

---

# **suricate\_robot Documentation**

***Release 0.1***

**Francisco Javier Garcia R.**

**Jul 08, 2017**



---

## Contents

---

|          |   |          |
|----------|---|----------|
| <b>1</b> | <b>Contents:</b>                          | <b>3</b> |
| 1.1      | Description of suricate project . . . . . | 3        |
| 1.2      | How to install suricate robot . . . . .   | 5        |
| 1.3      | Suricate robot tutorials . . . . .        | 6        |
| 1.4      | Frequent questions . . . . .              | 9        |
| 1.5      | Contact us . . . . .                      | 10       |



Suricate project was developed as a master project on the institute for control systems in the University of Kaiserslautern.

Features:

- ROS enable (Tested under ubuntu 16.04 - ROS kinetic)
- Integrate Raspberry PI 2, ODROID XU4 and Arduino
- Web interface using rosbridge, NodeJS and MongoDB
- Joystick + tablet control
- Sensors: IMU (BNO055), Asus xtion pro Camera, Encoders, Voltage, current.
- Simulation of robot using Gazebo 7



## Description of suricate project

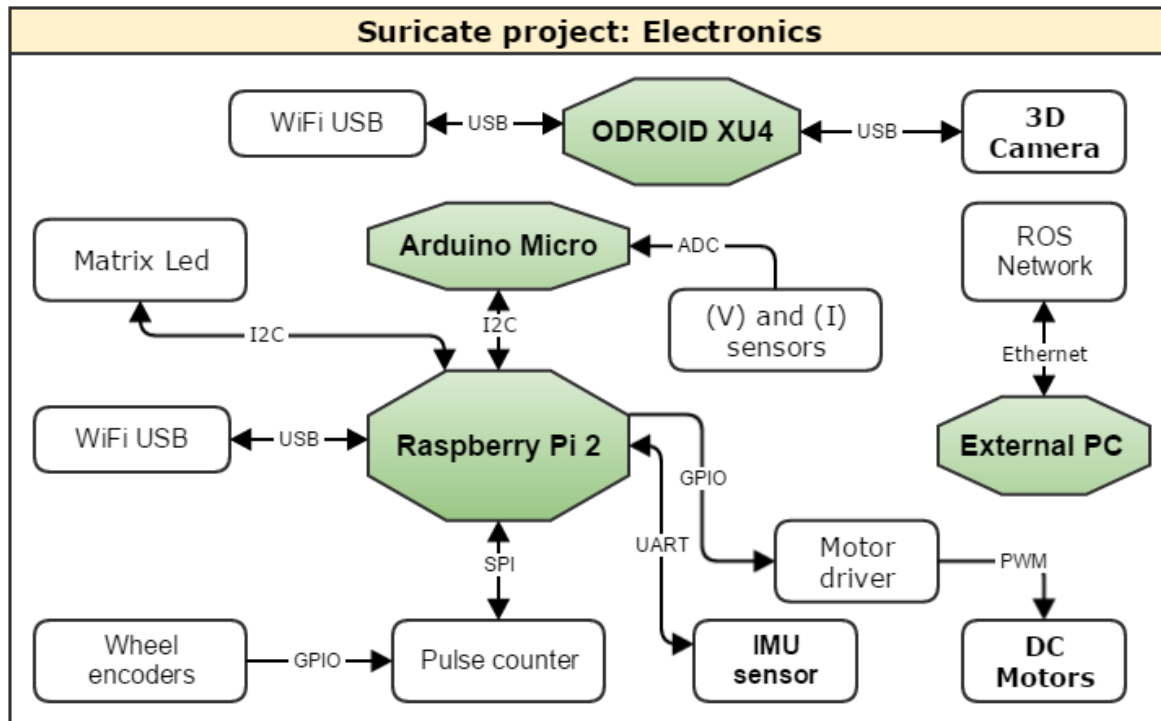
Developed using ROS (Robotic operating system) Kinetic under Ubuntu 14.04 Source code was written in Python and C++ Integrate external libraries such as OpenCV, NodeJS and MongoDB.

