

Gymnázium, Praha 6, Arabská 14

Obor programování

# DOKUMENTACE K ROČNÍKOVÉ PRÁCI

Ekaterina Bakhtina,

Filip Dubják,

Antonín Jirásek,

David Přeučil

(3. E)

2019

Gymnázium, Praha 6, Arabská 14

Obor programování

# Bezpečnostní systém pro domácnost „Smart lock“

Ekaterina Bakhtina, Filip Dubják,  
Antonín Jirásek, David Přeučil (3. E)

Předmět: programování  
Vyučující: Mgr. Jan Lána

2019

*Název práce:* Bezpečnostní systém pro domácnost „Smart lock“

*Autoři:* Ekaterina Bakhtina, Filip Dubják, Antonín Jirásek, David Přeučil

*Třída:* 3. E

*Předmět:* Programování

*Vyučující předmětu:* Mgr. Jan Lána

*Třídní učitelka:* Mgr. Jana Urzová

*Abstrakt:* Cílem této ročníkové práce bylo vytvořit bezpečnostní systém pro domácnost (neboli „Smart lock“), pomocí kterého by se dalo ovládat již existující mechanický zámek na dveřích. Součástí práce bylo také navrhnout a vyrobit univerzální mechanismus otáčející klíč v dveřích. Dokumentace obsahuje návod na instalaci systému pro případné uživatele.

*Title:* Home security system "Smart lock"

*Authors:* Ekaterina Bakhtina, Filip Dubják, Antonín Jirásek, David Přeučil

*Class:* 3. E

*Subject:* Programming

*Teacher:* Mgr. Jan Lána

*Class teacher:* Mgr. Jana Urzová

*Abstract:* The goal of this end of term project is to build a home security system (also "Smart lock") which is to be used used to control mechanical locks. A part of the project was to design and create a universal mechanism for turning a key. The documentation includes instructions for installing the system for potential users.

## Prohlášení

Prohlašujeme, že jsme jedinými autory tohoto projektu, všechny citace jsou řádně označené a všechna použitá literatura a další zdroje jsou v práci uvedené. Tímto dle zákona 121/2000 Sb. (tzv. Autorský zákon) ve znění pozdějších předpisů uděluji bezúplatně škole Gymnázium, Praha 6, Arabská 14 oprávnění k výkonu práva na rozmnožování díla (§ 13) a práva na sdělování díla veřejnosti (§ 18) na dobu časově neomezenou a bez omezení územního rozsahu.

Ekaterina Bakhtina

Filip Dubják

Antonín Jirásek

V Praze dne

David Přeučil

# Obsah

<b>1.</b>	<b>Úvod .....</b>	<b>6</b>
<b>1.1.</b>	<b>O vzniku projektu .....</b>	<b>6</b>
1.1.	Zadání .....	6
<b>2.</b>	<b>Vývoj projektu .....</b>	<b>7</b>
1.1.	Použité technologie .....	7
1.1.	Uchycení mechanismu .....	7
1.1.	Obvod .....	9
1.1.	Software .....	10
1.1.	Grafické uživatelské rozhraní .....	10
1.1.	Mobilní aplikace .....	11
<b>3.</b>	<b>Návody k použití .....</b>	<b>12</b>
1.1.	Instalace nástavby ke dveřím .....	12
1.1.	Ovládání zámku .....	12
1.1.	Mobilní aplikace .....	12
<b>4.</b>	<b>Problémy při vývoji .....</b>	<b>13</b>
<b>5.</b>	<b>Závěr .....</b>	<b>13</b>
<b>6.</b>	<b>Zdroje .....</b>	<b>14</b>
1.1.	Odkazy .....	14
1.1.	Obrázky .....	14

# 1. Úvod

## 1.1. O vzniku projektu

Dnešní doba nabízí řadu způsobů, jak zabezpečit svoje bydliště a zároveň navrhuje alternativy mechanismům, které se neustále používali během desetiletí. Jednou z takových technologií je smart lock (česky „chytrý zámek“) – elektronická varianta zámku, která umožňuje ovládání dveří většinou pomocí chytrého telefonu, nebo také prostřednictvím SMS, Wi-Fi, otisku prstu, systému rozpoznávání tváře a jiných technologií. (CHANG, 2013)

