

Středoškolská technika 2025

Setkání a prezentace prací středoškolských studentů na ČVUT

Automatizace ovládání tabulky GRAFO

Tomáš Mészároš

Lauderovy školy Belgická 25, Praha 2

Abstrakt

Práce se zabývá automatizací ovládání kreslící tabulky Grafo pomocí krokových motorů řízených mikroprocesorem Arduino. Dále je cílem vytvořit uživatelsky přívětivé rozhraní umožňující laikovi tabulku automatizovaně ovládat.

V prvních kapitolách práce jsou představeny a přiblíženy hlavní komponenty použité v práci, tedy tabulka Grafo, Arduino a krokové motory. Následně je detailně popsána provedená práce s elektronikou a programováním. Jsou uvedeny ukázky uživatelského rozhraní a také video ukazující tabulku během procesu vykreslování.

Hotové řešení umožňuje uživateli na tabulce automatizovaně zakreslovat hranice vybraných států, průběhy mocninných funkcí a také poskytuje uživateli možnost vytvořit si svůj vlastní náčrt.

Klíčová slova

Grafo, Arduino, krokový motor, Python, automatizace, uživatelské rozhraní

Obsah

Úvod			
1. Tabulka Grafo			
2. Ardu	1 ino		
2.1.	Historie7		
2.2.	Software – Arduino IDE		
2.3.	Připojení periferie		
3. Krokový motor			
4. Zapojení zařízení			
4.1.	Zapojení elektroniky		
4.2.	Propojení mechanických součástí 11		
4.3.	Front-end v Pythonu s Arduinem		
4.3.1	. Použité knihovny Pythonu		
4.3.2	2. Vytvoření uživatelského rozhraní		
4.3.3	8. Vykreslení hranic států		
4.3.4	Vykreslení průběhu matematických funkcí 15		
4.3.5	5. Možnost volné kresby		
4.3.6	5. Architektura propojení		
5. Disk	use		
Závěr			
Zdroje			
Seznam příloh Chyba! Záložka není definována.			

Úvod

Tabulka Grafo je ikonické zařízení, které se stalo od svého uvedení na trh (1960)¹ populární hračkou. Na první pohled se velmi jednoduše ovládá, ale nakreslit na ní cokoliv složitějšího (např. i rovné šikmé čáry) je poměrně náročné. Proto by bylo přínosné automatizovat její ovládání a tím zpřístupnit kreslení složitějších obrazců.

Prvním cílem této práce je tedy právě automatizace ovládání tabulky Grafo, a to konkrétně pomocí krokových motorů řízených mikroprocesorem Arduino Nano. Arduino Nano lze však ovládat jen specifickým programovacím jazykem. Aby bylo umožněno využít tuto automatizaci i uživatelům neovládajícím programování, bude potřeba vytvořit uživatelské rozhraní umožňující zadávání instrukcí pro Arduino Nano přívětivějším způsobem. Druhým cílem práce bude tedy vytvoření uživatelsky přívětivého rozhraní, které dovolí uživateli bez znalostí programování a elektroniky tabulku dálkově ovládat. Tento front-end² bude napsán v programovacím jazyku Python, s využitím knihoven jako *matplotlib* a *pyserial*. S jejich pomocí bude uživatel schopen na soustavě souřadnic vytvářet obrázky a poté je odesílat do Arduina.

Očekávané komplikace, které mohou vyvstat při vyhotovování praktické části práce, se mohou týkat zejména mechanických omezení tabulky Grafo přímo plynoucích z její konstrukce. Právě kvůli specifické konstrukci, která byla navržena přímo pro manuální ovládání, nelze s jistotou vyloučit potenciální problémy. Jsou spojeny hlavně s možnými nepřesnostmi způsobenými přenosem složitějších obrazců na mechanický pohyb otáčivých knoflíků, respektive na pohyb kreslícího pera tabulky. Mohou být zapříčiněny například setrvačností pohybu knoflíků nebo pera tabulky, nedodržení úhlu linky (způsobené nedokonalou synchronizací obou knoflíků) nebo ku příkladu nepřesností v délkách linky (zapříčiněné mechanickým odporem působícím na pero). Kumulací těchto faktorů může docházet k výrazné deformaci přenášených obrazců.

Práce je pojata primárně jako technický projekt opírající se o fungování svých hardwarových, softwarových a mechanických součástí. Z její povahy tedy plyne, že použité zdroje jsou spíše technické dokumentace a uživatelské manuály nežli vědecké články a publikace nebo teoretické koncepty.

