RoBat
DIY Robotics Kit

v0.4.2, 9/10/2021

Table of Contents

The RoBat robotics kit

Hi there!

It‘s nice to have you here and that you are interested in robots! The RoBat robotics kit is a DIY kit with which you can build your own robot. It's not hard at all and you can design your robot following your own ideas. These instructions explain how to use the kit.

With this tutorial we build a moving robot together. You can either control it with a small joystick or let it find its own way.

Unlike many other robot kits, RoBat does not have a predefined chassis. On the one hand this would make the construction easier, but on the other hand it would limit the possibilities. With a piece of sturdy cardboard and a hot glue gun you can easily build a chassis yourself. There is also a template that you can use, but you don't have to.

"Building a robot" might sound pretty complicated to you at first. But with the RoBat kit, it's not.

Let's have fun!

What is a robot anyway?

A robot is originally a machine that takes work from humans, especially with a human-like shape. The word comes from the Czech language and means "compulsory service" or "forced labor". During the years, the meaning has changed to some extent and at the moment there are many "robots" that are actually vehicles and move more or less independently. In industry, a robot is still a working machine that can perform tasks similar to a human being, often much more accurately.

To do work or move somewhere purposefully, you have to...

1. ... be able to perceive your surroundings
2. ... being able to influence / move the environment or yourself
3. ... know and decide what to do

Sensors of all kinds are used for perception, for example for temperature, brightness, humidity, distance or acceleration.

The environment or the robot itself can be influenced with **actuators.**

These can be motors or servos (movement), but also heating elements and coolers (temperature) or lamps (brightness).

Finally, we need a **controller** that executes actions in certain perceived situations. In the RoBat, this is done by a microcontroller, a kind of mini-computer that processes the values measured by the sensors and controls the actuators to perform actions.

In the expert instructions you will find hints on how to build other robots with the RoBat kit.

Check your kit

The first thing you should do is check that your RoBat kit is complete. The following parts are included (see illustrations on the next page):

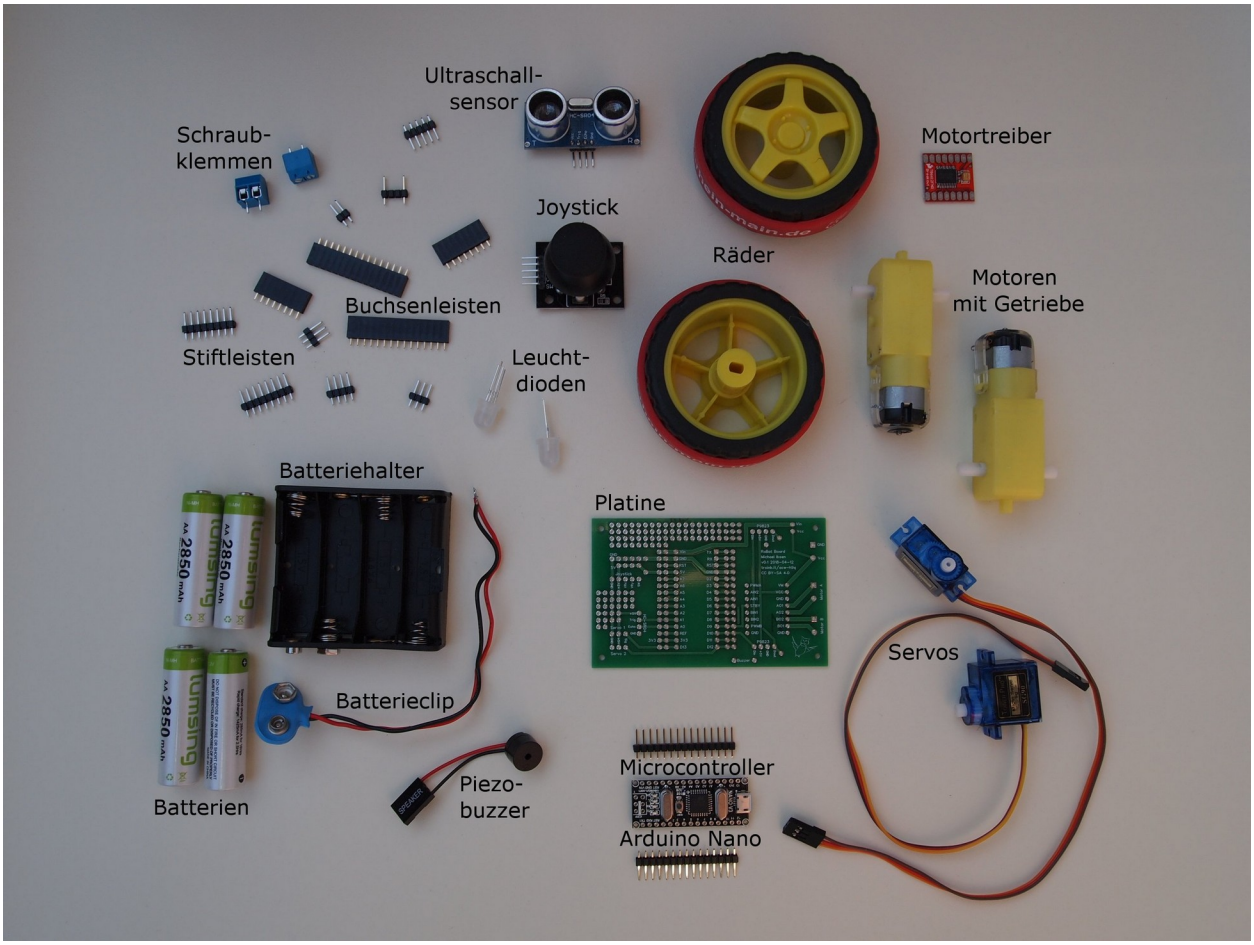
Quantity	**Component**	**available?**
1	Cuircuit board	
3	Screw terminals	
4 or 5	AA batteries	
1	Battery holder	
1	Arduino Nano Microcontroller	
1	Motor driver TB6612FNG	
2 or 4	yellow geared motors	
2 or 4	Wheels	
0 or 1	Furniture caster wheel (not shown)	
1	Ultrasonic sensor HC SR-04	
1	Piezo buzzer	
2	RGB Light Emitting Diodes (LEDs) PL9823	
1	Mini joystick	
1m	5-wire joystick cable (not shown)	
20cm	4-pole Dupont cable (not shown)	
2 or 4	two-wire cable for the motors	
1	Servo	

There is a variant of the kit with four motors and wheels, which does not include a caster wheel(aka as furniture roll).

