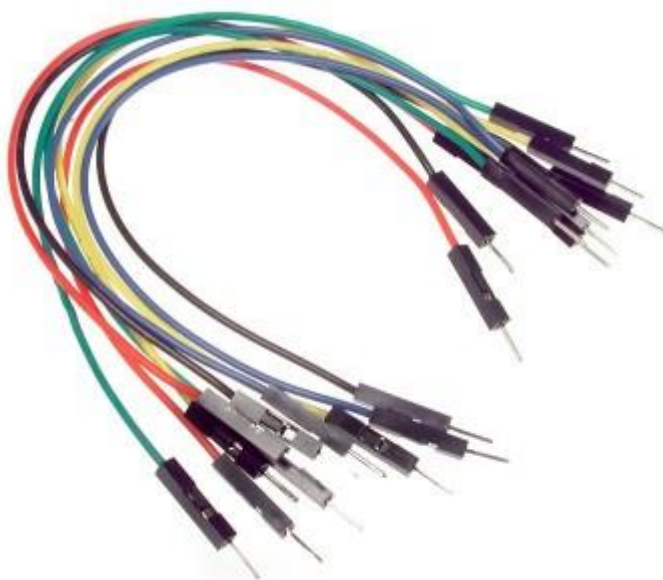


Фиг. 2.18 Схема, HC-06 с ардуино

Тъй като модул НС-06 е подчинен модул (Slave) е необходимо наличието и на друг комуникационен модул от тип Master или смартфон, чрез който да се потърси модул НС-06. Паролата за свързване с модул НС-06 е “1234” и може да бъде сменяна.

## ***6. Свързващи проводници***

За свързването на всичките компоненти ще използваме малки кабелчета познати повече като “Jumper cables”.



*Фиг. 2.19 Кабели за връзване на компоненти мъжко-мъжко*

Има различни видове кабели, тези на фигурата са мъжко-мъжко, в проекта се използват и мъжко-женско.

