VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE

Brno, 2019 Ondřej Kupka

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ
DEPARTMENT OF TELECOMMUNICATIONS

MODEL ŠIFRÁTORU ENIGMA ZE STAVEBNICE LEGO TECHNIC
MODEL OF ENIGMA ENCRYPTION MACHINE FROM LEGO TECHNIC

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — **Ondřej Kupka**
AUTHOR

VEDOUCÍ PRÁCE — **doc. Ing. Jan Hajný, Ph.D.**
SUPERVISOR

BRNO 2019

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Bakalářská práce

bakalářský studijní obor **Informační bezpečnost**
Ústav telekomunikací

Student: Ondřej Kupka ***ID:*** 195158
Ročník: 3 ***Akademický rok:*** 2018/19

NÁZEV TÉMATU:

Model šifrátoru Enigma ze stavebnice Lego Technic

POKYNY PRO VYPRACOVÁNÍ:

Cílem práce je rozšířit stávající model šifrátoru z Lego Technic Mindstorms o moduly pro zadávání zprávy a zobrazování šifrového textu. Předpokládá se využití klávesnice a displeje propojeného přes Bluetooth. Výstupem práce je model šifrátoru umožňující snadné zadávání vlastního textu pro šifrování a přehledné zobrazení výstupů šifrátoru, plná dokumentace a návod k použití modelu.

DOPORUČENÁ LITERATURA:

[01] MENEZES, Alfred, Paul C VAN OORSCHOT a Scott A VANSTONE. Handbook of applied cryptography. Boca Raton: CRC Press, c1997. Discrete mathematics and its applications. ISBN 0-8493-8523-7.

[02] PARK, Eun Jung. Lego Mindstorms EV3: stavíme a programujeme roboty. Brno: Computer Press, 2015. ISBN 978-80-251-4385-8.

Termín zadání: 1.2.2019 ***Termín odevzdání:*** 27.5.2019

Vedoucí práce: doc. Ing. Jan Hajný, Ph.D.
Konzultant:

prof. Ing. Jiří Mišurec, CSc.
předseda oborové rady

ABSTRAKT

Cílem práce je návrh a realizace možnosti propojit model šifrovacího stroje Enigma, ze stavebnice LEGO Technic a LEGO Mindstorm, spolu se zobrazovacím zařízením a rozšířit model o možnost zadat jakýkoliv text pomocí Bluetooth klávesnice. Vstupní text je pak spolu s demonstrací funkcionality šifrovacího stroje zašifrován a výstupní šifrový text je zobrazen na obrazovce zobrazovacího zařízení. První část práce se věnuje počátkům šifrování a vývoji šifrovacího stroje Enigma. Následně je podrobněji zaměřena na model Wehrmacht Enigma, na jehož principu byla vytvořena webová aplikace pro obsluhu modelu. Aplikace je navržena na základě speciálního firmware řídící jednotky Ev3dev a napsána pomocí jazyka Python a Flask. Práce popisuje dílčí části aplikace, jednotlivé funkce a principy. Součástí práce je návod na obsluhu modelu.

KLÍČOVÁ SLOVA

Enigma, LEGO, kryptografie, šifra, model šifrátoru, Python, Flask, EV3dev

ABSTRACT

The aim of the thesis is to design and implement the possibility of interconnecting the Enigma encryption machine model, built from LEGO Technic and LEGO Mindstorm kit, with the display device and extend the model by the possibikty to enter any text using the Bluetooth keyboard. The input text is then encrypted along with the demonstration of the encryption by the model and the output encryption text is then displayed on the display device screen. The first part of the work is devoted to the beginnings of encryption and development of Enigma encryption machine. Subsequently, it is focused in more detail on the Wehrmacht Enigma model, on the basis of which a web application for model operation was created. The application is designed based on the Ev3dev special firmware for the control unit and written using Python and Flask. The work describes partial parts of the application, individual functions and principles. Part of the work is a manual for model operation.

KEYWORDS

Enigma, LEGO, cryptography, cipher, cipher model, Python, Flask, EV3dev

KUPKA, Ondřej. *Model šifrátoru Enigma ze stavebnice Lego Technic*. Brno, Rok, 65 s. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce: doc. Ing. Jan Hajný, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Model šifrátoru Enigma ze stavebnice Lego Technic“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno ....

podpis autora

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu bakalářské práce panu doc. Ing. Janu Hajnému, Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno ....

podpis autora

Tato práce vznikla jako součást klíčové aktivity KA6 - Individuální výuka a zapojení studentů bakalářských a magisterských studijních programů do výzkumu v rámci projektu OP VVV Vytvoření double-degree doktorského studijního programu Elektronika a informační technologie a vytvoření doktorského studijního programu Informační bezpečnost, reg. č. CZ.02.2.69/0.0/0.0/16_018/0002575.

EVROPSKÁ UNIE
Evropské strukturální a investiční fondy
Operační program Výzkum, vývoj a vzdělávání

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY

Projekt je spolufinancován Evropskou unií.

Obsah

Seznam obrázků

Seznam tabulek

Seznam výpisů

Úvod

Tato práce navazuje na práci *Demonstrátor šifrátoru z Lego Technic*, Bc. Jakuba Jančíka [1]. Hlavním předmětem této práce bylo rozšířit původní model demonstrátoru o možnost zadat text pomocí klávesnice a následně zobrazit výstup šifry na zobrazovacím zařízení.

Teoretická část slouží jako úvod do problematiky. V první části je stručně shrnut vývoj šifrování a pohled na kryptografickou situaci v Německu za první světové války.

V druhé části je představen šifrovací stroj Enigma. Kapitola se věnuje historii vývoje jednotlivých verzí stroje. Následně se pak práce zaměřuje zejména na model Wermacht Enigma, který je podrobně popsán. Použité fotografie byly přebrány ze stránky *Technical Details of the Enigma Machine* [2].

Třetí část popisuje původní stav demonstrátoru. Představuje jeho konstrukci a funkcionalitu.

Praktická část se zaměřuje na návrh a realizaci řešení modelu šifrátoru. Věnuje se úpravě původní konstrukce a aplikaci, napsané pro zajištění funkcionality šifrátoru. Podrobněji popisuje jednotlivé součásti aplikace a jejich logiku.

1 Počátky šifrování

1.1 Proč je šifrování potřeba?

Určité informace měly už od nepaměti velmi vysokou hodnotu. Obzvláště cenné pak byly zprávy kolující v období konfliktu či válek, jakožto znalosti, které by mohly výrazně ovlivnit vývoj událostí. Aby se takovéto informace nedostaly do nepovolaných rukou, či k nepříteli, je nutné utajit jejich význam. Tento úmysl v minulosti vedl k vývoji kódů a šifer.

1.2 Stručný vývoj

Kódy a šifry vždy existovaly společně s luštiteli a analytiky, kteří se snaží rozluštit tajné kódy a proniknout tak k utajeným informacím. Jakmile je však technika použitá k ukrytí zprávy odhalena, stává se její použití zbytečným. Na druhé straně se pak tvůrci šifer snaží vymyslet co nejefektivnější metodu. Vzniká tak neustálý konflikt, který podněcuje další vývoj.

Rané počátky přenosu zpráv vyžadovaly prostředníka mezi odesílatelem a příjemcem. Posla, který doručí zprávu. Jeho bezpečnost však nebylo možné zaručit, mohl být zastaven třetí stranou. Ve většině případů bylo nutné zprávu doručit co nejrychleji a pokud možno diskrétně. Tady vzniká myšlenka skrytí zprávy, tedy aby se o ní v případě narušení komunikace nikdo nedozvěděl.

1.2.1 Steganografie

Název vznikl kombinací řeckých výrazů pro schovaný a psát. Jejím smyslem je skrýt zprávu, nebo zatajit průběh komunikace. Příkladem může být text vepsán do kůže hlavy posla, skryt pod novou vrstvou vlasů nebo zpráva vyrytá pod vrstvou vosku psací destičky. Později byly vymyšleny speciální inkousty, které byly za normálních podmínek neviditelné. Inkoustem Italského vědce Giovanni Porta bylo možné psát přes skořápku vařeného vejce na bílek. Jiné inkousty bylo nutné pro zviditelnění nahřát nebo nasvítit speciálním světlem. V neposlední řadě se zprávy daly schovat do všemožných předmětů denní potřeby v podobě dutých prostor a tajných přihrádek [3].

Výše uvedené metody však nejsou příliš bezpečné. Ve chvíli odhalení úkrytu je vyzrazen celý obsah zprávy. Proto bylo nutné přijít s novou, účinnější metodou.

1.2.2 Kryptografie

Smyslem kryptografie je utajení obsahu zprávy. Využívá předem ustanovená pravidla mezi příjemcem a odesílatelem, podle kterých pak mění text zprávy. Tak aby jej, bez předchozí znalosti nebo vyvinutí dostatečného úsilí, nebylo možné přečíst. Vznikají tak šifry, které můžeme rozdělit do dvou kategorií, a to transpoziční a substituční [3].

1.2.3 Transpoziční šifry

Principem transpozičních šifer je záměna pořadí znaků. Samotné znaky vyskytující se v původní zprávě se objevují i v šifrovém textu. Pro dešifrování je nutné znovu aplikovat pravidlo použité pro šifrování v opačném pořadí. Jednoduchým příkladem transpozice je text psaný pozpátku.

Jednou z historicky používaných metod byl tenký kožený proužek namotaný na válci o pevně daném průměru. Na ten se po řádcích napsala zpráva. Následně se proužek odmotal a tvořil tak zdánlivě nesmyslný sled písmen. Pro přečtení zprávy bylo nutné opět daný proužek namotat na kužel o stejném průměru.

Příkladem transpoziční šifry je Sloupcová metoda, neboli Column cipher. Otevřený text se po řádcích napíše do tabulky s pevně daným počtem sloupců. Permutací počtu sloupců vzniká klíč, u příkladu 1.1 tedy „3412“, nebo například „STAR“. V psané podobě odpovídá pořadí znaku v otevřené abecedě danému číslu sloupce.

3	4	1	2
V	I	T	E
J	T	E	N
A	V	U	T

Tab. 1.1: Příklad zápisu do sloupcové šifry

Znaky se postupně po sloupcích sepíší na řádek. Vzniká tak šifrový text:

VJA ITV TEU ENT

U delších textů nemusí být poslední řádek zcela naplněn. V takovýchto případech se dá volit mezi dvěma postupy: s doplňujícími znaky a bez doplňujících znaků. Postupy se pak liší v dešifrování [4].

1.2.4 Substituční šifry

Oproti transpozičním šifrám zůstává pořadí původní zprávy stejné. Dochází zde k záměně původní znakové sady za jinou. První zmínky ohledně substitučních šifer

se nacházejí v Kámasútře, bráhman Vátsjájana. Kniha radí ženám studovat šedesát čtyři umění, mezi nimiž se na čtyřicátém pátém místě nachází umění tajného písma. Popisuje pak metodu náhodného spárování písmen abecedy. Při psaní zprávy se pak písmena nahradí adekvátní dvojicí [3].

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
↕
N R O Z Q T U C A M S X B W F Y G K E D H I L V P J

Monoalfabetická substituční šifra

Tato šifra pracuje pouze s jednou šifrovací abecedou. Jedno písmeno otevřené abecedy se zamění za jiné. Nejznámějším zástupcem je Caesarova šifra. Spočívá v nahrazení znaku otevřené abecedy, znakem o tři pozice dále. Ze znaku „A“ se tak stává znak „D“.

C A E S A R
↓
F D H V D U

Šifrovou abecedu lze vytvořit i pomocí hesla, nebo fráze. Z fráze se odstraní mezery a duplicitní znaky. Nakonec se pak doplní zbytek abecedy v originálním pořadí, vyjma znaků již obsažených v úvodním sledu [3].

Tajné heslo → TAJNEHSLO + BCDFGIKMNPQRSUVWXYZ

Je možno nahradit i celá slova za jednotlivé znaky, či slova jiná. Tato metoda se nazývá kódování. Patří zde například Morseova abeceda.

Polyalfabetická substituční šifra

Šifra využívá dvě a více šifrových abeced. Ty se při šifrování pravidelně střídají. Stejné znaky otevřeného textu se tak nemusí projevit jako stejné znaky v textu šifrovém. Opakovaný znak "A"pak může být zašifrován poprvé jako "T", podruhé jako "E". Počet možností výstupu pro jeden znak se poté odvíjí od počtu použitých abeced.

Známým zástupcem je Vigenèrova šifra. Šifra používá 26 abeced sepsaných pod sebe, tvořic Vigenèrův čtverec 1.2. První řádek odpovídá otevřené abecedě. Další řádky jsou tvořeny posunutím abecedy na předchozím řádku o jeden znak dále. Pro šifrování se používá klíčové slovo. V prvním kroku se klíčové slovo opakovaně sepíše nad otevřený text, tak aby dosáhlo stejné délky. Tímto vznikají souřadnice pro Vigenèrův čtverec. Šifrový znak se získá vyhledáním odpovídajícího řádku a sloupce v tabulce.

A	B	C	D	E	F	...
B	A	B	C	D	E	...
C	B	C	D	E	F	...
D	C	D	E	F	G	...
E	D	E	F	G	H	...
...	.	.	.	.	.	...

Tab. 1.2: Část Vigènerova čtverce

Při použití klíčového slova kratšího než otevřený text však šifra dostává cyklický charakter. Na příkladu níže si lze všimnout, že každý třetí znak je šifrován stejnou abecedou.

klíčové slovo	V U T V U T
otevřený text	E N I G M A
šifrový text	Z H B B G T

Později tak přichází myšlenka použití náhodného klíče stejné délky jako otevřený text. Za podmínky použití každého klíče pouze jednou se šifra stává nerozluštitelnou. Roku 1918 vznikají knížky, jejichž listy obsahovaly unikátní seřazení náhodných znaků. Po prvním použití stránky s unikátním klíčem, byl předpoklad stránku zničit. Takovéto metody se nazývají jednorázové tabulkové šifry. V praxi, zejména pak v armádním prostředí, kde denně koluje nespočet zpráv, však navzdory své bezpečnosti nejsou velmi efektivní [3].

1.3 Německá kryptografie během 1. světové války

Během první světové války využívalo Německo řadu kryptografických metod. U telegrafické komunikace se jednalo například o diplomatické kódy s označením 13040 a později 0075. Ty jsou známé především díky Zimmermannovu telegramu, ve kterém nabízí Mexiku spojenectví ve válce proti Spojeným státům americkým. Kódy využívaly knížky s výčtem nejčastěji užívaných slov. Každému slovu byl přiřazen náhodný tří až pěti místní číselný kód. Některá slova se mohla v jednotném textu vyskytnout vícekrát, například „stop“. Pro zvýšení komplexnosti bylo těmto slovům přiřazeno více různých kódů [5].

Hlavním komunikačním médiem bylo radiové spojení. To přineslo možnost bezdrátové komunikace. Stěžejní se tento typ komunikace stal zejména pro námořnictvo a letectvo. Tedy hlavně tam, kde nebylo možné komunikovat pomocí drátového telegrafu. Vzhledem k plošnému šíření radiových vln však bylo velmi jednoduché tako-

vouto komunikaci odposlouchávat. Pro bezpečné použití tedy bylo nutno komunikaci šifrovat.