If all parts are there, we can start. If any parts are missing, please let us know so that we can supply them (free of charge). You can find our address at the very end of these instructions.

Components

In the following section, the individual components of RoBat are explained and RoBat is assembled in the process.

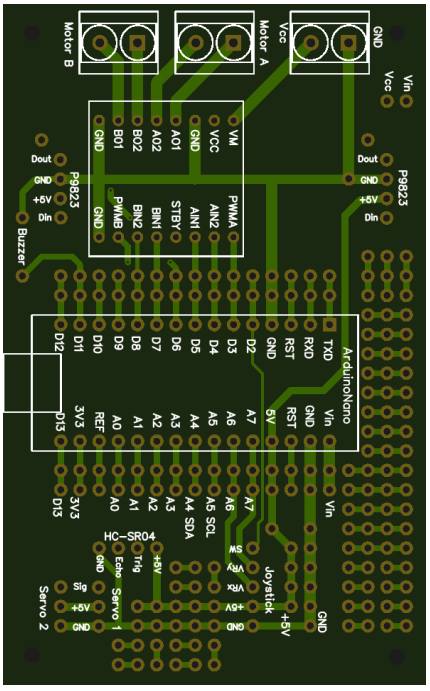


Important: With all components, you must make sure that you install them the right way round. If they are installed the wrong way round, not only will the robot not work, but the parts can also get damaged or destroy other parts.

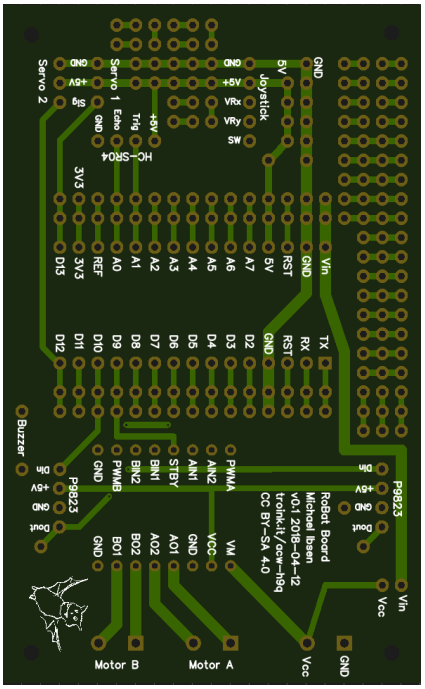
Board

We start with the circuit board, onto which all parts are soldered. The task of the PCB is to connect the contacts of the components with each other. This could also be done with wires, but that would be quite a lot of work and might also look a bit messy.

On the board the places of the components are marked. This is intended for the basic version of RoBat. When you have gained some experience, you can deviate from this and connect other components.



Top side of the board



Bottom side ofthe board

The board has a top and a bottom side. The top **side** is the one with the outline of the components printed on it. There‘s a little bat on the **bottom side**. In the description of the assembly, the board is always placed with the top side up and the components are put on top of it. **Soldering is always done on the bottom side**.

Wiring diagram

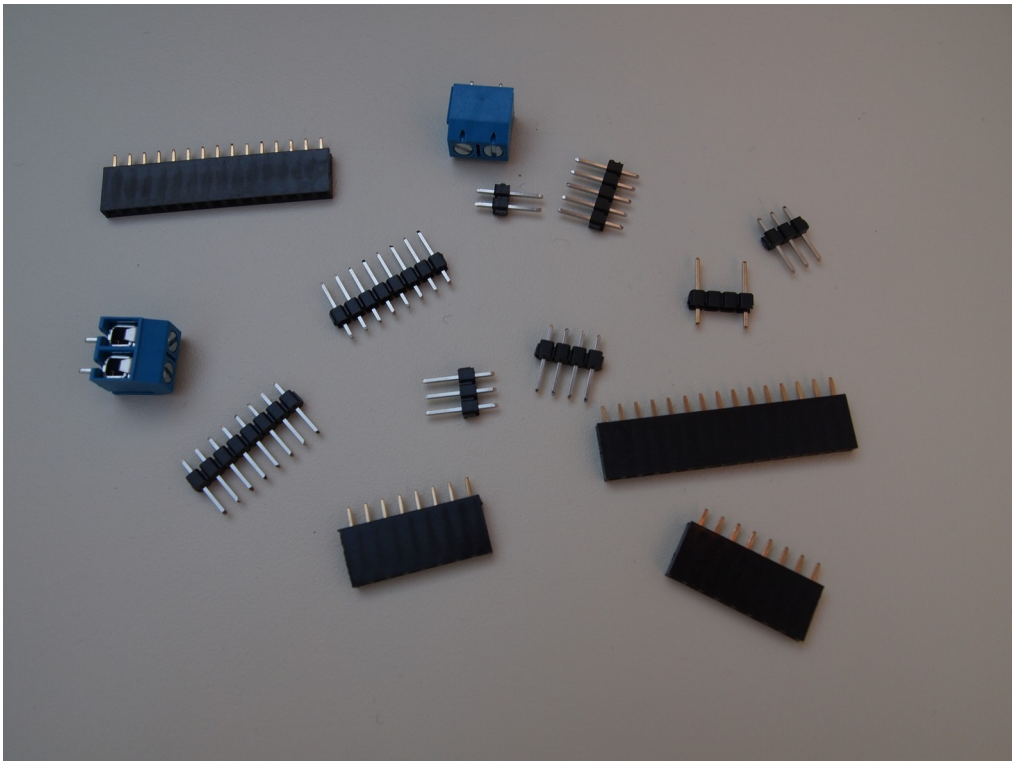
The connection diagram shows which contacts of the components have to be connected with each other. The circuit board does that for you. You only need the diagram if you want to expand or better understand the robot.

Arduino	TB 6612	Joystick	Servo 1	Servo 2	SR04	LED	Buzzer	Motor A	Motor B	Probe 1	Probe 2	Bluetooth
GND	GND	GND	GND	GND	GND	GND	GND			GND	GND	GND
+5V	Vcc	+5V	Vcc	Vcc	Vcc	Vcc						Vcc
3.3V												
Vin	VM											
D0 / RX												TX
D1 / TX												RX
D2		SW										
D3	PWMB											
D4	BIN2											
D5	BIN1											
D6	PWMA											
D7	AIN1											
D8	AIN2											
D9	STBY											
D10						Din						
D11							+					
D12			Signal									
D13				Signal								
A0					Echo							
A1					Trig							
A2										x		
A3											x	
A4												
A5												
A6		VRx										
A7		VRy										
	A01							+				
	A02							-				
	B01								-			
	B02								+			

Connections

Pin headers

The pin headers lead the contacts of the components to the outside.



