

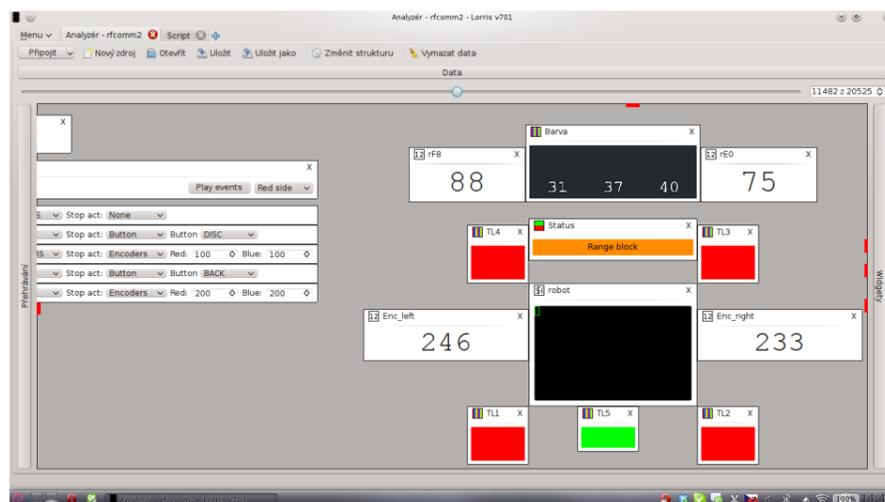
# LORRIS TOOLBOX

## Sada nástrojů pro vývoj a řízení robotů

Lorris Toolbox je sada několika nástrojů, které mají společný cíl – pomáhat při vývoji, ladění a řízení robotů, ale i jiných elektronických zařízení.

### 1. Analyzátor

- Soustřeďuje se na zobrazování dat z robota v grafické podobě.
- Analyzátor pro zobrazování používá tzv. widgety – malá „okna“, která zobrazují určitou část dat.
- Widgety mají individuální nastavení a uživatel si je může umístit na libovolné místo na pracovní ploše.
- Lorris obsahuje několik typů widgetů, například *Číslo*, *Barva*, *Sloupcový bar*, *Kolo* (zobrazení úhlu v kružnici) či *Graf*.
- Pomocí widgetů lze sestavit rozhraní vyhovující prakticky jakémukoliv robotovi.
- Analyzátor je ideální i pro snadné zobrazování dat z prvků, u kterých není vhodné jako výstup použít čísla – například barevný senzor.
- Některé widgety mohou posílat data i směrem do robota. Díky tomu je možné kromě zobrazování dat jejich prostřednictvím robota i ovládat.
- Pozornost si zaslouží widget „script“. Uživatel v něm může napsat vlastní script, který zpracovává příchozí data. Script může využít ostatní widgety a další části Lorris, díky tomu lze zobrazit i jinými způsoby interpretovat takřka jakákoliv data.
- Pomocí scriptu lze upravovat i samotný program Lorris.



*Hlavní okno programu*

### 2. Uživatelské prostředí pro programátor Shupito

- Shupito je programátor mikrokontrolérů. Na jeden konec programátoru se připojí čip, na druhý počítač – bez programátoru nelze do mikrokontrolérů nahrát program.
- Lorris obsahuje uživatelské rozhraní pro ovládání Shupita – zapisování programu, čtení a mazání paměti čipu a programování pojistek.

### 3. Terminál

- Klasický terminál - zobrazuje příchozí data jako text nebo vypisuje byty jako hexadecimální čísla.

