

UNIVERZITA PALACKÉHO V OLMOUCI

PEDAGOGICKÁ FAKULTA

Katedra technické a informační výchovy

## **BAKALÁŘSKÁ PRÁCE**

Raspberry Pi Pico vs. Arduino Nano,  
vzorová zapojení a programy

Petr Jalůvka

## **Prohlášení**

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně a uvedl jsem v ní veškerou literaturu a ostatní informační zdroje, které jsem použil.

V Olomouci dne 17.06.2024

.....

**Petr Jalůvka**

## **Poděkování**

Rád bych poděkoval mému vedoucímu práce panu Mgr. Radimu Děrdovi za vedení mé bakalářské práce, za poskytnutí informací pro mou tvorbu, jeho trpělivost a ochotu při konzultacích. Děkuji rodině a přátelům za podporu v mém studiu a za podporu při psaní této práce.

## Anotace

<b>Jméno a příjmení:</b>	Petr Jalůvka
<b>Katedra:</b>	Katedra technické a informační výchovy
<b>Vedoucí práce:</b>	Mgr. Radim Děrda
<b>Rok obhajoby:</b>	2024

<b>Název práce:</b>	Raspberry Pi Pico vs. Arduino Nano, vzorová zapojení a programy
<b>Název v angličtině:</b>	Raspberry Pi Pico vs. Arduino Nano, sample connections and programs
<b>Zvolený typ práce:</b>	Aplikační
<b>Anotace práce:</b>	Tato bakalářská práce se zaměřuje na porovnání dvou populárních mikropočítačových platforem – Arduino Nano a Raspberry Pi Pico. Cílem práce je poskytnout komplexní přehled o technických specifikacích, možnostech programování a praktickém využití obou platforem. Prostřednictvím experimentálních testů robotů sledujících čáru jsou analyzovány jejich rozdíly, přednosti a nedostatky.
<b>Klíčová slova:</b>	Arduino, Nano, Raspberry, Pico, robot.
<b>Anotace v angličtině:</b>	This bachelor's thesis focuses on the comparison of two popular microcomputer platforms - Arduino Nano and Raspberry Pi Pico. The goal of the work is to provide a comprehensive overview of the technical specifications, programming options and practical use of both platforms. Through experimental tests of line-following robots, their differences, strengths and weaknesses are analyzed
<b>Klíčová slova v angličtině:</b>	Arduino, Nano, Raspberry, Pico, robot.
<b>Přílohy vázané v práci:</b>	1
<b>Rozsah práce:</b>	79 stran
<b>Jazyk práce:</b>	čeština

## Podklad pro zadání BAKALÁŘSKÉ práce studenta

Jméno a příjmení: Petr JALŮVKA  
Osobní číslo: D210257  
Adresa: Olšany u Prostějova 114, Olšany u Prostějova, 79814 Olšany u Prostějova, Česká republika  
Téma práce: Raspberry Pi Pico vs. Arduino Nano, vzorová zapojení a programy  
Téma práce anglicky: Raspberry Pi Pico vs. Arduino Nano, sample connections and programs  
Jazyk práce: Čeština  
Vedoucí práce: Mgr. Radim Děřda  
Katedra technické a informační výchovy

### Zásady pro vypracování:

1. studium literatury
2. popis a funkce Raspberry Pi Pico
3. popis a funkce Arduino Nano
4. Popis jejich rozdílů, výhod nevýhod
5. ukázky zapojení a programů
6. průběžné konzultace s vedoucím závěrečné práce

### Seznam doporučené literatury:

WARREN, John-David, Josh ADAMS a Harald MOLLE. *Arduino robotics*. [New York, NY]: Apress, c2011. Technology in action series. ISBN 978-1430231837.  
VODA, Zbyšek. *Průvodce světem Arduina*. Vydání druhé. Bučovice: Martin Stříž, 2017. ISBN 978-80-87106-93-8.  
*Arduino – Home* [online]. Dostupné z: <https://www.arduino.cc/>  
Lekce 1 – Úvod do Raspberry Pi. *itnetwork.cz – Učíme národ IT* [online]. Copyright © 2023 itnetwork.cz. Veškerý obsah webu [cit. 31.01.2023]. Dostupné z: <https://www.itnetwork.cz/hardware-pc/raspberry-pi/uvod-do-raspberry-pi/>  
*Arduino vs Raspberry Pi: What's the difference?* – Great Learning. Great Learning: Online Courses, PG Certificates and Degree Programs [online]. Copyright © 2013 [cit. 31.01.2023]. Dostupné z: <https://www.mygreatlearning.com/blog/arduino-vs-raspberry-pi/>  
*What are the differences between Raspberry Pi and Arduino?*. Electronics Hub – Tech Reviews | Guides & How-to | Latest Trends [online]. Copyright © 2022 Electronicshub.org [cit. 31.01.2023]. Dostupné z: <https://www.electronicshub.org/raspberry-pi-vs-arduino/>

Raspberry Pi Documentation - Raspberry Pi Pico and Pico W. Raspberry Pi [online]. Copyright © 2012 [cit. 31.01.2023]. Dostupné z: <https://www.raspberrypi.com/documentation/microcontrollers/raspberry-pi-pico.html>

Stav schvalování: Vedoucím katedry schválen studentův podklad VŠKP

Podpis studenta:

Datum:

Podpis vedoucího práce:

Datum:

Podpis vedoucího pracoviště:

Datum:

# Obsah

Úvod .....	9
Teoretická část .....	10
1 Historie .....	11
2 Základy programování a elektrotechniky .....	14
2.1 Základy programování .....	14
2.2 Základy elektrotechniky a elektrosoučástky .....	15
2.2.1 Rezistor .....	15
2.2.2 Kondenzátory .....	16
2.2.3 Cívky .....	17
2.2.4 Diody a LED diody .....	17
2.2.5 Tranzistory .....	17
3 Arduino nano .....	18
3.1 Mikrokontrolér .....	19
3.2 Paměť Arduino nano .....	19
3.3 Programovatelné periferie .....	20
4 Raspberry Pi Pico .....	22
4.1 Procesor v Raspberry Pi Pico .....	23
4.2 Paměti v Raspberry Pi Pico .....	23
4.3 Programovatelné periferie .....	24
5 Porovnání Arduino Nano vs Raspberry Pi Pico .....	26
6 Příslušenství k mikropočítačům .....	28
6.1 Motor .....	28
6.2 H-můstek .....	29
6.3 Optický senzor .....	31
6.4 Nepájivé pole .....	31
7 3D modelování a 3D tisk .....	33
7.1 Programy pro 3D modelování .....	33
7.1.1 Tinkercad .....	33
7.1.2 Autodesk Inventor Professional .....	34
7.1.3 Fusion 360 .....	35
7.1.4 SolidWorks .....	36
7.2 3D Tisk .....	36
8 Využití ve škole .....	38
9 Programovací prostředí .....	40

9.1	Scratch .....	40
9.2	Arduino IDE .....	41
9.3	Micropython .....	42
	Praktická část .....	43
10	Ukázka zapojení a programů.....	44
10.1	Železniční přejezd .....	44
10.2	Házecí kostka .....	48
10.3	Soumrakový spínač .....	52
10.4	Dýchající dioda.....	55
10.5	Postupné roztočení elektromotoru .....	60
11	Robot.....	64
11.1	Srovnání práce s Arduino Nano a Raspberry Pi Pico.....	71
	Závěr.....	72
	Seznam použitých zdrojů .....	73
	Seznam příloh.....	79



# Úvod

V dnešní době, kdy technologie hraje stále větší roli ve všech aspektech našeho života, se mikropočítače staly nezbytným nástrojem pro vývoj inovativních elektronických projektů. Mezi nejpopulárnější mikropočítače patří Arduino Nano a Raspberry Pi Pico. Tyto dvě platformy nabízejí široké spektrum možností pro nadšence, profesionály a vzdělávací instituce, a to díky jejich dostupnosti, výkonnosti a podpoře komunity.

Arduino Nano, známé svou jednoduchostí a robustností, je často volbou pro začátečníky i pokročilé uživatele, kteří hledají spolehlivou a snadno programovatelnou platformu. Arduino ekosystém nabízí bohatou knihovnu zdrojů, tutoriálů a projektů, což usnadňuje učení a rychlé nasazení do praxe. Díky svému kompaktnímu designu a všestranným možnostem připojení se Arduino Nano stalo základem mnoha úspěšných projektů, od jednoduchých senzorických aplikací po komplexní řídicí systémy.

Raspberry Pi Pico, na druhé straně, představuje nový přístup k mikropočítačům s důrazem na vyšší výpočetní výkon a flexibilitu. Srdcem této platformy je čip RP2040, navržený přímo Raspberry Pi Foundation, který nabízí dvoujádrový procesor ARM Cortex M0+ s vysokou frekvencí a bohatou sadou periferií. Raspberry Pi Pico je také známé svou schopností běžet na různých programovacích jazycích, včetně MicroPython a C/C++, což poskytuje uživatelům větší svobodu při vývoji aplikací.

Tato bakalářská práce se zaměřuje na komplexní porovnání Arduino Nano a Raspberry Pi Pico z několika hledisek. Nejprve se podíváme na historický kontext a technické specifikace obou platforem. Následně prozkoumáme jejich možnosti programování a praktické využití, včetně experimentálních testů robotů sledujících čáru. Důraz bude kladen na reálné aplikace a experimentální výsledky, které ukazují, jak se tyto dva systémy liší a v čem jsou jejich přednosti a nedostatky.

Hlavním cílem této práce je poskytnout čtenářům ucelený přehled o výhodách a nevýhodách obou platforem a pomoci jim při rozhodování, která z nich je pro jejich specifické projekty nejvhodnější. Doufám, že tato práce přispěje k lepšímu pochopení a efektivnějšímu využití těchto výkonných nástrojů v oblasti výuky, prototypování a inovací.

Vedlejším cílem je sestavit robota s využitím obou technologií, robota následně naprogramovat s využitím dvoustavové regulace a vyzkoušet jej.

## **Teoretická část**

# 1 Historie

V této části se budeme zabývat celkovou historií mikroprocesorů a sledovat jejich postupný vývoj, který ovlivnil moderní technologii a změnil způsob, jakým lidé interagují s elektronikou a počítači.

Historie mikroprocesorů začíná v 60. letech 20. století, kdy byl vyvinut první mikroprocesor, Intel 4004, který se dostal na trh v roce 1971. Tento revoluční čip, navržený Marcianem E. Hoffem, byl velmi omezený výkonem, ale položil základy pro budoucí vývoj a inovace v oblasti výpočetní techniky. (Mikropočítače; 2024)

První mikroprocesory, jako je Intel 4004, byly primárně používány v průmyslových aplikacích, ale postupem času se jejich výkon a funkce zlepšovaly. V roce 1974 byl uveden na trh Intel 8080, první mikroprocesor, který se využil v osobním počítači. Následovala série inovací a vylepšení mikroprocesorů, včetně čipů jako Intel 8080 a Motorola 6800, které nabízely vyšší výkon a více funkcí.

V 80. a 90. letech 20. století začaly vznikat jednodeskové počítače, které kombinovaly mikroprocesor s dalšími perifériemi na jedné desce. Tím byly zjednodušeny návrhy a vývoj elektronických zařízení. Mezi příklady jednodeskových počítačů patří Amiga, Atari ST a první IBM (International Business Machines Corporation) PC. (Shirriff; 2016)

S rozvojem technologií výroby polovodičů a návrhu čipů se zlepšovala výkonnost a funkčnost mikroprocesorů. V roce 2005 byl založen projekt Arduino na Interaction Design Institute Ivrea v Itálii. Tento projekt měl za cíl vytvořit snadno použitelnou platformu pro vývoj embedded projektů a experimentování s elektronikou. První Arduino deska, nazvaná Arduino Serial, byla vyvinutá v roce 2005 a následně byla upravena s několika vylepšeními a přejmenována na Arduino NG. Další deskou uvedenou na trh v roce 2007 byla Arduino Diecimila. Následovala ji deska Arduino nano, která je známá dneska díky své velikosti a kompaktnímu provedení. Byla uvedena na trh v roce 2008 a i dnes pořád hojně využívána. (Historie mikrokontrolerů; 2014)

V roce 2009 vyšla deska Arduino Duemilanove, což byla hojně využívaná deska, kterou ovšem nahradilo Arduino Uno, které je i dnes jedno z nejoblíbenějších verzí arduina. Je to jednodeskový mikrokontrolér postavený na platformě ATmega328P. Obsahuje všechny potřebné prvky pro zapojení a programování elektronických projektů, včetně digitálních a analogových vstupů a výstupů, sériové komunikace a možnosti napájení přes USB nebo externí zdroj. Díky svým vlastnostem, spolehlivosti a cenové dostupnosti se Arduino Uno stalo jedním z nejpopulárnějších vývojových nástrojů pro výuku, prototypování a tvorbu embedded

systemů po celém světě. Kromě Una vyšla v téže roce ještě jedna deska s názvem Arduino Mega, který oproti Unu měla výhodu v tom, že obsahoval více vstupů a výstupů, což bylo výhodné pro rozsáhlejší projekty. Další z desek uvedených na trh byla deska Arduino Leonardo, která má mikrokontrolér postavený na platformě ATmega32u4. Jednou z hlavních vlastností Arduina Leonardo je jeho schopnost emulovat USB zařízení přímo z Arduina, což znamená, že může fungovat jako klávesnice, myš, nebo jiné typy USB zařízení. Arduino Due, bylo první Arduino s 32bitovým mikrokontrolérem ARM, uvedené na trh v roce 2012. Poskytovalo vyšší výkon a možnosti pro náročné projekty. (Arduino Duemilanove; 2024; ARDUINO ČILI MIKROKONTROLÉRY PRO KAŽDÉHO; 2021 )

Raspberry Pi začalo být dostupné na trhu v roce 2012 a tímto způsobem začalo konkurovat Arduinu v určitých oblastech. Zatímco Arduino bylo primárně zaměřeno na jednoduché embedded projekty a experimentování s elektronikou, Raspberry Pi nabízelo plnohodnotný počítačový systém s operačním systémem Linux, což umožnilo provozovat širokou škálu aplikací a programů, včetně webových serverů, mediálních center, her a dalších operací. (What is a Raspberry Pi?; 2021)

První modely, Raspberry Pi 1 Model A a B, byly postaveny na jednoduchém ARM procesoru a obsahovaly omezenou paměť. V roce 2015 byl představen Raspberry Pi 2 Model B, který přinesl vylepšený procesor, více paměti RAM a lepší výkon. Tento model umožnil provozování širší škály aplikací a projektů. Kromě hlavních modelů Raspberry Pi existuje také Raspberry Pi Zero, které je extrémně kompaktní a cenově dostupné. Raspberry Pi Zero bylo představeno v roce 2015 a nabízí základní funkce Raspberry Pi ve velmi malém balení. Roku 2016 byl uveden na trh Raspberry Pi 3 Model B, který přinesl integrované Wi-Fi a Bluetooth, což znamenalo další krok vpřed v možnostech komunikace a připojení k internetu. V roce 2019 byl uveden na trh Raspberry Pi 4 Model B, který přinesl významné vylepšení výkonu a funkcí. Tento model obsahuje výkonnější čtyřjádrový procesor, více paměti RAM a možnost připojení dvou monitorů. Roku 2020 byl uveden na trh Raspberry Pi 400, který je unikátní tím, že integrovaný Raspberry Pi do klávesnice. V podstatě Raspberry Pi 400 vypadá jako klasická klávesnice, ale v jejím těle se nachází kompletní počítačový systém Raspberry Pi. To zahrnuje procesor, paměť, konektory pro periferie (například HDMI, USB, Ethernet), a další prvky potřebné pro provozování operačního systému a aplikací. V roce 2021 byl uveden na trh Raspberry Pi pico, kterým se podrobněji budu zabývat v dalších kapitolách. (Raspberry Pi Models and Specs: A Comprehensive Guide; 2023) Jako nejnovější produkt je Raspberry Pi 5 poháněn 64bitovým čtyřjádrovým procesorem. (Raspberry Pi 5 je tu; 2023)

Od prvních mikroprocesorů po moderní čipy s více jádry, byla neustále zvyšována výpočetní síla mikroprocesorů. Každá nová generace mikroprocesorů přinášela zvýšení výkonu, které umožnilo zpracování složitějších úloh a aplikací. Trendem vývoje mikroprocesorů je neustálé zmenšování jejich velikosti a zvyšování hustoty integrace. Díky pokroku v technologii výroby polovodičů se velikost tranzistorů na čípech neustále zmenšuje, což umožňuje výrobu čipů s vyšší výpočetní silou a nižší spotřebou energie. (The Future of Microprocessors; 2011)

## 2 Základy programování a elektrotechniky

Tato kapitola je zaměřena na seznámení se základními pojmy v programování, elektrotechnice a představuje základní elektrosoučástky.

### 2.1 Základy programování

V této podkapitole se seznámíme se základními pojmy a koncepty programování, které jsou nezbytné pro práci s mikrokontroléry, jako je Raspberry Pi Pico a Arduino Nano.

#### Co je Programování?

Programování je proces vytváření instrukcí, které počítač nebo mikrokontrolér vykonává. Tyto instrukce jsou psány v programovacím jazyce, který je srozumitelný pro vývojáře a přeložitelný do strojového kódu, kterému rozumí počítačový hardware. (What is programming; 2019)

Základní koncepty programování jsou proměnné, datové typy, operátory, podmínky, cykly, funkce.

Proměnné jsou úložiště dat, která mohou obsahovat různé typy hodnot, jako jsou čísla, text nebo logické hodnoty. Příklad:  $x = 10$  (číslo),  $y = \text{"Ahoj"}$  (text, řetězec),  $z = \text{True}$  (logická hodnota).

Datové typy jsou typy dat, se kterými můžeme pracovat. Mezi základní datové typy patří: Integer (celá čísla), Float (desetinná čísla), String (řetězec), Boolean (logické hodnoty).

Operátory umožňují provádět operace s proměnnými, jako je sčítání, odčítání, násobení a dělení, porovnání hodnot, zda jsou menší, větší, jsou si rovné, nerovné.

Podmínky umožňují provádět rozhodnutí v kódu na základě určitých kritérií. Klíčovým slovem pro podmínku je `if`, za kterým následuje logický výraz. Pokud je výraz pravdivý provede se následující část programu, pokud není pravdivý, přechází se na `else if`, což je další podmínka, která ovšem nemusí být použita a v tom případě program skočí přímo na `else` a vykoná příkaz co je v této části. (Podmínky větvení; 2024)

Cykly umožňují opakované provádění bloku kódu, dokud nejsou splněny dané podmínky. Jsou dva hlavní `while` a `for`. Existuje ještě další `do while`, ale ten používat nebudeme.

Funkce je blok kódu, na který se odkazujeme kdekoli v programu a vykoná se kód obsažený ve funkci. (Základy programování; 2006; Funkce; 2024)

## 2.2 Základy elektrotechniky a elektrosoučástky

V této podkapitole se seznámíme se základními principy elektrických obvodů, součástek, porozumění základním elektronickým veličinám a jejich vzájemným vztahům jako základ pro jakýkoliv projekt v oblasti elektrotechniky.