Socket and pin headers

The RoBat Kit contains two pin headers with 3 pins for the connection of the servos, one pin header with 4 pins for the ultrasonic sensor and one pin header with 5 pins for the joystick. Two pin headers with 15 pins each are included for the microcontroller and two pin headers with 8 pins each are included for the motor driver.

Here again at a glance:

Component	Number	Pins
Servo	2	3
Ultrasound	1	4
Joystick	1	5
TB6612FNG	2	8
Microcontroller	2	15

The individual pin headers are plugged into the board with the **short end.**

Secure each with a small adhesive strip or rubber band so that they cannot fall out when the board is turned over.

Now turn the board over and solder each pin.

A jumper must be soldered between the solder tags labeled "Vin" and "Vcc" on the board near the input voltage terminal. To do this, take a piece of wire and solder one end to each of the two soldering eyes.

Socket connectors

Individual components can be plugged into socket connectors in order to be able to remove them again later. This is especially useful for the microcontroller and the motor driver, e.g. for programming or to replace them. You can also work without the socket connectors and solder the microcontroller and the motor driver directly with the pin headers on the board.

Two socket connectors with four pins for the LEDs should be soldered in any case. They make it possible to remove the LEDs again and to attach them with a cable at another place at the RoBat.

Overview of the socket connectors:

Component	Number	Pins
Arduino (Microcontroller)	2	15
TB6612FNG	2	8
LEDs	2	4

Screw terminals

The screw terminals are used to connect the wires of the motors and the battery box.

Battery box

The batteries are used to power your RoBat. You can use 4 or 5 AA batteries (1.5V), depending on your kit. This results a voltage of 6V (or 7.5V in the case of 5 batteries), which is perfect for the Arduino and the motors. Instead of normal batteries you can also use rechargeable batteries. They only have 1.2V each, for a total of 4.8V (6V for 5 of them). 4.8V is actually a bit too low, but works fine most of the time. Your RoBat will run a bit slower with rechargeable batteries than with batteries. But you can recharge the batteries again and again and don't have to buy new ones all the time. This saves the environment and your wallet.

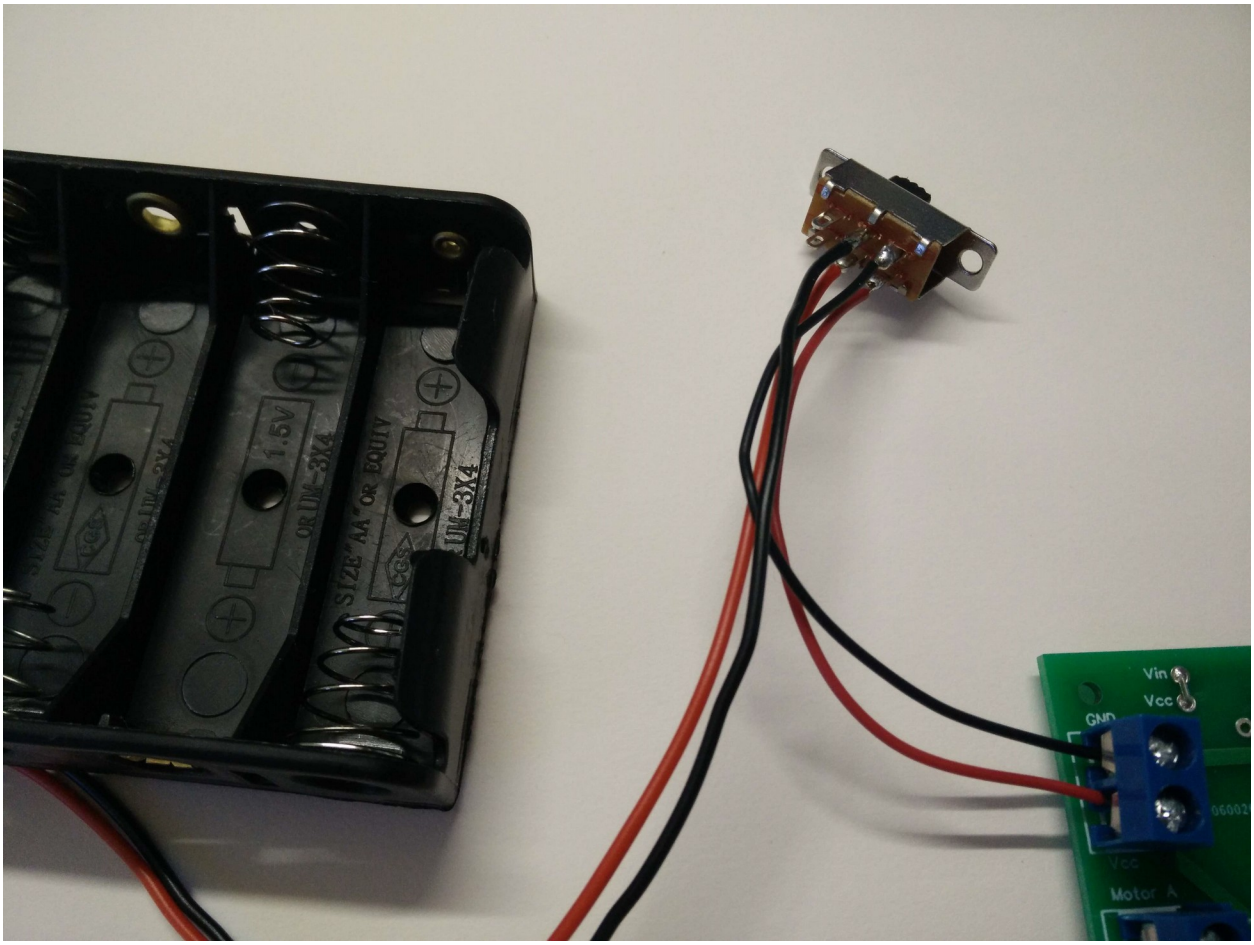
The batteries are inserted into the battery box only **after assembly.**

Battery box

Switch

With the switch you can turn RoBat on and off. Solder the red wire of the battery box to the middle contact on one side of the switch and the black wire to the middle contact on the other side of the switch.