Řada různých firem nabízí svoje varianty smart locků. Nevýhodou ale je, že moderní chytré zámky jsou velice často finančně nákladné, kromě toho většinou vyžadují úpravu stávajícího zámku, popřípadě dokonce i jeho výměnu. Tohle nás inspirovalo vytvořit systém, který by měl přechod na chytré zámky ulehčit. Velmi podstatnou součástí naše práce je univerzálním způsob chycení, který umožňuje připojit celou konstrukci k většině typů dveří a zároveň je finančně dostupný. Náš zámek je možné jednoduše ovládat pomocí webového rozhraní, poskytujícího přístup k ovládání zámku pouze oprávněným osobám. Výhodou rozhraní je možnost zámek odemknout na dálku (např. otevřít dveře pro kurýra z jiné místnosti v budově).

## 1.2. Zadání

Cílem dané rovníkové práce bylo vyrobit zámek, který by se dalo ovládat dálkově pomocí aplikace (Android, webové) a sdílet přístup s dalšími uživateli. Aplikace by uživatelům umožňovala ovládání přístroje a poskytovala informace o aktuálním stavu zařízení. Zámek by bylo přitom stále možné otevřít klíčem, jelikož vyrobený doplněk by byl ve formě visacího zámku anebo doplňku k již existujícímu zámku. Vývoj doplňku je součástí projektu, bude se skládat z motoru a jednočipového počítače, např. Arduino nebo Raspberry Pi).

## 2. Vývoj projektu

### 2.1. Použité technologie

HARDWARE:

**Raspberry Pi Zero:**

*Malá a cenově dostupnější verze jednodeskového počítače.*

**DC motor s převody**

SOFTWARE:

**Django:**

*Vysokoúrovňový webový framework v jazyce Python.*

**OS Raspbian:**

*Oficiální podporovaný operační systém pro počítače Raspberry Pi.*

**SelfCAD, TinkerCAD:**

*Online aplikace pro 3D modelování.*

**Android Studio**

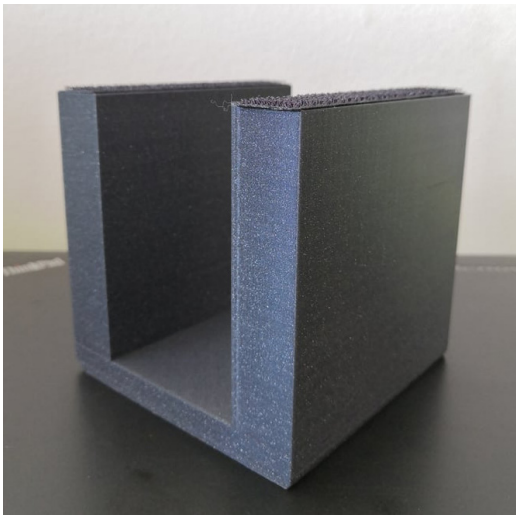
*Vývojové prostředí pro Android aplikace.*

### 2.2. Uchycení mechanismu

Problematika uchycení byla jedním z hlavních témat práce. Uchycení bylo nutné vytvořit pevné a zároveň využít stávající vybavení dveří.

Nástavba se skládá ze dvou dílů a násadky. První díl (*obr. 1*) je pomocí suchého zipu připevněn na dveře a také na počítač Raspberry. Na druhý díl (*obr. 2*) je upevněn obvod a motor. Násadka je připevněna z jedné strany na motor a z druhé je do ni zasunut klíč.

Celá nástavba byla speciálně navržena pro tento projekt, není však vytvořena na specifický typ dveří či zámku, aby ji bylo možné použít na maximální počet různých typů dveří. Modularita nástavby umožňuje upravit vzdálenost motoru od klíčové dírky, proto je možné ji přemístit. Instalace nástavby nevyžaduje žádné zvláštní úpravy dveří, proto nutné není při výměně dveří či zámku řešit jejich úpravy.



