Kobuki User Guide

Contents

About

Introducing Korea’s first robotic turtle.

kobuki [거북이] n. turtle

Kobuki is robotically engineered to be long-lived, tough and fast. With high performance batteries, Kobuki will tirelessly work alongside you through those long coffee-powered nights. He’ll also happily burden himself with your modded array of sensors, actuators, laptops, embedded boards, portside cannons and do it all at a speed that makes his real world cousins seem like … well, turtles.

Use him for serving 치맥 (chi-mek), chasing your neighbour’s kids or simply, to make your own robot ideas become reality.

Kobuki is still young, don’t expect him to remain as he is . Kobuki’s development has already been significantly influenced by the community and as he marches towards old age, we will continue to work with the community and you to ensure he becomes better with time.

Sincerely,
Kobuki Team.

Safety Guideline

Precautions related to power

- Do not force to bend the power cord or pressure it with heavy weight.
- Keep the pin and interface of the power plug clean from dust or water.
- Do not pull the power cord or touch the power plug with wet hands.
- Do not use damaged power plug or power cord, loose outlet.
- In case the water went inside the product, you must contact us for check-up even the product was dried thoroughly.
- Do not put or pull the power plug consecutively.
- Do not touch the power cord of the charger with wet hands.

Kobuki use restrictions

- Kobuki is for indoor use only.
- Do not pour or spray water onto Kobuki.
- Do not use Kobuki to pick up anything that is burning or smoking.
- Always remove the battery before long-term storage or transportation.
- Do not sit or stand on this device.

Get Started

In order to gain access to all of Kobuki's features, you need to provided an external computing unit, which communicates with Kobuki. This could be a laptop, netbook, tablet, an embedded board or other devices.

To let you quickly start using Kobuki, we added tutorials for each supported environment to the documentation section on the Kobuki website:

- Linux (including ROS)
    - http://kobuki.yujinrobot.com/documentation/get-started/on-linux
- Windows
    - http://kobuki.yujinrobot.com/documentation/get-started/on-windows
- Embedded
    - http://kobuki.yujinrobot.com/documentation/get-started/on-embedded

Batteries

Specification

Kobuki by the default ships with a small Lithium-Ion battery pack (4S1P, 2200mAh, 14.8V). For extra long operation, a big battery pack (4S2P, 4400mAh, 14,8V) can be ordered as well.

More details can be found on Kobuki's website:

- http://kobuki.yujinrobot.com/documentation/hardware/battery-details/

Replacing the battery

- Turn off Kobuki.
- Flip it and open the battery bay in the centre.
- Unplug and remove the battery.
- Plug in the new battery.
- Close the battery bay and flip it again.
- Turn on Kobuki.

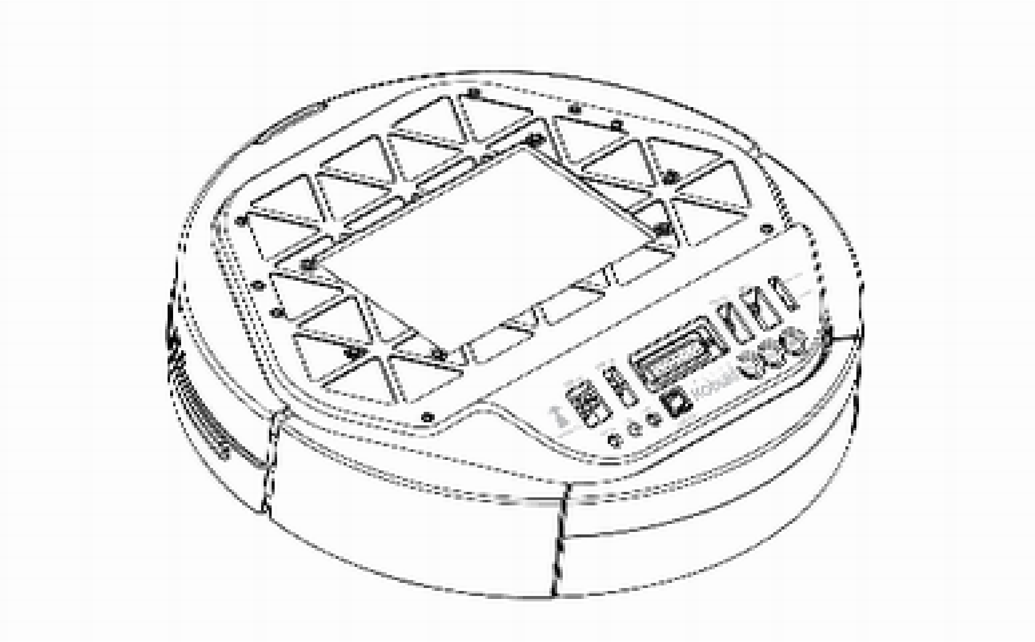
Note: It is not possible to use multiple battery packs at the same time.