### 4. Proxy mezi sériovým portem a TCP socketem

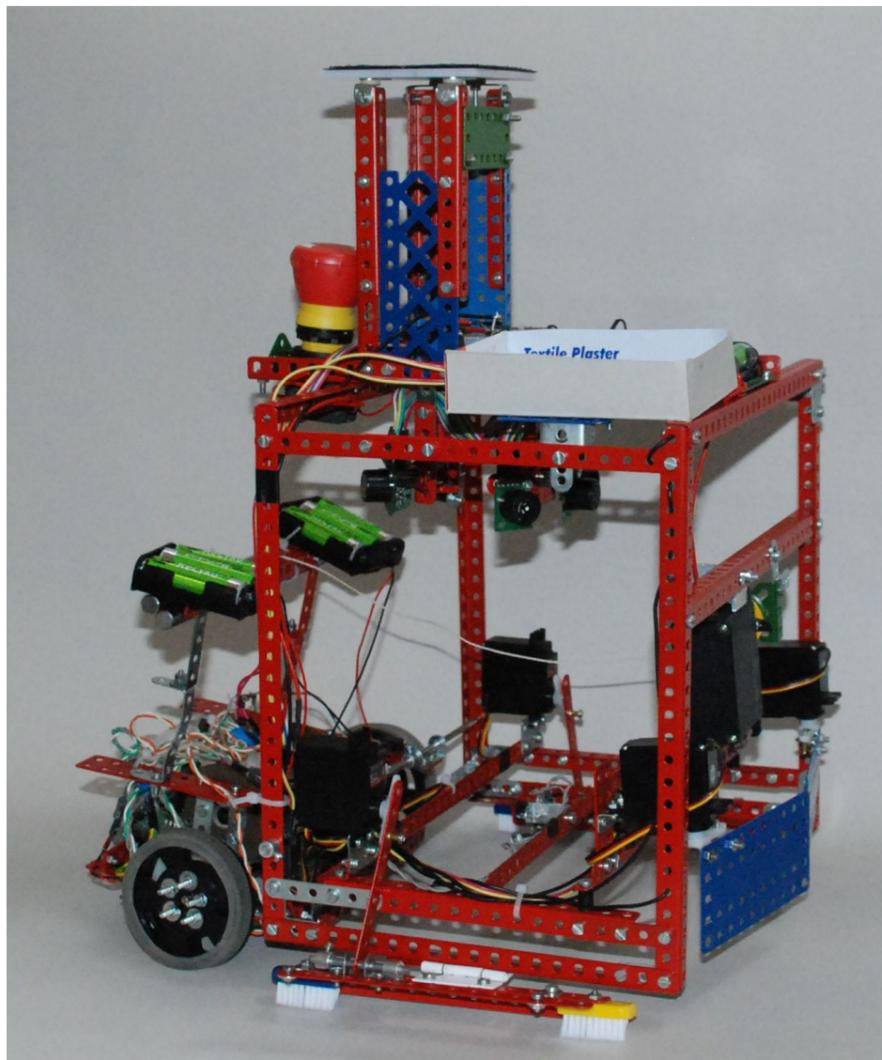
- Vytvoří server připojený na sériový port - k tomuto portu se pak lze připojit odkudkoliv z internetu.

# PŘÍKLAD POUŽITÍ

## Stavba robota pro soutěž Eurobot

Použití mého programu Lorris je zde prezentováno na příkladu stavby robota, který vznikl na naší škole (SPŠ a VOŠ technická, Sokolská 1, Brno) v roce 2011 pro soutěž Eurobot. Právě při vývoji tohoto robota vyvstala palčivá potřeba mít k dispozici nástroj, který by umožňoval ve všech fázích jeho vývoje snadné a rychlé testování a ladění všech funkcí a komponent robota. Vzhledem k tomu, že nejviditelnější částí programu Lorris je nástroj Analyzátor, je v této ukázce prezentováno především jeho použití, ostatní nástroje (Shupito, Terminál) však byly také použity, například při programování mikročipu v robotovi.

V příkladu je vytvořeno jednoduché uživatelské prostředí pro ovládání, testování a programování pro jednoho robota. Toto prostředí však lze znovu použít i pro jiného robota anebo vytvořit nové, pokud je robot příliš atypický a vyžaduje jiný typ ovládání.



### Část 1: mechanická kostra robota

Jako první byla navržena mechanická konstrukce robota. Již v této fázi byla využita moje aplikace Lorris. Pro otestování funkčnosti a chování motorů a servomotorů bylo použito ovládání pomocí joysticku. V Lorris jsem sestavil menší skupinu widgetů: „Script“, který čte data z joysticku, přepočítává je na rychlosti, které je třeba nastavit motorům a odesílá je do robota. Dále widget „Vstup“, ve kterém je nastavení ovládání pomocí joysticku a 2 widgety „Číslo“, které zobrazují aktuální rychlosti motorů.

**Widget Script:**  
čte data z joysticku, vypočítává podle nich rychlosti motorů a odesílá je do robota

```
Script X
0: GreenAsia Inc. USB Joystick
L: 187 R: 187
```

**Widget Vstup:**  
nastavení joysticku  
- výběr joysticku  
- maximální rychlost  
- poloměr zatáčení  
- záměna směru zatáčení

Joystick setting [S]  
0: GreenAsia Inc. v  
Refresh joystick list  
Max speed: 255  
Turn speed: 90  
Invert left/right

Speed left [S]  
187

Speed right [S]  
187

**Widget Číslo:**  
výsledná rychlost

# PŘÍKLAD POUŽITÍ

## Část 2: ladění a nastavení senzorů

Po vyladění mechanické části robota byl osazen senzory. Po jejich umístění jsem v nástroji Analyzér vytvořil rozhraní, které využívá zejména widgetů „Script“, „Číslo“, „Barva“ a „Status“. Každý z těchto widgetů je možné na pracovní ploše Analyzáru přesouvat, zmenšovat nebo zvětšovat, díky čemuž je možné jejich rozmístění tak, aby odpovídalo skutečným pozicím senzorů na robotu. Jako optimální se jeví zobrazení jako při pohledu shora.

