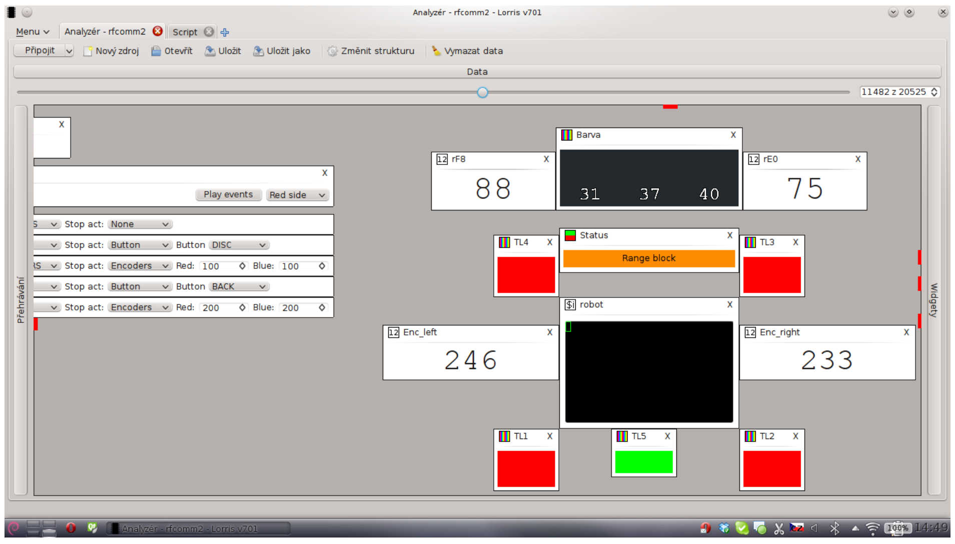
LORRIS TOOLBOX
Set of tools designed for developement and controling of robots

Lorris Toolbox is a suit of multiple tools, which all share the same goal – to help with developement, debugging and controling of not only robots, but also other electronic devices.

1. Analyzer

- Its main purpose is to display graphically the data from robot.
- Analyzer uses widgets to display data – small "windows", each showing certain part of data.
- Each widget has individual settings and the user can place them anywhere on the workspace.
- Using widgets, it is possible to assemble interface suitable for virtually any robot.



Stop act: None

Stop act: Button Button DISC

Stop act: Encoders Red: 100 Blue: 100

Stop act: Button Button BACK

Stop act: Encoders Red: 200 Blue: 200

Main application window

- Several types of widgets are available in Lorris, for example *Number, Color, Column bar, Circle* (displaying angle within circle) or *Graph*.
- Analyzer is also ideal for easy displaying of data from components for which using only numbers is not eligible, e.g. color sensor.
- Some widgets can also send data to the robot. It consequently means that beside displaying data, widgets can also control the robot.
- From all the widget types, „Script“ is the most notable one. The user writes his own script, which processes the incoming data. User's script can use other widgets and other parts of Lorris, which means that it can display or in other ways interpert virtually any data.
- Using script, the user can modify the behavior of Lorris itself.

2. User interface for Shupito programmer

- Shupito is a programmer of microcontrollers. One end of the programmer goes to the computer, the second one to the microcontroller. The programmer is needed to write program into microcontrollers.
- Lorris contains user interface for Shupito – writing program into chip, reading and erasing chip memory and programming of chip's fuses.

3. Terminal

- Regular terminal – displays incoming data either as text or as hexadecimal byte dump.

