EXPERIMENT MANUAL

ROBOTICS WORKSHOP

THAMES & KOSMOS

620377-03-181016

Franckh-Kosmos Verlags-GmbH & Co. KG, Pfizerstr. 5-7, 70184 Stuttgart, Germany | +49 (0) 711 2191-0 | www.kosmos.de
Thames & Kosmos, 301 Friendship St., Providence, RI, 02903, USA | 1-800-587-2872 | www.thamesandkosmos.com
Thames & Kosmos UK Ltd, Goudhurst, Kent, TN17 2QZ, United Kingdom | 01580 212000 | www.thamesandkosmos.co.uk

›› › IMPORTANT INFORMATION

Safety Information

WARNING! Only for use by children aged 10 years and older. Instructions for parents or other supervising adults are included and have to be observed. Keep the packaging and instructions as they contain important information.
WARNING! Not suitable for children under 3 years. Choking hazard — small parts may be swallowed or inhaled.
WARNING! Do not aim at eyes or face.
Never launch heavy, sharp-pointed, or sharp-edged objects.

Dear Parents and Supervising Adults,

This experiment kit will introduce your child to the exciting world of robotics and programming in a fun and simple way. Please be available to provide your child with help, advice, and support.

It is natural to have questions about safety. This kit meets U.S. and European safety standards. These standards impose obligations on the manufacturer, but also stipulate that adults should provide their children with advice and assistance during the experiments.

Tell your child to read all the relevant instructions and safety information, and to keep these materials on hand for reference. Be sure to stress the importance of following all the rules and information when performing the experiments.

We wish your child, and of course you as well, lots of fun and success with the experiments!

FCC Part 15 Statement

This device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) this device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

Warning: Changes or modifications to this unit not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment.

NOTE: This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications.
However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:
• Reorient or relocate the receiving antenna.
• Increase the separation between the equipment and receiver.
• Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
• Consult the dealer or an experienced radio/TV technician for help.

Shielded cables must be used with this unit to ensure compliance with the Class B FCC limits.

FCC RF Exposure Statement

To comply with the FCC RF exposure compliance requirements, this device and its antenna must not be co-located or operating in conjunction with any other antenna or transmitter.

For body worn operation, this device has been tested and meets FCC RF exposure guidelines when used with an accessory that contains no metal and that positions the device a minimum of 20 cm from the body. Use of other accessories may not ensure compliance with FCC RF exposure guidelines.

IC Statement

This Class B digital apparatus complies with Canadian ICES-003.

Cet appareil numérique de classe B est conforme à la norme NMB-003.

This device complies with Industry Canada licence-exempt RSS standard(s). Operation is subject to the following two conditions: (1) this device may not cause interference, and (2) this device must accept any interference, including interference that may cause undesired operation of the device.

Le présent appareil est conforme aux CNR d'Industrie Canada applicables aux appareils radio exempts de licence. L'exploitation est autorisée aux deux conditions suivantes:
(1) l'appareil ne doit pas produire de brouillage, et
(2) l'utilisateur de l'appareil doit accepter tout brouillage radioélectrique subi, même si le brouillage est susceptible d'en compromettre le fonctionnement.

RF Radiation Exposure Statement:
This equipment complies with IC radiation exposure limits set forth for an uncontrolled environment. This equipment should be installed and operated with minimum distance 20 cm between the radiator and your body. This transmitter must not be co-located or operating in conjunction with any other antenna or transmitter.

Déclaration d'exposition aux radiations:
Cet équipement est conforme aux limites d'exposition aux rayonnements IC établies pour un environnement non contrôlé. Cet équipement doit être installé et utilisé avec un minimum de 20 cm de distance entre la source de rayonnement et votre corps.

Disposal of Electrical and Electronic Components
This product's electronic parts are reusable and, for the sake of protecting the environment, they should not be thrown into the regular household trash at the end of their lifespan. Instead, they must be delivered to a collection location for electronic waste, as indicated by the following symbol:
Please consult your local authorities for the appropriate disposal location.

Disposal of Battery
The battery does not belong in the household trash! In some states and countries, it is required by law to deliver batteries and rechargeable batteries to a local collection location or to a store. This will ensure that they will be disposed of in an environmentally responsible manner. Batteries containing hazardous substances are identified by this image or by chemical symbols (Cd = cadmium, Hg = mercury, Pb = lead).

Safety for Experiments with Batteries
››› To operate the models, you will need six AA batteries (1.5-volt, type AA/LR6) or six AA rechargeable batteries (1.2-volt, type AA, HR6/KR6), which could not be included in the kit due to their limited shelf life.
››› The supply terminals are not to be short-circuited. A short circuit can cause the wires to overheat and the batteries to explode.
››› Different types of batteries or new and used batteries are not to be mixed.
››› Do not mix old and new batteries.
››› Do not mix alkaline, standard (carbon-zinc), or rechargeable (nickel-cadmium) batteries.
››› Rechargeable batteries are only to be charged under adult supervision.
››› Non-rechargeable batteries are not to be recharged. They could explode!
››› Never perform experiments using household current! The wires are not to be inserted into socket-outlets. The high voltage can be extremely dangerous or fatal!
››› Batteries are to be inserted with the correct polarity. Press them gently into the battery compartment. See page 2.
››› Always close the battery compartment with the lid.
››› Avoid deforming the batteries.
››› Rechargeable batteries are to be removed from the toy before being charged.
››› Exhausted batteries are to be removed from the toy.
››› Be sure not to bring batteries into contact with coins, keys, or other metal objects.
››› Keep the kit out of the reach of small children.

DC Power Supply (Not Included)
A 5V 2.5A power supply is recommended, such as an AC adapter, portable charger, or other type of power supply.

The toy is only to be connected to Class II equipment bearing the following symbol:

Kosmos Quality and Safety

Quality & Safety KOSMOS 100 YEARS

More than one hundred years of expertise in publishing science experiment kits stand behind every product that bears the Kosmos name. Kosmos experiment kits are designed by an experienced team of specialists and tested with the utmost care during development and production. With regard to product safety, these experiment kits follow European and US safety standards, as well as our own refined proprietary safety guidelines. By working closely with our manufacturing partners and safety testing labs, we are able to control all stages of production. While the majority of our products are made in Germany, all of our products, regardless of origin, follow the same rigid quality standards.

1st Edition 2016 Thames & Kosmos, LLC, Providence, RI, USA
Thames & Kosmos® is a registered trademark of Thames & Kosmos, LLC.

This work, including all its parts, is copyright protected. Any use outside the specific limits of the copyright law without the consent of the publisher is prohibited and punishable by law. This applies specifically to reproductions, translations, microfilming, and storage and processing in electronic systems and networks. We do not guarantee that all material in this work is free from copyright or other protection.