**Tento měřič vzdálenosti svítí červeně, protože naměřená hodnota je menší než 35 cm. Robot se zastaví aby nenarazil.**

**Widget Barva:** Data z barevného senzoru. Čísla jsou RGB hodnoty.

Vzdálenost naměřená ultrazvukovým měřičem. Hodnota je v cm.

**Widget status:** Zobrazuje aktuální stav robota, například jestli před ním nestojí nějaká překážka.

**Widget Číslo:** Dva enkodéry měří ujetou vzdálenost. Na tomto snímku robot couval a mírně zatáčel doleva.

Toto tlačítko je uvnitř robota a je stisknuto, pokud v sobě robot má hrací prvek (pěšce).

**Widget Barva:** Na každém rohu robota je tlačítko detekující náraz. Zelená znamená stisknuto.

The interface displays the following data:

- rF8: 30 (Red)
- Barva: 29 34 37
- rE0: 110
- Status: OK (Green)
- TL3: Red
- Enc\_left: -1141
- robot: [Image]
- Enc\_right: -781
- TL1: Green
- TL5: Red
- TL2: Green

# PŘÍKLAD POUŽITÍ

## Část 3: programování reaktivního chování robota

Vrcholem vývoje robota bylo programování jeho chování na herní ploše. Při této příležitosti se v plné míře uplatnil widget „Script“ programu Lorriss. V tomto widgetu bylo vytvořeno scriptovací prostředí, které zapouzdřilo nejtypičtější povelové sady, pomocí kterých lze s výhodou konstruovat složitější vzorce chování robota. Widget „Script“ by umožnil i přímé psaní scriptu pro řízení robota, ale zmíněné prostředí tuto práci výrazně zjednodušilo. Za povšimnutí stojí také to, že zde byl widget script využit nejen pro řízení robota, ale i pro vylepšení fungování samotného nástroje Analyzátor.

The screenshot shows the Lorriss Script widget interface. On the left is a terminal window titled "Program" showing a log of events: "Event done 3", "Encoder event 0 done", "Event done 0", "Encoder event 3 done", "Event done 3", "Encoder event 0 done", "Event done 0", "Encoder event 5 done", "Event done 5", "Encoder event 5 done", and "Event done 5". The main area is titled "ev\_head" and contains a list of three actions. Each action has a checkbox, a "Moveflags" dropdown, a "SpecAct" dropdown, and a "Stop act" dropdown. Action 1: MOVE NONE, OPEN DOORS, None. Action 2: MOVE FORWARD, CLOSE DOORS, Button DISC. Action 3: MOVE NONE, CLIMB, None. Above the list are buttons for "Add back", "Add front", "0", "Insert to pos", "Play events", and a "Red side" dropdown. Callouts point to various features: "Widget Script: Obstarává veškerou funkčnost této části." (points to the terminal), "Přidávání nové akce - dozadu, dopředu a na určitou pozici" (points to the "Add back" and "Add front" buttons), "Tlačítko, které spustí program" (points to the "Play events" button), "Výběr strany hřiště" (points to the "Red side" dropdown), "Směr pohybu robota" (points to the "Moveflags" dropdown), "Akce, kterou robot vykoná poté, co dojde na určené místo" (points to the "SpecAct" dropdown), "Kdy se robot zastaví. Možnosti: - když ujede určitou vzdálenost - když je stisknuto určité tlačítko - když robot dojde na určitou barvu" (points to the "Stop act" dropdown), and "Widgety 'Vstup'" (points to the right side of the interface).

V tomto příkladu používám jednoduché „akce“, které robot postupně provádí. Každá akce má 3 hlavní parametry - směr jízdy, kdy se má robot zastavit a co má vykonat, když se zastaví na cílovém místě. Všechny akce je možné ve scriptovacím prostředí rovnou měnit, bez nutnosti přeprogramovávat robota. Všechny ostatní části prostředí Lorriss stále fungují, i když robot je právě řízen nastaveným scriptem. Díky tomu lze sledovat stav robota i všech jeho senzorů a rychle zjistit zdroj případného neočekávaného chování.