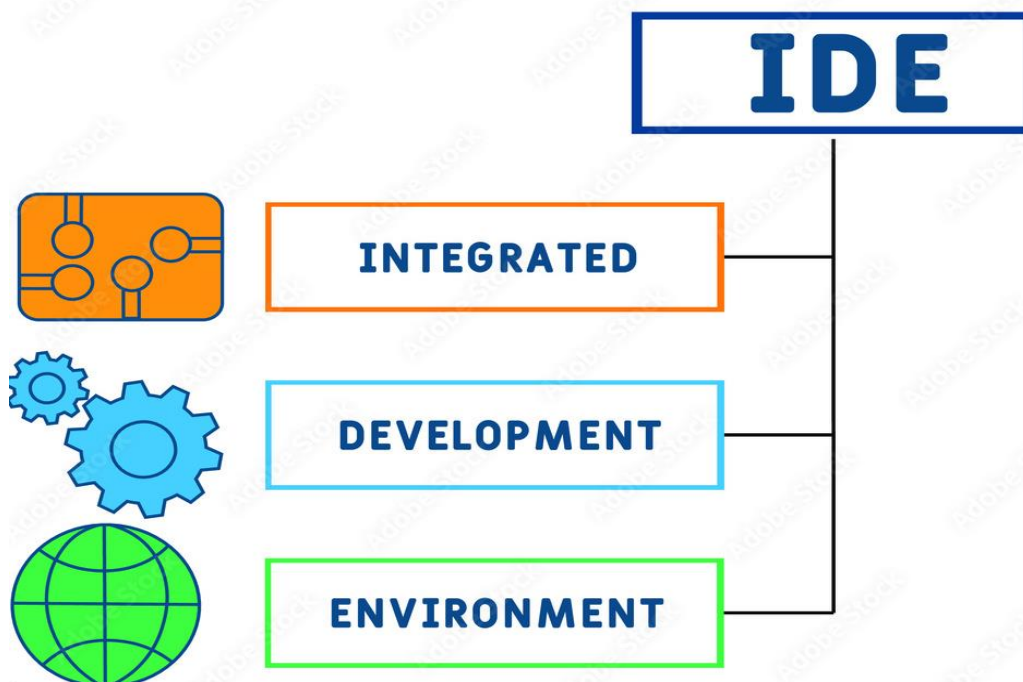


*Фиг. 2.20 Кабели за връзване на компоненти мъжко-женско*



## 7. Arduino IDE

- *Какво е IDE?*



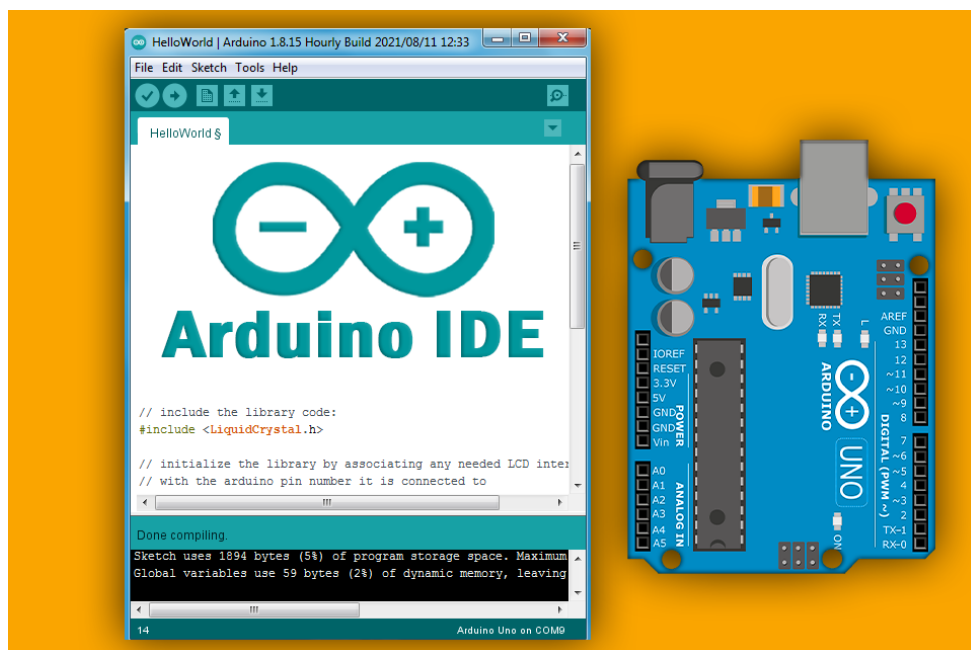
Фиг. 2.21 Какво е IDE?

IDE е абревиатура, която означава "Integrated Development Environment" (интегрирана среда за разработка). Това е програма, която предоставя инструменти на програмиста за разработка на софтуер. IDE се използва за създаване на различни видове софтуер, включително уеб приложения, мобилни приложения, настолни приложения и други.

IDE включва обикновено текстов редактор, средства за компилация и дебъгване на кода, графичен интерфейс за управление на файлове и проекти, интегрирани системи за управление на версиите и други инструменти, които помагат на програмистите да създават, тестват своите приложения.

Някои от най-известните IDE-та включват Visual Studio, Eclipse, IntelliJ IDEA, NetBeans и Xcode. В зависимост от програмния език и типа на софтуера, който програмистът разработва, може да бъде избрана и специализирана IDE.

- *Arduino IDE*



Фиг. 2.22 Arduino IDE

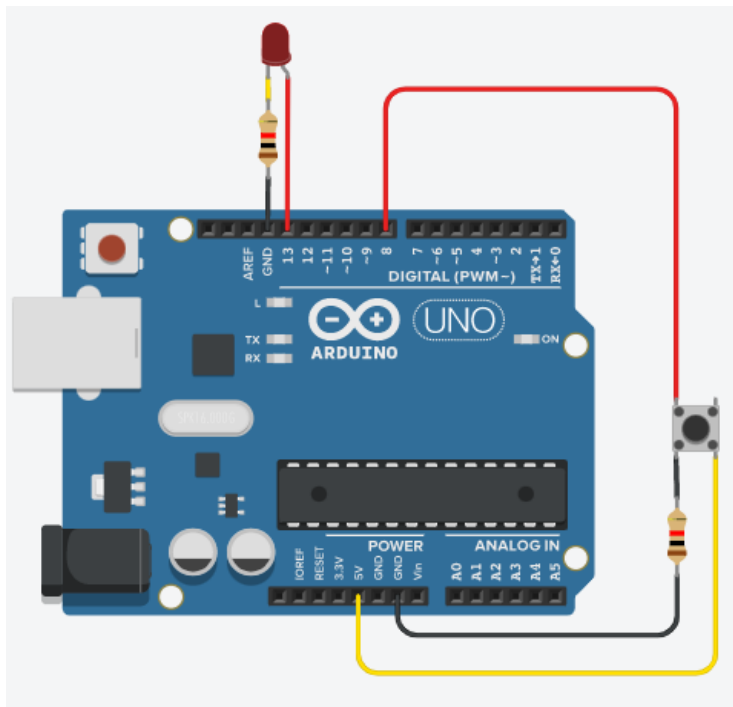
Arduino IDE е интегрирана среда за разработка на софтуер, която се използва за програмиране на Arduino микроконтролери. Това е безплатен софтуер, който може да бъде изтеглен и инсталиран на компютъра ви.

