



## **Středoškolská technika 2016**

**Setkání a prezentace prací středoškolských studentů na ČVUT**

**„Kiwi“**

**Jakub Kolár, Daniel Veltruský**

SPŠE Ječná  
Ječná 30, Praha 2

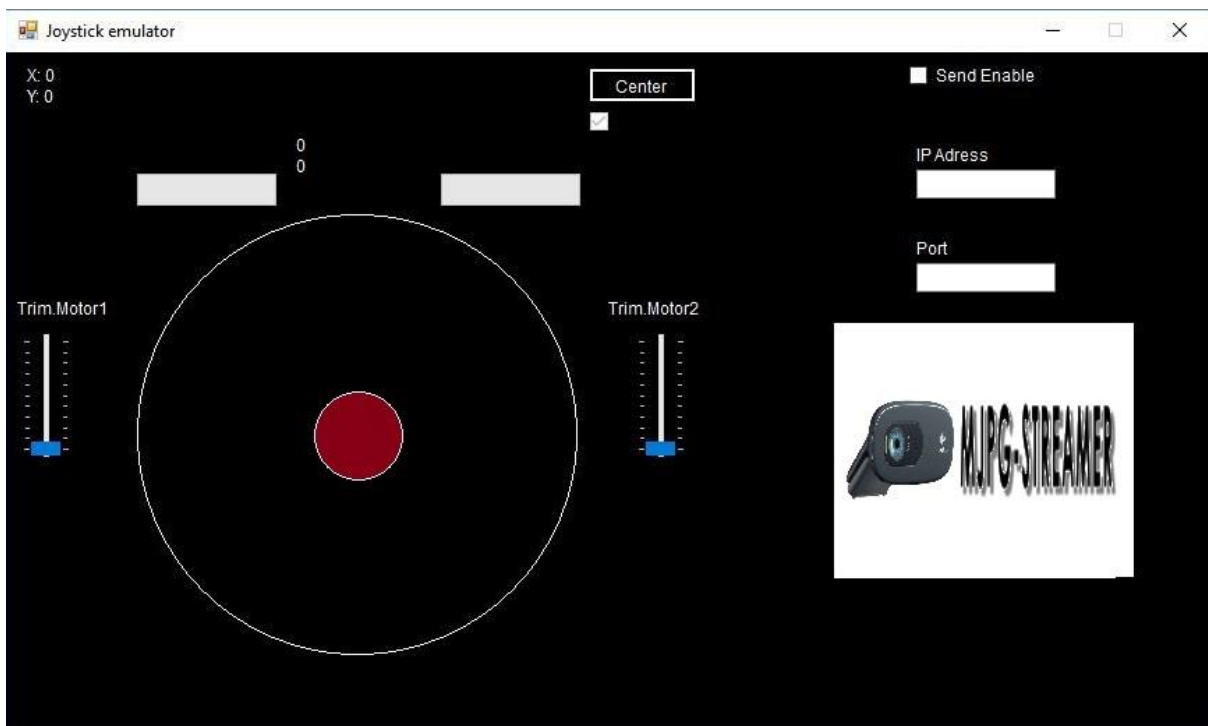
## Úvod

V rámci školního projektu LSD (Linux server development) jsme se rozhodly zkonstruovat robota, se záměrem ho využít jako průvodce po škole ve dnech otevřených dveří. Tento koncept byl poupraven a je možné ho využít i jinými způsoby např. v průmyslu jako skladového robota (který doveze určenou věc). V současné době už v tomto školním programu dále nepůsobíme a na tomto projektu děláme sami v našem volném čase, koncept zůstal vesměs stejný.

