



# Ein Prozess lernt laufen

LEGO® MINDSTORMS® Steuerung mit BPMN

Oliver Hock  
Videa Project Services GmbH



Videa Services GmbH



oliver.hock@videa.services



JUG Hamburg



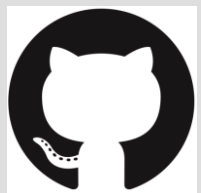
Kids4IT Hamburg



Oliver\_Hock3



@oliver\_hock



olihock / bepemind

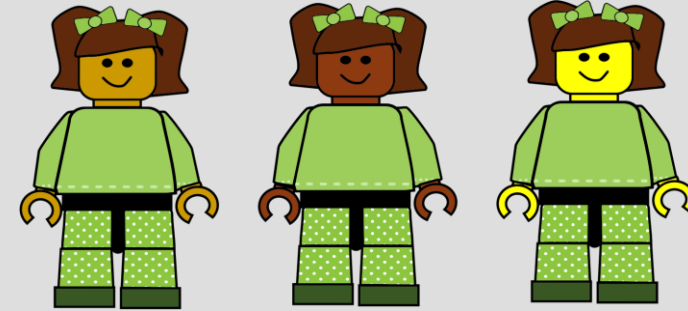
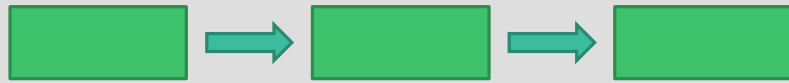