Някои от основните функции на Arduino IDE включват:

- Създаване на нови проекти и файлове за Arduino програми.
- Редактиране на програмен код в текстов редактор.
- Компилиране на програмния код в машинен код, който може да се изпълни на Arduino.
- Качване на компилирания код на Arduino микроконтролера, като използвате сериен порт или други методи на комуникация.
- Мониторинг на входящи данни от Arduino микроконтролера.

Arduino IDE поддържа синтаксиса на програмния език C++, който се използва за програмиране на Arduino микроконтролери. Също така, той има вградени библиотеки за управление на различни компоненти, като например светодиоди, сензори, мотори и дисплеи.

Arduino IDE е лесен за използване, дори за начинаещи програмисти, които нямат много опит в програмирането. Той също така има голяма общност от потребители, която може да предостави поддръжка и помощ за проблеми при програмирането на Arduino.



Фиг. 2.23 Схема с ардуино, бутон и лед лампа

```
sketch_apr18a.ino  sketch_apr18a.ino
1  int led = 13;
2  int button = 8;
3  void setup() {
4      pinMode(led,OUTPUT);
5      pinMode(button,INPUT);
6
7  }
8
9  void loop() {
10
11      if(digitalRead(button)== HIGH){
12          digitalWrite(led,HIGH);
13      }else{
14          digitalWrite(led,LOW);
15      }
16
17  }
```

Фиг. 2.24 Код от Arduino IDE, който го програмира да светка, когато е натиснат бутон.

## 8. MIT App Inventor

**MIT App Inventor** е безплатна онлайн среда за разработка на мобилни приложения за операционната система Android. Това е инструмент за програмиране, който позволява на потребителите без опит в програмирането да създават мобилни приложения чрез "блоково програмиране", което означава, че създаването на приложението става чрез влачене и поставяне на блокове с логически операции и функции.





*Фиг. 2.25 MIT APP INVENTOR лого*

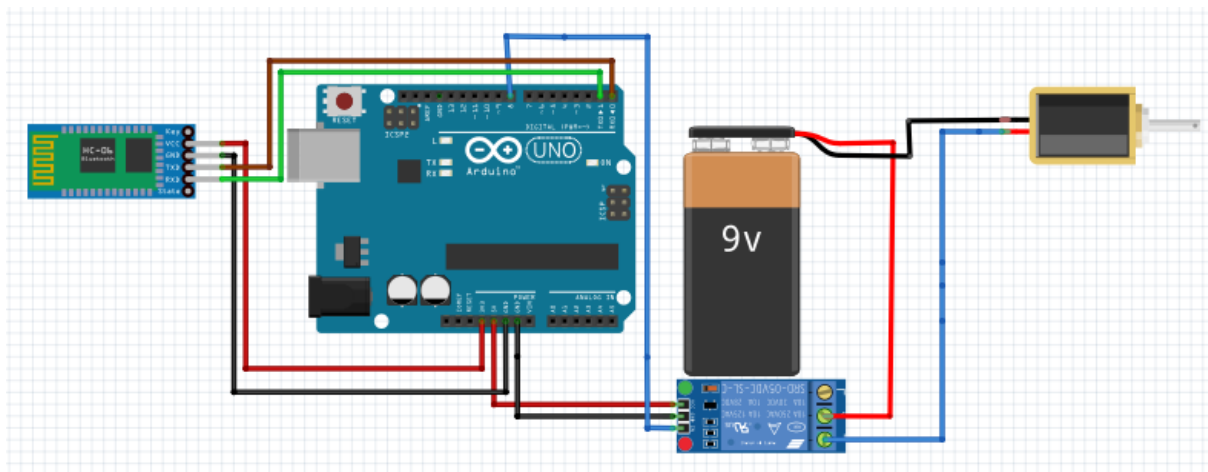
**MIT App Inventor** позволява на потребителите да създават приложения за Android с помощта на вградените компоненти на средата, като сензори, визуални елементи, бази данни и други. След като приложението е създадено, потребителите могат да го тестват директно на своите устройства или да го публикуват в Google Play Store.

