# http://knork.metropolia.fi/website/wp-content/uploads/2014/05/Tues-logo.png

# Технологично училище

# „Електронни системи“ към ТУ-София

Компютърни Архитектури

# Проект:

# Ключалка с тайно почукване

## Изготвили:

## 1. Атанас Чомполов №2, 11 Б

## 2. Борислав Михайлов №5, 11 Б

## 3. Мартин Ангелов №19, 11 Б

### ­­Ръководител: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Дата: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (Росен Витанов)

# 

# Съдържание на проекта

## Информация за проекта...............................................................3

## Инструменти и материали...........................................................4

## Програмиране на Ардуино..........................................................6

## Схема на проекта..........................................................................11

## Свързване на проекта..................................................................12

## Приготвяне на кутията................................................................13

## Приготвяне на въртящата скоба................................................15

## Запояване на схемата..................................................................16

## Сглобяване на кутията.................................................................19

## Прикачване, тестване и използване..........................................21

# 

# Информация за проекта

Ключалката, която ще се отвори само когато „чуе тайното почукване“, е полезна срещу престъпници. С повече работа по нея може да се постигне и различаване на човека, който се опитва да отвори вратата. Въпреки плюсове си, ключалката може да бъде доста досадна ,ако бързате.  
Тя е програмируема. Натискате бутона за въвеждане на ново почукване и го въвеждате, а ключалката ще се отключи само при чуване на новото почукване. Подразбиране тайното почукване е "[Shave and a Haircut](http://en.wikipedia.org/wiki/Shave_and_a_Haircut)", но можете да програмиране ключалката с каквото си искате почукване до 20 почуквания. Използвайте любимата си песен, морзов код и т.н.

# Инструменти и материали

### ****Инструменти:****

* Бормашина
* Трион, можещ да реже PVC тръби
* Поялник и спойка
* Клещи
* Отвертка
* Изолирбанд
* Клещи за кабел
* Предпазни очила
* Ръкавици

### ****Материали:****

#### Електроника:

* 1 x Arduino Duemilanove
* 1 x 5v Gear мотор
* 1 x Пиезо говорител (30мм)
* 1 x SPST моментен бутон
* 1 x червен светодиод
* 1 х зелен сетодиод
* 1 х NPN Транзистор (P2N2222A)
* 1 х токоизправителен диод (1N4001)
* 1 х 2.2 kΩ резистор
* 1 х 10 kΩ резистор
* 1 х 1 MΩ резистор
* 2 х 560 Ω резистор
* 1 x малка платка за прототипи (5x15 дупки или повече)
* 1 x 9v щипка за батерия и 9v батерия
* Кабел за свързване (20 размер или по-тясна – гъвкав кабел за свързване на електрониката)

#### Кутия:

* 1 x 20" PVC тръба 1/2".
* 3 х 1/2" PVC съединител с десен ъгъл
* 1 x 1/2" PVC 5-пътен съединител
* 2 х 1/2" PVC накрайника
* 3 х 1 1/2" вендузи.
* 1 x 6" на 1/2" широка от 1/64" дебела метална лента
* 1 x 4.5" на 1" широк метален лист, 1/32" дебел
* 2 x 3/32"x3/8" винтове с гайки
* 2 x 1.6M 16мм винтове