V rámci vytvoření uživatelského rozhraní jsou jako hlavní zdroje využity dokumentace knihoven programovacího jazyka Python (viz kapitola Použité knihovny Pythonu) detailně popisující všechny funkce v knihovnách zabudované.

Pro účely napsání teoretické části práce byla vybrána publikace Arduino³ popisující historii projektu Arduino a jeho využití a publikace Stepping Motors: A Guide to Theory and Practice⁴ popisující fungování krokových motorů.

Při navrhování výtisku násad na otáčivé knoflíky tabulky Grafo byla využita data o rozměrech krokového motoru 24byj-48.⁵

V případě samotného programování, ať už Pythonu nebo v Arduinu nebyla potřebná dodatečná studijní literatura, protože autor již měl v tomto ohledu dostatečné dovednosti.

¹ Wikipedia. (2024). Etch A Sketch. Wikipedia, The Free Encyclopedia. Dostupné z: <u>https://en.wikipedia.org/wiki/Etch_A_Sketch</u>. [cit. 2025-03-08].
²část systému viditelná pro uživatele, sloužící k přímé interakci

³ SELECKÝ, Matúš. Arduino. Vyd. 1. Computer Press, 2016. ISBN 978-80-251-4840-2.

⁴ Atherton, D. P., & Irwin, G. W. (1992). Stepping motors: A guide to theory and practice (4th ed.). IET. Online. Dostupné z <u>https://nirmt.com/storage/uploads/E-BOOK_BE-</u>

AUTOMOBILE/Stepping%20motors%20%20a%20guide%20to%20theory%20and%20practice.pdf

⁵ Maintex. (2025). 24BYJ48 Stepper Motor. Maintex Motors. Dostupné z: https://maintexmotors.com/product/24byj48-stepper-motor/

1. Tabulka Grafo

Tabulka Grafo (viz Obrázek 1) je mechanická kreslící hračka vytvořená André Cassagnfesem na konci 50. let.⁶ Stala se celosvětově prodávanou a její popularita jí do jisté míry vydržela až dodnes. V této kapitole se práce bude věnovat zejména jejímu fungování, tedy tomu, jak se z pohybu rotace knoflíků dostaneme ke kreslení úseček s požadovanou směrnicí a délkou.

Jedná se o plastovou tabulku (nejčastěji v červené barvě) s průhlednou obrazovkou vyrobenou z plexiskla, která má po stranách dva otočné knoflíky sloužící k ovládání vnitřního mechanismu. Tento mechanismus umožňuje kreslení na obrazovku. Uvnitř tabulky se nachází velmi jemný hliníkový prášek, který se přirozeně přichytí na vnitřní stranu obrazovky. Tento prášek tvoří rovnoměrnou vrstvu, která slouží jako podklad pro kreslení.





Spolu s práškem jsou uvnitř tabulky malé skleněné kuličky, které pomáhají prášku, aby se volně pohyboval a nerozpadal se na shluky.⁸

Kreslení je zajištěno hrotem, připevněným na dvě navzájem kolmé kolejnice. Ovládací knoflíky jsou přímo spojeny s těmito kolejnicemi (viz Obrázek 2). Jeden knoflík posouvá hrot vodorovně, zatímco druhý ovládá jeho svislý pohyb. Při otáčení knoflíky se hrot posouvá po vrstvě hliníkového prášku na vnitřním povrchu obrazovky a setře ho v místě svého pohybu, čímž vzniká nepřerušovaná čára. Při současném pohybu knoflíků lze docílit nakreslení přímky s jakoukoliv směrnicí.⁹

⁶ COOPEE, Todd. *Etch A Sketch* [online]. Dostupné z: <u>https://toytales.ca/etch-a-sketch-from-ohio-art-company-1960/</u>. [cit. 2025-03-08].

⁷ Wikipedia. (2024). *Etch A Sketch*. Wikipedia, The Free Encyclopedia. Dostupné z: <u>https://en.wikipedia.org/wiki/Etch A Sketch</u>. [cit. 2025-04-07].

⁸ Shaw, A. W. Etch A Sketch Toy. US3055113. 18. 9. 1962. Dostupné z:

https://patentimages.storage.googleapis.com/67/4f/71/27e699d9875e3d/US3055113.pdf [cit. 2025-03-08].

