RoboTiCan

Innovative Robotics Solutions

Lizi Robot Manual

Manual version: 1.2

Email: support@RoboTiCan.net Website: http://www.RoboTiCan.net

Table of Contents

Safety instructions 2
1. Robot Specification 3
2. Included Hardware 3
3. Getting started 4
3.1. Turning on and using the on-board computer to run the robot 4
4. Maintenance 4
4.1. Batteries 4
5. The Lizi Robot ROS manual 5
5.1. packages 5
5.1.1. lizi package 5
5.1.2 lizi_urdf package 7
5.1.3 lizi_arduino package 7
5.1.3 lizi_base_station package 7
5.2 Running the lizi launch file (from the lizi robot computer) 8
5.3 IMU Calibration 8
5.4 Programming the lizi micro-controller (Arduino DUE) 9
5.5 Clean installation guide 9
6. Warranty 11

Safety instructions

Safety in operation

Distance: Keep a safety distance of few meters between the robot and people when in operation. If the robot may hit a person, it may cause an injury.

Battery safety

The robot's battery are Sealed Lead Acid (SLA) battery. The safety instructions for the battery and charger are as follows:

(1) Do not load the battery in airtight case. This may cause an explosion or injury.
(2) Charge the battery using the exclusive charger. Charging the battery with other chargers may cause the battery to overheat, emit hydrogen gas, leak, ignite, or burst.
(3) Do not put the battery near a device that may cause sparks. The battery may generate flammable gas when charged, so keep the battery away from fire or an open flame to prevent any sparks from igniting or causing explosions.
(4) The operation temperature is -15^{O}C to 40 ^{O}C. Avoid placing the battery near a heat source. This may cause the battery to overheat, leak, ignite, or burst.
(5) Do not immerse the battery in water.
(6) Be careful not to drop the battery onto feet to avoid injury.

1. Robot Specification

The Lizi- Mobile Robotic Platform is designed for indoor and good-weather outdoor operation. The Lizi robot use skid-steering for maximum maneuverability. The robot is designed to be small and light to be able to operate and navigate indoor and outdoor. The Lizi platform is rugged, light weight, and compact. The robot is equipped with a whole suite of sensors for autonomous navigation. The robot is fully compatible with the ROS (Robot operating system) and can be controlled using ready-made ROS packages. Following are the robot's detailed specifications:

- Size: 38 L X 31 W X 20 H cm
- 4 solid rubber wheels, differential drive.
- Wheels diameter: 14 cm
- 2x5W motors (Lizi2), 4x50W motors (Lizi4).
- High resolution encoders (56000(Lizi2)\4288(Lizi4) counts per revolution) for closed loop control.
- Weight: 9.1 Kg (with battery).
- Battery: 12V SLA 9Ah (Charging socket available on the robot, Charger included).
- Wireless 802.11N connection.
- Sensors: GPS, 9 DOF IMU (3 Gyros, 3 Accelerometers, 3 Magnetometers), laser scanner, RGB-D (Kinect like) mounted on pan-tilt system, 3 Ultrasonic Range Finders (URF) mounted on the sides and back of the robot.
- Fully compatible with the ROS (Robot operating system).
- Maximum payload: 2 kg.
- Maximum speed: 1 Km/hr (Lizi2), 3.6 Km/hr (Lizi4).
- MTBC: 2 Hour.

2. Included Hardware

The included hardware in the box is:
1. Lizi Robot.
2. 12V battery charger.
3. Charging adapter.

3. Getting started

In this section we describe the required steps in order to operate the Lizi robot using its onboard computer.

3.1. Turning on and using the on-board computer to run the robot

To turn the robot on do the following:

- On the back of the robot turn the power switch to ON position. The blue LED should be lit.
- Turn the computer ON (this is the button just under the power switch), a short push is enough.

Now you can control the robot from its on-board computer. If you want to work directly on the robot you can connect a keyboard mouse and monitor to the back connectors panel of the robot. Alternatively, you can use remote desktop to connect to the robot and log into its computer.
Another, more common option, is to run the ROS core on a master machine, while one or more robots are working on the same network. For more instructions on how to configure ROS to run on multiple machines see: http://wiki.ros.org/ROS/Tutorials/MultipleMachines
While operating the robot one may connect the charger and work directly from the electric outlet, and charging at the same time.

4. Maintenance

4.1. Batteries

The battery of the robot can be charged while connected to the robot, and even when the robot is in ON mode. The battery is located at the back part of the robot. To replace the battery follow these steps:

1. Open the cover by unscrewing the four screws that hold the battery cover in place. Open the screws all the way, and remove them away.
2. Pull the battery off the robot and then disconnect the two cable connectors from the battery
3. Connect the connectors to the new battery with the correct polarity red (+12V) connector to the red battery terminal and blue (GROUND) connector to black battery terminal.
4. Place the new battery inside the robot, close the cover and screw the four screws back in place.