- Main task distribution:



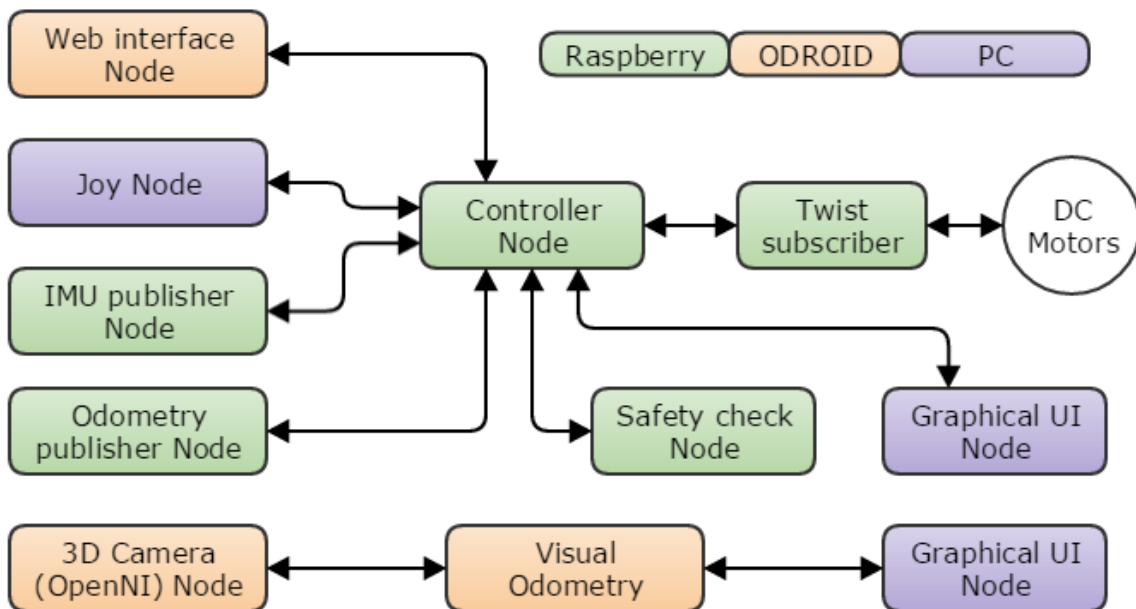
- Electronics:

The following figure shows how components are connected and which protocols were implemented.



- ROS nodes:

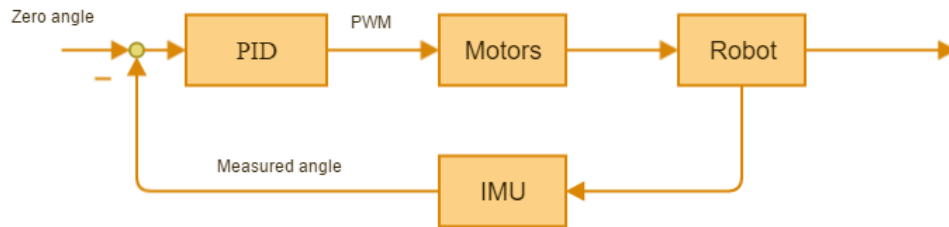
The following figure shows the ROS nodes which were developed for this project and which device execute each of them.



- PID controller:

The following figure shows a basic structure that represent the controller that we have developed.





## How to install suricate robot

In order to use this project, we must install first ROS (tested on Jade and kinetic), then compile the project and finally run some simulations and tests.

### Install ROS and dependencies

Follow ros installation procedure:

<http://wiki.ros.org/kinetic/Installation/Ubuntu>

we can summarize the steps:

Open a command windows on ubuntu and run the following commands:

- Prepare ubuntu for installation:

```

sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release -sc) main" > /
→etc/apt/sources.list.d/ros-latest.list'
sudo apt-key adv --keyserver hkp://ha.pool.sks-keyservers.net:80 --recv-key 0xB01FA116
sudo apt-get update

```

- Install ROS

For PC/Laptop we should install full desktop version:

```

sudo apt-get install ros-kinetic-desktop-full

```

For Raspberry PI 2 and Odroid we can install ROS-Base:

```

sudo apt-get install ros-kinetic-ros-base

```

Install rosdep:

```

sudo rosdep init
rosdep update

```

And finally prepare environment:

```

echo "source /opt/ros/kinetic/setup.bash" >> ~/.bashrc
source ~/.bashrc