⁹ Shaw, A. W. Etch A Sketch Toy. US3055113. 18. 9. 1962. Dostupné z:

https://patentimages.storage.googleapis.com/67/4f/71/27e699d9875e3d/US3055113.pdf [cit. 2025-03-08].



*Obrázek 2 - Schéma vnitřního mechanismu tabulky Grafo. Otačením knoflíku A se rozpohybují kolečka B, C, D a E a vlákno, na které je připevněna kolej F na sobě nesoucí hřídlo s hrotem.*¹⁰

Kromě kreslícího mechanismu je tabulka vybavena také systémem mazání. Když chce uživatel obrazovku vyčistit, stačí tabulku otočit a zatřepat s ní. Prášek se rozprostře po celé ploše obrazovky a znovu pokryje všechny oblasti, které byly hrotem dříve setřeny.

¹⁰ Wikipedia. (2024). *Etch A Sketch*. Wikipedia, The Free Encyclopedia. Dostupné z: <u>https://en.wikipedia.org/wiki/Etch A Sketch</u>. [cit. 2025-04-07].

2. Arduino

Arduino je značka mikroprocesorů, které je možné programovat i bez detailních znalostí jejich architektury a logiky celého systému.¹¹ Poskytuje uživatelské rozhraní, které se stará o procesy na pozadí (například řízení paměti). Jedná se o nástroj vhodný pro laické domácí projekty.

2.1. Historie

Příběh Arduina začíná v roce 2003, když italský univerzitní student Hernando Barragán chtěl v rámci své diplomové práce vytvořit nástroj, který by ulehčil práci s elektronikou umělcům a designérům.¹² Vytvořil zařízení s názvem Wiring, které bylo poměrně úspěšné. V roce 2005 se z projektu Wiring odpojilo několik Hernandových spolupracovníků a založili vlastní projekt s názvem Arduino, který úspěšně komercializovali. Nyní existuje velké množství modelů Arduina. V tomto projektu je využíván konkrétně model Arduino Nano (viz Obrázek 3), který se začal vyrábět v roce 2008.¹³

Obrázek 3 - model Arduino Nano¹⁴



¹¹ SELECKÝ, Matúš. (2016). Arduino. Computer Press. ISBN 978-80-251-4840-2. s.16.

¹² SELECKÝ, Matúš. (2016). Arduino. Computer Press. ISBN 978-80-251-4840-2. s.16.

¹³ Wikipedia contributors. Arduino Nano. In: Wikipedia, The Free Encyclopedia [online]. 2025. Dostupné z: <u>https://en.wikipedia.org/wiki/Arduino_Nano</u>. [cit. 2025-03-08].

¹⁴ Wikimedia Commons. (2024). *Arduino Nano isometr* [obrázek]. Wikimedia Commons. Dostupné z: <u>https://commons.wikimedia.org/wiki/File:Arduino nano isometr.jpg</u>. [cit. 2025-04-07].

2.2. Software – Arduino IDE

Pro programování mikroprocesorů Arduino je možné použít několik populárních uživatelských rozhraní. V tomto projektu je využíván konkrétně Arduino IDE (viz

Obrázek 4). Je to open-source rozhraní napsané v jazyce Java umožňující uživateli psát a nahrávat kód do paměti Arduina.¹⁵ Programování zde probíhá v jazyku C++ s vestavěnými knihovnami. Arduino IDE poskytuje uživateli plnohodnotnou podporu od psaní kódu, přes jeho kompilaci až po jeho odeslání do Arduino mikroprocesoru.

Obrázek 4 - Ukázka kódu v prostředí Arduino IDE, příloha s celým kódem - Chyba! Nenalezen zdroj odkazů., zdroj – archiv



2.3. Připojení periferie

K Arduinu lze připojit periferie přes takzvané piny (specializované konektory). Dva nejčastěji využívané druhy pinů jsou digitální piny a analogové piny. U digitálních pinů lze regulovat napětí binárně (ON/OFF) a u analogových pinů lze číst hodnotu napětí (využívají se například s potenciometry). V tomto projektu jsou využity pouze digitální piny, a to celkem osm. Jako periferie jsou v projektu využity dva krokové motory (pro každý knoflík tabulky jeden motor), z nichž každý využívá čtyři piny, které ovládají cívky v motoru.

¹⁵ SELECKÝ, Matúš. (2016). Arduino. Computer Press. ISBN 978-80-251-4840-2. s.55.