Technical product development: Genius Toy Taiwan Co., Ltd., Taichung, Taiwan, R.O.C.
Text and Editing: Camille Duhamel, Ted McGuire
Additional Graphics and Packaging: Dan Freitas, Ashley Greenleaf
Manual design concept: Atelier Bea Klenk, Berlin
Manual illustrations: Genius Toy Taiwan Co., Ltd., Taichung, Taiwan, R.O.C., and Thames & Kosmos
Manual photos: Bin im Garten, CC BY-SA 3.0, p. 4 (hex bot); Neobotix GmbH, p. 4 (transport robot); Courtesy of NASA, p. 4 (Mars rover); iStock, p. 4 (ASIMO and robotic arm); Fotolia, p. 4 (vacuum bot); iStock, p. 6 (animals); iStock, p. 7 (button);
All other photos: Genius Toy Taiwan Co., Ltd., Taichung, Taiwan, R.O.C., and Thames & Kosmos

The publisher has made every effort to locate the holders of image rights for all of the photos used. If in any individual cases any holders of image rights have not been acknowledged, they are asked to provide evidence to the publisher of their image rights so that they may be paid an image fee in line with the industry standard.

Android, Google Play and the Google Play logo are trademarks of Google Inc.
Apple and the Apple Logo are trademarks of Apple Inc., registered in the USA and other countries. App Store is a service mark of Apple Inc.

Distributed in North America by Thames & Kosmos, LLC. Providence, RI 02903
Phone: 800-587-2872; Web: www.thamesandkosmos.com

Distributed in United Kingdom by Thames & Kosmos UK, LP. Goudhurst, Kent TN17 2QZ
Phone: 01580 212000; Web: www.thamesandkosmos.co.uk

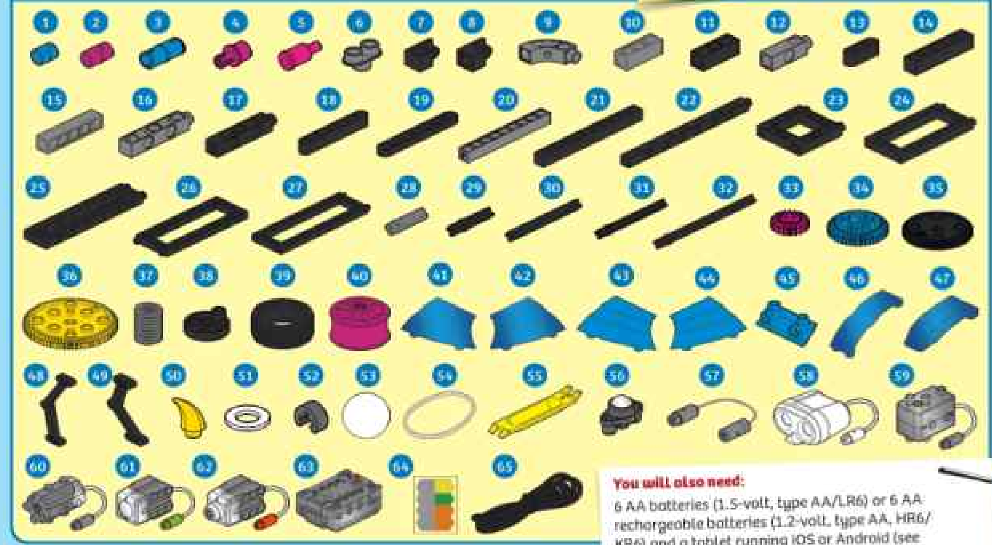
We reserve the right to make technical changes.

Printed in Taiwan / Imprimé en Taiwan

››› KIT CONTENTS

GOOD TO KNOW! If you are missing any parts, please contact Thames & Kosmos customer service.
US: techsupport@thamesandkosmos.com
UK: techsupport@thamesandkosmos.co.uk

What's inside your experiment kit:



You will also need:
6 AA batteries (1.5-volt, type AA/LR6) or 6 AA rechargeable batteries (1.2-volt, type AA, HR6/KR6) and a tablet running iOS or Android (see page 8 for hardware requirements)

Checklist: Find – Inspect – Check off

✔	No.	Description	Qty.	Item No.
○	1	Short anchor pin	46	7344-W10-C2B1
○	2	Anchor pin	25	7061-W10-C2R1
○	3	Joint pin	21	1116-W10-A1B
○	4	Shaft plug	12	7026-W10-H1R
○	5	Shaft pin	8	7026-W10-J1R1
○	6	Two-to-one converter	12	7061-W10-G1S2
○	7	90-degree converter X, black	2	7061-W10-J1D
○	8	90-degree converter Y, black	2	7061-W10-J2D
○	9	Curved rod, gray	2	7026-W10-V1S2
○	10	3-hole rod	5	7026-W10-Q1S1
○	11	3-hole cross rod, black	5	7026-W10-K1D
○	12	3-hole dual rod, gray	12	7061-W10-R1S2
○	13	3-hole wide rounded rod, black	6	7404-W10-C1D
○	14	5-hole rod B, black	6	7413-W10-K2D
○	15	5-hole rod C, gray	1	7413-W10-A1S1
○	16	5-hole dual rod B, gray	3	7026-W10-S2S1
○	17	5-hole dual rod C, black	2	7026-W10-S3D
○	18	7-hole wide rounded rod, black	5	7404-W10-C2D
○	19	7-hole flat rounded rod, black	5	7406-W10-C3D
○	20	9-hole rod	4	7407-W10-C1S
○	21	11-hole rod	7	7413-W10-P3D
○	22	15-hole dual rod	3	7413-W10-H1D
○	23	5x5 square frame	6	7026-W10-T2D
○	24	5x10 frame	2	7413-W10-I1D
○	25	3x13 dual frame	2	7406-W10-A1D
○	26	5x13 dual frame	3	7060-W10-U1D
○	27	5x15 frame	4	7413-W10-J1D
○	28	Motor shaft	6	7026-W10-L1S1
○	29	35-mm axle	4	7413-W10-D1D
○	30	65-mm axle	2	7406-W10-C1D
○	31	70-mm axle	11	7061-W10-Q1D
○	32	100-mm axle	1	7413-W10-L2D
○	33	Small gear, magenta	11	7026-W10-D1R
○	34	Medium gear, blue	9	7344-W10-C1B1
○	35	Large gear, black	2	7026-W10-W5D
○	36	Extra large gear, yellow	3	7328-W10-G1Y
○	37	Worm gear	4	7344-W10-A1S1
○	38	Rod connector	3	7026-W10-L2D
○	39	Tire	2	7407-W10-A1D
○	40	Wheel	2	7407-W10-B1R
○	41	Body plate left, transparent	1	7392-W10-L1B
○	42	Body plate right, transparent	1	7392-W10-L2B
○	43	Body plate left, blue	1	7392-W10-L1B1
○	44	Body plate right, blue	1	7392-W10-L2B1
○	45	Side plate	2	7392-W10-M1B1
○	46	Large body plate	2	7398-W10-C1B
○	47	Small body plate, B	2	7398-W10-C2B
○	48	Left leg	1	7397-W10-C1D
○	49	Right leg	1	7397-W10-C2D
○	50	Horn	2	7128-W10-G1Y
○	51	Washer	2	R1343630
○	52	Axle lock	1	3620-W10-A1D
○	53	Large foam ball	3	K3087366-2
○	54	Rubber band	1	R50-02
○	55	Anchor pin lever	1	7061-W10-B1Y
○	56	Ball roller	1	1247-W85-C1S
○	57	Extension cord	1	1246-W85-D
○	58	Ultrasonic sensor	1	7414-W85-B
○	59	Motor (40x DDM)	2	7400-W85-A1
○	60	Motor (32x DDM)	2	7392-W85-B1
○	61	Light sensor	1	1246-W85-B
○	62	Touch sensor	2	1246-W85-C
○	63	CB1 Core controller	1	1246-W85-A1-US
○	64	Sticker sheet	1	R20812464-2
○	65	USB 2.0 cable	1	E3081247A

620377-02-071016

››› TIPS AND TRICKS

Here are a few tips for assembling and using the models. Read them carefully before starting.

