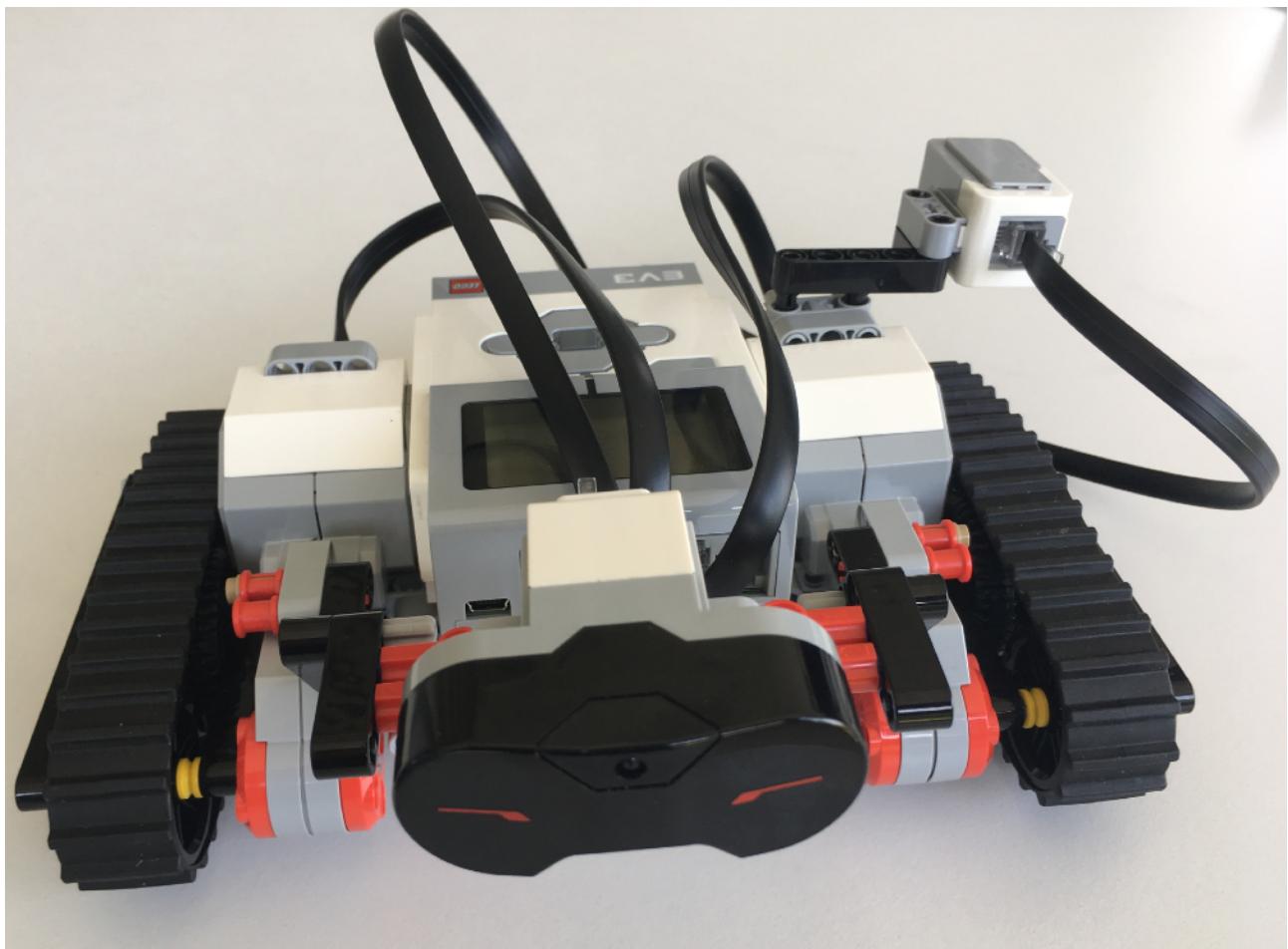


MAN4Kids : ManRobot

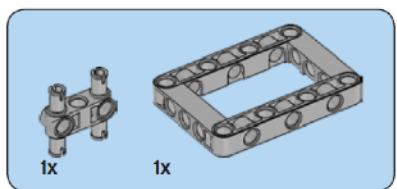
instructions

by
Miro Wengner
Brian Porter
Valeria Kalteis

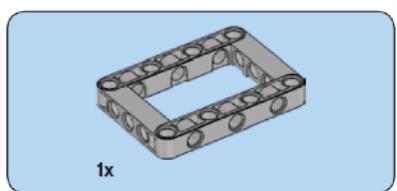
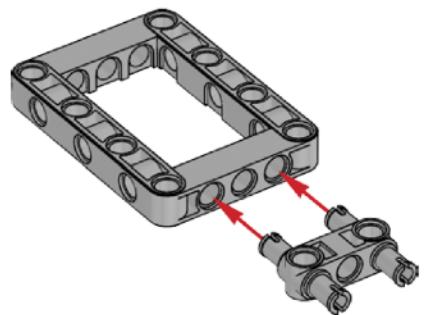


content :

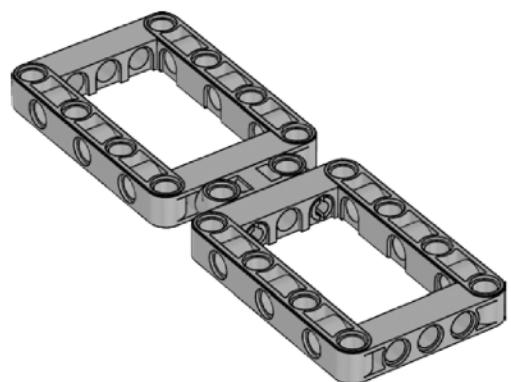
1. Assemble part



1

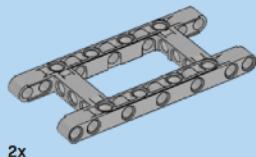
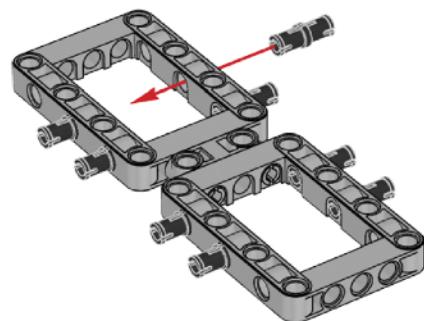