Obr. 1



Obr. 2

Všechny díly jsou vytištěné pomocí technologie 3D tisku, což napomáhá jejich pevnosti a snížení finančních nákladů. První díl je připevněn ke dveřím suchým zipem se dvěma styčnými plochami velikosti 20 cm, které zajišťují dostatečnou oporu motoru. Motor je do prvního dílu připevněn suchým zipem. Suchý zip umožňuje motoru pohyb, a přitom obstarává jeho stabilitu. Na motor je nasazena násadka (obr. 3), jejíž rozměry je možné upravit díky využití 3D tisku na velikosti různých klíčů (případně jiných motorů).

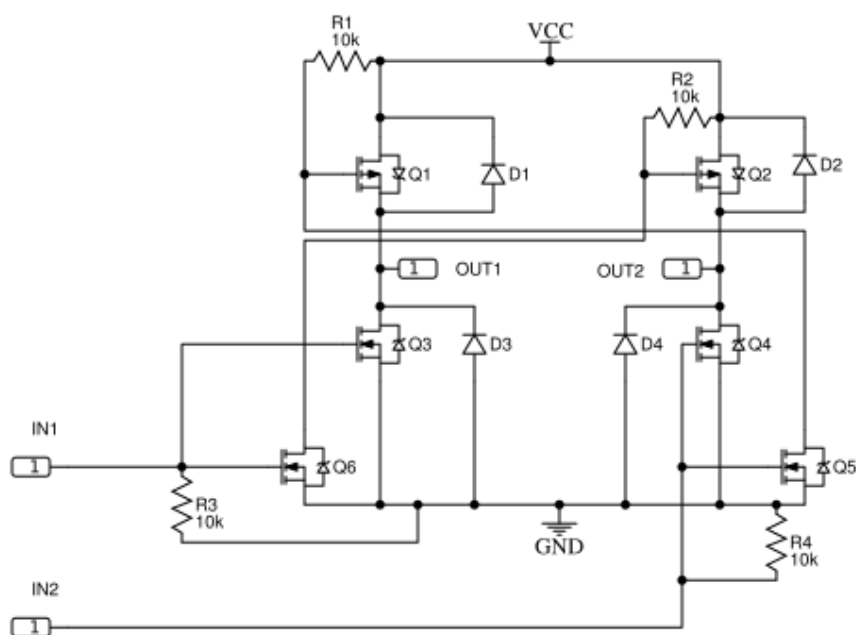


Obr. 3



## 2.3. Obvod

Obvod (*obr. 4*) se jmenuje H-Můstek (H-Bridge). Název H-Můstek vznikl z rozložení MOSFETů do písmene H. H-Můstek slouží k ovládání DC motorů. Skládá se ze 4 MOSFETů: 2 typu  $p$  a 2 typu  $n$ . Můstek můžeme rozdělit na 2 půl-můstky. Každý půl-můstek se skládá z mosfetu typu  $p$  a  $n$  a má 3 vývody. Prostřední vývod je určený pro napájení jednoho pólu motoru, ostatní 2 vývody slouží pro napájení. Směr otáčení motoru se ovlivňuje spuštěním  $p$ -mosfetu v jednom půl-můstku a  $n$ -mosfetu v druhém. V obvodu jsou navíc 4 diody, které zabraňují napěťovým špičkám, vzniklým změnou magnetického pole na cívkách motoru (při zastavení nebo při změně směru otáčení). U MOSFETů typu  $p$  se nachází pull-up rezistor, který zaručuje vypnutí MOSFETů, když není přiveden ovládací signál. U MOSFETů typu  $n$  je pull-down resistor ze stejného důvodu. Obvod obsahuje navíc 2 tranzistory typu  $n$ , které slouží jako invertory pro mosfety typu  $p$ , protože se otevírají opačnou polaritou než mosfety typu  $n$ .