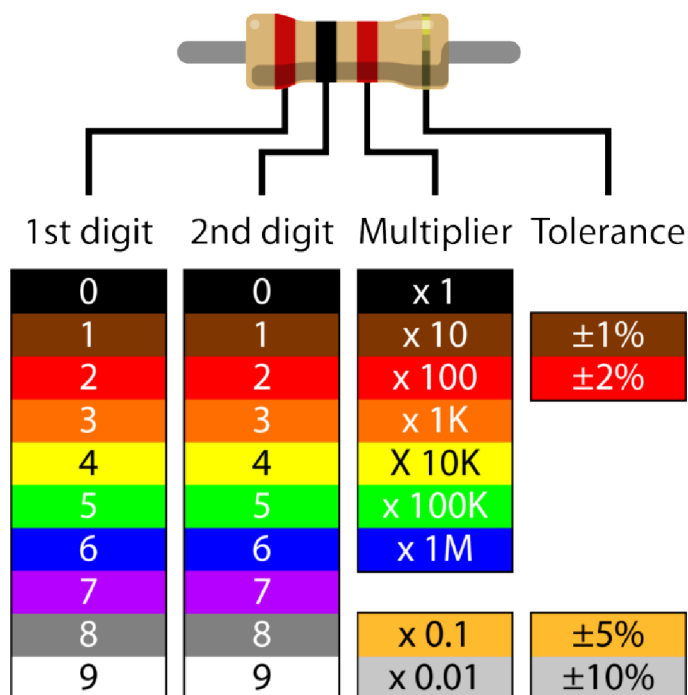
Elektrotechnika je vědní obor, který se zabývá studiem a aplikací elektrických a elektromagnetických jevů. Klíčovými pojmy v elektrotechnice jsou elektrické veličiny, které jsou základem pro pochopení fungování elektrických obvodů a zařízení. Mezi základní elektronické veličiny patří elektrické napětí, elektrický proud a odpor. Elektrické napětí je rozdíl potenciálů mezi dvěma body v elektrickém obvodu tedy mezi kladným a záporným pólem zdroje. Měří se v jednotkách voltů (V). Elektrický proud je měření toku náboje, kterým prochází elektrickým obvodem, jeho jednotka je ampér (A). Odpor je schopnost materiálu nebo součástky omezit tok elektrického proudu. Jednotkou je ohm ( $\Omega$ ). (Elektrické napětí; 2024; Elektrický proud; 2024)

Základní elektrické obvody máme sériové a paralelní. V sériovém obvodu jsou součástky spojeny za sebou po jedné fázi což má nevýhodu velkého odporu, který je součtem jednotlivých součástek. Pokud jedna součástka selže, obvod se přeruší. V paralelním obvodu jsou součástky připojeny tak, že mají společný počáteční a koncový bod, což vytváří uzly, kde se proud dělí do jednotlivých větví. Každá větev obsahuje jednu nebo více součástek, které jsou zapojeny paralelně. To znamená, že celkový odpor obvodu je menší než nejmenší odpor v jednotlivých větvích. Výhodou paralelního zapojení je, že pokud jedna součástka selže, ostatní větve obvodu mohou nadále fungovat. (Elektrotechnický základ; 2011; Elektronika; 2006)

### 2.2.1 Rezistor

Rezistor je základní komponenta v elektronice, která omezuje tok elektrického proudu v obvodu. Jeho hlavní funkcí je poskytovat odpor, který je měřen v ohmech ( $\Omega$ ), což vede ke snížení napětí a regulaci proudu protékajícího obvodem. Rezistory jsou klíčové pro řízení výkonu a ochranu dalších komponent před poškozením v důsledku nadměrného proudu. Rezistory jsou obvykle vyrobeny z materiálů, které mají specifický odpor, jako jsou uhlíkové kompozity, kovové filmy nebo dráty. Existuje několik typů rezistorů a jsou to vrstevné rezistory, drátové rezistory a potenciometry. Vrstevné rezistory jsou ty, u kterých se jejich odpor nijak nemění a používají se tak kde je potřeba konstantní odpor. Mají svoje specifické

rozlišování. Každý rezistor má na sobě proužky, podle kterých lze určit jeho odpor. Podrobněji to vysvětluje následující obrázek.



Obr. 2.1 Vyznačení odporu rezistoru

(Zdroj: [https://dotekomanie.cz/wp-content/uploads/2015/09/Resistor\\_color\\_code\\_chart-1009x1024-1009x1024x.png](https://dotekomanie.cz/wp-content/uploads/2015/09/Resistor_color_code_chart-1009x1024-1009x1024x.png))

U Arduina a Raspberry se nejčastěji s nimi potkáme při zapojení LED diod. Dalšími rezistory jsou proměnné neboli potenciometry, který umožňují nastavení odporu v určitém rozsahu. Typickým příkladem z praxe jsou ovladače hlasitosti či stmívače světla. Existují ještě další rezistory jako jsou termistory, jejichž odpor se mění s teplotou a fotorezistory, kde se odpor mění intenzitou světla. (Elektrotechnický základ; 2011)

## 2.2.2 Kondenzátory

Kondenzátory jsou další elektrosoučástkou, které slouží k uchovávání a uvolňování elektrické energie. Jsou široce používány v různých elektronických obvodech a zařízeních díky svým unikátním vlastnostem a funkcím. Skládá se ze dvou vodivých desek, které jsou oddělené nevodivým materiálem, který se nazývá dielektrikum. Když se mezi desky přivede napětí, tak se na deskách vytvoří pole. Jedno se nabije kladně a druhá záporně. Množství uloženého náboje je pak úměrné kapacitě kondenzátoru. Příklad využití s Arduinem a Raspberry může být měření napětí, kdy nám kondenzátor zmírní nežádoucí elektrické signály, které by mohli ovlivnit přesnost měření. (Elektrotechnický základ; 2011)



### 2.2.3 Cívky

Cívky, známé také jako indukory, jsou pasivní elektronické součástky, které ukládají energii v magnetickém poli, když prochází elektrický proud. Jsou důležitou součástí mnoha elektronických obvodů a aplikací. Cívka se skládá z vodiče, často navinutého do spirály. Když prochází cívkou elektrický proud, vytváří se kolem ní magnetické pole. Změna proudu v cívce způsobí změnu magnetického pole, což indukuje napětí opačné polarity. Tento jev se nazývá indukčnost. (*FRIK's WEB*; 2013; Pasivní elektronické součástky; 2024)

### 2.2.4 Diody a LED diody

Diody jsou elektronické součástky, které umožňují proudění proudu pouze jedním směrem. Tento jednosměrný průtok proudu je klíčový pro mnoho aplikací v elektronice, protože pomáhá chránit součástky před poškozením zpětným proudem. Diody se často používají v usměrňovačích a jako ochranné prvky v obvodech, aby zabránily zpětnému toku proudu, který by mohl poškodit citlivé komponenty. LED diody (Light Emitting Diodes) jsou speciálním typem diod, které mají schopnost vydávat světlo, když jimi prochází elektrický proud. Jsou velmi efektivní, mají dlouhou životnost a nízkou spotřebu energie. Díky těmto vlastnostem se často používají jako indikátory různých stavů v elektronických zařízeních, například pro signalizaci zapnutí, vypnutí nebo chyby. LED diody jsou dostupné v různých barvách a velikostech a nacházejí uplatnění v široké škále aplikací od jednoduchých indikátorů po složité displeje a osvětlovací systémy. (*FRIK's WEB*; 2013)

### 2.2.5 Tranzistory

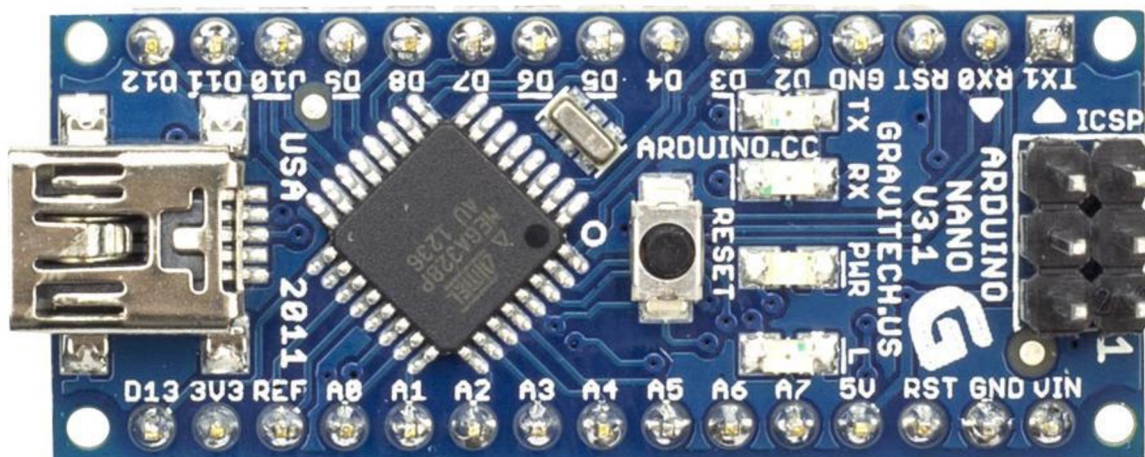
Tranzistory jsou základní a velmi důležité součástky v elektronice, které mohou fungovat jako spínače nebo zesilovače. Skládají se ze tří vrstev polovodičových materiálů, které tvoří tři elektrody: kolektor, báze a emitor. Existují dva základní typy tranzistorů: bipolární tranzistory (BJT) a tranzistory s efektním polem (FET). (Tranzistor; 2024)

### 3 Arduino nano

V této kapitole se budeme zabývat Arduino Nano, kompaktním a výkonným mikropočítačem, který nabízí široké možnosti pro vývoj elektronických projektů a aplikací. Arduino Nano je oblíbeným vývojovým nástrojem mezi inženýry, studenty a nadšenci, díky svému kompaktnímu provedení a snadnému použití. Na obrázku 3.1 můžete vidět samotnou desku Arduino Nano, na níž je umístěn mikrokontrolér ATmega328. Tento mikrokontrolér je jádrem celé desky a zajišťuje provoz a funkčnost Arduino Nano. ATmega328 je výkonný a spolehlivý mikrokontrolér, který poskytuje dostatečný výkon pro provádění různých úloh a operací. (Arduino memory guide; 2023)

Kromě mikrokontrolerů je deska Arduino Nano vybavena krystalovým oscilátorem, který zajišťuje stabilitu hodinového signálu. Tento oscilátor je klíčový pro správnou funkci mikropočítače a umožňuje přesné časování operací a komunikace. Co se týče napájení, deska Arduino Nano může být napájena buď přes USB port, který slouží i pro programování desky, nebo externím napájecím zdrojem připojeným k pinům VIN (napájecí napětí) a GND (zem). Tato flexibilita v napájení umožňuje snadné integrování Arduino Nano do různých projektů a aplikací, ať už se jedná o připojení k počítači nebo k externím zdrojům energie. (Arduino nano; 2020)

Dále v kapitole se budu detailně zabývat hardwarovými parametry a schopnostmi Arduino Nano, abych popsal jeho potenciál a možnosti pro vývoj elektronických projektů.



*Obr. 3. 1 Arduino Nano*

(Zdroj: [https://cz.farnell.com/productimages/large/en\\_GB/1848691-40.jpg](https://cz.farnell.com/productimages/large/en_GB/1848691-40.jpg))

### 3.1 Mikrokontrolér

Arduino Nano využívá mikrokontrolér ATmega328P (u starších verzí) nebo ATmega328 (u novějších verzí). Tento mikrokontrolér je založen na architektuře AVR od společnosti Microchip (dříve Atmel) a je 8bitový. Architektura AVR je známá svou spolehlivostí, snadnou použitelností a širokou podporou vývojových prostředí. Standardní frekvence procesoru ATmega328P/ATmega328 na Arduino Nano je 16 MHz. Tato frekvence poskytuje dostatečný výkon pro provádění různých úloh a operací, ať už se jedná o řízení periferií, zpracování dat nebo komunikaci s externími zařízeními. (Arduino nano; 2020)

Procesor je dostatečně výkonný pro většinu projektů, které jsou vytvářeny na platformě Arduino. Mikrokontrolér ATmega328P/ATmega328 je navržen s ohledem na nízkou spotřebu energie, což je užitečné pro aplikace s bateriovým napájením nebo pro zařízení, která mají být v provozu po dlouhou dobu. Tato vlastnost umožňuje efektivní využití energie a prodloužení výdrže baterie, což je důležité pro mobilní zařízení a projekty, které mají být provozovány v terénu nebo na místech, kde není dostupný trvalý zdroj energie. (ATmega328P Microcontroller; 2018)

### 3.2 Paměť Arduino nano

Arduino Nano má více typů pamětí, které jsou klíčové pro jeho funkci a použití v různých projektech a aplikacích.

Arduino Nano obsahuje 32 KB flash paměti, což je místo, kde je uložen uživatelský kód (program). Tato paměť je nepostradatelná, protože bez ní by nebylo možné uložit programy do desky Arduino. K dispozici je také bootloader, který umožňuje programování desky přes sériové rozhraní. Flash paměť je trvalá a uchovává programy i po vypnutí desky, což je klíčové pro opakované spouštění a aktualizace programů.

Další pamětí je statická paměť (anglicky Static Random Access Memory; dále SRAM), kterou má arduino nano 2 KB, což je paměť používaná pro ukládání proměnných a dat v průběhu běhu programu. SRAM poskytuje dočasné úložiště pro běžící program a umožňuje manipulaci s daty v reálném čase. Velikost SRAM je omezená, a proto je důležité správně řídit a optimalizovat použití paměti v programu.

Arduino Nano disponuje také 1 KB EEPROM, což je paměť určená pro ukládání dat, která mají být trvale uložena i po vypnutí desky. To umožňuje uchovávat například nastavení nebo různé konfigurační údaje.

Kromě pamětí typu flash, SRAM a EEPROM můžeme v Arduino Nano také využívat dynamickou alokaci paměti pomocí funkcí jako je *malloc()* a *free()*. To umožňuje dynamicky alokovat paměť pro proměnné a datové struktury v průběhu běhu programu. Dynamická alokace paměti poskytuje flexibilitu a možnost efektivního využití dostupné paměti pro různé účely a aplikace. Je však důležité pamatovat si na správné uvolňování paměti, aby nedocházelo k problémům s paměťovou únavou nebo únikem paměti. (Arduino memory guide; 2023)

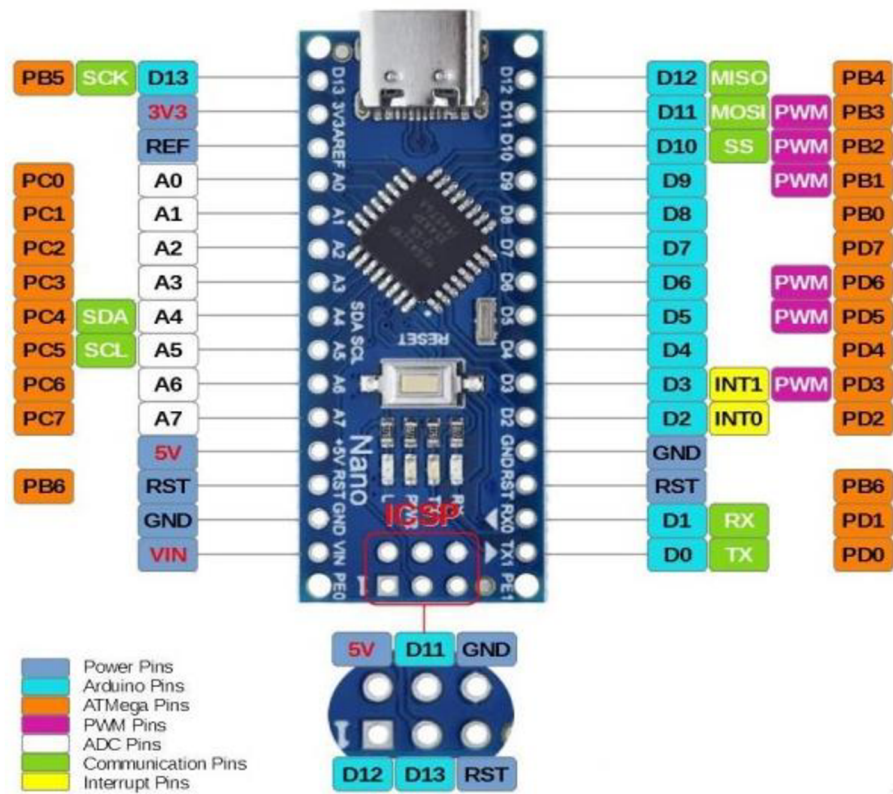
### 3.3 Programovatelné periferie

Arduino Nano disponuje několika rozhraními pro komunikaci s různými zařízeními a senzory.

Digitální piny jsou základními stavebními bloky pro interakci s digitálními zařízeními. Každý pin může být konfigurován jako vstup nebo výstup, což umožňuje snadnou komunikaci s tlačítky, spínači, LED diodami a dalšími digitálními zařízeními. Tímto způsobem lze jednoduše ovládat různé elektronické prvky v projektu.

Analogové vstupy rozšiřují možnosti měření, protože umožňují čtení analogových hodnot. Arduino Nano má několik analogových pinů, které mohou být využity k připojení různých analogových senzorů, jako jsou teplotní senzory, světelné senzory nebo senzory vlhkosti. Tímto způsobem lze snadno získávat data z fyzického světa. PWM piny umožňují generování modulovaných signálů, což je užitečné pro ovládání různých zařízení, jako jsou například servomotory nebo regulace jasu LED diod. Tato funkcionalita poskytuje možnost jemného řízení výstupního signálu s různou úrovní napětí.

Arduino Nano také podporuje sériovou komunikaci pomocí rozhraní UART, I2C a SPI. To umožňuje komunikaci s jinými mikrokontroléry, senzory, displeji a dalšími zařízeními, což rozšiřuje možnosti propojení s dalšími elektronickými komponenty. Možnosti připojení k jednotlivým pinům je znázorněno na obr.3.2. (Arduino Nano Tutorial – Pinout & Schematics; 2020)



*Obr. 3. 2 Arduino Nano Piny*

(Zdroj: [https://www.hwpro.cz/oc/image/cache/catalog/produkty/HW0007\\_14-800x600.jpg](https://www.hwpro.cz/oc/image/cache/catalog/produkty/HW0007_14-800x600.jpg))

## 4 Raspberry Pi Pico

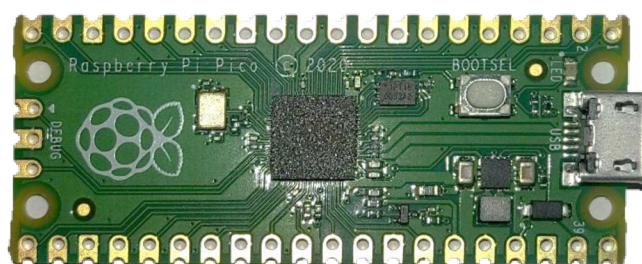
Raspberry Pi Pico představuje celkem nový (rok 2021) přírůstek do rodiny produktů od Raspberry Pi Foundation, který se vyznačuje kompaktním designem a vysokým výkonem. V této kapitole se podíváme podrobněji na hardwarové parametry a schopnosti této desky, která je zobrazená na obr. 4. 1.

Samotná deska Raspberry Pi Pico je vybavena mikrokontrolérem RP2040, který je srdcem tohoto zařízení. RP2040 je výkonný dvoujádrový mikrokontrolér, který byl speciálně navržen pro aplikace Internetu věcí (IoT) a embedded systémy. Díky svému vysokému výkonu a nízké spotřebě energie je RP2040 ideální volbou pro širokou škálu projektů, od jednoduchých senzorických aplikací po pokročilé robotické projekty. (How to connect Raspberry Pi Pico W to ThingsBoard?; 2024)

Na desce Raspberry Pi Pico jsou kromě mikrokontroléru RP2040 umístěny také krystaly pro vytváření přesného hodinového signálu. Tyto krystaly jsou klíčové pro správnou funkci mikrokontroléru a zajišťují přesnost časování a synchronizaci dat v rámci projektů.

Další součástky na desce zahrnují regulátory napětí pro napájení mikrokontroléru, USB konektor pro programování a komunikaci s počítačem, a další periferie jako jsou GPIO piny, které umožňují připojení externích senzorů, displejů, motorů a dalších komponent. (Vaněček; 2022)

Tato kapitola se bude podrobně zabývat samotným mikrokontrolérem RP2040 a jeho hardwarovými periferiemi, včetně GPIO pinů, UART, SPI, I2C a dalších.



*Obr.4. 1 Raspberry Pi Pico*

(Zdroj: Vlastní fotografie)

## 4.1 Procesor v Raspberry Pi Pico

Procesor v Raspberry Pi Pico je integrovaný čip nazvaný RP2040. Tento čip byl vyvinut přímo společností Raspberry Pi Foundation speciálně pro Raspberry Pi Pico. RP2040 je vybaven dvoujádrovým ARM Cortex-M0+ procesorem. Tento procesor obsahuje 32bitový RISC (Reduced Instruction Set Computing), navržený pro nízký výkon a nízkou spotřebu energie, což je ideální pro aplikace s omezenými zdroji. Každé jádro procesoru Cortex-M0+ běží na frekvenci 133 MHz, což umožňuje efektivní zpracování úloh a zajišťuje rychlou odezvu zařízení. Dvoujádrová architektura procesoru RP2040 poskytuje větší flexibilitu a výkon pro zpracování úloh. Procesor může efektivně spouštět různé úlohy současně na obou jádrech nebo optimalizovat výkon a spotřebu energie podle potřeby. RP2040 může být napájen širokým rozsahem napětí, obvykle od 1,8 V do 5,5 V. To umožňuje připojení k různým zdrojům napájení, včetně baterií, solárních panelů, adaptérů nebo jiných zdrojů energie.