2

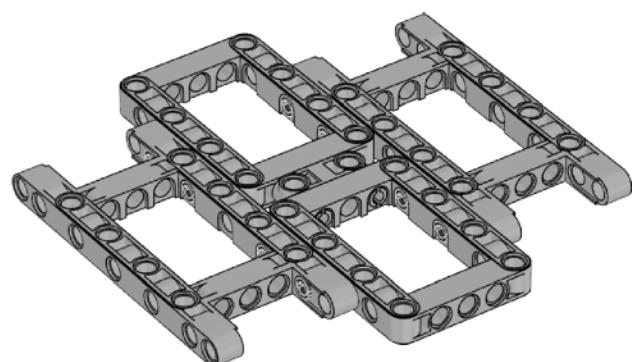


8x

3

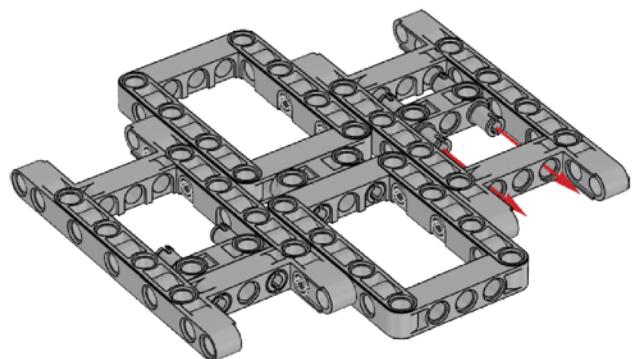


4

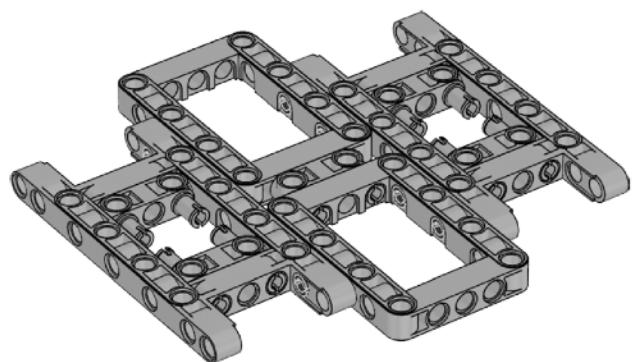


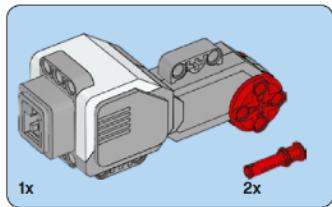


5

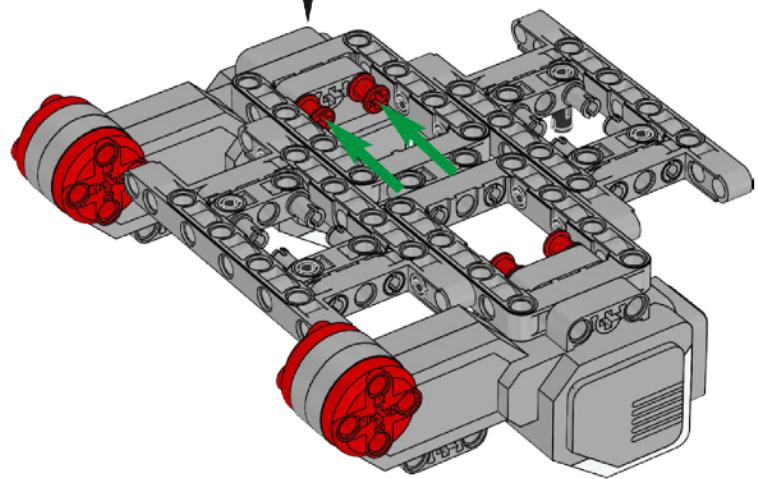
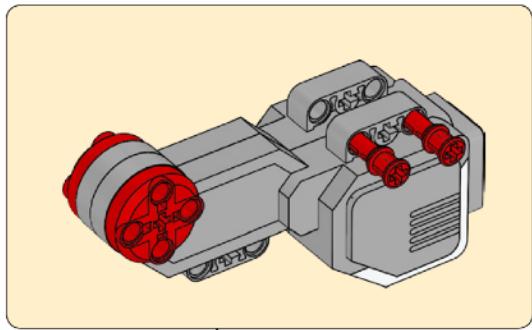


6

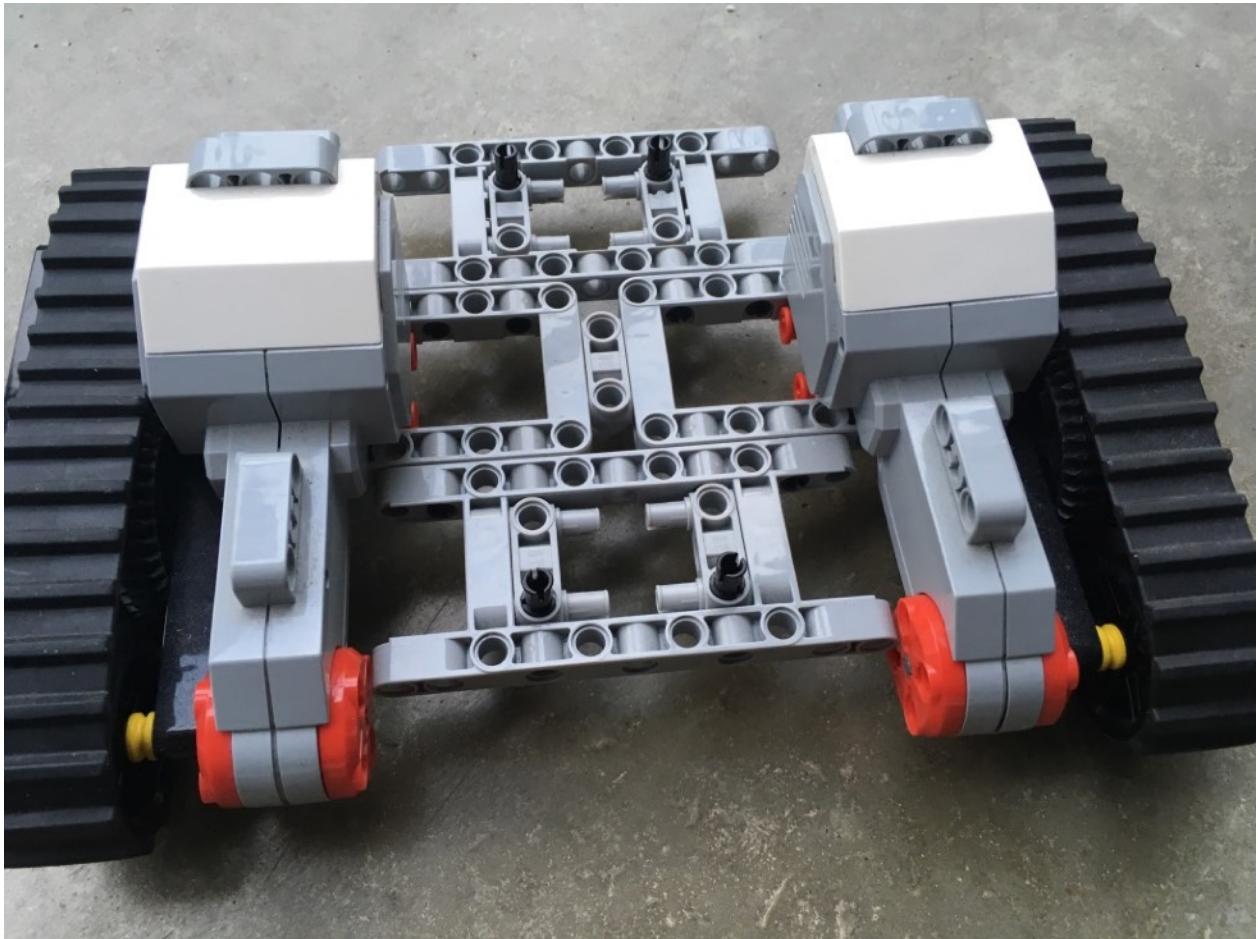


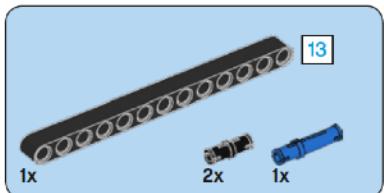
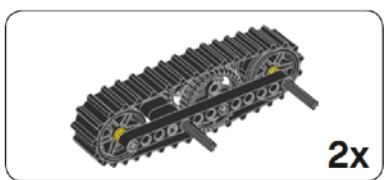