A more detailed tutorial including pictures is located on Kobuki's website:

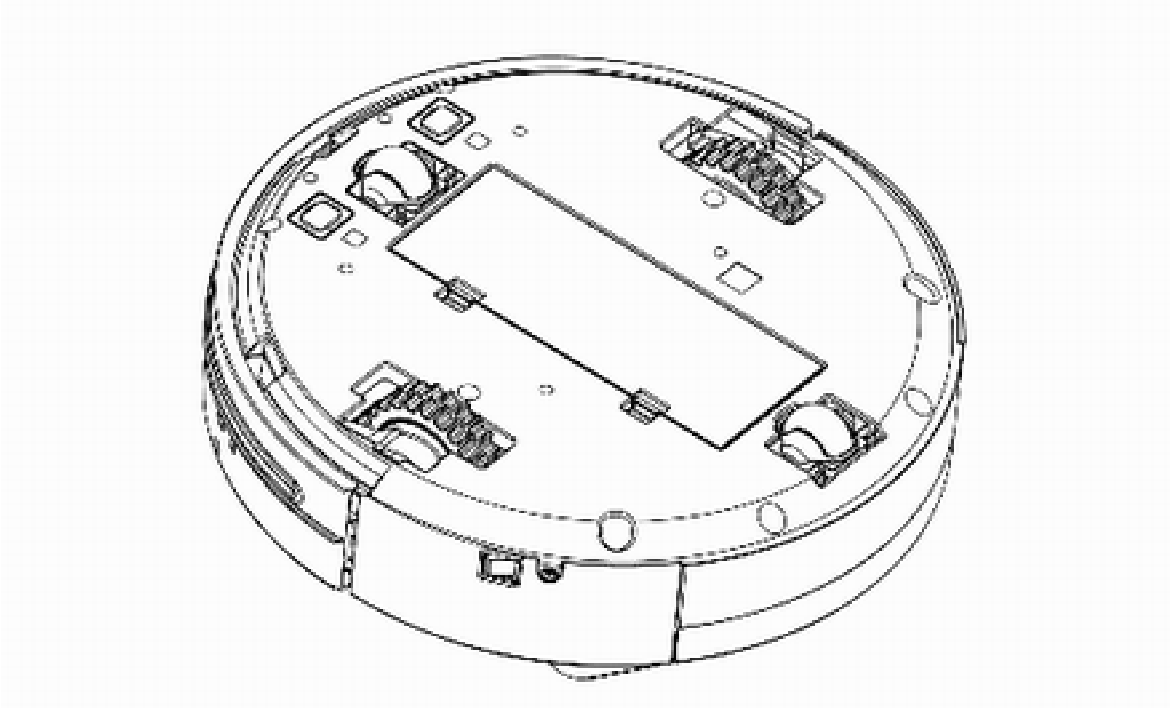
- http://kobuki.yujinrobot.com/documentation/howtos/battery/