A. Pay attention to the hole alignment!

It is very important that you pay close attention to the alignment of the holes in the gear wheels. Make sure that you insert the shaft pins into the correct holes and that the gears are oriented exactly as shown in relation to each other and to the model. Otherwise, the models will not move properly.

B. The anchor pin lever

In the box, you will find a little yellow tool called the anchor pin lever or part separator tool. End A of the tool makes it easy to remove anchor pins from the frames. End B can be used to pry pieces apart.

C. Gear wheels

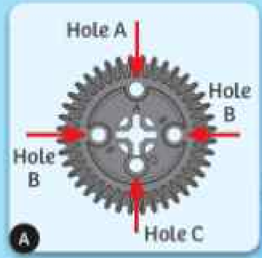
The models will often have several gear wheels installed in a row, or gear train. In order for the models to work well, these gears will have to mesh well. Otherwise, the force from one gear wheel won't be properly transferred to the next.

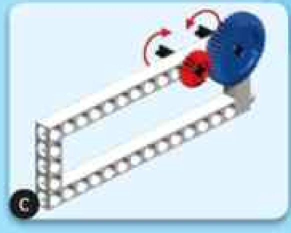
D. Installing batteries in the core controller

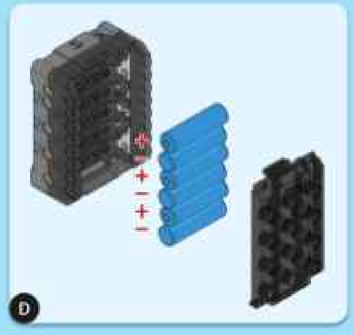
Slide the transparent cover open. Insert the batteries according to the indicated plus-minus polarity. Close the compartment by snapping the cover back on.

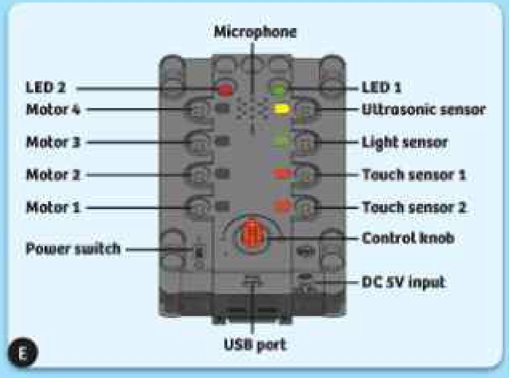
E. Placing the stickers on the CB1 core controller

Place the stickers from the sticker sheet on the core controller in the right spots. This will make it easier to plug the sensors into the correct locations.







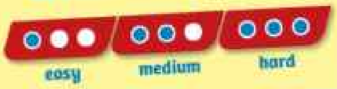


››› TABLE OF CONTENTS

TIP!

Above each set of assembly instructions, you will find a red bar:

››› It shows you the difficulty level for the model's assembly:



Robotics Workshop: The Robot Builder's Toolkit

Robots are mechanical agents controlled by computer programs. They can be programmed to perform all sorts of tasks and movements. Robots can assemble cars, play soccer, vacuum floors, deliver packages, map terrain, climb mountains, entertain people, cook dinner — the list goes on and on. With this kit, you can build robots that use different sensors to sense their environment. With the app, you can program the robots' motors, lights, and speaker to behave in different ways depending on the data coming from their sensors. First, build the robots in this manual to learn how everything works, and then design your own robots!

GETTING STARTED

There are six primary functional components in this kit that enable the robots to work:

A. **The CB1 core controller** connects to the app on your tablet via a wireless Bluetooth connection, and provides power to the motor units via wires. It connects to the sensor units to transmit sensor input back to the program. The core controller also contains a microphone that can be used to detect sounds, a speaker to make sounds, and a rotary knob.

B. **Motor units 1 and 2** connect to axles and motor shafts to turn gears and wheels, activating your models. They are powered by the core controller.

C. **The ultrasonic sensor** sends out ultrasound waves and "listens" for them to bounce off of objects. It gives this information to the app, enabling it to estimate the distance to the objects.

D. **The light sensor** is able to detect changes in the amount of light shining on the sensor. It then sends this information to the app.

E. **The touch sensor**, like your fingers, is able to sense if it has come into contact with an object. It then sends this information to the app.

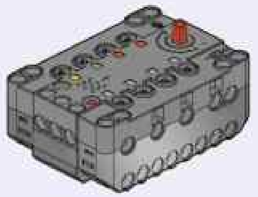
F. **CB1 Blockly: The mobile app** (or desktop PC application) is the "brain" of your robotic models. It uses the feedback from the sensors along with programmed instructions to control the models via the core controller.

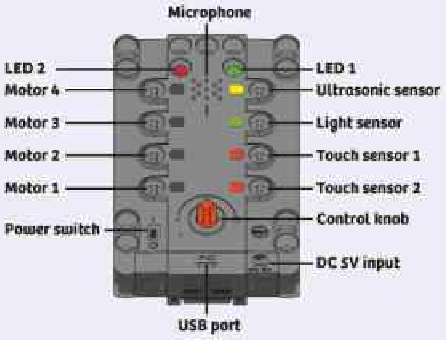
These six elements, in combination with all the mechanical parts — rods, gears, axles, frames, and so on — allow you to build and program mechanical robots that can sense and respond to their surroundings.

First, follow the assembly instructions starting on page 12 to build one of the models. Make sure you have inserted the batteries correctly according to the battery information on page 2.

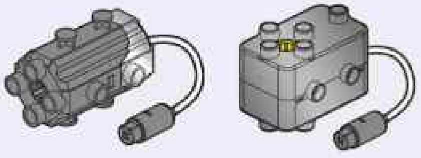
Now you can connect the tablet to the model via the free app. Instructions to download and use the app start on page 8.

A

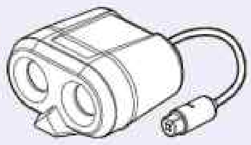




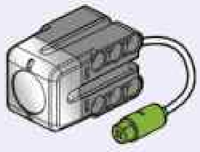
B



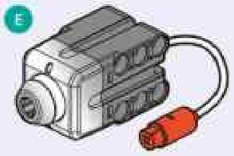
C



D



E



F

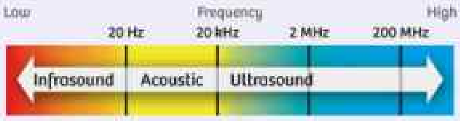
CB1 Blockly

ABOUT THE SENSORS

WHAT IS ULTRASOUND?

Ultrasound is a sound pressure wave that moves through substances (gases, liquids, and solids) and has a frequency greater than that which humans can hear.

Frequency is simply the number of waves in a given period of time. Humans can hear sound waves in the frequency range of 20 hertz (which means cycles per second) to 20,000 hertz (20 kilohertz).