1.3.1 ADFGX a ADFGVX šifra

ADFGVX je jedna z nejznámějších válečných šifer určena pro bezdrátovou komunikaci. Dříve se však používala její starší verze ADFGX. Obě šifry jsou založeny na kombinaci substituce a transpozice [6].

ADFGX

Šifra ADFGX se začala používat 5 března roku 1918.Jejím tvůrcem byl německý radiový operátor Fritz Nebel. Substituční část této šifry funguje na principu Polybiova čtverce, tabulky o 5 řádcích a 5 sloupcích viz tab. 1.3.

	A	D	F	G	X
A	G	W	R	E	T
D	Q	D	N	I/J	L
F	C	O	B	K	Y
G	F	U	H	M	Z
X	P	V	S	A	X

Tab. 1.3: příklad Polybiova čtverce

Každý řádek a sloupec je označen příslušným znakem. V tomto případě znaky ADFGX. Dvojice znaků určujících řádek a sloupec pak substituuje znak otevřené abecedy. Například „XG“ představuje znak „a“. Předpokládá se, že znaky ADFGX byly použity kvůli své velmi rozdílné podobě v Morseově kódu. Ten se používal právě pro přenos zpráv skrze radiový telegram.

V dalším kroku se na šifrový text aplikuje transpozice. Šifra využívá sloupcovou transpoziční metodu popsanou v kapitole 1.2.3. Dlouhé zprávy se rozdělily na bloky různé délky. Použité klíče se denně měnily [6].

Později se začalo předpokládat, že šifra byla prolomena. Bylo proto nezbytné ji upravit.

ADFGVX

Roku 1918 dne 1. června byla poslána první zpráva šifrována pomocí ADFGVX šifry. Její princip zůstal stejný, avšak Polybiuv čtverec byl rozšířen o dodatečný řádek a sloupec s označením „V“. Rozšíření dovolilo přidání číslic 0,. . . ,9 a rozdělení znaků „i“ a „j“.

	A	D	F	G	V	X
A	1	W	R	E	T	3
D	Q	D	4	I	8	G
F	C	0	B	K	Y	F
G	2	U	H	M	5	A
V	N	9	Z	S	L	7
X	P	V	6	J	X	O

Tab. 1.4: příklad rozšířeného Polybiova čtverce

Při znalosti původní verze se dala šifra snadno identifikovat. Z výskytu šestého písmene v šifrovém textu se dalo usoudit, že byl čtverec rozšířen. Samotnou substituci by bylo snadné prolomit. Hlavní silou však byla transpozice. Při použití sloupcové metody se dvojice substituující jeden znak rozdělí do dvou sloupců. Po přehození sloupců se od sebe oddělí a odstraní tak frekvence otevřeného textu. Právě kvůli tomuto se Německo určitou dobu domnívalo, že je šifra neprolomitelná [6].

2 Enigma

Manuální šifrování se s přibývajícím množstvím zpráv a jejich délek stává méně efektivní a časově náročné. Lidský faktor se rovněž může při postupu dopustit chyby. Z tohoto důvodů se lidé pokoušeli vymyslet různé pomůcky či šifrovací stroje, které by práci usnadnili.

Historickým příkladem mohou být šifrovací disky. Ty byly tvořeny dvěma soustřednými kotouči. Po obvodu každého z nich byla sepsána abeceda. Vnější kotouč zastupoval otevřenou abecedu, vnitřní pak abecedu šifrovou. Natočením vnějšího disku, tak lze získat jednoduchou monoalfabetickou šifru. Vícenásobným natočením vnějšího disku během jedné zprávy lze šifru rozšířit na polyalfabetickou [3].

Nejznámnějším šifrovacím strojem je Enigma. Stroj, který reprezentuje bitvu mezi kryptografií a kryptoanalýzou. Stroj na němž záviselo tolik životů, jako nikdy dříve.

2.1 Historie

Arthur Scherbius byl německý vynálezce. Zabýval se především modernizací kryptografických metod. Díky jeho znalostem elektrotechnického inženýrství navrhl šifrovací stroj využívající pohyblivé rotory s elektrickým vedením. Roku 1918 si nechal návrh patentovat a založil firmu Scherbius & Ritter. Návrh prezentoval vojenským kruhům, Německému námořnictvu a zahraničním kancelářím. V této době však o stroj nebyl zájem. Důvodem byla jeho vysoká cena, ale taky víra v bezpečnost původních šifrovacích metod [7].

Roku 1923 přechází patent pod firmu Chiffriermaschinen-AG, kde Scherbius zaujímá pozici člena představenstva. V tomto roce byly publikovány britské dokumenty pojednávající o první světové válce. V těchto dokumentech byl podrobně popsán odposlech Německé komunikace a jakým způsobem byli Britové schopni jej dešifrovat. Německo se zde poprvé dovídá, že jejich kryptografické metody byly prolomeny a začíná hledat metody nové. Byla zvolena právě Enigma firmy Chiffriermaschinen-AG [3].

První model Enigma A se začal komerčně prodávat roku 1923. Stroj byl velmi rozměrný a vážil přibližně 50 kg. Krátce poté byl představen model Enigma B. Oba stroje byly velmi podobné psacím strojům. Šifrování bylo zajištěno pomocí 4 neodnimatelných rotorů, ovládaných pomocí systému ozubených kol. Výchozí pozice rotorů se dala nastavit pomocí otočných knoflíků umístěných na boku stroje. Díky svým rozměrům nebyly tyto modely pro armádu atraktivní.

Enigma C byl první model využívající systém žárovek pro výstup místo psacího stroje. Rovněž zde byl poprvé využit reflektor, speciální rotor s kontakty pouze na

jedné straně. Všechny rotory byly pořád pevnou součástí stroje. Krom reflektoru bylo možné nastavit pozice rotorů pomocí ozubených disků v horní části stroje. Díky novému způsobu zpracování byl tento model výrazně menší a kompaktnější. Roku 1925 se začala masově vyrábět jeho upravené verze, Funkschlüssel C, která se o rok později začala využívat Německou armádou.

Roku 1927 se začal komerčně vyrábět a prodávat model Enigma D. Tento model měl snímatelný svrchní kryt, který umožnil přístup k nastavení rotorů. Využíval tři rotory umístěny na odnimatelné ose díky čemuž bylo možné rotory mezi sebou zpřeházet a vytvořit tak šest různých kombinací zapojení. Stejně jako klasické rotory byl u tohoto modelu nastavitelný i reflektor. Model se stal hlavním produktem firmy Chiffriermaschinen-AG a všechny následující modely jsou z velké části založeny právě na jeho designu. Téhož roku začal vývoj verze pro Německou armádu a roku 1932 byla představena jako Enigma I, známá také jako Wehrmacht Enigma.

Wehrmacht Enigma byla založena na modelu D, ale měla nenastavitelný reflektor. Od komerční verze se také lišila přidaným propojovacím polem v přední části stroje. To bylo přidáno z důvodu zvýšení bezpečnosti šifry. Původně byla dodávána se 3 rotory s odlišným vnitřním propojením od komerční verze. Od roku 1939 se začala dodávat s pěti rotory, zvyšujíc počet různých kombinací zapojení na 60 ($5 \times 4 \times 3$). Tato verze byla používána Německou armádou „Heer“ a letectvem „Luftwaffe“. Roku 1934 implementovalo lehce upravenou verzi i Německé námořnictvo „Kriegsmarine“. Verze pro námořnictvo nesla označení M3 a byla dodávána spolu se sadou osmi rotorů. Začátkem roku 1942 byl pro námořnictvo vyvinut model M4, ten využíval 4 rotory a speciální tenký reflektor.

Šifrovací stroj s obecným názvem Enigma nebyl využíván pouze Německem. Státy jako Itálie a Španělsko využívaly komerční modely D. Jiné státy, jako Švédsko a Japonsko, pak využívaly své upravené verze modelu D. Celkově se však konečný počet prodaných kusů odhaduje na 100 000 [7] [8].

2.2 Popis šifrátoru

Tato práce se dále věnuje zejména modelu Wehrmacht Enigma viz obr. 2.1. Teoretická část popisu šifrátoru a funkcionality čerpá z technických detailů dostupných na stránkách *Crypto Museum* [8] a *Technical Details of the Enigma Machine* [2].

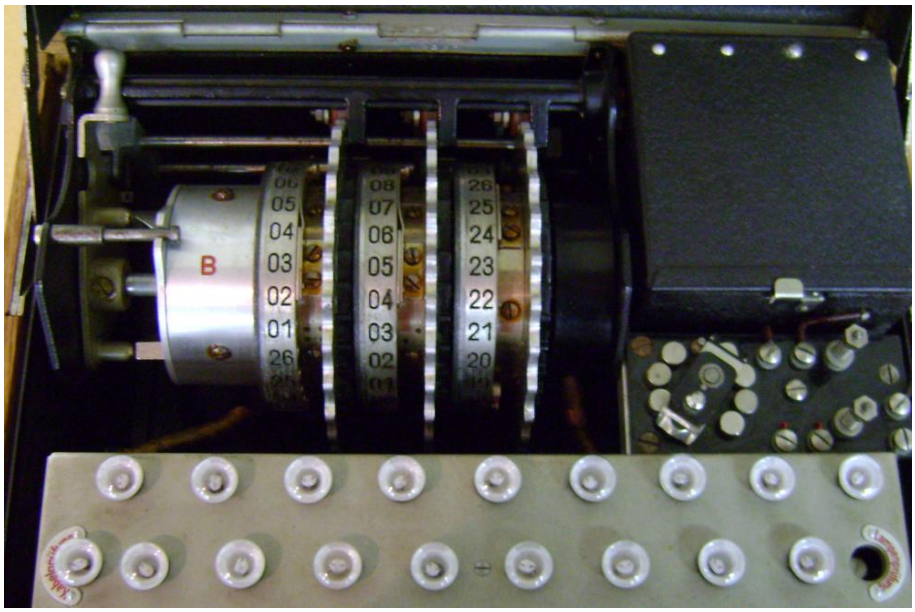
Stroj byl vestavěn v dřevěném boxu pro lepší přenositelnost. Box bylo možné uzamknout klíčem. Ve vrchním víku bylo po odklopení možné najít návod k obsluze, sadu náhradních žárovek, případně náhradní propojovací kabely.

Obr. 2.1: Wermacht Enigma. Autor: Dirk Rijmenants, dostupné z [2].

Samotný šifrovací stroj tvořilo několik částí:

- **Vstupní klávesnice**: Tvořena 26 znaky v Německém QWERTZ rozložení. Byly použity pouze velké znaky. Na klávesnici se nenacházely číslice, znaky s diakritikou ani speciální znaky.

- **Sada žárovek**: Nad klávesnicí se nacházel panel s 26 žárovkami ve stejném rozložení jako klávesnice. Tyto žárovky sloužily jako podsvícení znaků ve svrchním krytu, a také jako výstup šifry. V panelu se v prostřední řadě po obou koncích nacházely testovací body. Po pravé straně byla umístěna zdířka pro testování žárovek, na levé straně se pak nacházela žárovka pro testování kabelů. Na obrázku 2.2 lze vidět z leva do prava: Reflektor B, tři rotory, vstupní rotor (černý válec) a box s baterií. Pod baterií se nachází kontakty pro vstupní zdroj. Níže pak pole žárovek.



Obr. 2.2: Wermacht Enigma s odklopeným krytem. Autor: Dirk Rijmenants, dostupné z [2].

- **Vyměnitelné rotory na společné ose**: Rotory, též zvané jako „scramblery“ byly hlavním šifrovacím prvkem stroje. Byly to kovové, nebo bakelitové disky s vnitřním propojením. Větší ozubený disk sloužil pro nastavení polohy rotoru. Jádro disku mělo na pravé straně pružinové kontakty, na straně levé pak kontakty ploché viz obr. 2.3. Uvnitř jádra byly různě propojeny kontakty pravé a levé strany. Kolem jádra byl pohyblivý kroužek s drážkou na levé straně, některé rotory pozdějších verzí měly drážek více. Kroužek byl označen čísly od 1 do 26 zastupujíc znaky abecedy od A do Z. Rotory stroje Enigma M3 byly značeny znaky abecedy. Natočením kroužku se dala nastavit pozice abecedy na kroužku vzhledem k vnitřnímu posunutí rotoru, kroužek se poté zajistil pinem. Kolem jádra na straně s pružinovými kontakty bylo ozubené kolo se stejným průměrem jako kroužek s drážkou. To využíval krokový mechanismus pro natočení rotoru.

Obr. 2.3: Rotor, z leva: levá strana rotoru, pravá strana rotoru. Autor: Dirk Rijmenants, dostupné z [2].

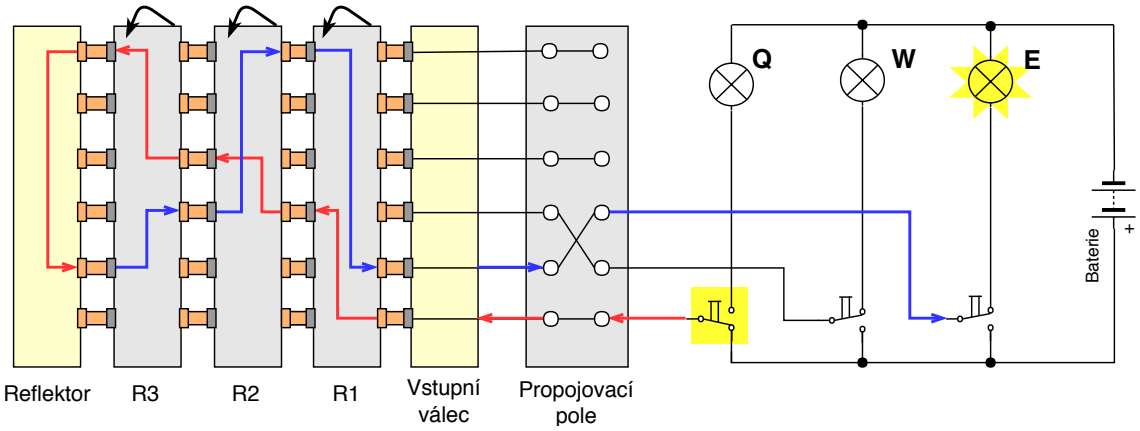
Se strojem byly původně dodávány tři rotory I,II,III. Později byly přidány dva další rotory IV a V.