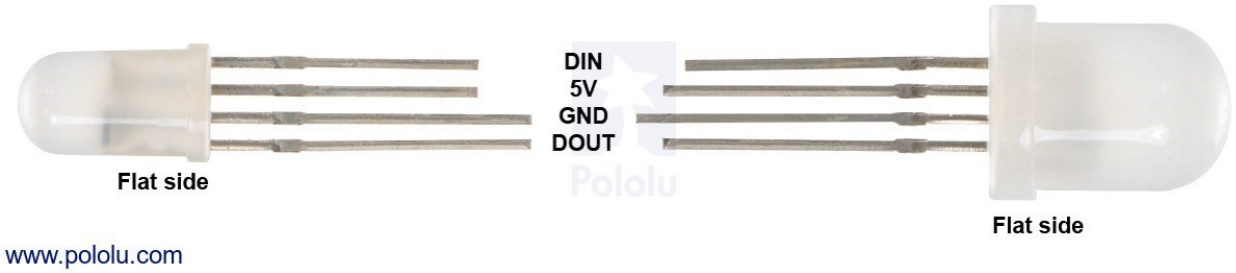
Now we use two of the still unused contacts of the switch, which are next to each other, and solder a red and a black wire there again, as shown in the picture. The red cable goes to the screw terminal marked "Vcc", the black cable to the terminal marked "GND".



Connection of the switch between battery holder and screw terminals

LED

Light emitting diodes or LEDs are components that can emit light, hence the name. RoBat has two P9823 type LEDs. They can display all colors of the rainbow. These special LEDs have their own very small computer built in, which can be programmed via a single wire. This allows the color of the LED to be set. The P9823 can even be strung together. So you can address many LEDs with only one control line.



P9823 Pinout

With the LEDs you have to make sure that they are inserted the right way round. The longest of the four LED connections must go into the soldering eye marked "GND" on the board.

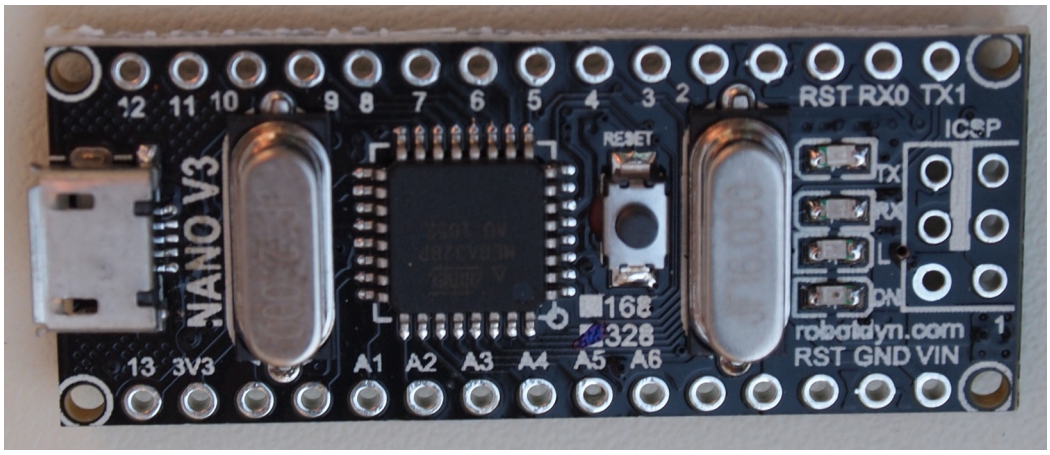
If you don't solder the LEDs directly to the boards, but use one of the supplied female headers (4-pin), you can extend the LEDs later with a 4-wire cable and attach them in other places of the RoBat, e.g. as "eyes" or lamps.

Microcontroller Arduino Nano

The microcontroller is the brain of your robot, so to speak. It processes the measured values of the sensors (e.g. the distance to the next object) and controls the motors, the servo and the LEDs.

The microcontroller included in the kit is called Arduino Nano. Besides the Nano, there are many more Arduino models for many different purposes. We have chosen the Nano because it is small and inexpensive and already offers a lot of functions.

You can find more about the Arduino microcontrollers on their website at https://www.arduino.cc/.



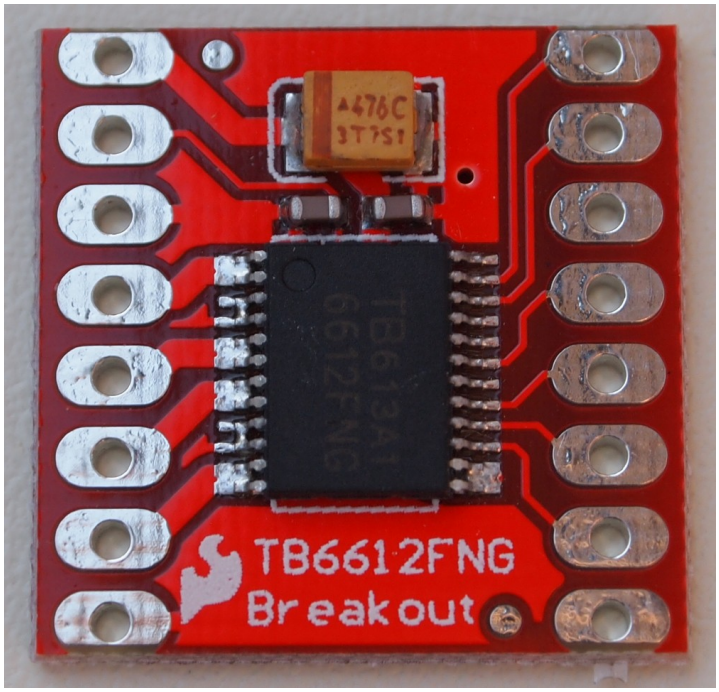
Arduino Nano

With the Arduino it is especially important that it is soldered in the right way round. Orientate yourself by the labeling and make sure that the USB connector of the Arduino is facing outwards. There is a small white box printed on the board at this point that marks the USB port.

The Arduino can perform many different tasks. To do this, it must be programmed. Without programming, your robot does nothing. We will explain this to you right after the assembly.

Motor driver

The outputs of the Arduino microcontroller can only supply a small amount of current, much less than the motors need. Therefore you need a motor driver to control the motors. You can think of it as a current amplifier. The motor driver is instructed by the Arduino via control lines which motor should turn how fast in which direction.