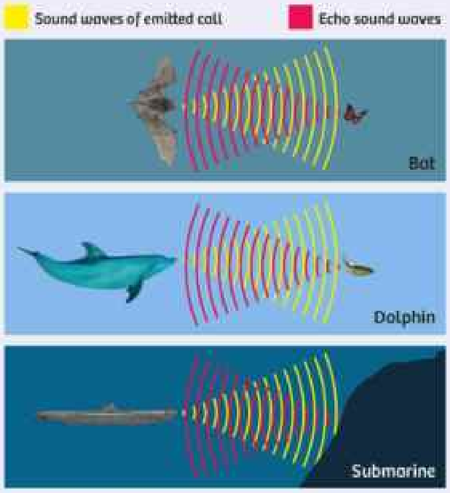


ANIMAL SONAR

In the natural world, bats, whales, and some birds use sound waves to detect objects around them. This is especially useful in the darkness of night or underwater, where seeing visible light is difficult or impossible.

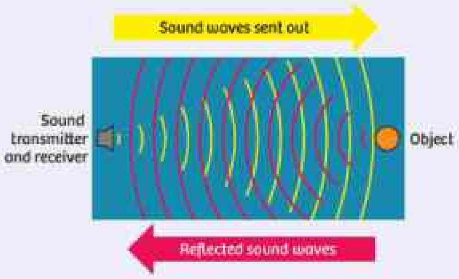
This type of sensing is called **echolocation**, or biosonar. It works like this: The animal emits sound waves that move outward in all directions around it. When the sound waves hit an object, they bounce off of it and travel back to the animal's ears. The sound waves reach each of the animal's two ears at slightly different times. The animal can interpret this time difference to perceive the size, direction of movement, and speed of objects.

Animals use echolocation to hunt prey in the dark of night. They can also navigate and find their way around without normal eyesight.



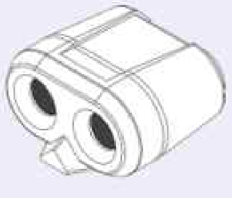
ACTIVE SONAR

Humans have developed a technology to replicate echolocation, which is called **sonar.** Sonar is an acronym for **SO**und **N**avigation **A**nd **R**anging. Sonar is used for navigation in submarines, ships, and airplanes. A similar technology using electromagnetic radio waves instead of sound waves is called **radar** (**RA**dio **D**etection **A**nd **R**anging).



With sonar and radar, airplane pilots are able to find their way and avoid collision with other planes, even in complete darkness or in thick clouds. Police use radar to detect speeding cars.

Your ultrasonic sensor also sends out sound waves. You can't hear them because they are ultrasonic! One "eye" on the sensor head is a transmitter and the other is a receiver. The transmitter sends out ultrasonic waves, like a speaker, and the receiver senses the waves that bounce back, like a microphone. In this way, the sensor is able to sense objects in front of it, even in the dark.


The ultrasonic sensor has one transmitter and one receiver.

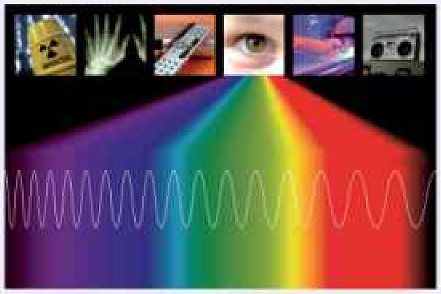
ABOUT THE SENSORS

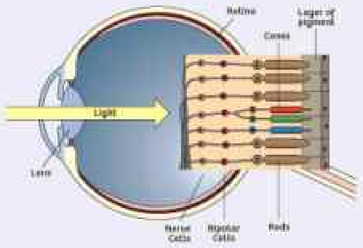
WHAT IS LIGHT?

Have you ever thrown pieces of gravel into a lake? When you do that, you can see how each pebble creates a small circular ripple in the water. The same thing happens when a ship passes by the shore of the ocean or a river. If you watch closely, you may also be able to tell that larger stones create bigger waves than smaller stones do.

According to one explanation by scientists, light also spreads out like a wave. Each color in the rainbow is a slightly different wave with its own wavelength, as you can see in the picture to the right. Light is just one part of a much larger spectrum of waves — specifically, the part that we can see with our eyes. Other waves from this spectrum, called the electromagnetic spectrum, flow as electric current out of your wall outlet, or supply a radio with signals it turns into music.

A person is able to see light because the retina in his or her eye has special mini-sensors, or sensory cells. There are two different types: The 125 million or so rods in each eye can perceive something when there are only very few light portions, but they can only discern bright from dark, or black from white. The seven million or so cones, on the other hand, perceive colors. But to do that, they need a lot more light than the rods. On top of that, the two types of sensors are not distributed evenly over the retina.


Electromagnetic spectrum



Cross section diagram of human eye

SENSING LIGHT

The digital light sensor in this kit is not nearly as sophisticated a sensor as the human eye, but it can detect different light levels. It is similar to the light sensor used in mobile phones and computer keyboards to measure ambient light so that the backlighting can be adjusted.

The light sensor outputs its light measurements in units called lux. Lux is a measure of the amount of light that hits or passes through a surface. A given amount of light will illuminate a surface more dimly if it is spread over a larger area. For example, if you place a lamp very close to a table it will illuminate a small area very brightly. If you move the lamp far away from the table the whole table will be illuminated but not as brightly because of the increased area.

SENSING TOUCH

The touch sensor in this kit is simply a push button switch with a special housing. A push button is a type of switch that electrically connects two terminals, usually when a spring-loaded button is depressed, allowing current to flow as long as the button is depressed. When the button is released, the circuit is broken again and the current stops flowing. The touch sensor in this kit only has two states: on or off. It cannot sense varying levels of force or pressure.

Push button electrical component

INSTALLING THE MOBILE APP

You can download the free app for iOS devices from the iOS App Store or for Android devices from Google Play. The official app is published by T2T Inc.

- iOS devices must support Bluetooth 4.0 and must be running iOS 8 or later. iOS 9 or 10 is recommended. Supported devices include iPad 3rd generation or later, iPad mini, iPad Air, and iPad Pro. An iPad display of 9.7 inches or larger is recommended.
- Android devices must support Bluetooth 4.0 and must be running Android 4.4 or later. A display of 9.7 inches or larger is recommended. Because of the large number of Android devices on the market, it is impossible for us to recommend specific Android devices. NOTE: The CB1 Blockly mobile app was developed on Android 6.0. It is expected to work on Android 5.x or 4.4, but the compatibility cannot be verified on all devices.

To find, download, and install the app:

1. In a web browser on your tablet, navigate to the Robotics Workshop product page on the Thames & Kosmos website.
   http://thamesandkosmos.com/index.php/product/category/science-kits/robotics-workshop
2. On this web page, there are links to the app pages in the iOS App Store and Google Play. Click the link to go to the appropriate store for your device.
   Alternatively, you can search for "CB1 Blockly" or "Robotics Workshop" in the app stores.
3. Follow the steps on the app page to download and install the app on your device.

CONNECTING THE MOBILE APP TO THE CB1 CORE CONTROLLER

The mobile app will automatically connect to the CB1 core controller via Bluetooth. Make sure that Bluetooth is enabled on your mobile device and the core controller is turned on.

