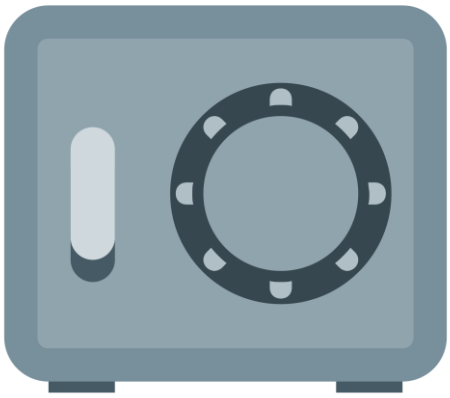
Inhaltsverzeichnis



1. Projektbeschreibung

Einen eigenen Safe bauen

Diesen stelle ich mir wie folgt vor:

Ich will einen eigenen Holzkasten bauen mit einer Tür, an der ich innen ein Schiebeschloss festschraube. Dieses Schiebeschloss soll mit einem Servomotor geöffnet/geschlossen werden können.

Außerhalb des Safes schraube ich dann ein Zahlenfeld fest, welcher dafür zuständig ist den Zugang zum Safe zu gewähren/abzulehnen.

Über dem Zahlenfeld befestige ich noch ein Display der den Zustand des Safes anzeigt (z.B. Geschlossen/Offen, Richtiger Code, Falscher Code).

Dazu noch einen Buzzer der ein Geräusch auslöst (z.B. bei falscher Code Eingabe).

Ein RGB-Licht befestige ich zusätzlich neben dem Display, welches Grün leuchtet beim Zugang gewähren und Rot beim Zugang ablehnen/Schloss verriegeln.

Der Mikrocontroller wird logischerweise innen befestigt sein.

Wie ich dies verwirklicht habe, zeige ich auf den folgenden Seiten.

2. Die Komponenten

1x Arduino UNO

1x RGB-Licht

1x Passive Buzzer

1x Servomotor

1x LCD-Display

1x (3x4) Zahlenfeld

1x Breadboard

18x Female-to-Male Kabel

5x Male-to-Male Kabel

7-9 Male-to-Male Headers

Draht

6 DIN-A4 Holzplatten (3mm dick)

4 Holzbalken (21cm*2cm*2cm)

4 Holzbalken (17cm*2cm*2cm)

1 kleiner Holzbalken (8cm*2cm*2cm)