Anatomy

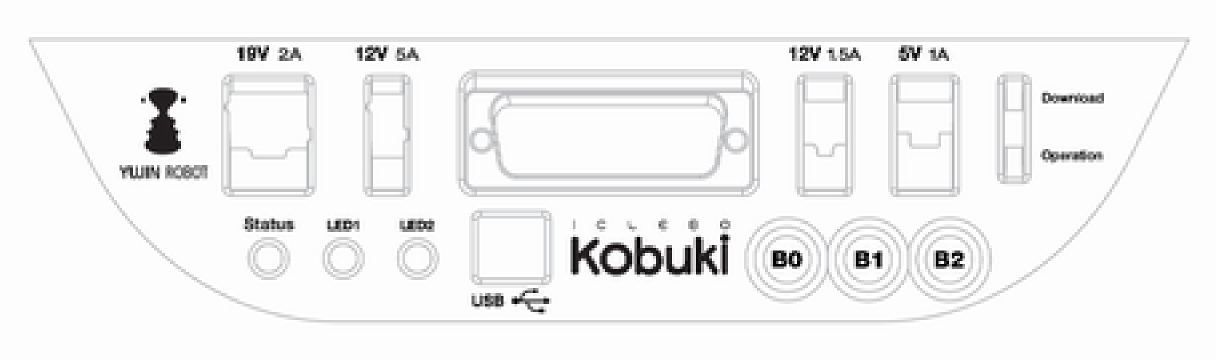
Top View



Bottom View

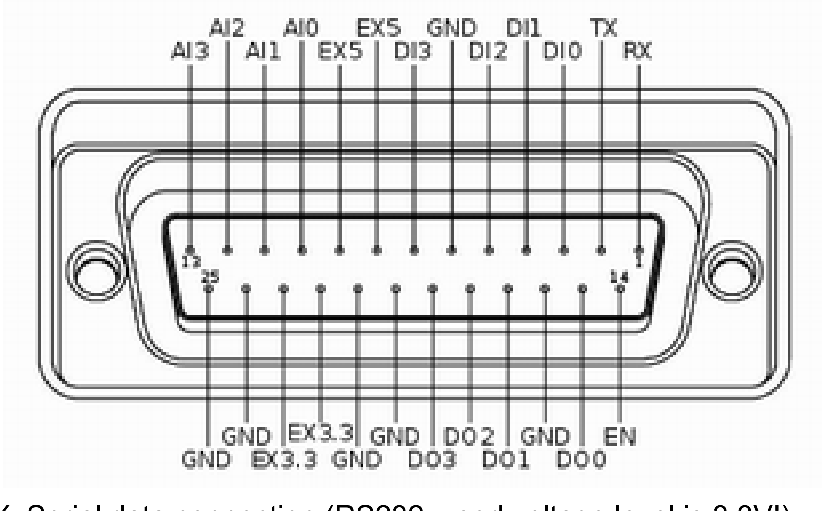


Control Panel



- 19V/2A**:** Laptop power supply
- 12V/5A: Arm power supply
- 12v/1.5A: Microsoft Kinect power supply
- 5V/1A: General power supply
- Status LED: Indicates Kobuki’s status
  - Green: High battery voltage level
  - Orange: Low battery voltage level (please charge soon)
  - Green blinking: Battery charging
- LED1/2: Programmable LEDs
- USB: Data connection
- BO/1/2: Buttons
- Firmware switch: Enable/disables the firmware update mode

Serial Port



- RX / TX: Serial data connection (RS232; used voltage level is 3.3V!)
- EX3.3 / EX5: 3.3V/1A and 5V/1A power supply
- DI0 - 3: 4 x Digital input (high: 3.3 - 5V, low: 0V)
- DO0 - 3: 4 x Digital output (open-drain, pull-up resistor required)
- AI0-3: 4 x Analog input (12bit ADC: 0 - 4095, 0 - 3.3V)
- GND: Ground
- EN: Used for detecting an external board (connect to external ground)

Specifications

Functional

- Maximum translational velocity: 70 cm/s
- Maximum rotational velocity: 180 deg/s (>110 deg/s gyro performance will degrade)
- Payload: 5 kg (hard floor), 4 kg (carpet)
- Cliff: will not drive off a cliff with a depth greater than 5cm
- Threshold Climbing: climbs thresholds of 12 mm or lower
- Rug Climbing: climbs rugs of 12 mm or lower
- Expected Operating Time: 3/7 hours (small/large battery)
- Expected Charging Time: 1.5/2.6 hours (small/large battery)
- Docking: within a 2mx5m area in front of the docking station