3. Krokový motor

Krokové motory jsou speciální elektromotory navržené pro přesné řízení polohy a rychlosti. Jejich hlavní výhodou je schopnost rozdělit plnou otáčku na několik kroků, což umožňuje přesné ovládání.

Existují dva hlavní typy krokových motorů: bipolární a unipolární. Bipolární motory jsou navrženy tak, že pro své ovládání vyžadují změnu směru proudu. Kvůli tomu se složitěji řídí, ale poskytují vyšší točivý moment než unipolární motory. Unipolární motory mají každé vinutí rozdělené na dvě části se společným středovým vývodem. Tento model umožňuje jednodušší řízení, protože změna směru proudu není nutná, stačí aktivovat odpovídající polovinu vinutí (viz Obrázek 5).





Konkrétně model 24BYJ-48 (viz Obrázek 6), který je využit v tomto projektu, je motor unipolární. Jednou z důležitých vlastností motoru 24BYJ-48 je jeho schopnost dosáhnout vysokého rozlišení krokování. Hřídlo motoru jsme schopni mikroprocesorem ovládat až na úrovni 1/2048 jedné otáčky, tedy přibližně velikost úhlu 10° 33°. Díky tomu jsme schopni vyvinout velkou přesnost hodící se na tento projekt¹⁷.

¹⁶ Maintex. (2025). 24BYJ48 Stepper Motor. Maintex Motors. Dostupné z:

https://maintexmotors.com/product/24byj48-stepper-motor/. [cit. 2025-04-07].

¹⁷ Atherton, D. P., & Irwin, G. W. (1992). *Stepping motors: A guide to theory and practice* (4th ed.). IET. Online. Dostupné z <u>https://nirmt.com/storage/uploads/E-BOOK_BE-AUTOMOBILE/Stepping%20motors%20%20a%20guide%20to%20theory%20and%20practice.pdf.</u>



Obrázek 6 - Model krokového motoru 24byj-48¹⁸

¹⁸ Maintex. (2025). 24BYJ48 Stepper Motor. Maintex Motors. Dostupné z: <u>https://maintexmotors.com/product/24byj48-stepper-motor/</u>. [cit. 2025-04-07].

4. Zapojení zařízení

Pro ovládání tabulky Grafo pomocí Arduina bylo potřeba správně zapojit jednak hardware elektroniky, ale také mechanické části, které drží vše pohromadě.

4.1. Zapojení elektroniky

Zapojení elektroniky se dělilo na několik částí. Bylo to jednak napojení krokových motorů 28byj-48 ke zdroji, dále jejich připojení k Arduinu Nano, a nakonec připojení Arduina Nano k počítači, ze kterého získává pokyny.

Pokud jde o napájení motorů, byl využit 2A zdroj s maximem 12 V (motory byly napájeny 6 V). Následně byly motory připojeny k odpovídajícím pinům Arduina Nano (viz Obrázek 7).



Obrázek 7 - Připojení krokových motorů drátky k Arduinu Nano přes piny 3-6 a 8-11, zdroj – archiv autora

Arduino Nano s konektorem mini USB bylo připojeno k počítači portem COM3 (lze ho připojit jakýmkoliv COM¹⁹ portem).

4.2. Propojení mechanických součástí

Aby bylo možné s nějakou přesností cokoliv vykreslit na tabulce, bylo nutné zajistit pevné uchycení motorů na ovládacích knoflících tabulky Grafo. Proto byla v této souvislosti věnována největší pozornost právě mechanickému upevnění motorů. K tomu byly využity jednak násady

¹⁹ COM – komunikační rozhraní sériového portu na PC

na motory vytištěné na 3D tiskárně umožňující motory kdykoliv jednoduše odpojit, ale zároveň schopny motory k tabulce pevně uchytit (viz



Obrázek 8). Obrázek 8 - Spojky mezi krokovými motory a tabulkou Grafo vytištěné na 3D tiskárně, zdroj – archiv autora

Návrh pro vytištění byl vytvořen autorem v programu Fusion 360. Dále byly motory připevněny k sobě konstrukcí vyrobenou ze stavebnice Merkur (viz Obrázek 9).