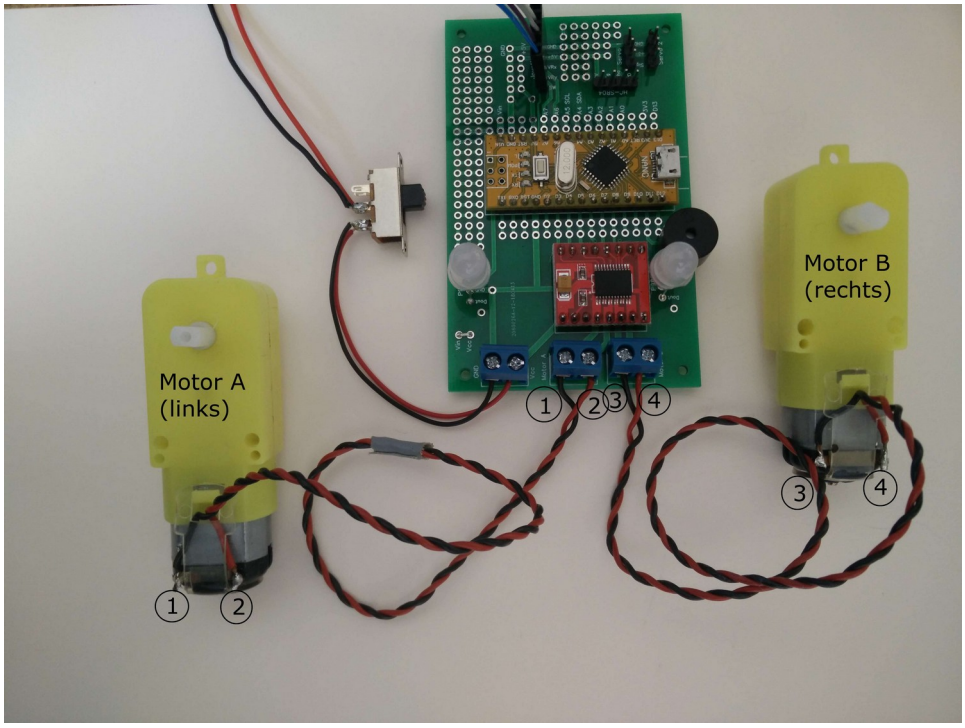


Motor driver TB6612FNG

Engines

The motors are connected to the screw terminals with cables. The cables must be soldered to the motor terminals and should be long enough to reach from the PCB to the place on the robot where you attach the motors. About 10 - 20cm is a good estimate. Connect the left terminal of motor A to the left screw terminal for motor A on the board, and the right terminal of the motor to the right terminal. Do the same with motor B. Motor A is the left motor when viewed in the direction of travel, motor B is the right.

If you have four motors in your kit, then the two on one side of the robot are connected in parallel. See below for more details.



Connecting the motors

If you connect the cables for the motors the wrong way round, nothing will break, but your robot will make strange movements. If the left and right motors are reversed, the RoBat will turn left when it should turn right and vice versa. If one motor has the wires connected to the terminal the wrong way round, the motor will run backwards when it

should be running forwards. Your robot will then turn instead of going straight. If the cables of both motors are connected to the terminals the wrong way round, the robot will run backwards when it should be running forwards and vice versa.

When connecting the motors, you must be aware that the contacts of the motors are relatively sensitive and break off easily when they are pulled. Therefore, a strain relief is attached as soon as the cables are soldered on. You can do the strain relief with a drop of hot glue to fix the wires to the motor. Don‘t apply glue to the contacts directly, though. You wouldn‘t be able to desolder them should the need arise. If you have them, you can also use cable ties for the strain relief.

Robot with four motors

If you want your robot to drive in difficult terrain or to be extra strong, you can connect two more motors instead of the furniture roller, making a total of four. The two motors on each side of the robot are connected in parallel, i.e. the left contact of one motor is connected to the left contact of the other motor and the same for the right contacts.

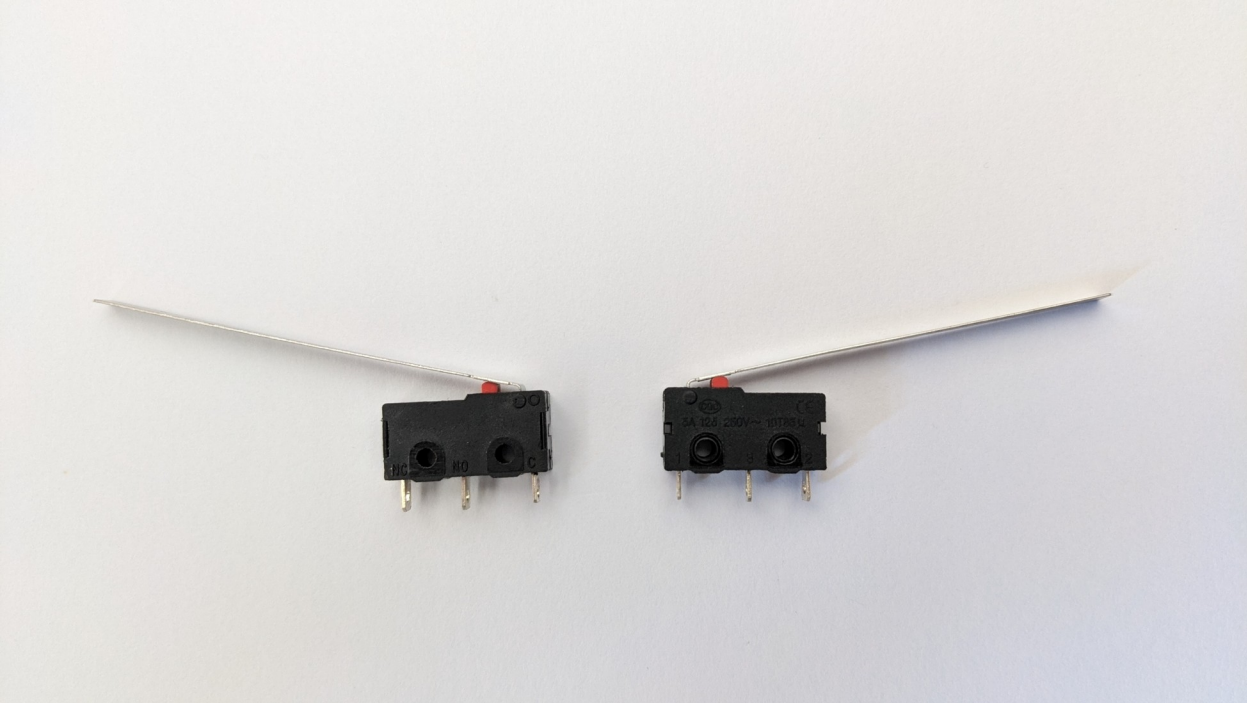
That means, you have to connect both wires from the left connection of the two motors on one side of the robot to one of the sockets of the left screw terminal and both wires from the right connection of the two motors to the other socket of the left srwe terminal.

Antennae

Your robot can get two antenna-like sensors as feelers with which it can touch its environment. The sensors are switches that close when the robot hits an obstacle. Normally one sensor is placed on the front left and one on the front right. We also call them bumper switches sometimes. Depending on what the robot should be able to do, this can be different.