9

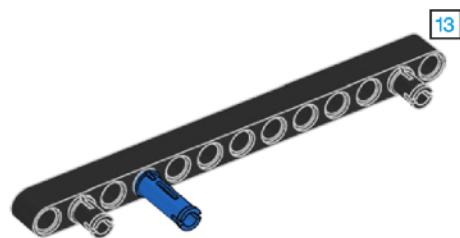


Robot Number42: platform with strips



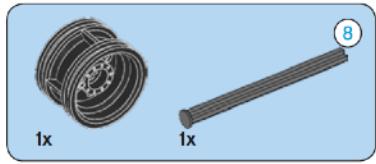


1

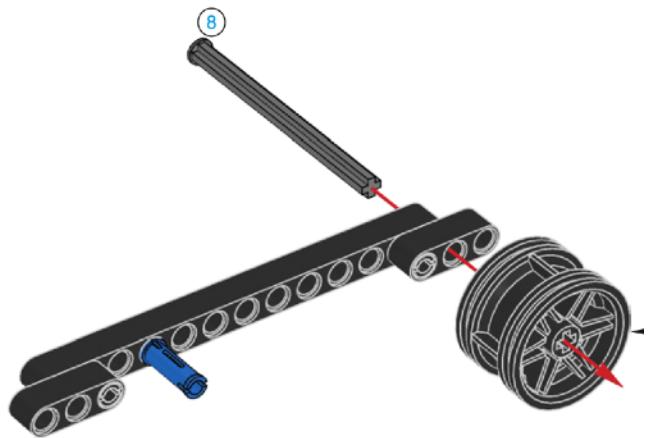


2

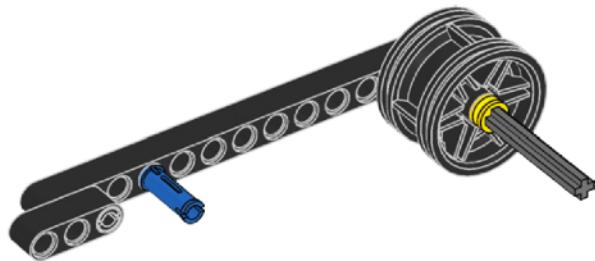


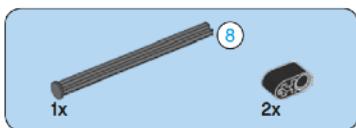


3

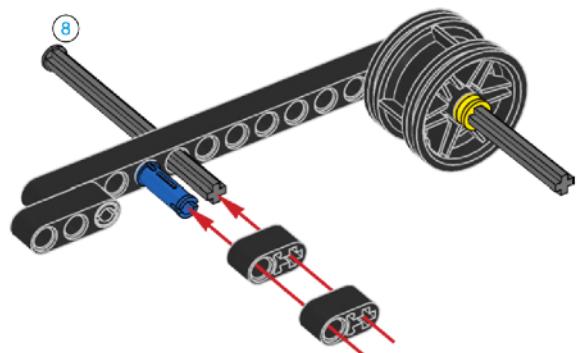


4