Díky širokému rozsahu napětí je Raspberry Pi Pico ideální pro aplikace, které vyžadují použití baterií. Lze použít různé typy baterií, včetně alkalických, lithiových, dobíjecích baterií a dalších. Můžeme jej napájet přes USB konektor, který je umístěný na desce, což umožňuje snadné napájení z počítače nebo jiného zařízení s USB portem. Mimo něj můžeme Pico napájet přes GPIO piny, což umožňuje další flexibilitu při návrhu a implementaci napájecího systému. (Vaněček; 2022; How to connect Raspberry Pi Pico W to ThingsBoard?; 2024)

## 4.2 Paměti v Raspberry Pi Pico

Raspberry Pi Pico obsahuje několik typů pamětí, které slouží k ukládání programu, dat a dalších informací.

Raspberry Pi Pico má 264 KB SRAM (Static Random Access Memory), což je rychlá paměť používaná pro ukládání proměnných, datových struktur a provádění operací programu. SRAM je velmi rychlá a umožňuje rychlý přístup k datům, což je klíčové pro efektivní běh programu. Nicméně je SRAM závislá na napájení a ztrácí svá data při vypnutí, což je důležité vzít v úvahu při návrhu aplikace.

Raspberry Pi Pico má integrovanou flash paměť o velikosti 2 MB. Tato paměť slouží k ukládání uživatelského programu, který běží na mikrokontroléru. Flash paměť je trvalá a uchovává svá data i po vypnutí napájení, což znamená, že programy mohou být uloženy

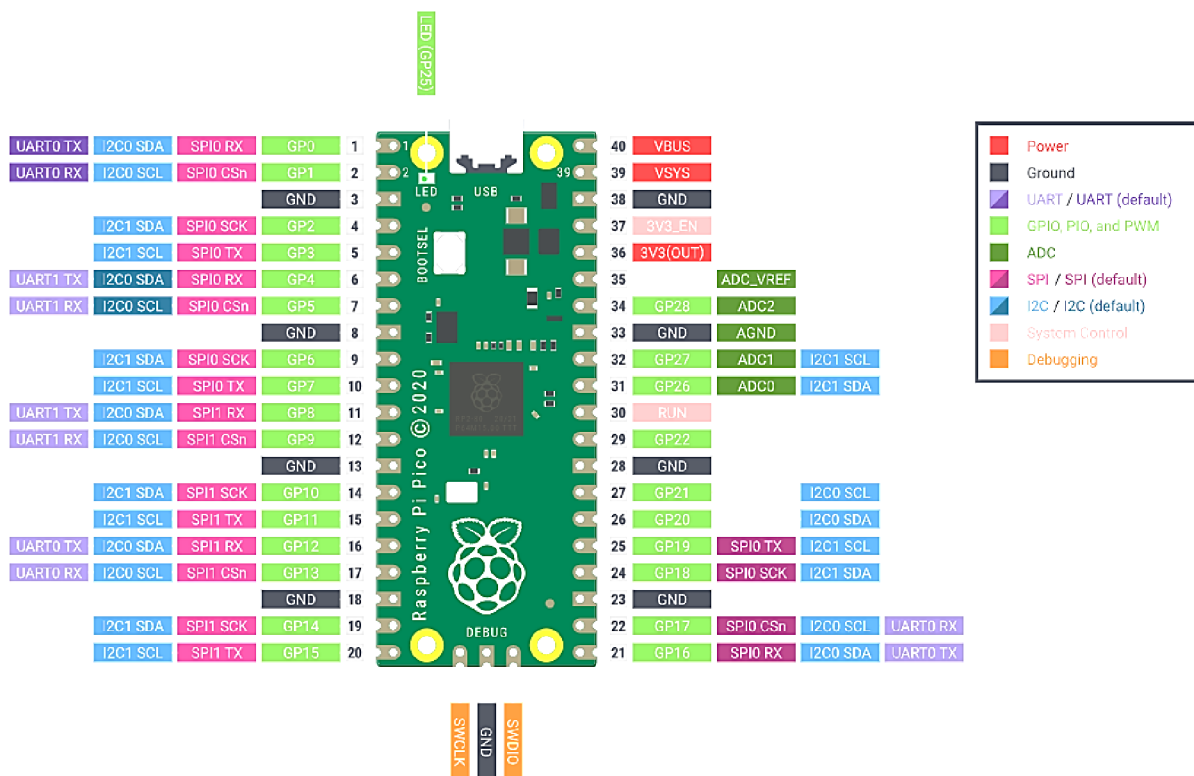
a spuštěny opakovaně bez nutnosti znovu nahrávat. To je velmi výhodné pro vývojáře, protože umožňuje snadnou aktualizaci a správu programů.

Kromě těchto interních pamětí může Raspberry Pi Pico také využívat externí paměti. To zahrnuje paměťové karty, externí flash paměti nebo další typy pamětí připojené přes rozhraní jako SPI nebo I2C. Externí paměti umožňují rozšíření paměťové kapacity Raspberry Pi Pico a umožňují ukládání větších datových sad a programů.

Tyto různé typy pamětí poskytují Raspberry Pi Pico dostatečný úložný prostor pro ukládání programů, dat a dalších informací, což je zásadní pro jeho funkci jako mikrokontrolér a pro provádění různých projektů a aplikací. Tento paměťový systém umožňuje vývojářům vytvářet komplexní aplikace a experimentovat s různými funkcemi. (Raspberry Pi Pico and Pico W; 2024)

### 4.3 Programovatelné periferie

Programovatelné periferie jsou klíčovými prvky Raspberry Pi Pico, které umožňují uživatelům připojit různá zařízení a senzory a komunikovat s nimi. Možnosti připojení k jednotlivým pinům je znázorněno na obr.4.2. (Raspberry Pi Pico and Pico W; 2024)



Obr. 4. 2 Raspberry Pi Pico Piny

(Zdroj: <https://www.raspberrypi.com/documentation/microcontrollers/images/pico-pinout.svg>)



GPIO (General Purpose Input/Output) piny jsou základními vstupy a výstupy, které umožňují Raspberry Pi Pico komunikovat s externími zařízeními. Tyto piny mohou být nastaveny jako digitální vstupy nebo výstupy a mohou být použity k čtení stavu externích signálů nebo ovládání různých zařízení, jako jsou LED diody, tlačítka, senzory a další. GPIO piny poskytují základní možnost interakce s okolním světem a jsou klíčovým prvkem pro vytváření různých projektů.

PWM (Pulse Width Modulation) je technika, která umožňuje generovat signály s proměnlivou šířkou pulzu. Raspberry Pi Pico obsahuje hardwarové PWM kanály, které umožňují generovat PWM signály s různými frekvencemi a rozlišením. Tato funkce je užitečná pro řízení různých zařízení, jako jsou motory, LED diody, servomotory a další. PWM umožňuje přesné řízení intenzity světla, rychlosti motorů a dalších parametrů. (Raspberry pi pico datasheet; 2024)

SPI (Serial Peripheral Interface) je sériové komunikační rozhraní, které umožňuje komunikaci mezi mikrokontrolérem a externími zařízeními, jako jsou displeje, senzory, paměťové karty a další. (Obvody rozhraní SPI; 2015)

I2C (Inter-Integrated Circuit) je další sériové komunikační rozhraní, které umožňuje komunikaci mezi mikrokontrolérem a externími zařízeními. Toto rozhraní je vhodné pro připojení různých senzorů, displejů a dalších periférií, které podporují I2C komunikaci. (Stručný popis sběrnice I2C; 2024)

UART (Universal Asynchronous Receiver-Transmitter) je sériové komunikační rozhraní, které umožňuje přenos dat mezi mikrokontrolérem a externími zařízeními pomocí sériového protokolu. Toto rozhraní je často používáno pro komunikaci s počítačem nebo jinými mikrokontroléry. (Co je to UART; 2011)

## 5 Porovnání Arduino Nano vs Raspberry Pi Pico

Porovnání Raspberry Pi Pico a Arduino Nano lze provést z několika hledisek, jako jsou hardware, programování, komunita a ekosystém.

Raspberry Pi Pico je založen na mikrokontroléru RP2040, který má dvoujádrový ARM Cortex-M0+ procesor. Obsahuje 264 KB SRAM a 2 MB flash paměti. Velký rozdíl oproti Arduino je, že pico funguje na 3,3V logice a nano na 5V logice. Arduino Nano je založen na mikrokontroléru ATmega328P/ATmega328, který je 8bitový a založen na architektuře AVR. Obsahuje 2 KB SRAM a 32 KB flash paměti.

Raspberry Pi Pico má integrované PWM, SPI, I2C, UART, GPIO piny a další periferie.

Arduino Nano má digitální a analogové vstupy/výstupy, PWM, sériový port (UART) a další periferie.

Raspberry Pi Pico lze programovat v jazycích C/C++ nebo MicroPython, CPython, Arduino Core pro RP2040 a Rust. Arduino Nano je primárně programovatelný v jazyce C/C++ pomocí Arduino IDE, ale může být také použit s dalšími jazyky a prostředími.

Raspberry Pi Pico je novější a nemá tak rozsáhlou komunitu jako Arduino, ale rychle získává na popularitě a je podporován společností Raspberry Pi Foundation. Arduino Nano má velkou a aktivní komunitu uživatelů, rozsáhlý ekosystém knihoven a rozšíření, což usnadňuje vývoj a sdílení projektů. Arduino Nano bývá obvykle dostupný v cenovém rozmezí od 200 Kč do 700 Kč, v závislosti na verzi, ty nejlevnější jsou tzv. klony a ty dražší originály. Klony mohou mít potencionálně několik nevýhod jako jsou levnější součástky než originální, což způsobuje nižší životnost. Někdy se můžou vyskytnout problémy s kompatibilitou, mě osobně se stalo že se program někdy nenahrál. (Raspberry Pi Pico vs Arduino – Which Board to Choose?; 2023)

Raspberry Pi Pico je vhodný pro projekty vyžadující vyšší výpočetní výkon a komplexnější operace. Arduino Nano je vhodnější pro jednodušší projekty, kde není potřeba tak vysoký výpočetní výkon a kde je důležitější snadná integrace s existujícími knihovnami a rozšířeními.

## Pro přehlednost Tabulka

Charakteristika	Raspberry Pi Pico	Arduino Nano
Mikrokontrolér	čip RP2040 ARM Cortex-M0+	ATmega328P
počet jader	Dvě	Jedno
Šířka datové sběrnice	32bit	8bit
maximální frekvence	133MHz	16MHz
SRAM paměť	264KB	2KB
Flash paměť	2MB	32KB
programovací jazyky	C/C++, MicroPython, CircuitPython, Arduino Core pro RP2040, Rust	C/C++ pomocí Arduino IDE, scratch
digitální piny	26	22
Analogové vstupy	3	8
PWM piny	8	6
Komunikace	UART, SPI, I2C	UART, SPI, I2C
Operační napětí	3,3V	5V
Cena	obvykle od 100 Kč do 200 Kč	Od 200 Kč do 700 Kč, v závislosti na verzi
Velikost	51 × 21 mm	45 × 18 mm
Komunita	Rychle rostoucí komunita, podpora od Raspberry Pi Foundation	Velká a aktivní komunita uživatelů, rozsáhlý ekosystém knihoven a rozšíření

*Tab. 5. 1 Porovnání Nano a Pico*

(Zdroj: vlastní zdroj)

## 6 Příslušenství k mikropočítačům

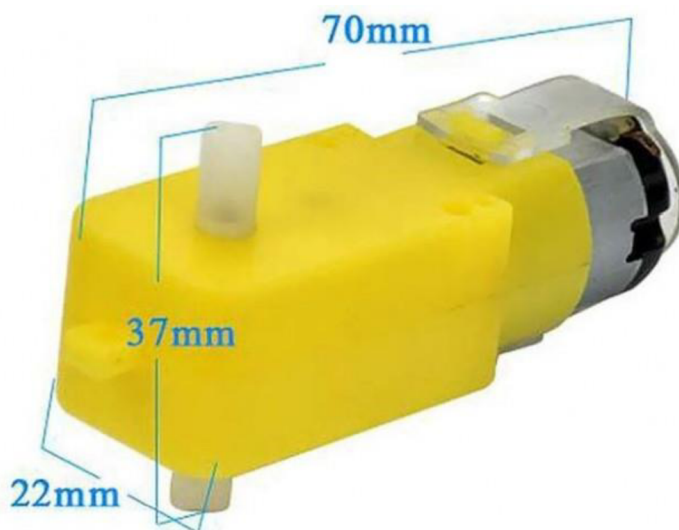
V této kapitole se zabýváme příslušenstvím, které lze zapojit k mikrokontrolerům. Konkrétně se jedná o motor, H-můstek, optický senzor. Nedílnou součástí této kapitoly je i nepájivé pole.

### 6.1 Motor

Elektromotory jsou základními komponentami v mnoha elektronických projektech, kde je potřeba mechanický pohyb. Při práci s platformami jako Arduino Nano a Raspberry Pi Pico je důležité rozumět základním typům elektromotorů, jejich vlastnostem a způsobům jejich ovládání.

Základními typy elektromotorů jsou stejnosměrné motory, krokové motory a servo motory. (Elektromotor, elektronický motor; 2019)

Stejnoseměrné motory jsou motory, které pracují na stejnosměrný proud. Jsou snadno ovladatelné pomocí PWM signálů, které jsou propojeny přes H-můstek a ten je až propojený s motorky. Napájení těchto motorů je od 3 až do 12 V. Převodový poměr mají 1:48. (MOTOR S PŘEVODOVKOU PRO ARDUINO ROBOTA; 2024) Používají se pro jednoduché projekty, jako jsou roboti či pohyblivé mechanismy. I v mém projektu využívám právě tyto motory.



*Obr. 6.1 Elektromotor s převodovkou*

(Zdroj: [https://xduino.cz/img/p/5/8/3/583-large\\_default.jpg](https://xduino.cz/img/p/5/8/3/583-large_default.jpg))

Krokové motory se pohybují v určitých krocích, což umožňuje jejich přesné řízení. Každý krok odpovídá pevně danému úhlu otáčení motoru, což umožňuje velmi přesnou

kontrolu nad polohou hřídele. Díky této přesnosti se krokové motory často využívají v aplikacích, kde je důležité mít plnou kontrolu nad pohybem. Ovládání krokového motoru se provádí pomocí driver modulu, který přijímá signály z Arduina či Raspberry. Využívají se v komplexnějších projektech. (MAŇÁK; 2014)

Servo motory kombinují DC motor s pozicovým senzorem, což umožňuje přesné natočení na daný úhel. Servo motory mají zabudovanou řídicí elektroniku, která přijímá PWM signály a podle nich nastavuje polohu hřídele. Tento přesný úhlový pohyb je ideální pro aplikace, kde je potřeba řídit přesnou polohu. Stejně jako krokové motory se dají využít v komplexnějších projektech. (Servo Motor Basics with Arduino; 2024)

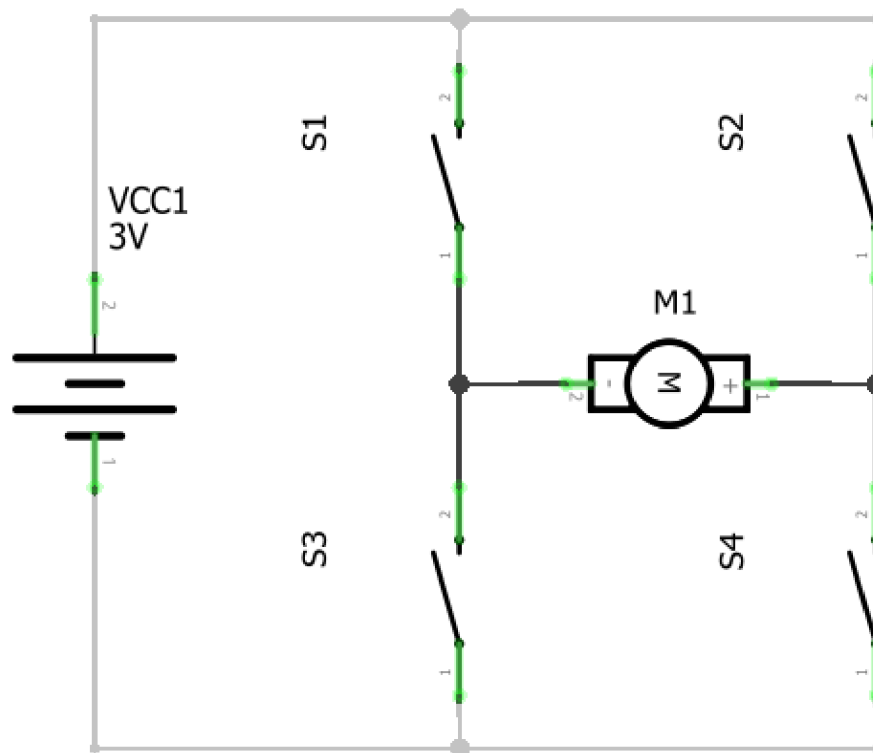
## 6.2 H-můstek

H-můstek je elektronický obvod, který slouží k řízení výstupních zařízení, jako jsou například motory nebo relé, pomocí malých řídicích signálů. Jeho hlavní vlastností je schopnost invertovat a zesílit signály. H-můstky se často používají pro řízení stejnosměrných motorů, kde mohou regulovat směr otáčení a rychlost motoru.

Principiálně se H-můstek skládá ze čtyř tranzistorů (obvykle jsou to MOSFETy nebo bipolární tranzistory), které jsou spojeny tak, aby umožňovaly průchod proudu ve dvou směrech. (H-můstek a otáčení motůrkem; 2024) Tato konfigurace umožňuje můstku přepínat mezi různými stavy:

- Proud teče z horního výstupu S1 na spodní výstup S4: Tento stav umožňuje motoru pohybovat se jedním směrem.
- Proud teče z horního výstupu S2 na horní výstup S3: Tento stav umožňuje motoru pohybovat se opačným směrem.
- Oba tranzistory na jedné straně jsou uzavřené: Tento stav zastavuje proud motoru a brzdí ho.

Schéma H-můstku je na obr. 6. 2



*Obr. 6. 2 Schéma H-můstek*

(Zdroj: [https://navody.dratek.cz/images/technikuv\\_blog/zmena-smeru-otaceni-dc-motoru/h\\_mustek\\_schema.png](https://navody.dratek.cz/images/technikuv_blog/zmena-smeru-otaceni-dc-motoru/h_mustek_schema.png))

### 6.3 Optický senzor

Optické senzory jsou klíčové komponenty v robotických projektech, kde je potřeba sledovat předdefinovanou dráhu nebo linii. Senzory na snímání čáry využívají optické technologie k detekci rozdílů v odrazu světla od povrchu. Typicky jsou sestaveny z infračervené (IR) LED diody a fotodetektoru (fototranzistoru nebo fotodiody). IR LED vysílá světlo, které se odráží od povrchu a je zachyceno fotodetektozem. Odražené světlo se liší podle barvy povrchu: světlý povrch odráží více světla než tmavý povrch. (IR Line Tracking Module; 2024) V mém projektu využívám tento typ senzoru, konkrétně MH-Sensor-Series.



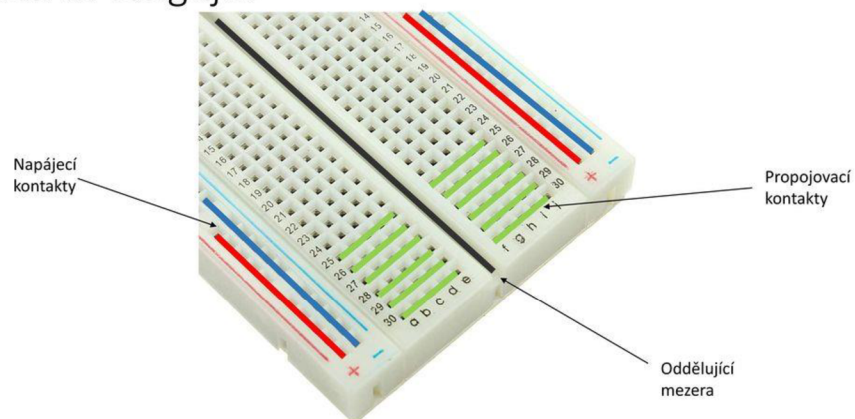
*Obr. 6. 3 Senzor na sledování čáry*

(Zdroj: [https://xduino.cz/index.php?id\\_product=212&id\\_product\\_attribute=0&rewrite=1-kanalovy-infra-modul-detekce-vzdalenosti&controller=product](https://xduino.cz/index.php?id_product=212&id_product_attribute=0&rewrite=1-kanalovy-infra-modul-detekce-vzdalenosti&controller=product))

### 6.4 Nepájivé pole

Nepájivé pole je nástroj pro stavbu a testování elektronických obvodů bez nutnosti pájení. Toto pole je velmi užitečné pro prototypování a experimentování s obvody, protože umožňuje snadné a rychlé sestavení a změny v obvodu. Nepájivé pole se skládá z napájecích lišt a z propojovacích lišt, viz obrázek.

Jak to funguje?