## Ovládací rozhraní

Rozhraní jsme po zvážení všech našich možností a znalostí napsali v jazyce **C#**, v rámci Microsoft Visual Studio. Jak vlastně rozhraní funguje a co to vlastně je? Rozhraní je pro nás panel na kterém je umístěn dvou osý joystick – z kterého bereme výchozí hodnoty pro PWM bajty, dále se tam nachází sada TextBoxů, Buttonů a TrackBarů, které udávají od hodnot: natáčení servomotorů (kamerky), rozsvěcení a zhasínání reflektorů, trimování joysticku, až po hodnoty určující připojení na port a ip adresu. Celé rozhraní funguje velmi jednoduše a je navrženo, tak aby s ním mohl pracovat v podstatě každý alespoň trochu počítače znalý uživatel, Pohyby Joystiku – pomocí myši, či spínání různých prvků.

### Náhled rozhraní (z rozpracované verze)



## Komunikace mezi ovládacím rozhraním a Raspberry PI

Při změně hodnoty, či stavu nějakého z prvků rozhraní, odešle program přes UDP spojení do Raspberry PI hodnoty. Raspberry PI čte na předem určeném portu a vlastní IP adrese o čtení a následné funkce se v mikropočítači stará program napsaný v jazyce Python. Tento program nedělá nic jiného než že, jak jsem již zmínil čte na portu..., přebere hodnoty a odešle je po USB sběrnici do Arduina. Jednotlivé pakety (UDP) nejsou zabezpečovány a není nad nimi kontrola, avšak jsou UDP je méně náročné připojení a lze přenášet s větší latencí než u TCP.

## Komunikace mezi Raspberry PI a Arduinem nano

Raspberry přijme pakety a okamžitě je odesílá po USB sběrnici do Arduina, které si je dále přebere.

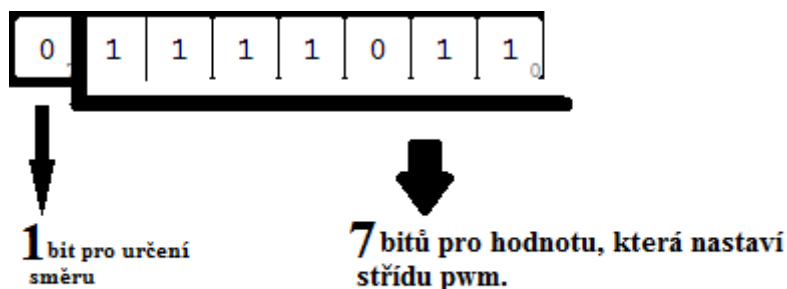
### Arduino

Arduino zastává funkci PWM driveru. Jak to funguje? Jak přijdou data z aplikace do Arduina jsme si už stanovili, Arduino tato data vymaskuje pomocí tzv. Bitových masek a dále si rozdělí jednotlivé bajty (bity) do příslušných metod.

#### Definice komunikačního protokolu mezi RPI a ARD:

- Posíláme **5 bajtů** po USB sběrnici na virtuální COM port, kde **2 bajty** jsou určeny k nastavení rychlosti a směru otáčení motorů, které zajišťují pohon Robota. Další **2 bajty** udávají úhel natočení servomotorů, které pohybují s kamerkou. Zbýlý bajt je prozatím využit na rozsvícení a zhasínání LED reflektorů (využit jeden bit).

#### Zde pro ukázkou rozložení bajtu pro motor :



## Konstrukce a elektronika robota Kiwi :

Robot je zkonstruován převážně z hliníku, díky čemuž má pevnost a relativně malou váhu, tvarově je to nepravidelný šestiúhelník. Jako pohonou jednotku jsme využili motory s převodovkami (z modelu RC tanků), kvůli jejich dostupnosti, klidnému chodu a nízkému napájecímu napětí, které snižuje nároky na baterii (větší kapacita při menší hmotnosti). Motory jsou řízeny H-můstky (L298N). Baterie jsme zvolili Gelovou 6V, 10Ah, která je umístěna v držáku vespod podvozku a napájí kompletně celého robota.

## Tělo robota v grafickém prostředí VariCAD (bez krytu a komponent)