44 Nägel

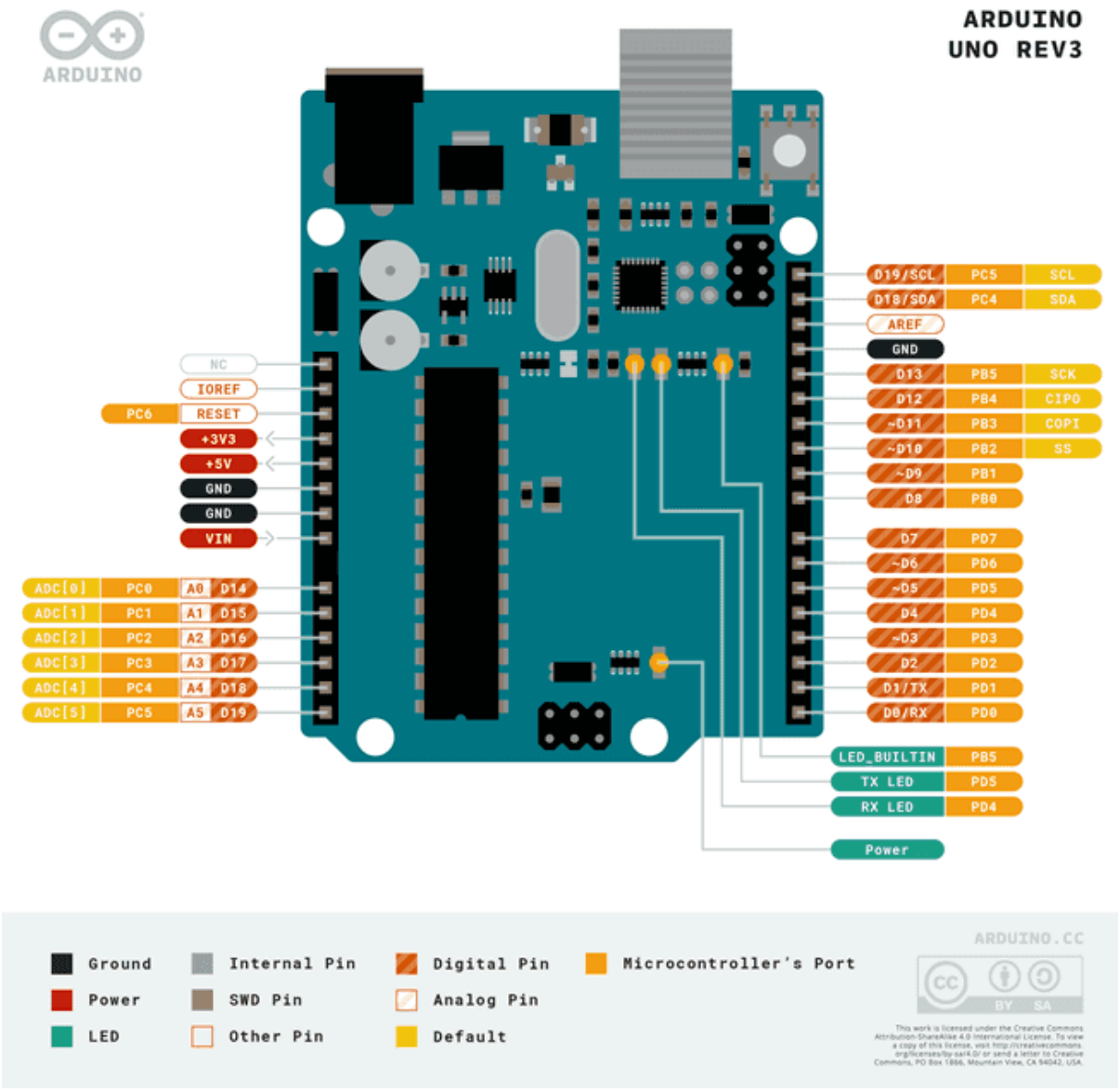
Schiebeschloss und 2 Scharniere

Heißklebepistole

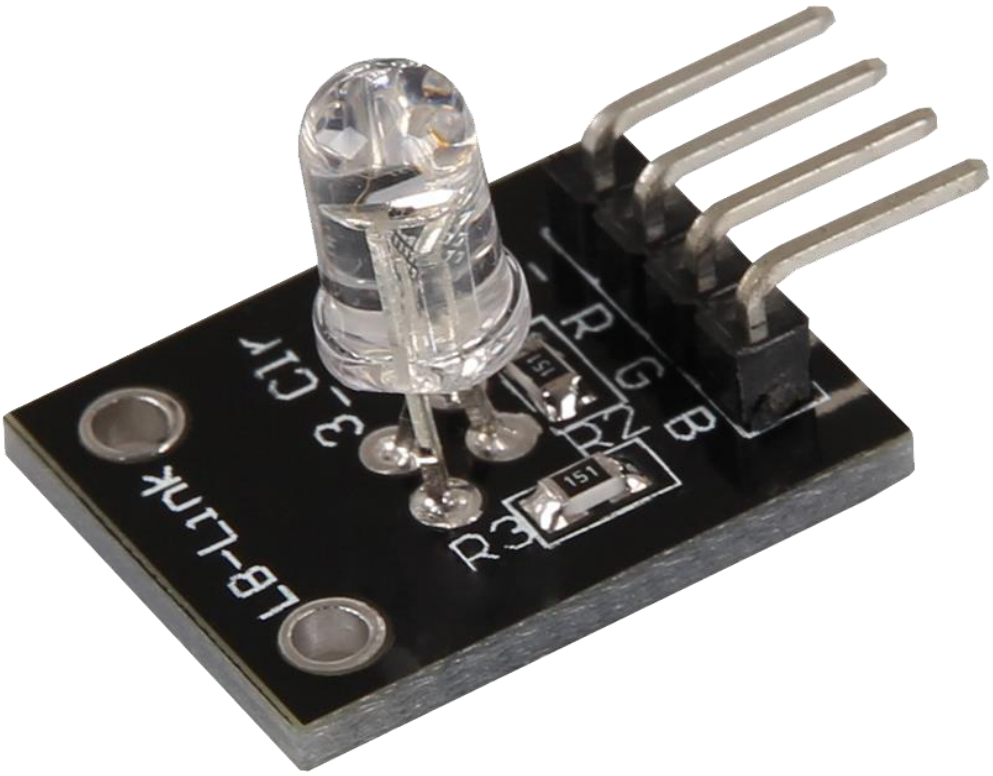
Lötwerkzeug

Akku/Batterien kompatibel mit Arduino

Arduino UNO R3 Board

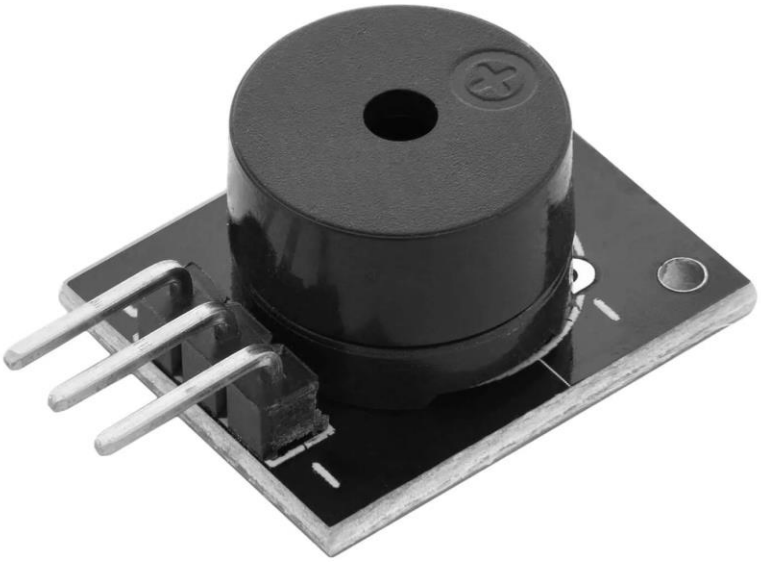


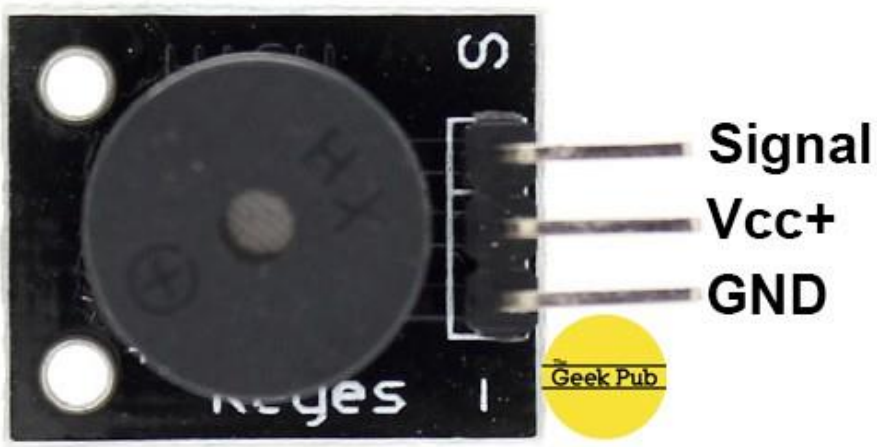
RGB-Licht



Pin Name	Description
"R"	Red light
"G"	Green light
"B"	Blue light
"-"	Ground

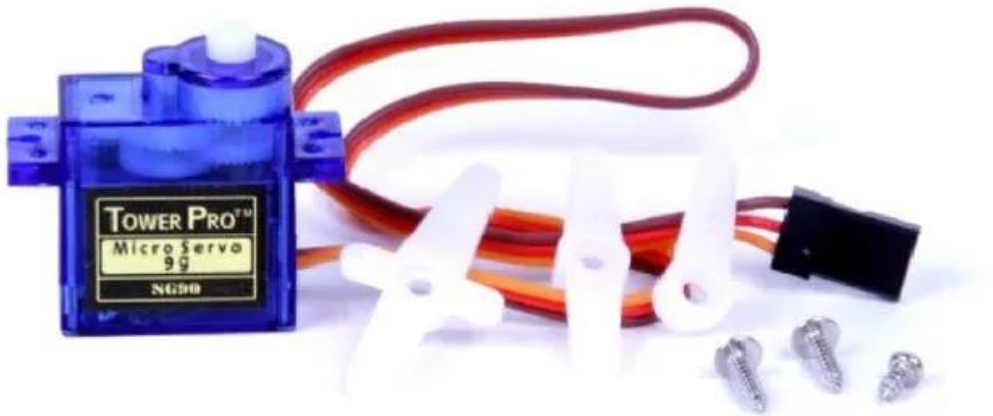
Passive Buzzer



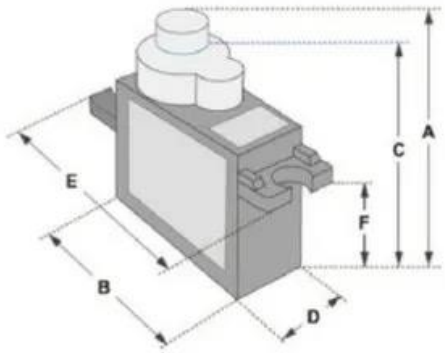


Servomotor

SERVO MOTOR SG90 DATA SHEET

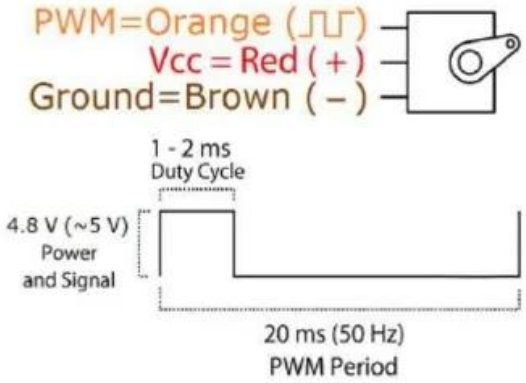