**Obr. 6. 4** *Nepájivé pole*

(Zdroj: [https://player.slideplayer.cz/70/12094265/slides/slide\\_5.jpg](https://player.slideplayer.cz/70/12094265/slides/slide_5.jpg))

Používá se k jednoduchému propojení komponentů, jako jsou rezistory, diody, tranzistory atd. Komponenty se jednoduše zasunují do děr v nepájivém poli a vodiče též, což umožňuje rychlé a snadné propojení. Napájení se přivádí do napájecích lišt a z nich se snadno distribuuje ke všem komponentům na poli. Protože spojení nejsou pevná lze snadno měnit a upravovat obvody, což se dá dobře využít ve výuce. (Nepájivé pole; 2024)



## **7 3D modelování a 3D tisk**

V této kapitole se budeme zabývat 3D modelováním a 3D tiskem, který jsem použil na výrobu robota v praktické části. Dozvíme se jaké programy jsou vhodné na použití na 3D modelování a též se dozvíme něco o 3D tisku a jeho nastavení.

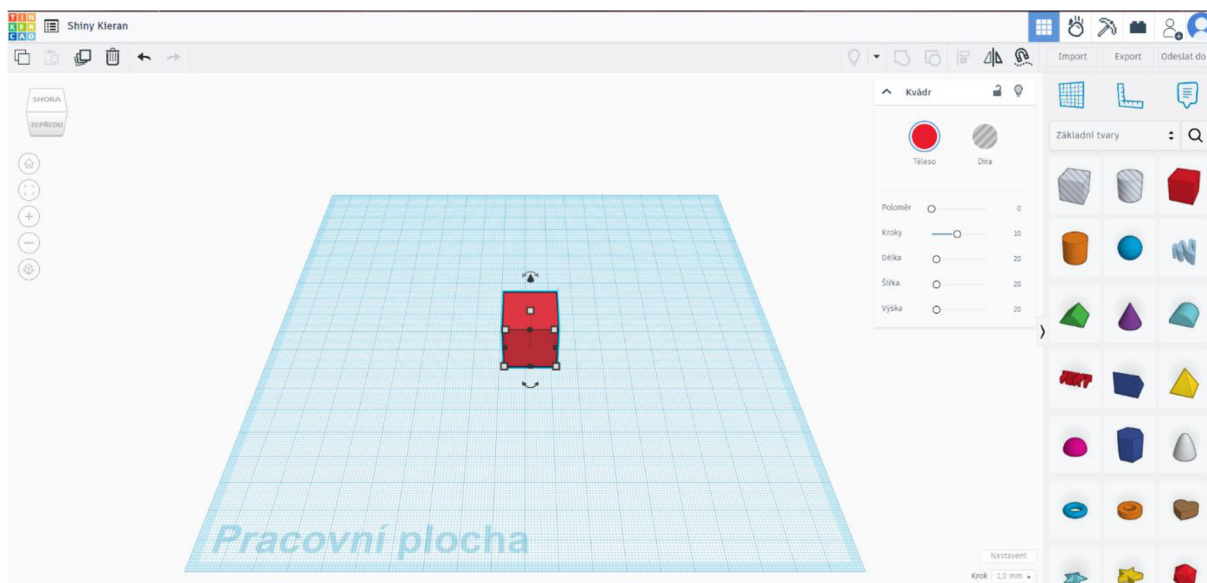
### **7.1 Programy pro 3D modelování**

3D modelování představuje klíčovou disciplínu v oblasti digitálního designu a vizualizace, která umožňuje tvorbu komplexních a realistických 3D objektů. V dnešní době, kdy vizualizace a interaktivní digitální prostředí získávají na významu, je schopnost efektivně modelovat a renderovat 3D objekty nezbytná pro mnoho profesních oblastí.

Modely téměř všeho si můžeme najít na internetu, avšak pouze jejich stažení neposkytuje dostatečnou znalost o procesu jejich vytvoření a manipulaci s nimi. Právě proto je zde důležité se seznámit s některými z předních programů určených k 3D modelování.

#### **7.1.1 Tinkercad**

Tinkercad je populární online platforma pro 3D modelování. Je to uživatelsky přívětivá online platforma vyvinutá společností Autodesk, která umožňuje uživatelům vytvářet jednoduché 3D modely přímo ve webovém prohlížeči. Je navržena s důrazem na jednoduchost a intuitivní ovládání, což činí z Tinkercadu ideální nástroj pro začátečníky v oblasti 3D modelování. Nespornou výhodou tinkercadu je, že nestojí vůbec nic a pro žáky je to skvělý základ, na který mohou později navázat s profesionálnějšími programy. (Programy pro 3D modelování; 2023)



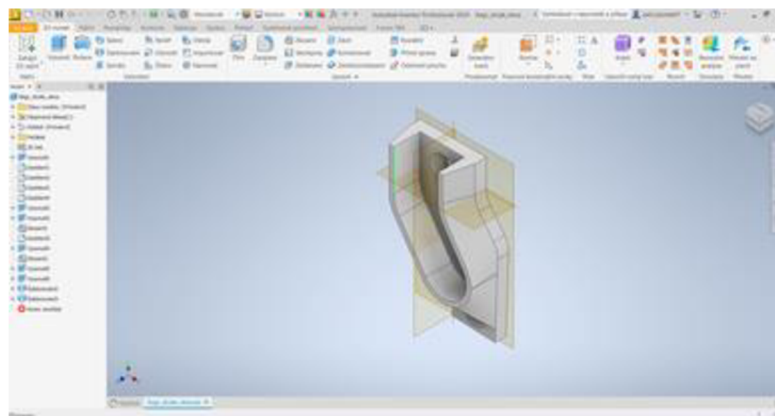
**Obr. 7. 1 Tinkercad**  
(Zdroj: vlastní zdroj)

### 7.1.2 Autodesk Inventor Professional

Autodesk Inventor Professional je vysoce výkonný CAD (Computer-Aided Design) software určený pro profesionály v mnoha oblastech. Jeho pokročilé nástroje pro 3D modelování umožňují vytvářet detailní a přesné digitální prototypy. Umožňuje vytvářet složité sestavy, které mohou obsahovat tisíce dílů. Jsou zde možné provádět simulace pohybů mechanismů. Umožňuje simulovat fyzikálního chování modelů, jako jsou statické a dynamické zatížení. Každý rok vychází nová verze s různými novinkami.

Oproti Tinkercadu jde o složitější software, kde je potřeba mít už nějaké znalosti v rámci 3D modelování.

Samotný program je celkem drahý, zhruba 70 000 Kč ročně, ale to platí v rámci osobního užití. Autodesk podporuje samotné vzdělávání a žákům, studentům i pedagogům je nabízen zdarma s roční licenci, která lze obnovovat. (Autodesk Inventor;2024; Studentská licence inventor; 2024)



**Obr. 6. 6 Autodesk Inventor Professionals**

(Zdroj: vlastní zdroj)

### **7.1.3 Fusion 360**

Autodesk Fusion 360 je pokročilý software, který nabízí výkonné nástroje pro 3D modelování, navržený pro profesionály i nadšence v oblasti produktového designu, strojírenství a výroby. Fusion 360 umožňuje uživatelům vytvářet, upravovat a analyzovat komplexní 3D modely v intuitivním a integrovaném prostředí. Mimo samotné vytváření 3D modelů umí program CAM (Computer-Aided Manufacturing) a CAE (Computer-Aided Engineering).

CAM je technologie využívající počítačový software k řízení strojů a procesů ve výrobě. CAM systémy automatizují výrobní procesy tím, že převádějí 3D modely a návrhy do instrukcí, které řídí CNC stroje (počítačem řízené obráběcí stroje), jako jsou frézky, soustruhy, laserové řezačky a 3D tiskárny.

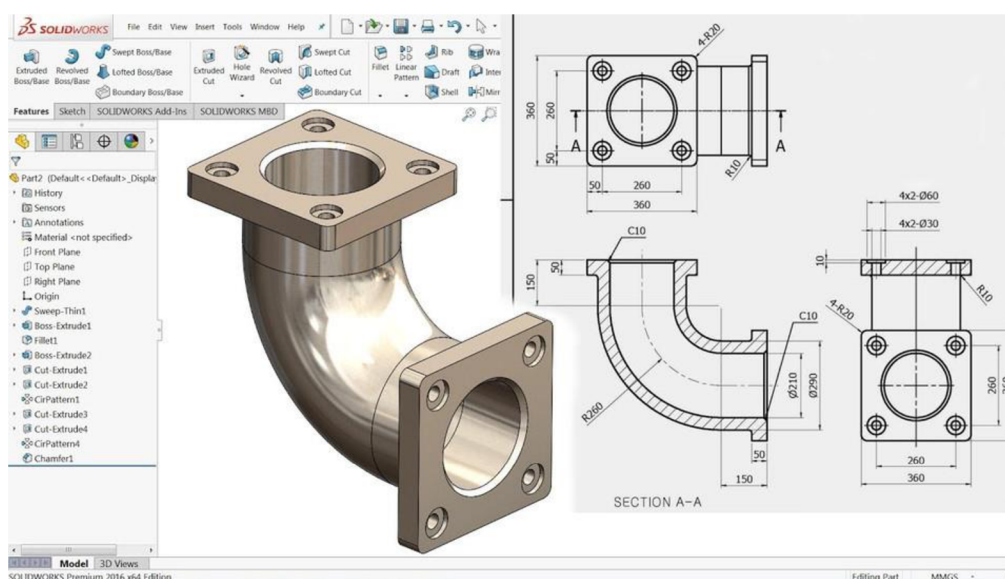
CAE je technologický přístup využívající počítačový software k provádění technických analýz a simulací. To zahrnuje širokou škálu funkcí, jako jsou simulace fyzikálních vlastností, termální analýzy, dynamické simulace, analýza proudění tekutin a strukturální analýzy. (CAD, CAE a CAM; 2021; Popis Fusion 360; 2024)

Stejně jako Inventor se za jeho používání platí zhruba 19 000 Kč ročně. Jsou určité výjimky, kdy lze získat licenci zdarma. Spadají do ní studenti, žáci a pedagogové, nadšenci a neziskové organizace. Stejně jako Inventor se hodí spíše pro pokročilejší uživatele. (Programy pro 3D modelování; 2023)

## 7.1.4 SolidWorks

Jako posledním zástupcem tu mám SolidWorks vyvinutý společností Dassault Systèmes, který je široce používán v různých odvětvích. SolidWorks je známý svou robustní sadou nástrojů pro 3D modelování, která umožňuje vytvářet detailní a přesné 3D modely. Podobně jako Inventor umožňuje vytvářet a spravovat složité sestavy. Obsahuje nástroje pro simulace pohybů, kontroly kolizí a mnoho dalších užitečných funkcí.

Podobně jako Inventor a Fusion 360 jde o placený software, kde lze získat studentské licence zadarmo. Svoji náročností se hodí pro žáky, kteří již mají zkušenosti s nějakým 3D modelováním. (What is SOLIDWORKS 3D CAD? ; 2024)



Obr. 6. 7 SolidWorks

(Zdroj:

<https://d2t1xqejof9utc.cloudfront.net/screenshots/pics/6316fe2a59832a5d8f073b6ee443d7c4/large.jpg>)

## 7.2 3D Tisk

3D tisk, známý také jako aditivní výroba, je revoluční technologie, která umožňuje vytvářet trojrozměrné objekty přidáváním materiálu vrstvu po vrstvě podle digitálního modelu. Tento proces se stává stále populárnějším díky své schopnosti rychle a efektivně vyrábět složité a detailní součástky přímo z 3D modelů.

Abychom mohli něco vůbec vytisknout, tak potřebujeme tyto věci. 3D model, ten můžeme získat několika způsoby viz. předchozí podkapitola, převést tento model v nějakém slicer softwaru, tzn. že se vygeneruje dráha, po které tiskárna bude tisknout a v poslední řadě je

potřeba 3D tiskárna s materiálem, z něhož bude objekt vyroben, známý jako filament. Filament může být z různých materiálů, jako jsou plasty, kovy či pryskyřice v závislosti na typu tiskárny a požadovaných vlastnostech výsledného objektu.

3D tisk přináší mnoho výhod, oproti jiným metodám výroby jako je rychlost výroby, schopnost vyrábět složité tvary, které by byly obtížné nebo nemožné vyrobit jinými metodami. Snižují náklady na výrobu u jednotlivých dílů či malých sérií. (Co je 3d tisk; 2024)

## 8 Využití ve škole

V této kapitole se budu zabývat informatikou ve školství, kde má klíčový význam v dnešní digitální společnosti.

V rámci školství se v od vzniku RVP ZV v roce 2004 dlouho nekonaly žádné změny ve výuce informatiky. V hodinách se většinou zaměřovalo především na základní principy práce s počítačem, jako je práce s operačním systémem, textovými editory, tabulkovými procesory a dalšími běžnými programy.

Až MŠMT ve spolupráci s pedagogickými fakultami a desítkami škol udělalo v letech 2018 až 2020 ověřování znalostí v rámci informatiky a digitální gramotnosti. (MŠMT spustilo web na podporu moderní výuky informatiky; 2022)

Na základě výsledků tohoto průzkumu ministerstvo provedlo revizi rámcového vzdělávacího programu. Na základě této revize by se měla informatika nově zaměřovat na rozvoj inforatického myšlení a porozumění základním principům digitálních technologií.

Na prvním stupni základního vzdělávání žáci skrze hry, experimenty a diskuse začínají chápat, jak se data a informace zaznamenávají, a objevují inforatické aspekty světa kolem nich. Postupně rozvíjejí schopnost popsat problém, analyzovat ho a hledat řešení, a v příhodném programovacím prostředí ověřují algoritmicke postupy. Informatika na tomto stupni spolu s dalšími obory zavádí základy uživatelských dovedností a pomáhá žákům porozumět digitálním technologiím a inforatickým konceptům. Důraz je také kladen na bezpečné zacházení s technologiemi a prevenci rizikového chování.

Na druhém stupni základního vzdělávání žáci aktivně tvoří, provádějí experimenty a testují své hypotézy. Hledají a navrhuji různá řešení, diskutují s ostatními a prohlubují si porozumění základním inforatickým konceptům a fungování digitálních technologií. Při analýze problémů se zaměřují na podstatné aspekty a učí se vytvářet a systematicky posuzovat postupy pro automatizaci a zpracování dat. Díky poznávání principů digitálních technologií chápou základy kódování a modelování a lépe chrání své soukromí, data a zařízení. Právě na druhém stupni se žáci mohou setkat s již zmíněným Arduinem či Raspberry. Je to skvělý způsob, jak ukázat žákům propojení mezi teorií a praxí. Dávají dohromady své obvody a programují je pomocí Scratch. (INFORMATIKA; 2021)

Vzdělávání v oblasti informatiky není jen o tom, jak používat počítače a aplikace, ale spíše o tom, jak porozumět základním principům informačních technologií, algoritmů, programování a digitálního myšlení.

Se vzděláváním informatiky, někdy označované jako nová informatika, na základní škole a digitálními kompetencemi se také využívá pojem informatické myšlení.

Informatické myšlení souvisí se zaměřením na problém, porozumění problému a řešení problému. V informatickém myšlení se uplatňují různé principy (Informatické myšlení; 2018):

- Pokus – omyl je cesta vpřed;
- Učíme se tím, že to děláme;
- Podstatná je vytrvalost;
- Spolupracujeme.

V této oblasti je pro základní školy velice nápomocen portál informatického myšlení, kde je možné blíže pojem pochopit a využívat vhodné připravené materiály do výuky. Jsou zde také modelové školní vzdělávací programy pro základní vzdělávání. Tyto programy jsou zde v několika verzích: ŠVP Oparně vpřed; ŠVP Nebojácně vpřed, ŠVP Progresivně vpřed; ŠVP Kreativně vpřed. (Informatické myšlení; 2018)

Zmíněné modelové školní vzdělávací programy se liší v organizačním a obsahovém vymezení vyučovacího předmětu, od kterého se dále odvíjí jeho náplň. Je zde možnost využití:

- pouze běžných počítačů;
- robotických stavebnic LEGO WeDo / Mindstorms EV3 nebo programovatelné desky Micro:bit;
- robotická hračka Bee-bot / Blue-bot a deska Tac-Tile
- edukační software pro algoritmizaci Emil.

Vždy je nutné uvažovat nad tím, jaké podmínky jsou v konkrétní škole poskytnuty. I když Arduino a Raspberry nejsou v těchto ŠVP uvedené, tak jistě mají své možné místo ve výuce na základní škole. I třeba v případném kroužku zaměřený na novu informatiku.

## 9 Programovací prostředí

V této části se budu zabývat programovacími prostředím vhodnými pro programování Arduino nano a Raspberry Pi Pico. Výběr správného programovacího prostředí může výrazně ovlivnit efektivitu a pohodlí při vývoji projektů. Porozumění těmto prostředím a jejich funkcím umožňuje maximálně využít potenciál obou těchto populárních mikrokontrolerových desek.

### 9.1 Scratch

Scratch je vizuální programovací jazyk a prostředí vyvinuté na Massachusettském technologickém institutu (MIT), které umožňuje uživatelům vytvářet interaktivní příběhy, hry a animace. Jedná se o intuitivní prostředí, ve kterém uživatelé manipulují s grafickými bloky reprezentujícími různé programové příkazy a operace. Tímto způsobem mohou vytvářet složité programy bez potřeby psaní kódu v textové podobě.

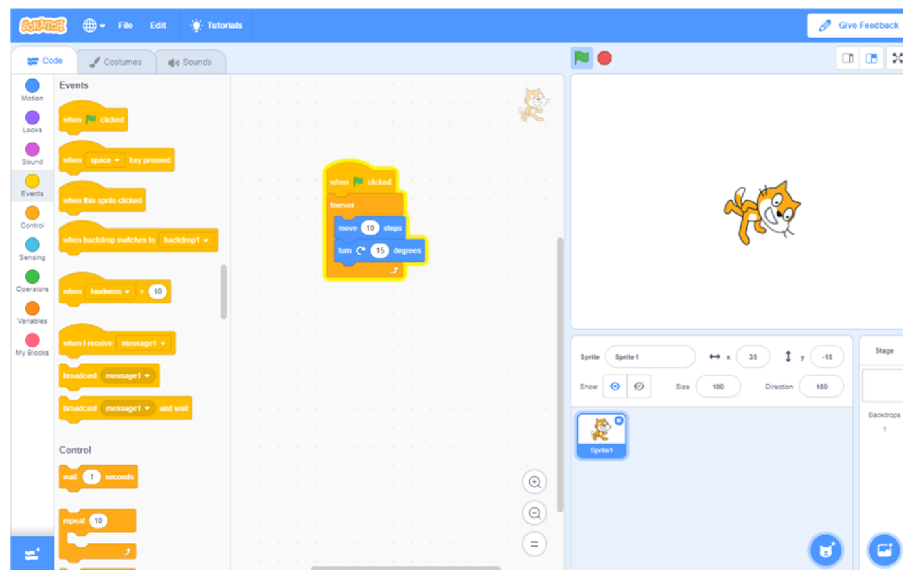
Scratch nabízí prostředky pro vytváření pohyblivých postaviček, interaktivních scén a reakcí na uživatelské vstupy. Uživatelé mohou definovat podmínky, smyčky, proměnné a další programové konstrukce pomocí grafických bloků, které jednoduše přetahují a spojují spolu.

Toto prostředí je ideální pro začátečníky v programování, děti i dospělé, protože umožňuje snadný a zábavný vstup do světa algoritmů a programování bez nutnosti předchozích znalostí. Scratch podporuje rozvoj kreativity, logického myšlení a řešení problémů a může sloužit jako skvělý vstupní bod pro další studium programování v pokročilejších jazycích. (Úvod do jazyka Scratch; 2024)

Pro programování Arduina pomocí Scratche je ale nejprve nutné nainstalovat speciální verzi Arduino IDE, o tomto prostředí se budeme bavit v další podkapitole, následně nainstalovat rozšíření Scratch for Arduino (S4A) do Scratche.