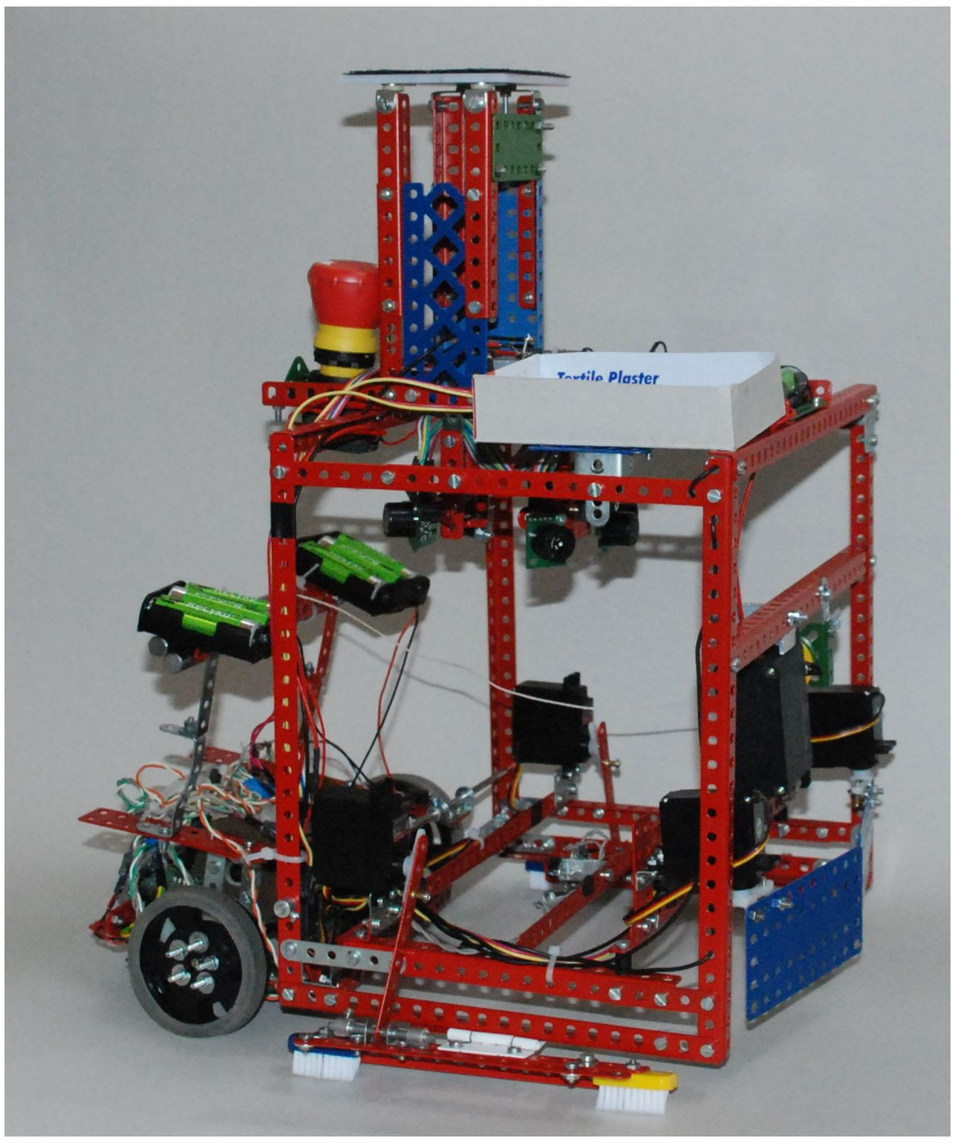
4. Proxy between serial port and TCP socket

- Creates server connected to serial port, which makes connection to that serial port from anywhere on the internet possible.