Tiny and lightweight with high output power. Servo can rotate approximately 180 degrees (90 in each direction), and works just like the standard kinds but smaller. You can use any servo code, hardware or library to control these servos. Good for beginners who want to make stuff move without building a motor controller with feedback & gear box, especially since it will fit in small places. It comes with a 3 horns (arms) and hardware.

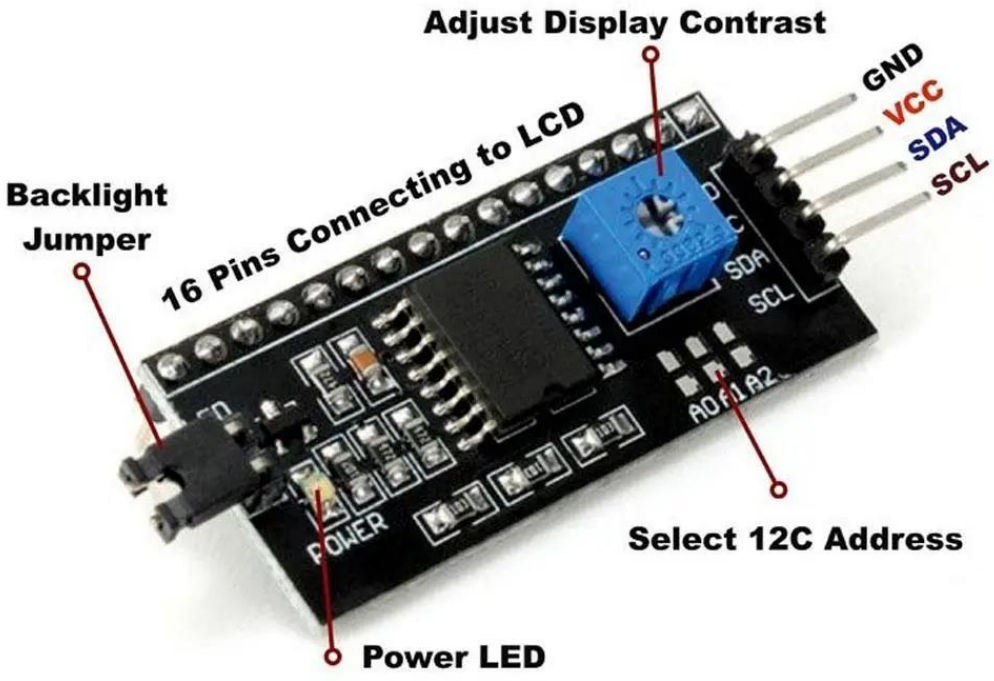


Dimensions & Specifications
A (mm) : 32
B (mm) : 23
C (mm) : 28.5
D (mm) : 12
E (mm) : 32
F (mm) : 19.5
Speed (sec) : 0.1
Torque (kg-cm) : 2.5
Weight (g) : 14.7
Voltage : 4.8 - 6