Bohužel jsem nenašel že by existovalo nějaké rozšíření pro Raspberry Pi Pico, ale existují programy, ve kterých se skládají bloky stejně jako ve Scratchi. Jako příklad tu mám Piper Make. Pomocí tohoto programu mohou uživatelé vytvářet interaktivní projekty i bez znalosti textového programování. (Drag-n-drop coding for Raspberry Pi Pico; 2021)





Obr. 9. 1 Scratch

(Zdroj: <https://store-images.s-microsoft.com/image/apps.48610.14205598541777108.d0471565-31cc-4061-beba-0830f2c9a787.8d42f056-8a80-43d0-bb5c-a7c571b01321?h=576>)

## 9.2 Arduino IDE

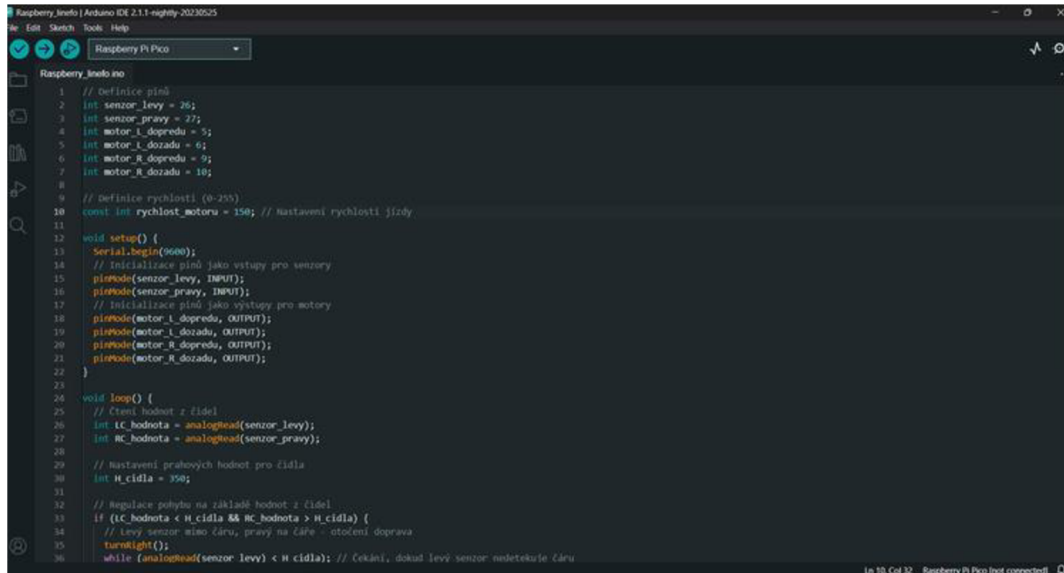
Arduino IDE (Integrated Development Environment) je softwarové prostředí navržené speciálně pro programování mikrokontrolérů z rodiny Arduino. Je to zdarma dostupný software, který poskytuje uživatelům prostředí pro psaní, editaci a nahrávání kódu do Arduino. Arduino IDE je k dispozici pro různé operační systémy, včetně Windows, macOS a Linux. (Arduino dokumentace; 2024)

Hlavními funkcemi Arduino IDE jsou:

- Editor kódu: Poskytuje prostředí pro psaní a editaci kódu v jazyce Wiring, což je zjednodušený dialekt jazyka C/C++, který je populární mezi uživateli Arduino.
- Správce knihoven: Umožňuje snadnou instalaci a aktualizaci knihoven, což jsou balíčky obsahující předdefinované funkce a procedury, které usnadňují programování různých zařízení a periférií.
- Nahrávání kódu: Arduino IDE umožňuje nahrávání připraveného kódu do desek Arduino pomocí USB kabelu. Tento proces se nazývá "nahrání" a provádí se jednoduše stisknutím tlačítka "Upload".
- Sériová monitorovací konzole: Poskytuje nástroj pro monitorování sériové komunikace mezi Arduino a počítačem. To umožňuje uživatelům zobrazovat

zprávy, výstupy nebo chyby generované jejich programem přímo v Arduino IDE. (Arduino ide; 2024)

Díky své jednoduchosti a uživatelsky přívětivému rozhraní je Arduino IDE vhodné pro začátečníky i pokročilé uživatele, kteří chtějí programovat a experimentovat s Arduinem.



```
1 // Definice pinů
2 int sensor_levy = 26;
3 int sensor_pravy = 27;
4 int motor_L_dopredu = 5;
5 int motor_L_dozadu = 6;
6 int motor_R_dopredu = 9;
7 int motor_R_dozadu = 10;
8
9 // Definice rychlosti (0-255)
10 const int rychlost_motoru = 150; // nastavení rychlosti jízdy
11
12 void setup() {
13   Serial.begin(9600);
14   // Inicializace pinů jako vstupy pro senzory
15   pinMode(sensor_levy, INPUT);
16   pinMode(sensor_pravy, INPUT);
17   // Inicializace pinů jako výstupy pro motory
18   pinMode(motor_L_dopredu, OUTPUT);
19   pinMode(motor_L_dozadu, OUTPUT);
20   pinMode(motor_R_dopredu, OUTPUT);
21   pinMode(motor_R_dozadu, OUTPUT);
22 }
23
24 void loop() {
25   // Čtení hodnot z čidel
26   int LC_hodnota = analogRead(sensor_levy);
27   int RC_hodnota = analogRead(sensor_pravy);
28
29   // Nastavení prahových hodnot pro čidla
30   int H_cidla = 350;
31
32   // Regulace pohybu na základě hodnot z čidel
33   if (LC_hodnota < H_cidla && RC_hodnota > H_cidla) {
34     // Levý senzor málo čáru, pravý na čáře - otočení doprava
35     digitalWrite();
36     while (analogRead(sensor_levy) < H_cidla); // Čekání, dokud levý senzor nedetekuje čáru
```

Obr. 9. 2 Arduino IDE  
(Zdroj: Vlastní zdroj)

### 9.3 Micropython

MicroPython je implementace programovacího jazyka Python optimalizovaná pro mikrokontroléry a vestavné systémy. Je navržena tak, aby byla co nejmenší a nejefektivnější, což umožňuje běh Pythonu na zařízeních s omezenou pamětí a výpočetním výkonem například pyboard, ESP8266, ESP32, Raspberry Pi Pico. Dá se použít i na Arduino nano ale jen na novější verze jako je Arduino Nano 33 IoT a Arduino Nano RP2040 Connect. (MicroPython.; 2024)

Klíčovými vlastnostmi jsou kompaktnost a efektivita, tzn. Že je optimalizován pro běh na hardwaru s omezenými prostředky. Zachovává mnoho základních vlastností a syntaxí z Python, což zjednodušuje jeho programování a umožňuje používat známé konstrukce a knihovny. MicroPython poskytuje interaktivní repl (read-eval-print loop) prostředí, kde mohou programátoři psát a testovat kód přímo na zařízení. MicroPython obsahuje mnoho standardních knihoven, které jsou k dispozici v Pythonu. Kromě toho má specifické moduly pro práci s hardwarem, jako jsou GPIO, I2C, SPI, UART, a další. (MicroPython ako vývojová platforma pre IoT; 2020)

## **Praktická část**

## 10 Ukázka zapojení a programů

Kapitola se zaměřuje na ukázkou typických zapojení elektrických obvodů s využitím Arduino a Raspberry. Vždy je doloženo schéma zapojení pro oba typy mikropočítačů. Uvádíme také programy, které by po zapojení měly plnit příslušnou funkci.

Zaměřili jsme se na konkrétní příklady:

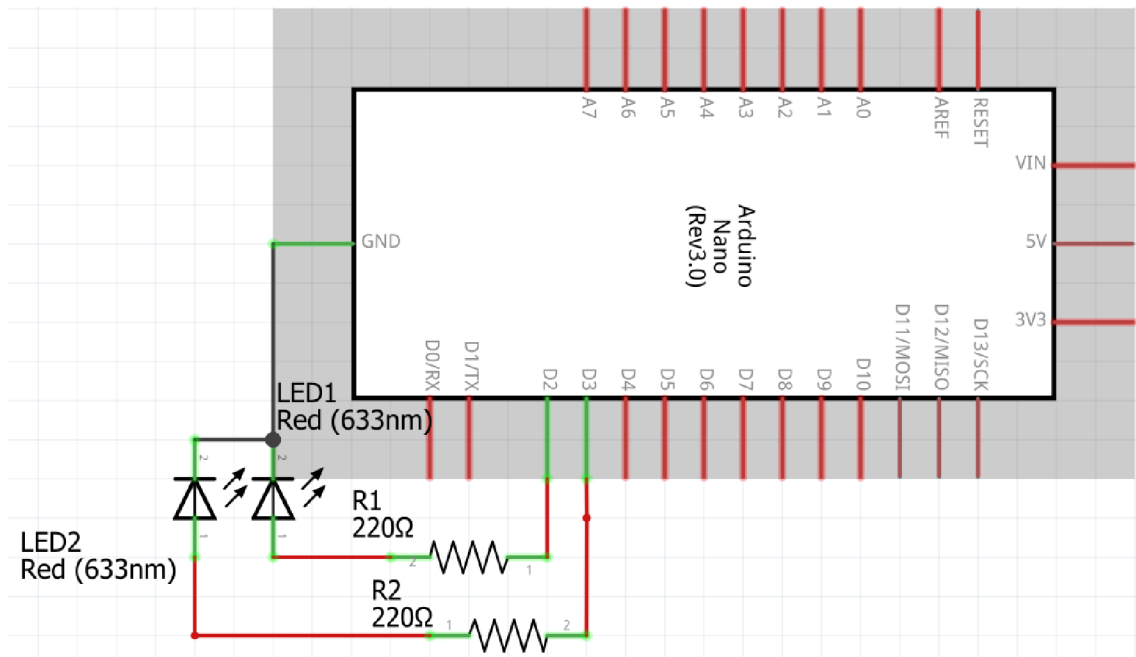
- železniční přejezd;
- házecí kostka;
- soumrakový spínač;
- dýchací dioda;
- postupné roztočení elektromotoru.

### 10.1 Železniční přejezd

Jako první tu mám jednoduché zapojení a program pro blikání dvou LED diod jako simulace železničního přejezdu. Je to krásná ukáзка toho, jak se dají využít digitální výstupy.

#### Zapojení pro Arduino nano

Na nepájivém poli si umístíme Arduino, následně z pinu 1 vyvedeme rezistor(220ohmů) na který napojíme LED diodu na anodu. Následně z katody vyvedeme drátek zpátky do arduina na pin GND. To samé provedeme pro druhou LED diodu s tím rozdílem, že ji vyvedeme z pinu 2.



**Obr. 9. 1** Zapojení železniční přejezd Nano  
(Zdroj: Vlastní zdroj)

Následně napíšu jednoduchý program v C++, který rozbliká LED diodu. Pro nahrání programu do arduina použiji program Arduino IDE.

```

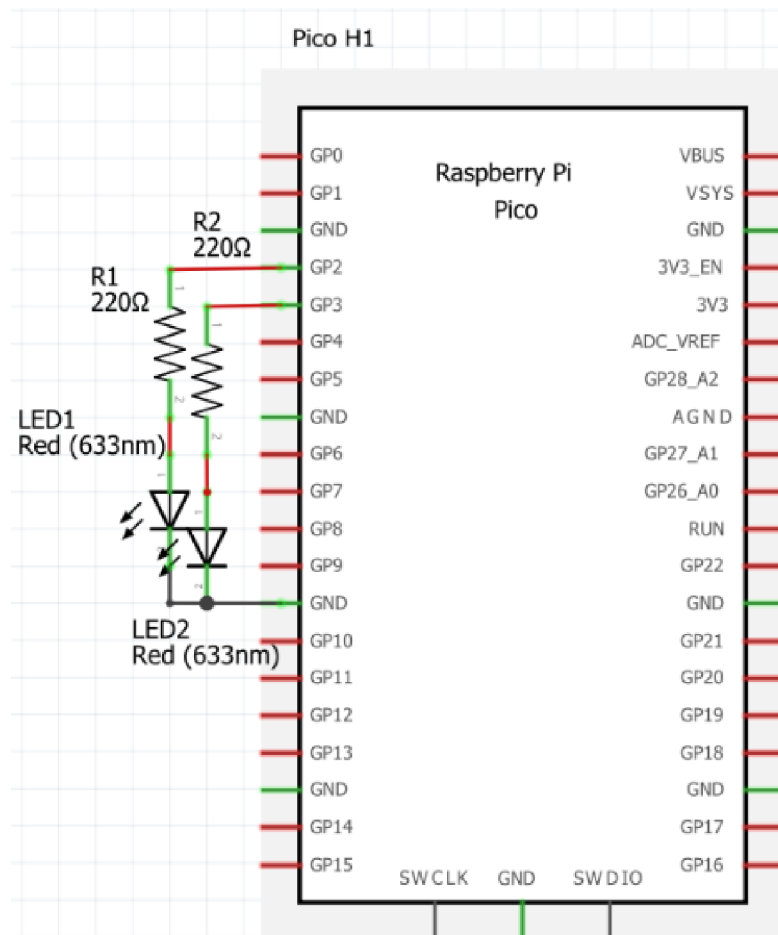
void setup()
{
pinMode(2, OUTPUT);      //První LED dioda
pinMode(3, OUTPUT);      //Druhá LED dioda
}

void loop()
{
digitalWrite(2, HIGH);    //Zapnutí první LED diody
digitalWrite(3, LOW);     //Vypnutí druhé LED diody
delay(1000);
digitalWrite(3, HIGH);    //Zapnutí druhé LED diody
digitalWrite(2, LOW);     //Vypnutí první LED diody
delay(1000);
}

```

## Zapojení pro Raspberry Pi Pico

Pro zapojení na raspberry pi pico použijí podobný postup, jak na arduino nano. Tak z pinu GP02 vyvedu rezistor (220 ohmů), který spojím s anodou na LED diodě. Následně z katody vyvedu drátek zpátky na pin GND. Stejný postup provedeme pro druhou LED diodu s tím rozdílem, že použijeme pin GP03.



Obr. 9. 2 Zapojení železniční přejezd Pico

(Zdroj: Vlastní zdroj)

Následně v MicroPythonu napiši a následně nahraji program do raspberry pi pico.

```
import machine
import utime

# Nastavení pinů pro LED diody
dioda1_pin = machine.Pin(2, machine.Pin.OUT)
dioda2_pin = machine.Pin(3, machine.Pin.OUT)

def blink_led(active_pin, inactive_pin):
    active_pin.on() # Zapne aktivní LED
    inactive_pin.off() # Vypne neaktivní LED
    utime.sleep(0.5) # Počká 0.5 sekundy
    active_pin.off() # Vypne aktivní LED
    utime.sleep(0.5) # Počká 0.5 sekundy

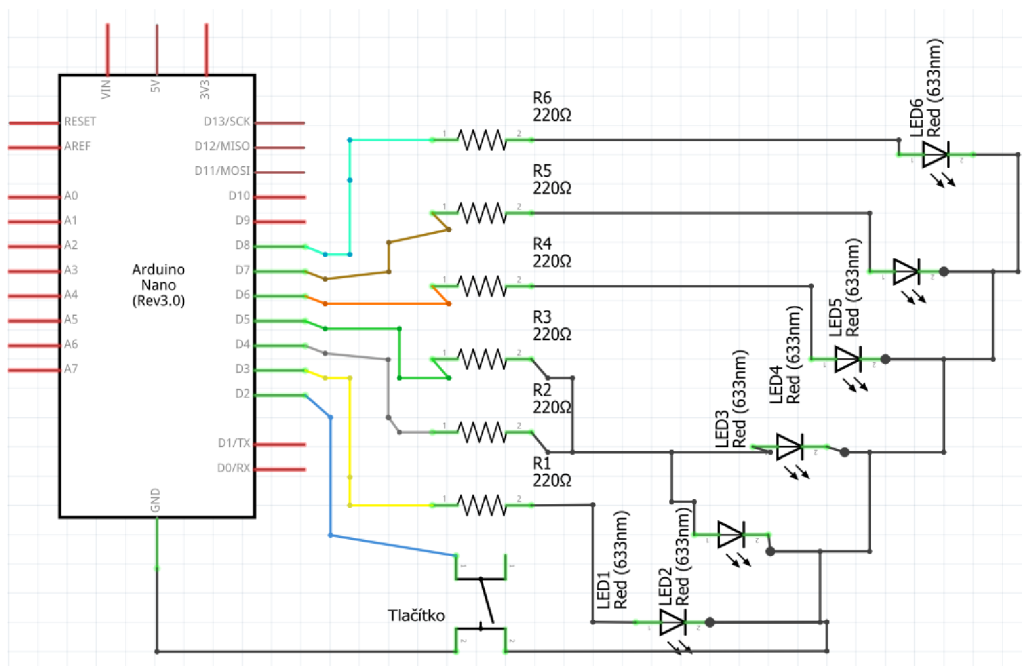
# Hlavní smyčka programu
while True:
    blink_led(dioda1_pin, dioda2_pin) # Blikání LED 1
    blink_led(dioda2_pin, dioda1_pin) # Blikání LED 2
```

## 10.2 Házecí kostka

Házecí kostka je ukázka toho, jak se dají kombinovat digitální vstupy a výstupy.

### Zapojení pro Arduino nano

Pro zapojení v arduinu použijeme piny dva až osm. Do pinu dva zapojíme tlačítko. Piny tři až osm použijeme pro zapojení LED diod, které budeme zapojovat stejně jako u železničního přejezdu.



*Obr. 9.3 Zapojení házecí kostka Nano*

(Zdroj: Vlastní zdroj)



```

const int buttonPin = 2; // Pin pro připojení tlačítka
const int ledPins[] = {3, 4, 5, 6, 7, 8}; // Piny pro připojení LED diod
const int numLEDs = 6; // Celkový počet LED diod

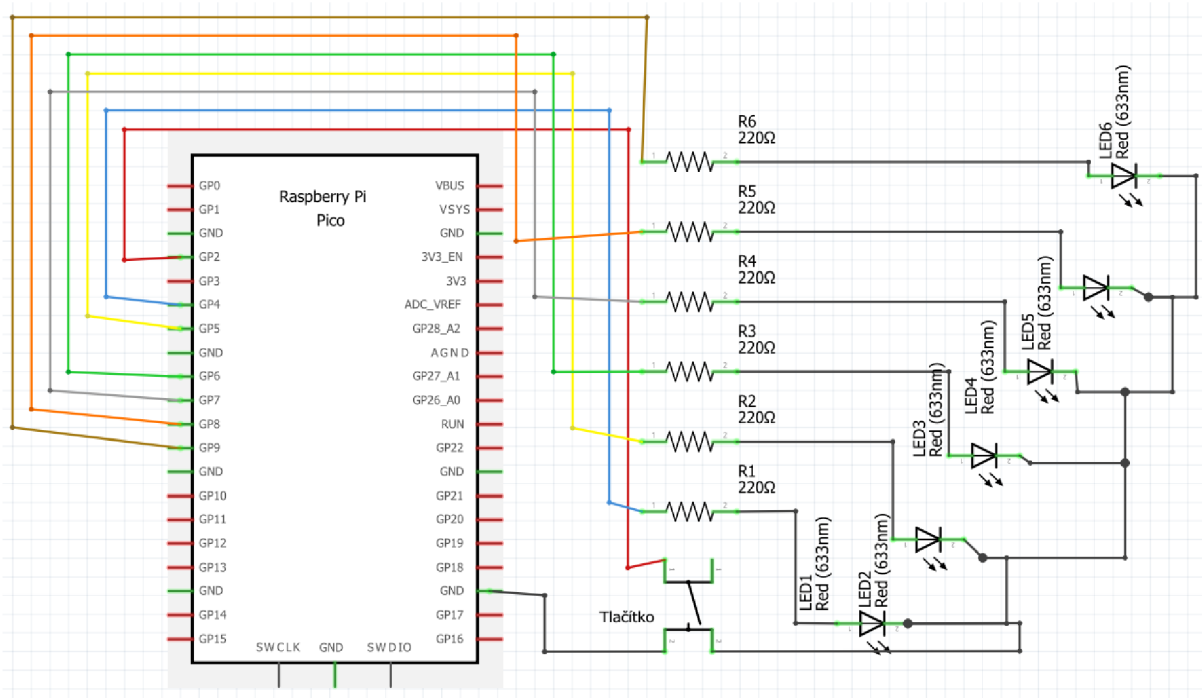
void setup() {
  pinMode(buttonPin, INPUT_PULLUP); // Nastavení tlačítka jako vstupu s pull-up rezistorem
  for (int i = 0; i < numLEDs; i++) {
    pinMode(ledPins[i], OUTPUT); // Nastavení všech LED diod jako výstupů
  }
}

void loop() {
  if (digitalRead(buttonPin) == LOW) { // Pokud je tlačítko stisknuto
    int numToLight = random(1, numLEDs + 1); // Náhodně určíme počet LED diod k rozsvícení (1 až numLEDs)
    for (int i = 0; i < numToLight; i++) {
      int randomLED = random(0, numLEDs); // Generování náhodného čísla od 0 do numLEDs - 1
      digitalWrite(ledPins[randomLED], HIGH); // Rozsvítí náhodnou LED
    }
    delay(3000); // Počká 3 sekundu
    for (int i = 0; i < numLEDs; i++) {
      digitalWrite(ledPins[i], LOW); // Zhasne všechny LED diody
    }
    delay(500); // Krátká pauza
  }
}

```

## Zapojení pro Raspberry Pi Pico

Pro Raspberry pi pico bude tlačítko zapojeno na GP02 a led diody na GP03- GP08, jak je vidět na schéma níže.