Tae Kwon Do + Sailing

# The Idea



Knowledge

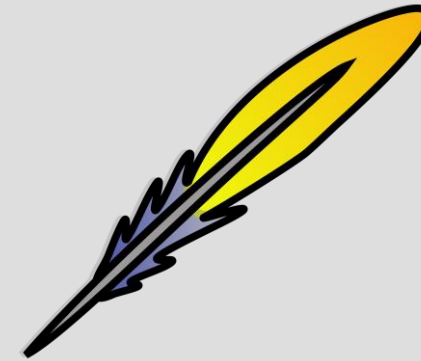


Curiosity

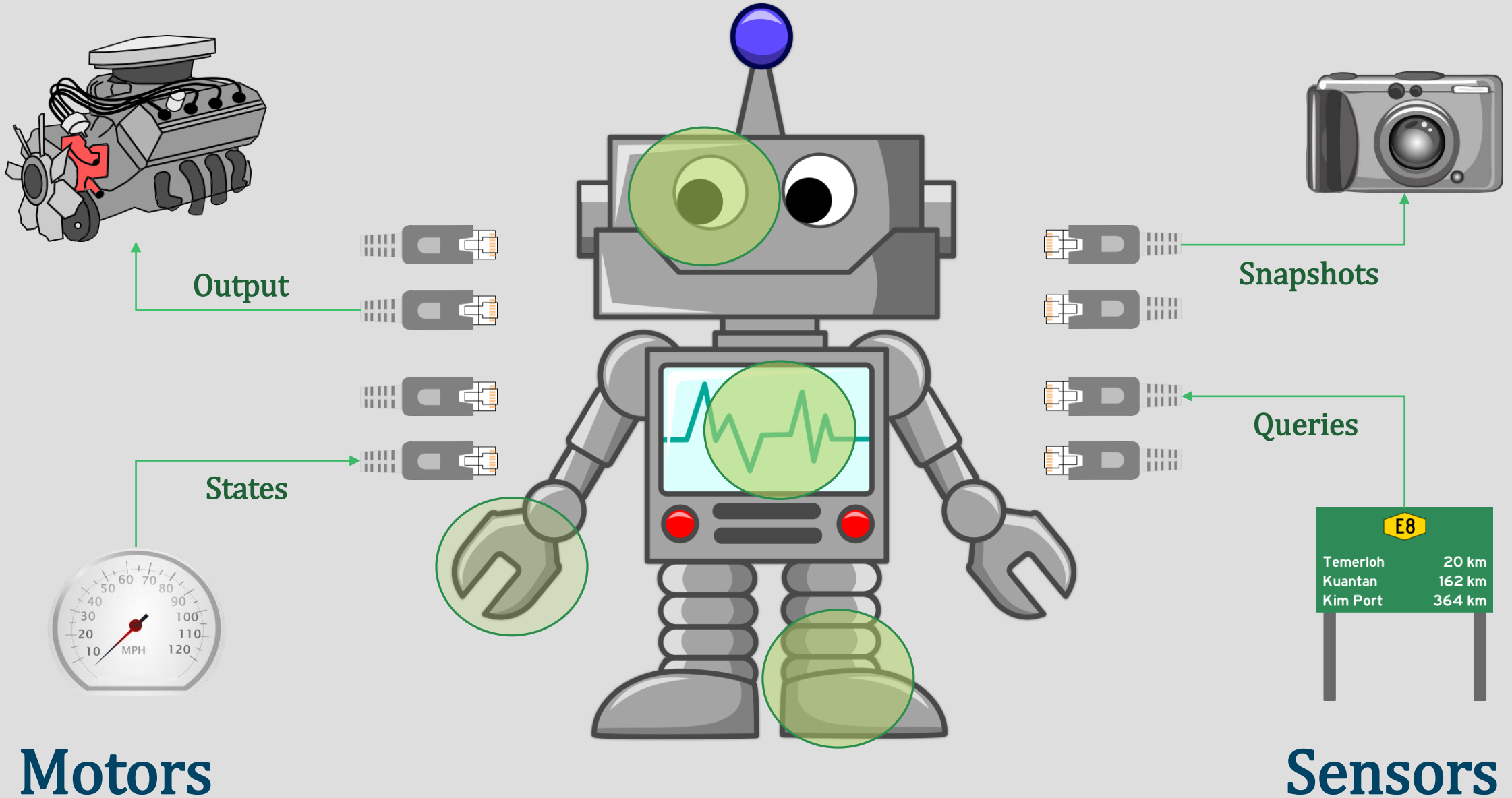
Precision



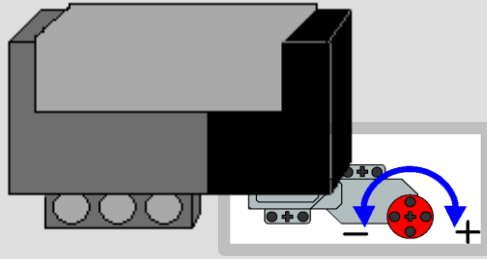
Fun



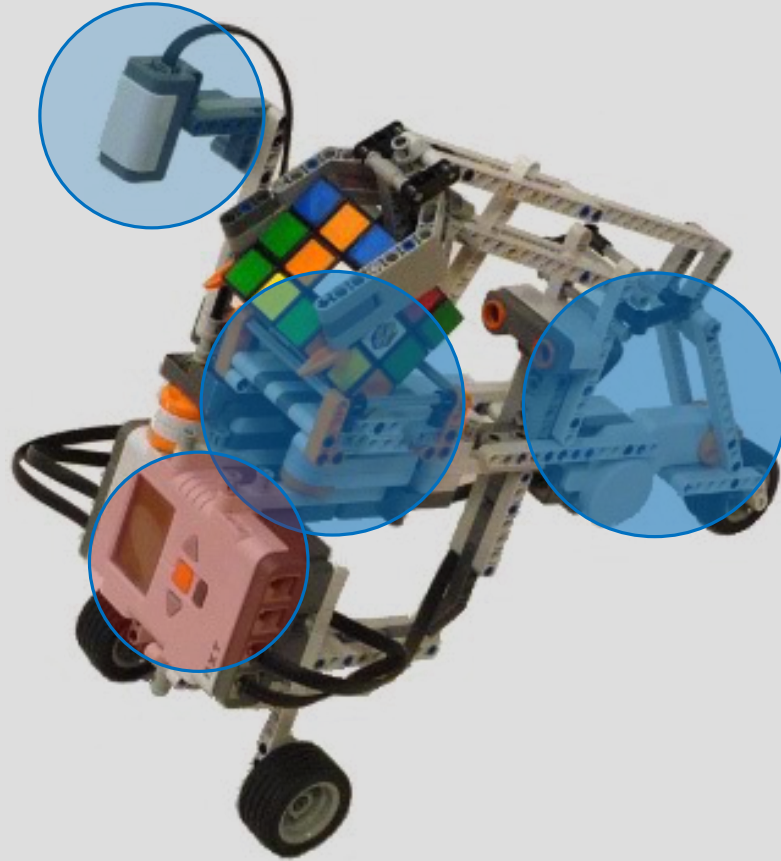
# A Robot



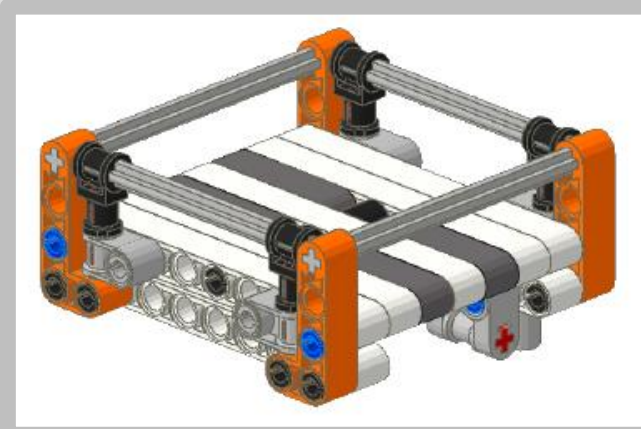