```

## Compile project

First we need to install some dependencies:

Install pigpio library for GPIO access into raspberry PI 2, source: <http://abyz.co.uk/rpi/pigpio/download.html>

```
rm pigpio.zip
sudo rm -rf PIGPIO
wget abyz.co.uk/rpi/pigpio/pigpio.zip
unzip pigpio.zip
cd PIGPIO
make -j4
sudo make install
```

Then we should install some additional dependencies and ROS packages:

```
sudo apt-get install libqwt-dev ros-kinetic-teleop-twist-joy ros-kinetic-rviz-imu-
↳plugin python-smbus ros-kinetic-rqt-multiplot
```

Finally we create a workspace for project, clone github repository, install dependencies and compile it:

```
mkdir -p ~/catkin_ws/src
cd ~/catkin_ws/src
git clone https://github.com/francisc0garcia/suricate_robot
cd ..
source devel/setup.bash
rosdep install suricate_robot
catkin_make
```

## Test project

Once the project has been compiled successfully, we can run a simulation that includes suricate robot using a simple controller for stabilization.

```
cd ~/catkin_ws
source devel/setup.bash
roslaunch suricate_robot PC_simulation_project.launch
```

Now you are ready to play and extend the project, let's go to section Tutorials and extensions.

## Suricate robot tutorials

This section explain some examples of how to use the gazebo simulation and ROS in order to play with the suricate robot and develop some extensions.

### Initial considerations

- Credentials for Raspberri PI 2: user **ubuntu** - Password **ubuntu**
- Credentials for ODROID: user **odroid** - Password **odroid**
- Suricate robot is pre-configured for connecting to WiFi network **suricate** and password **suricate2015**, router should assign automatically static IP address

Raspberry PI2- *192.168.0.201*

ODROID - *192.168.0.203*

PC - *192.168.0.105*

this is important for following steps.

- If you want to change WiFi network configuration on Raspberry or ODROID, please modify config file (you can connect MicroSD on external PC with linux):

```
sudo nano /etc/network/interfaces
```

- You can change I2C and SPI parameters modifying file:

```
sudo nano /boot/config.txt
```

## How to start suricate robot

To run suricate robot, you need a working ROS project (explained on installation section). Once it is working, you can execute the following steps:

1. Turn on robot and wait for WiFi connection (default network **suricate** and static IP **192.168.0.201**).
2. By default, ODROID is configured to act as ROS Master using static IP *192.168.0.203*, and it will start automatically as soon as you power it on.
3. Use console to connect using SSH to raspberry PI2 (example: `ssh user@IP_address`):

```
ssh ubuntu@192.168.0.201
```

4. Start ROS nodes on RPI2 using the following commands:

```
cd ~/Documents/code/suricate_micro/
source devel/setup.bash
export ROS_IP=192.168.0.201
export ROS_MASTER_URI=http://192.168.0.203:11311
roslaunch robot_starter robot_starter.launch
```

By default, robot uses LQR controller that will start automatically, meaning that robot will stand up as soon as you execute `robot_starter.launch`, you might also try PID controlled that requires an initial velocity command given by **/robot/cmd\_vel** (usually provided if you connect a gamepad).

5. Now, if everything is correct, LED matrix should show a happy face. If does not do it, please check Frequent Questions section.
6. Open a new console and start PC ROS nodes for visualization and control, using following commands:

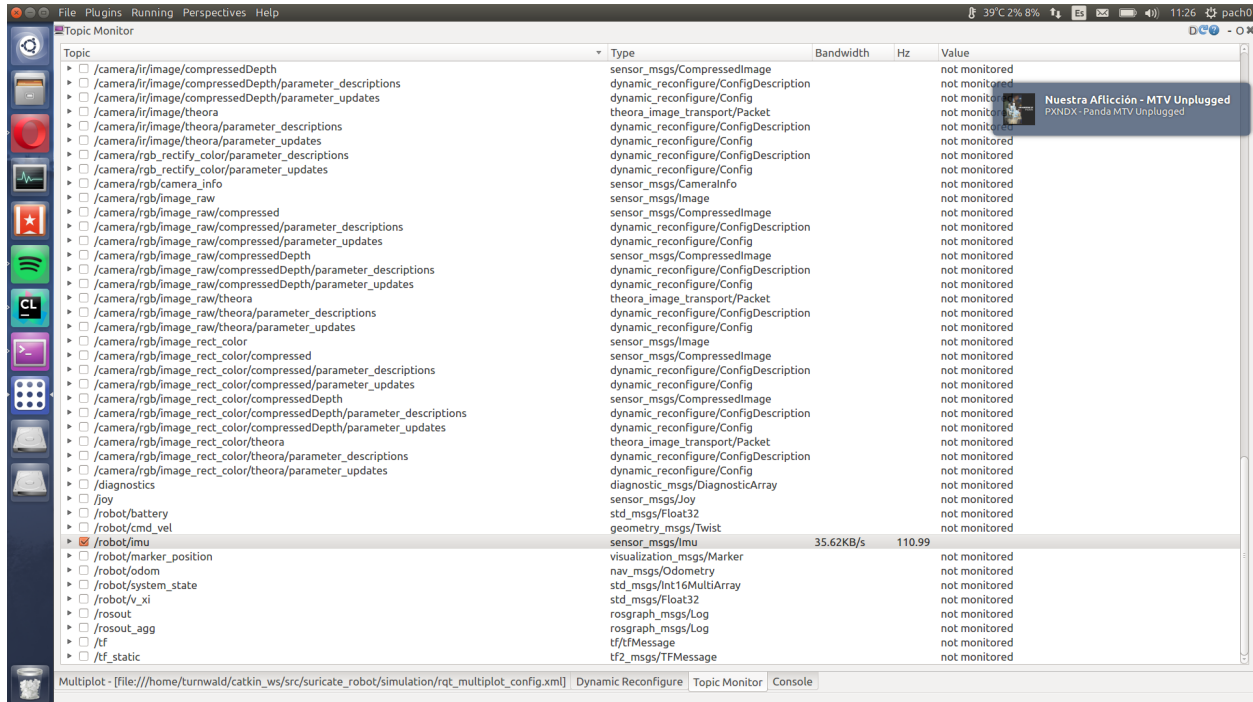
```
cd ~/catkin_ws/
source devel/setup.bash
export ROS_IP=192.168.0.105
export ROS_MASTER_URI=http://192.168.0.203:11311
roslaunch suricate_robot PC_launcher.launch
```

Recommendations:

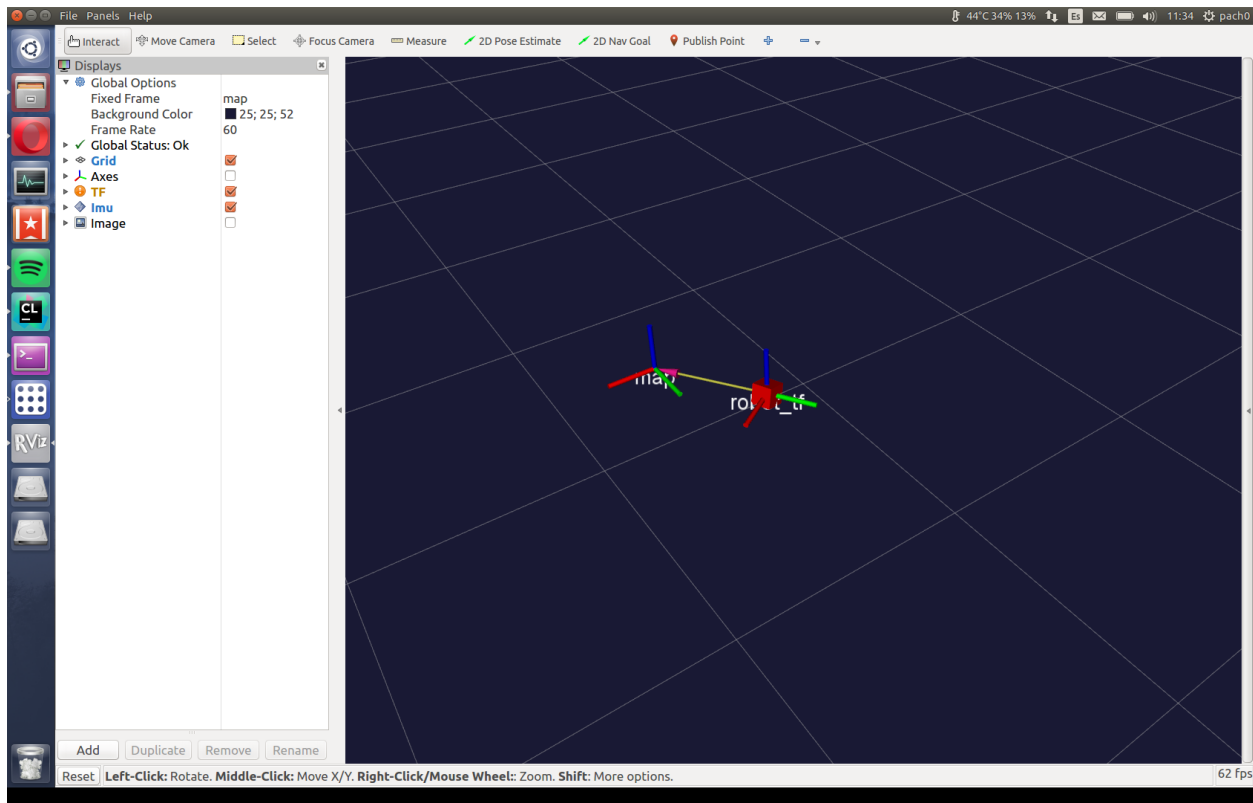
It is possible that your ROS project is place on different location.