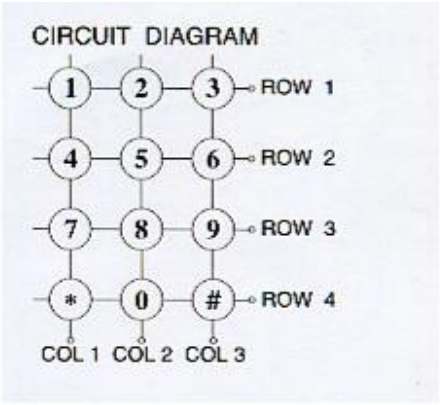
Position "0" (1.5 ms pulse) is middle, "90" (~2ms pulse) is middle, is all the way to the right, "-90" (~1ms pulse) is all the way to the left.



LCD-Display I2C



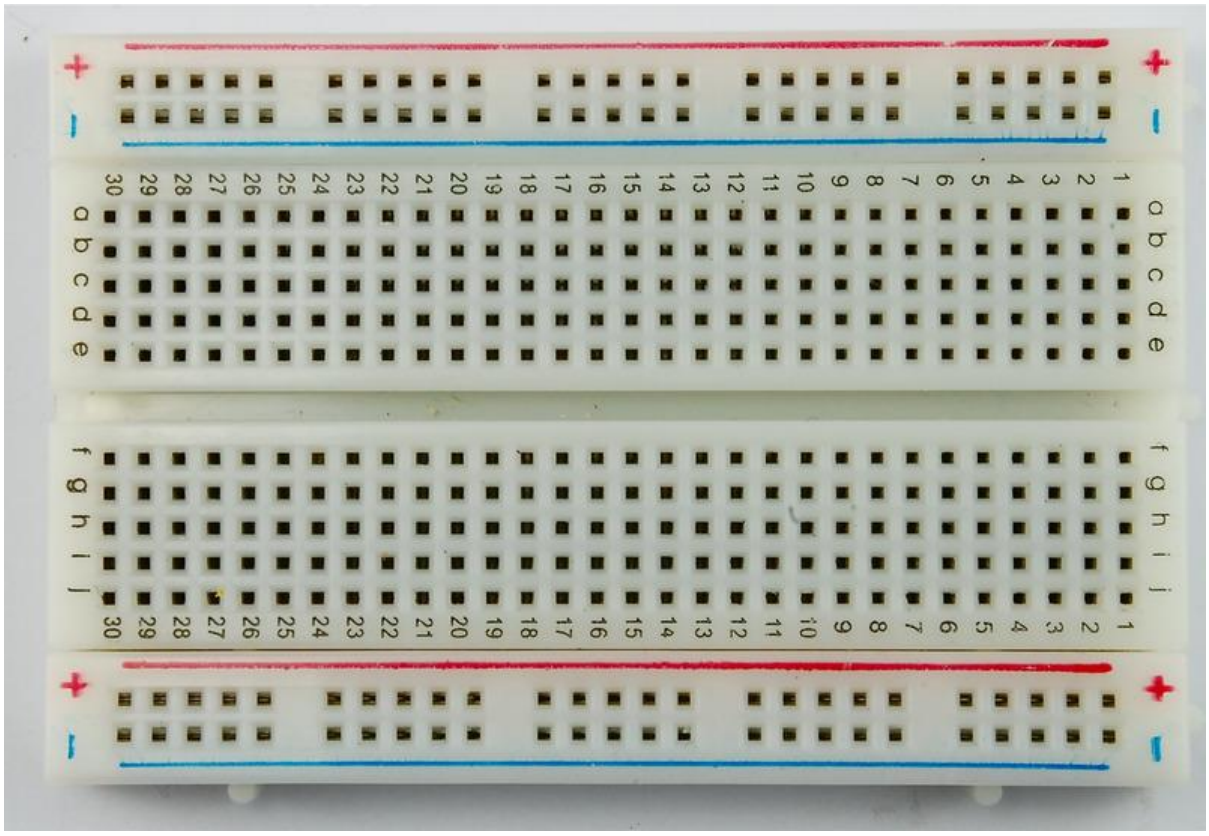
(3x4) Zahlenfeld



AK-207

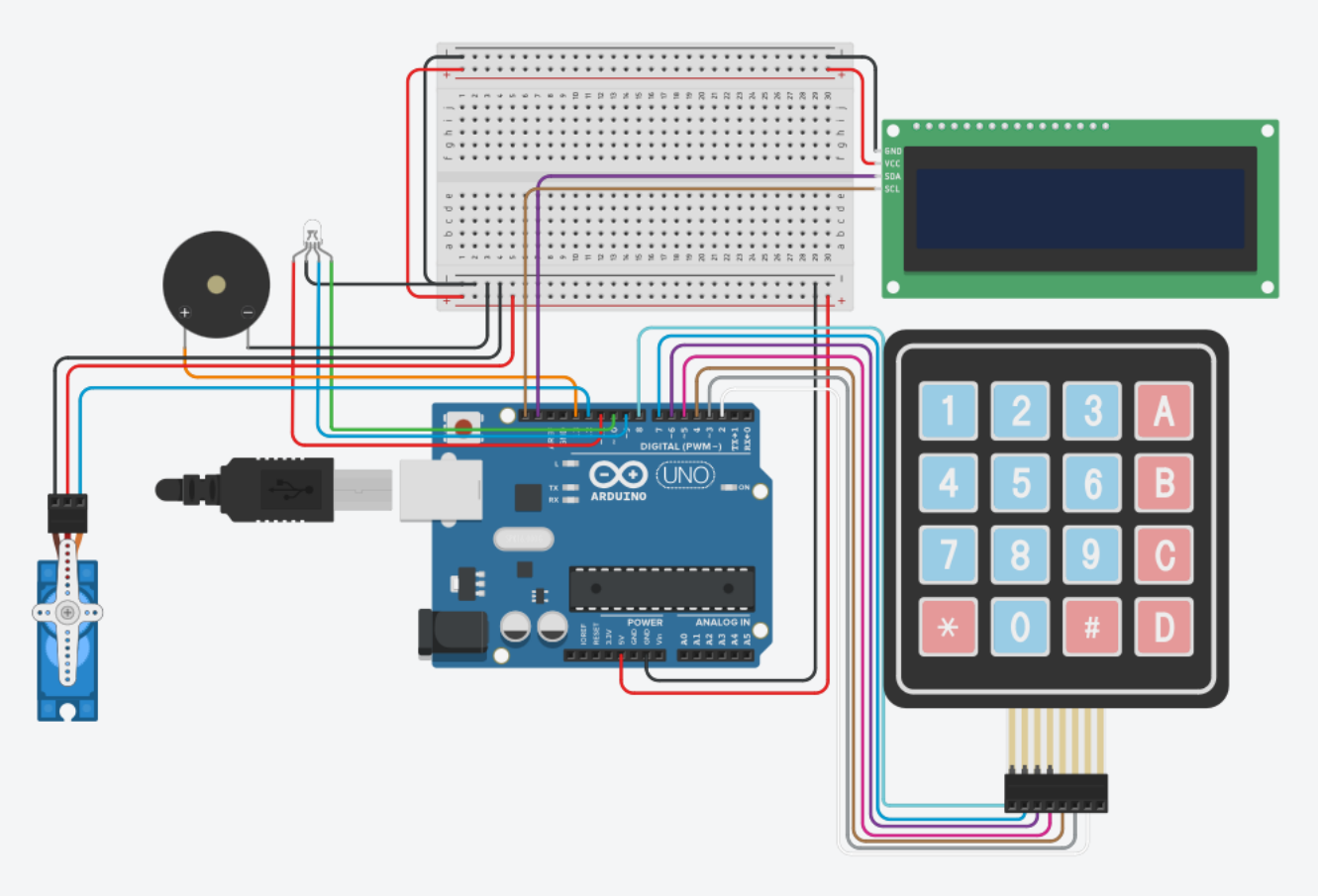
OUTPUT ARRANGEMENT	
OUTPUT PIN NO.	SYMBOL
1	COL 2
2	ROW 1
3	COL 1
4	ROW 4
5	COL 3
6	ROW 3
7	ROW 2

Breadboard