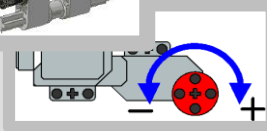
# TiltedTwister 2.0 Set-up



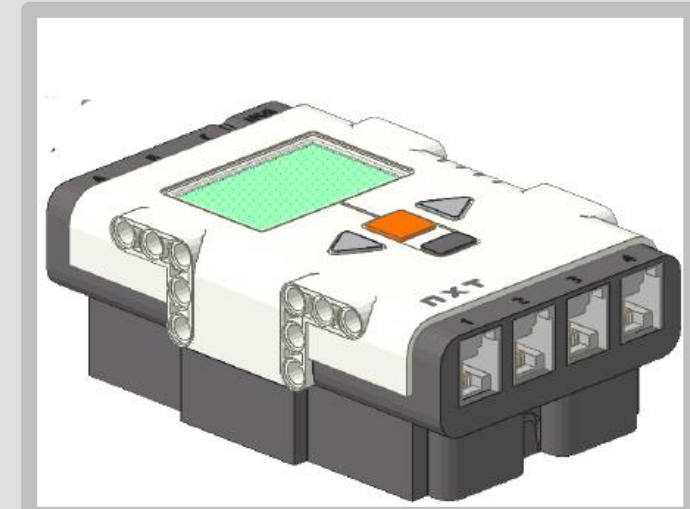
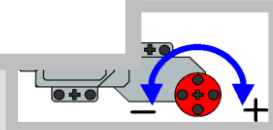
Color Sensor



Cube Tilter



Turn Table

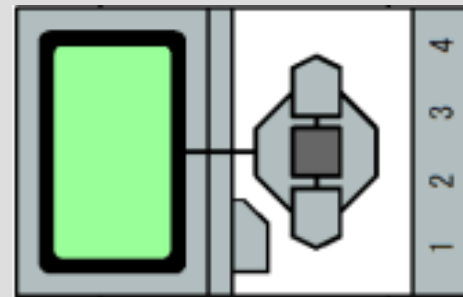
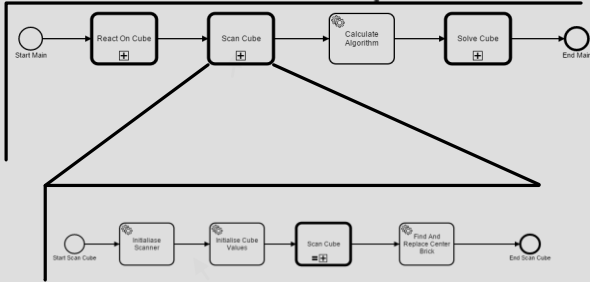