APPLICATION EXAMPLE
Developement of robot for the Eurobot contest

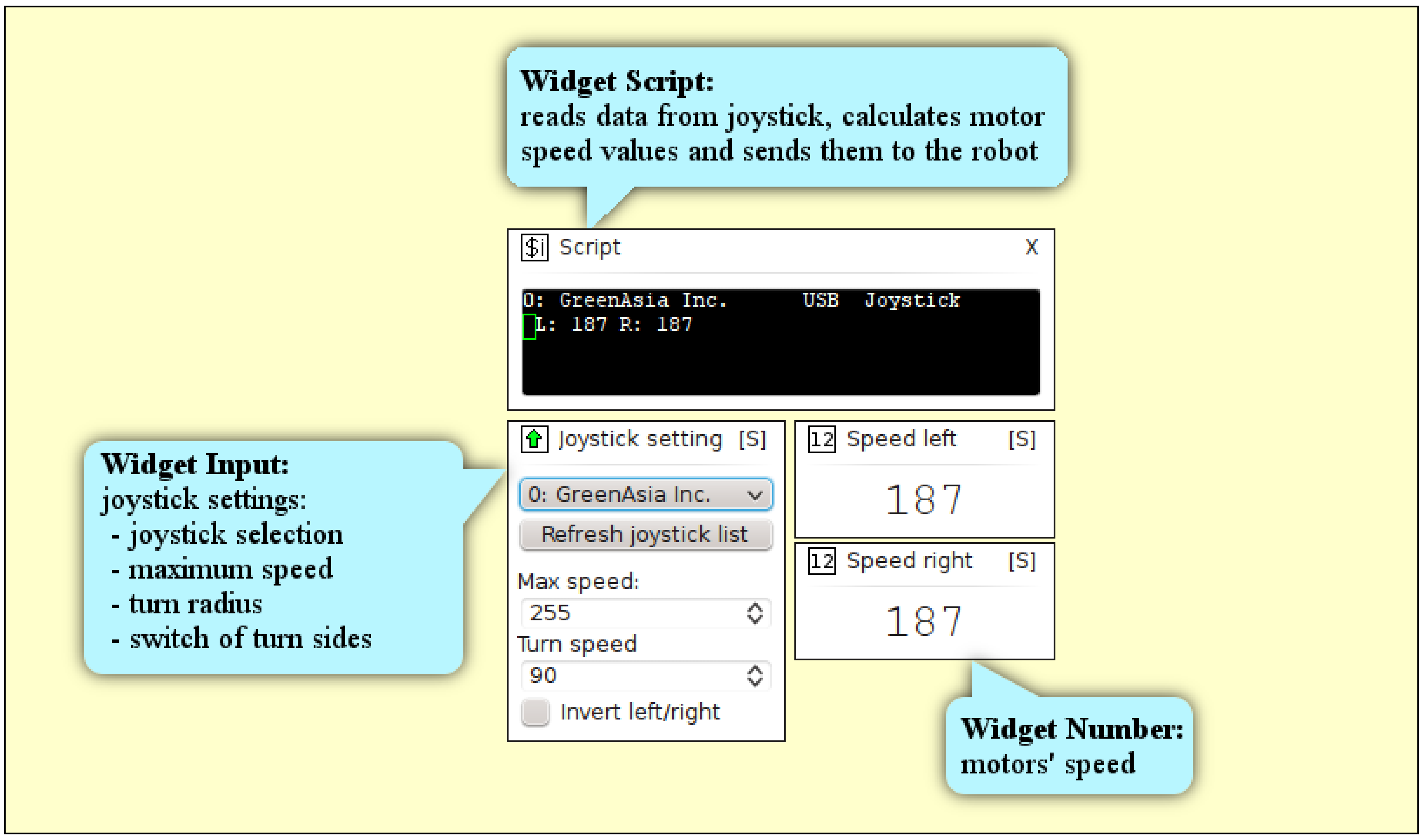
Usage of my Lorris program is exaplained here using example case of robot, which was developed on our school (SPŠ a VOŠ technická, Sokolská 1, Brno) in 2011 to compete in Eurobot contest. Just during the developement, the most pressing need for tool, which would let us test and debug all of the robot's functions and components quickly and easily, arised. Mainly usage of the Analyzer tool is presented here, due to the fact that it is the most visible part of the program. Other tools (Shupito, Terminal) were also used, for example for writing program into the robot's chip.

This example contains simple user interface for controling, testing and debugging of our robot, but this interface can also be used for other robots. You can also create new interface to fulfill your needs, for example when the robot is too atypic and requires different type of controls.



Part 1: mechanical frame of the robot

Body of the robot, alongside with motors and servomotors, was constructed first. Even in this early stage, my Lorris program was already used. We needed to test if all the motors and servos work properly and how exactly they behave, so I assembled a small group of widgets in Lorris, which would allows to control the robot via joystick. Several widgets were used, namely „Script“, which was reading data from joystick, calculating the speed values for motors and sending them to the robot. Next, widget „Input“, which contained settings joystick parameters and lastly, 2 widgets „Number“ with actual motors' speeds.



Widget Script:
reads data from joystick, calculates motor speed values and sends them to the robot

Script

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

Joystick setting [S]

0: GreenAsia Inc.