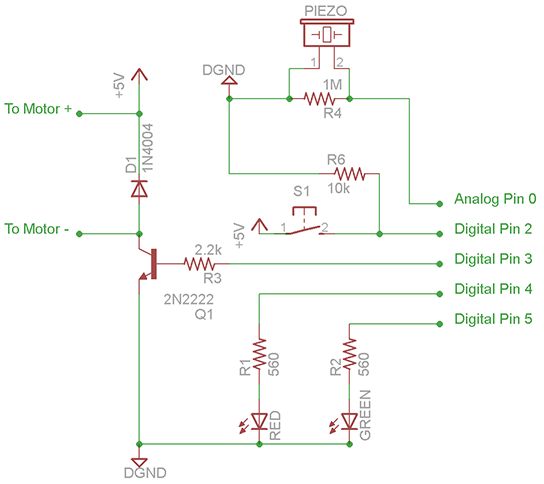
# Програмиране на Ардуино

1. /\*
2. Analog Pin 0: Piezo speaker (connected to ground with 1M pulldown resistor)
3. Digital Pin 2: Switch to enter a new code.  Short this to enter programming mode.
4. Digital Pin 3: DC gear reduction motor attached to the lock. (Or a motor controller or a solenoid or other unlocking mechanisim.)
5. Digital Pin 4: Red LED.
6. Digital Pin 5: Green LED.
7. \*/
9. // Pin definitions
10. **const** **int** knockSensor = 0;         // Piezo sensor on pin 0.
11. **const** **int** programSwitch = 2;       // If this is high we program a new code.
12. **const** **int** lockMotor = 3;           // Gear motor used to turn the lock.
13. **const** **int** redLED = 4;              // Status LED
14. **const** **int** greenLED = 5;            // Status LED
16. // Tuning constants.  Could be made vars and hoooked to potentiometers for soft configuration, etc.
17. **const** **int** threshold = 3;           // Minimum signal from the piezo to register as a knock
18. **const** **int** rejectValue = 25;        // If an individual knock is off by this percentage of a knock we don't unlock..
19. **const** **int** averageRejectValue = 15; // If the average timing of the knocks is off by this percent we don't unlock.
20. **const** **int** knockFadeTime = 150;     // milliseconds we allow a knock to fade before we listen for another one. (Debounce timer.)
21. **const** **int** lockTurnTime = 650;      // milliseconds that we run the motor to get it to go a half turn.
23. **const** **int** maximumKnocks = 20;       // Maximum number of knocks to listen for.
24. **const** **int** knockComplete = 1200;     // Longest time to wait for a knock before we assume that it's finished.

