

# PROJEKTY NA PROGRAMOVÁNÍ

---

Na předmětu Programování ve třídě 3.IT jsme se rozhodli, že nabídneme studentům netradiční výuku v podobě dlouhodobého projektu. Studenti Zbyněk Vajchart a Richard Ochotný výzvu přijali. Tento způsob výuky je velmi náročný, protože klade velký důraz na vnitřní motivaci, organizaci práce a vytrvalost. Studenti si postup práce navrhli s učitelem programování Michalem Frčkem a následně už pouze konzultovali postupy při realizaci.

Chtěli bychom také poděkovat kolegovi Vaškovi Zahradníkovi, který nám vyřešil tisk konstrukčních prvků robotického jeřábu.

Chtěli bychom studenty pochválit za výborný přístup ke vzdělání a doufáme, že budou i vzorem pro další studenty.

Vyučovací projekt jsme rozdělili na několik částí.

- stanovení cíle
- studium podkladů a potřebných informací
- rozplánování výuky
- zahájení prací
- ladění projektu
- hodnocení a ověření funkčnosti

## YTdown - Richard Ochotný

### Jaké jsi si vybral téma?

*Jako projekt jsem si zvolil rozšíření do Firefoxu, pomocí kterého si uživatel může stáhnout jakékoli audio z platformy YouTube a přidat k ní následující metadata: název skladby, jméno interpreta, název alba i obrázek alba.*

*Toto rozšíření jsem původně vytvořil s cílem pomoci mému kamarádovi, který písničky z YouTube stahoval na denní bázi, ale způsob který používal, byl velmi zdlouhavý a náročný. Tímto rozšířením jsem chtěl tento proces co nejvíce zjednodušit.*

## Jaký jsi si stanovil cíl?

*Předtím než jsem se rozhodl z tohoto rozšíření udělat svůj školní projekt, rozšíření sice fungovalo, ale nebyl jsem spokojen jak se vzhledem, tak ani s funkčností. Toto byly hlavní dvě věci, na které jsem se chtěl zaměřit a vylepšit je.*

## Jaká byla příprava před začátkem práce?

*Pro vývoj rozšíření do Firefoxu je potřeba znát jazyky HTML, CSS a JavaScript. Zatímco jazyky HTML a CSS jsem se naučil v prvním a druhém ročníku střední školy při hodinách webových technologií, JavaScript jsem se učil ve svém volném čase, protože mě zajímala tvorba interaktivních webových stránek. Postupem času jsem si tento jazyk velmi oblíbil, a proto jsem se jej rozhodl použít i k vývoji serveru, který má na starosti samotné stahování. Dále jsem se rozhodl místo JavaScriptu použít TypeScript a to z důvodu, že umožňuje používat datové typy, což vývoj velmi zjednodušuje, a dá se jednoduše převést na JavaScript, aby mohl být spuštěn v prohlížeči.*

## Jak práce probíhala?

*Jako první jsem se zaměřil na předělání serveru. Chtěl jsem ho přepsat z JavaScriptu do TypeScriptu a začít používat jinou knihovnu na stahování z YouTube, protože ta předchozí poměrně často nefungovala a rychlost stahování také nebyla optimální. Po chvíli hledání jsem se rozhodl pro nástroj s názvem „yt-dlp“, a to právě kvůli její rychlosti stahování. Protože je to ale nástroj konzolový, potřeboval jsem použít také Node knihovnu „yt-dlp-exec“, která umožňuje yt-dlp jednoduše spustit a kontrolovat její výstup. Nejprve jsem písničky stahoval pouze pomocí yt-dlp. Tento proces je ovšem velmi pomalý. Nejprve se totiž písnička stáhne ve formátu webm, poté se překonvertuje na mp3 a po přidání metadat se pošle uživateli celá najednou. Toto je velmi neefektivní a pro moje potřeby příliš pomalé. Přešel jsem tedy později na následující způsob. Písnička se bude stále stahovat přes yt-dlp ale daný webm výstup začnu ihned zpracovávat pomocí knihovny s názvem „fluent-ffmpeg“, ta změní formát zvuku na mp3, a tento výstup budu ihned odesílat uživateli. Metadata se změní až u uživatele, což celý proces zjednoduší a hlavně zrychlí. Výsledná rychlost stahování by mohla být mnohem vyšší, ale protože VPS, na kterém tento server běží je velmi pomalý, zrychlení se pohybuje okolo dvou třetin až jedné poloviny původní rychlosti.*