Obrázek 9 - Konstrukce ze stavebnice Merkur určená k propojení motorů navzájem, zdroj – archiv autora

4.3. Front-end v Pythonu s Arduinem

Cílem práce bylo umožnit laickému uživateli komfortně automatizovaně vykreslovat požadované kresby na tabulce Grafo podle vlastního zadání. Za tímto účelem bylo nutné vytvořit uživatelské rozhraní a propojit ho s funkčními části Arduina Nano. K tvorbě tohoto rozhraní byl vybrán programovací jazyk Python pro své široké možnosti v oblastech knihoven s předpřipravenými funkčnostmi a komfortní práci s daty, která byla pro naše účely potřebná.

4.3.1. Použité knihovny Pythonu

Náš program vytvořený v Pythonu by měl poskytnout vytvoření samotného uživatelského rozhraní, umožnit uživateli načrtnout návrh (mimo jiné také pracovat s hranicemi států a s průběhy matematických funkcí), následně jej převézt do formátu zpracovatelného Arduinem, a nakonec provézt odeslání do Arduina.

K těmto konkrétním funkcím existují v Pythonu specifické knihovny (viz Obrázek 10). Jedná se konkrétně o *matplotlib*, *pyserial*, *geojson*, *shapely a numpy*.

Knihovna *matplotlib* je nástroj umožňující jednoduše zobrazit soustavu souřadnic na vyskakovacím okně, také je možné na okno přidat tlačítka pro interakci s uživatelem nebo obrázky.²⁰

Knihovna pyserial umožňuje komunikaci přes komunikační porty počítače.²¹

Knihovna geojson umožňuje zpracovat data ze souborů v geografických vektorových formátech. $^{\rm 22}$

1	import geojson
	from shapely.geometry import shape
	import matplotlib.pyplot as plt
	<pre>from matplotlib.widgets import Button, TextBox</pre>
	import matplotlib.image as mpimg
	import numpy as np
	import serial
	<pre>import serial.tools.list_ports</pre>
	import tkinter as tk

Obrázek 10 - Ukázka z Python kódu zobrazující využití knihoven, příloha s celým kódem -**Chyba! Nenalezen zdroj odkazů.**, zdroj – archiv autora

Knihovna *shapely* umožňuje pracovat s polygony. Nám pomohla obejít limity funkčnosti tabulky, konkrétně neschopnost přerušit vykreslovanou linku.²³

Knihovna numpy slouží k jednodušší práci s daty, v našem případě se seznamy a poli.²⁴

4.3.2. Vytvoření uživatelského rozhraní

Aby bylo rozhraní uživatelsky přívětivé, je důležité zajistit jednoduchý způsob ovládání. Po zpuštění programu jsou pro uživatele zobrazena tři tlačítka poskytující odlišné nabídky (viz Obrázek 11).

²⁰ MATPLOTLIB, *Matplotlib: Visualization with Python*. Online. 2012 – 2024. Dostupné z: <u>https://matplotlib.org/</u>. [cit. 2025-03-08].

 ²¹ PYTHONHOSTED. *PySerial*. Online. 2024. Dostupné z: <u>https://pythonhosted.org/pyserial/</u>. [cit. 2025-03-08].
 ²² GeoJSON Online. Verze 3.2.0 Python Package Index Dostupné z: <u>https://pypi.org/project/geojson/</u>. [cit. 2025-03-08].

²³ SHAPELY. Shapely. Online. 2024. Dostupné z: <u>https://shapely.readthedocs.io/en/stable/</u>. [cit. 2025-03-08].

²⁴ NUMPY. *NumPy*. Online. 2025. Dostupné z: <u>https://numpy.org/</u>. [cit. 2025-03-08].

První funkčnost programu poskytuje uživateli na souřadnicové soustavě znázornit hranice státu (pouze největší souvislý polygon vybraného státu).

Další funkčnost umožňuje vykreslovat průběhy specifických matematických funkcí.

Poslední funkčnost poskytuje možnost načrtnutí vlastního nákresu.

U všech funkčností program simuluje zobrazení náčrtu na reálné tabulce Grafo (v případě, že vykreslování začne ve středu tabulky).

Následně lze jednoduše zmáčknutím tlačítka odeslat zobrazený útvar do Arduina, po čemž se automaticky vykreslí na tabulce.

4.3.3. Vykreslení hranic států

Funkčnost programu vykreslit hranice států spočívá v několika krocích programu, většinově týkajících se zpracování textového souboru geojson nesoucího vektorový zápis hranic. Pro jednodušší práci s takovýmito soubory je důležité využít knihovnu *geojson* (zmíněnou v předchozí kapitole).