*Obr. 9. 4 Zapojení házečí kostka Pico*

(Zdroj: Vlastní zdroj)

```

import machine

import utime

import urandom

# Definice pinů
button_pin = machine.Pin(2, machine.Pin.IN, machine.Pin.PULL_UP)
led_pins = [machine.Pin(pin, machine.Pin.OUT) for pin in range(3, 9)] # Piny 3 až 8 pro LED diody

# Funkce pro rozsvícení náhodného počtu LED diod
def light_random_leds():
    num_to_light = urandom.randint(1, len(led_pins)) # Náhodný počet LED diod
    leds_to_light = urandom.sample(led_pins, num_to_light) # Náhodný výběr LED diod
    for led in leds_to_light:
        led.value(1) # Zapnutí LED diody

# Hlavní smyčka programu
while True:
    if button_pin.value() == 0: # Pokud je tlačítko stisknuto
        while button_pin.value() == 0: # Pokračuj ve svícení LED, dokud je tlačítko stisknuto
            light_random_leds() # Rozsvítí náhodné LED diody
        for led in led_pins:
            led.value(0) # Vypnutí všech LED diod po uvolnění tlačítka

```

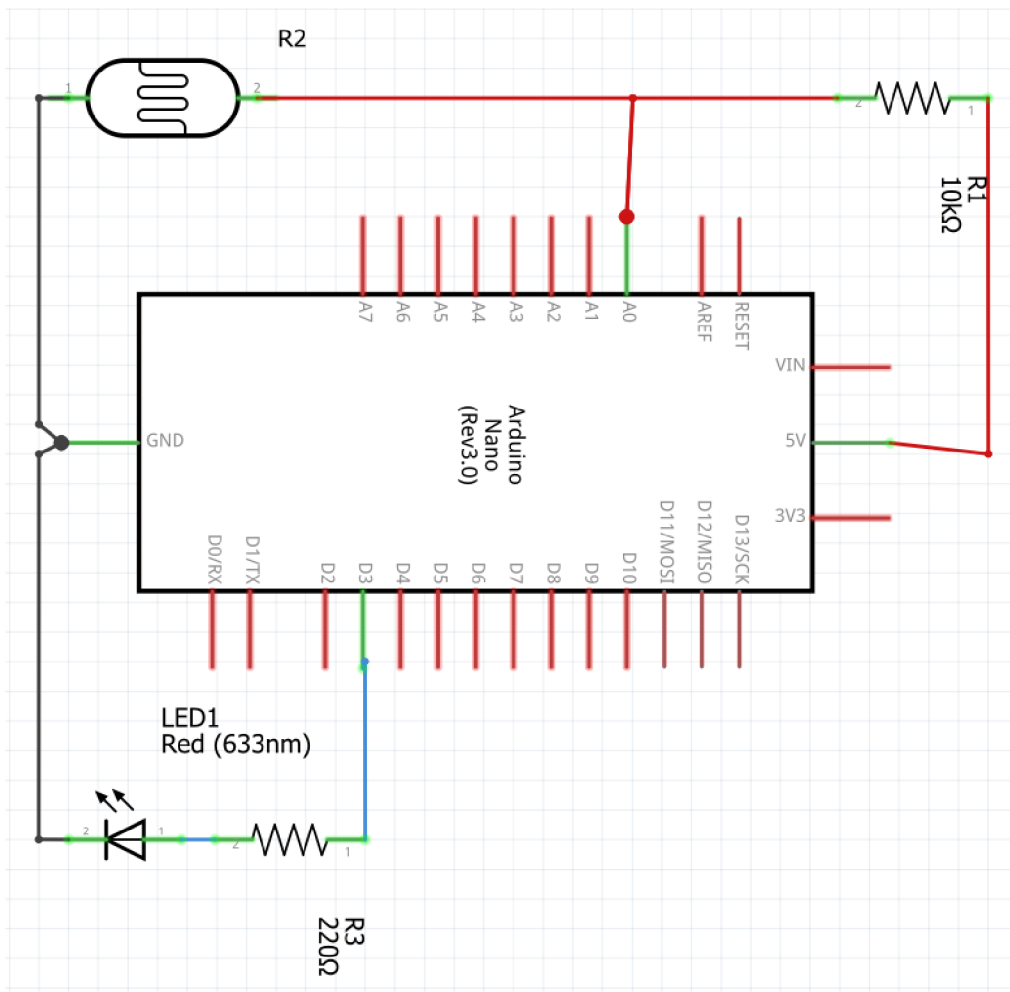
## 10.3 Soumrakový spínač

Další problematikou jsou analogové vstupy. Ty nám umožňují připojení analogových senzorů a dalších zařízení, které generují analogové signály, jako jsou například fotorezistory, termistory, potenciometry atd.

### Zapojení pro Arduino nano

Arduino Nano má několik pinů, které lze použít jako analogové vstupy, označené jako A0, A1, A2 atd. Tyto piny jsou schopny číst napětí v rozsahu od 0V do 5V a převádět je na číselnou hodnotu mezi 0 a 1023 (10bitový převod).

Jako skvělou ukázkou zapojení je soumrakový spínač. Pro jeho realizaci je potřeba fotorezistor, pár rezistorů a žárovku nebo LED diodu, která nám poslouží jako kontrola. Zapojení je jednoduché, prvně si zapojíme fotorezistor. Vytáhneme si 5V z arduina na lištu pro vedení el. proudu na nepájivém poli. Z lišty vyvedeme proud přes rezistor 10k $\Omega$  někam na nepájivé pole. Pak vezmeme drátek a nad nebo pod zapojený rezistor jej připojíme. Druhý konec dáme do Arduina na kterýkoliv analogový vstup například A0. Teď vezmeme fotorezistor a připojíme ho do lajny s rezistorem. Pak zapojíme LED diodu a do pinu 3 přes rezistor 220  $\Omega$ .



Obr. 9. 5 Zapojení Soumrakový spínač Nano

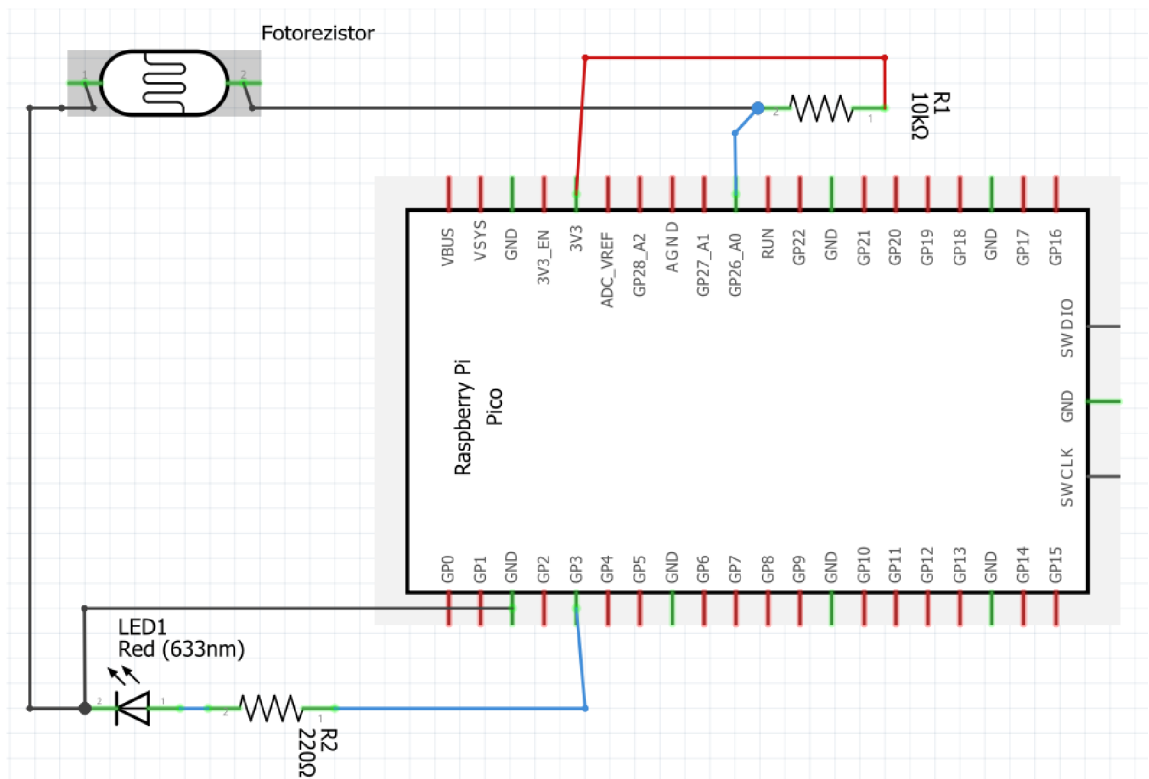
(Zdroj: Vlastní zdroj)

```
void setup() {
  pinMode(3, OUTPUT); // Nastaví pin 3 jako výstupní pin pro ovládání LED
}

void loop() {
  if (analogRead(A0) > 500) { // Pokud hodnota na analogovém pinu A0 je vyšší než 500
    digitalWrite(3, HIGH); // Zapne pin 3 (rozsvítí LED)
  } else { // Pokud hodnota na analogovém pinu A0 není vyšší než 500
    digitalWrite(3, LOW); // Vypne pin 3 (zhasne LED)
  }
}
```

## Zapojení pro Raspberry Pi Pico

Pro Raspberry pi pico použijeme podobné zapojení. Zapojíme z Raspberry 3,3V na lištu, odkud dáme rezistor  $5k\Omega$  a za rezistorem to vrátíme drátem do pinu GP26. Pokračujeme za rezistorem fotorezistorem a druhý konec zapojíme do GND. Ted zapojíme LED diodu stejně jak u arduina s tím rozdílem, že ji zapojíme do GP3.



Obr. 9. 6 Zapojení Soumrakový spínač Pico

(Zdroj: Vlastní zdroj)

```

import machine

import utime

# Definujeme pin pro připojení světelného senzoru (analogový vstup)

light_sensor_pin = machine.ADC(26) # použijeme GP26 pro analogový vstup

# Definujeme pin pro ovládání výstupního zařízení (např. LED)

output_pin = machine.Pin(3, machine.Pin.OUT)

# Hlavní smyčka programu

while True:

    # Přečteme hodnotu ze světelného senzoru

    light_intensity = light_sensor_pin.read_u16()

    # Porovnáme hodnotu s práhovou hodnotou

    if light_intensity > 20000: # intezitu můžeme dle potřeby měnit

        output_pin.on() # Zapne výstupní pin (LED nebo jiné zařízení)

    else:

        output_pin.off() # Vypne výstupní pin

```

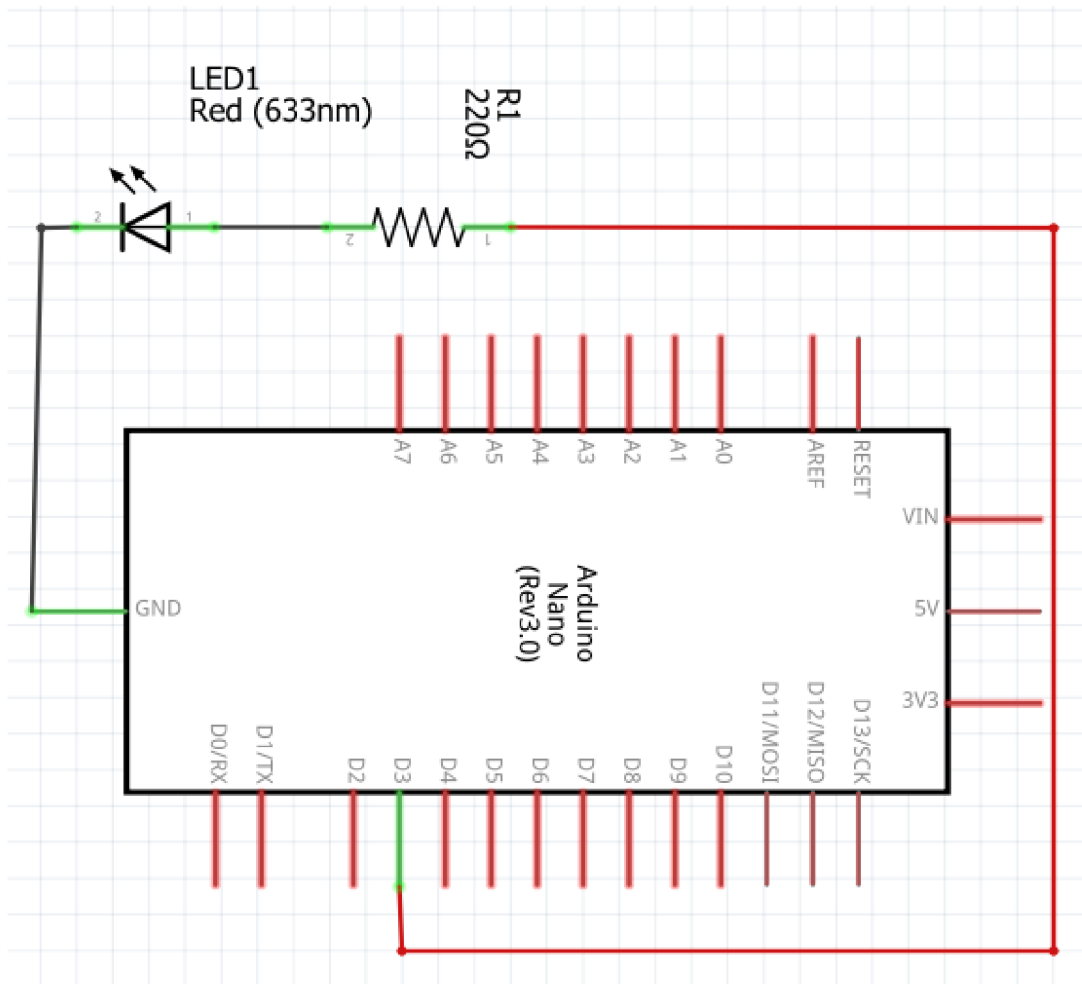
## 10.4 Dýchající dioda

V této části si ukážeme využití analogových výstupů (PWM pinů). Princip spočívá v tom, že digitální signál je periodicky přepínán mezi logickou hodnotou HIGH (1) a LOW (0) s určitou frekvencí a variabilní délkou impulzů. Například, když používáte PWM pro řízení jasu LED diody, rychlost motoru nebo polohy servo motoru, můžeme pomocí softwarového řízení nastavit požadovanou hodnotu. Tím získáme efekt podobný analogovému signálu, i když ve skutečnosti používáte digitální signál.

Hezkým příkladem z praxe je dýchající dioda, tzn. Dioda, která se bude postupně rozsvěcovat a zase zhasínat.

## Zapojení pro Arduino nano

Zapojení v arduinu je jednoduché. Potřebujeme jen Arduino, rezistor a LED diodu, kterou spojíme s LED diodou a připojíme k arduinu na pin s podporou PWM signálů. Následně nahrajeme do arduina následující program a máme hotovo.



Obr. 9. 7 Zapojení Dýchající dioda Nano

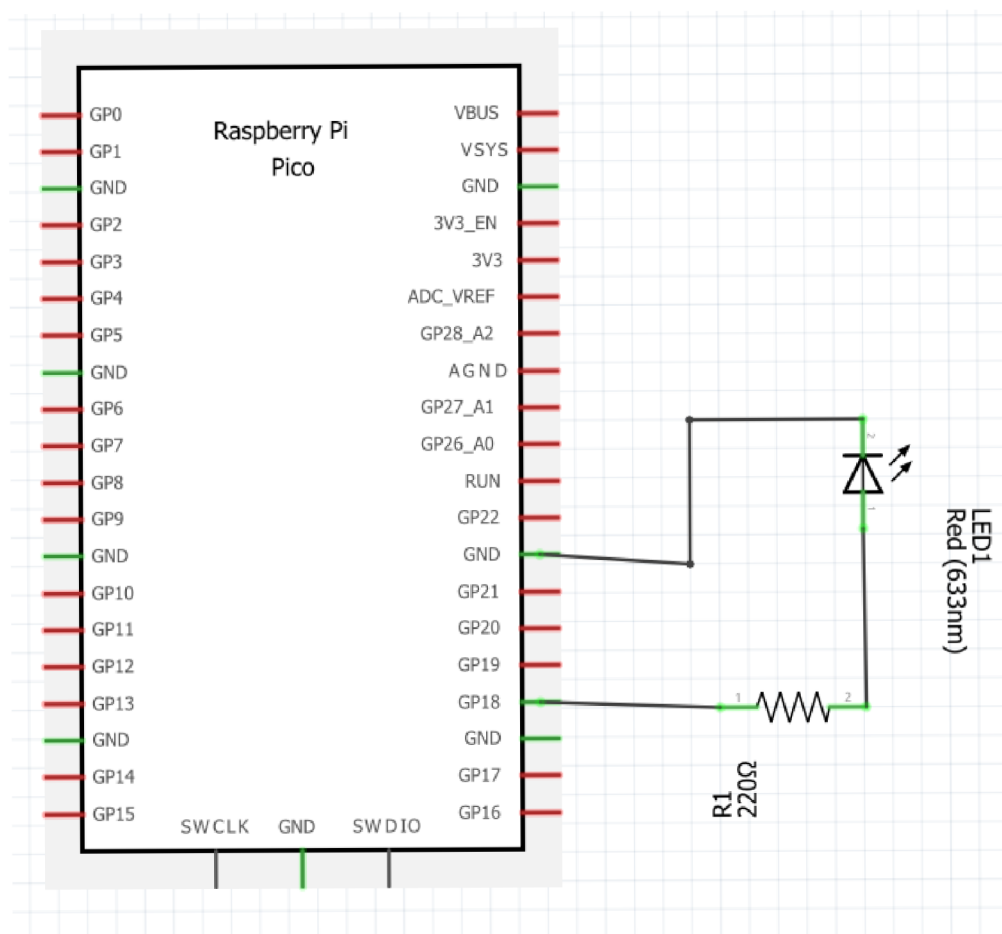
(Zdroj: Vlastní zdroj)



```
// Definice pinu pro připojení LED
const int ledPin = 3; // Použijte pin s podporou PWM
void setup() {
  // Nastavení pinu jako výstupní s podporou PWM
  pinMode(ledPin, OUTPUT);
}
void loop() {
  // Postupně zvyšujeme jas LED
  for (int brightness = 0; brightness <= 255; brightness++) {
    analogWrite(ledPin, brightness); // Nastaví jas LED
    delay(10); // Čekáme na chvíli pro efekt dýchání
  }
  // Postupně snižujeme jas LED
  for (int brightness = 255; brightness >= 0; brightness--) {
    analogWrite(ledPin, brightness); // Nastaví jas LED
    delay(10); // Čekáme na chvíli pro efekt dýchání
  }
}
```

## Zapojení pro Raspberry Pi Pico

Pro zapojení na Raspberry pi pico použijeme stejné zapojení jako na arduinu s tím rozdílem, že použijeme pin GP18.