F-to-M Kabel, M-to-M Kabel, 7-9 M-to-M Headers

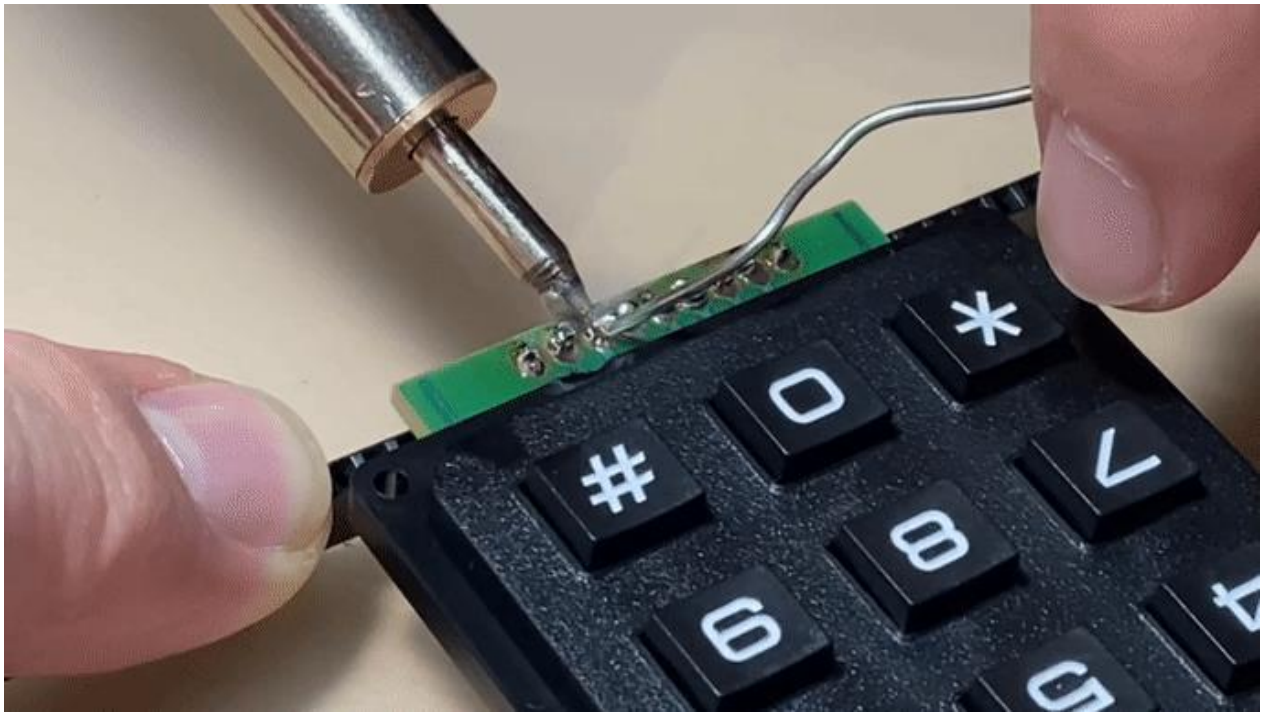
3. Schaltplan



Erstellt auf: tinkercad.com

4. Vorbereitung

Mit den von mir verwendeten Komponenten kann man noch nicht so einfach loslegen, denn das Zahlenfeld muss noch entsprechend gelötet werden, dafür habe ich die 9er Male-to-Male Headers verwendet. (Lötkolben und Lötzinn werden logischerweise benötigt)



Nun kann man alle Komponenten entsprechend dem Schaltplan anschließen.