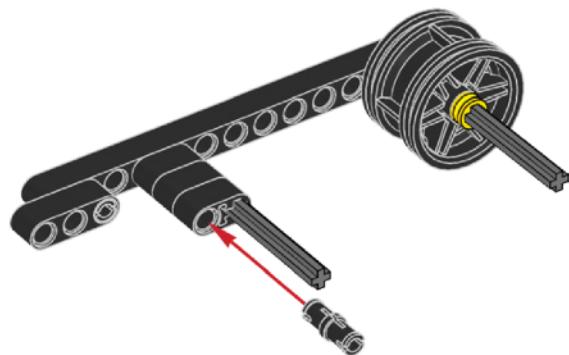


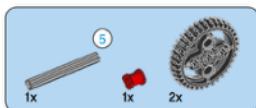


5

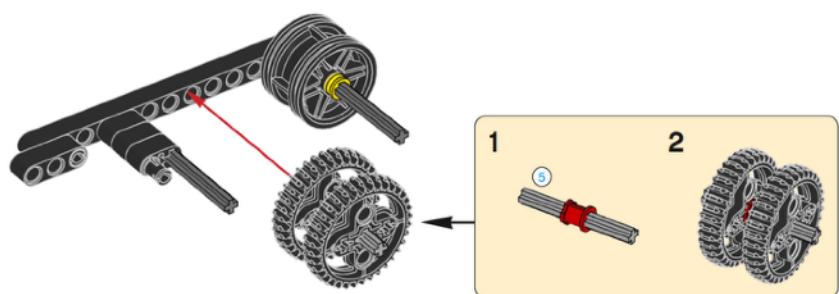


6

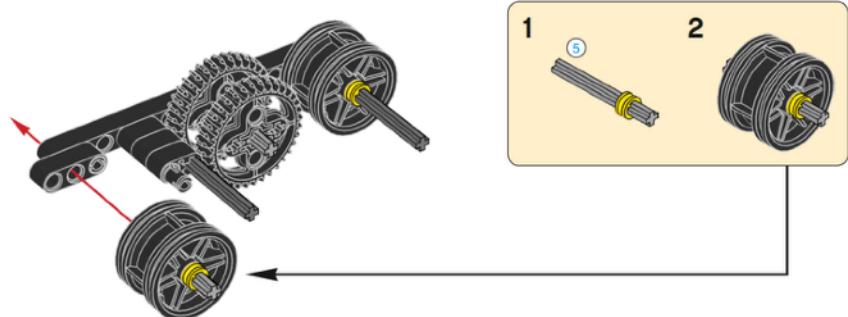


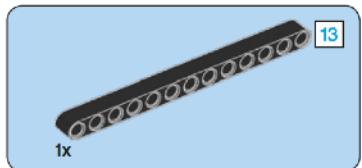


7

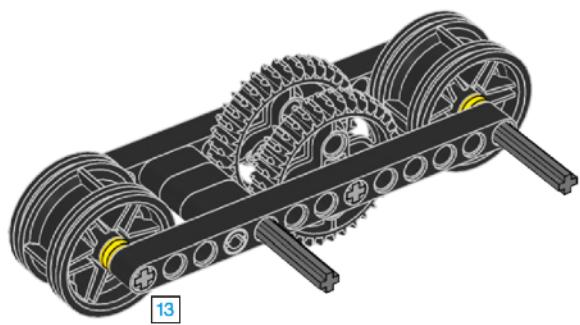


8

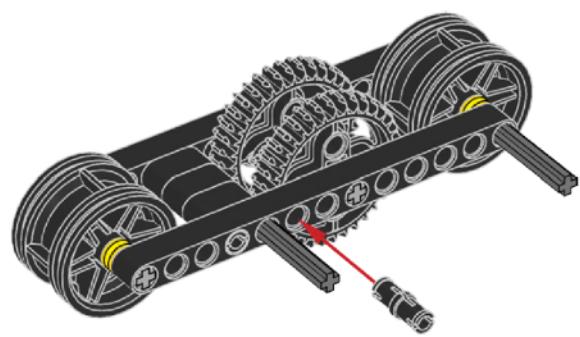


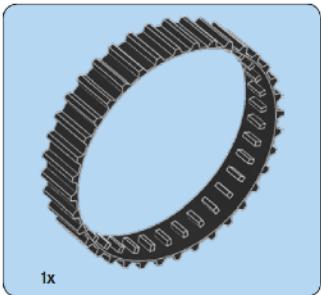