*Obr. 9. 8 Zapojení Dýchající dioda Pico*

(Zdroj: Vlastní zdroj)

```
import machine
import utime

# Definice pinu pro připojení LED
led_pin = machine.Pin(18, machine.Pin.OUT)

# Inicializace PWM pro řízení jasu LED
led_pwm = machine.PWM(led_pin)

while True:

    # Postupně zvyšujeme jas LED
    for brightness in range(0, 65535):
        led_pwm.duty_u16(brightness) # Nastaví jas LED
        utime.sleep_ms(10) # Čekáme na chvíli pro efekt dýchání

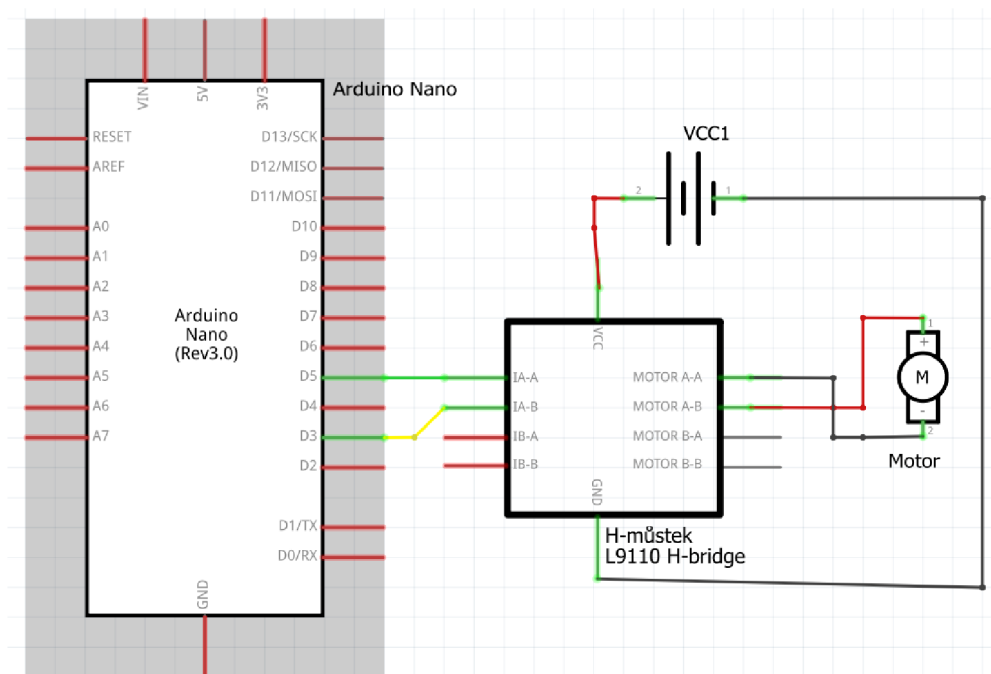
    # Postupně snižujeme jas LED
    for brightness in range(65535, 0, -1):
        led_pwm.duty_u16(brightness) # Nastaví jas LED
        utime.sleep_ms(10) # Čekáme na chvíli pro efekt dýchání
```

## 10.5 Postupné roztočení elektromotoru

V této podkapitole si ukážeme zapojení a kód pro postupné roztočení elektromotorů na jednu a následně na druhou stranu. Příklad zapojení je znázorněn na obrázcích níže jak pro Arduino nano tak i pro Raspberry Pi pico.

### Zapojení pro Arduino nano

Zapojení pro Arduino vypadá tak, že vezmeme motorek, ten připojíme na H-můstek, který následně připojíme k arduinu. Je vhodné napájet H-můstek odděleně od arduina, ne z něj, z praxe vím, že to může jinak zlobit, například, že se vůbec nerozjede.



*Obr. 9. 9 Zapojení Postupní roztočení elektromotoru Nano*

(Zdroj: Vlastní zdroj)

```

// Definice pinů pro řízení H-můstku
const int in1Pin = 5; // Připojeno k IN1 pinu H-můstku
const int in2Pin = 3; // Připojeno k IN2 pinu H-můstku

// Definice proměnné pro PWM signál
int motorSpeed = 0; // Počáteční rychlost motoru

void setup() {
  // Nastavení pinů jako výstupy
  pinMode(in1Pin, OUTPUT);
  pinMode(in2Pin, OUTPUT);}

void loop() {
  // Postupné roztočení motoru jedním směrem
  for (motorSpeed = 0; motorSpeed <= 255; motorSpeed++) {
    analogWrite(in1Pin, motorSpeed);
    analogWrite(in2Pin, 0);
    delay(10); // Krátká pauza pro postupnost }
  for (motorSpeed = 255; motorSpeed >= 0; motorSpeed--) {
    analogWrite(in1Pin, motorSpeed);
    analogWrite(in2Pin, 0);
    delay(10); // Krátká pauza pro postupnost }

  // Krátké zastavení motoru
  analogWrite(in1Pin, 0);
  analogWrite(in2Pin, 0);
  delay(1000); // Počkej 1 sekundu

  // Postupné roztočení motoru opačným směrem
  for (motorSpeed = 0; motorSpeed <= 255; motorSpeed++) {
    analogWrite(in1Pin, 0);
    analogWrite(in2Pin, motorSpeed);
    delay(10); // Krátká pauza pro postupnost }

```

```

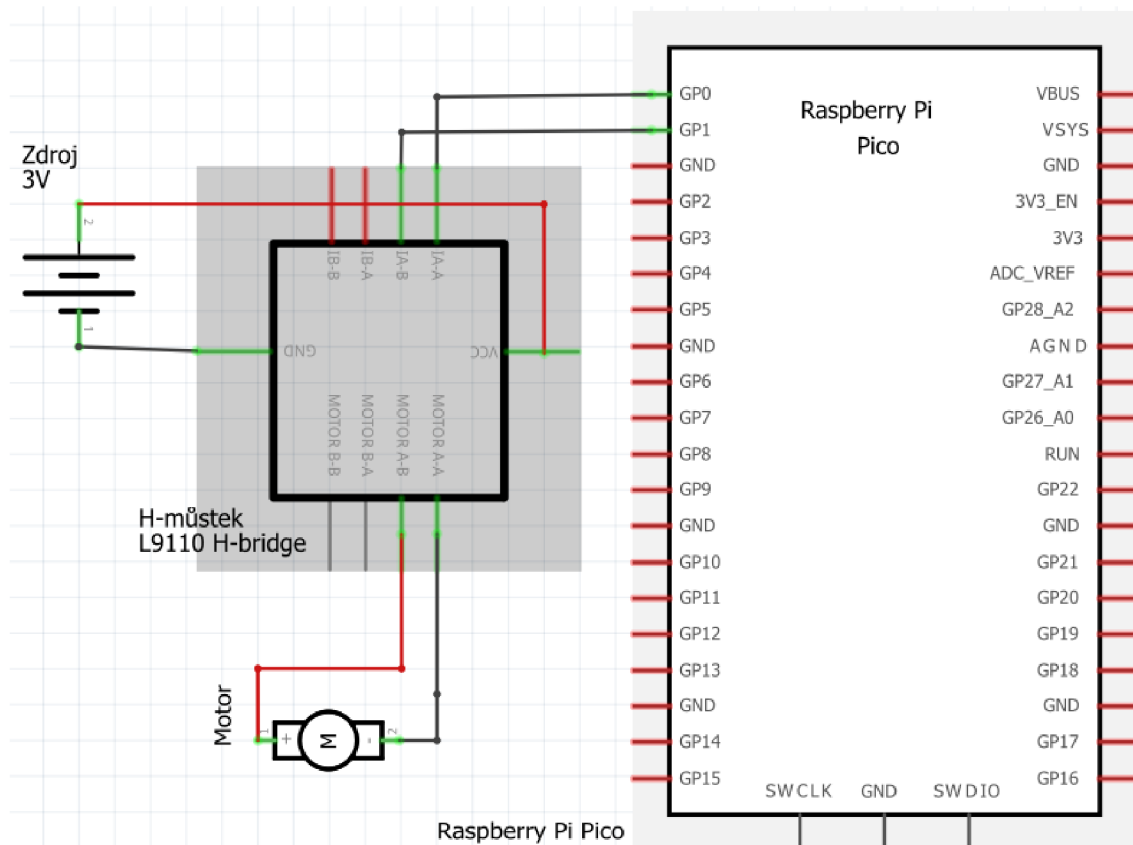
for (motorSpeed = 255; motorSpeed >= 0; motorSpeed--) {
    analogWrite(in1Pin, 0);
    analogWrite(in2Pin, motorSpeed);

    delay(10); // Krátká pauza pro postupnost }

// Krátké zastavení motoru
analogWrite(in1Pin, 0);
analogWrite(in2Pin, 0);
delay(1000); // Počkej 1 sekundu }

```

### Zapojení pro Raspberry Pi Pico



Obr. 9. 10 Zapojení Postupní roztočení elektromotoru Pico

(Zdroj: Vlastní zdroj)

```

from machine import Pin, PWM
from time import sleep

# Definice pinů pro řízení H-můstku
in1Pin = PWM(Pin(1)) # Připojeno k IN1 pinu H-můstku
in2Pin = PWM(Pin(2)) # Připojeno k IN2 pinu H-můstku

# Definice proměnné pro PWM signál
motorSpeed = 0 # Počáteční rychlost motoru

# Nastavení frekvence PWM signálu
in1Pin.freq(1000)
in2Pin.freq(1000)

# Nastavení PWM signálů na nulu
in1Pin.duty_u16(0)
in2Pin.duty_u16(0)

while True:

    # Postupné roztočení motoru jedním směrem
    for motorSpeed in range(0, 65535, 256):
        in1Pin.duty_u16(motorSpeed)
        in2Pin.duty_u16(0)
        sleep(0.1) # Krátká pauza pro postupnost

    # Postupné roztočení motoru opačným směrem
    for motorSpeed in range(65535, 0, -256):
        in1Pin.duty_u16(0)
        in2Pin.duty_u16(motorSpeed)
        sleep(0.1) # Krátká pauza pro postupnost

```

## 11 Robot

Ted' všechny tyto funkce dám dohromady a vzniknou nám dva roboti. Jeden bude mít hlavní část Arduino nano a druhý Raspberry Pi Pico. Budou fungovat na jednoduchém principu dvoustavové regulace, tzn. zapnuto a vypnuto. K robotům použiji bitBeam4 což je open-source konstrukční stavebnice pro vytváření různých mechanických a robotických konstrukcí. Jedná se o sadu standardizovaných dílů, které lze kombinovat a spojovat do různých tvarů a funkcí. Bitbeam umožňuje vytvářet různé konstrukce, jako jsou roboti, modely, herní hračky a další.

Bitbeam je oblíbeným nástrojem v oblasti vzdělávání a hobby projektů, protože umožňuje uživatelům snadno a levně vytvářet vlastní mechanické konstrukce. Systém je otevřený a flexibilní, což znamená, že uživatelé mohou upravovat a rozšiřovat své modely podle svých potřeb a představ. Tyto díly jsem vytiskl na 3D tiskárně konkrétně na tiskárně od společnosti Creality. Typ tiskárny, co jsem použil je Ender 3. Slicer, který jsem použil je PrusaSlicer. Do tohoto sliceru jsem ale musel přidat svoji 3D tiskárnu, protože původně je tento slicer zaměřený především na 3D tiskárny od Prusy. Po vytištění jednotlivých dílů jsem si pořídil ještě potřebné šroubky a následně jsem se pustil do skládání robotka. (Slicer: Příprava modelu pro tis; 2022)

Nejdříve jsem na motorky umístil drátky a zajistil jsem je vytištěnou součástkou, tak aby nevypadly. Následně jsem na tyto motorky namontoval držáčky. Pak jsem vzal základnu, na kterou jsem namontoval motorky. Dalším krokem montáže byly čidla, se kterých se sundala originální krytka, co je u snímačů a nahradil jsem ji novou krytkou, aby čidla lépe snímaly dráhu a také k jednodušší montáži. Jednotlivé piny na čidlech jsem ohnul asi o 45 stupňů dozadu. Potom jsem vzal šroubek, který jsem protáhl plastovou podložkou a skrze čidlo a namontoval jsem ho na základnu k motorům. Tyto čidla jsem umístil 2 vedle sebe. Do čidel jsem zapojit drátky na piny VCC, GND a A0, je to důležité zapojit, dokud je k nim dobrý přístup.

Dalším krokem je namontování zadní části, ve které je třetí bod robotka skleněnka, případně by se dala vytisknout i kulička, která by tam pasovala, ale já jsem si zvolil skleněnku. Následně jsem namontoval vrchní část, na které bude zbytek elektroniky a mikrokontroler.

Jako poslední část je potřeba namontovat kolečka na motorky. Tato část má čtyři části, které se do sebe poskládají a zajistí šroubkem a maticí a následně se nacvaknou na motorky. V tento moment můžeme začít dávat dohromady zbytek zapojení.

Základ pro toto zapojení bude nepájivé pole. Na ní bude umístěno Arduino nano na jednom robotu a na druhém bude Raspberry Pi Pico. Na nepájivé pole si zapojíme piny na lišty označené + a – z baterky nebo powerbanky. Z této lišty zapojíme na pin VIN kladné napětí a na

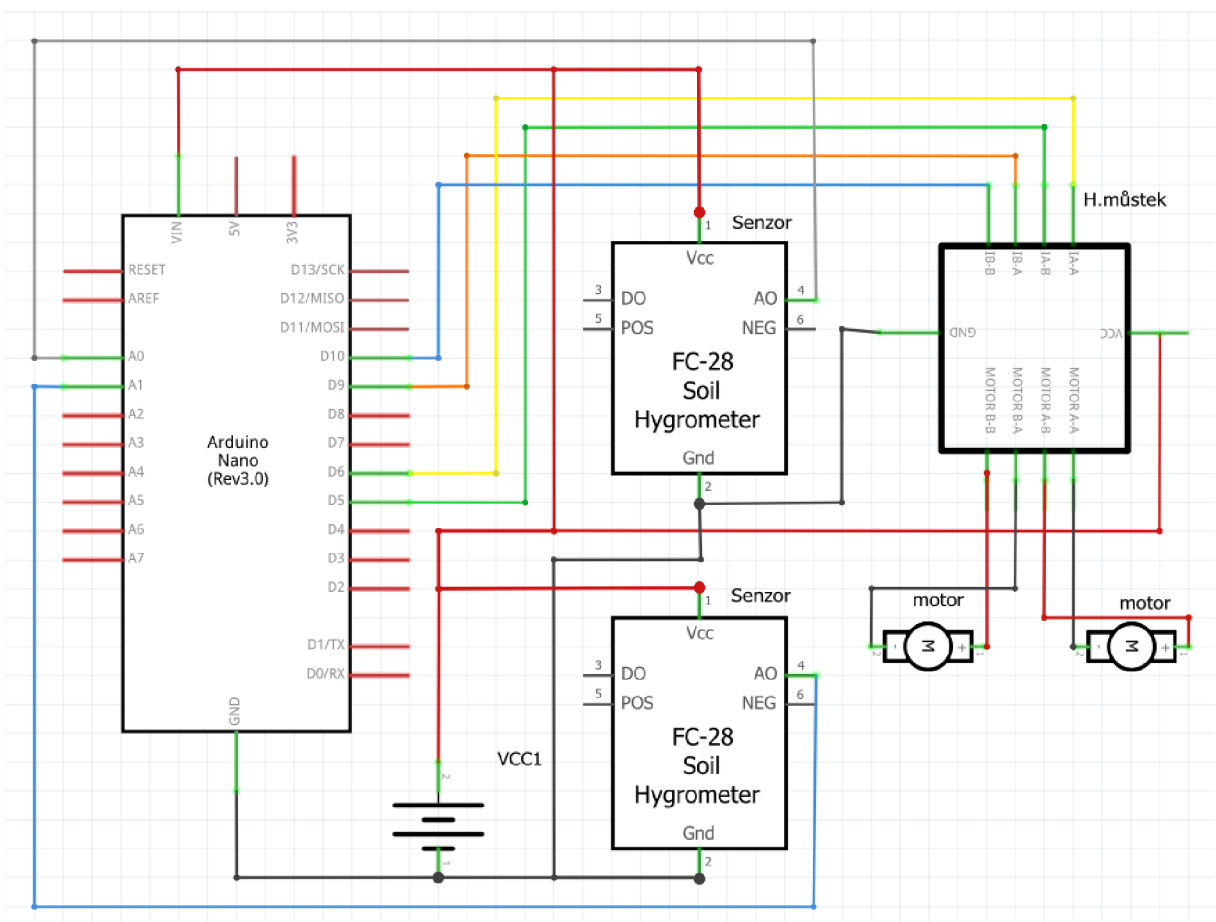


pin GND záporné. Následně dopojíme čidla, od kterých máme už vyvedené drátky. Drátek, co vede z pozice A0 na čidlu zapojíme do jakéhokoliv analogového vstupu, v mém případě jsou to piny A0 a A1. Další pin GND zapojíme na nepájivém poli na lištu s označením mínus a poslední pin VCC připojíme na nepájivém poli na lištu označenou plusem.

Ted' zapojím motorky, které jednoduše napojím do H-můstku. Ten následně připojím. Piny B-1A, B-1B, A-1A a A-1B připojím na PWM piny na Arduino, v mém případě piny D5, D6, D9 a D10. Zbývající piny GND zapojíme na nepájivé pole na lištu označenou mínusem a VCC na nepájivé pole na lištu označenou plusem.

Zapojení na Raspberry Pi Pico je stejné jen se používají piny GP 26, GP 27 pro senzory a GP5, GP6, GP9, GP10 pro ovládání motorků. Programy jsem udělal jak pro Arduino Nano tak Raspberry Pi Pico co možná nejpodobnější, abych viděl jestli se budou chovat rozdílně na dráze.

### Schéma zapojení a program pro Arduino Nano



*Obr. 10.1 Zapojení Robot Nano*

(Zdroj: Vlastní zdroj)

```

// Definice pinů
int senzor_levy = A0;
int senzor_pravy = A1;
int motor_L_dopredu = 5;
int motor_L_dozadu = 6;
int motor_R_dopredu = 9;
int motor_R_dozadu = 10;

void setup() {
  Serial.begin(9600);
  // Inicializace pinů jako vstupy pro senzory
  pinMode(senzor_levy, INPUT);
  pinMode(senzor_pravy, INPUT);
  // Inicializace pinů jako výstupy pro motory
  pinMode(motor_L_dopredu, OUTPUT);
  pinMode(motor_L_dozadu, OUTPUT);
  pinMode(motor_R_dopredu, OUTPUT);
  pinMode(motor_R_dozadu, OUTPUT);
}

void loop() {
  // Čtení hodnot z čidel
  int LS_hodnota = analogRead(senzor_levy);
  int RS_hodnota = analogRead(senzor_pravy);

  // Nastavení prahových hodnot pro čidla
  int H_cidla = 350;

```

```

// Regulace pohybu na základě hodnot z čidel
if (LS_hodnota < H_cidla && RS_hodnota > H_cidla) {
    // Levý senzor mimo čáru, pravý na čáře - otočení doprava
    turnRight();

    while (analogRead(senzor_levy) < H_cidla); // Čekání, dokud levý senzor nedetekuje čáru
} else if (LS_hodnota > H_cidla && RS_hodnota < H_cidla) {
    // Levý senzor na čáře, pravý mimo čáru - otočení doleva
    turnLeft();

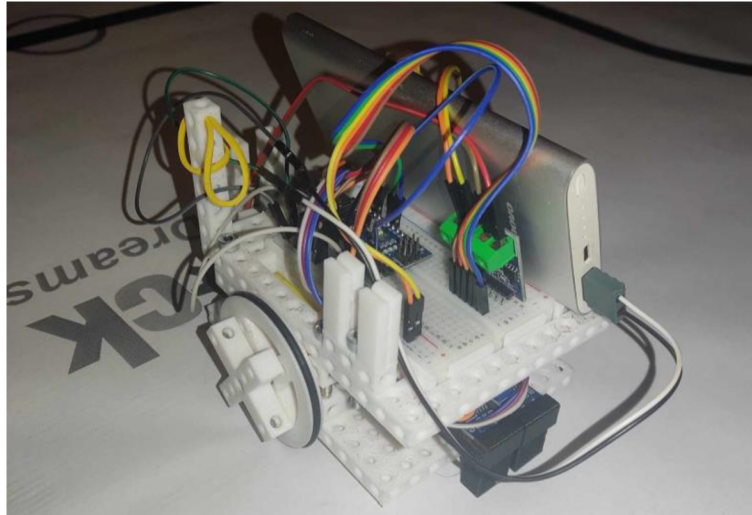
    while (analogRead(senzor_pravy) < H_cidla); // Čekání, dokud pravý senzor nedetekuje
čáru
} else {
    // Oba senzory mimo čáru - otočení vlevo
    turnLeft();

    while (analogRead(senzor_pravy) < H_cidla && analogRead(senzor_levy) < H_cidla);
}
}

void turnRight() {
    analogWrite(motor_L_dopredu, 150);
    analogWrite(motor_L_dozadu, 0);
    analogWrite(motor_R_dopredu, 0);
    analogWrite(motor_R_dozadu, 0);
}

void turnLeft() {
    analogWrite(motor_L_dopredu, 0);
    analogWrite(motor_L_dozadu, 0);
    analogWrite(motor_R_dopredu, 150);
    analogWrite(motor_R_dozadu, 0);
}

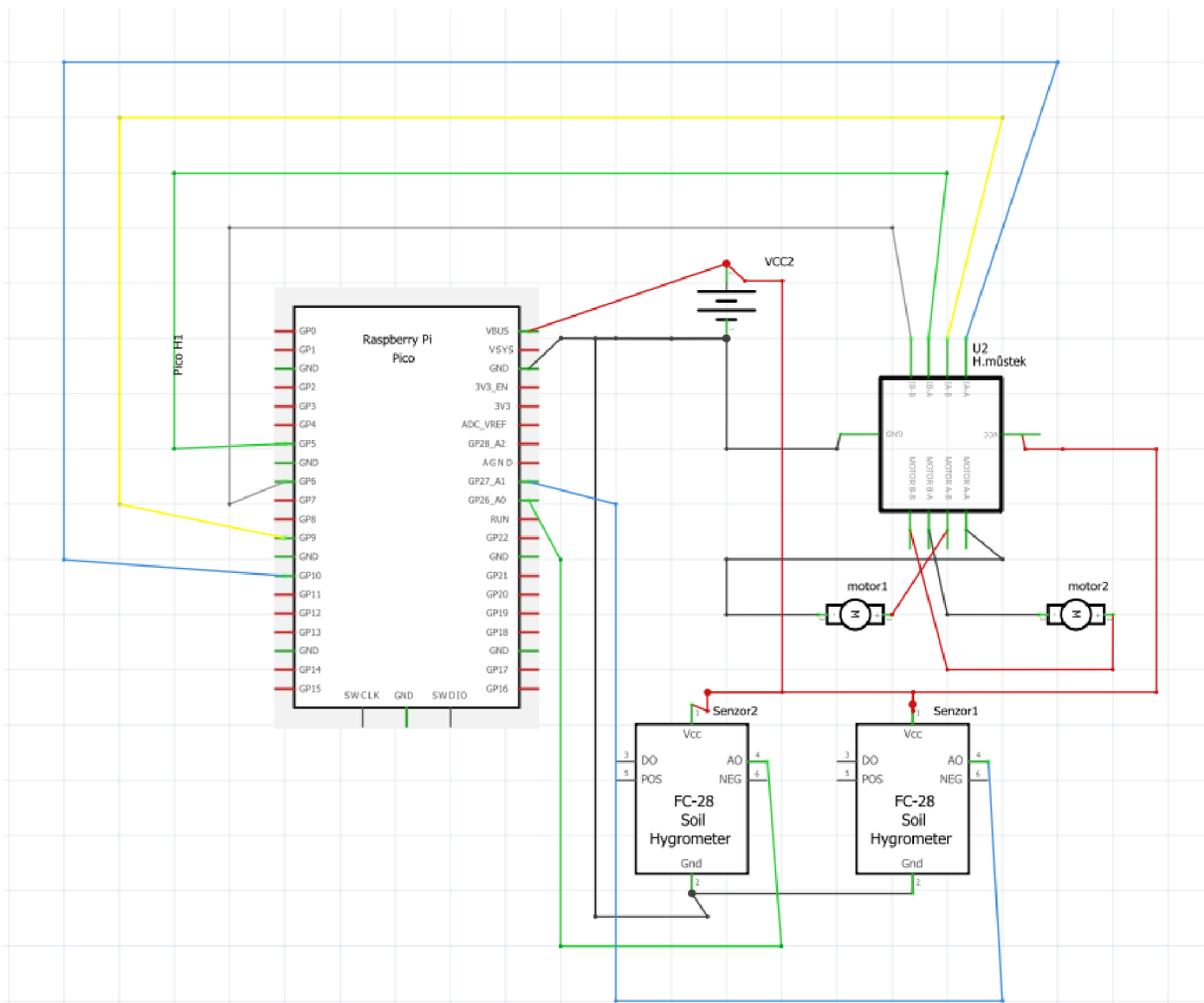
```