Prvním krokem pro vykreslení hranic státu je výběr státu uživatelem. V programu je k dispozici výběr států v podobě vlajek (viz Obrázek 12).



Obrázek 11 - Hlavní menu po otevření programu, zdroj – archiv autora

Po kliknutí na vlajku uživatelem program lokalizuje geojson soubor nacházející se ve stejné složce a převezme si z něj souřadnice polygonů (souvislých tvarů) představujících hranice odpovídající země v podobě zeměpisných souřadnic.²⁵ Kvůli omezením tabulky (lze kreslit pouze nepřerušovanou čáru) je nutné vybrat největší polygon a pouze ten bude vykreslen (například v případě Velké Británie bude vykreslen pouze hlavní ostrov). Následně je největší polygon z hranic vybraného státu upraven tak, aby se vešel na displej tabulky Grafo a je vizualizován pro uživatele v programu. Nyní už uživateli zbývá pouze zmáčknout tlačítko odeslání souřadnic polygonu přes komunikační port do Arduina.

²⁵ Kyd, A. (2023). *GeoJSON Maps of the globe*. GeoJSON Maps. Dostupné z: <u>https://geojson-maps.kyd.au/</u>. [cit. 2025-03-08].



Obrázek 12 - Program po vybrání funkčnosti vykreslování hranic země a kliknutí na českou vlajku, zdroj – archiv autora

4.3.4. Vykreslení průběhu matematických funkcí

Program poskytuje vykreslování průběhu mocninných funkcí (viz Obrázek 13). Průběhy funkcí vykreslené programem nemají za cíl vykreslit přesný průběh funkce, pouze vizualizovat křivky jako např. parabola nebo hyperbola. Proto program neposkytuje měřítko a zobrazí pouze množinu bodů odpovídající stejnému typu průběhu funkce.



Obrázek 13 - Program po vybrání funkčnosti vykreslování průběhu funkcí a zadání mocninné funkce s exponentem 3, zdroj – archiv autora

V případě nespojitých funkcí (například u hyperboly) nastává problém týkající se nemožnosti nadzvednutí pera tabulky Grafo, a tedy přerušení linky. Po dosažení hraniční hodnoty souřadnic vykreslování prvního ramena hyperboly se nakreslí propojovací úsečka mezi rameny a pokračuje se ve vykreslování druhého ramena (viz Obrázek 14 - Vykreslení hyperboly na tabulce, jsou zakresleny také dvě pomocné linie).



Obrázek 14 - Vykreslení hyperboly na tabulce, jsou zakresleny také dvě pomocné linie, zdroj – archiv autora

4.3.5. Možnost volné kresby

Část programu umožňující uživateli provádět volnou kresbu je ze všech nejjednodušší. Při každém kliknutí na tabulku v programu dojde k zobrazení bodu a úsečky spojující ho s bodem předchozím. Následně je převeden do seznamu souřadnic připraveného k odeslání do Arduina. Lze také jednoduše zmáčknutím tlačítka vymazat nákres a začít od znovu (viz Obrázek 15).



Obrázek 15 - Program po výběru funkčnosti volné kresby a vykreslení jednoduchého náčrtu, zdroj – archiv autora

4.3.6. Architektura propojení

Komunikace mezi všemi zařízeními bude probíhat následovně. Arduino Nano máme připojené do počítače přes port COM3. Nejdříve přes COM port nahrajeme kód z Arduino IDE do paměti našeho Arduina. Následně lze spustit program v Pythonu. Nejprve si uživatel vybere funkčnost programu, kterou využije při zakreslení svého náčrtu, a náčrt zakreslí. Poté zmáčkne tlačítko, kterým odešle souřadnice všech vykreslených bodů ve formě řetězce znaků přes COM port do Arduina. Arduino si převede řetězec znaků, který dostalo, na jednotlivé souřadnice a následně na argumenty do funkcí, které jsme mu v Arduino IDE napsali. Hned potom začíná posílat signály do svých pinů, ke kterým jsou připojeny krokové motory nasazené na tabulku Grafo. Ty se začnou otáčet, čímž docílí vykreslení obrázku na tabulce (viz Obrázek 16).