You can either build the switches yourself or use ready-made end switches. Each switch connects one of the input pins with GND. That is input pin A2 for the left switch and pin A3 for the right switch.

There are many possibilities for the construction of the sensors. In principle, two pieces of metal are always connected together, one of which is connected to GND and the other to the input pin.

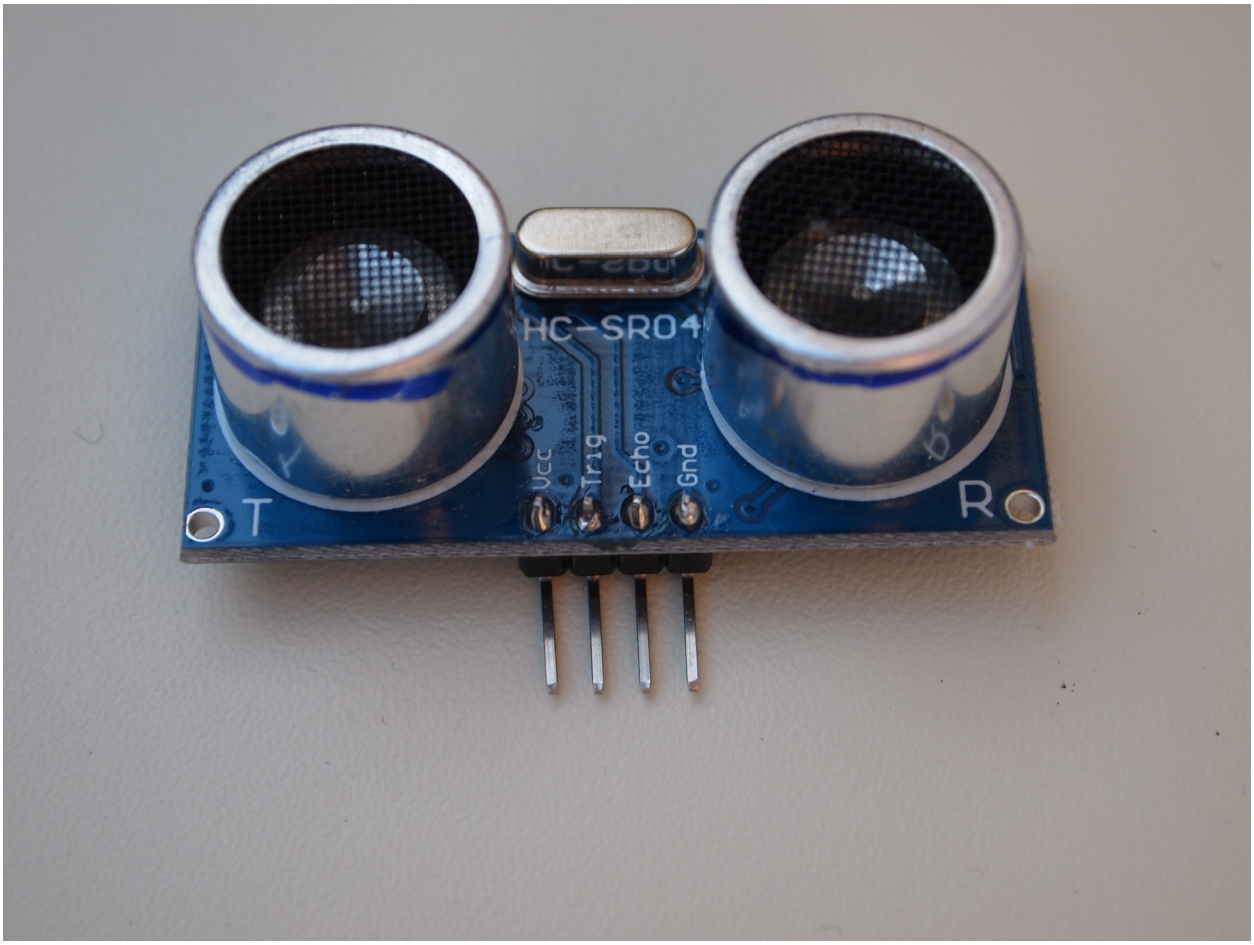


Two end switches as sensors for the robot

Ultrasound range finder

With the ultrasonic rangefinder HC SR-04 RoBat can detect obstacles.

This works similar to a bat (RoBat got its name from the word "bat"). The ultrasonic sensor emits a very high pitched sound inaudible to humans. The sound gets reflected from the surface of objects. Then the microcontroller measures the time until the echo returns and calculates the distance to the object from that time. Sometimes this doesn't work, i.e. if the sound is not reflected by small objects. In this case RoBat cannot detect the obstacle.



Ultrasonic range finder HC SR-04

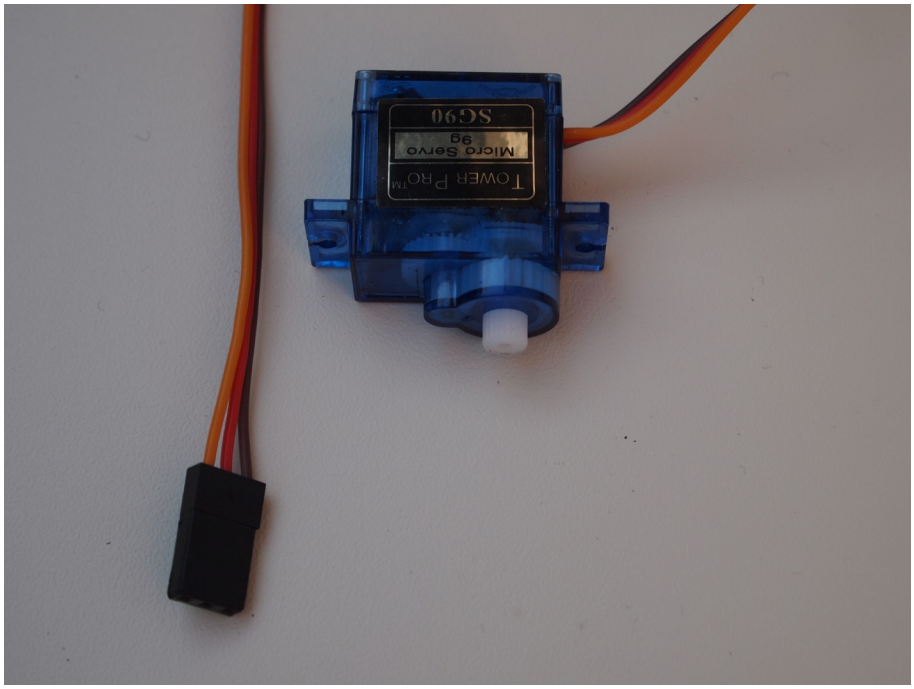
Servo

A servo is used to move something to a certain position, such as the rudder on an airplane. RoBat uses a servo to turn the ultrasonic sensor in different directions.