*Jako další jsem se zaměřil na klienta do prohlížeče. Také zde jsem přešel na jazyk TypeScript a chtěl jsem také vylepšit grafickou stránku. Nejprve jsem se rozhodl zaměřit na funkční stránku aplikace. Po změně jazyka jsem také přidal velmi potřebná chybová okna, která informují uživatele v případě nějaké chyby při stahování.*

*Když jsem nechával aplikaci testovat mému kamarádovi, pro kterého jsem aplikaci původně vyvíjel, stěžoval si na to, že pokud chce stáhnout písničku s názvem obsahující háčky nebo čárky, zobrazuje se název písničky úplně jinou abecedou. Povětšinou šlo o čínské znaky nebo o jiné naprosto náhodné znaky. Dlouho jsem si s tímto problémem nevěděl rady, protože na mém i jeho počítači se název zobrazoval správně. Až po dlouhé době studování a marných pokusů jsem přišel na to, že všechny knihovny, které jsem použil na změnu metadat písničky, používaly ID3 verze 2.3, která text ukládá s kódováním UTF-16, což jsem usoudil, že telefon mého kamaráda není schopen správně zpracovat. Zkoušel jsem tedy vyhledat nějakou jinou knihovnu, která by používala ID3 verze 2.4, která už text umí uložit i s kódováním UTF-8. Toto kódování je z jakéhosi důvodu telefon schopen zpracovat bez problémů, jak jsem se po testování mohl přesvědčit. Bohužel ani po velmi zdlouhavém hledání jsem vhodnou knihovnu nebyl schopen najít, a tak jsem si musel specifikaci ID3 sám nastudovat a pokusit se udělat vlastní editor na metadata. Po několika dnech jsem editor úspěšně dokončil a přidal jsem ho do webového rozšíření.*

*Další menší části vývoje bylo navrhování a vytváření nového vzhledu a zobrazování průběhu stahování a zbývající čas do stažení. Až na návrh vzhledu tyto části příliš času nezabrali a aplikace je nyní mnohem uživatelsky přívětivější.*

## **Jak hodnotíš svůj projekt?**

*Když porovnáme, jak aplikace vypadala před zahájením projektu a jak vypadá teď, musím říci, že jsem velmi spokojen s finální podobou aplikace i serveru. Všechny stanovené cíle jsem splnil, ale rozhodně bych tento projekt chtěl nadále rozvíjet a přidávat nějaké nové funkce. Například bych ještě rád přidal anglický překlad a možnost přidat další jazyky.*

## **Link k projektu YTdown**

Rozšíření GitHub: <https://github.com/Histmy/YTDown>

ZŠ Komenského 17, učitelé x Bakaláři - Úvod x Nová karta x YTDown - rozšíření do Fire... x +


addons.mozilla.org/cs/firefox/addon/ytdown/

Google zskom17 intranet ZŠ K17 OA Domažlice Bakaláři - zskom17 Bakaláři - OA-Dom Seznam.cz YouTube RetroGames.cz ... NCDDB obj ISIC Sigma R19

Přihlásit se

Firefox Browser  
**ADD-ONS** Rozšíření Motivy vzhledu Více... v

Najít doplňky



## YTDown od Histmy

Nejlepší možnost jak stahovat písničky z YouTube.

Pro používání tohoto rozšíření potřebujete Firefox

**Stáhnout Firefox a nainstalovat toto rozšíření**

[Stáhnout soubor](#)

39  
Uživatelů

1  
Recenze

★★★★★  
5 hvězdiček

5	★	1
4	★	0
3	★	0
2	★	0
1	★	0

# Robotický jeřáb: EEZYbotARM MK2- Zbyněk Vajchart

## Co jsi si vybral ta téma?

*Toto téma jsem si vybral na základě návrhu od mého učitele. Projekt mě zaujal, protože jsem si chtěl vyzkoušet programování a implementaci mechanických částí s mikroprocesorem. Také mě zajímalo, jaká bude práce s Arduinem, které je stěžejní částí projektu, a jak obstojím při sestavování elektroniky.*

## Jaký jsi si stanovil cíl?

*Cílem bylo fyzicky sestavit robotickou „ruku“ a především vytvořit program, který umožní její pohyb na základě přednastaveného cyklu, případně aby ji bylo možno živě ovládat.*