Connect a gamepad before running this nodes, if you want to control robot with it.

7. Previous command will open a RQT instance (graphical interface), you can check robot state by clicking tab “Topic monitor”, and select a topic, for instance `/robot/imu`. If Hz shows something between 80 to 130 HZ IMU is properly sensed and robot work OK.



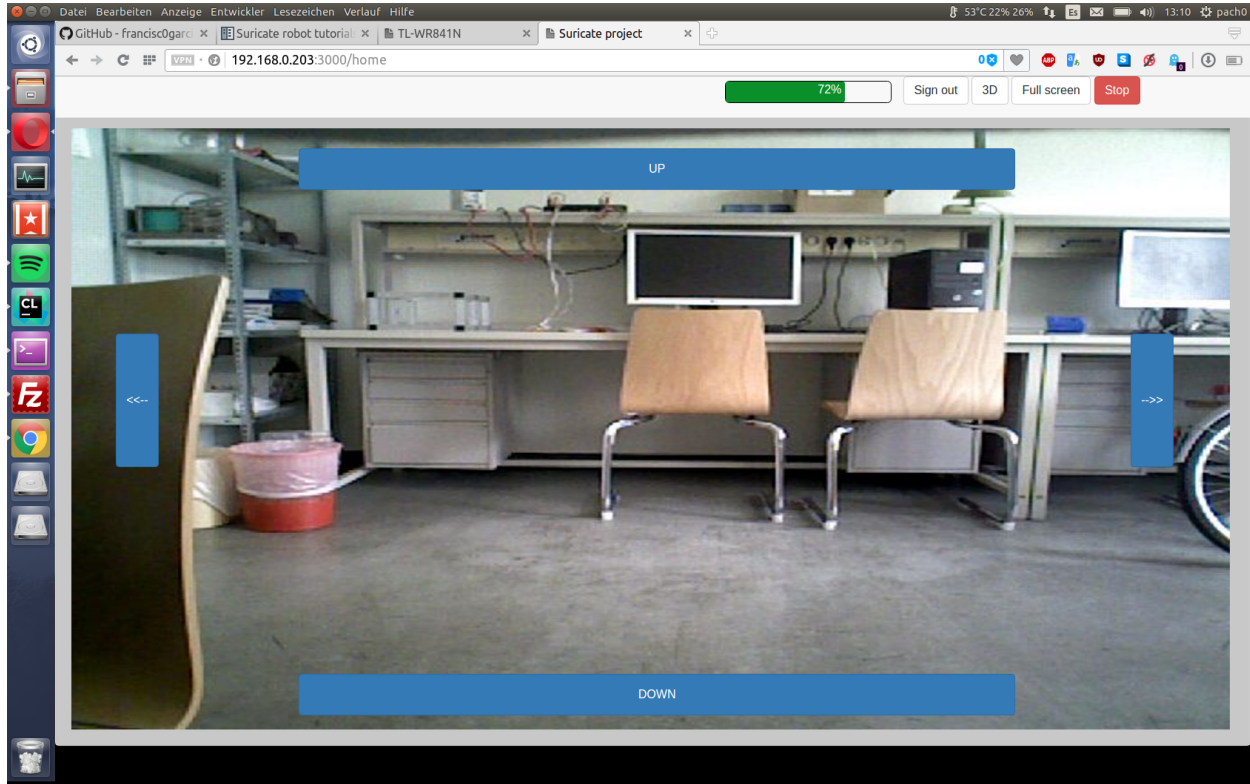
You should also get a RVIZ 3D graph, which, as soon as you move the robot, shows a computed position and pose.