Brick

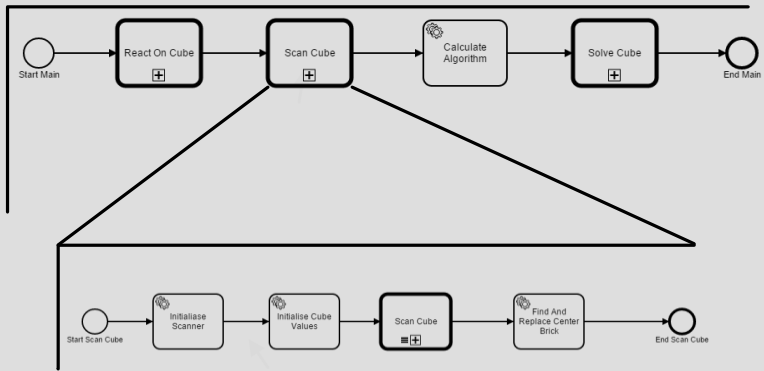
# The Architecture



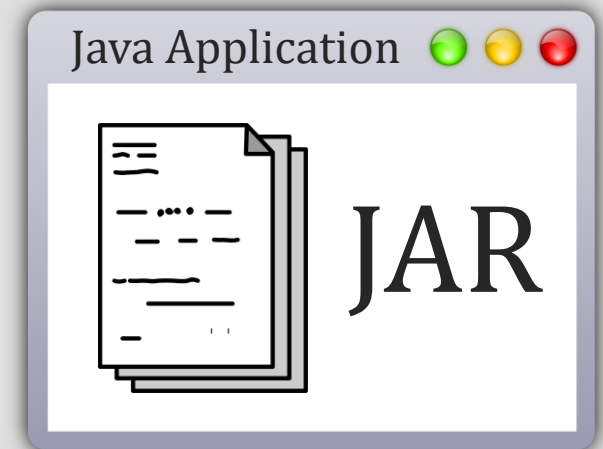
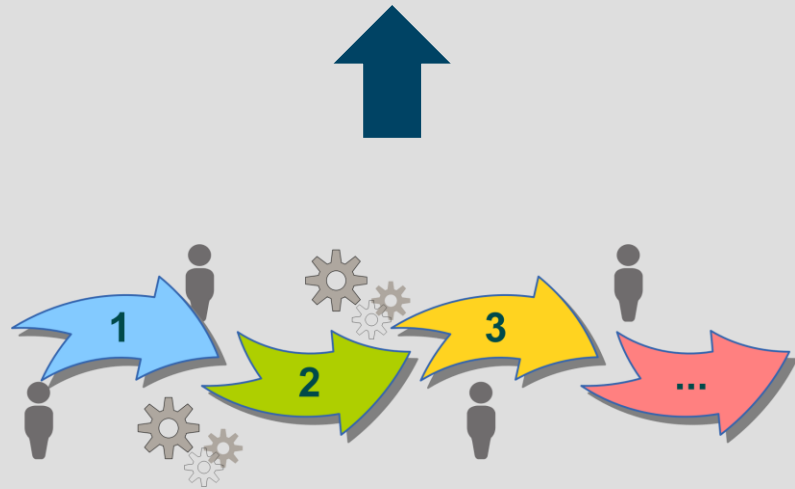
camunda BPM platform