9

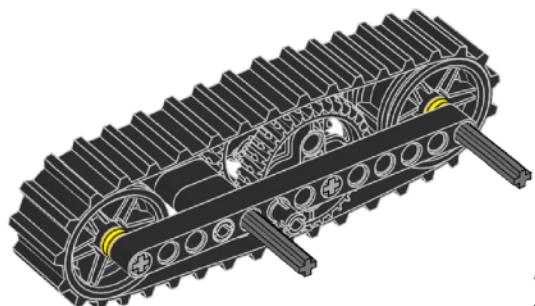


10

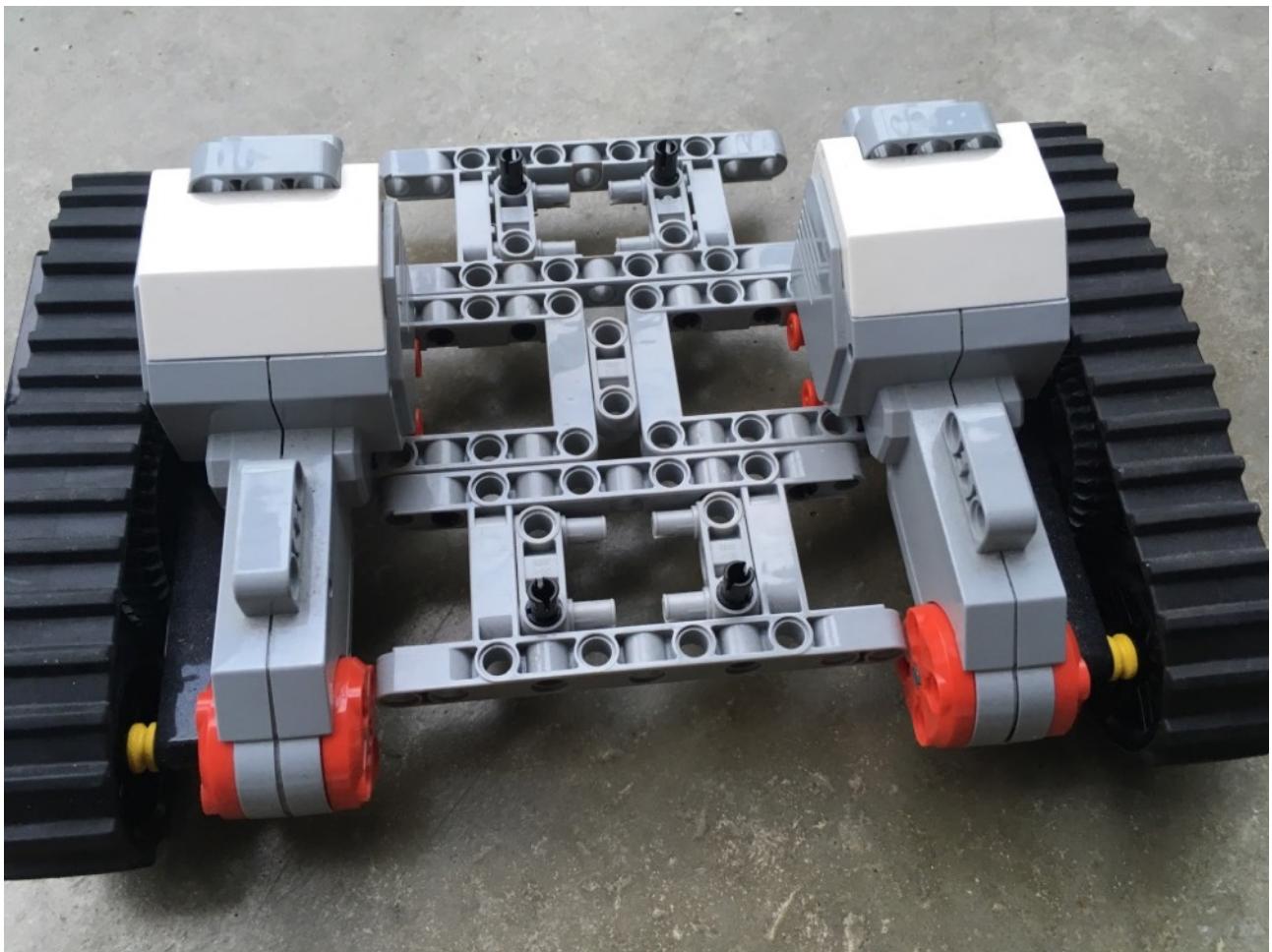




11



2x

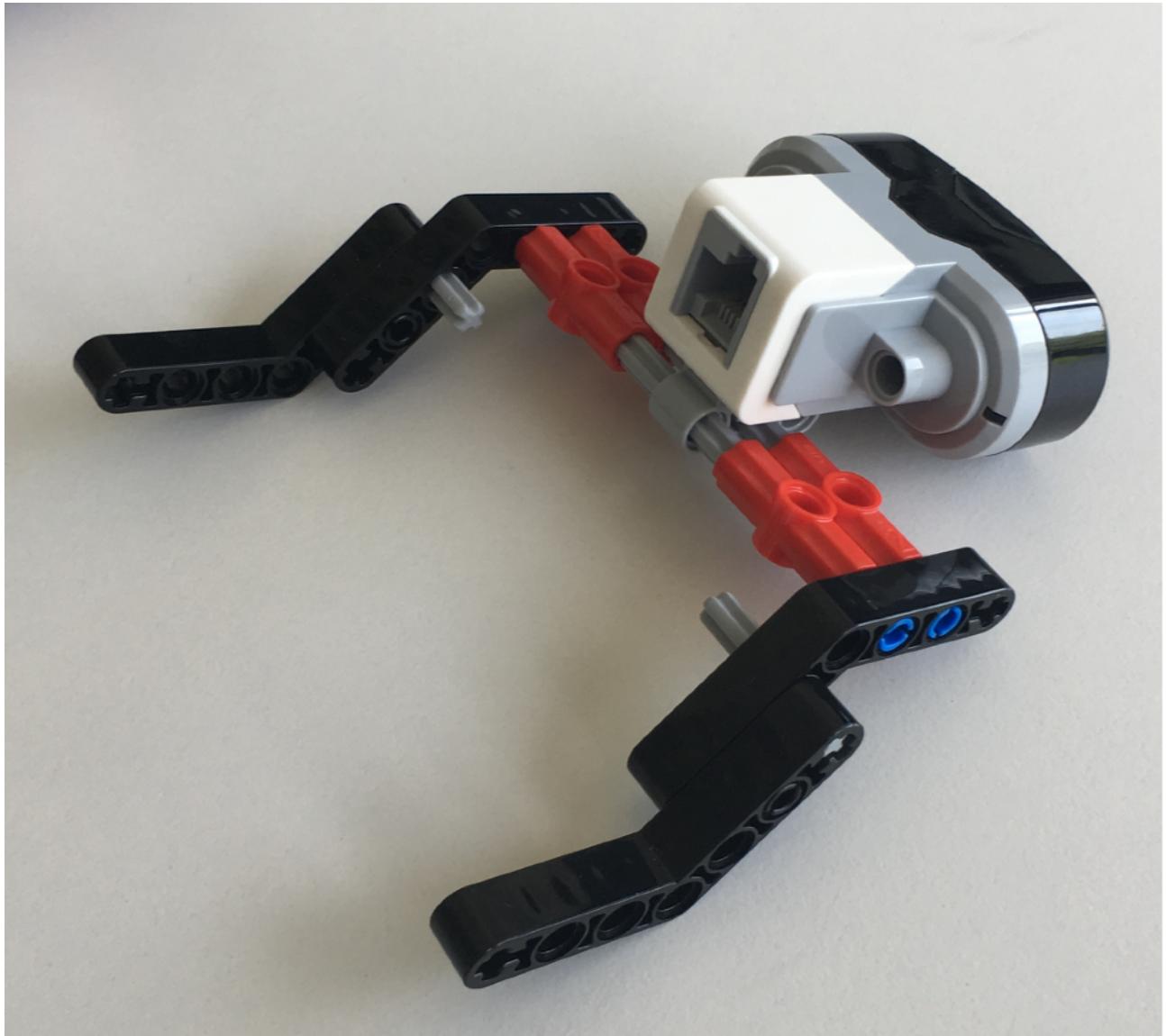


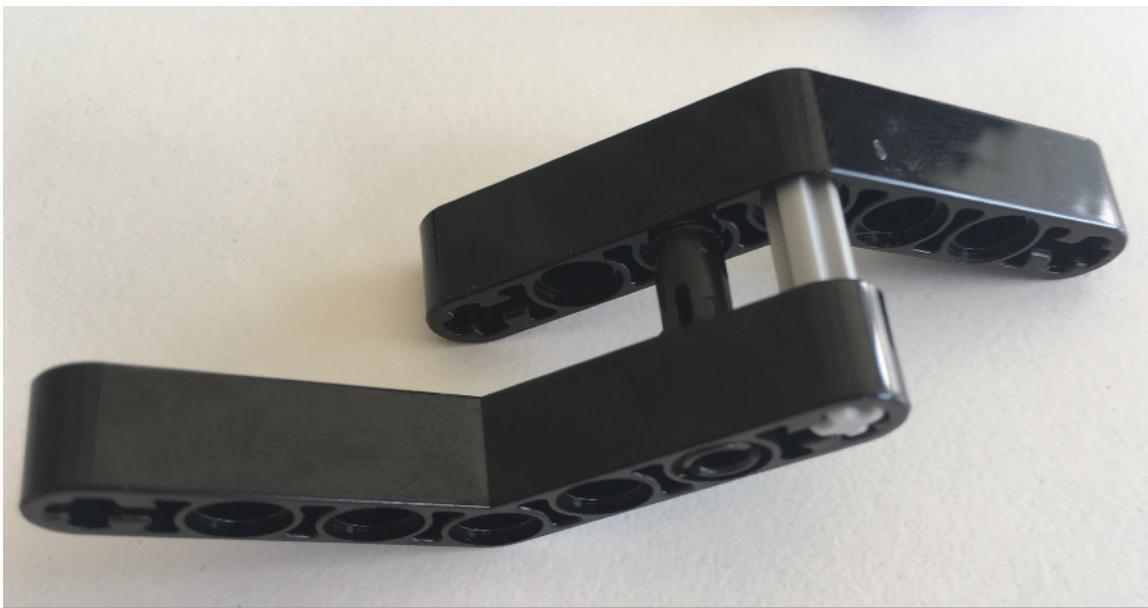
Robot ManRobot: wings and action button

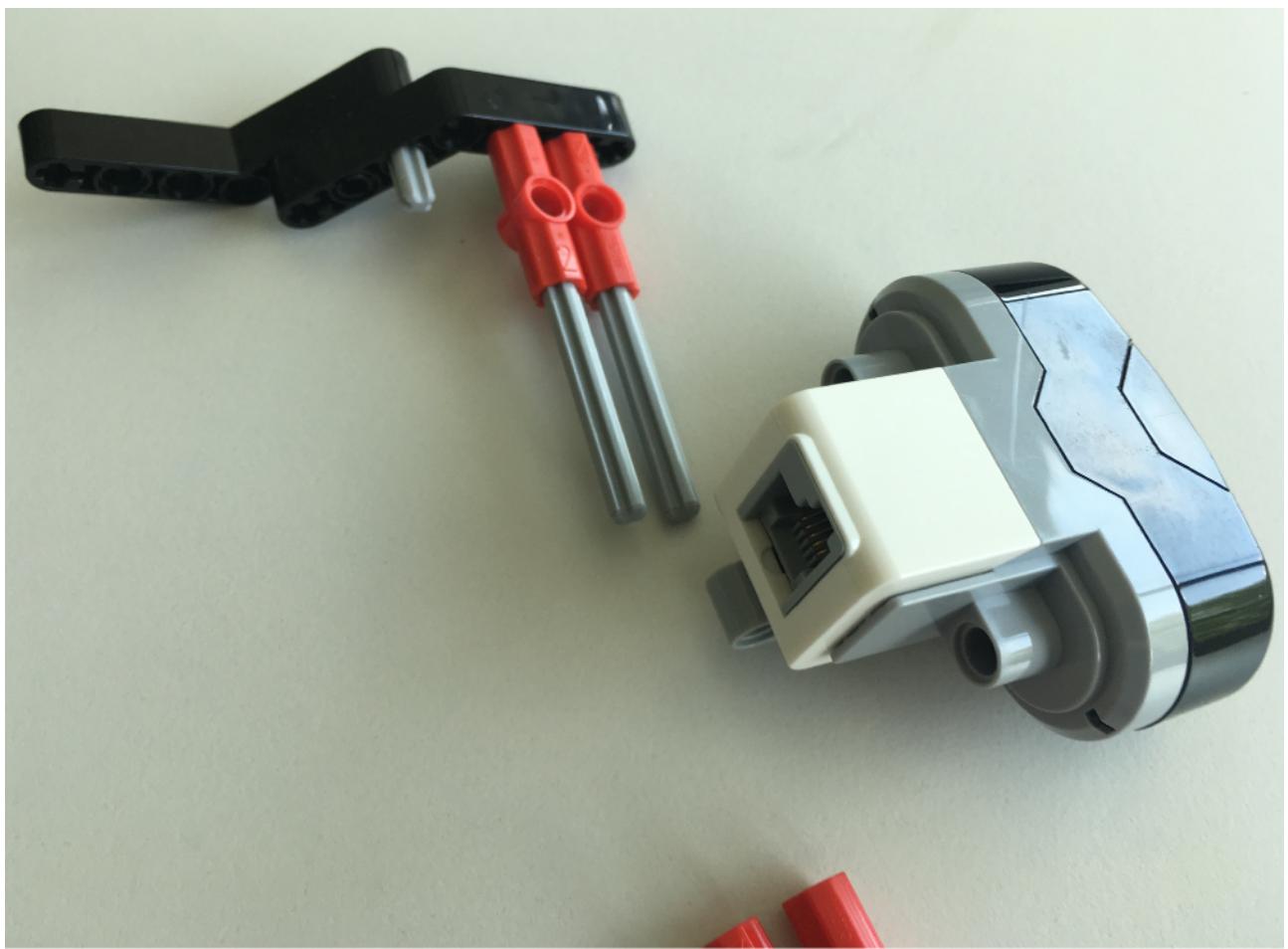
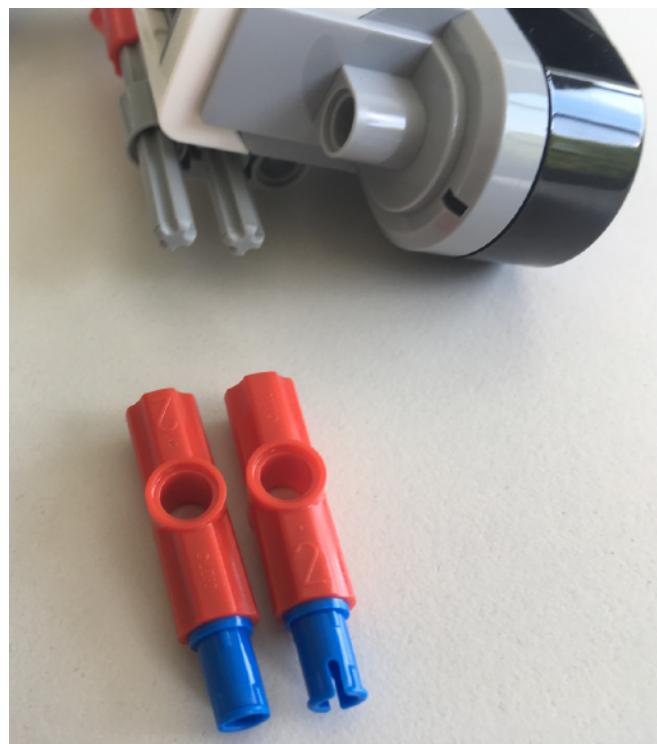