27. // Variables.
28. **int** secretCode[maximumKnocks] = {50, 25, 25, 50, 100, 50, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};  // Initial setup: "Shave and a Hair Cut, two bits."
29. **int** knockReadings[maximumKnocks];   // When someone knocks this array fills with delays between knocks.
30. **int** knockSensorValue = 0;           // Last reading of the knock sensor.
31. **int** programButtonPressed = **false**;   // Flag so we remember the programming button setting at the end of the cycle.
33. **void** setup() {
34. pinMode(lockMotor, OUTPUT);
35. pinMode(redLED, OUTPUT);
36. pinMode(greenLED, OUTPUT);
37. pinMode(programSwitch, INPUT);
39. Serial.begin(9600);                           // Uncomment the Serial.bla lines for debugging.
40. Serial.println("Program start.");             // but feel free to comment them out after it's working right.
42. digitalWrite(greenLED, HIGH);      // Green LED on, everything is go.
43. }
45. **void** loop() {
46. // Listen for any knock at all.
47. knockSensorValue = analogRead(knockSensor);
49. **if** (digitalRead(programSwitch)==HIGH){  // is the program button pressed?
50. programButtonPressed = **true**;          // Yes, so lets save that state
51. digitalWrite(redLED, HIGH);           // and turn on the red light too so we know we're programming.
52. } **else** {
53. programButtonPressed = **false**;
54. digitalWrite(redLED, LOW);
55. }
57. **if** (knockSensorValue >=threshold){
58. listenToSecretKnock();
59. }
60. }
62. // Records the timing of knocks.
63. **void** listenToSecretKnock(){
64. Serial.println("knock starting");
66. **int** i = 0;
67. // First lets reset the listening array.
68. **for** (i=0;i<maximumKnocks;i++){
69. knockReadings[i]=0;
70. }
72. **int** currentKnockNumber=0;                     // Incrementer for the array.
73. **int** startTime=millis();                       // Reference for when this knock started.
74. **int** now=millis();
76. digitalWrite(greenLED, LOW);                  // we blink the LED for a bit as a visual indicator of the knock.
77. **if** (programButtonPressed==**true**){
78. digitalWrite(redLED, LOW);                         // and the red one too if we're programming a new knock.
79. }
80. delay(knockFadeTime);                                 // wait for this peak to fade before we listen to the next one.
81. digitalWrite(greenLED, HIGH);
82. **if** (programButtonPressed==**true**){
83. digitalWrite(redLED, HIGH);
84. }
85. **do** {
86. //listen for the next knock or wait for it to timeout.
87. knockSensorValue = analogRead(knockSensor);
88. **if** (knockSensorValue >=threshold){                   //got another knock...
89. //record the delay time.
90. Serial.println("knock.");
91. now=millis();
92. knockReadings[currentKnockNumber] = now-startTime;
93. currentKnockNumber ++;                             //increment the counter
94. startTime=now;
95. // and reset our timer for the next knock
96. digitalWrite(greenLED, LOW);
97. **if** (programButtonPressed==**true**){
98. digitalWrite(redLED, LOW);                       // and the red one too if we're programming a new knock.
99. }
100. delay(knockFadeTime);                              // again, a little delay to let the knock decay.
101. digitalWrite(greenLED, HIGH);
102. **if** (programButtonPressed==**true**){
103. digitalWrite(redLED, HIGH);
104. }
105. }
107. now=millis();
109. //did we timeout or run out of knocks?
110. } **while** ((now-startTime < knockComplete) && (currentKnockNumber < maximumKnocks));
112. //we've got our knock recorded, lets see if it's valid
113. **if** (programButtonPressed==**false**){             // only if we're not in progrmaing mode.
114. **if** (validateKnock() == **true**){
115. triggerDoorUnlock();
116. } **else** {
117. Serial.println("Secret knock failed.");
118. digitalWrite(greenLED, LOW);          // We didn't unlock, so blink the red LED as visual feedback.
119. **for** (i=0;i<4;i++){
120. digitalWrite(redLED, HIGH);
121. delay(100);
122. digitalWrite(redLED, LOW);
123. delay(100);
124. }
125. digitalWrite(greenLED, HIGH);
126. }
127. } **else** { // if we're in programming mode we still validate the lock, we just don't do anything with the lock
128. validateKnock();
129. // and we blink the green and red alternately to show that program is complete.
130. Serial.println("New lock stored.");
131. digitalWrite(redLED, LOW);
132. digitalWrite(greenLED, HIGH);
133. **for** (i=0;i<3;i++){
134. delay(100);
135. digitalWrite(redLED, HIGH);
136. digitalWrite(greenLED, LOW);
137. delay(100);
138. digitalWrite(redLED, LOW);
139. digitalWrite(greenLED, HIGH);
140. }
141. }
142. }