Es wird Zeit den Code zu schreiben.

5. Code

Includes/PIN-Zuweisungen/Objekte erstellen

```
#include <Key.h>
#include <Keypad.h>
#include <LiquidCrystal_I2C.h>
#include <Servo.h>

//Servo Einstellungen ZU/AUF
#define ZU 0
#define AUF 90

//RGB PIN Zuweisung
const byte red = 11;
const byte green = 10;
const byte blue = 9;

//Buzzer Pin Zuweisung
const byte buzzer = 13;

//Servo Objekt
Servo schloss;

//LCD Objekt
LiquidCrystal_I2C lcd(0x27,20,4);

//Reihen-/Zeilenanzahl vom Zahlenfeld
const byte row = 4;
const byte col = 3;

//Tasten auf Zahlenfeld definieren
char hexaKeys[row] [col] = {
  {'1', '2', '3'},
  {'4', '5', '6'},
  {'7', '8', '9'},
  {'*', '0', '#'}
};

//Zahlenfeld Pins entsprechend nach Datenblatt
byte rowPins[row] = {7, 2, 3, 5};
byte colPins[col] = {6, 8, 4};
```

Variablen deklarieren/setup()

```
40  //Zahlenfeld Objekt
41  Keypad zahlenfeld = Keypad(makeKeymap (hexaKeys), rowPins, colPins, row, col);
42
43  //Passwort standartmäßig auf 1234
44  String passwort = "1234";
45
46  //String für Passwortänderung
47  String passwortneu;
48
49  //String um Eingabe auf dem LCD anzuzeigen
50  String anzeige;
51
52  //Eingegebene Zahl
53  char zahl;
54
55  //Status geschlossen wird standardmäßig angenommen, insgesamt 3: geschlossen,offen,pwd(n
56  String status = "geschlossen";
57
58
59  void setup() {
60
61    //RGB einstellen
62    pinMode(red, OUTPUT);
63    pinMode(green, OUTPUT);
64    pinMode(blue, OUTPUT);
65
66    //Buzzer einstellen
67    pinMode(buzzer, OUTPUT);
68
69    //Servo Pin Zuweisung
70    schloss.attach(12);
71
72    //LCD einstellen
73    lcd.init();
74    lcd.backlight();
75
76    //Schloss standardmäßig zu
77    schloss.write(ZU);
78
79  }
```

loop()

```
81
82  void loop() {
83
84
85    //geschlossener Zustand
86    if (status == "geschlossen") {
87
88      eingabeZahlenfeld(); //Methodenaufruf
89
90      //Display bearbeiten
91      lcd.setCursor(0,0);
92      lcd.print("Geschlossen");
93      lcd.setCursor(0,1);
94      lcd.print("Code : " + anzeige); //Eingabe wird hier angezeigt
95
96      //Enter drücken
97      if (zahl == '#') {
98        if (anzeige == passwort) {
99          korrekterCode(); //Methodenaufruf
100       } else {
101         falscherCode(); //Methodenaufruf
102       }
103     }
104     //Eingabe löschen
105     if (zahl == '*'){
106       anzeige = "";
107       lcd.clear();
108     }
109     delay(100); //Schleife alle 100ms
110
111   }
```

```
113
114   //geöffneter Status
115   if (status == "offen") {
116
117     eingabeZahlenfeld(); //Methodenaufruf
118
119     //Display bearbeiten
120     lcd.setCursor(0,0);
121     lcd.print("Offen");
122     lcd.setCursor(0,1);
123     lcd.print("Verriegeln: #");
124
125     //neues Passwort
126     if (zahl == '*') {
127       lcd.clear();
128       anzeige = "";
129       status = "pwd"; //Übergang zum Status pwd
130     }
131     //Bei Enter drücken schließen
132     if (zahl == '#') {
133       verriegeln();
134     }
135     delay(100); //Schleife alle 100ms
136
137   }
138
```

```
140   //Passwort ändern Status
141   if (status == "pwd") {
142
143     eingabeZahlenfeld();//Methodenaufruf
144
145     //Display bearbeiten
146     lcd.setCursor(0,0);
147     lcd.print("Neues Passwort");
148     lcd.setCursor(0,1);
149     lcd.print(anzeige); //Neues Passwort anzeigen
150
151     //Neues Passwort speichern
152     if (zahl == '#') {
153       lcd.clear();
154       passwort = anzeige; //Passwort wird übernommen
155       anzeige = "";
156       status = "offen"; //Übergang zum Status offen
157     }
158     //Eingabe löschen
159     if (zahl == '*') {
160       lcd.clear();
161       anzeige = "";
162     }
163     delay(100); //Schleife alle 100ms
164
165   }
166
167 }
```