- **Reflektor**: Speciální nepohyblivý rotor s kontakty pouze na jedné straně. Jeho interní propojení tvořilo 13 párů znaků. Propojení byly oboustranné. Díky jeho funkci nemohl být výstup šifry nikdy stejný jako vstupní znak. Používaly se dva typy: A a B. Pozdější čtyř rotorová verze Enigma M4 používala speciální tenké verze reflektorů A a B.
- **Vstupní válec**: Válec byl fixně upevněn a sloužil jako propojení mezi propojovacím polem a pravým rotorem.
- **Propojovací pole**: Nacházelo se za předním panelem přenosného boxu. Pomocí krátkých kabelů dodávaných se strojem bylo možné vytvořit páry znaků. Propojením znaku „A“ a „N“ vznikla obousměrná vazba. Při stisku klávesy „A“ do šifry vstoupil díky propojení znak „N“. Obdobně u výstupu šifry, pokud byl výstupem znak „N“ rozsvítila se žárovka „A“. Bylo možné vytvořit 0 až 13 párů, běžně však bylo doporučeno použití deseti.
- **Napájecí zdroj**: Stroje měly integrovanou baterii. Krom baterie bylo však možné připojit i externí zdroj. V horní části stroje se nacházel přepínač, kterým bylo možné zvolit vstupní zdroj energie. Proud generovaný napájecím zdrojem pouze procházel rotory, reflektorem, popřípadě propojovacím polem a napájel žárovky. Krokovací mechanismus pro pohyb s rotory byl mechanický [8] [2].

2.3 Popis funkcionality

Wermacht Enigma pracovala na elektro-mechanickém principu. Šifrování samotné bylo zajištěno pomocí elektrického obvodu procházejícího skrze rotory. Kdyby byly rotory pevně umístěny, produkoval by stroj pouze jednoduchou substituční šifru. Síla Enigmy však spočívala v pohybu rotorů. Rotory byly označeny jako rychlý, středně rychlý a pomalý. Každé stlačení klávesy vyvolalo krok rychlého rotoru. Další rotory se otáčely o jednu dvaceti šestinu a pouze v případě, že byl předchozí rotor v pozici s drážkou. Díky tomuto systému i po opakovaném stlačení stejné klávesy stroj produkoval rozdílný výstup. Mechanismus, který s rotory pohyboval byl čistě mechanický [2].

2.3.1 Vnitřní zapojení



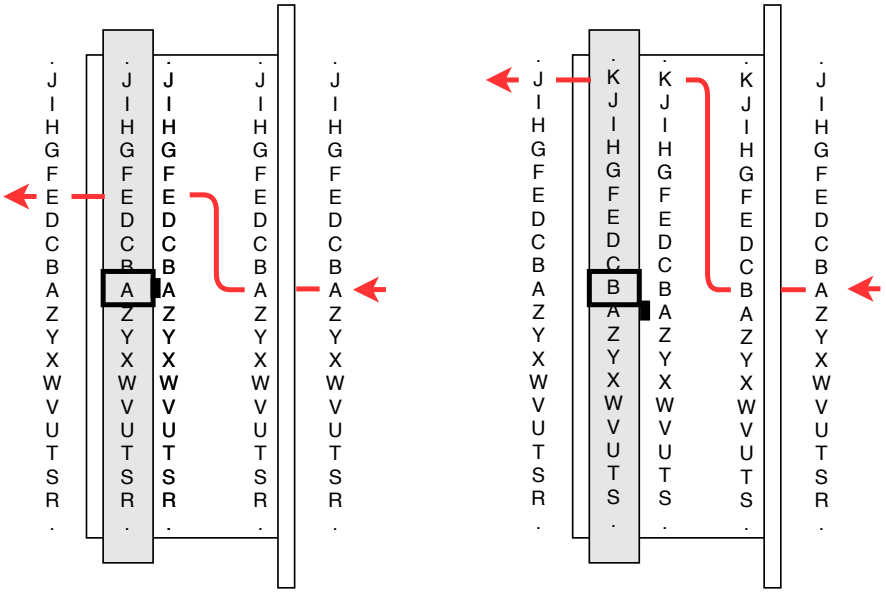
Obr. 2.4: Zjednodušené blokové schéma zapojení

Na obr. 2.4 je zobrazeno vnitřní zapojení Wermacht Enigmy. Schéma je kvůli přehlednosti velmi zjednodušeno. Všechny páry propojovacího pole, rotorů i reflektoru jsou obousměrné.

Stlačení klávesy nejdříve vyvolá krok patřičných rotorů. Proud poté ze směru od napájecího zdroje prochází stlačenou klávesou a vstupuje do propojovacího pole. Propojovací pole umožňuje přenastavit spojení mezi klávesnicí a vstupním válcem. U příkladu není spojení stlačeného znaku „Q“ změněno a do vstupního válce tak vstupuje beze změny. Skrze vstupní válec prochází proud postupně vnitřním zapojením rotorů R1, R2 a R3. V reflektoru se směr proudu otočí, a dále tak v opačném pořadí opět prochází všemi rotory R3,R2 a R1. Skrze vstupní válec vstupuje opět do propojovacího pole. V tomto případě je simulováno využité propojení znaků „W“ a „E“. Nakonec proud projde nestlačenou klávesou přes pole žárovek a rozsvítí tak příslušný znak odpovídající výstupu šifry [2].

2.3.2 Průchod signálu rotorem

Rotor měl dvě hlavní části. Kroužek s drážkou označený abecedou, jehož natočením se dala měnit jeho pozice vzhledem k vnitřnímu zapojení rotoru a samotné jádro s vnitřním propojením vstupních a výstupních pinů reálně vykonávajíc šifrování. Jádro bylo fixně spojeno s ozubeným diskem, pomocí jenž se dala manuálně nastavit pozice rotoru.



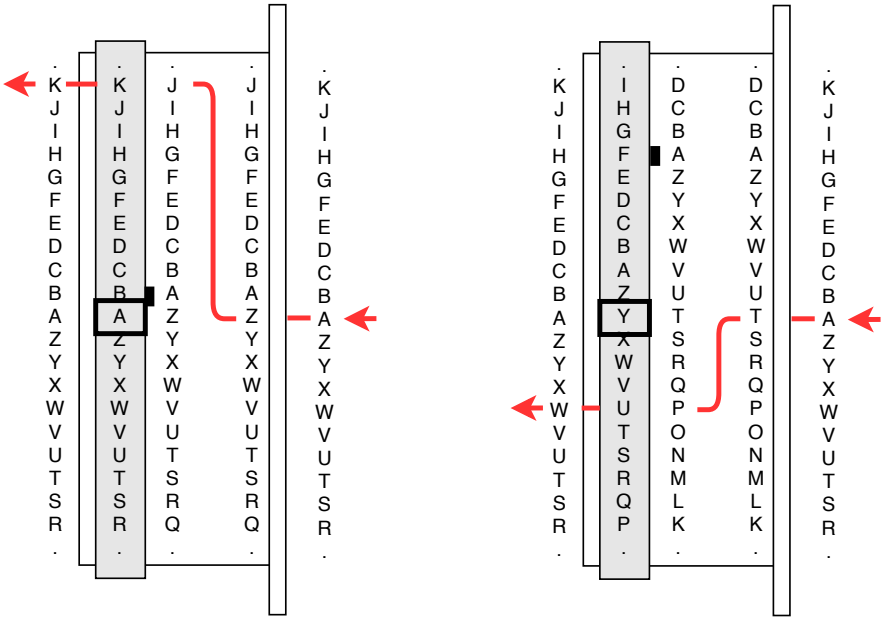
Obr. 2.5: Průchod signálu rotorem I s nastavením A-01 po dvojitém stlačení klávesy A

Na obr. 2.5 je zobrazena simulace průchodu signálu rotorem I po dvojitém stlačení klávesy A. Šedé pole představuje nastavitelný kroužek s drážkou, značka po jeho pravé straně značí pozici pinu, natočení kroužku vzhledem k vnitřnímu propojení. V tomto případě je kroužek nastaven v pozici A-01. Zvýrazněný obdélník v šedém poli znázorňuje pozici rotoru, znak který lze vidět v okénku v panelu stroje. Bílá část představuje jádro s vnitřním propojením spojené s ozubeným diskem.

Levá část představuje první stlačení klávesy A. Rotor je nastaven v pozici A. Signál vychází ze vstupního válce v pozici A a vstupuje do rotoru pinem A, který je vnitřně propojen s pinem E. Do dalšího rotoru vstupuje signál v pozici E.

V pravé části se rotor po opětovném stlačení klávesy A pootočil na pozici B. Signál opět vychází ze vstupního válce v pozici A, nově ale vstupuje do rotoru pinem B, který je spojen s výstupním pinem K. Díky natočení rotoru však nyní vstupuje do dalšího rotoru v pozici J.

Pohyblivý kroužek s drážkou je zajištěn v pozici pomocí pinu. Tento pin značí první kontakt vnitřního propojení. Nastavení s označením B-02 znamená, že je pohyblivý kroužek zajištěn pinem v pozici B. Obrázek 2.6 znázorňuje průchod signálu rotorem I s nastavením B-02 a F-06.



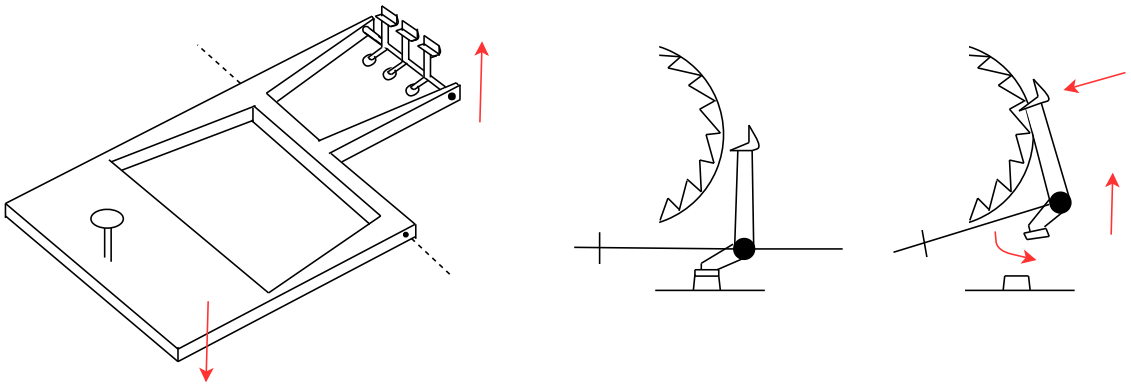
Obr. 2.6: Průchod signálu rotorem I, z leva: s nastavením B-02 a F-06

V levé části se rotor s nastavením B-02 nachází v pozici A. Signál vycházející ze vstupního válce v pozici A vstupuje do rotoru pinem Z, který je vnitřně propojen s pinem J. Díky nastavení kroužku je však vnitřní propojení posunuto o jednu pozici a tudíž vstupuje signál do dalšího rotoru v pozici K.

Vpravo je znázorněn průchod signálu rotorem I s nastavením F-06 v pozici Y. Signál vycházející ze vstupního válce v pozici A vstupuje do rotoru pinem T, který je vnitřně propojen s pinem P. Zde je však díky kombinaci pootočení rotoru a nastavení kroužku rotor natočen o 7 pozic vůči vstupům dalšího rotoru. Výstupní pin P je tedy srovnán se vstupním pinem W dalšího rotoru. Signál tak vchází do dalšího rotoru v pozici W [2].

2.3.3 Krokový mechanismus

Krok rotorů byl zajištěn pomocí překlopné krokovací lišty viz obr. 2.7.
Stlačení klávesy zatlačí lištu dolů a vyzdvihne konec se třemi západkami na společné ose směrem k rotorům. Každá západka dopadá na rozmezí ozubeného kola jednoho rotoru a kroužku s drážkou sousedního rotoru. Je-li rotor v pozici s drážkou západka zapadne do drážky, zachytí zub ozubeného kola a posune rotor o jednu



Obr. 2.7: Překlopná krokovací lišta a znázornění kroku rotoru

pozici. V opačném případě západka pouze sjede podél kroužku. Rychlý rotor rotuje při stisku každé klávesy, protože po své pravé straně nemá sousední rotor a západce tak nic nebrání v zachycení ozubeného kola.

Enigma M4 se čtyřmi rotory využívaná námořnictvem používá stejnou krokovou lištu se třemi západkami jako Wermacht Enigma. Čtvrtý rotor se tedy neotáčel a byl fixně uchycen. Bylo možné pouze manuálně nastavit jeho pozici [2].

Dvojitý krok

Krokový systém pracuje v principu tak, že se pravý rotor otáčí při každém stisku klávesy a ostatní rotory se otáčí pouze tehdy, pokud je rotor po jejich pravé straně v pozici s drážkou. Prostřední rotor však vykoná extra krok pokud je sám v pozici s drážkou. Toho je docíleno tak, že západka pootočí rotor i v případě zatlačí-li do jeho drážky.

Dvojitý krok lze demonstrovat na příkladu použití rotorů III-II-I v případě, že rotor I vykoná krok při skoku z Q na R, rotor II při skoku z E na F a rotor III z V na W.

KDO, KDP, KDQ, KER, LFS, LFT, ...

Rotory jsou ve výchozím nastavení v pozicích KDO. Každá další trojice reprezentuje stlačení klávesy. Při prvním stlačení klávesy se pravý rotor dostane do pozice P. Při druhém stlačení klávesy se pravý rotor dostane do své pozice s drážkou Q. Při následujícím stlačení klávesy tedy západka zapadne do drážky pravého rotoru a posune zároveň prostřední rotor do pozice E. Prostřední rotor se tedy dostal do své pozice s drážkou a při následujícím stlačením klávesy tak západka zapadne do rozhraní mezi prostředním a levým rotorem a posouvá tak všechny tři rotory o jednu pozici [2].

2.3.4 Šifrování

Enigma šifruje otevřený text algoritmem založeným na polyalfabetické substituční šifře. Algoritmus má pár specifických rysů

- znak nikdy nemůže být šifrován sám sebou,
- symetrie šifry – byl-li znak „Q“ zašifrován na znak „E“, bude při naprosto identickém počátečním nastavení zašifrován i znak „E“ na znak „Q“.

Obě specifika jsou dosaženy použitím obousměrných párů vnitřního zapojení rotorů, propojovacího pole a reflektoru.

Pro zašifrování zprávy a její úspěšné dešifrování bylo nutné naprosto stejné počáteční nastavení stroje. Díky symetrii šifry operátor napsal na stroji šifrový text a dostal zpět původní zprávu. Pro zajištění bezpečnosti bylo nastavení každý den měněno. Každý měsíc byly rozesílány tabulky, které pro jednotlivé dny definovaly

- použité rotory v daném pořadí,
- natočení nastavitelného kroužku rotoru,
- použité páry propojovacího pole,
- čtyři různá počáteční nastavení rotorů (hesla).