Obr. 4

Pro připojení počítače k motoru jsme využili obvod připojený ke dvěma output pinům na Raspberry (*obr. 4* – *IN1* a *IN2*) a k síti pomocí USB adaptéru. Obvod je poté připojen k motoru (*obr. 4* – *OUT1* a *OUT2*), kde je směr proudu kontrolován podle napětí na *IN1* a *IN2*.

## 2.4. Software

Při vývoji webové aplikace byl použit autentizační systém, který je již nainstalovaný v Django: `django.contrib.auth.urls`. Aplikace `auth` řídí účty uživatelů, s její pomocí se dá jednoduše zařídit přihlášení a odhlášení, jelikož není třeba vytvářet vlastní šablony. Pro registraci byl implementován formulář `UserCreationForm`, který bylo třeba rozšířit za účelem možnosti přidání e-mail adresy. (ANON)

Ovládání motoru pomocí Raspberry Pi přes GPIO (general-purpose input/output) piny je možné díky nainstalovanému balíčku `Rpi.GPIO`. Otáčení se uskutečňuje prostřednictvím následujících příkazů:

```
GPIO.setmode(GPIO.BOARD)
GPIO.setup(37, GPIO.OUT)
GPIO.output(7, False)
GPIO.output(37, True)
time.sleep(10.5)
GPIO.cleanup()
```

`GPIO.BOARD` označuje číslování pinů podle čísel vytištěných na desce. `GPIO.setup(#, GPIO.OUT)` nastaví vybraný pin, `GPIO.output(#, True)` ho zapne, `time.sleep` určuje čas, po kterém se splní další řádek (v našem případě je to doba otačení). (ALEX. 2013)

Velice důležité je zabránit tomu, aby souběžně byly nastaveny oba piny 37 a 7 na 3 V. Kdyby se to stalo, mohlo by to spálit obvod.

## 2.5. Grafické uživatelské rozhraní

Grafické uživatelské rozhraní bylo vytvořeno v jazyce CSS (Cascading Style Sheets), pomocí kterého byly upraveny HTML šablony. Webová aplikace je udělána v moderním minimalistickém flat designu a má jednoduché barevné schema. Na každé stránce se nachází malé okno s obsahem dané stránky, pozadí představuje gradient postupně měnící barvy. Část souboru `style.css`:

```

body {
background: linear-gradient(-45deg, #843EFF,
#365EEB, #38D7FF, #27E89A, #37FF30);
background-size: 400% 400%;
-webkit-animation: Gradient 15s ease infinite;
-moz-animation: Gradient 15s ease infinite;
animation: Gradient 15s ease infinite;
padding: 10px;
}

@keyframes Gradient {
0% {background-position: 0% 50%}
50% {background-position: 100% 50%}
100% {background-position: 0% 50%}
}

```

Tag `@keyframes` se používá pro nastavení cyklu animace a přechodů v CSS. Pomocí tagů `-webkit-animation` a `-moz-animation` se určují nastavení pro různé prohlížeče (v daném případě pro Chrome, Safari, Opera a Firefox). Tag `infinite` zaručuje nekonečné opakování cyklu.

## 2.6. Mobilní aplikace

Při psaní naší mobilní aplikace bylo použito již vytvořené webové rozhraní, jelikož nebylo třeba se zabývat novou grafikou a její kompatibilitou s Androidem. Pro programování této aplikace posloužil program Android Studio. Úkolem tedy bylo konvertovat naši webovou stránku do Android aplikace. Nejprve jsme si v layoutu upravili `activity_main.xml` tak, aby se tam nacházelo okno, ve kterém by se zobrazoval web a textové pole nad ním (vyhledávač IP adresy) společně s tlačítkem `refresh`. V části `main` jsme si naicializovali tlačítko tak, po kliknutí na něj se vzal text z textového pole a znovu se načetla vyžadovaná URL adresa.

### **3. Návod k použití**

#### **3.1. Instalace nástavby ke dveřím**

1. Přiložené kusy suchého zipu přilepte na dveře, každý z jedné strany klíčové dírky ve vzdálenosti 5 cm od ní.
2. Držák připevníme pomocí suchého zipu ke dveřím.
3. Klíč zasuneme do klíčové dírky a motor připevníme do držáku.
4. Raspberry a motor připojíme ke zdroji.