145. // Runs the motor (or whatever) to unlock the door.
146. **void** triggerDoorUnlock(){
147. Serial.println("Door unlocked!");
148. **int** i=0;
150. // turn the motor on for a bit.
151. digitalWrite(lockMotor, HIGH);
152. digitalWrite(greenLED, HIGH);            // And the green LED too.
154. delay (lockTurnTime);                    // Wait a bit.
156. digitalWrite(lockMotor, LOW);            // Turn the motor off.
158. // Blink the green LED a few times for more visual feedback.
159. **for** (i=0; i < 5; i++){
160. digitalWrite(greenLED, LOW);
161. delay(100);
162. digitalWrite(greenLED, HIGH);
163. delay(100);
164. }
166. }
168. // Sees if our knock matches the secret.
169. // returns true if it's a good knock, false if it's not.
170. // todo: break it into smaller functions for readability.
171. boolean validateKnock(){
172. **int** i=0;
174. // simplest check first: Did we get the right number of knocks?
175. **int** currentKnockCount = 0;
176. **int** secretKnockCount = 0;
177. **int** maxKnockInterval = 0;                     // We use this later to normalize the times.
179. **for** (i=0;i<maximumKnocks;i++){
180. **if** (knockReadings[i] > 0){
181. currentKnockCount++;
182. }
183. **if** (secretCode[i] > 0){                      //todo: precalculate this.
184. secretKnockCount++;
185. }
187. **if** (knockReadings[i] > maxKnockInterval){    // collect normalization data while we're looping.
188. maxKnockInterval = knockReadings[i];
189. }
190. }
192. // If we're recording a new knock, save the info and get out of here.
193. **if** (programButtonPressed==**true**){
194. **for** (i=0;i<maximumKnocks;i++){ // normalize the times
195. secretCode[i]= map(knockReadings[i],0, maxKnockInterval, 0, 100);
196. }
197. // And flash the lights in the recorded pattern to let us know it's been programmed.
198. digitalWrite(greenLED, LOW);
199. digitalWrite(redLED, LOW);
200. delay(1000);
201. digitalWrite(greenLED, HIGH);
202. digitalWrite(redLED, HIGH);
203. delay(50);
204. **for** (i = 0; i < maximumKnocks ; i++){
205. digitalWrite(greenLED, LOW);
206. digitalWrite(redLED, LOW);
207. // only turn it on if there's a delay
208. **if** (secretCode[i] > 0){
209. delay( map(secretCode[i],0, 100, 0, maxKnockInterval)); // Expand the time back out to what it was.  Roughly.
210. digitalWrite(greenLED, HIGH);
211. digitalWrite(redLED, HIGH);
212. }
213. delay(50);
214. }
215. **return** **false**;     // We don't unlock the door when we are recording a new knock.
216. }
218. **if** (currentKnockCount != secretKnockCount){
219. **return** **false**;
220. }
222. /\*  Now we compare the relative intervals of our knocks, not the absolute time between them.
223. (ie: if you do the same pattern slow or fast it should still open the door.)
224. This makes it less picky, which while making it less secure can also make it
225. less of a pain to use if you're tempo is a little slow or fast.
226. \*/
227. **int** totaltimeDifferences=0;
228. **int** timeDiff=0;
229. **for** (i=0;i<maximumKnocks;i++){ // Normalize the times
230. knockReadings[i]= map(knockReadings[i],0, maxKnockInterval, 0, 100);
231. timeDiff = abs(knockReadings[i]-secretCode[i]);
232. **if** (timeDiff > rejectValue){ // Individual value too far out of whack
233. **return** **false**;
234. }
235. totaltimeDifferences += timeDiff;
236. }
237. // It can also fail if the whole thing is too inaccurate.
238. **if** (totaltimeDifferences/secretKnockCount>averageRejectValue){
239. **return** **false**;
240. }
242. **return** **true**;
244. }

# Схема на проекта



# Свързване на проекта

#1: Свързване на Пиезо сензора  
Запояваме чифт 12" (30cm) проводници към Пиезо говорителя. Свързваме между Analog pin 0 и земята. Също така свързваме 1 MΩ резистор между Analog pin 0 и Digital Ground.

#2: Свързване на светодиодите  
Свързваме чевения светодиод към Digital pin 4 ,а зеления към Digital pin 5 със съответващи им 560 Ω резистрори.

#3: Свързване на бутона за програмиране  
Запояваме 8" проводник към бутона. Свързваме една страна на бутона към +5v. Другият щифт на бутона свързваме към Digital pin 2 и към Digital Ground ,с 10 K резистор.

#4: Свързване на мотора   
Запояваме 8" проводник към мотора. Проверяваме диодите наред ли са и pin-овете на транзистора.

# Приготвяне на кутията

#1: Закачане на бутона  
В един от 2та PVC накрайника пробиваме дупка през центъра такава ,че да бъде подходяща за бутона.  
Вкарваме накрайника в една от 4те дупки на 5-пътния съединител(не в дупката ,която сочи надолу).

#2: Закачане на мотора  
В другия PVC накрайник пробиваме дупка ,достатъчно голяма моторът ни да мине през нея.   
Използваме шаблон хартия ,за да поставим дупките за винтове за закрепване. Пробиваме дупките и свързваме мотора към захранването.  
Вкарваме го в “гледащата надолу“ дупка от 5-пътния суединител.  
  