Obrázek 16 - Celkový systém propojení jednotlivých komponentů, A – Tabulka Grafo, B – krokový motor (zakrytý konstrukcí ze stavebnice Merkur), C – řadiče pro krokové motory, D – Arduino Nano, E – zdroj energie, F – počítač, zdroj – archiv autora

5. Diskuse

Tato kapitola se zaměřuje na ověření splnění cílů práce. Prvním cílem bylo dosáhnout automatizovaného ovládání tabulky Grafo pomocí krokových motorů 24byj-48 kontrolovaných mikroprocesorem Arduino Nano.

Podařilo se vykreslit automatizovaně nejen prosté čáry, ale i mnohem složitější obrazce jako například hranice států nebo průběhy nelineárních funkcí (viz **Chyba! Nenalezen zdroj odkazů.**). Bylo tedy zároveň umožněno vykreslovat na tabulku útvary, které by jen stěží šly ručně vytvořit s přijatelnou přesností.

Dalším cílem bylo vytvoření uživatelsky přívětivého rozhraní v programovacím jazyku Python umožňující uživateli jednoduše dálkově kreslit na tabulku.

S pomocí vybraných knihoven programovacího jazyka Python byl vytvořen program s jednoduchým a intuitivním ovládáním, kde si uživatel může vybrat z několika funkčností (viz Obrázek 11) poskytujících několik předpřipravených typů náčrtů, konkrétně hranice států (viz Obrázek 12) a průběhy funkcí (viz Obrázek 13). Program také poskytuje uživateli možnost vytvořit si volný nákres (Obrázek 15). Pro vykreslení požadovaného náčrtu na tabulku Grafo je potřeba pouze zmáčknout tlačítko "Odeslat do Arduina" a tabulka vzápětí začne kreslit.

Uživatelské rozhraní mimo jiné ještě před odesláním do Arduina simuluje samotné vykreslení na tabulce Grafo, čímž umožňuje uživateli vidět srovnání návrhu s realitou.

V průběhu práce na projektu jsme narazili na několik komplikací a omezení. Hlavním je pravděpodobně limitace velikosti paměti v rámci Arduina Nano. Proměnné v kódu totiž mohou dohromady zabírat maximálně 2048 bytů plynoucích z použitého čipu v hardwaru Arduina Nano (ATmega328P).²⁶ Omezuje nás to na vykreslování maximálně 100 bodů na tabulku Grafo najednou. Pravděpodobně by šlo množství přenášených bodů zvýšit využitím jiného datového typu. Pro přenos byl totiž pro přehlednost a čitelnost použitý textový řetězec. I v případě komprimace do jiného datového typu by však omezení na 2048 bytů zůstávalo a bránilo by nám k přenosu řádově vyššího počtu bodů. Vyšší počet bodů by nám umožnil jiný model Arduina, například Arduino Mega (čtyřikrát vyšší objem paměti).²⁷

Stávající řešení by šlo zdokonalit nahrazením provizorní konstrukce pro uchycení krokových motorů účinnějším řešením nežli použití stavebnice Merkur. Jednotlivé součástky stavebnice se pohybem přirozeně uvolňovaly. Nabízí se proto využití 3D tiskárny i pro vyhotovení této spojující součásti.

Výzvy předpovězeny v úvodu práce týkající se problémů s přesností vykreslené čáry souvisejících s mechanickou konstrukcí tabulky Grafo se vždy do jisté míry vyskytují. Nejpatrnější jsou v případě uzavřených útvarů, kdy se začátek a konec linky často nespojí. Čím více bodů je na tabulku vykresleno, tím větší tato odchylka bývá. Po identifikaci těchto problémů proběhlo pozorování s cílem objevit jejich systémový charakter. Chyby však byly spíše náhodné a nebyl objeven žádný trend eliminovatelný úpravou kódu programu.

²⁶ Microchip Technology Inc. ATmega328/P – 8-bit AVR Microcontrollers. [online]. Dostupné z: https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf

²⁷ Arduino. *Arduino Mega* 2560 *Rev3*. [online]. Dostupné z: https://docs.arduino.cc/resources/datasheets/A000067-datasheet.pdf [cit. 2025-03-10].

Závěr

Z výsledků celkové práce v rámci projektu vyplývá, že automatizace kreslení na tabulku Grafo pomocí krokových motorů a Arduina je velmi účinný způsob, jak umožnit na tabulku kresbu složitých obrazců vyžadující značnou přesnost, a jejichž nakreslení manuálně by bylo velmi náročné. Díky automatizovaným metodám bylo možné s poměrně vysokou přesností na tabulce vykreslit i nepravidelné a komplexní tvary jako hranice států.