#### **3.2. Ovládání zámku**

1. Po registraci se uživatel přihlásí pomocí vytvořených přihlašovacích údajů.
2. Uživatel je přesměrován na webovou stránku, ze které je možné zámek odepknout či zamknout, dostane tam upozornění o aktuálním stavu zámku.

#### **3.3. Mobilní aplikace**

1. K použití naší mobilní aplikace je k zapotřebí mít na svém mobilním telefonu operační systém Android.
2. Soubor `app-release.apk` si můžete stáhnout z našeho GitHub repozitáře, který najdete na webovém odkazu: `github.com/gyarab/chytry-lock`.
3. Po stažení souboru musíte veškerou vaši důvěru vložit do našeho neautorizovaného instalačního programu, tzn. povolte všechno co vám bylo systémem nedoporučeno povolovat a pokračujte.
4. Po instalaci můžete na ploše najít ikonku SmarLock 2.3, klikněte na ní a můžete pokračovat (aplikace funguje stejně jako webová stránka).
5. Ve vyhledávacím poli zadávejte IP adresu serveru daného zámku ve formátu `xxx.xxx.xxx.xxx`, port je automaticky nastavený na 8000.

## 4. Problémy při vývoji

Během vývoje nás potkalo nemalé množství problémů. Většinou se týkaly hardwaru, jelikož Raspberry Pi Zero nemá dostatečný výkon na hostování serveru zároveň s grafickým uživatelským rozhraním operačního systému. Nejspíše se důsledkem těchto nedostatků rozbila naše Raspberry a byli jsme nuceni zakoupit novou. Po nyní nabytých zkušenostech bychom se nejspíš přiklonili k výkonějšímu zařízení.

Tato hardwarová chyba (počítač se automaticky vypínal během načítání dat) také dvakrát způsobila korupci naší SD karty s operačním systémem, kterou jsme poprvé museli přeformátovat a podruhé vyměnit za novou.

Mezi softwarové překážky se řadí například spouštění motoru pomocí skriptu, jelikož skriptů napsaný v Pythonu je složitější spustit přes HTML stránku, než např. napsaný v JavaScriptu.

## 5. Závěr

Vytvořili jsme chytrý zámek, který je schopen otevřít a zavřít dveře. Zámek lze ovládat přes aplikaci nebo web, kde po přihlášení je také možné sledovat stav zámku. Při zhotovování ročníkového projektu jsme narazili na pár zásadních překážek, mezi které patří i nefunkční Raspberry. V další verzi projektu bychom nejspíše zvolili krokový motor, u kterého se lépe kontroluje, zda se zámek opravdu kompletně otočil. Další evoluce projektu by mohli zahrnout tabulku s historií odemykání a zamykání zámku.

## 6. Zdroje

### 6.1. Odkazy

**ALEX.** 2013. RPi.GPIO basics 5 – Setting up and using outputs with RPi.GPIO. RasPi.TV. 13. July. <https://raspi.tv/2013/rpi-gpio-basics-5-setting-up-and-using-outputs-with-rpi-gpio> (accessed: 6. May 2019).

**ANON.** Django. [HTtps://docs.djangoproject.com/en/2.2/ref/contrib/auth/](https://docs.djangoproject.com/en/2.2/ref/contrib/auth/) (accessed: 6. May 2019).

**BESTASTTUNG.** 2015. . Stackoverflow. 18. July. <https://stackoverflow.com/questions/26510242/django-how-to-login-user-directly-after-registration-using-generic-createview> (accessed: 6. May 2019).

**CHANG, ALEXANDRA.** 2013. Your Door Is About to Get Clever: 5 Smart Locks Compared. Wired Magazine. 19. June. <https://www.wired.com/2013/06/smart-locks/> (accessed: 6. May 2019).

### 6.2. Obrázky

*Obr. 1: vlastní fotografie.*

*Obr. 2: vlastní fotografie.*

*Obr. 3: vlastní fotografie.*

*Obr. 4: vlastní tvorba.*