#3a: „Ръцете“  
Изрязваме парче 5" от PVC тръба. Това е „дългата ръка“. Поставяме съединител с десен ъгъл в края. Поставяме другия край в 5-пътния съединител.

Изрязвам две парчета 2" от PVC тръба. Това са „късите ръце“. На половината на дължината пробиваме 1/4" дупка. Поставяме съединители с десен ъгъл в единия край на всяка „ръка“. Вкарваме „късите ръце“ в останалите дупки на 5-пътния съединител.

#3б: Екстра дупки  
С молив начертаваме линия в центъра на горната и долната част на „дългата ръка“. На горната страна ,правим шаблон за две дупки ,едната на 3/4" ,другата на 1 1/2" от 5-пътния съединител. Пробиваме 3/16 (5мм) дупка на всяко от тези места. Това са местата за светодиодите.  
  
#4: „Краката“  
Това са частите ,които се закачат за вратата. Дължината им е зависима от дизайна на ключалката ,от големината на мотора и от въртящата скоба в следващата стъпка.  За „краката“ залепяме вендузите в единия край и прихващаме другите краища към съединителите с десен ъгъл върху краищата на „краката“.

# 

# Приготвяне на въртящата скоба

**Пробиване на дупките**  
Взимаме металното парче(4 1/2" дълго и 1 1/4" високо) и го режем наполовина ,така че да станат 2 парчета(2 1/4").  
Залепяме ги ,маркираме всяка страна ,за да знаем коя е „отвън“ ,маркираме едно от дългите краища като „горе“.   
Измерваме и маркираме центъра от горе до долу. 1/2" всяка страна от централната линия и 1/4" от горните маркирани дупки за дупчене.  
Продупчваме 1/8" дупки на маркираните места. Краят с дупките е страната ,към която се закача моторът.

**Прегъване на метала**  
Измерваме ширината на заключващото резе(тесният път)  и делим на 2. Това разстояние е колко ще прегънем всяко парче метал. Маркираме разстоянието по протежение на всяка лента. Прегъваме едното парче наляво ,другото надясно.

**Довършване и оразмеряване**  
Поставяме винтовете през отворите на горната част ,така че прави с наопаки "Y" и поставете вала на мотора на върха между винтовете.  Завиваме гайките на всяка страна и затягаме ,докато не са здраво прикрепени. Малкото място ,оставено между метала и вала, позволява на мотора да се завърти ,ако има твърде много съпротива вместо да счупи нещо важно.

Завиваме друга гайка върху винта и я затягаме срещу другите. Това ще им помогне да се задържат на едно място.

# Запояване на схемата

#1: Подготвяне на платката за прототипи  
Нашата платка е с 0.10" разстояните ,където всяка дупка има индивидуална медна подложка. Разрязваме платката в размер(5х15 дупки) и след това изпилваме краищата ,така че да пасва перфектно в 1/2" PCV тръбата.  
За напред наричаме страната с медта „задна част“ , страната с компонентите „предна част“.

#2: The ground line.  
По задната част пускаме „ground line“ , за да свържем компонентите по нея. Използваме 9-инча от сърцевината на кабела, който използваме за заземяване и го оголваме 3 см от единия край. Запоената между дупка 1 и дупка 10. След това взимаме другите заземени компоненти и ги поставяме. Другият края на проводника го свързваме с пина за заземяване на Ардуиното.

#3: +5V лента.

Има няколко точки към които свързваме +5V лента. Идеята е същата както при „the ground line“ , но се нуждаем от 1,5 см оголен кабел. Другият край на проводника свързваме с +5V пин на Ардуиното.

#4: LED светодиодите.  
Наместете LED резисторите (560 ohm по подразбиране) на показаното място. Има два варианта как да свържем светодиодите. Може да ги монтираме на отгоре(по-лесният начин) или отдолу, което изглежда е по-добре, но е трудно, защото 1/2" PVC не ни предоставя много място за работа. Ако ги монтираме отгоре трябва да сме сигурни , че сме вдигнали жилата преди да запоим. Вкарваме всички 4 краища на светодиодите с най-близкият край на ‚дългото рамо‘ и с 5-изходният до рамото на борда. Запояваме изхода на катода (най-късият) с всеки от светодиодите. Анодът свързваме с 4-ят дигитален пин (червен) и 5-ят (зелен) на Ардуиното.