A servo consists of a small geared motor and a potentiometer that can be used to measure the current position of the motor. This allows the microcontroller to set the servo to an angle between 0 and 180 degrees.

The circuit board has connectors for two servos. The one we use for moving the ultrasonic sensor is labelled „servo 2“. The other one, labelled „servo 1“ can be used to control a small gripper to let the robot grab objects or for some other purpose.

A servo is connected with a plug to the previously soldered pin header. You have to pay attention to the polarity. The servo has three cables in the colors brown, yellow and red. The brown cable must be connected to pin GND.



Servo with connector

Joystick

The joystick allows you to manually control your RoBat. It has two axes (X- and Y-axis). It detects how far the joystick is pressed in X-direction and Y-direction with an adjustable resistor (a so called „potentiometer“. Additionally, it has a small switch that closes when you press down the control button.

The joystick is connected with 5 cables. Take 5 wires from the ribbon cable included in the kit and connect the identically labeled solder points on the PCB with those on the joystick.

If you have access to a crimping tool, you can also put small terminals on each side of the cable. Then you can pull the cable off when you don't need it.

Joystick

Manual control or autonomous robot?

Now you have to decide: do you want to control your robot yourself or do you want to let it drive by itself?

If you want to control it, you need the joystick. If you want it to drive itself, your robot needs the servo and the ultrasonic sensor and/or the two bumber switches.

You can also connect all five components and decide how the robot is controlled each time you turn it on.

Control with the joystick

When RoBat is turned on, if the joystick button is pressed (or pin 2 is set to LOW) or the ultrasonic sensor detects an obstacle 1cm - 10cm away, RoBat will start in manual mode so you can control it with the joystick.

Battle mode

If RoBat sees an obstacle at a distance of about 10cm to 50cm, it goes into battle mode, chooses the nearest object as an opponent and attacks it. This mode is also selected if both bumper switches are pressed when switching on. We also call this the Hebocon mode. Hebocon is a japanese word for a competition with homemade "junk" robots with the goal of pushing an opponent out of the fighting arena.

Obstacle avoidance

If RoBat detects the next obstacle in 50cm to 1m distance when it is switched on, it starts in obstacle avoidance mode. It then tries to avoid obstacles by itself and does not react to the joystick.

Control via the serial interface

Your RoBat can also be controlled via the serial interface on pins TX and RX on the Arduino. This control is for when you want another computer or microcontroller to control the RoBat. Furthermore you can control RoBat via Bluetooth if you connect a Bluetooth module HC-05 or HC-06 to these pins.

The serial mode is selected if RoBat has a clear view when switched on, i.e. the next obstacle is further than 1m (100cm) away.

The serial mode can also be set via the interface itself. To do this, the command '`serialcmd`' must be sent via the serial interface.

Serial interface and Bluetooth

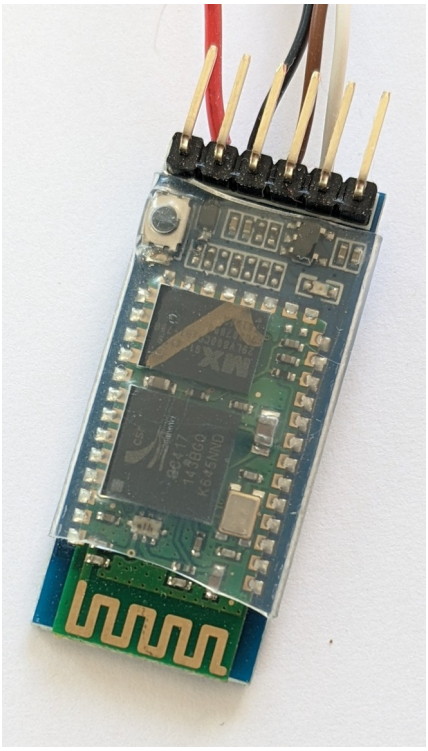
The serial interface of the Arduino is connected to the pins RX and TX. The R in RX stands for "receive" and the T in TX for "transmit". The serial interface must be configured. In particular the transmission rate must match that of the remote station. Common transmission rates are 9600 baud (bit/s), 19200, 38400, 57600 and 115200. The baud rate is set with the command `Serial.begin(BAUDRATE);`.

For the transmission of data, the TX pin of the Arduino is connected to the RX pin of the remote terminal and the RX pin of the Arduino is connected to the TX pin of the remote terminal. The connection is therefore crossed. Clearly, the data that goes out on one side comes in at the remote station and vice versa.

The serial interface of the Arduino (pins RX and TX) is also used to transfer programs to the Arduino. Therefore they must not be connected to other modules during transmission.

Bluetooth module HC-05 / HC-06

Bluetooth modules are available in different variants, the most common are HC-05 and HC-06. For RoBat we use the HC-05 module.



Usually the baud rate of the module is 38400 bit/s. But it can also be 9600 bit/s by default. If the module was delivered with the RoBat kit, the baud rate is 38400 bit/s and you don't have to do anything else. If you got the module yourself, you have to adjust the baudrate. Please follow the instructions from Instructables:

https://www.instructables.com/Change-the-Baud-Rate-of-HC-05-Bluetooth-Module-Usi/

Serial commands

The RoBat software processes various commands that are sent to the Arduino via the serial interface. The Arduino must be in serial mode for this, otherwise only the command to switch on the serial mode is considered. You reach serial mode when RoBat is more than one meter away from the next object when it is switched on.

The following commands are currently available:

Command	**Parameters and meaning**
`serialcmd`	Switching on the serial command mode
`manual`	Switching on manual control (joystick control)
`battle`	Switching on the battle mode (attack)
`autonomous`	Switching on the obstacle avoidance mode
`dist`	Query of the distance to the next object. The distance in cm is sent back via the serial interface.
`lf nnn`	Switch on left motor with speed nnn forward. nnn is an integer between 0 and 255. example: `lf 120`
`lb nnn`	Switch on the left motor in reverse at speed nnn.
`rf nnn`	Right motor forward
`rb nnn`	Right motor backward
`st`	stop both motors (stop)

To keep the motors running, the commands must be repeated as long as the motor or motors are to run. If RoBat does not receive any command for more than half a second, the motors will stop for safety reasons. We assume that in this case the connection is broken and RoBat can no longer be controlled.