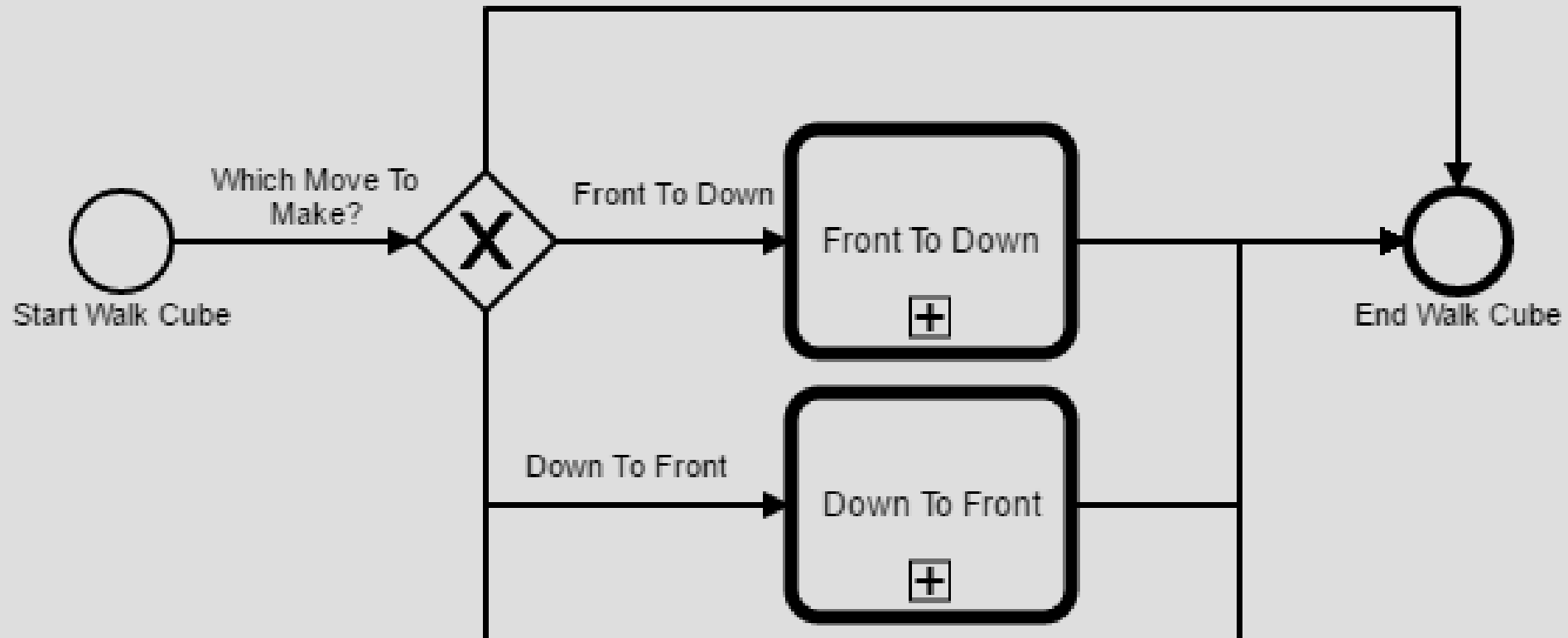
# BPM Platform



  
camunda BPM platform

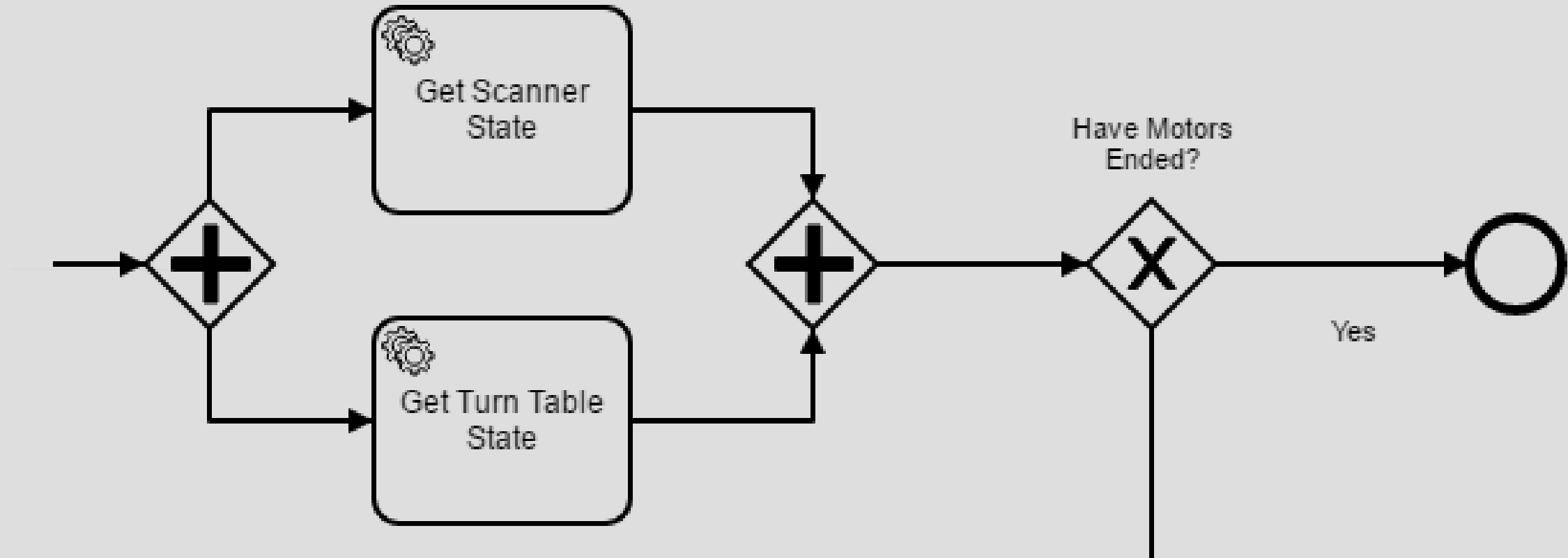


# Business Process Model and Notation

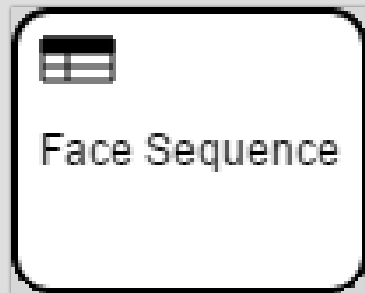





# Business Process Model and Notation



# Business Process Model and Notation



Face Sequence		Hide Details			
faceSequence					
U	Input +		Output +	Annotation	
	From Face	To Face	Move Sequence		
	fromFace	toFace	moveSequence		
	string	string	string		
1	"U"	"F"	"upperToFront"	-	
2	"F"	"D"	"frontToDown"	-	
3	"D"	"L"	"downToLeft"	-	
4	"L"	"R"	"leftToRight"	-	
5	"R"	"B"	"rightToBack"	-	
6	"B"	"U"	"backToUpper"	-	
+	-	-	-	-	