8. It is also possible to use web interface, please open a web browser (tested on Google Chrome and Opera) and write this address:

```
http://192.168.0.203:3000
```

you must create a new user with password in order to get access to robot interface



It is possible to check camera image, battery status and 3D computed pose. If you use PID controller (select on `raspberry_init.launch`), you can also control robot using arrows on display.

## Frequent questions

This section explain some common problems and how to solve it!

### Installation and compilation problems

- If you want to change default GCC compiler to version 5, open a terminal and run:

```
export CC=/usr/bin/gcc-5
export CXX=/usr/bin/g++-5
```

then you can compile using GCC 5.

### ROS related problems

- When I relaunch Raspberry PI2 nodes, sad face is showed by LED matrix.

Answer: sometimes, when you relaunch nodes, an instance of PIGPIO remain active and generate conflicts, you can stop nodes and wait for couple of minutes before everything is closed by ubuntu, and then start again ros nodes.

## Interface problems

- When I close a node that uses gazebo, it takes too long before it closes.

Answer: You can close manually the process, run in a command window:

```
sudo killall gzserver gzclient
```

- How can I visualize the GPIO state of my raspberry PI 2 remotely?

Answer: You can use pigpio scope [<http://abyz.co.uk/rpi/pigpio/piscope.html>], install piscope in your PC and then run the following command in a terminal.

```
export PIGPIO_ADDR=IP_OF_RASPBERRY
piscope
```

- I get error: initInitialise: bind to port 8888 failed (Address already in use)

Answer: This is caused by PIGPIO's daemon, you can stop all ros process, and kill its process using

```
sudo killall -9 pigpiod
sudo rm /var/run/pigpio.pid
```

then wait some seconds before it is closed by ubuntu. After that, you should be able to execute normally.

## Other questions

- How can I monitor the performance of my Raspberry PI 2 or ODROID:

Answer: you can monitor remotely in real-time the RAM and CPU consumption using SSH, open two different terminal windows and run:

```
watch -n 5 free -m
htop
```

- How can I change router configuration?

Answer: You can connect to router, and open a web browser. Type **<http://tplinklogin.net>**, credentials are: user **admin** and password **admin**.

You can, for instance, change static IP address assigned to RPI2, ODROID or PC (*DHCP/Address Reservation*). As well as checking if all devices are properly connected (*DHCP/DHCP Client List*)

## Contact us

- If you want more info about the project or you want to contribute, we are happy to contact you!

Francisco J. Garcia R. - [garcia@rhrk.uni-kl.de](mailto:garcia@rhrk.uni-kl.de)

13. Sc. Alen Turnwald - [turnwald@eit.uni-kl.de](mailto:turnwald@eit.uni-kl.de)

- Suricate project license:

Copyright (C) 2016 Francisco Garcia

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.

This project is released under GPLv2 license, please consider that external plugins may have a different type of license.

Developed by:

- Francisco J. Garcia R. - [garcia@rhrk.uni-kl.de](mailto:garcia@rhrk.uni-kl.de)
- Dina Martynova

Supervised by:

- 13. Sc. Alen Turnwald - [turnwald@eit.uni-kl.de](mailto:turnwald@eit.uni-kl.de)