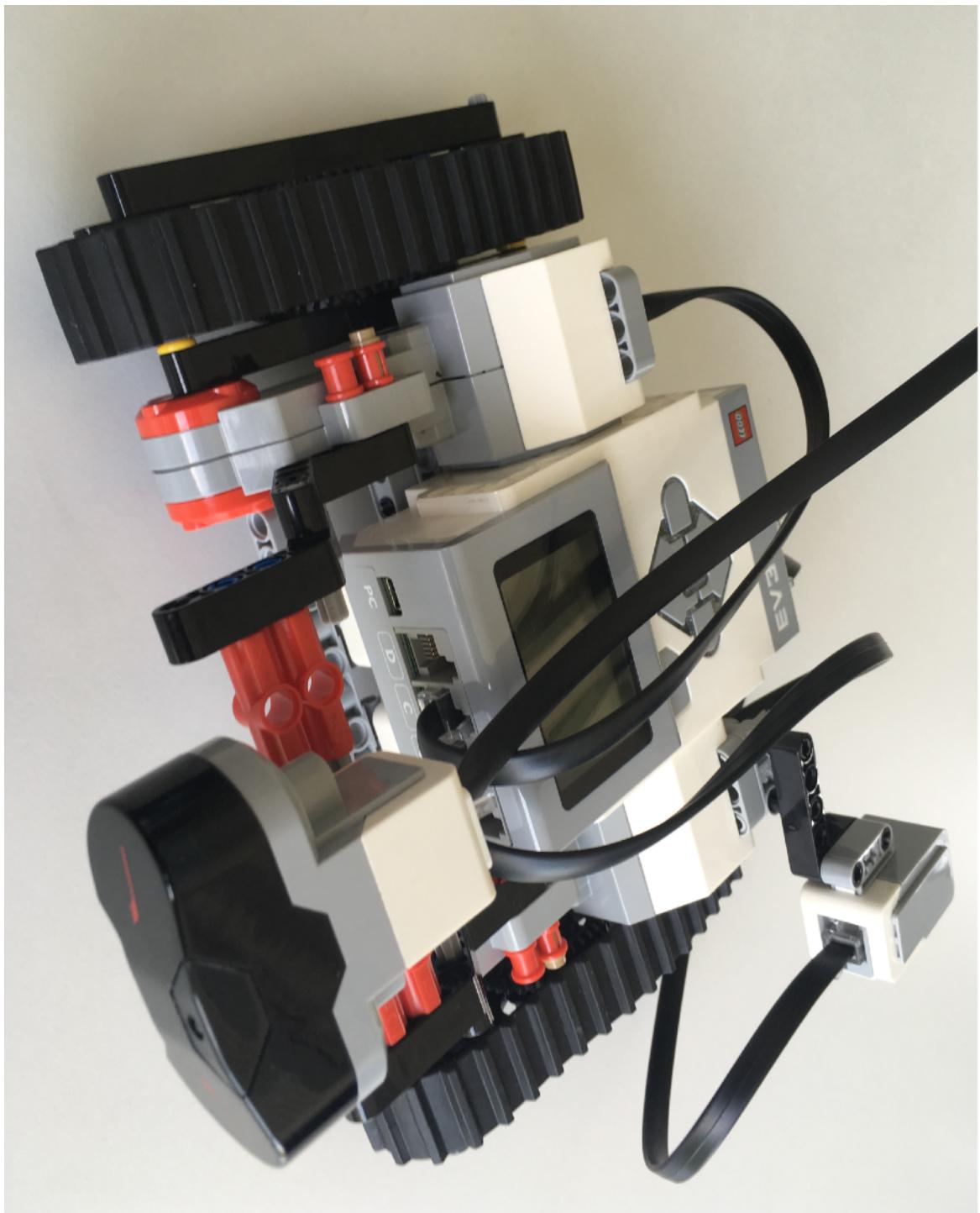


ManRobo Sensor : infra red sensor





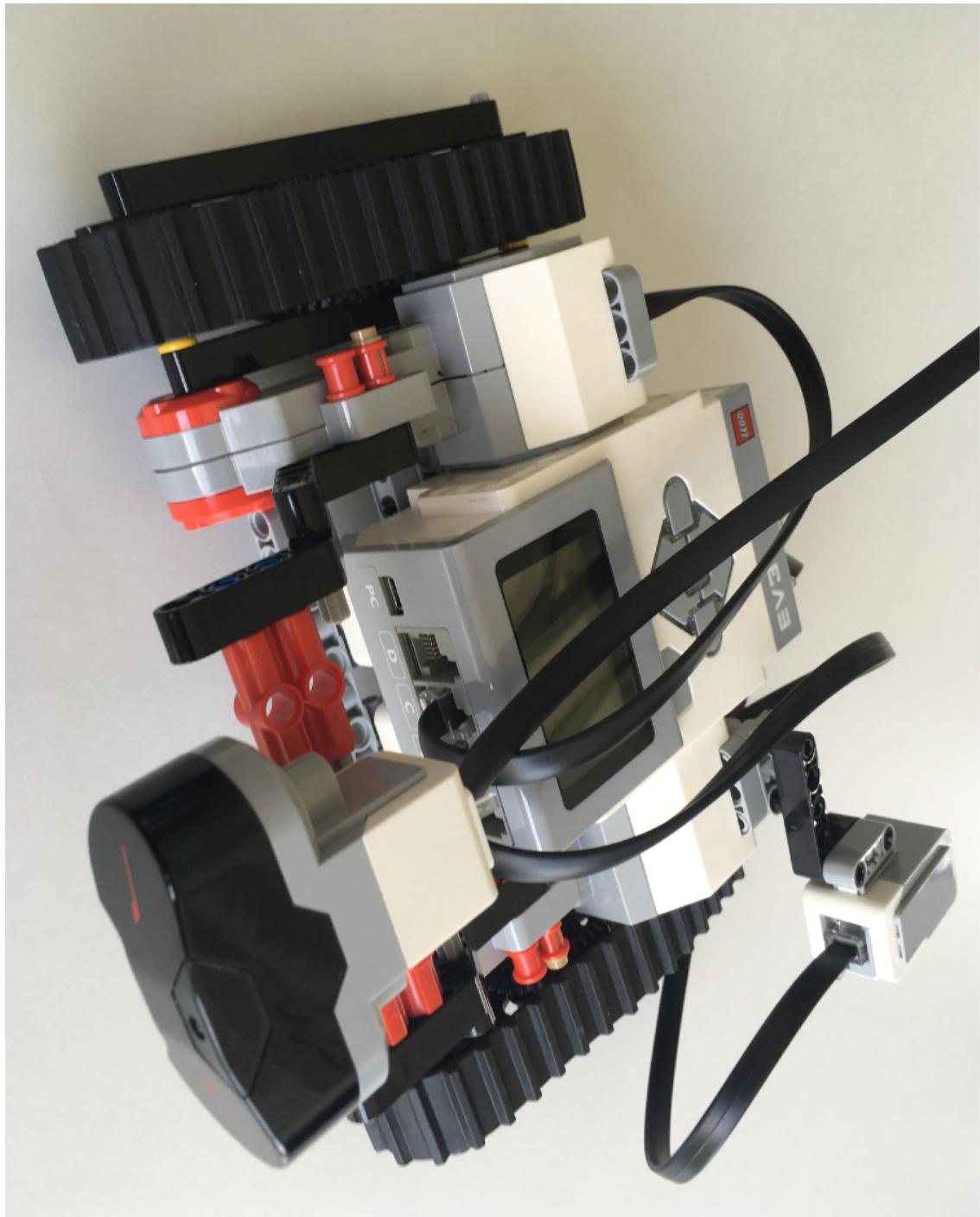




ManRobot wire motor and button

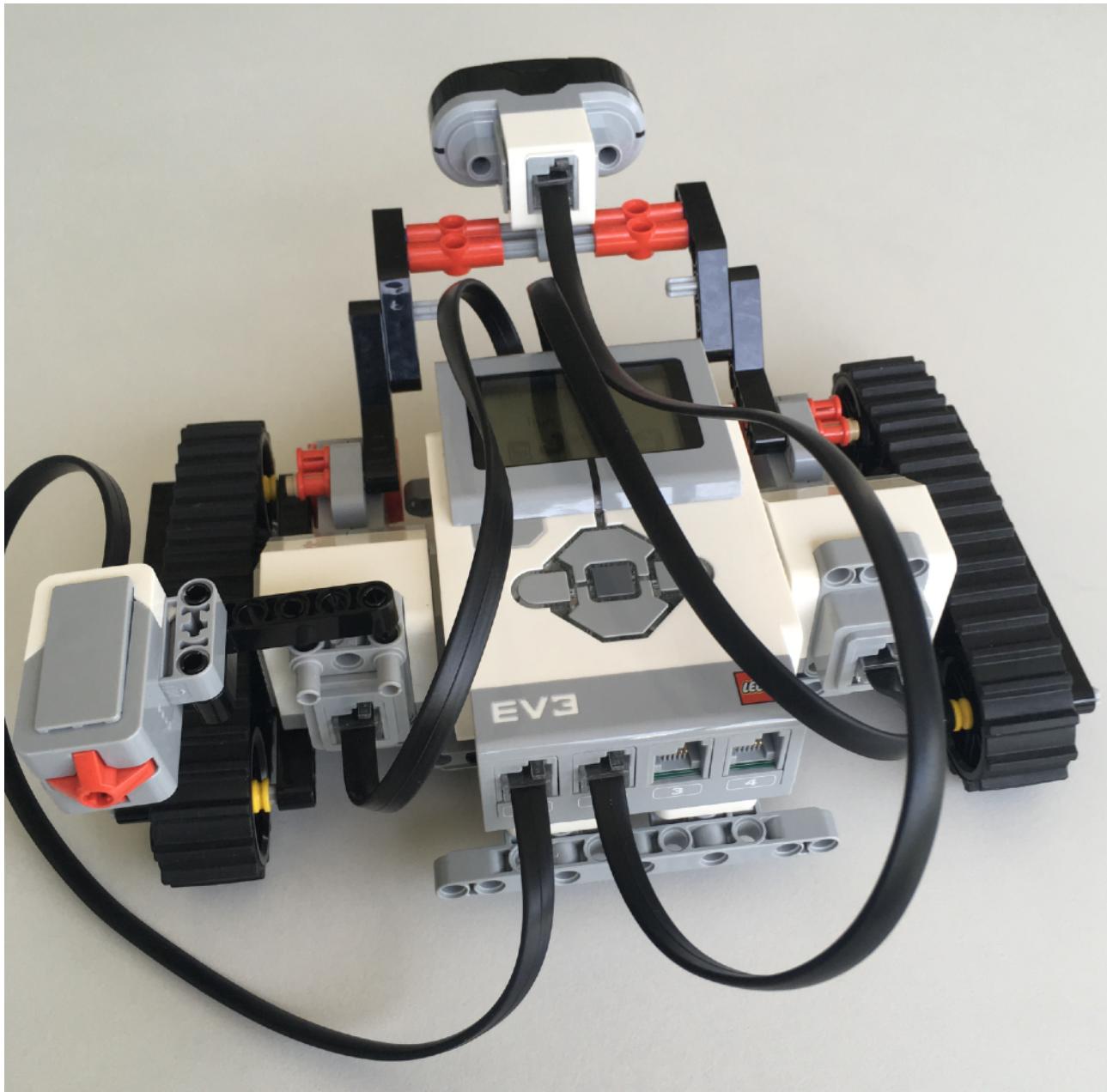
Motors:

1. Left Motor to the Socket B
2. Right Motor to the Socket C



Push Button and InfraRed Sensor:

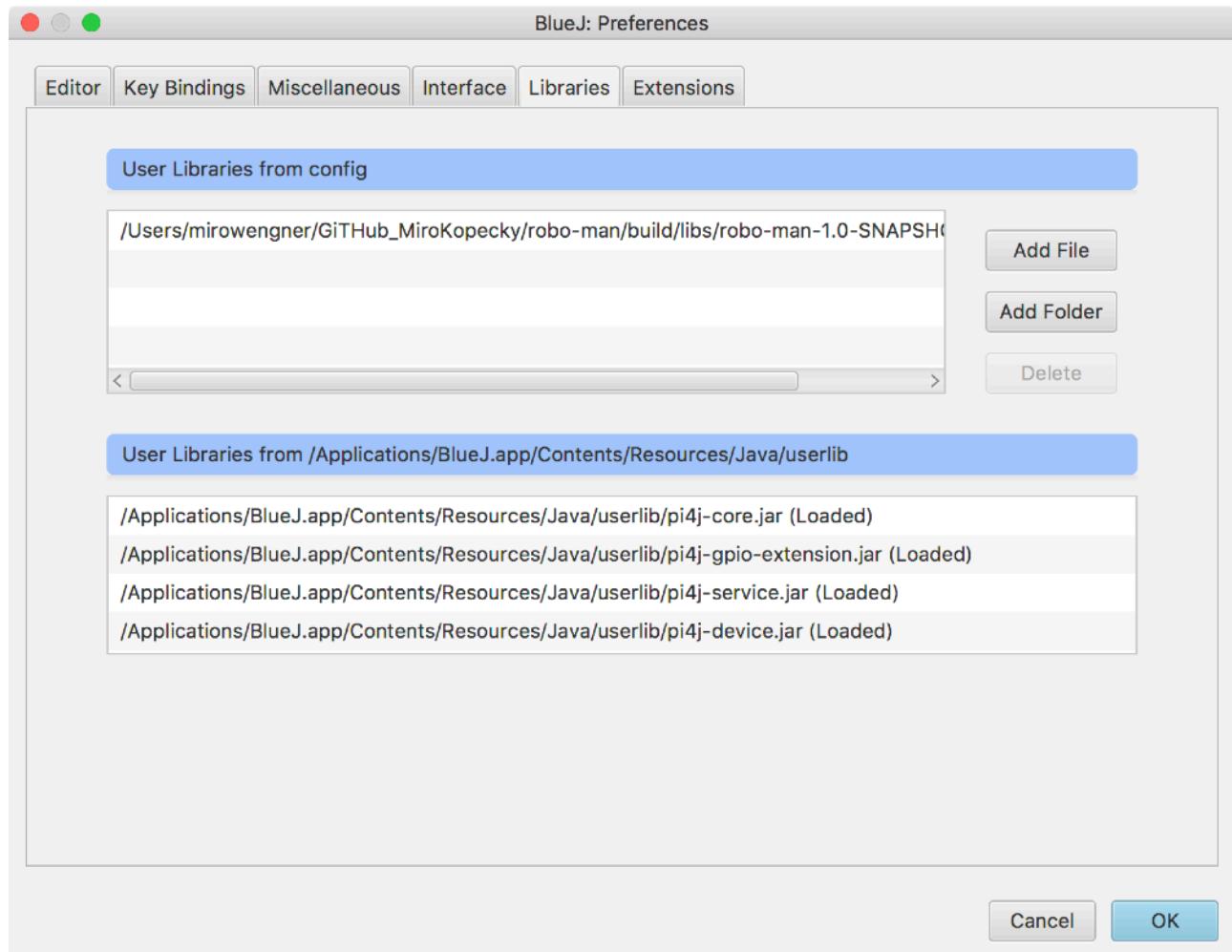
1. Push Sensor to the Socket 1
2. InfraRed Sensor to the Socket 2



BlueJ IDE Project

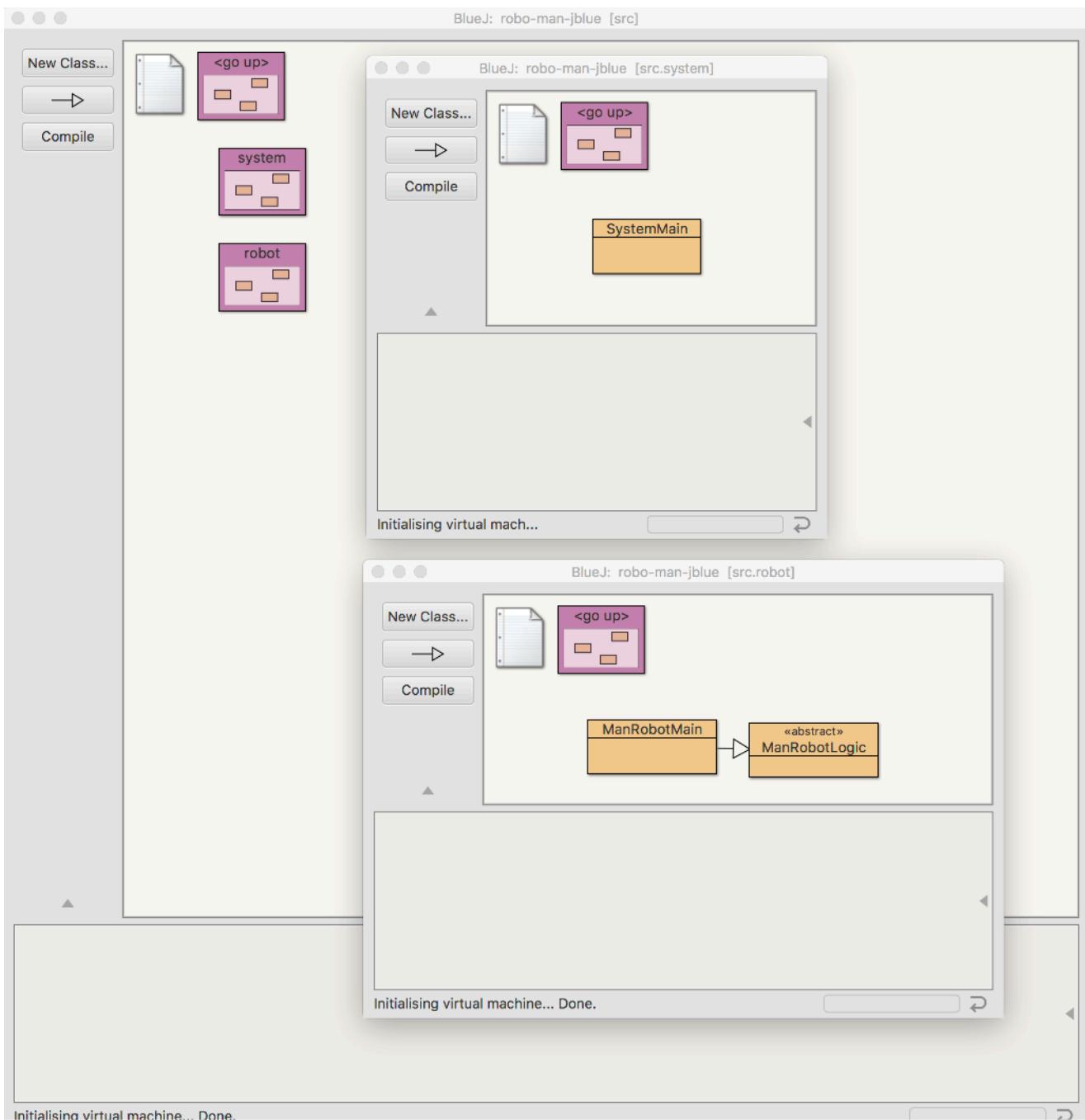
Open BlueJ IDE

1. go to Preferences -> Libraries -> Add File
2. select robo-man-1.0-SNAPSHOT.jar
3. close BlueJ IDE
4. start BlueJ