Charging the battery outside the robot is also possible using the charging adapter.

WARNING:
Incorrect connections or connections with reverse polarity of the battery will damage the robot internal components.

5. The Lizi Robot ROS manual

This section describes the structure of the lizi_robot metapackage and details on the role and configurations of each part.

5.1. packages

5.1.1. lizi package

The lizi package includes the following components and files:

- **config** – This directory includes the default.yaml file which is the default configuration file of the robot. This file includes some robot parameters. One should leave them on their default values. The two last ones are as follows

  `fuse_imu_roll_pitch: true`

  This parameter when true use the IMU's roll and pitch measurements. If false it assumes the robot is on a horizontal surface.

  `fuse_imu_yaw: false`

  This parameter when true use the IMU's yaw measurement instead of the calculated odometry heading. If false uses the robot odometry to determine its heading.

- **launch** - This directory contains the main launch file (`lizi.launch`). See Section 5.2 for more details.
- Nodes
  - `lizi_node` – This is the main node that runs the robot and sensors.
  - `serial_node` – This is an internal node that is responsible for the communication between the internal Micro Controller and the computer. This node is used from http://wiki.ros.org/rosserial
- Topics

  (the prefix "lizi_1" can be different according to the lizi id)
  - Subscribed

    /lizi_1/cmd_vel (`geometry_msgs/Twist`) –

    The robot listens to this topic and set its forward and angular velocity accordingly.

    /lizi_1/command (lizi/lizi_command) –

    Instead of using the cmd_vel topic, one may want to directly command the robot motors. This is done using the /lizi_1/command. The lizi_command message contains two fields: int32 left_wheel, int32 right_wheel . The values are in encoder ticks per second units.

    /lizi_1/pan_tilt (lizi/lizi_pan_tilt) –

    This topic sends command to the pan tilt system that orients the RGB-D sensor. The lizi_pan_tilt message contains two fields: float32 pan_angle

float32 tilt_angle. The values are in radians units. Note that the positive directions of the pan-tilt are left and up respectively. A single message is enough to set the required angles (no need for publishing these messages continuously).
The pan angle range of motion is ±35°, and the tilt range is ±30°. These limitations are constrained by software.

- Published

/lizi_1/Rangers/Left_URF (`sensor_msgs/Range`)
/lizi_1/Rangers/Rear_URF (`sensor_msgs/Range`)
/lizi_1/Rangers/Right_URF (`sensor_msgs/Range`)

These three topics publish the range measured by the Ultrasonic Range Finders (URF) on the Left Rear and Right respectively. The published message is of the standard type `sensor_msgs/Range`.

/lizi_1/lizi_status (lizi/lizi_status)

This topic publishes the battery voltage in Volts and the faults report. The voltage should be in the range of 11V to 14V. The faults parameter is constructed from 3 bits as follows:
Bit 0 – Set if motors driver has faults.
Bit 1 – Set if motors torque is off.
Bit 2 – Set if GPS location is not valid.
Bit 3 – Set if IMU readings are not valid.

faults = (Bit 0)*1 + (Bit 1)*2 + (Bit 2)*4 + (Bit 3)*8

/lizi_1/gps_pub (sensor_msgs/NavSatFix)

This topic publishes the GPS location of the robot. The published message is of the standard type sensor_msgs/NavSatFix.

/lizi_1/imu_pub (sensor_msgs/Imu)

This topic publishes the orientation of the robot as measured by the internal IMU (Inertial Measurement Unit). The published message is of the standard type sensor_msgs/Imu.

/lizi_1/odom_pub (`nav_msgs/Odometry`)

This topic publishes the location and orientation of the robot based on the odometry computation of the wheels rotations. The published message is of the standard type `nav_msgs/Odometry`.

Internal topics

The following are internal topics which are used for the communication between the internal micro controller (and sensors) to the main computer.

/lizi_1/raw_gps (lizi/lizi_gps)
/lizi_1/lizi_raw (lizi/lizi_raw)

- Services

  /lizi_1/set_odom (lizi/set_odom)

  This service set the odometry of the robot to any desired location and orientation it should receive three fields: float32 x, float32 y, float32 theta.

  /lizi_1/reset_encoders (std_srv/Empty)

  This service reset the encoder readings on the micro-controller counters.