*Zpočátku jsem plánoval, že se bude ovládat pomocí dálkového ovladače, nakonec jsem se ale – vzhledem k jeho omezením (například není možné zaznamenat stisknutí i uvolnění tlačítka) – rozhodl pro ovládání přes sériové rozhraní Arduina (dostupné přes USB). Jako součástí projektu jsem tedy vytvořil i ovládací software na počítač.*

*Jaká byla příprava před začátkem práce?*

*Zpočátku bylo klíčové najít na internetu vhodný model ruky pro vytištění na 3D tiskárně. Zvolil jsem EEZYbotARM MK2 od daGHIZmo ([thingiverse.com](http://thingiverse.com)), protože se mi zdál nejsolidnější a byl k němu dostupný seznam dalších potřebných součástí (šroubů, elektrických) a návod na sestavení. Tyto součástky bylo třeba následně objednat.*

*Dále jsem se seznámil s Arduinem. Ve volném čase jsem se již setkal s programováním mikroprocesorů PICAXE a PIC. Tyto projekty byly ale primitivní a nemusel jsem dělat elektroniku. Arduino je snazší, protože jeho součástí je již deska s konektory a převodníkem sériového rozhraní USART na USB.*

Jak práce probíhala?

Mechanika a elektronika

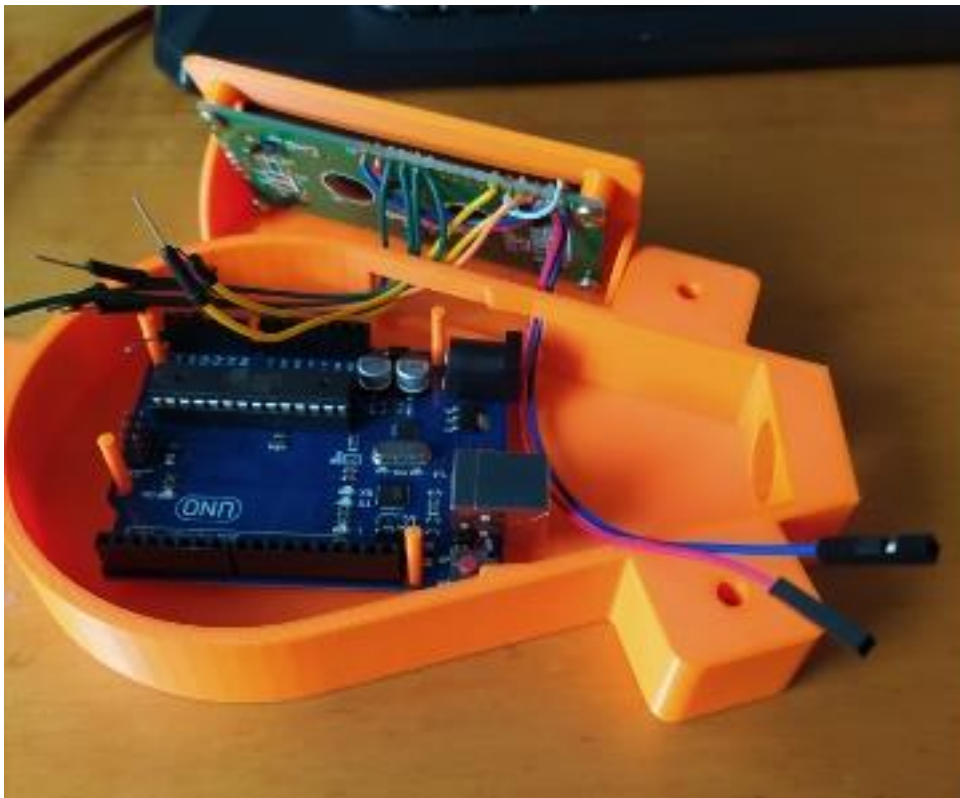


*Začal jsem sestavováním plastových a elektrických částí. Nejprve jsem složil základnu, kde bylo potřeba vložit ložiskové kuličky mezi vytištěné díly. Dále jsem skládal rameno, a nakonec úchyt na jeho konci. Použil jsem nové i použité šrouby a osičky.*