If you are having trouble connecting to the core controller, first try closing and then reopening the app.

CB1 Blockly

TROUBLESHOOTING THE CONNECTION

If the connection isn't working:

››› Disconnect and then reestablish the Bluetooth connection.

››› Make sure the sensor and motor cables are securely plugged into the core controller.

››› Exit the program you are in and relaunch it. Or try a different program.

CB1 BLOCKLY DESKTOP INSTRUCTIONS

- The desktop app is not supported on all computers. A Windows desktop or laptop PC with USB 2.0/3.0 port and Internet connection are required. NOTE: The CB1 Blockly Desktop application was developed on Windows 10. It is expected to work on Windows 7 or higher, but the compatibility cannot be verified on all computers.
- Minimum system requirements: Dual core processor, 2.4 GHz (i5 or i7 Intel processor or AMD equivalent); 4 GB RAM; 100 MB free hard drive space for the application files; 802.11g/n wireless (for laptops; WPA 2 support required); 19-inch LCD monitor (for desktops).

To find, download, and install the app on your PC:

1. In a web browser, navigate to the Robotics Workshop product page on the Thames & Kosmos website.
http://thamesandkosmos.com/index.php/product/category/science-kits/robotics-workshop

2. Click the link to download the CB1 Blockly Desktop application. Download and run the installer.

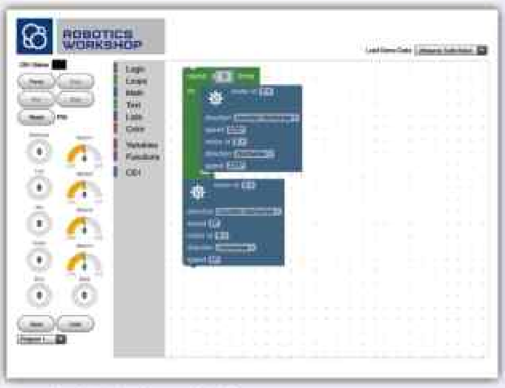
Use the included USB cable to connect the CB1 core controller to the computer.

WRITING PROGRAMS

APP OVERVIEW

CB1 Blockly is a visual programming tool built using the open-source Google Blockly library. It is configured to interact with the CB1 core controller in this kit.

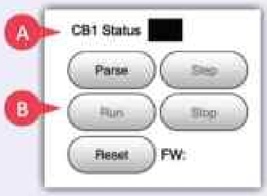
The app uses visual blocks of code that can be easily inserted, moved around, configured, and deleted. The goal is to make it easy to write programs to command the robots you build with this kit, so that you can get them to do what you want them to do.


The main CB1 Blockly app interface

FEATURES OVERVIEW

The main user interface of the CB1 Blockly App for Robotics Workshop is shown above. The primary features are described here.

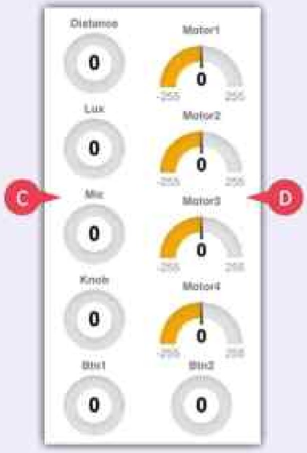
Control Panel Area



A. CB1 status indicator: This changes color when the CB1 core controller is connected to the app, and it blinks when there is activity between the core controller and the app.

B. Program operation buttons:

Parse: This button checks the code and uploads it to the core controller.
Run: This button runs, or executes, the program on the core controller.
Step: This button allows you to run through the code blocks one by one, to see the effects of each one. This is useful when debugging new programs.
Stop: This button stops the entire program from running.
Reset: This button removes the program from the core controller so that a new program can be uploaded.

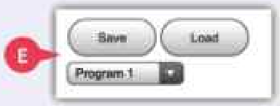


C. Sensor status display: These gauges show the real-time readings from the sensors plugged into the CB1 core controller:

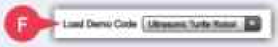
Distance: Ultrasonic sensor input
Lux: Light sensor input
Mic: Sound sensor (microphone) input
Knob: Rotary knob (variable resistor) input
Button 1 and 2: Touch sensor input

D. Motor status display: These gauges show the real-time output of each of the four motors (the speed and the direction).

E. Save and load programs: To save a program, choose one of the programs listed in the menu and then press the save button. To load a previously saved program, choose one of the programs listed in the menu and press the load program. The program will load in the block coding area.



F. Load demo code: To load the demo program for one of the ten robotic models included in this manual, simply tap the menu and select the demo program you want to load in the block coding area.



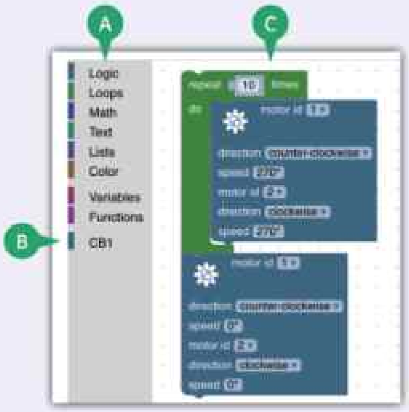
Block Toolbox and Block Coding Area

This is the main coding area where you can assemble programs to control your robots.

A. Block toolbox: Tap one of the category names in the block toolbox to show the code blocks available in that category. This toolbox contains all of the common blocks of code that you will need.

B. CB1 code blocks: This part of the toolbox contains all of the code blocks that were developed specifically for use with the CB1 core controller in this kit.

C. Block coding area: This is where you assemble and configure the code blocks into the active program.



WRITING A PROGRAM

You can modify the existing demo programs or write your own program. Here's how:

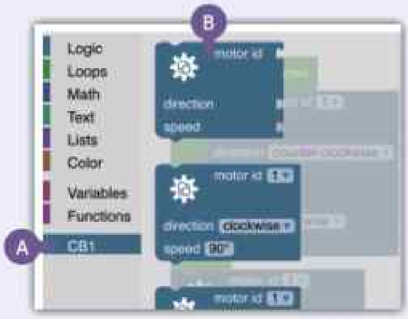
Tap one of the sections in the block toolbox (A). This will open a bar which shows all the functions within that menu (B). Then drag and drop one of the blocks into the center coding area of the app.

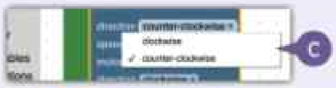
Blocks can be connected and placed within one another to form new blocks of code. Code blocks can also be stacked on top of each other.

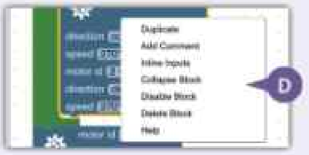
Important: When you run the program, the app will carry out your code from top to bottom, starting at the top.

You can change the variable parameters or values in code blocks by tapping on the menus or fields (C).

You can duplicate, collapse, disable, or delete a code block by tapping and holding on a blank part of the block to open the edit menu (D).