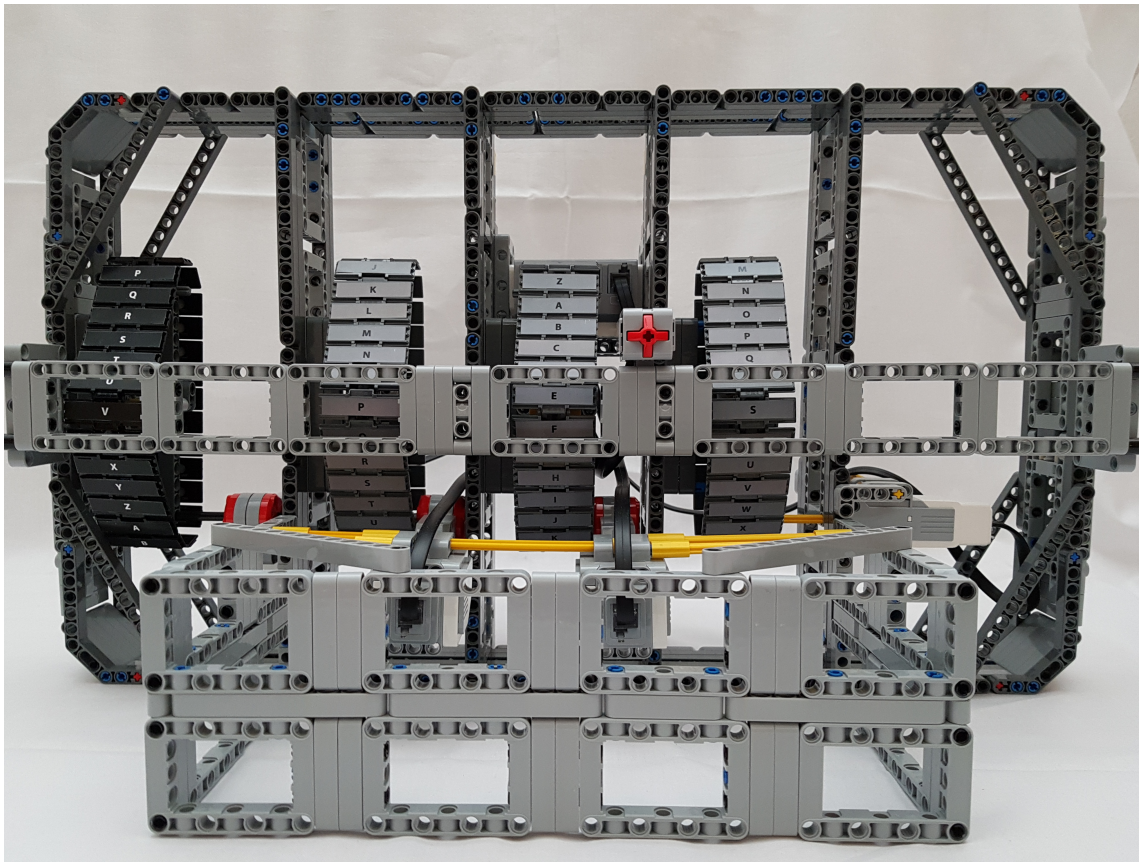
V posledním sloupci byly uvedeny čtyři hesla pro konkrétní den. Aby se dalo rozlišit jaké heslo na danou zprávu použít, bylo na začátek zpráv přidáváno pěti znakové slovo. Toto slovo bylo složeno ze dvou náhodně zvolených znaků a jednoho ze čtyř hesel. Slovo sloužilo pouze jako heslo pro nastavení stroje a nebylo dešifrováno se zbytkem zprávy. Byla-li zpráva rozdělena na více částí používalo se vždy jiné heslo pro každou z nich [9].

3 Původní stav demonstrátoru

V rámci bakalářské práce Bc. Jakuba Jančíka [1] byl zkonstruován demonstrátor šifrátoru Enigma. Model byl vytvořen pomocí kombinace stavebnic LEGO Technic a LEGO Mindstorm EV3. Model demonstruje proces šifrování pomocí znázornění průchodu signálu rotory.

3.1 Konstrukce demonstrátoru

Konstrukce modelu viz obr. 3.1 je tvořena vnějším rámem a funkčními prvky. Černý pohyblivý pás po levé straně představuje vstup a následně výstup šifry. Zbylé tři šedé pohyblivé pásy pak reprezentují rotory šifrátoru. Všechny pásy jsou označeny znaky abecedy a ovládány pomocí motorů skrze řídící jednotku LEGO EV3. Model rovněž využívá tlakového senzoru, který zaujímá funkci tlačítka [1].



Obr. 3.1: Demonstrátor šifrátoru Enigma

3.2 Programy

Pro zajištění funkcionality demonstrátoru byly v rámci práce sepsány dva programy. Tyto programy byly vytvořeny pomocí programu LEGO Mindstorms EV3 Home edition. S prací byl dostupný pouze strojový kód. Následující popis daných programů je vyvozen z analýzy chodu programů a na základě popisu v původní práci Bc. Jakuba Jančíka [1].

3.2.1 Kalibrační program

Program slouží k uvedení zařízení do výchozího nastavení. Spuštěním daného programu započne kalibrace čtyř pásů. Stisknutím tlakového senzoru se nejdříve spustí kalibrace prvního pásu. Pás se otáčí po dobu stlačeného senzoru. Po uvolnění senzoru pár zastaví v aktuální pozici. Zaznění zvukové signalizace indikuje přechod k dalšímu pásu. Po zaznění zvukové signalizace je vymezen čas 2s na stlačení tlakového senzoru. Pokud ve vymezené době ke stlačení senzoru nedojde zazní opět zvuková signalizace indikující přechod k dalšímu pásu. Po kalibraci čtvrtého pásu je program ukončen [1].

Pro správný chod demonstračního programu je nutné nastavit pásy do pozic v daném pořadí: A, X, D, H.

3.2.2 Demonstrační program

Demonstrační program simuluje chod šifrovacího stroje Enigma pomocí znázornění průchodu signálu jednotlivými rotory. Program ve smyčce šifruje otevřený text „VITEJTENAVUT“. Výstupem šifry je poté šifrový text „WXKYYRBMUMDH“.

Po spuštění programu se čeká na stlačení tlakového senzoru. Při stlačení započne demonstrace šifrování prvního znaku otevřeného textu. Nejprve je nastaven černý pás na pozici shodnou se znakem otevřeného textu. Černý pás reprezentuje znak stlačený na klávesnici.

Následně se začíná točit první šedý pás zleva. Šedé pásy reprezentují rotory. Na rozdíl od skutečného stroje jsou pásy v opačném pořadí. První pás zleva tedy reprezentuje pravý rotor. Pás se nastaví na pozici, ve které by do rotoru vstupoval signál. Po krátkém časovém intervalu je pás znovu natočen, tentokráte do pozice, ze které by signál z rotoru vystupoval.

Proces je opakován zbylými pásy. Po natočení posledního pravého pásu by signál vstoupil do reflektoru, ten však není reprezentován. Po krátkém časovém intervalu se pravý šedý pás natočí do pozice, do které by signál vstoupil do rotoru skrze reflektor. Celý proces natočení do vstupní a výstupní pozice se zopakuje v opačném směru.

Po konečném natočení prvního šedého pásu zleva, do jeho výstupní pozice, se nakonec natočí černý pás na pozici výstupu šifry. Pozice černého pásu po skončení jednoho cyklu reprezentuje rozsvícenou žárovku pod daným výstupním znakem šifry.

Zakončením cyklu pro jeden znak program čeká na opětovné stlačení tlakového senzoru. Po stlačení poté program pokračuje demonstrací šifrování dalšího znaku. Při ukončení demonstrace na posledním znaku otevřeného textu program po stlačení tlakového senzoru nastaví pásy na výchozí pozice a je připraven začít proces znovu [1].

4 Návrh komunikačního rozhraní

Cílem práce bylo rozšířit stávající demonstrátor šifrátoru o možnost zadat text pomocí klávesnice a zobrazit výstup šifry na obrazovce. Bylo navrženo řešení, které využívá systému Ev3dev [10] pro rozšíření možností práce s řídící jednotkou Lego Ev3. Jako programovací jazyk pro realizaci funkcionality Enigmy byl zvolen Python, který je systémem Ev3dev podporován. Dále bylo pro práci navrženo použití Bluetooth klávesnice a mobilního telefonu či tabletu jako zobrazovače.

4.1 Ev3dev

Ev3dev je operační systém založený na Debian Linuxu. Systém je kompatibilní s LEGO Mindstorms zařízeními. Implementovaný framework umožňuje sytému pracovat s originálními periferiemi LEGO, jako jsou senzory a motory. Linuxové jádro pak dodává přístup k běžně dostupným Linuxovým balíčkům a možnost pracovat s USB (*Universal Serial Bus*), Bluetooth a dalšími klasickými periferiemi. Ev3dev podporuje mnoho programovacích jazyků mezi, které patří například Python, Java, C/C++ a další.

Systém samotný není firmware, proto nijak nezasahuje do původního systému řídící jednotky LEGO. Ev3dev je nainstalován na microSD (*Secure Digital*) kartu. Po vložení karty do zařízení a následném zapnutí daného zařízení se načte systém z SD karty. Pro odstranění systému stačí zařízení vypnout a odstranit SD kartu. Při opětovném zapnutí zařízení načte opět původní firmware.

Další výhodou, kterou systém přináší je možnost konfigurace zařízením pomocí SSH shellu.

Systém Ev3dev je dostupný ve dvou verzích a to ev3dev-jessie a ev3dev-strech beta. Pro realizaci práce byla zvolena verze ev3dev-strech beta, která dodává plnou podporu jazyka Python3 a nové funkce pro práci s periferiemi [10].

4.2 Návrh zobrazovače

Bylo navrženo použití mobilního telefonu či tabletu jako zobrazovače. Pro spojení zobrazovače a řídící jednotky je možné využít Bluetooth, nebo Wifi s použitím adaptéru. Samotné uživatelské rozhraní pro ovládání modelu šifrátoru bylo navrženo jako webová aplikace. Tato webová aplikace pak předpokládá využití jazyka Python3 a microframeworku Flask.

4.2.1 Flask

Flask je microframework pro tvorbu webových aplikací pomocí programovacího jazyka Python. Jakožto microframework není náročný a má malou až žádnou závislost na externí knihovny [11].

Webová aplikace byla nejdříve navržena a testována pomocí Python CGI (*Common Gateway Interface*), PHP (*Hypertext Preprocessor*) a JQuery. Všechny zmíněné přístupy však byly postaveny na přímém spuštění externího python scriptu. Komunikace s webovým serverem tak vyžadovala individuální spuštění Python překladače, který předal vstupní informace scriptu, ten informace zpracoval a výstup předal zpět webovému serveru. Celý tento proces způsobil zhruba 8 sekundovou odezvu a tím i velmi pomalý běh aplikace.

Z výše uvedeného důvodu byl nakonec zvolen Flask. Navržená aplikace využívá vestavěný vývojový webový server, a tak dokáže přímo pracovat s funkcemi pro ovládání periferií a zároveň poskytnout klientské straně danou webovou stránku. Informace jsou v tomto případě předávány přímo aplikaci, která je zpracovává a vrací výstup přímo klientské straně. Nevzniká zde nadbytečná odezva a chod aplikace je tak plynulý.

Vestavěný vývojový server se nedoporučuje používat v produkci, protože dokáže zpracovávat pouze jeden požadavek zároveň. Díky jeho specifickému zpracování informací, ale disponuje mnohem rychlejším načtení stránky. Vzhledem k povaze lokální aplikace jsou tak jeho vlastnosti vyhovující [11].

5 Výsledky studentské práce

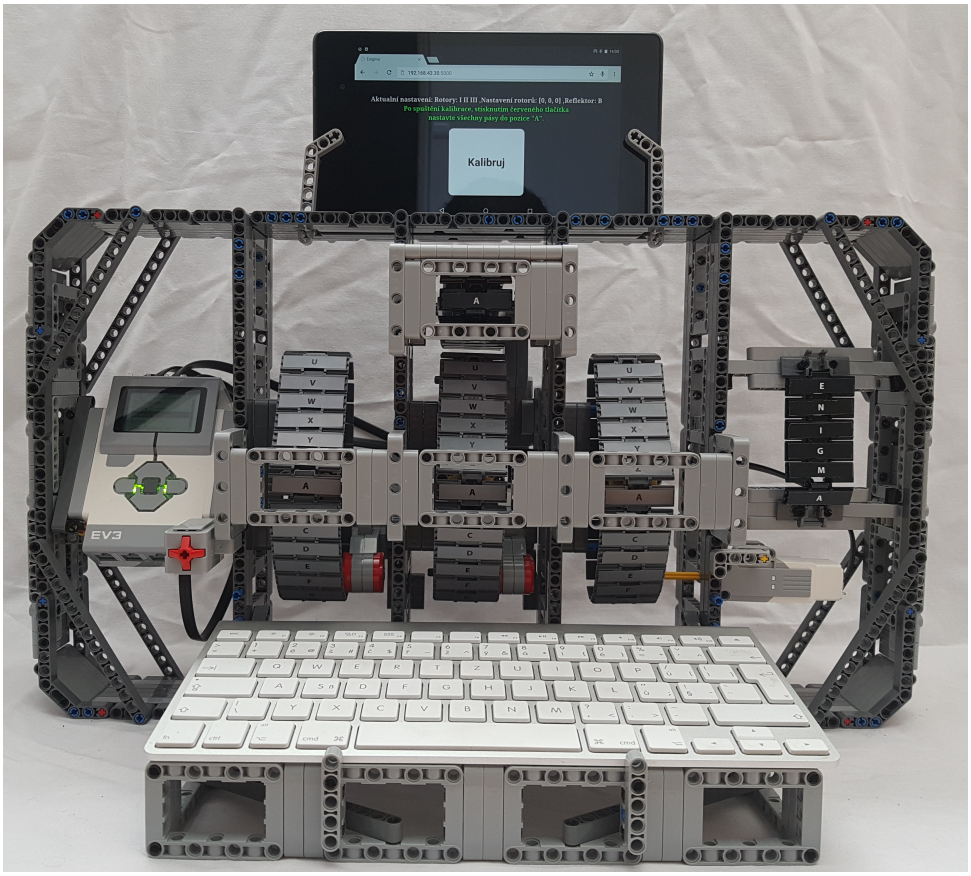
V rámci bakalářské práce byl upraven původní model demonstrátoru.

Nově upravený model nyní funguje jako šifrátor. Černý pás slouží pro zobrazení vstupu a výstupu šifry, šedé pásy pak demonstrují pohyb rotorů při stisku kláves. Pomocí klávesnice je možné zadat jakýkoliv text, který je následně zašifrován. Výsledný šifrový text je zobrazen na obrazovce tabletu.

Šifrátor byl navržen podle funkcionality Wermacht Enigmy. Využívá stejných možností nastavení, které poskytoval skutečný stroj. Toto nastavení je z důvodu udržení jednoduchosti aplikace možno změnit pouze přímou úpravou skriptu, dále popsáno v kapitole 5.3.1.

5.1 Konstrukce

Model využívá z velké části původní konstrukce demonstrátoru viz obr. 5.1.