BlueJ IDE Project

Open robo-man-jblue project



BlueJ IDE Project

Implement ManRobot logic

The screenshot shows the BlueJ IDE interface with the title bar "ManRobotMain - robo-man-jblue". The toolbar includes "Compile", "Undo", "Cut", "Copy", "Paste", "Find...", "Close", and a dropdown menu "Source Code". The code editor displays the following Java code:

```
package src.robot;

import com.robo4j.man.demo.LegoBrick;
import com.robo4j.man.demo.LeftMotor;
import com.robo4j.man.demo.RightMotor;
import com.robo4j.man.demo.TouchSensor;
import com.robo4j.man.demo.InfraRedSensor;
import com.robo4j.man.demo.RobotCenter;

/**
 * ManRobotMain represent the main entry point for the
 * ManRobot program. It is necessary to implement the logic.
 *
 * After the robot logic has been implemented the SystemMain.main
 * method can be executed in order to process defined action.
 * For more information follow the doc in SystemMain class
 *
 * @author Miroslav Wengner(@miragemiko)
 */
public final class ManRobotMain extends ManRobotLogic
{
    public static void main(String[] args){
        ManRobotMain robot = new ManRobotMain();
        robot.run();
    }

    protected void logic(){
        // ManRobot logic is here
    }
}
```

The code editor highlights the main() method and the logic() method with a yellow background. The status bar at the bottom right shows the word "saved".

BlueJ IDE Project

Implemented ManRobot logic

The screenshot shows the BlueJ IDE interface with the title bar "ManRobotMain - robo-man-jblue". The main window displays the source code for the `ManRobotMain` class. The code implements the `ManRobotLogic` interface, which includes a `main` method and a `logic` method. The `main` method creates a new `ManRobotMain` object and calls its `run` method. The `logic` method adds various devices to a brick: a LeftMotor ("B"), a RightMotor ("C"), a TouchSensor ("S1"), and an InfraRedSensor ("S2"). The code is annotated with Javadoc-style comments explaining the purpose of the class and its methods.

```
package src.robot;

import com.robo4j.man.demo.LegoBrick;
import com.robo4j.man.demo.LeftMotor;
import com.robo4j.man.demo.RightMotor;
import com.robo4j.man.demo.TouchSensor;
import com.robo4j.man.demo.InfraRedSensor;
import com.robo4j.man.demo.RobotCenter;

/**
 * ManRobotMain represent the main entry point for the
 * ManRobot program. It is necessary to implement the logic.
 *
 * After the robot logic has been implemented the SystemMain.main
 * method can be executed in order to process defined action.
 * For more information follow the doc in SystemMain class
 *
 * @author Miroslav Wengner(@miragemiko)
 */
public final class ManRobotMain extends ManRobotLogic
{
    public static void main(String[] args){
        ManRobotMain robot = new ManRobotMain();
        robot.run();
    }

    protected void logic(){
        brick.addDevice(new LeftMotor("B"));
        brick.addDevice(new RightMotor("C"));
        brick.addDevice(new TouchSensor("S1"));
        brick.addDevice(new InfraRedSensor("S2"));
    }
}
```

BlueJ IDE Project

Execute ManRobot project and upload to the LegoBrick

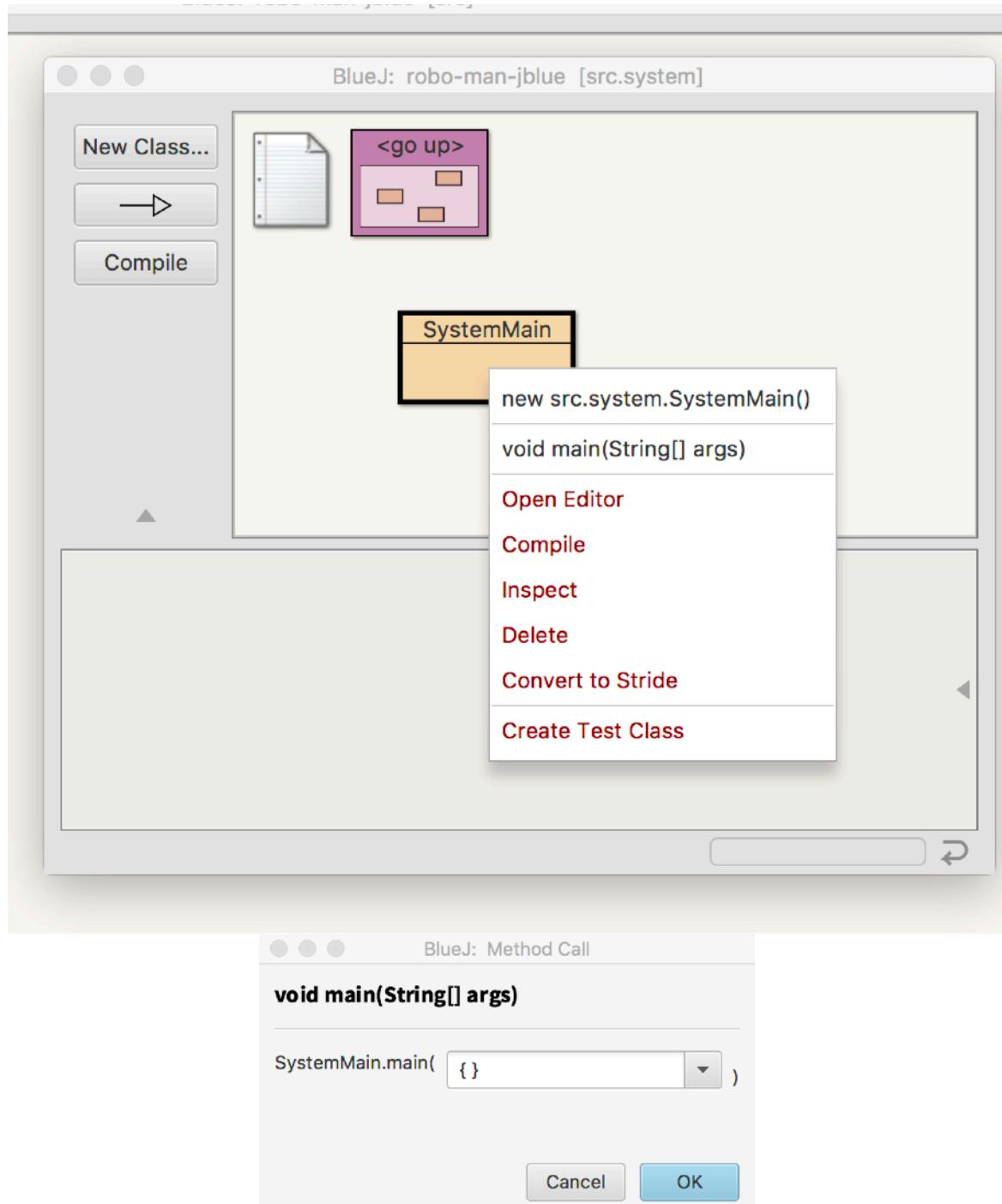
requirements:

LegoBrick is connected with the Computer System over W-LAN, Bluetooth or USB cable.

LegoBrick is present on IP address: 10.0.1.1

Steps:

1. Compile SystemMain class
2. right click and execute void main(String[] args) method
3. small execution window is displayed -> Click OK



BlueJ IDE Project

1. Execution process will be displayed in separate console window.
2. wait until the process is FINISHED

The screenshot shows a terminal window titled "BlueJ: Terminal Window - robo-man-jblue". The window contains the following text output:

```
System Start
Compile STARTS
MANIFEST.MF
JAR file ManRobot has been created
FINISHED...
```

At the bottom of the window, there is a message: "Can only enter input while your programming is running".

EV3 Lego Brick Run ManRobot

1. In Menu go to Samples
2. Find ManRobot
3. Excetue