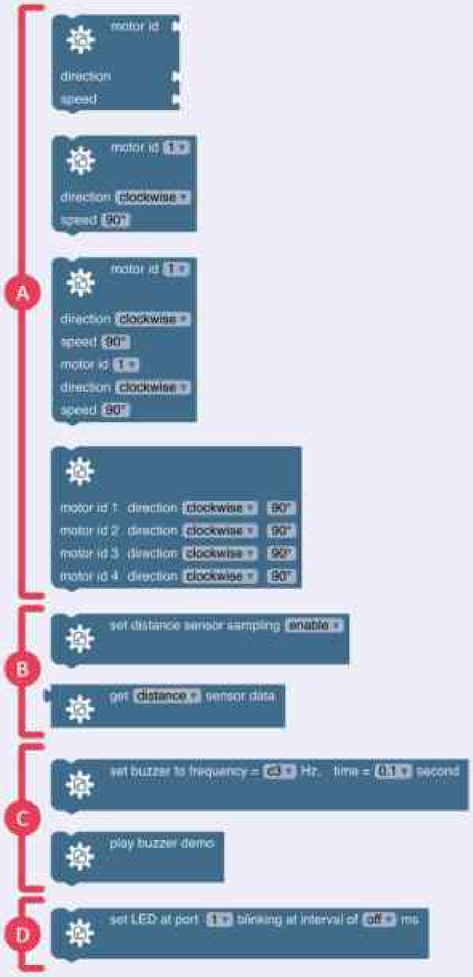






WRITING PROGRAMS

This manual does not explain the function of every block in the Blockly library. The educational intent is for you to learn how many of the blocks function by building the models in this manual, loading the demo code for each model, and experimenting with how the demo code functions with each model. If you have a question about a specific code block, we suggest you look it up online with the help of an adult. Because Blockly is open-source, there is a considerable amount of information available about it online. The CB1 code blocks are explained below.


A
motor id
direction
speed
motor id 1
direction clockwise
speed 90°
motor id 1
direction clockwise
speed 90°
motor id 1
direction clockwise
speed 90°
motor id 1 direction clockwise 90°
motor id 2 direction clockwise 90°
motor id 3 direction clockwise 90°
motor id 4 direction clockwise 90°
B
set distance sensor sampling enable
get distance sensor data
C
set buzzer to frequency = c4 Hz. time = 0.1 second
play buzzer demo
D
set LED at port 1 blinking at interval of off ms

CB1 BLOCKS

A. Motor Blocks

Use the motor blocks to control one to four of the motors. You can select which motor port you want to control with the block, and set the direction and the relative speed (power level) you want the motor to turn.

B. Sensor Blocks

The sensor blocks allow you to use the sensor input data in your programs. With the sensor blocks, you can choose to get sensor data from the ultrasonic sensor (distance), light sensor (luminance), sound sensor (microphone), touch sensors (button 1 and 2), and rotary variable resistor (knob). You can plug the sensor blocks into logic blocks.

C. Buzzer Blocks

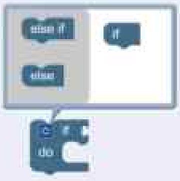
The buzzer blocks allow you to play sounds from the speaker on the CB1 core controller.

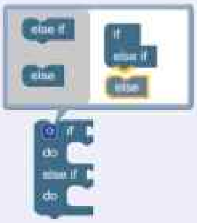
D. LED Blocks

The LED blocks allow you to program the LEDs on the CB1 core controller to light up.

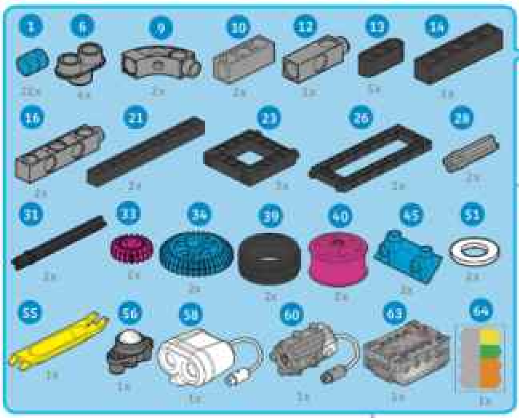
CREATING IF-ELSE STATEMENTS

As you will learn, if-else and if-else-if-else function blocks are important to many programs. To create an if-else block, tap on the small blue gear on an if block. Drag an else-if or else block to the right side of the window to make a new type of block.