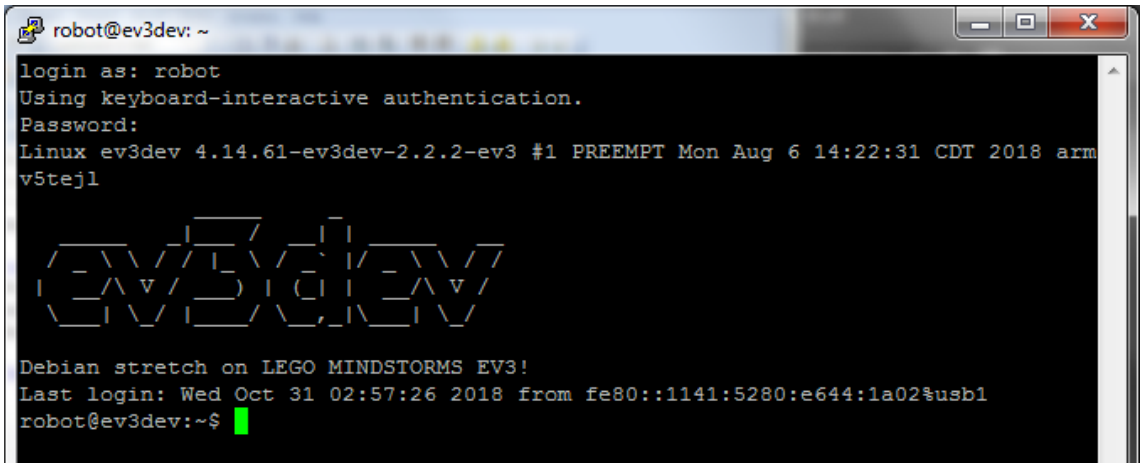
Obr. 5.1: Konstrukce modelu šifrátoru

Černý pás byl upraven a přemístěn nad šedé pásy, pro vizuální odlišení jeho funkce od ostatních pásů. Motory, které byly uchyceny ve vystupující konstrukci v přední

části modelu byly přesunuty do zadní části modelu. Přední konstrukce byla poté snížena a nově slouží jako místo pro uchycení klávesnice. Vnější rám byl doplněn o stojan pro zobrazovací zařízení. Z estetických důvodů byla nakonec řídící jednotka přesunuta do přední části modelu. Jednotku je pak možné pro snadnější přístup k výměně baterií zaklopit.

5.2 Systém Ev3dev

Pro rozšíření možností práce s řídící jednotkou EV3 byl použit systém Ev3dev viz obr. 5.2. Systém je nainstalován na microSD kartě. Systém nijak nezasahuje do původního firmware řídící jednotky. Po vytažení microSD karty se po zapnutí jednotky opět načte původní systém.



Obr. 5.2: Systém Ev3dev

5.2.1 Vytvoření spojení

Řídící jednotku je možné spojit s počítačem nebo chytrým telefonem či tabletem přes USB, Bluetooth nebo Wifi (*Wireless Fidelity*) s využitím adaptéru.

Při využití Bluetooth je však spojení velmi nestabilní a v některých případech se při prvním spuštění systému modul Bluetooth z důvodu neznámé chyby neinicializuje a systém je nutné restartovat. Z tohoto důvodu byla vybrána možnost spojení skrze Wifi, která poskytuje snazší a stabilnější spojení. Aby však bylo možné využít spojení Wifi bylo nutné dokoupit micro Wifi adaptér, jelikož jednotka tuto možnost samostatně neposkytuje.

Pro přímou práci se systémem je nutné navázat SSH spojení skrze terminál např. Putty, případně JuiceSSH.

5.2.2 Úprava systému

Pro zajištění fungování webové aplikace bylo nutné uživateli www-data, který je využíván webovým serverem, přiřadit dodatečná práva pro práci s funkcemi pro ovládání periferií. Toho bylo docíleno přidáním uživatele www-data do uživatelských skupin: *tty, dialout, plugdev, users, i2c, input, ev3dev, bluetooth, robot, audio.*

Do systému pak bylo rovněž třeba doinstalovat potřebné moduly:

- python-pip – Balíčkový manažer pro softwarové python moduly.
- virtualenv – Virtuální prostředí pro python.
- Flask – Python microframework pro tvorbu webových aplikací.
- cron – Softwarový démon pro automatizaci.

5.2.3 Příprava prostředí

Před tvorbou samotné aplikace bylo potřeba vytvořit virtuální prostředí pro Python *venv*. Virtuální prostředí poskytuje izolovanou sadu adresářů a vlastní binární soubor Python, kterým je spouštěna Flask aplikace. Toto virtuální prostředí bylo nutné doplnit o moduly *ev3, ev3dev2* umožňující práci s řídící jednotkou a jejími periferiemi. Tyto moduly byly převzaty z výchozího Python adresáře systému Ev3dev.

Dále byla vytvořena adresářová struktura viz obr. 5.3. Struktura byla tvořena podle požadavků Flask aplikace, která požaduje umístění určitých souborů ve specifických adresářích.

```
/var/www/Flask
├── /venv
└── /Enigma
    ├── enigma.py
    ├── /templates
    │   ├── index.html
    │   ├── calibrate.html
    │   ├── cipher.html
    │   └── output.html
    ├── /static
    │   └── /styles
    │       └── styles.css
    ├── /Controler
    │   └── enigmaV4.py
    └── /Config
        └── config.py
```

Obr. 5.3: Adresářová struktura aplikace

5.3 Programová část

Aplikace je tvořena třemi Python soubory: *enigmy.py* zajišťující řízení aplikace, *enigmaV4* definující potřebné funkce a třídy umožňující šifrování a *config.py* kde jsou uloženy globální proměnné a seznamy pro nastavení stroje.

Samotné uživatelské prostředí je pak tvořeno pomocí čtyř HTML (*Hypertext Markup Language*) souborů a stylisováno pomocí kaskádových stylů *styles.css*.

5.3.1 Řízení aplikace: enigma.py

Hlavní soubor „enigma.py“ se stará o řízení aplikace. V souboru je vytvořen objekt šifrového stroje s daným výchozím nastavením viz kap. 5.3.1. Dále se soubor stará o příjem požadavků na zobrazení konkrétní webové stránky. Pomocí definovaných funkcí hlídá zadanou adresu a vrací příslušnou stránku. Na příkladu 5.1 je vidět funkce, která po zadání IP (*Internet Protocol*) adresy řídící jednotky vrátí domovskou stránku *index.html*.

```
@app.route('/')
def home():
    return render_template('index.html',settings=settings)
```

Výpis 5.1: Funkce pro domovskou stránku

Aplikace umožňuje také pracovat s uživatelským vstupem skrze formulář na webové stránce, který dokáže zpracovat a následně předat výstupní webové stránce.

```
@app.route('/sifruj/', methods=['GET','POST'])
def cipher():
    if request.method == 'POST':
        input=request.form['input']
        out=machine.enc_text(input)
        return render_template('output.html',out=out)
```

Výpis 5.2: Funkce pro zadání vstupního textu

Na příkladu 5.2 jsou data poskytnutá webovým formulářem uložena do proměnné *input* a předána funkci *machine.enc_text*, která zadaný text zašifruje. Výsledný šifrový text je pak uložen do proměnné *out*, která je předána výstupní webové stránce, na které je šifrový text zobrazen.

Soubor nakonec definuje jak má být aplikace spuštěna viz 5.3.

```
if __name__ == '__main__':
    app.run(host='0.0.0.0', debug=True)
```

Výpis 5.3: Definice spuštění aplikace

Pro spuštění je využit vestavěný vývojový server a tento zápis umožňuje jeho spuštění pomocí příkazu *venv/bin/python3 enigma.py*. Daný server je spuštěn na zadané IP adrese 0.0.0.0 a výchozím portu 5000. Konkrétní IP adresa dělá server dostupný na všech lokálních IP adresách.

Výchozí nastavení stroje

Šifrátor je vytvořen s daným nastavením pomocí řádku 5.4. Pro změnu nastavení je nutné přímo upravit skript.

```
machine = enigmaV4.Machine(rotors='I II III',
                           reflector='B',
                           ring_settings=[0, 0, 0],
                           plugboard=[('A', 'X')])
```

Výpis 5.4: Inicializace stroje

Na příkladu je vytvořen stroj využívající rotory I - II - III v daném pořadí a reflektor B. Argument *ring_settings* udává posunutí kroužku rotoru vůči vnitřnímu propojení. Posunutí je možné zadat v rozmezí od 0 po 25. Ve výchozím nastavení jsou použity rotory s nulovým posunutím. Posledním argumentem je *plugboard*, neboli propojovací pole. U příkladu jsou propojeny pouze znaky „A“ a „X“.

5.3.2 Konfigurační modul: config.py

Soubor config.py slouží jako konfigurační modul, který inicializuje periferie. Aby se s inicializovanými motory a senzory dalo dále pracovat jsou zde přiřazeny globálním proměnným. V souboru jsou pak také uloženy další globální proměnné jako je systémová pozice motorů *relative_pos[]* nebo seznam rotorů a reflektorů viz 5.5. Tento seznam je využívaný pro inicializaci systémového stroje a je zhotoven podle technických detailů reálně využívaných částí dostupných na stránkách *Technical Details of the Enigma Machine* [2].

Klíče v seznamu označují modely. U rotorů jsou uvedeny vnitřní propojení (*wiring*) a pozice označující drážku na vnějším kroužku rotoru (*stepping*).

Pozice drážky neoznačuje její skutečnou pozici, ale znak, který je vidět v okénku stroje. Pokud by se rotor I nacházel v pozici „Q“ a došlo by ke stisku klávesy, rotor by se posunul do pozici „R“. Ve stejnou chvíli by se západka zachytila v jeho drážce a posunula tak do následující pozice i sousední rotor.

```
ROTORS = {
    'I': {
        'wiring': 'EKMFLGDQVZNTOWYHXUSPAIBRCJ',
        'stepping': 'Q',
    },
    ...
}

REFLECTORS = {
    'B': 'YRUHQSLDPXNGOKMIEBFZCWVJAT',
    'C': 'FVPJIAOYEDRZXWGCTKUQSBNMHL',
}
```

Výpis 5.5: Seznam rotorů a reflektorů

Vnitřní propojení rotorů a reflektoru je vždy v závislosti k otevřené abecedě. U rotoru I je tedy vstupní pin „A“ propojen s výstupním pinem „E“.

abe: ABCD...
wiring: EKMF...

5.3.3 Ovládací modul: enigmaV4.py

Soubor enigmaV4.py slouží jako přídavný modul definující třídy pro vytvoření systémového šifrovacího stroje a funkce umožňující vlastní šifrování a kalibraci.

Kalibrace

Pro správný chod modelu je nutné synchronizovat softwarovou pozici rotorů s reálnou pozicí pásů. Pásy nelze do pozice jednoduše nastavit manuálně. Z tohoto důvodu je nutná softwarová kalibrace.

Spuštění kalibrace je indikováno zvukovým signálem. Poté se postupně začínají roztáčet jednotlivé pásy. Program vstoupí do smyčky, ve které otáčí motorem a čeká na stisknutí tlakového senzoru. Při stisku tlakového senzoru je motor zastaven v aktuální pozici a program opouští smyčku. Po krátkém časovém intervalu program následně opakuje proces pro ostatní motory. Ukončení kalibrace je opět indikováno zvukovým signálem.

Pro pohyb motorů byla zvolena funkce *run_to_rel_pos()*, která otáčí s motorem o zvolený úhel. Funkce z výpisu 5.6 tedy otočí s motorem o 60° rychlostí 125°/s. Argument *stop_action=""* s hodnotou „hold“ aktivně zastaví motor v aktuální pozici.

Jednotlivé hodnoty byly zvoleny tak, aby vyvolávaly plynulý pohyb, který je možné snadno zastavit v určité pozici.

```
def calibrate():
    st = True
    Sound.speak('Kalibruji').wait()
    while st:
        co.mA.run_to_rel_pos(position_sp=60, speed_sp=125,
            stop_action="hold")
        if ts.is_pressed:
            co.mA.stop(stop_action="hold")
            co.relative_pos[3]=0
            st = False
    ...
```

Výpis 5.6: Kalibrační funkce

Synchronizace reálné a softwarové pozice je zajištěna hodnotou *relative_pos*. Při stisku tlakového senzoru je hodnota nastavena na 0, což odpovídá znaku „A“. Jednotlivé pásy je tedy nutno při kalibraci nastavit do pozic „A“.

Funkce šifrování

Šifrování je zajištěno čistě softwarově. Model demonstruje rotaci rotorů při stisku jednotlivých kláves.

Šifrování zadaného textu je realizováno pomocí funkce *enc_text()* viz výpis 5.7.

```
def enc_text(self, text):
    result = []
    for key in text:
        c = key.upper()
        result.append(self.enc(c))

    return ''.join(result)
```

Výpis 5.7: Hlavní šifrovací funkce

Funkce převádí znaky zadaného textu do velkých písmen a jednotlivě je posílá jako argumenty funkci *enc()*. Funkce *enc()*, viz výpis 5.8, simuluje průchod signálu rotory po stlačení znaku z klávesnice.

Demonstrace šifrování začíná natočením černého pásu do pozice zadaného znaku spolu se zvukovou indikací. Poté se ověří jestli není zadaný znak speciálním znakem, pokud ano je zadaný znak přeložen na znak „X“. Znak se pak porovná s mapou

propojovacího pole. Je-li zjištěn výskyt znaku v mapě, uloží se do proměnné jeho protější pár. V opačném případě není provedena žádná změna.

```
abe = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
...
def enc(self, c):
    if c not in abe:
        c='X'
    ...
    c = self.plug.plugboard[c] if c in self.plug.plugboard else c
    self.rotation()
    res = self.refle.reflect(self.rotors[0].sig(self.rotors[1].sig(
       self.rotors[2].sig(c))))
    res = self.rotors[2].sig_out(self.rotors[1].sig_out(self.rotors
       [0].sig_out(res)))
    if res in self.plug.plugboard:
            res=self.plug.plugboard[res]
            ...
```

Výpis 5.8: Funkce simulující proces šifrování jednotlivého znaku

Před samotným vstupem signálu do rotorů je provedena rotace příslušných rotorů pomocí funkce *rotation()*, viz výpis 5.9.

```
    def rotation(self):
        self.rotors[2].rotate()
        step('mM')
        co.mM.wait_while('running')
        time.sleep(1)
        if self.rotors[2].notch(1) or self.rotors[1].notch(0):
            self.rotors[1].rotate()
            step('mC')
            ...
```

Výpis 5.9: Funkce rotation() pro rotaci rotorů

Rychlý rotor s indexem 2 rotuje při každém stisku klávesy. Funkce *rotate()* inkrementuje softwarovou pozici rotoru. Funkce *step()* viz výpis 5.10 pak aktivuje motor daného pásu.

```
def step(m):
    if m == 'mA':
        co.mA.run_to_rel_pos(position_sp=-60, speed_sp=200,
           stop_action="hold")
        co.relative_pos[3] = (co.relative_pos[3] + 1 ) % 26
    ...
```

Výpis 5.10: Funkce step() pro pohyb s motory

Pokud je pravý rotor v pozici s drážkou (funkce *notch()*) je posunut i sousední rotor. Podmínka zahrnuje i situaci dvojitého kroku prostředního rotoru. K rotaci prostředního rotoru tedy dojde i v případě je-li prostřední rotor sám v pozici s drážkou.

Po rotaci rotorů je simulován průchod signálu rotory a reflektorem. Vstupní znak je postupně předáván jako argument funkci *sig()* viz výpis 5.11.

```
abe = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
...
# proměnná rot[] představuje vnitřní propojení,
#   např. rot=['EKMFLGDQVZNTOWYHXUSPAIBRCJ']
def sig(self, c):
     global abe
     In = self.rot[(abe.index(c)+self.pos-self.rs) % 26]
     Ou = abe[(abe.index(In)+self.rs-self.pos) % 26]
     return Ou
...
```

Výpis 5.11: Funkce pro průchod signálu z prava do leva

Funkce přičte k pozici daného vstupního znaku aktuální pozici rotoru (*pos*) a nastavení posunutí kroužku s drážkou (*rs*). Pozici porovná s vnitřním nastavením rotoru a následně vrací znak v podobě, v jaké vstupuje do dalšího rotoru.