# CubeSolving Process Design

## Main Process

React on Cube

Scan Cube

Calculate

Solve Cube

Scan Face

Walk Cube

Move Face

Scan all Bricks

Face DMN

Move DMN

Face to Face

Scan single Brick

Rotate Motor

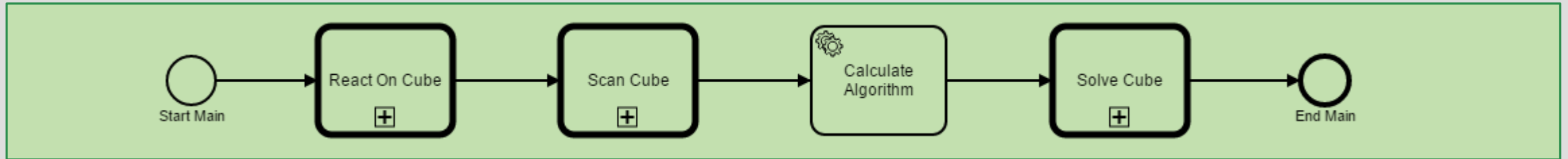
Turn

Tilt

Twist

# CubeSolving Project Design

Data Objects



CubeFinder Functions  
«JavaDelegate»

## Robot Access Delegate Layer

RotateMotorDelegate  
GetColorDelegate

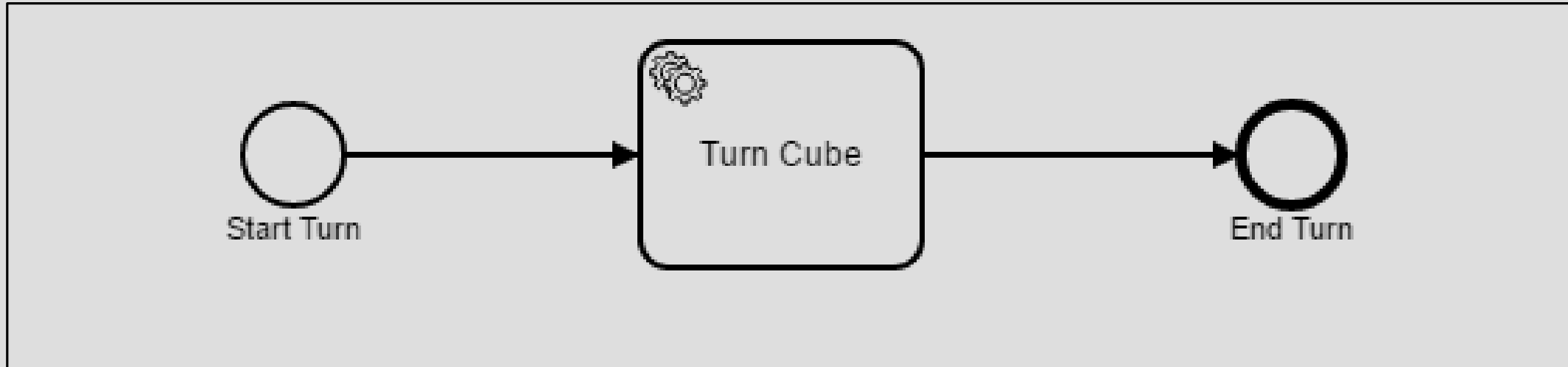
```
new MotorFactory()  
new SensorFactory()
```