  /lizi_1/set_parameters (lizi/set_parameters)

  This service set the control parameters for the closed-loop PID control of the robot wheels speed. The default parameters are:

  kp=0.001
  ki=0.01
  kd=0
  alpha=0.5 (for velocity low-pass filter)
  control_dt=1 (control loop interval in milliseconds)

  Calling this service will only change the current PID constants. To make the change permanently you will have to change the constants in the LiziDUE2.ino \ LiziDUE4.ino file and re-upload the code (see section 5.4).

  /lizi_1/imu_calib (lizi/imu_calib)

  This service allows calibrating the IMU accelerometers and magnetometer. See section 5.3 for more details.

5.1.2 lizi_urdf package

This package contains two subdirectories which describe the robot both kinematics and visual aspects. The "robots" directory includes the Unified Robot Description Format (URDF) file of the Lizi robot. The directory "meshes" includes the visual meshes of the robot's links.

5.1.3 lizi_arduino package

This package contains the code and libraries for the micro-controller (Arduino Due: http://arduino.cc/en/Main/arduinoBoardDue).

5.1.3 lizi_base_station package

TODO

5.2 Running the lizi launch file (from the lizi robot computer)

Users can run the main launch file: ***lizi.launch*** to initiate all the sensors and controllers with a single command:

>roslaunch lizi lizi.launch id:=1

The lizi id can be set from the roslaunch command or by changing the default value in the launch file.

One can add or remove components from this file to enable or disable some features. This launch file does the following:

- Set the lizi ID parameter.
- Reads the robot parameters from the `default.yaml` file.
- Load the robot description from the URDF file.
- Run the `static_transform_publisher` node from the `tf` package to publish the `odom` transformation. This should be activated in case there is no active mapping process running (optional).
- Run the `robot_state_publisher` node from the `robot_state_publisher` package to publish the transformations of all other links of the robot (optional).
- Run the `serial_node` node from the `rosserial_python` package to allow the communication between the internal micro-controller and the main computer.
- Run the `lizi_node` node from the `lizi` package to run the robot.
- Runs the `openni2.launch` to start the Asus RGB-D sensor (optional).
- Run the `usb_cam_node` node from the `usb_cam` package to start the front camera (optional).
- Run the hokuyo laser scanner node (optional).
- One can start the `rviz` or `rqt_gui` visualization software by uncommenting the last command of the launch file (optional).

5.3 IMU Calibration

To calibrate the IMU sensor: lizi/imu_calib service is used to enter calibration mode and saving the data. The com field in the service request can range from 1 to 4 as follows:

com=1 – Enter magnetometer calibration mode.
com=2 – Enter accelerometer calibration mode.
com=3 – Save current calibration data.
com=4 – Exit calibration mode.

For example to calibrate the magnetometer of Lizi 1 robot use:

> rosservice call /lizi_1/imu_calib "com: 1"

After entering calibration mode, move the robot around and then send com=3 request to save the data, followed com=4 request, as follows:

> rosservice call /lizi_1/imu_calib "com: 3"
> rosservice call /lizi_1/imu_calib "com: 4"

If something went wrong, just send com=4 request to discard the data.

The calibration data is stored in flash and is overwritten every time a new sketch is uploaded.

5.4 Programming the lizi micro-controller (Arduino DUE)

In order to upload new code to the micro-controller, follow these steps:

1. Install Java:
   >sudo apt-get install gcc-avr avr-libc
   >sudo apt-get install openjdk-6-jre
   >sudo update-alternatives --config java
2. Install Arduino 1.5.6 or newer.
3. In your Arduino installation folder go to hardware/arduino/sam/variants/arduino_due_x/ and edit the variant.h:
   Look for the line:
   #define PWM_FREQUENCY 1000
   And change it to:
   #define PWM_FREQUENCY 20000
4. Copy the content of the lizi/Arduino folder into you arduino sketchbook location.
5. Start the Arduino IDE and load the LiziDUE2/ LiziDUE4 sketch.
6. From Tools->Board select "Arduino Due (Programming Port)".
7. From Tools->Port select "/dev/ttyACM0 (Arduino Due (Programming Port))".
8. Edit the code as you like.
9. Click Upload.

For future id changes only steps 5-9 are required.

5.5 Clean installation guide

This section explains how to install all required software for the Lizi Robot on a new or formatted hard drive.

1. Install Ubuntu 12.04.01 LTS (32bit) from bootable USB drive.
* do not update graphic card drivers
The default lizi robot user name and password are:
Username: lizi
Password: a

2. Install ROS Hydro (Desktop-Full Install), see:
http://wiki.ros.org/hydro/Installation/Ubuntu

3. Configure the ROS environment, see:
http://wiki.ros.org/ROS/Tutorials/InstallingandConfiguringROSEnvironment

Don't forget: in ~/.bashrc replace: "source /opt/ros/hydro/setup.bash" with: "source ~/catkin_ws/devel/setup.bash" (without "")

4. Install the "`lizi`" package (change 'catkin_ws' to your ros workspace name) :

```
>cd ~/catkin_ws/src
>git clone https://github.com/robotican/lizi.git
>cd ..
>catkin_make
>roscd lizi/config/
>sudo ./setup.sh
```

The last command will also install rosserial and offer to install openni2 and usb-cam packages.