**MIT App Inventor** е проектиран да бъде достъпен за ученици, студенти, преподаватели и хоби програмисти, като осигурява една лесна за употреба и интуитивна среда за създаване на мобилни приложения, без да е необходимо да се научават сложни програмни езици или да се инсталират сложни програмни инструменти.

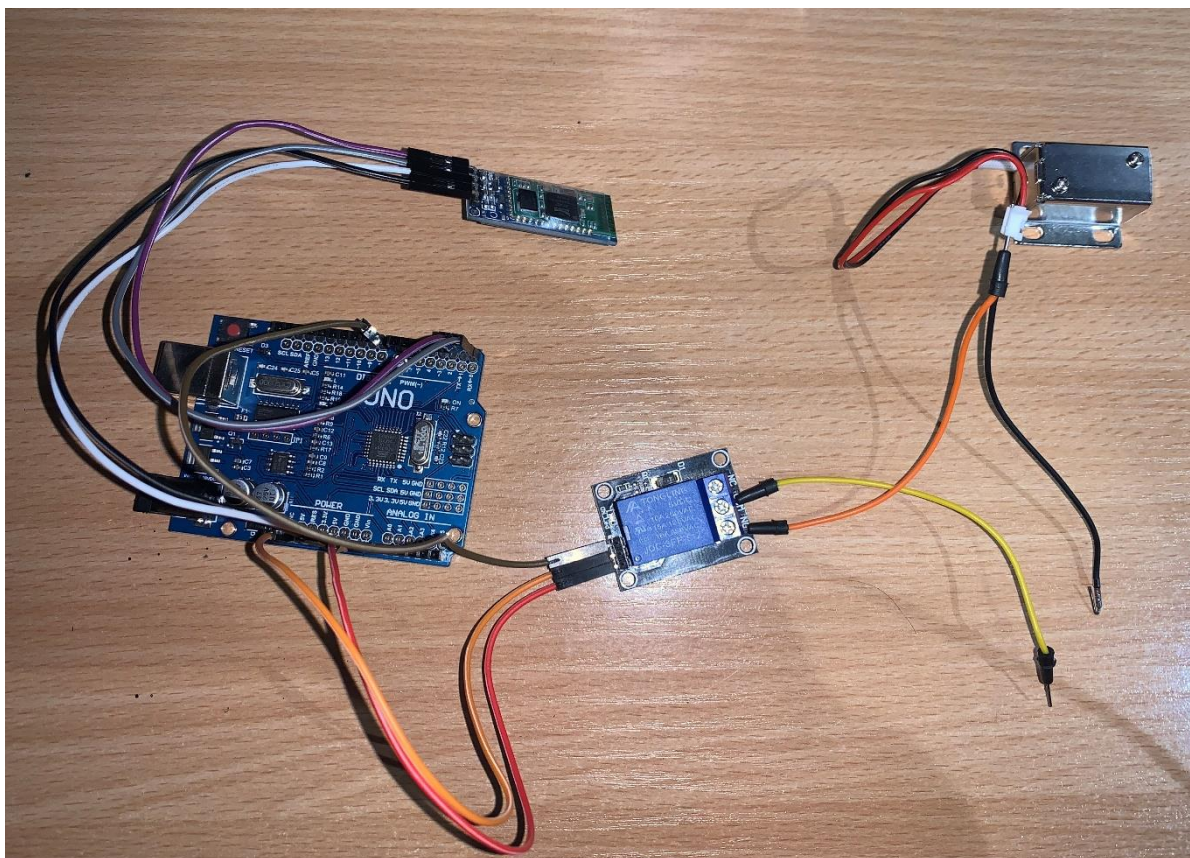
# Глава 3 Разработка и реализация

## 1. Хардуерна реализация

Това е схема на която е показано как са свързани различните компоненти от проекта.