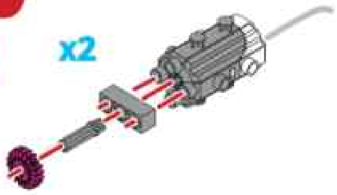




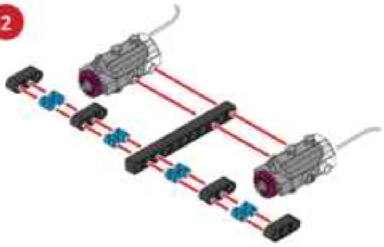
ULTRASONIC TURTLE ROBOT



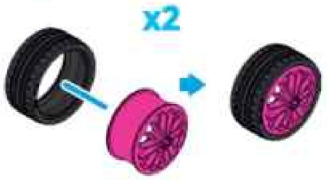
1



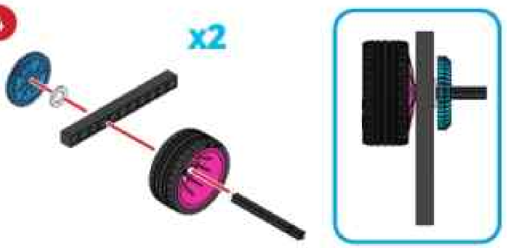
2



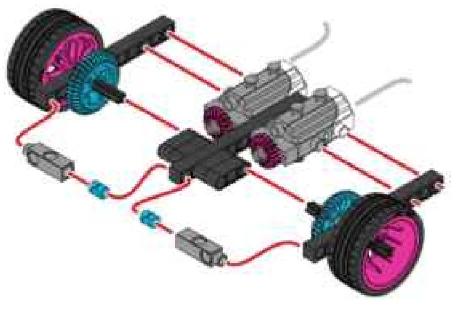
3



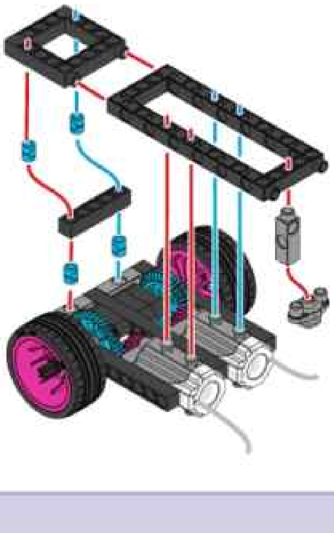
4



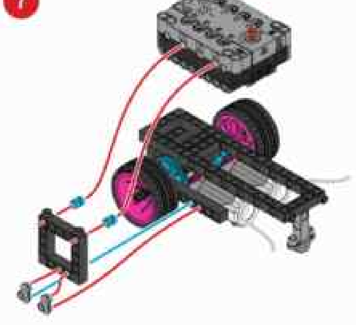
5



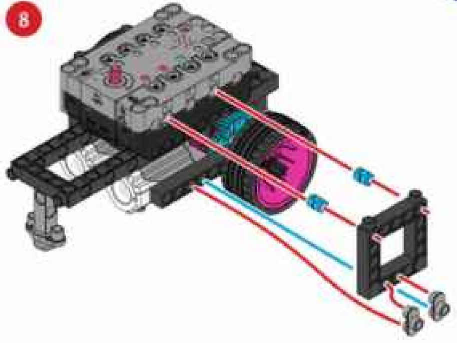
6



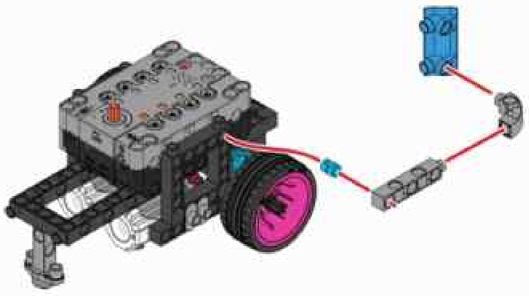
7



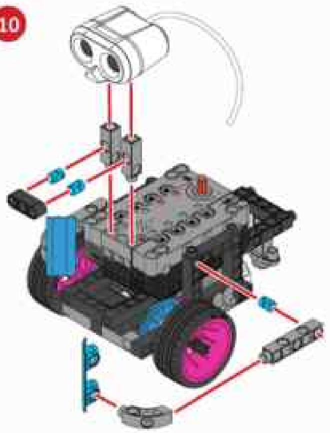
8



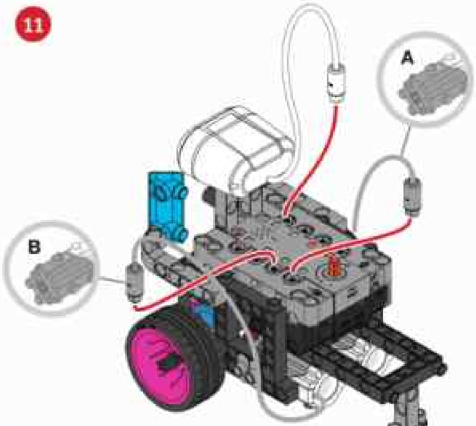
9



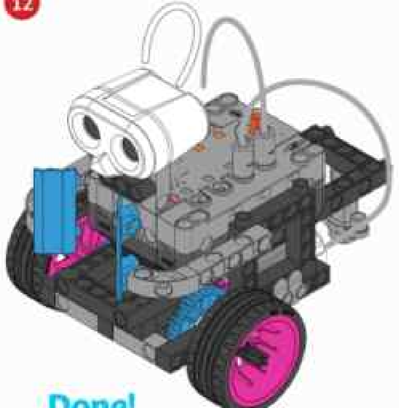
10



11



12



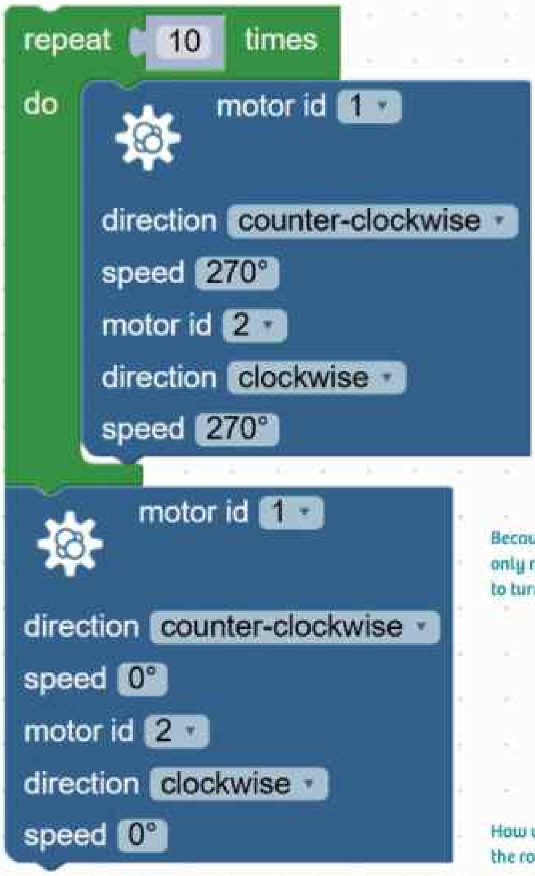
Done!

SAMPLE PROGRAM FOR THE TURTLE ROBOT

Use this program to make the ultrasonic turtle robot walk forward. Build the robot and test out the program. Then try modifying the program so that the turtle robot is able turn left or right.

Try building a simple maze for your turtle robot using objects such as books or paper cups. Then write a program to drive the turtle robot through the maze without bumping into anything.

This program and all of the programs in this manual are preloaded in the app in the Load Demo Code menu.



```
repeat 10 times
do
    motor id 1
    direction counter-clockwise
    speed 270°
    motor id 2
    direction clockwise
    speed 270°
motor id 1
direction counter-clockwise
speed 0°
motor id 2
direction clockwise
speed 0°
```

Loop

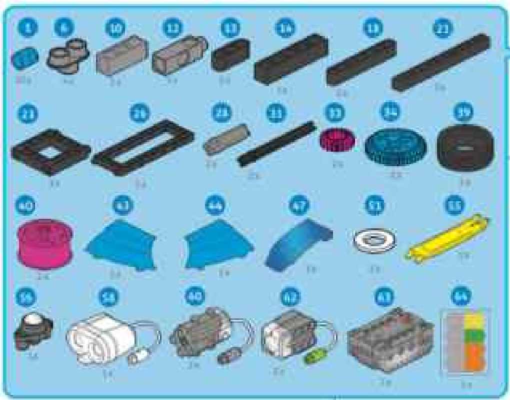
The green block is what is known as a **loop**. A loop in a program repeats the instructions within it over and over again, sometimes infinitely! This loop will repeat the blocks that are within the green bracket 10 times. Loops are used to simplify code so that you do not have to rewrite the same instructions over and over again. Imagine writing these steps 10 times!

This blue block of code tells the robot to go forward by turning on motor 1 and motor 2. The motors need to turn in opposite directions to move the robot forward because of the way that the gears connect to the wheels in the model. These blocks of code are repeated by the loop 10 times.

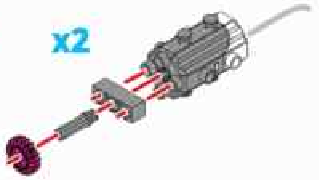
Because this block is outside of the loop, it only repeats once. This block tells the motors to turn off.

How would you modify this program to make the robot turtle turn left or right?