Methoden-Definitionen

```
169  //Methoden Definitionen
170
171  //Zahleneingabe
172  void eingabeZahlenfeld() {
173    zahl = zahlenfeld.getKey();
174    if (zahl != '#') { // Zeichen # soll logischerweise nicht mehr dem String hinzugefügt werden
175      anzeige += String(zahl); //Zahl wird dem String anzeige hinzugefügt
176    }
177  }
178
179  //Bei Falscher Eingabe im geschlossenen Zustand
180  void falscherCode() {
181
182    //Display bearbeiten
183    lcd.clear();
184    lcd.setCursor(0,0);
185    lcd.print("Falscher Code");
186
187    digitalWrite(red, HIGH); //Rotes Licht
188    tone(buzzer, 120); //Tiefen Ton abspielen
189    delay(1500); //Licht,Ton und Anzeige nach 1,5s ausschalten
190    digitalWrite(red, LOW);
191    noTone(buzzer);
192    anzeige = ""; //Eingabe wird zurückgesetzt
193    lcd.clear();
194  }
195
196  //Bei korrekter Eingabe im geschlossenen Zustand
197  void korrekterCode() {
198    status = "offen"; //Übergang zum Status offen
199
200    //Display bearbeiten
201    lcd.clear();
202    lcd.setCursor(0,0);
203    lcd.print("Richtiger Code");
204
205    digitalWrite(green, HIGH); //Grünes Licht
206    schloss.write(AUF); //Schloss öffnen
207
208    //Melodie abspielen
209    tone(buzzer, 523,140);
210    delay(150);
211    tone(buzzer, 659,140);
212    delay(150);
213    tone(buzzer, 784,140);
214    delay(150);
215    tone(buzzer, 1046,140);
216    delay(2000); //Licht und Anzeige nach 2s ausschalten
217    digitalWrite(green, LOW);
218    anzeige = "";
219    lcd.clear();
220  }
221
```

```
221
222 //Schloss verriegeln bei geöffnetem Status
223 void verriegeln() {
224   status = "geschlossen"; //Übergang zum Zustand geschlossen
225   //Display bearbeiten
226   lcd.clear();
227   lcd.setCursor(0,0);
228   lcd.print("Verriegelt");
229
230   schloss.write(ZU); //Schloss verriegeln
231   tone(buzzer, 400); //Ton abspielen
232   digitalWrite(red, HIGH); //Rotes Licht
233   delay(150); //Licht nach 150ms ausschalten
234   tone(buzzer, 700,150);
235   digitalWrite(red, LOW);
236   delay(2000); //Anzeige nach 2s löschen
237   lcd.clear();
238 }
239
240
```

Der Code sollte durch die Kommentation selbsterklärend sein, trotzdem möchte ich erklären, was in welchem Safe-Zustand passiert:

1. Zuerst wird der geschlossene Zustand eingenommen:
    a. Standard-Passwort: 1234
    b. Servo ZU(0°)
    c. LCD: 1.Zeile: “Geschlossen” / 2.Zeile: “Code : <Tasteneingabe>”
    d. Mit der Taste * setzt man die Tasteneingabe zurück
    e. Mit der Taste # bestätigt man den eingegebenen Code
    f. Bei korrekter Eingabe:
        i. LCD: 1.Zeile: “Richtiger Code” / 2.Zeile: *leer*
        ii. Servo AUF(90°)
        iii. Grünes Licht blinken
        iv. Melodie abspielen
        v. Übergang zum Status offen
    g. Bei falscher Eingabe:
        i. LCD: 1.Zeile: “Falscher Code” / 2.Zeile: *leer*
        ii. Rotes Licht blinken
        iii. Tiefen Ton abspielen

2. Offener Status:
    a. LCD: 1.Zeile: “Offen” / 2.Zeile: “Verriegeln: #”
    b. Bei der Taste * ändert man sein Passwort -> Zustandübergang pwd
    c. Bei der Taste # schließt man den Safe:
        i. LCD: 1.Zeile: “Verriegelt” / 2.Zeile: *leer*
        ii. Servo ZU
        iii. Rotes Licht blinken
        iv. 2 Töne abspielen
        v. Übergang zum Status geschlossen

3. Passwort ändern Status:
    a. LCD: 1.Zeile: “Neues Passwort” / 2.Zeile: “<Tasteneingabe>”
    b. Mit der Taste * setzt man die Tasteneingabe zurück
    c. Mit der Taste # bestätigt man sein neues Passwort
    d. Übergang zum Status offen

6. Gehäuse

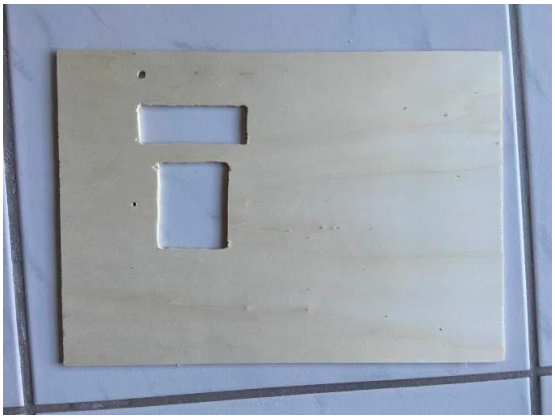
Das Gehäuse kann individuell gestaltet und gebaut werden, mein Gehäuse baut man folgendermaßen:

Zuerst 2 DIN-A4 Holzplatten zu einem Quadrat sägen (21cm * 21cm) und bei den Quadraten einen Holzrahmen bauen, indem man 4 Holzbalken (2x 21cm und 2x 17cm) an die Ränder festschraubt:

Diese 2 Quadrate sollen die Seiten vom Safe darstellen, an eine muss man noch 2 Scharniere befestigen:

Nun schraubt man 2 DIN-A4 Holzplatten jeweils unten und oben an den 2 Seiten an:

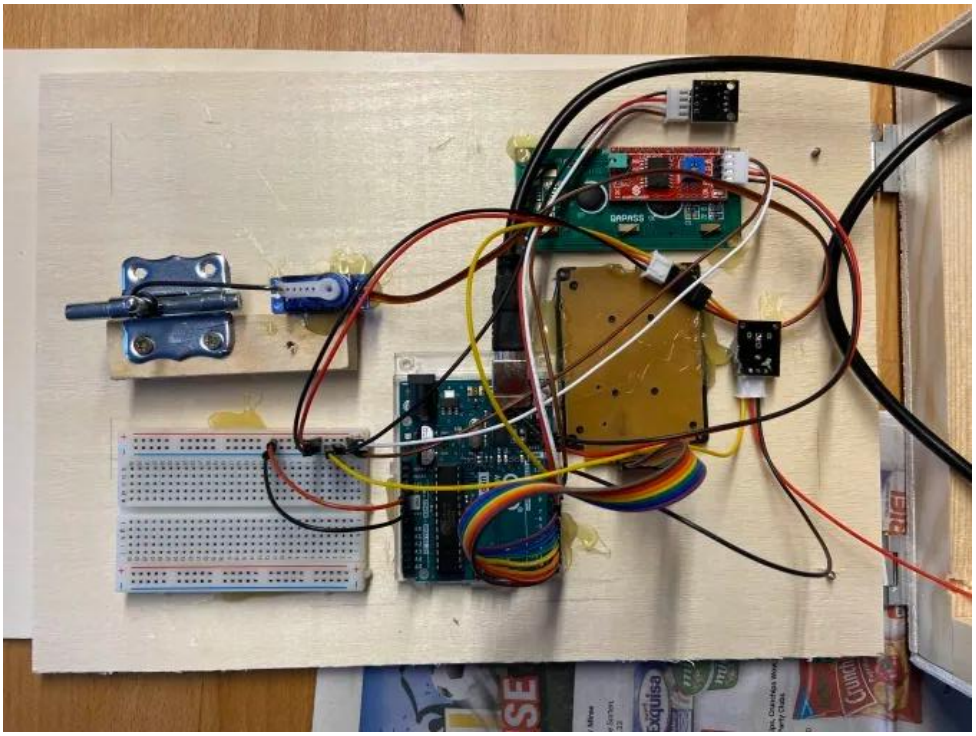
Als nächstes muss man in einer DIN-A4 Holzplatte die Komponentenplätze reinschneiden (Zahlenfeld, Display, RGB und Buzzer):



An diese Holzplatte das Schiebeschloss mit einem kleinen Holzbalken befestigen und die Holzplatte anschließend an die Scharniere festschrauben:

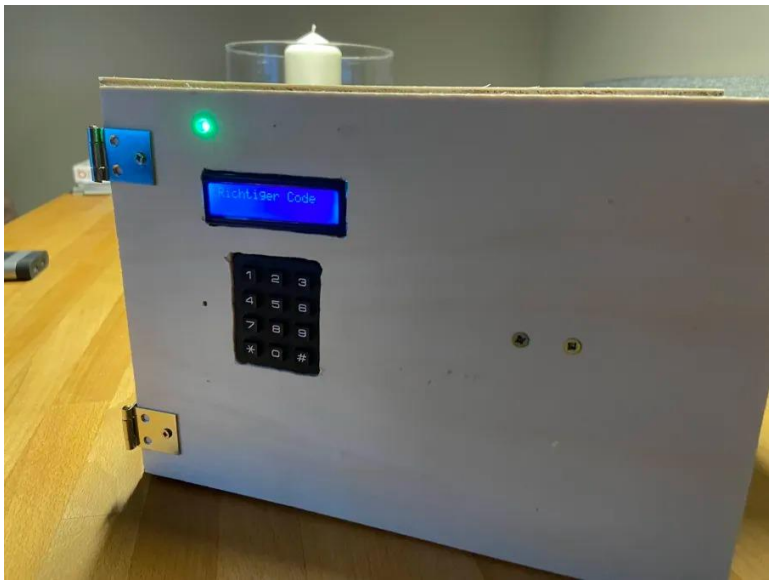
Ein kleines Holzplättchen aus den Resten schnitzen damit das Schloss folgendermaßen aussieht:

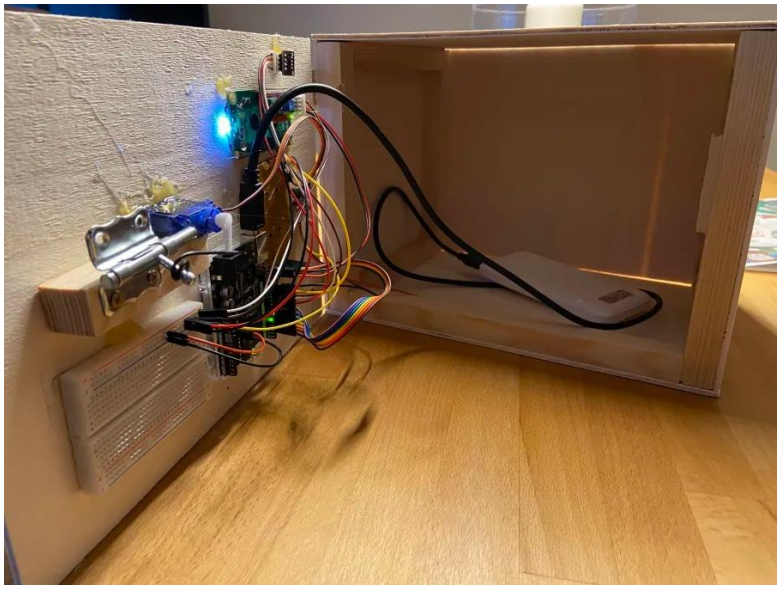
Mit einer Heißklebepistole kann man nun alle Komponenten an der Tür befestigen:



Zum Schluss nur noch die hintere Seite festschrauben und zB einen Akku an den Arduino anschließen:

7. Safe Vorstellung


Richtiger Code
1 2 3
4 5 6
7 8 9
* 0 #




Offen
Verriegeln: #
1 2 3
4 5 6
7 8 9
* 0 #