Фиг. 3.1 Електрическа Схема



Фиг 3.2 Системата сглобена

## 2. Софтуерна реализация:

Този код е написан за Arduino и има за цел да прочете данни от серийния порт и да управлява бравата:

```
char Incoming_value = 0;

void setup(){
  Serial.begin(9600);
  pinMode(8, OUTPUT);
}

void loop(){
  if(Serial.available() > 0){
    Incoming_value = Serial.read();
    Serial.print(Incoming_value);
    Serial.print ("\n");
    if(Incoming_value == '1'){
      digitalWrite(8, HIGH);
    }else if(Incoming_value == '0'){
      digitalWrite(8, LOW);
    }
  }
}
```

Фиг. 3.3 Код, който е качен на ардуиното

### Какво прави кода?

```
1 char Incoming_value = 0;
2
```

Фиг. 3.4 Дефиниране на променлива

Тази линия дефинира променлива "incoming\_value" като char с начална стойност 0.

```

3 void setup(){
4     Serial.begin(9600);
5     pinMode(8, OUTPUT);
6 }
7

```

Фиг. 3.5 Void setup

Тази част от кода се изпълнява само веднъж при стартирането на устройството. Тя задава скоростта на серийния порт (9600 бита/с) и определя пин 8 като изходен пин.

```

8 void loop(){
9     if(Serial.available() > 0){
10         Incoming_value = Serial.read();
11         Serial.print(Incoming_value);
12         Serial.print("\n");
13         if(Incoming_value == '1'){
14             digitalWrite(8, HIGH);
15         }else if(Incoming_value == '0'){
16             digitalWrite(8, LOW);
17         }
18     }
19 }
20

```

Фиг. 3.6 Void loop

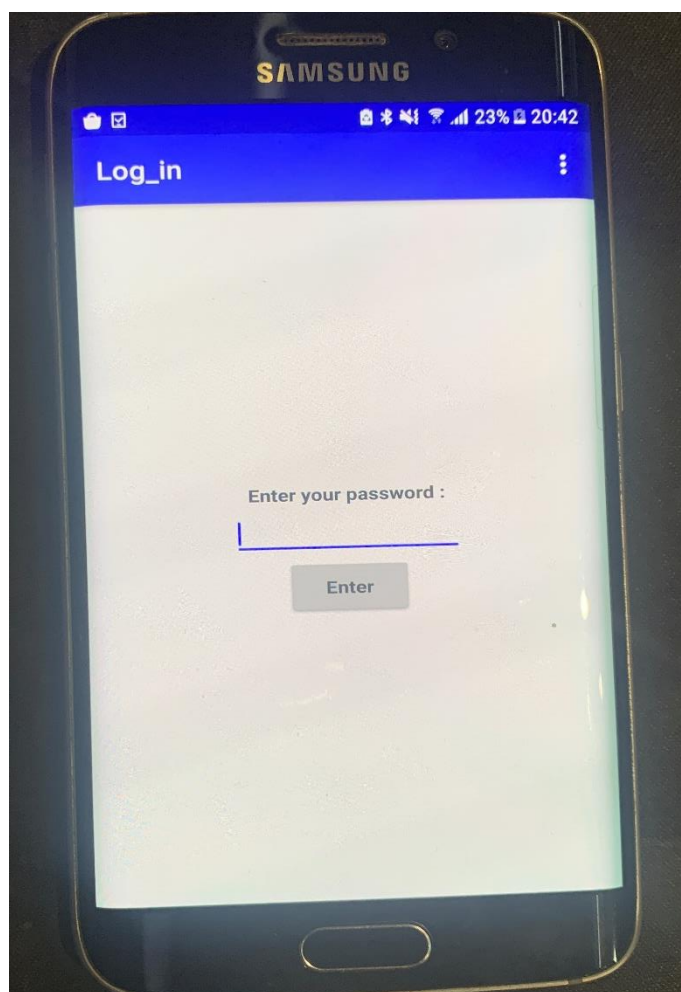
Този блок от кода се изпълнява постоянно в цикъл, докато устройството е включено. Той проверява дали има налични данни във серийния порт, използвайки функцията **Serial.available()**. Ако има налични данни, те се прочитат чрез функцията **Serial.read()**, след което се изпращат обратно на серийния порт чрез функцията **Serial.print()**.