Hardware

- PC Connection: USB or via RX/TX pins on the parallel port
- Motor Overload Detection: disables power on detecting high current (>3A)
- Odometry: 52 ticks/enc rev, 2578.33 ticks/wheel rev, 11.7 ticks/mm
- Gyro: factory calibrated, 1 axis (110 deg/s)
- Bumpers: left, center, right
- Cliff sensors: left, center, right
- Wheel drop sensor: left, right
- Power connectors: 5V/1A, 12V/1.5A, 12V/5A
- Expansion pins: 3.3V/1A, 5V/1A, 4 x analog in, 4 x digital in, 4 x digital out
- Audio : several programmable beep sequences
- Programmable LED: 2 x two-coloured LED
- State LED: 1 x two coloured LED [Green - high, Orange - low, Green & Blinking - charging]
- Buttons: 3 x touch buttons
- Battery: Lithium-Ion, 14.8V, 2200 mAh (4S1P - small), 4400 mAh (4S2P - large)
- Firmware upgradeable: via usb
- Sensor Data Rate: 50Hz
- Recharging Adapter: Input: 100-240V AC, 50/60Hz, 1.5A max; Output: 19V DC, 3.16A
- Netbook recharging connector (only enabled when robot is recharging): 19V/2.1A DC
- Docking IR Receiver: left, centre, right

Software

- C++ drivers for Linux and Windows
- ROS driver
- Gazebo simulation

Firmware

Communication

- Baudrate: 115200 BPS
- Electronic protocol: USB / RS232 (@ DB25 connector)

Bytestream Protocol

The driver communicates with the robot by using predefined protocol. In general, the driver sends the commands to the robot and the robot sends some feedback data or sensor readings. This command and feedback data is converted into bytestream for communication via serial port. Protocol specify that rules and forms of bytestream.

Structure of bytestream:

Header 0	Header 1	Length	Payload	Checksum
1 byte	1 byte	1 byte	n bytes	1 byte
0xAA (fixed)	0x55 (fixed)	Size of payload in byte	Described below	Xor'ed value of every bytes of command

Groups:

Header	Length	Data
1 byte	1 byte	n bytes
Predefined identifier	Size of data in byte	

Command Identifier:

1	Base Control
2	*Reserved*
3	Sound
4	SoundSequence
5	*Reserved*
6	*Reserved*
7	*Reserved*

8	SetPower
9	ReqestExtra
10	ChangeFrame
11	Request EEPROM
12	General Purpose Output

Feedback Data Identifier:

1	Basic Sensor Data
2	*Reserved*
3	Docking IR
4	Inertial Sensor Data
5	Cliff sensors
6	Current
7	*Reserved*
8	*Reserved*
9	*Reserved*
10	Hardware Version
11	Firmware Version
12	*Reserved*
13	Raw data of 3-axis gyro
14	*Reserved*
15	EEPROM
16	General Purpose Input
17	*Reserved*
18	*Reserved*
19	Unique Device Identifier
20	*Reserved*

Version Checking

Every time the robot comes alive (because it is connected or powered on), the driver checks the compatibility between software (i.e. driver) and firmware. Firmware versions are of the form M.m.p, where:

- M (ayor) represents a deep rebuild of the code that almost surely breaks protocol compatibility. In consequence, if mayor version doesn't match, the driver will show an error, suggest the required update and shutdown.
- m (inor) represents a new feature that could not work if the driver is outdated, but the protocol itself is spared. In consequence, if minor version doesn't match, the driver will show a warning suggesting the required update and continue working.
- p (atch) represents a fix on the code and is not checked at all.

Generally speaking, it's recommended to upgrade to the latest stable version. Next section explains how to do it.

Updating

The firmware update process is a bit different depending on which operating system used for flashing. The documentation section on the Kobuki websites contains detailed instructions on how to update the firmware using Linux and Windows:

- Linux tutorial
  - http://kobuki.yujinrobot.com/documentation/howtos/upgrading-firmware/upgrading-firmware-linux/
- Windows tutorial
  - http://kobuki.yujinrobot.com/documentation/howtos/upgrading-firmware/upgrading-firmware-windows/

Troubleshooting

Will be filled as problems appear.

Links

Website	http://kobuki.yujinrobot.com
Blog	http://blog.yujinrobot.com
Mailing List	kobuki-users@yujinrobot.com
Documentation	http://kobuki.yujinrobot.com/documentation
Contact	kobuki@yujinrobot.com

Appendix

Accessories

- Small (4S1P) and big (4S2P) battery packs
- Docking Station
- Kinect Modification Kit
- Netbook Modification Kit
- Turtlebot Accessories

Reference Platforms

- Turtlebot 2