Funkce reflektoru *reflect()* obdobně porovná vstupní znak s mapou vnitřního zapojení a vrací adekvátní znak.

Z výstupu reflektoru se signál opět vrací všemi rotory v opačném pořadí zpět. Průchod rotorem v opačném směru je simulován funkcí *sig_out()*. Funkce je sestavena velmi podobně jako funkce *sig()*, rozdílem je porovnání pozice znaku ve vnitřním zapojení k adekvátní pozici otevřené abecedy. Funkce rovněž vrací znak ve tvaru, ve kterém vstupuje do sousedního rotoru.

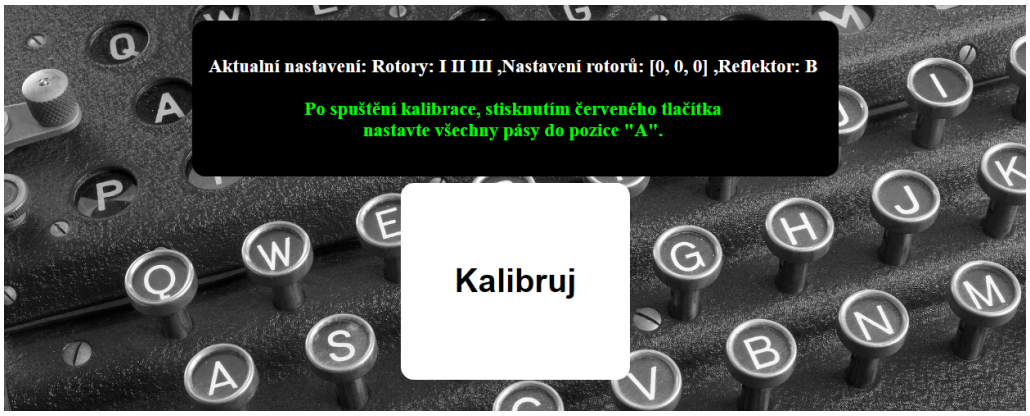
Výstupní znak rychlého rotoru je opět porovnán s propojovacím polem. Následně je černý pás natočen do pozice výstupního šifrového znaku, který je šifrovací funkcí *enc()*, viz výpis 5.8, vrácen jako její výstup.

Hlavní funkce *enc_text()*, viz výpis 5.7, nakonec spojí všechny jednotlivé znaky do jednoho slova a výsledný šifrový text vrátí jako výstup funkce. Tento šifrový text je následně předán řídící aplikací webové stránce, na které je výsledný šifrový znak zobrazen.

5.4 Uživatelské prostředí

Uživatelské prostředí bylo vytvořeno s ohledem na jeho jednoduchost a funkcionalitu. Model šifrátoru by měl sloužit k demonstraci šifrování na akcích pro širokou veřejnost. Proto bylo důležité zajistit jeho jednoduché ovládání a dát uživateli co nejmenší možnost se z procesu odchýlit.

Aplikaci tvoří 4 na sebe navazující webové stránky. První stránka zobrazuje aktuální nastavení stroje v podobě použitých komponent spolu se stručným návodem kalibrace a tlačítko umožňující spustit kalibraci viz obr.5.4. Po stisku tlačítka na



Obr. 5.4: Domovská stránka aplikace

obrazovce je spuštěna kalibrace. Jakmile je kalibrace dokončena je automaticky načtena druhá stránka.

Na druhé stránce se nachází pouze samotný formulář pro zadání hesla. Formulář je automaticky označen, není tedy nutné dále manipulovat se zobrazovací jednotkou a je možné přímé využití bluetooth klávesnice. Zadáním hesla a potvrzením stisknutí klávesy *enter* se pásy na modelu nastaví do zadaných pozic a je automaticky načtena následující stránka. Při pouhém potvrzení formuláře bez zadání hesla zůstává model v aktuálním nastavení pozic rotorů a je pouze načtena další stránka.

Třetí stránka obdobně jako stránka předchozí obsahuje pouze jeden formulář k zadání textu, který si uživatel přeje zašifrovat. Z důvodu časové náročnosti šifrování delších textů bylo pole omezeno na 20 znaků. Zadáním textu a potvrzením stisknutí klávesy *enter* je spuštěna demonstrace šifrování. Nejdříve je vysloven vstupní znak

spolu s natočením černého pásu do pozice zadaného znaku. Následně jsou natočeny příslušné rotory a nakonec je vysloven výstupní znak spolu s opětovným natočením černého pásu do pozice výstupního znaku. Tento proces je opakován pro všechny znaky zadaného textu. Po zašifrování konečného znaku daného textu je načtena poslední stránka.

Poslední stránka zobrazuje celý výstupní šifrový text a dvě tlačítka viz obr. 5.5. Stisknutím levého tlačítka je prostředí přesměrováno na stránku s možností zadat

Obr. 5.5: Konstrukce modelu šifrátoru

heslo. Stisknutí pravého tlačítka pak přesměruje prostředí na stránku s možností zadat vstupní text a umožňuje tak dále šifrovat s aktuálním nastavením pozic rotorů.

Pro opětovnou kalibraci je nutné se manuálně přesunout na domovskou stránku aplikace přepsáním adresního řádku vyhledávače. Za normálních okolností však po úvodní kalibraci není nutné model znovu kalibrovat. Výjimečný případ může nastat v nepřesném nastavení pozic pásu při prvotní kalibraci což způsobí, že při následném kroku rotorů nebude pás nastaven do správné pozice.

Jako pozadí byl vybrán obrázek volně dostupný na adrese [12].

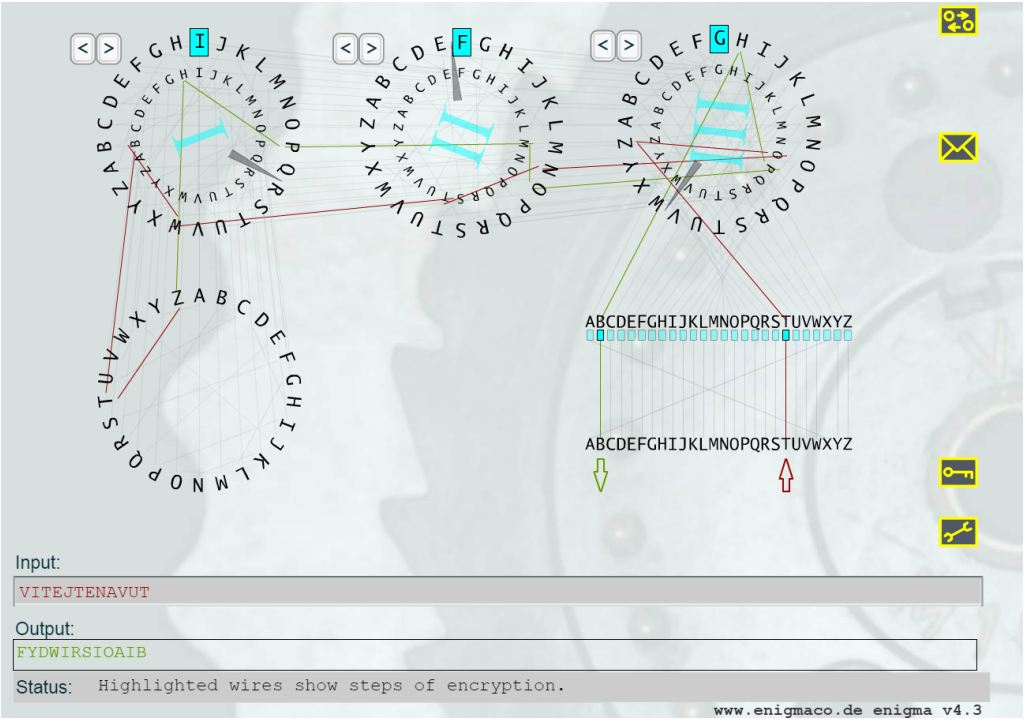
5.5 Ověření funkcionality

V rámci zpracování bakalářské práce byla pro ověření správné funkcionality modelu využita webová simulační aplikace Enigma, dostupná na adrese `http://enigmaco.de/enigma/enigma.html`.

Na obr. 5.6 je uveden příklad s následujícím počátečním nastavením stroje:

- Rotory: I - II - III bez posunutí kroužku s drážkou
- Výchozí pozice rotorů: H - D - U
- Páry v propojovacím poli: A-X

Byl zadán otevřený text „VITEJTENAVUT“.



Obr. 5.6: Webová aplikace Enigma

Výsledný šifrový text byl „FYDWIRSKVXIB“.

Obr. 5.7: Screenshot aplikace EnigmaV1

Aplikace modelu šifrátoru byla nastavena stejným způsobem. Výsledek šifry při zadání stejného otevřeného textu je možno vidět na obr. 5.7. Výsledným šifrovým textem je „FYDWIRSKVXIB“, což se shoduje s výsledkem webové simulační aplikace.

Stejným způsobem byly testovány různá nastavení. Výsledný šifrový text se vždy shodoval s předpokládaným výsledkem.

Správnost funkcionality modelu šifrátoru je tedy na základě provedených testů považována za potvrzenou.

6 Závěr

Během bakalářské práce byl sestaven teoretický podklad. Tato teoretická část byla zaměřena především na samotný šifrovací stroj Enigma. Byla představena historie vývoje různých verzí. Práce byla dále zaměřena na verzi Wermacht Enigma, kterou podrobně popisuje a rozebírá její princip funkcionality. Následně se teoretická část věnuje stručnému popisu původního modelu demonstrátoru.

Na základě rozboru funkcionality skutečného stroje a původního modelu demonstrátoru byl navržen způsob řešení zpracování.

V rámci praktické části byla upravena konstrukce původního modelu, tak aby umožnila použití klávesnice a zobrazovače. Dále byl zprovozněn a upraven speciální operační systém pro řídící jednotku Ev3dev. Pro tento systém byla pomocí programovacího jazyka Python3 a microframeworku Flask napsána webová aplikace. Aplikace umožňuje uživateli snadnou práci s modelem. Postupně dává uživateli možnost model kalibrovat, nastavit heslo a především šifrovat jakýkoli text zadaný prostřednictvím klávesnice. Následný výstupní text je poté zobrazen na obrazovce spolu s možností vrátit se na stránku, která umožňuje zadat jiné heslo nebo pokračovat v šifrování s aktuálním nastavením. Model samotný pak slouží k demonstraci principu fungování skutečného stroje Wermacht Enigma.

Výsledkem práce je fyzický model šifrátoru ze stavebnice LEGO a ověřená webová aplikace zajišťující jeho funkcionalitu. Součástí práce je také návod na obsluhu modelu.

Literatura

[1] JANČÍK, Jakub. *Demonstrátor šifrátoru z Lego Technic*Ḃrno, 2018. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií. Vedoucí práce Doc. Ing. Jan Hajný, Ph.D.

[2] CIPHER MACHINES AND CRYPTOLOGY: *Technical Details of the Enigma Machine* [online]. Dirk Rijmenants, c2004-2018 [cit. 2018-11-13]. Dostupné z: <http://users.telenet.be/d.rijmenants/en/enigmatech.htm?fbclid=IwAR3JpD9gS8hGRmGdsABL4Pf8gNGkGhtdTHhBtyo5LQhztrNe_utlBwPH4ns>

[3] SINGH, Simon. *Kniha kódů a šifer: tajná komunikace od starého Egypta po kvantovou kryptografii. 2. vyd. v českém jazyce.* Přeložil Dita ECKHARDTOVÁ, přeložil Petr KOUBSKÝ. Praha: Dokořán, 2009. Aliter (Argo: Dokořán). ISBN 978-80-7363-268-7.

[4] CHURCHHOUSE, R. F. *Codes and ciphers: Julius Caesar, the Enigma and the internet.* Cambridge: Cambridge University Press, 2002. ISBN 0-521-81054-X.

[5] REYNARD, Robert. *Secret code breaker III: a cryptanalyst's handbook.* Jacksonville Beach, FL: Smith & Daniel Marketing, c1999. ISBN 1889668133.

[6] CHRISTENSEN, Chris. *ADFGVX Cipher* [online]. Kentucky: Highland Heights, 2006 [cit. 2018-11-13]. Dostupné z: <https://www.nku.edu/~christensen/section%2010%20ADFGVX>. Northern Kentucky University.

[7] CIPHER MACHINES AND CRYPTOLOGY: *The German Enigma Cipher Machine* [online]. Dirk Rijmenants, c2004-2018 [cit. 2018-11-13]. Dostupné z: <http://users.telenet.be/d.rijmenants/en/enigma.htm>.

[8] *Crypto Museum* : Enigma [online]. 2009 [cit. 2018-11-13]. Dostupné z: http://www.cryptomuseum.com/crypto/enigma/index.htm

[9] CIPHER MACHINES AND CRYPTOLOGY: *Enigma Message Procedures* [online]. Dirk Rijmenants, c2004-2016 [cit. 2018-11-24]. Dostupné z: <http://users.telenet.be/d.rijmenants/en/enigmaproc.htm>

[10] *Ev3dev* [online]. [cit. 2018-12-03]. Dostupné z: <https://www.ev3dev.org/>

[11] *Flask: Web development, one drop at a time* [online]. c2010-2019 [cit. 2019-04-28]. Dostupné z: <http://flask.pocoo.org/>

[12] *Pixabay* [online]. c2019 [cit. 2019-04-28]. Dostupné z: <https://pixabay.com/photos/rotor-cipher-machine-enigma-1147801/?fbclid=IwAR2n-G8Hl8lOcJqfmLGfRj69FgJNQ_ASXUTaHrRfrnjaUSlN4vPQDDyDnyc>

[13] Setting up VS Code. *EV3 Pythonv2* [online]. [cit. 2019-05-15]. Dostupné z: <https://sites.google.com/site/ev3devpython/setting-up-vs-code>

[14] *EV3 Python* [online]. [cit. 2019-05-15]. Dostupné z: <v1.ev3python.com>

[15] HEMPEL, Ralph. Python language bindings for ev3dev. *Python-ev3dev* [online]. c2015 [cit. 2019-05-15]. Dostupné z: <https://ev3dev-lang.readthedocs.io/projects/python-ev3dev/en/2.0.0beta1/>

[16] Ev3dev language bindings. *Ev3dev-lang* [online]. c2016 [cit. 2019-05-15]. Dostupné z: <https://ev3dev-lang.readthedocs.io/en/latest/index.html>

Seznam symbolů, veličin a zkratek

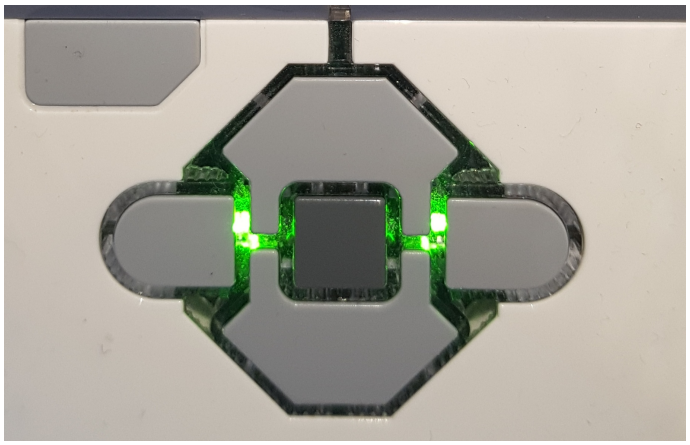
CSS	Kaskádové styly – Cascading Style Sheets
HTML	Značkovací jazyk pro tvorbu webových stránek– Hypertext Markup Language
IP	Internetový protokol – Internet Protocol
PHP	Hypertextový preprocesor – Hypertext Preprocessor
Python CGI	Modul umožňující tvorbu webových stránek pomocí jazyka Python – Common Gateway Interface
SD	Paměťová karta – Secure Digital
USB	Univerzální sériová sběrnice – Universal Serial Bus
Wifi	komunikační standard pro bezdrátový přenos dat – Wireless Fidelity

Seznam příloh

A Návod na obsluhu modelu

A.1 Spuštění modelu

Model je spuštěn stiskem středového tlačítka řídící jednotky, která se nachází v levé části modelu.