След като се изпратят данните, програмата проверява стойността на "incoming\_value", която представлява получената стойност от серийния порт. Ако стойността на "incoming\_value" е '1', тогава релето превключва чрез функцията digitalWrite(8, HIGH) и подава ток на електромагнитната бртва, която се отваря. Ако стойността на "incoming\_value" е '0', тогава релето не подава ток, като сменя своята посока чрез функцията digitalWrite(8, LOW).

Така, ако изпратим '1' чрез серийния порт, бравата ще се отвори, ако изпратим '0', тя ще се затвори. Този код е добър пример за управление на периферни устройства чрез комуникация по сериен порт на Arduino.

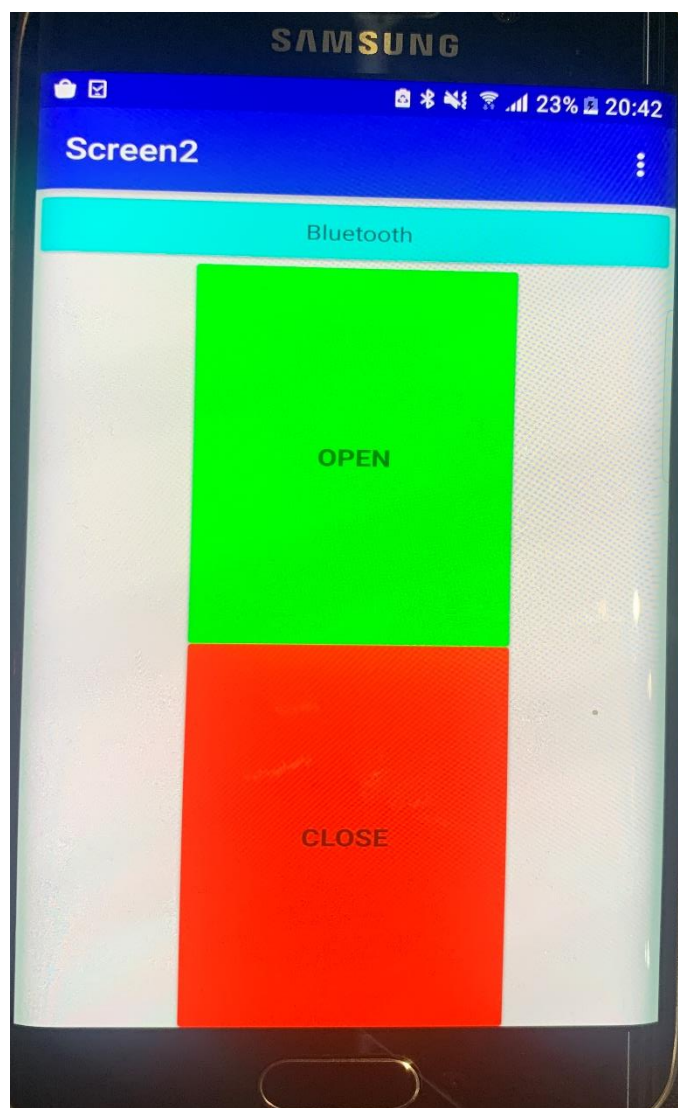
### ○ Приложение за телефон

Приложението както вече казахме е направено с MIT App Inventor. Първо, за да има допълнителна защита, добавих Log in екран, който изисква парола, за да продължиш напред.



Фиг. 3.7 Първата страница от приложението –  
Log in екран

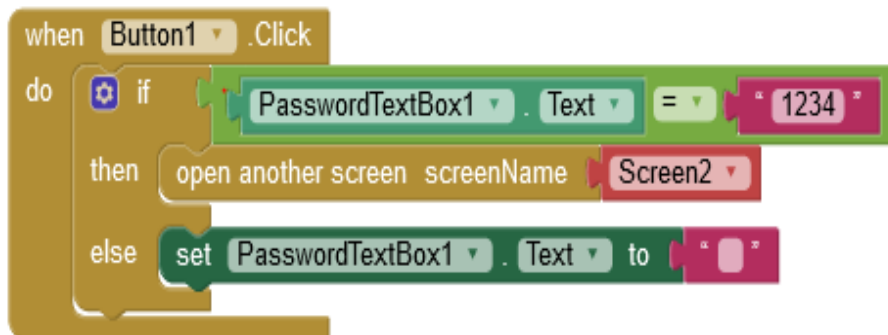
При въвеждане на правилна парола това те допуска до следващата страница, която има два бутона, за отваряне и затваряне и един бутон, който отваря списък с наличните Bluetooth устройства.