Refresh joystick list

Max speed: 255

Turn speed 90

□ Invert left/right

Speed left [S]: 187

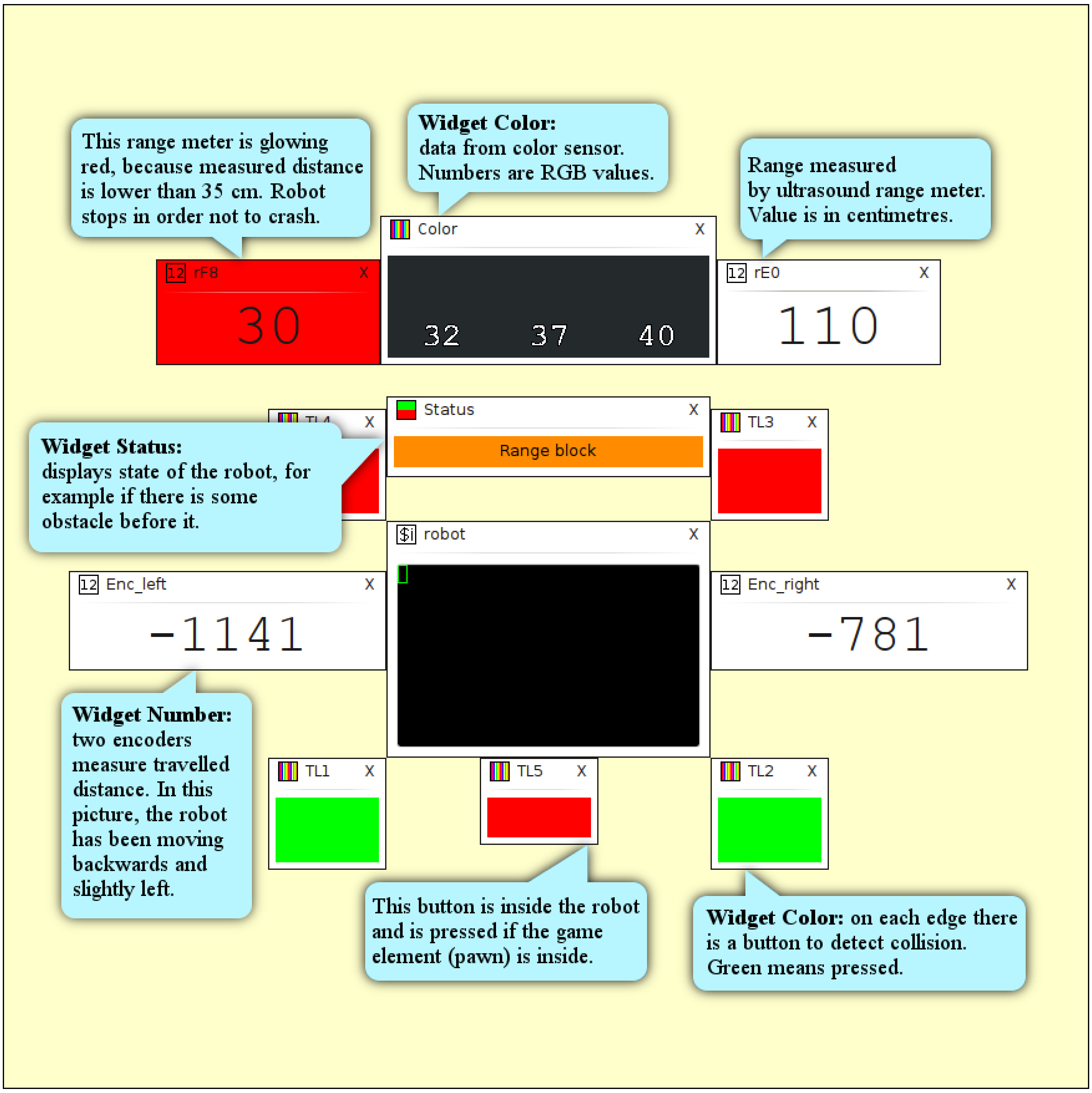
Speed right [S]: 187

Widget Input:
joystick settings:
- joystick selection
- maximum speed
- turn radius
- switch of turn sides

Widget Number:
motors' speed

APPLICATION EXAMPLE
Part 2: debugging and ajdusting of sensors

When the mechanical frame was complete and tested, all sensors were added to the robot. After that, interface to actually see sensor values was needed, so I created interface for this purpose in Analyzer. It uses mainly „Script“, „Number“ „Color“ and „Status“ widgets. Each and every one of these widgets can be moved around Analyzer's worspace and resized. That makes it possible to place widgets so that their positions coresponds with their real positions on the robot. Top view appears to be ideal for this task.



This range meter is glowing red, because measured distance is lower than 35 cm. Robot stops in order not to crash.

Widget Color: data from color sensor. Numbers are RGB values.

Range measured by ultrasound range meter. Value is in centimetres.

Widget Status: displays state of the robot, for example if there is some obstacle before it.