#5: Сензорът за почукване.  
Запояваме 1M ohm-овия резистор на говорителя. След като сме сигурни, че е монтиран здраво и в края на пружината, и проводникът е навит няколко пъти около нея. Вкарваме проводника през най-дългия крак до 5-изходния. Запояваме единият край на изходите на всяка страна на 1M-овият резистор. След това запояваме краят на 8‘‘ за не зазимения изход на резистора, който ще свържем към пин 0 (Аналогов).  
  
#6: Бутонът**.**  
Запояваме 10K ohm-овият резистор. Затягаме бутонът в дупката на края на щепсела и след това вкарваме всички проводници през дупката на рамото на борда. Запояваме единият край от ключа към резистора. Другият към +5v проводник. Запояваме една дължина на проводника от резистора и я означаваме ‚Digital 2‘.  
  
#7: Моторът.  
Запояваме диодът, тразисторът и резисторът на място. На свободния край на 2.2k ohm-овият резистор запояваме 8‘ край, който свързваме с пин 3 (Дигитален). Слагаме моторът на място в дупката на бутона на 5-изходния конектор, вкарваме изходите и ги запояваме на място от двете страни на диода. Подсигуряваме, че сме свързали правилно така, че когато го стартираме – отключва.  
  
#8: Pin-овете на Ардуиното  
Свързваме маркираните проводници към техните съответни места на Ардуиното.

# Сглобяване на кутията

Всички компоненти трябва да са поставени на място и цикълът да работи. Последното което трябва да се направи е да се натъпче всичко в тръбата.  
 **#1:** Свързването на проводниците в правилната страна.  
Ако сме вкарали проводниците към Ардуиното трябва да ги изкараме. Проводникът към Analog О и заземеният и +5V трябва да излзнат от рамото на борда. Останалите проводници към Digital 2,3,4 и 5 вкарваме в 5-изходният където е ‚късото рамо‘.  
  
**#2:**Късото рамо.  
Издърпваме проводниците за Digital 2,3,4 и 5 през дупката на средата на тръбата на ‚късото рамо‘.  
  
**#3:**Дългото рамо.  
Дългото рамо има свързани светодиодите и сензорите на него.  
  
**#4 The circuit board.**  
The circuit board should be the first part put in place inside the pipe for the Board Arm. Thread the wires for Ground, +5v and, Analog 0  through the small hole on top of the arm.    
Now make a tight bundle of the wires around the circuit board putting even pressure on it, being careful not to bend, break or spindle the thing.  Gently slide it into the Board Arm.  If you have a lot of extra wire lengths you might want to push it out the far side about half an inch so there's more room for wire on the inside.  
When it's in place, plug this short arm into the 5-way.    
  
**#5** Моторът.  
Моторът е свързан на мястото си - към долната част на 5-изходникът.

**#6:** Бутонът.  
Бутонът също е свързан на мястото си .

**#7:**Краката и вендузите.  
Краката трябва да са вкарани.

The arms should all be plugged in.  Attach 90" turns to the ends of the 3 arms, and plug the legs in to the other end.  
Suction cups should fit snug into the bottom of the legs.  If not, some hot glue will get them into shape.  
(If you're not using suction cups then this is where you use your alternate solution.)  
  
**#8:**Ардуиното и батерията.  
Прикрепваме Ардуино върху горната част на рамката. Прикрепяме батерията по подобен начин, някъде, където може да захрани Ардуиното. Вкарваме проводниците, където етикетите им показват и тестваме отново, че всичко е закачено. Вече сме готови да го закачим на вратата.

# Прикачване, тестване и използване

Прикрепяме към вратата.

Пускаме Ардуиното. Заключваме врата отвътре. Когато светне диода почукваме с *Shave and a Haircut*и отключваме.