Software

Before RoBat can do anything, you have to teach it. You can tell the microcontroller what to do in which situation with a program.

We have prepared a program for RoBat to get you started. You can find it here: https://github.com/merlanura/robat/ You can also customize it yourself if you want. In the GitHub directory you can also find the construction plan of the board and download this manual in PDF format.

To install the program, you must first install the Arduino IDE on your computer. An „IDE“ is an integrated development environment, basically a combination of an editor and a compiler. It is used for writing programs and translating them so the Arduino understands them. You‘ll find the program "Arduino IDE" for Windows, Mac and Linux on this website: https://www.arduino.cc/en/main/software. You can also write programs online, please see the website for details. We assume here that you have installed the Arduino software on your computer.

To program, you connect the Arduino to your computer via a USB cable.

Arduino software settings

To install a program on the Arduino, the software must be set. You can do this via the menu item "Tools". You will want to adjust the following settings:

Setting	**Value**
Board	Arduino Nano
Processor	ATmega328P (Old Bootloader)
Port	see below

To find out which port the Arduino is connected to, simply do the following: Look at the list of ports in the Port setting while the Arduino is not connected. Then plug in the Arduino and see which port has been added. This is the port that the Arduino is connected to and that you need to set here.

Platform and body

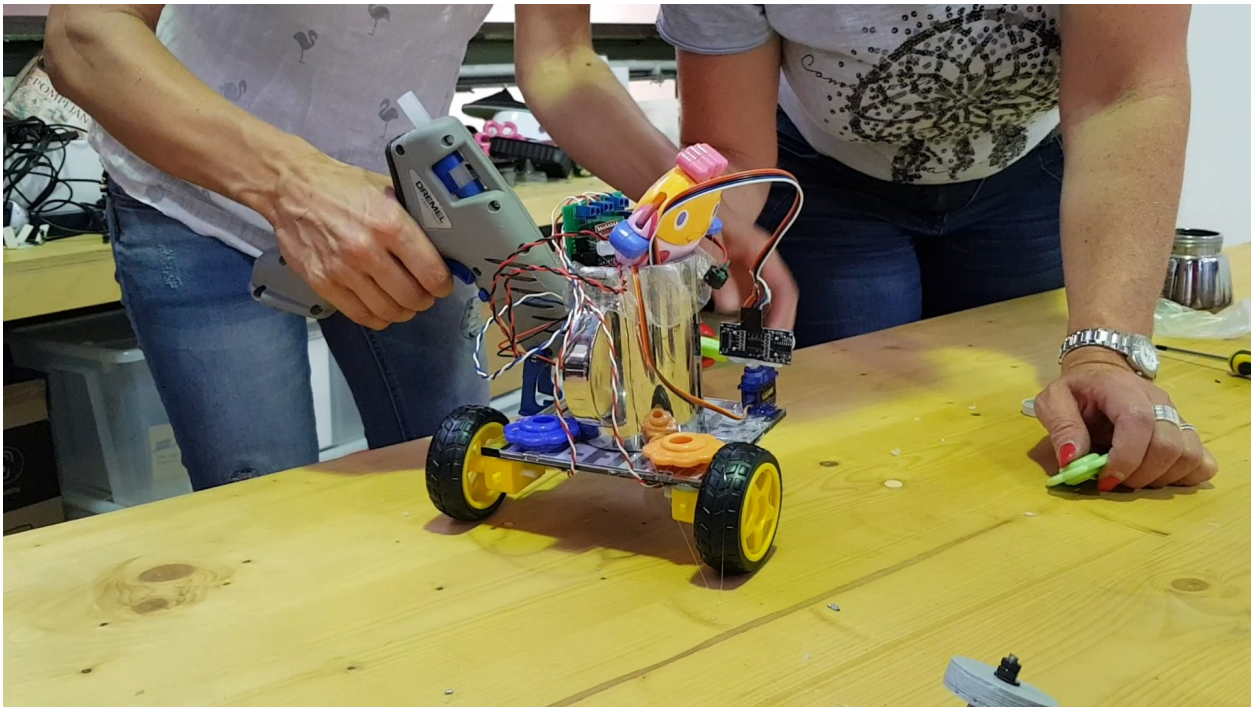
Now your RoBat is almost finished. But it doesn't have a body yet, the board and the other components are hanging loose on the cables. Time for the creative part!

First, find a platform that will form the basic framework of the robot. This can be a piece of sturdy cardboard or a small board (dimensions about 10cm x 15cm x 0.5cm), but also something else, e.g. an old CD, an empty milk carton or almost any other flat object. The only condition: the surface must not conduct electricity, so a metal plate is not possible.

You can also use our template to cut out or saw out a matching platform.

1. Glue the yellow gear motors with the hot glue gun under the platform on the left and right side. Then attach the caster wheel in front under the platform. This creates a kind of tricycle with the two drive motors at the back.
2. Switch on RoBat and wait until it has played the melody. The servo will align itself. Switch it off immediately after you hear the beep.
3. Screw one of the white plastic parts (so called „horns“) onto the servo and glue the ultrasonic sensor onto it. Glue the servo on the platform so that the ultrasonic sensor points to the front.
4. Attach the board to the platform with one or two small glue dots.
5. Use some hot glue to attach the battery holder and the switch in a suitable place on or under the platform. The batteries usually don't fall out when you turn the battery holder upside down, so you can also glue it to the bottom of your robot.

Design the robot as you like with additional parts made of cardboard, plastic, paper, etc.



Customize your RoBat as you like

Have fun with your RoBat!

Help

You can find more information about RoBat on our website https://robot-workshop.de.

There are a number of hardware vendors that provide very detailed and good documentation and tutorials related to the Arduino and their products. These include the following:

- Adafruit, https://www.adafruit.com
- Sparkfun, https://www.sparkfun.com
- Pololu, https://pololu.com
- Joy IT, http://anleitung.joy-it.net

And lots of great websites with robots:

- http://robotics.hobbizine.com
- https://dronebotworkshop.com
- https://servomagazine.com
- https://diyodemag.com/

This is a good place to buy robot parts:

- https://shop.pimoroni.com/
- https://www.roboter-bausatz.de/

Good instructions for the Bluetooth modules HC-05 and HC-06

- https://wolles-elektronikkiste.de/hc-05-und-hc-06-bluetooth-module
- https://www.instructables.com/Change-the-Baud-Rate-of-HC-05-Bluetooth-Module-Usi/

Contact

For questions and suggestions contact:

Michael Ibsen
Sandbergstrasse 59
64285 Darmstadt
Germany

Mail: ibsen@gmx.net

Twitter @merlanura

https://robot-workshop.de