## Brick Access Factory

```
static Motor.A  
static SensorPort.S1
```

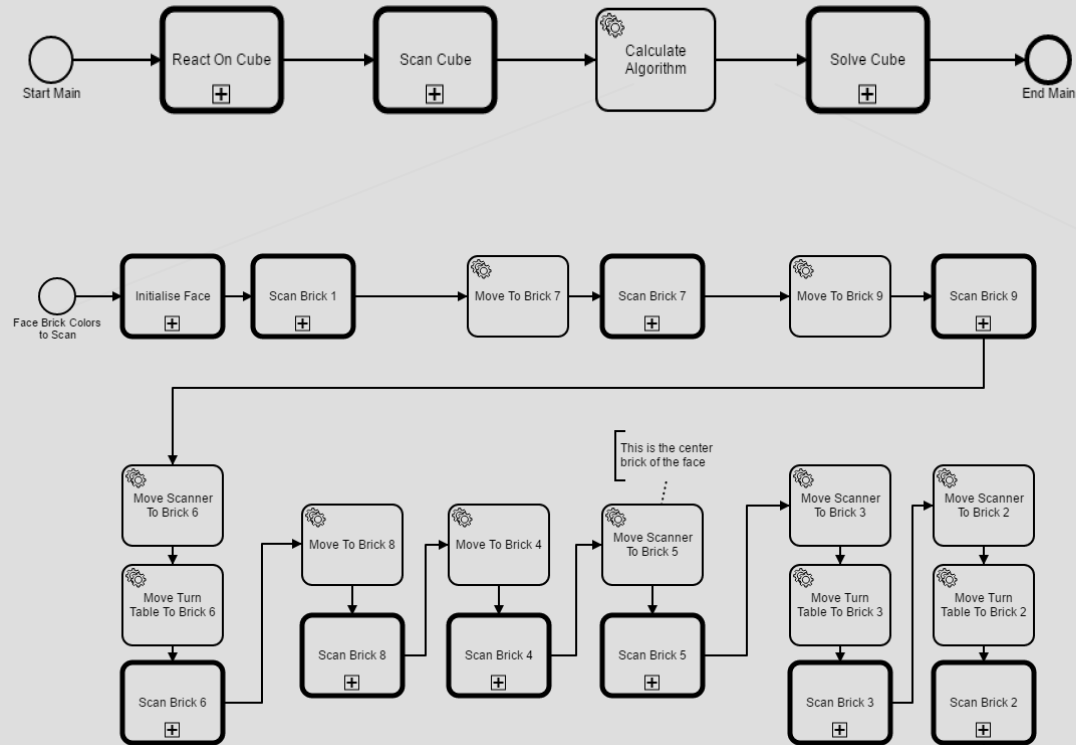
# Robot Access Delegates

## Process



```
public class RotateMotorDelegate implements JavaDelegate {  
  
    public void execute(DelegateExecution execution) throws Exception {  
        String motorPort = (String) execution.getVariable("motorPort");  
        Number speed = (Number) execution.getVariable("speed");  
  
        // leJOS stuff here
```

# Process Execution



```
@Deployment(resources = {
    "com/videaps/cube/solving/ScanCube.bpmn"
})
public class ScanCubeTest extends BaseTest {
```

```
@Test
public void test() {
    ProcessInstance processInstance =
        processEngine.getRuntimeService()
            .startProcessInstanceByKey(
                "Process_ScanCube");

    assertTrue(processInstance.isEnded());
}
```



Run as ...

# leJOS Remote API

```
package com.videaps.cube.solving.access.motor;

import lejos.nxt.Motor;

/**
public class MotorFactory {

    public RemoteMotor getMotor(String motorPort) {
        if("A".equalsIgnoreCase(motorPort)) {
            return Motor.A;
        } else if("B".equalsIgnoreCase(motorPort)) {
            return Motor.B;
        } else if("C".equalsIgnoreCase(motorPort)) {
            return Motor.C;
        }
        return null;
    }
}
```

# leJOS Motor API

```
int tachoCount = -1;
if(Features.USE_LEJOS.isActive()) {
    RemoteMotor motor = new MotorFactory().getMotor(motorPort);
    motor.setSpeed(speed!=null?speed.intValue():999);
    motor.setAcceleration(acceleration!=null?acceleration.intValue():0);
    motor.rotate(angle!=null?angle.intValue():0, immediateReturn);

    tachoCount = motor.getTachoCount();
}

execution.setVariable("rotateMotorTachoCount", tachoCount);
```