Obr. A.1: Ovládací rozhraní řídící jednotky

Úplné načtení systému je signalizováno svitem zelené diody viz obr. A.1.

A.2 Navázání spojení

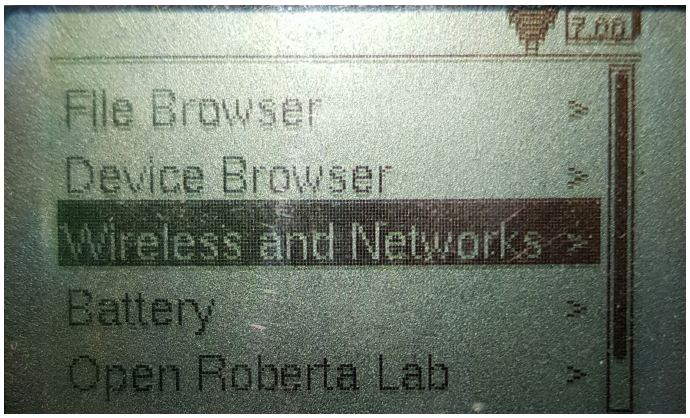
Na přiloženém tabletu se nachází již nastavený hotspot *LegoEnigma*, po jeho spuštění se řídící jednotka automaticky s tabletem spojí.

V případě, že si přejete model spojit s jiným zařízením pokračujte podle návodu níže. Řídící jednotku je možno s vybraným zobrazovacím zařízením spojit pomocí Bluetooth nebo Wifi.

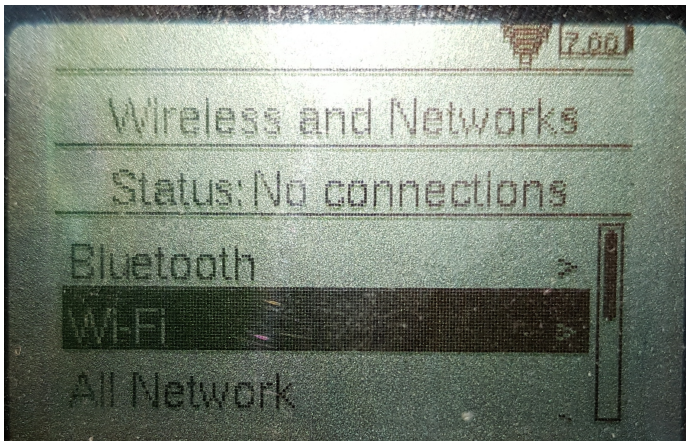
A.2.1 Navázání spojení pomocí Wifi

Po spuštění řídící jednotky je na obrazovce řídící jednotky zobrazeno ovládací menu. Pomocí směrových tlačítek řídící jednotky zvolte možnost *Wireless and network* viz obr. A.2 a potvrďte stiskem středového tlačítka.

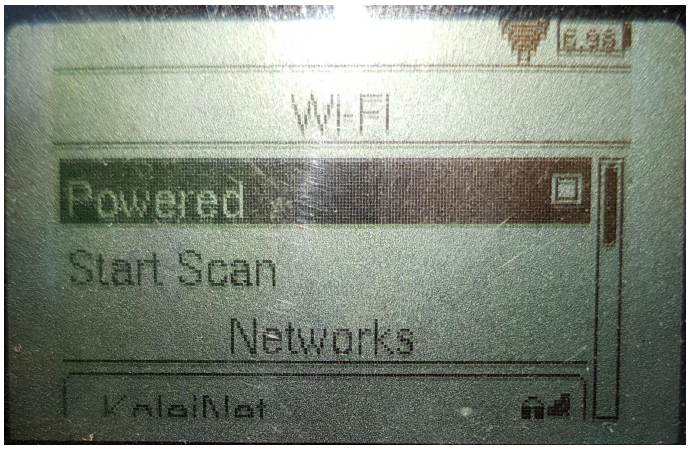
Pro připojení pomocí Wifi zvolte možnost *Wifi* viz. obr A.3 a potvrďte.
Pokud není možnost *Powered* zaškrtlá viz. obr A.4, potvrďte středovým tlačítkem. Následně zvolte možnost *Start Scan*. Zařízení začne vyhledávat dostupné bezdrátové sítě a jejich názvy zobrazí v seznamu níže.



Obr. A.2: Výchozí menu systému EV3dev

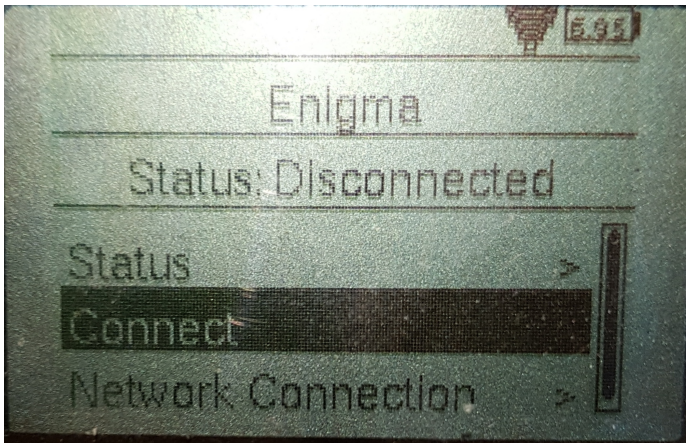


Obr. A.3: Menu sytému EV3dev - Wireless and network



Obr. A.4: Menu sytému EV3dev - Wifi

Ze seznamu vyberte požadovanou bezdrátovou síť a potvrďte. Bude zobrazeno menu dané sítě, zde zvolte možnost *Connect* viz obr. A.5.

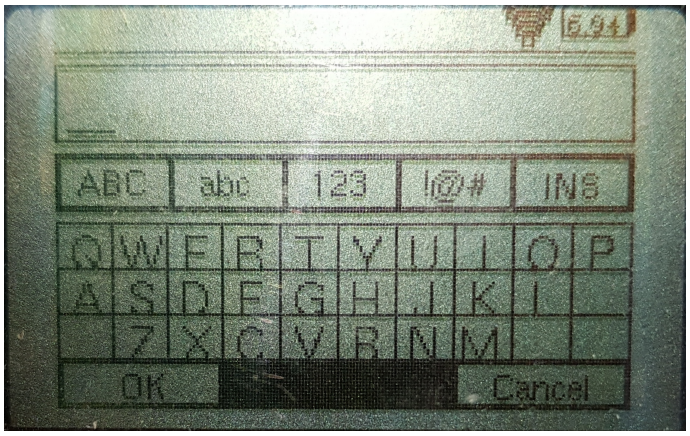


Obr. A.5: Zvolená síť

Zobrazí se požadavek na zadání hesla viz obr. A.6, stiskem středového tlačítka potvrďte.

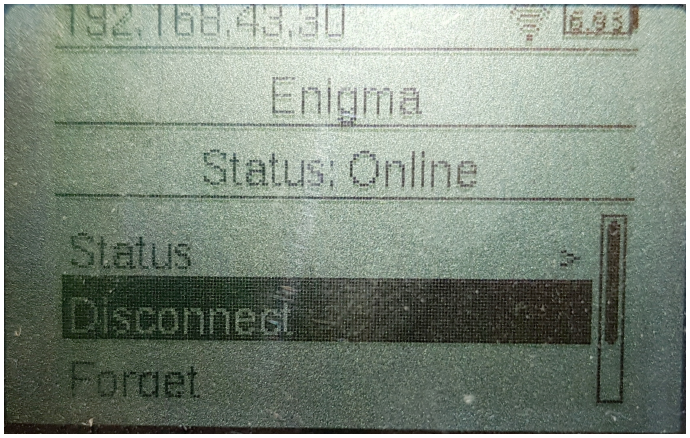
Obr. A.6: Zadání hesla

Potvrzením je zobrazena systémová klávesnice viz obr. A.7. Po klávesnici se navigujte pomocí směrových tlačítek, pro potvrzení znaku stiskněte středové tlačítko, pro smazání posledního znaku stiskněte tlačítko „zpět“ (samostatné tlačítko pod levým rohem obrazovky). Po zadání hesla potvrďte tlačítkem „ok“ na systémové klávesnici.



Obr. A.7: Systémová klávesnice

Potvrzením zadaného hesla začne jednotka navazovat spojení. Na obrazovce se v poli status zobrazí nejprve *Associating*, následně *Configuring*. Úspěšné navázání spojení signalizuje text *Online* v poli status a přidělená IP adresa v levém horním rohu obrazovky viz obr. A.8.



Obr. A.8: Úspěšné navázání spojení

Prvním připojením se heslo dané sítě uloží a při následujícím startu řídící jednotky stačí danou síť pouze spustit a jednotka automaticky naváže spojení. V některých případech je-li bezdrátová síť spuštěna před spuštěním řídící jednotky není spojení automaticky navázáno. V tomto případě se stačí v menu přesunout do *Wireless and Network*, následně *Wifi* a zvolit *Start Scan*. Systém by měl po vyhledání požadované sítě automaticky navázat spojení.

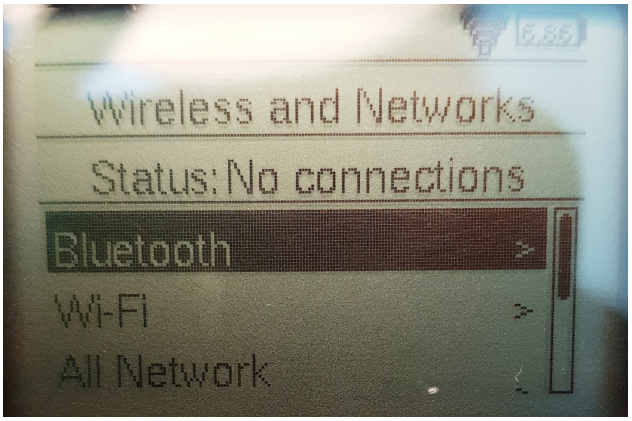
A.2.2 Navázání spojení pomocí Bluetooth

Spojení pomocí bluetooth je ve verzi systému ev3-dev-stretch-ev3-generic-2018-08-06 velmi nestabilní. V některých případech není při startu systému modul bluetooth vůbec načten a je potřeba celý systém restartovat.

Důležitým prvkem pro úspěšné navázání spojení je *Sdílení připojení Bluetooth*, na některých zařízeních je nutno zapnout před samotným navázáním spojení.

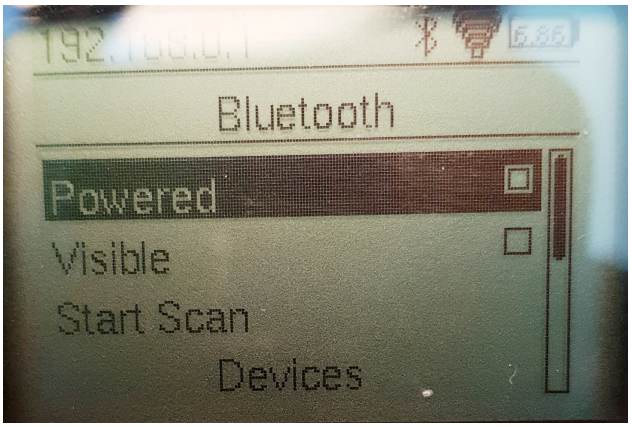
Pro navázání spojení pomocí Bluetooth vyberte v hlavním menu možnost *Wireless and Networks* viz obr. A.2.

Následně vyberte možnost Bluetooth viz obr. A.9. Pokud je po vybrání možnosti zobrazeno *No Bluetooth available.* je nutné celý systém restartovat.



Obr. A.9: Výchozí menu systému EV3dev - Bluetooth

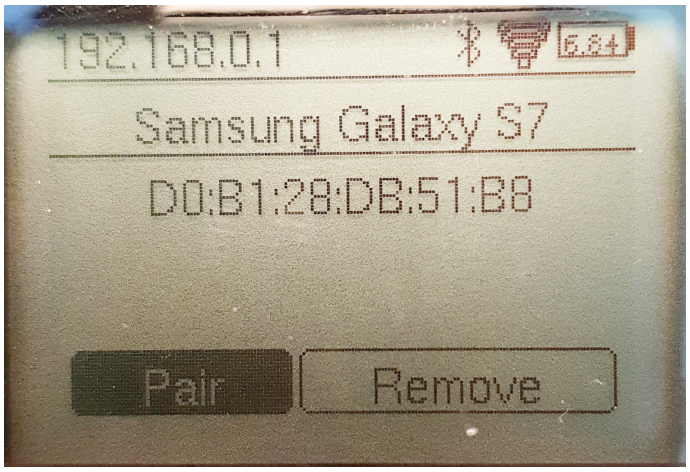
Pokud není zaškrtlá možnost *Powered*, potvrďte stiskem středového tlačítka viz obr. A.10.



Obr. A.10: Nabídka Bluetooth

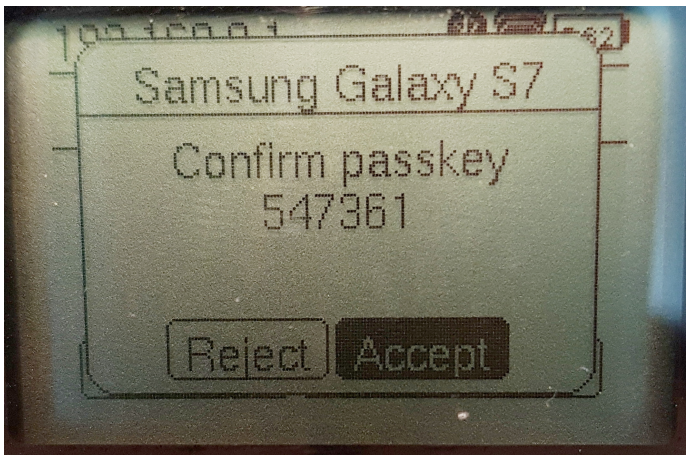
Pokračujte výběrem možnosti *Start Scan*. Systém vyhledá viditelná zařízení a zobrazí je v seznamu níže.