**Obr. 10.2 Robot s Arduino Nano**

(Zdroj: Vlastní zdroj)

### Schéma zapojení a program pro Raspberry Pi Pico



**Obr. 10.3 Zapojení Robot Pico**

(Zdroj: Vlastní zdroj)

```

// Definice pinů
int senzor_levy = 26;
int senzor_pravy = 27;
int motor_L_dopredu = 5;
int motor_L_dozadu = 6;
int motor_R_dopredu = 9;
int motor_R_dozadu = 10;

// Definice rychlosti (0-255)
int rychlost_motoru = 150; // Nastavení rychlosti jízdy

void setup() {
  Serial.begin(9600);
  // Inicializace pinů jako vstupy pro senzory
  pinMode(senzor_levy, INPUT);
  pinMode(senzor_pravy, INPUT);
  // Inicializace pinů jako výstupy pro motory
  pinMode(motor_L_dopredu, OUTPUT);
  pinMode(motor_L_dozadu, OUTPUT);
  pinMode(motor_R_dopredu, OUTPUT);
  pinMode(motor_R_dozadu, OUTPUT);
}

void loop() {
  // Čtení hodnot z čidel
  int LC_hodnota = analogRead(senzor_levy);
  int RC_hodnota = analogRead(senzor_pravy);
  // Nastavení prahových hodnot pro čidla
  int H_cidla = 350;

```

```

// Regulace pohybu na základě hodnot z čidel
if (LC_hodnota < H_cidla && RC_hodnota > H_cidla) {
    // Levý senzor mimo čáru, pravý na čáře - otočení doprava
    turnRight();

    while (analogRead(senzor_levy) < H_cidla); // Čekání, dokud levý senzor nedetekuje čáru
} else if (LC_hodnota > H_cidla && RC_hodnota < H_cidla) {
    // Levý senzor na čáře, pravý mimo čáru - otočení doleva
    turnLeft();

    while (analogRead(senzor_pravy) < H_cidla); // Čekání, dokud pravý senzor nedetekuje
čáru
} else {
    // Oba senzory mimo čáru - otočení vlevo
    turnLeft();

    while (analogRead(senzor_pravy) < H_cidla && analogRead(senzor_levy) < H_cidla);
}
}

void turnRight() {
    analogWrite(motor_L_dopredu, rychlost_motoru);
    analogWrite(motor_L_dozadu, 0);
    analogWrite(motor_R_dopredu, 0);
    analogWrite(motor_R_dozadu, 0);
}

void turnLeft() {
    analogWrite(motor_L_dopredu, 0);
    analogWrite(motor_L_dozadu, 0);
    analogWrite(motor_R_dopredu, rychlost_motoru);
    analogWrite(motor_R_dozadu, 0);
}

```

## 11.1 Srovnání práce s Arduino Nano a Raspberry Pi Pico

V této kapitole bych se rád vyjádřil, jak se mi pracovalo s Arduinem i Raspberry Pi Pico. Vzhledem k tomu, že jsem měl více zkušeností s Arduinem, práce s ním byla pro mě samozřejmě jednodušší. Už jsem věděl, co a jak zapojit, jak nahrát program a co očekávat. Na rozdíl od toho jsem s Raspberry Pi Pico neměl žádné předchozí zkušenosti, což vedlo k určitým potížím, ale nakonec se vše podařilo vyřešit.

Překvapilo mě, že po připojení napájení není zřejmé, zda Raspberry Pi Pico běží, zatímco Arduino začne hned po připojení blikat, což dává jasnou indikaci, že vše funguje správně. Zmátlo mě také číslování pinů na Raspberry Pi Pico, kde některé piny mají čísla, která neodpovídají daným GP pinům, což je trochu matoucí. Naproti tomu Arduino Nano má perfektně popsané jednotlivé piny, což umožňuje snadné zapojení.

Celkově se mi lépe pracovalo s Arduinem, což může být způsobeno zvykem nebo tím, že je pro projekty uživatelsky přívětivější.

Co se týče fungování robotů, na první pohled se mi zdá, že Raspberry Pi Pico funguje na dráze o něco lépe než Arduino Nano. Jak je vidět i na krátkém videu v příloze, které jsem zpracoval, Raspberry Pi Pico projíždí dráhu rychleji než Arduino Nano. Lze uvažovat malé odlišnosti v programech, které však podle mého názoru nemají vliv na výsledky.

## Závěr

Porovnání Arduino Nano a Raspberry Pi Pico ukázalo, že obě zařízení mají své jedinečné výhody a jsou vhodné pro různé typy projektů. Arduino Nano vyniká svou jednoduchostí, velkou komunitou a širokou podporou knihoven, což ho činí ideálním pro začátečníky a jednodušší projekty. Na druhé straně, Raspberry Pi Pico nabízí vyšší výpočetní výkon, flexibilitu v programování a možnost využití pokročilých funkcí díky dvoujádrovému procesoru a bohaté nabídce periférií.

Praktické testy, včetně srovnání robotů sledujících čáru, ukázaly, že Raspberry Pi Pico může být přesnější a stabilnější díky svým technickým parametrům, i když Arduino Nano stále drží krok ve většině běžných aplikací. Závěrem lze říci, že volba mezi těmito dvěma mikropočítači závisí na konkrétních požadavcích projektu, technických schopnostech uživatele a dostupných zdrojích.

Tato práce naplnila stanovené cíle a poskytla ucelený přehled a může sloužit jako základ pro rozhodování při výběru vhodného mikropočítače pro různé aplikace.



## Seznam použitých zdrojů

*Autodesk Inventor.* Online. Adeon. Dostupné z: <https://www.adeon.cz/katalog/autodesk/inventor/>. [cit. 2024-04-03].

*ARDUINO ČILI MIKROKONTROLÉRY PRO KAŽDÉHO.* Online. 2021 DPS elektronika od A do Z. Dostupné z: <https://www.dps-az.cz/clanky/id:78455/arduino-cili-mikrokontrolery-pro-kazdeho>. [cit. 2024-04-03].

*Arduino dokumentace.* Online. Vývojové prostředí arduina. Dostupné z: <https://docs.uart.cz/docs/arduino-ide/>. [cit. 2024-04-03].

*Arduino Duemilanove.* Online. Arduino. 2024. Dostupné z: <https://docs.arduino.cc/retired/boards/arduino-duemilanove/>. [cit. 2024-04-03].

*Arduino ide.* Online. Bastlírna kitchen. Dostupné z: <https://bastlirna.hwkitchen.cz/arduino-ide/>. [cit. 2024-04-03].

*Arduino memory guide.* Online. Arduino. 2023. Dostupné z: <https://docs.arduino.cc/learn/programming/memory-guide/>. [cit. 2024-04-03].

*Arduino nano.* Online. Makerguides. 2020. Dostupné z: <https://www.makerguides.com/arduino-nano/>. [cit. 2024-04-03].

*Arduino Nano Tutorial – Pinout & Schematics.* Online. Circuits today. 2020. Dostupné z: <https://www.circuitstoday.com/arduino-nano-tutorial-pinout-schematics>. [cit. 2024-04-03].

*ATMega328P Microcontroller.* Online. 2018. Dostupné z: <https://components101.com/microcontrollers/atmega328p-pinout-features-datasheet>. [cit. 2024-04-03].

*CAD, CAE a CAM.* Online. Cad club. 2021. Dostupné z: <https://www.cadclub.cz/5776/cad-cae-a-cam-jake-jsou-mezi-nimi-rozdily>. [cit. 2024-04-03].

*Co je 3d tisk.* Online. Josef Průša. Dostupné z: <https://josefprusa.cz/o-3d-tisku/>. [cit. 2024-04-03].

*Co je to UART.* Online. Uart. 2011. Dostupné z: <https://uart.cz/139/arduino-a-seriova-komunikace/>. [cit. 2024-04-03].

*Drag-n-drop coding for Raspberry Pi Pico.* Online. Raspberrypi. 2021. Dostupné z: <https://www.raspberrypi.com/news/drag-n-drop-coding-for-raspberry-pi-pico/>. [cit. 2024-04-03].

*Elektrické napětí.* Online. Eluc. Dostupné z: <https://eluc.ikap.cz/verejne/lekce/418>. [cit. 2024-04-03].

*Elektrický proud.* Online. Eluc. Dostupné z: <https://eluc.ikap.cz/verejne/lekce/420>. [cit. 2024-04-03].

*Elektromotor, elektronický motor.* Online. One industry. Dostupné z: <https://www.oneindustry.cz/lexikon/elektromotor-elektricky-motor/>. [cit. 2024-04-03].

*Elektronika.* Online. Sse na jizdarně. 2006. Dostupné z: [https://www.sse-najizdarne.cz/dokumenty/studijni\\_materialy/elektronika.pdf](https://www.sse-najizdarne.cz/dokumenty/studijni_materialy/elektronika.pdf). [cit. 2024-04-03].

*Elektrotechnický základ.* Online. SOŠE, COP Hluboká nad Vltavou. 2011. Dostupné z: [https://www.sosehl.cz/uploads/2020/11/Elektrotechnicky\\_zaklad\\_ucebni\\_text.pdf](https://www.sosehl.cz/uploads/2020/11/Elektrotechnicky_zaklad_ucebni_text.pdf). [cit. 2024-04-03].

*Funkce.* Online. Git Hub. Dostupné z: <https://mrlvsb.github.io/upr-skripta/c/funkce/funkce.html>. [cit. 2024-04-03].

*FRIK's WEB.* Online. 2013. Dostupné z: <http://www.frik.cz/>. [cit. 2024-04-03].

*Historie mikrokontrolerů.* Online. Circuitstoday. 2014. Dostupné z: <https://www.circuitstoday.com/story-and-history-of-development-of-arduino>. [cit. 2024-04-03].

*How to connect Raspberry Pi Pico W to ThingsBoard?* Online. Thingsboard Open-source IoT Platform. Dostupné z: <https://thingsboard.io/docs/devices-library/raspberry-pi-pico-w/>. [cit. 2024-04-03].

*H-můstek a otáčení motůrkem.* Online. Itnetwork. Dostupné z: <https://www.itnetwork.cz/hardware-pc/raspberry-pi/raspberry-pi-h-mustek-a-otaceni-moturkem>. [cit. 2024-04-03].

*Informatické myšlení*, 2018. Online. Dostupné z: <https://www.imysleni.cz/>. [cit. 2024-06-14].  
*INFORMATIKA*. Online. Edu.cz. 2021. Dostupné z: <https://revize.edu.cz/files/informatika-2021.pdf>. [cit. 2024-04-03].

*IR Line Tracking Module*. Online. Cytron. Dostupné z: <https://my.cytron.io/p-ir-line-tracking-module>. [cit. 2024-04-03].

MAŇÁK, Petr. *Řízení krokového motoru pomocí platformy Arduino*. Fakulta elektrotechnická Katedra elektrických pohonů a trakce, 2014. Bakalářská práce. České vysoké učení technické v Praze. Vedoucí práce: Ing. Hlinovský Vít CSc.

MicroPython. *Raspberry pi*. Online. Dostupné z: <https://www.raspberrypi.com/documentation/microcontrollers/micropython.html>. [cit. 2024-01-12].

*MOTOR S PŘEVODOVKOU PRO ARDUINO ROBOTA*. Online. Kw kitchen. Dostupné z: <https://www.hwkitchen.cz/motor-s-prevodovkou-pro-arduino-robota/>. [cit. 2024-04-03].

*MŠMT spustilo web na podporu moderní výuky informatiky*. Online. 2022 Edu.cz. Dostupné z: <https://www.edu.cz/msmt-spustilo-web-na-podporu-moderni-vyuky-informatiky/>. [cit. 2024-04-03].

*MicroPython ako vývojová platforma pre IoT*. Online. Root.cz. 2020. Dostupné z: <https://www.root.cz/clanky/micropython-ako-vyvojova-platforma-pre-iot/>. [cit. 2024-04-03].

*Mikropočítače.* Online. Wiki knihovna. Dostupné z: <https://wiki.knihovna.cz/index.php?title=Mikropo%C4%8D%C3%ADta%C4%8De>. [cit. 2024-04-03].

*Nepájivé pole.* Online. It net work. Dostupné z: <https://www.itnetwork.cz/hardware-pc/arduino/hardware/nepajive-pole-a-desky-plosnych-spoju-pro-arduino>. [cit. 2024-04-03].

*Obvody rozhraní SPI.* Online. Eluc. 2015. Dostupné z: <https://eluc.ikap.cz/verejne/lekce/873>. [cit. 2024-04-03].

*Pasivní elektronické součástky.* Online. It net work. Dostupné z: <https://www.itnetwork.cz/hardware-pc/arduino/hardware/elektronicke-soucastky-pro-arduino-rezistor-kondenzator>. [cit. 2024-04-03].

*Podmínky větvení.* Online. IT net work. Dostupné z: <https://www.itnetwork.cz/csharp/zaklady/c-sharp-tutorial-podminky-vetveni-if-switch>. [cit. 2024-04-03].

*Popis Fusion 360.* Online. 1 C pro. Dostupné z: <https://www.1cpro.cz/fusion.php>. [cit. 2024-04-03].

*Programy pro 3D modelování.* Online. MaterialPRO3D. 2023. Dostupné z: <https://www.materialpro3d.cz/blog/programy-pro-3d-modelovani/>. [cit. 2024-04-03].

*Raspberry Pi Models and Specs: A Comprehensive Guide.* Online. HowChoo. 2023. Dostupné z: <https://howchoo.com/pi/raspberry-pi-models/>. [cit. 2024-04-03].

*Raspberry Pi Pico and Pico W. Raspberry Pi.* Online. Dostupné z: <https://www.raspberrypi.com/documentation/microcontrollers/raspberry-pi-pico.html>. [cit. 2023-11-07].

*Raspberry pi pico datasheet.* Online. Raspberry pi. 2024. Dostupné z: <https://datasheets.raspberrypi.com/pico/pico-datasheet.pdf>. [cit. 2024-04-03].

*Raspberry Pi Pico vs Arduino – Which Board to Choose?*. Online. Electrocredible. 2023. Dostupné z: <https://electrocredible.com/raspberry-pi-pico-vs-arduino-which-board-to-choose/>. [cit. 2024-04-03].

*Raspberry Pi 5 je tu.* Online. Computerworld. Dostupné z: <https://www.computerworld.cz/clanky/raspberry-pi-5-je-tu-nabizi-vyrazne-vyssi-vykon-nej-ctyri-roky-stary-predchudce/>. [cit. 2024-04-03].

*Servo Motor Basics with Arduino.* Online. Arduino cc. 2024. Dostupné z: <https://docs.arduino.cc/learn/electronics/servo-motors/>. [cit. 2024-04-03].

*Slicer: Příprava modelu pro tisk.* Online. Abc. 2022. Dostupné z: <https://www.abicko.cz/clanek/3d-tisk/29019/slicer-priprava-modelu-pro-tisk.html>. [cit. 2024-04-03].

*Studentská licence inventor.* Online. Autodesk. Dostupné z: <https://www.autodesk.com/support/account/education/onboarding/students-guide>. [cit. 2024-04-03].

SHIRRIFF, Ken. THE SURPRISING STORY OF THE FIRST MICROPROCESSORS. *IEEE Spectrum*. Online. Dostupné z: <https://spectrum.ieee.org/the-surprising-story-of-the-first-microprocessors>. [cit. 2024-04-14].

*Stručný popis sběrnice I2C.* Online. Vývoj hw cz. Dostupné z: <https://vyvoj.hw.cz/navrh-obvodu/strucny-popis-sbernice-i2c-a-jeji-prakticke-vyuziti-k-pripojeni-externi-EEPROM-24lc256>. [cit. 2024-04-03].

*Tranzistor.* Online. Na jízdně. Dostupné z: [https://www.sse-najizdarne.cz/dokumenty/studijni\\_materialy/ov/uc\\_text\\_polo\\_sou.pdf](https://www.sse-najizdarne.cz/dokumenty/studijni_materialy/ov/uc_text_polo_sou.pdf). [cit. 2024-04-03].

*The Future of Microprocessors.* Online. Communications of the acm. 2011. Dostupné z: <https://cacm.acm.org/research/the-future-of-microprocessors/>. [cit. 2024-04-03].

*Úvod do jazyka Scratch.* Online. Itnetwork. Dostupné z: <https://www.itnetwork.cz/pro-deti/scratch/uvod-do-jazyka-scratch>. [cit. 2024-04-03].

VANĚČEK, Vít. *Logický analyzátor s Raspberry Pi Pico.* Fakulta elektrotechnická Katedra měření, 2022. Diplomová práce. České vysoké učení technické v Praze. Vedoucí práce Doc. Ing. Jan Fischer, CSc.

*What is a Raspberry Pi?* Online. OpenSource. 2021. Dostupné z: <https://opensource.com/resources/raspberry-pi>. [cit. 2024-04-03].

*What is SOLIDWORKS 3D CAD?* Online. Solidworks. 2022. Dostupné z: <https://www.solidworks.com/product/solidworks-3d-cad>. [cit. 2024-04-03].

*What is programming.* Online. Codecademy. 2019. Dostupné z: <https://www.codecademy.com/article/what-is-programming>. [cit. 2024-04-03].

*Základy programování.* Online. Programovací učebnice. 2006. Dostupné z: [https://programovaci-ucebnice.g6.cz/ucebnice/UcebniceJazykaJava/0\\_Predmluva.xhtml](https://programovaci-ucebnice.g6.cz/ucebnice/UcebniceJazykaJava/0_Predmluva.xhtml). [cit. 2024-04-03].

## **Seznam příloh**

Příloha č. 1: Video Arduino Nano vs Raspberry Pi Pico - odkaz:  
[https://youtu.be/ZlAYFzCD2\\_s](https://youtu.be/ZlAYFzCD2_s)