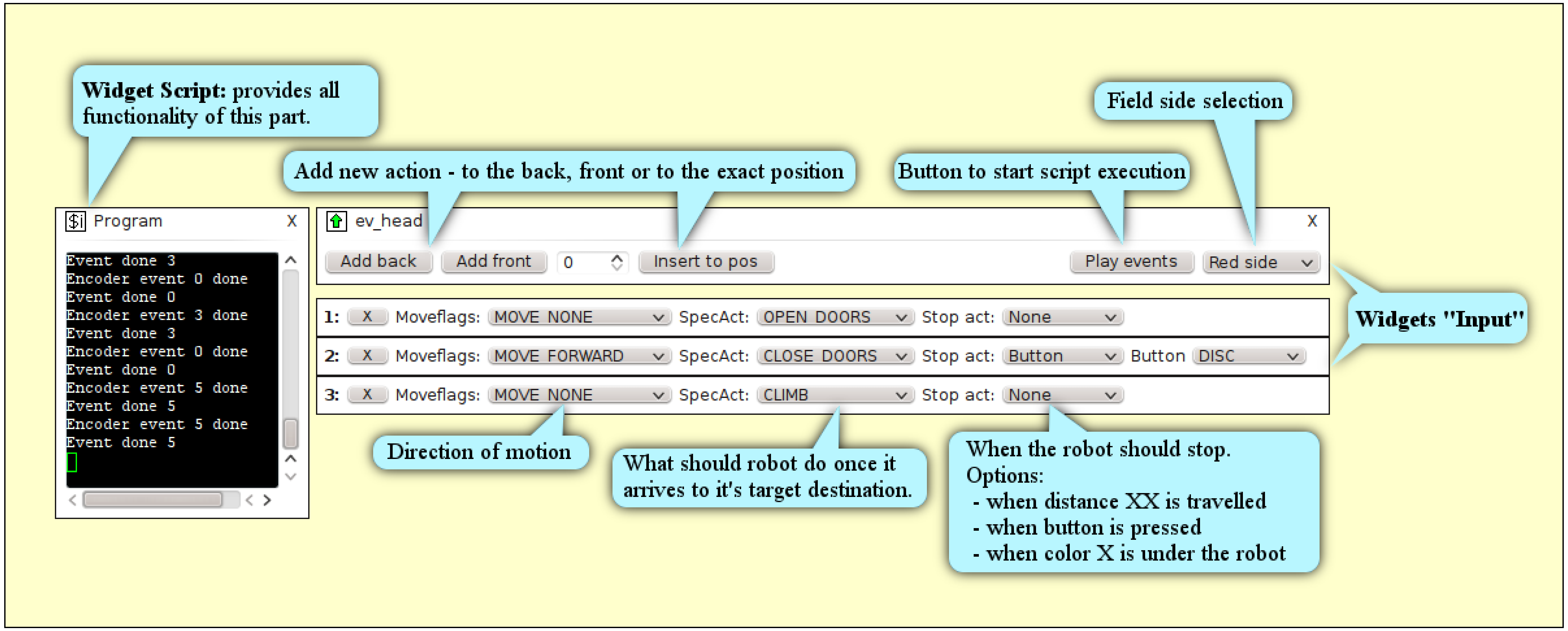
Widget Number: two encoders measure travelled distance. In this picture, the robot has been moving backwards and slightly left.

This button is inside the robot and is pressed if the game element (pawn) is inside.

Widget Color: on each edge there is a button to detect collision. Green means pressed.

APPLICATION EXAMPLE
Part 3: programming of robot’s reactive behavior

Peak of the developement was programming of its behavior on the game field. For this occasion, widget „Script“ in my Lorris program was used to large extent. Scripting enviroment which encapsulated robot’s basic command sets was created in this widget. These command sets allows us to create more complicated behavioral patterns for the robot. It would be possible to write script for the robot directly, but this enviroment considerably simplified and sped up the developement. Another fact is also worth noticing – widget „Script“ was used here not only to control the robot, but also to improve functionality of the Analyzer tool itself.



```
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
Event done 5
```

1: X Moveflags: MOVE NONE SpecAct: OPEN DOORS Stop act: None

2: X Moveflags: MOVE FORWARD SpecAct: CLOSE DOORS Stop act: Button Button DISC

3: X Moveflags: MOVE NONE SpecAct: CLIMB Stop act: None

Widget Script: provides all functionality of this part.

Add new action - to the back, front or to the exact position

Button to start script execution

Field side selection

Widgets "Input"

Direction of motion

What should robot do once it arrives to it's target destination.

When the robot should stop.
Options:
- when distance XX is travelled
- when button is pressed
- when color X is under the robot

In this example, I use simple „actions“, which are executed by the robot step by step. Each action has 3 main parameters – direction of movement, when the robot should stop and what should it do when it arrives at it’s target destination. Each action can be changed directly in the scripting enviroment, bypassing the need to re-program robot after every change. All other parts of Lorris are still working, even when the robot is controlled by the script. That makes it possible to keep track of robot’s state as well as all his sensors and quickly find the source of possible unexpected behavior.