Vyberte vaše zařízení a potvrďte stiskem středového tlačítka. Bude zobrazena nabídka párování s daným zařízením viz obr. A.11.



Obr. A.11: Bluetooth párování

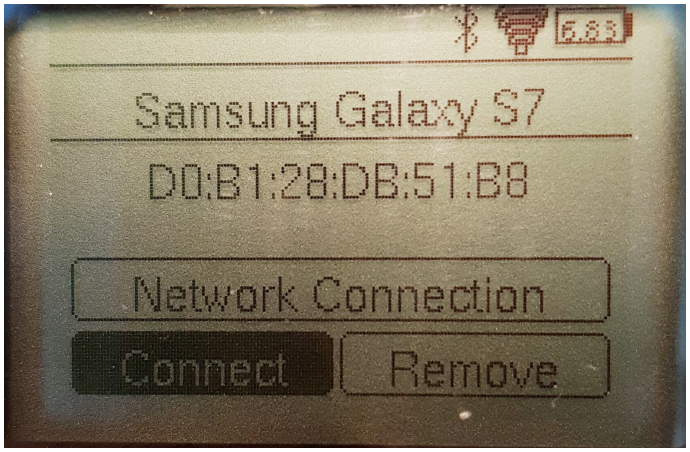
Potvrzením stiskem středového tlačítka je zobrazena výzva s ověřovacím kódem viz obr. A.12. Podobná výzva by se měla objevit na obrazovce cílového zařízení. Výzvu na obou zařízeních potvrďte.



Obr. A.12: Bluetooth párování - potvrzení

U některých zařízení se může výzva lišit. Na obrazovce řídící jednotky může být zobrazeno pole s heslem. Výchozí heslo je nastaveno na *1234*. V tomto případě stačí potvrdit stiskem středového tlačítka. Na obrazovce cílového zařízení by se pak měla objevit výzva k zadání hesla. Zadejte heslo z obrazovky řídící jednotky a potvrďte.

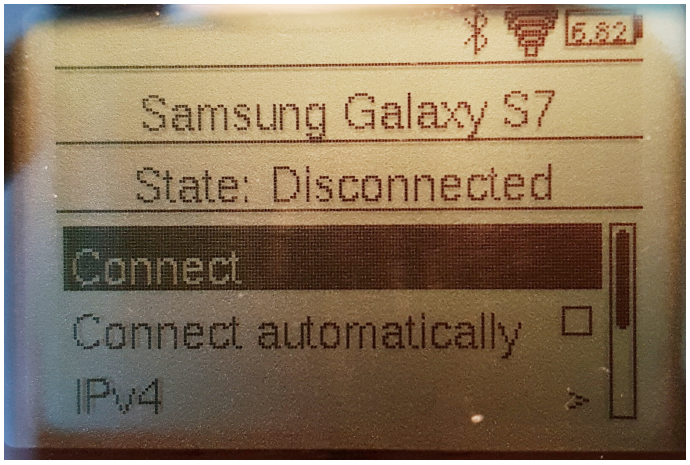
Po úspěšném spárování je zobrazena nabídka daného zařízení viz obr. A.13.



Obr. A.13: Bluetooth úspěšné spárování

Možnost *Connect* v úvodní nabídce bohužel není ve verzi systému ev3-dev-stretch-ev3-generic-2018-08-06 funkční. Je nutné z nabídky zvolit *Network Connection*.

V rozšířené nabídce Bluetooth viz obr. A.14 vyberte možnost *Connect*.



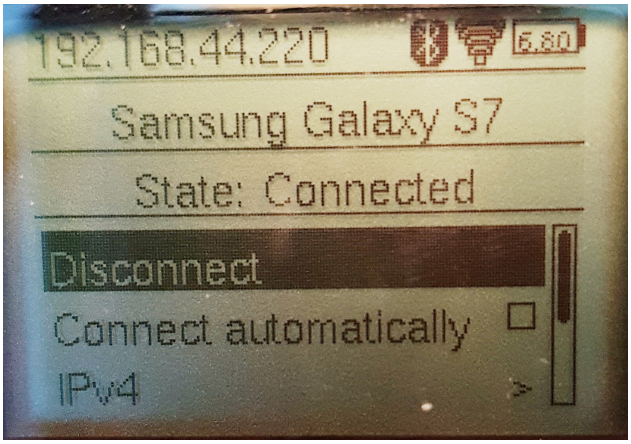
Obr. A.14: Rozšířené nastavení

Pokud na vybraném zařízení nebylo dříve spuštěno *Sdílené připojení Bluetooth* zobrazí se žádost o povolení služby. Žádost potvrďte. Některá zařízení žádost o povolení automaticky nezobrazí, u takovýchto zařízení je nutné spustit sdílení před samotným navázáním spojení. Na obrazovce řídící jednotky bude v případě pozdního povolení *Sdíleného připojení Bluetooth* zobrazena chyba. Chybovou hlášku potvrďte a znovu vyberte možnost *Connect*.

Jak již bylo zmíněno dříve spojení Bluetooth je velmi nestabilní a chybová hláška se může zobrazit i po opětovném pokusu o spojení. V některých případech může

k úspěšnému připojení dojít až při pátém pokusu. Pokud problém přetrvává zařízení restartujte a postup zopakujte.

Úspěšné propojení je indikováno změnou *State* na *Connected* a zobrazením IP adresy v levém horním rohu obrazovky viz obr. A.15.



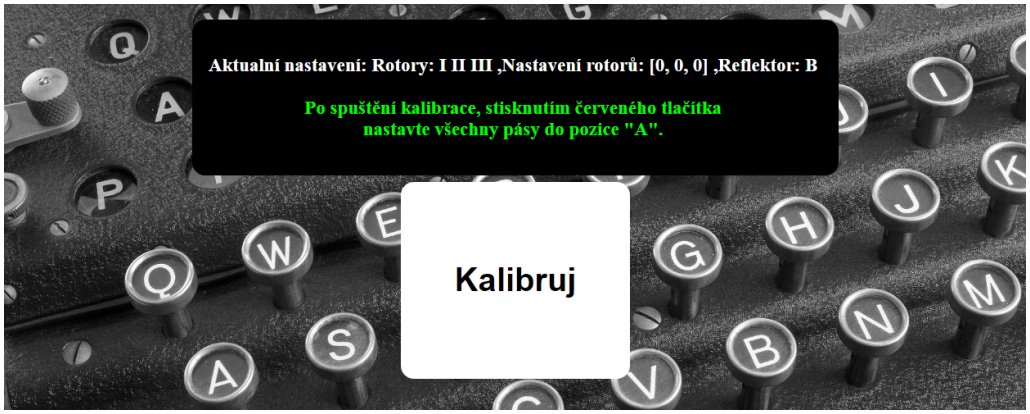
Obr. A.15: Úspěšné navázání Bluetooth spojení

A.3 Ovládání webová aplikace

Na tabletu (jiném zobrazovacím zařízení) spusťte webový prohlížeč. Do adresního řádku zadejte IP adresu viditelnou v levém horním rohu na obrazovce řídící jednotky spolu s portem 5000. V případě adresy přidělené na obrázku A.8:

192.168.43.30:5000

Po načtení se zobrazí úvodní stránka viz obr. A.16.



Obr. A.16: Domovská stránka aplikace

Stiskem tlačítka *Kalibruj* se spustí kalibrace. Postupně se začnou otáčet pásy modelu. Stiskem červeného tlakového senzoru na modelu se pás zastaví a začne se otáčet pás následující. Všechny páse je nutné nastavit do pozice „A“.

Ukončení kalibrace je indikováno zvukovým signálem a následně je načtena následující stránka umožňující zadat heslo viz obr. A.17.

Obr. A.17: Stránka pro zadání hesla

Heslo je ve tvaru *AAA* a jeho zadáním se pásy nastaví do zadaných pozic. Levý šedý pás je nastaven do pozice prvního znaku, prostřední do pozice druhého znaku a pravý pás pak do pozice třetího znaku. Zadáním méně než 3 znaků jsou nastaveny pouze příslušné pásy. Pouhým potvrzením bez zadání hesla zůstávají pásy v aktuálních pozicích.

Nastavení pásů do zadaných pozic je opět indikováno zvukovým signálem a je načtena stránka umožňující zadat text viz obr. A.18.

Obr. A.18: Stránka pro zadání textu

Zde je možné zadat vstupní text s maximální možnou délkou 20 znaků. Zadané číslice, mezery a speciální znaky jsou šifrovány jako znak „X“. Potvrzením začne proces šifrování. Nejprve je černý pás nastaven do pozice vstupního znaku, následně se natočí příslušné pásy demonstrující rotaci rotorů. Výstupní znak je poté znovu zobrazen na černém pásu. Tento proces se opakuje pro všechny znaky zadaného textu.

Po ukončení demonstrace je načtena stránka zobrazující výstupní šifrový text viz obr. A.19. Na stránce se rovněž nachází 2 tlačítka.

Obr. A.19: Zobrazení výstupní šifrového textu

Stiskem tlačítka *Změn heslo* se aplikace přesměruje na stránku umožňující zadat heslo. Stiskem tlačítka *Pokračuj* je zobrazena stránka umožňující znovu zadat vstupní text. V tomto případě je dále šifrováno s aktuálním nastavením pásů (rotorů).

A.4 Přístupové údaje

Systém Ev3dev nahraný na microSD kartě může být dále upraven. Po úspěšném spojení řídící jednotky s počítačem je možné se k systému připojit pomocí terminálu. Přihlašovací údaje jsou:

Username: robot
Passwoord: maker

Po odebrání microSD karty z řídící jednoty při vypnutém stavu, je při následujícím spuštění jednotky načten originální systém LEGO.

B Užitečné odkazy

Pro návrh aplikací pro systém Ev3dev v jazyce Python je vhodné použít vývojové prostředí Microsoft Visual Studio Code. Visual Studio poskytuje rozšíření, které umožňuje přímé stažení vytvořených aplikací do řídící jednotky a přímý přístup k souborům v domovském adresáři. Návod na zprovoznění je dostupný na webu *EV3 Pythonv2* viz. [13].

Webové stránky *EV3 Pythonv2* poskytují i další užitečné návody na práci se systémem Ev3dev, některé funkce jsou však lépe popsány na starší verzi webu *EV3 Python* viz. [14].

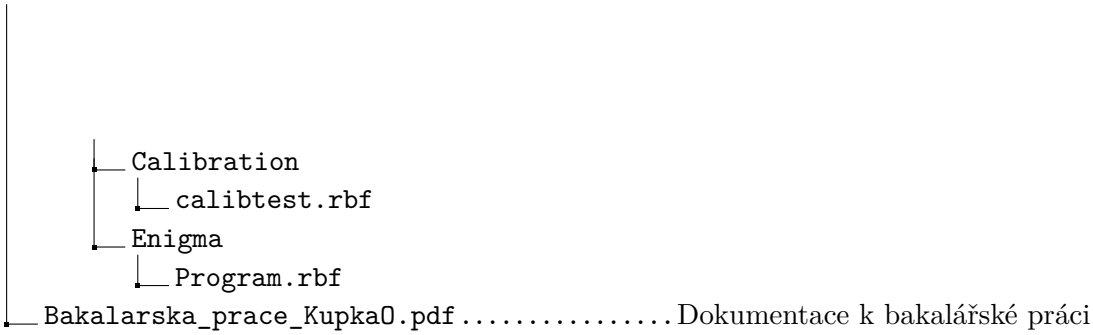
Oficiální dokumentace knihovny je dostupná na stránce *Python language bindings for ev3dev* viz. [15], případně na *Ev3dev language bindings* viz. [16].

C Obsah přiloženého CD

- / ..... kořenový adresář přiloženého CD
  - fotodokumentace
    - EnigmaLEGO_aktualni ..... Fotografie aktuálního modelu
      - Motor_detail
      - Pas
      - Pas_horni_obnazen
      - Pas_pravy_obnazen
      - Pas_pravy_obnazen_detail
      - Pasy_detail
      - Pohled_shora
      - Pohled_shora_s_tabletem
      - Pohled_zboku
      - Pohled_zepredu_bez
      - Pohled_zepredu_bez_klavesnice
      - Pohled_zepredu_detail
      - Pohled_zepredu_s
      - Pohled_zepredu_s_klavesnici
      - Pohled_zespod
      - Pohled_zezadu
      - Pohled_zezadu_detail
      - Prava_cast_detail
      - Prava_cast_zezadu
      - Ridici_jednotka_zepredu
      - Ridici_jednotka_zepredu_2
      - Ridici_jednotka_zezadu
      - Ridici_jednotka_zezadu_2
      - Stojan_detail
    - EnigmaLEGO_puvodni ..... Fotografie původního modelu Bc.Jakuba Jančíka
      - Leva_zadni_cast
      - Okenka
      - Pas
      - Pas_levy_detail
      - Pas_levy_zepredu
      - Pas_levy_zezadu
      - Pas_levy_zezadu_2
      - Pas_odhalen
      - Pas_odhalen_detail
      - Pas_odhalen_zezadu
      - Pas_pravy_zezadu
      - Pas_pravy_zezadu_shora
      - Pasy_detail
      - Pasy_detail_2
      - Pasy_detail_3
      - Pasy_odhaleny_detail

- Pasy_zezadu_detail
- Pohled_na_pasy
- Pohled_shora
- Pohled_zboku
- Pohled_zboku_2
- Pohled_zbokuv_3
- Pohled_zboku_zespod
- Pohled_zepredu
- Pohled_zepredu_2
- Pohled_zespod
- Pohled_zespod_detail
- Pohled_zezadu
- Pohled_zezadu_2
- Pohled_zezadu_shora
- Prava_cast_zepredu
- Prava_cast_zespod_zezadu
- Prava_predni_cast_detail
- Prava_predni_cast_odhalena
- Pravy_pas_zepredu_detail
- Predni_konstrukce
- Ridici_jednotka
- Uchyceni_ridici_jednotky

- Webova_aplikace ..... Soubory webové aplikace v dané souborové struktuře
  - Enigma
    - Config
      - __init__.py
      - config.py ..... Konfigurační modul
    - Controler
      - __init__.py
      - enigmaV4.py ..... Ovládací modul
    - static
      - bg1.jpg ..... Obrázek pozadí stránky
      - styles
        - style.css ..... Kaskádové styly
    - templates
      - cipher.html ..... Stránka pro zadání textu
      - index.html ..... Úvodní stránka
      - output.html ..... Výsledná stránka
      - password.html ..... Stránka pro nastavení hesla
    - enigma.py ..... Řídící soubor
  - launch.sh ..... Spouštěcí script
  - README.txt ..... Návod na instalaci
- Zálohy
  - Konzolova_aplikace ..... Program semestrální práce
    - enigmaV1.py
  - Puvodni_program ..... Původní program Bc.Jakuba Jančíka


Calibration
calibtest.rbf
Enigma
Program.rbf
Bakalarska_prace_Kupka0.pdf
Dokumentace k bakalářské práci