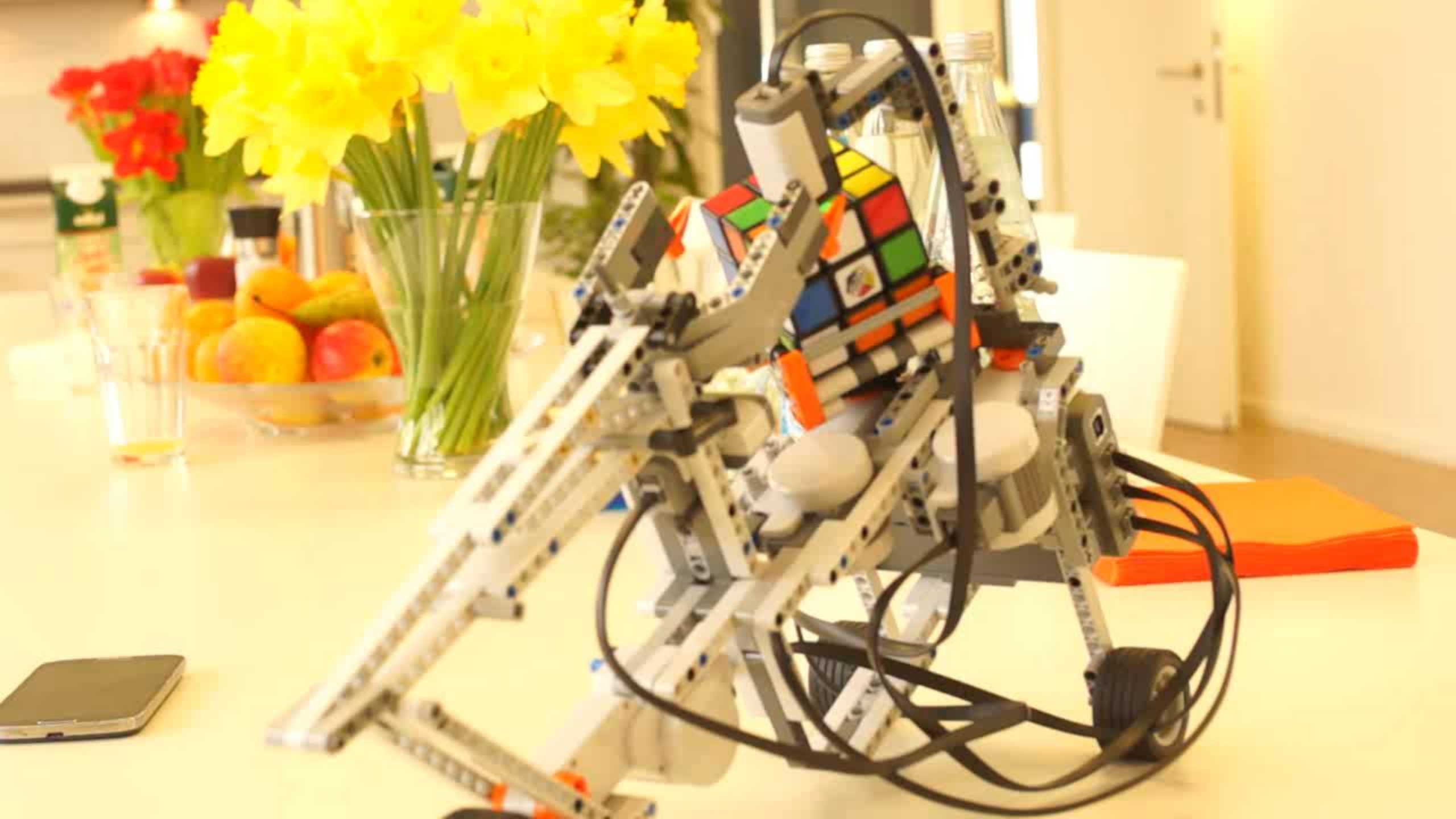
# leJOS Sensor API

```
int colorId = -1;
if(Features.USE_LEJOS.isActive()) {
    I2CPort port = new SensorFactory().getSensor(sensorPort);
    ColorHTSensor sensor = new ColorHTSensor(port);

    Color color = sensor.getColor();
    colorId = color.getColor();

    // The ColorHTSensor mixes up Red and Orange and always returns Red.
    // Therefore, the red color is treated separately in such, that
    // the RGB values are fetched and evaluated if red or orange.
    if(colorId == Color.RED) {
        colorId = distinguishRedAndOrange(colorId, color);
    }
}
String theColor = new ColorPicker().pickColor(colorId);

execution.setVariable("getColorColor", theColor);
```



# Thank you



**Oliver Hock**

Videa Project Services GmbH



Gustav-Heinemann-Ufer 56  
50968 Köln

Telefon: +49 221 3007 9270  
Mobil: +49 176 2949 9727

[oliver.hock@videa.services](mailto:oliver.hock@videa.services)



[www.videa.services](http://www.videa.services)

# Questions? Answers.