*V tuto chvíli jsem vytvořil jednoduchý program pro Arduino na otestování serv. Zjistil jsem, že USB port počítače nebyl schopen unést zátěž více serv najednou a Arduino se při prudších pohybech resetovalo. Z informací na internetu jsem vypočítal, že přes všechna serva dohromady může procházet mnohonásobně větší proud, než je limit USB portu. Bylo tedy nutné zakoupit externí zdroj.*

*V Blenderu jsem pomocí modelu spodní části ruky vytvořil základnu, do které lze umístit desku Arduina, LCD displej a rozvod serv. Tento model nebyl vyhovující, a proto jsem ho musel předělat a tisknout znovu.*



*Rozvodová deska napěťově spojuje externí zdroj s Arduinem, LCD displejem a servy. Jsou z ní také vyvedeny PWM (řídící) vývody serv do Arduina. S její konstrukcí mi výrazně pomohl jakožto elektrikář můj děda.*

## Programování

### „Firmware“ v Arduinu

*Arduino se programuje v jazyce C++, který je mi velmi blízký. Po seznámení s Arduino IDE (v té době verze 1.x.x) jsem zjistil, že je velmi chudé – sotva zvýrazňuje syntaxi, natož aby nabídlo doplňování kódu a další užitečné nutnosti moderních editorů. Proto jsem pro vývoj*

*použil editor Visual Studio Code s rozšířením pro Arduino, který nabízí IntelliSense a mnohem více. (Toto již není aktuální: nová verze Arduino IDE 2.0, která vyšla během vývoje tohoto projektu, je shodou okolností založená právě na VS Code a sdílí s ním tak i jeho funkce.) Program v Arduinu obsluhuje serva, zpracovává data ze sériového rozhraní a zobrazuje status na LCD displeji.*

### *Ovládací software*

*Ovládací software jsem se rozhodl vytvořit v jazyce C#. Neměl jsem s ním velké zkušenosti, ale vybral jsem ho proto, že je k němu snadné vytvořit uživatelské rozhraní WinForms s WYSIWYG designerem ve Visual Studiu. V .NET Frameworku, ve kterém pak tato aplikace běží, je také překvapivě snadné ovládat sériový port. Při vývoji v C# jsem narážel na jeho přílišnou abstraktnost (vzhledem k tomu, na co jsem ho potřeboval použít) a některé operace, které jsou v C++ samozřejmostí, byly v C# překvapivě složité (například kopírování objektů hodnotou, nikoli referencí), ale možná jsem jenom nenašel správný způsob. Nicméně nakonec program funguje, jak má, a vzhledem k tomu, že je pouze vedlejší částí projektu, jsem s ním dostatečně spokojený.*

*Předmětem tohoto programu je možnost jednak živě ruku ovládat pomocí klávesnice, a dále také nastavené pozice zaznamenat a vytvořit tak cyklus, který je možno přehrát a také uložit jako soubor ve formátu JSON (a pak otevřít). Cyklus se skládá z jednotlivých checkpointů (pozic), mezi kterými je při spuštění lineárně interpolováno během nastaveného času každého checkpointu, a rameno se tak souvisle pohybuje.*

### *Komunikace*

*Pro komunikaci jsem „nadesignoval“ jednoduchý protokol, který spočívá v tom, že každý bajt přijatý Arduinem může být buď instrukce, nebo hodnota. Instrukce pomyslně nastavují adresy paměti, do kterých budou zapsány hodnoty, a hodnoty jsou zapsány do pozic určených předchozími instrukcemi. Bajt může nabývat 256 různých hodnot, nejvyšších 180 program v Arduinu bere jako hodnoty (vyhovující pro nastavení úhlů – rozsah serv je 0° - 180°), zbylé hodnoty jsou interpretovány jako instrukce (od 0 do 75).*

*Používat k ovládní vždy počítač mi přišlo nepraktické, proto jsem algoritmus na interpolaci checkpointů implementoval i do Arduina. Do něj je možno cyklus nahrát z*



*ovládacího programu pomocí výše uvedeného protokolu, tam je uložen do EEPROM paměti, odkud je následně možné ho kdykoli spustit tlačítkem i bez počítače.*

## **Závěr**

Nejvíce zdoluhavou částí bylo modelování základny – musel jsem ji víckrát předělávat. Sestavování mechanických mě zpočátku příliš nebavilo, ale výsledek byl uspokojivý. Také jsem rád, že jsem si to mohl vyzkoušet a vidím to jako největší přínos tohoto projektu. Nejvíce zábavné bylo programování.