Фиг. 3.8 Втората страница от приложението –  
Същинска страница

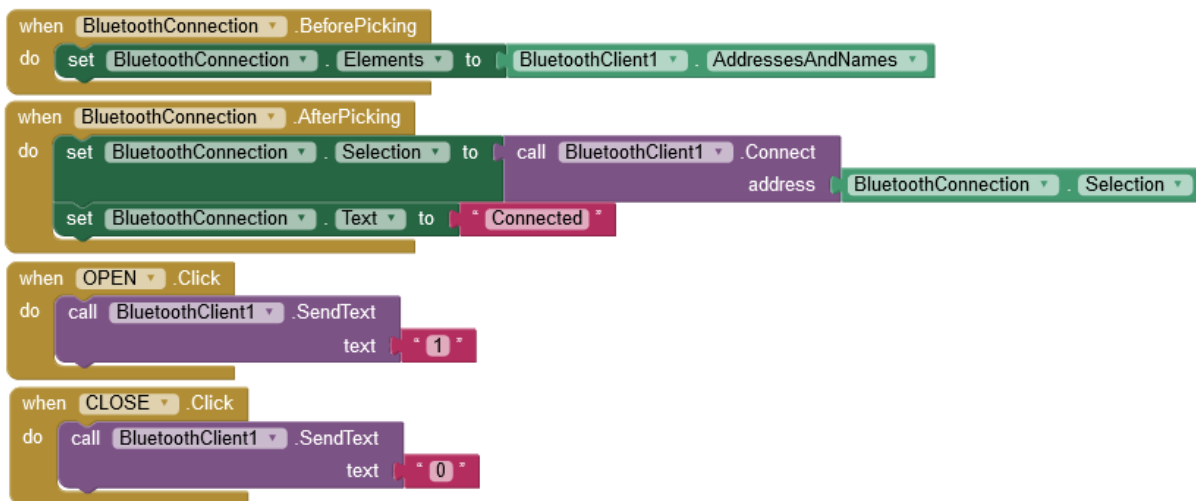


Логиката зад тези две страници изглежда по този начин :



Фиг. 3.9 Логика за първата страница

Логиката за първата страница не е сложна, когато натиснеш бутона “Enter”, се проверява дали въведената парола е правилна (в случая се проверява дали е “1234”) ако е правилна те препраща към втората страница , но ако не е, не те.



Фиг. 3.10 Логика за втората страница

Логиката зад втората страница е малко по-сложна. При натискане на бутона “Bluetooth”, се отваря списък с устройства, които са близо до теб и можеш да се свържеш към тях. След избиране на устройство и свързване, текста на този бутон се променя от “Bluetooth” на “Connected”. След свързване към блутут модула HC-06 можеш вече чрез натискането на двата бутона да контролираш електромагнитната брава.

## Заклучение

В бъдеще се планира промяна на работата на проекта, като ще бъде реализиран в по-голям мащаб и ще бъде направено да работи с интернет връзка вместо Bluetooth. Също така ще бъдат добавени и още няколко разновидности на отваряне, примерно с чип или карта.



## Използвана литература

- [1] <https://bg.wikipedia.org/wiki/%D0%90%D1%80%D0%B4%D1%83%D0%B8%D0%BD%D0%BE>
- [2] <https://bg.wikipedia.org/wiki/C%2B%2B>
- [3] <https://www.blog.vikiwat.com/relay/>
- [4] <https://exallto.com/elektronika/215-bezzhichen-bluetooth-priemo-predavatelen-modul-rs232.html>
- [5] [https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P\\_Datasheet.pdf](https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf)
- [6] <http://www.lmpt.univ-tours.fr/~volkov/C++.pdf>