Za účelem zjednodušení byl vytvořen v programovacím jazyku Python program kompatibilní s kódem pro Arduino komunikující s ním přes port a umožňující jednoduše uživateli zadat své požadavky ohledně vykreslování na tabulku.

Kvůli technickým omezením tabulky Grafo, například nemožnosti přerušit souvislou linii, limitované citlivosti kreslícího hrotu nebo relativně malé plochy samotné tabulky, by se téma tohoto projektu dalo do velké míry označit za vyčerpané. Na druhou stranu se jedná o dobrý příklad toho, že lze v případě využití správných nástrojů poměrně jednoduše automatizovaně ovládat periferie Arduina pomocí softwaru v počítači, a to i bez příliš velkého rozhledu v detailním fungování elektroniky. Na trhu je k dispozici velké množství různých periferií Arduina. Díky tomu je možné i pro laika v dnešní době pomocí programování ovládat fyzické součásti reálného světa. I princip fungování využitý v tomto projektu by šel relativně jednoduše přenést do jiného praktického využití (například automatizování otevírání dveří nebo ovládání rolet).

Zdroje

- 1. Arduino. *Arduino* Mega 2560 Rev3. [online]. Dostupné z: <u>https://docs.arduino.cc/resources/datasheets/A000067-datasheet.pdf</u> [cit. 2025-03-10].
- Atherton, D. P., & Irwin, G. W. (1992). Stepping motors: A guide to theory and practice (4th ed.). IET. Online. Dostupné z <u>https://nirmt.com/storage/uploads/E-BOOK_BE-AUTOMOBILE/Stepping%20motors%20%20a%20guide%20to%20theory%20and%</u>20practice.pdf
- 3. COOPEE, Todd. *Etch A Sketch*. Online. Toytales. Dostupné z: <u>https://toytales.ca/etch-a-sketch-from-ohio-art-company-1960/</u>.
- 4. GeoJSON Online. Verze 3.2.0 Python Package Index Dostupné z: https://pypi.org/project/geojson/
- 5. Howstuffworks. Online. "Inside an Etch-a-Sketch" 21 September 2011. *HowStuffWorks.com.* Dostupné z: <u>https://entertainment.howstuffworks.com/etch-a-sketch-pictures.htm</u>.
- 6. Kyd, A. (2023). *GeoJSON Maps of the globe*. GeoJSON Maps. Dostupné z: <u>https://geojson-maps.kyd.au/</u>
- 7. Maintex. (2025). 24BYJ48 Stepper Motor. Maintex Motors. Dostupné z: https://maintexmotors.com/product/24byj48-stepper-motor/
- 8. MATPLOTLIB, *Matplotlib: Visualization with Python*. Online. 2012 2024. Dostupné z: <u>https://matplotlib.org/</u>
- Microchip Technology Inc. ATmega328/P 8-bit AVR Microcontrollers. [online]. Dostupné z: <u>https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-</u> Automotive-Microcontrollers-ATmega328P_Datasheet.pdf [cit. 2025-03-10].
- 10. NUMPY. NumPy. Online. 2025. Dostupné z: https://numpy.org/
- 11. PYTHONHOSTED. *PySerial*. Online. 2024. Dostupné z: <u>https://pythonhosted.org/pyserial/</u>
- 12. SELECKÝ, Matúš. *Arduino*. Vyd. 1. Computer Press, 2016. ISBN 978-80-251-4840-2.
- 13. SHAPELY. *Shapely*. Online. 2024. Dostupné z: <u>https://shapely.readthedocs.io/en/stable/</u>
- 14. Shaw, A. W. Etch A Sketch Toy. US3055113. 18. 9. 1962. Dostupné z: <u>https://patentimages.storage.googleapis.com/67/4f/71/27e699d9875e3d/US3055113.p</u> <u>df</u>
- 15. Wikimedia Commons. (2024). Arduino Nano isometr [obrázek]. Wikimedia Commons. Dostupné z: https://commons.wikimedia.org/wiki/File:Arduino_nano_isometr.jpg
- 16. Wikipedia. (2024). *Etch A Sketch*. Wikipedia, The Free Encyclopedia. Dostupné z: <u>https://en.wikipedia.org/wiki/Etch_A_Sketch</u>
- 17. Wikipedia. Arduino Nano. In: Wikipedia, The Free Encyclopedia [online]. Dostupné z: <u>https://en.wikipedia.org/wiki/Arduino_Nano</u>