Finally, don't forget to run catkin_make

```
> cd ~/catkin_ws
>catkin_make
```

5. Optional: install the Multi Robot Tele-Operator, see:
http://wiki.ros.org/mr_teleoperator/Tutorials/Getting%20Started
run this if the plugin does not appear in the rqt gui:

```
>rm ~/.config/ros.org/rqt_gui.ini
```

6. Optional: install "Terminator" from Ubuntu Software Center.

7. Reboot.

6. Warranty

Limited Warranty Coverage:

Subject to the limitations provided below, the Lizi Robot is warranted against defects in materials and workmanship, under normal use, from the date of shipment through the period identified in the purchase quote (one year). If warranty period is not specified then the deliverable is "as is" and no warranty is provided.

Where a warranty is provided, RoboTiCan liability for such warranty is limited to: 100% of the parts necessary to repair the covered defect. Technical support by telephone, e-mail, fax or other means of correspondence during the warranty period for issues covered by warranty is provided at no charge for covered defects.

If travel to a location (other than RoboTiCan facility) is required to address an issue determined to be covered by warranty, parts will be covered but all labor and travel expenses will be billed. Travel will be billed at expenses plus $50.00 per hour for work time and $20.00 per hour travel. Determination of warranty coverage is the responsibility of RoboTiCan and cannot be assigned or delegated to any other party.

The following are specifically not covered by warranty: (1) failure due to abuse and neglect and/or improper operating environment (including, but not limited to, improper power supply, temperature, humidity, and environmental conditions); (2) down time and related costs due to failure; (3) items such as batteries, and (4) software.

RoboTiCan shall have sole authority to determine type and means of repair in the event of a warranty claim. RoboTiCan shall have the right to require the return of the defective part or system to RoboTiCan, transportation prepaid, to establish the claim. RoboTiCan shall in no event be held liable for repairs or alterations made without RoboTiCan’s written consent or approval. RoboTiCan shall not be held responsible for repairs made by others. If unauthorized service is performed, the warranty provided herein shall be void. This warranty is void if the equipment is altered, improperly operated, improperly maintained or payments are not made according to the agreement.

Any additional software requirements or options are the responsibility of the end user. RoboTiCan does not warrant or guarantee the performance or functionality of any software, or that Customer is acquiring the legal right to use any software embedded within, or provided in connection with, reconditioned robots. Any software provided to customer is provided "as is" without warranty of any kind, either express or implied, and RoboTiCan specifically disclaims any warranty of merchantability, fitness for a particular purpose, satisfactory quality, lack of viruses, title, quiet enjoyment, and/or non-infringement with respect to any software. Customer acknowledges and agrees that use of any software provided to customer is at customer's own risk. Customer agrees to indemnify and defend RoboTiCan from and against any liability arising from customer's use of software in connection with products sold to customer hereunder.

Except for the warranties expressly provided herein, all other guarantees, warranties, conditions or representations, either expressed or implied, whether arising under statute, common law, and commercial usage or otherwise, including implied warranties of merchantability and fitness for a particular purpose, are excluded and hereby specifically disclaimed. In addition to writing here RoboTiCan company will not be responsible for any

damage, injury, loss of ability to work, disability and even death caused arising directly or indirectly from the use of the Lizi Robot. RoboTiCan has made no express warranties except as provided herein, and no oral or written information or advice given by RoboTiCan its agents or employees shall create a warranty of any kind or in any way increase the scope of the limited warranty provided herein.

Under no circumstances whatsoever shall RoboTiCan be liable to any person, firm, corporation or other entity for any special, punitive, indirect, or consequential damages, whether for breach of contract, negligence, misrepresentation, or otherwise, and whether resulting in lost profits, loss of interest in money borrowed or invested, impairment of goods, business interruption, work stoppage, or otherwise, in any way arising out of any transaction to which these standard terms apply. RoboTiCan maximum liability for any claim arising hereunder or relating to the products sold to customer hereunder (whether for breach of warranty or otherwise) shall be limited to the obligation to repair the non-conforming product.

Customer agrees that this sale is completely made in the Country of Israel in that the laws of the Israel country shall apply concerning any controversy or claim regarding the equipment. Customer agrees to indemnify and hold harmless RoboTiCan from all claims, damages, liabilities, attorney's fees, and expenses arising from the ownership or use of the equipment.