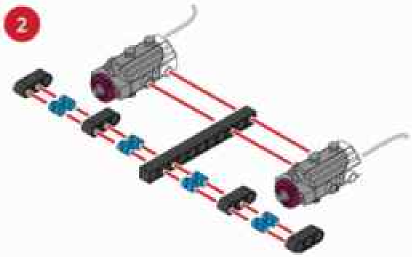
TOUCH SENSOR ROBOT



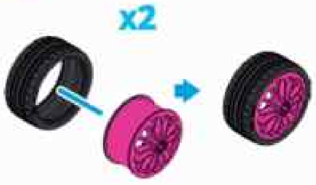
1



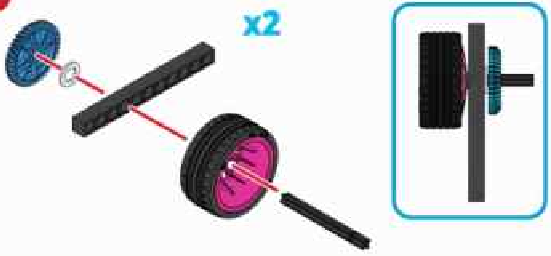
2



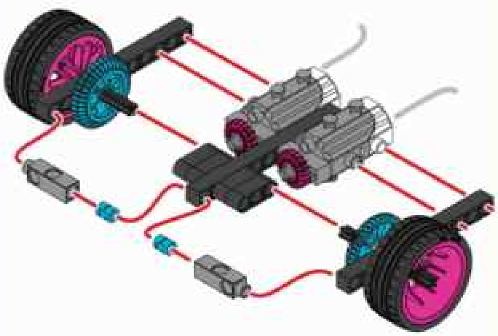
3



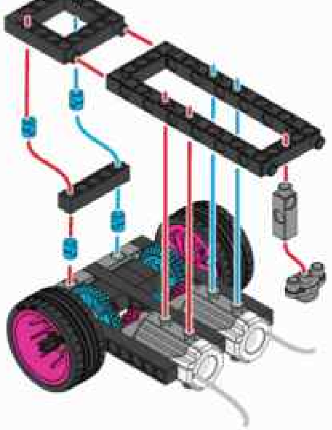
4



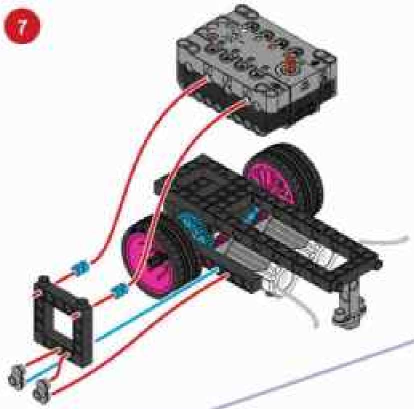
5



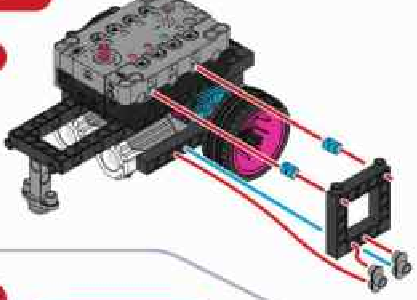
6



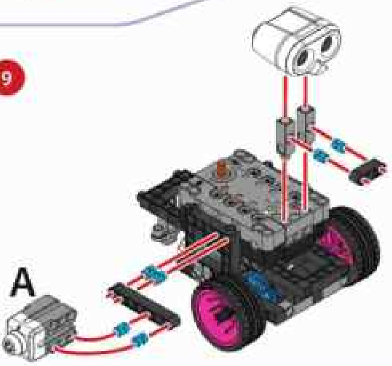
7



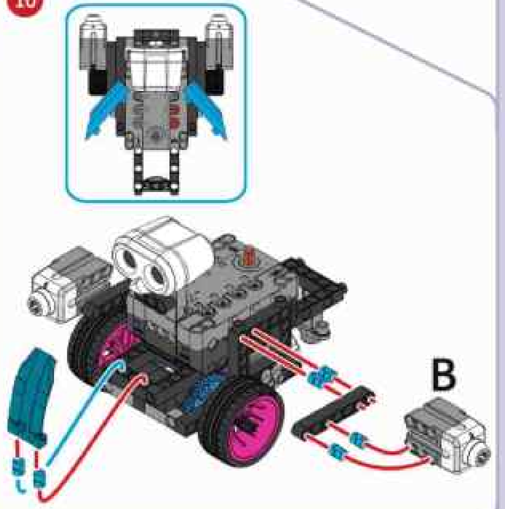
8



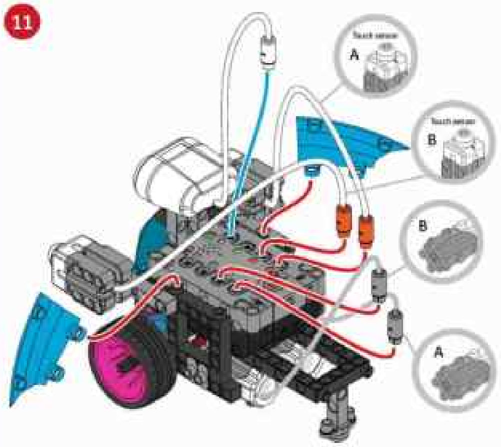
9



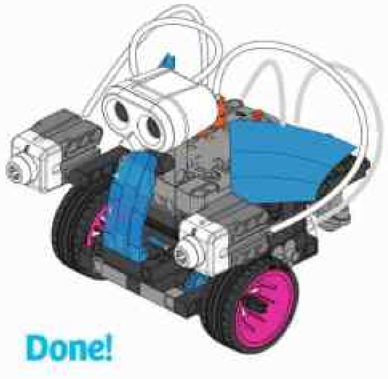
10



11



12



Done!

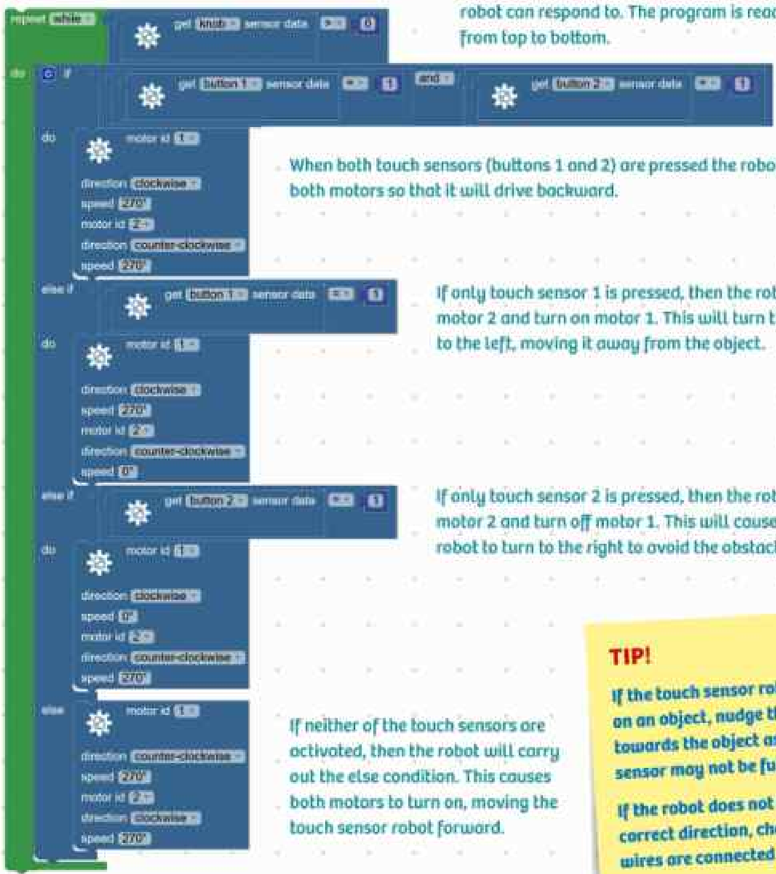
PROGRAMMING

SAMPLE PROGRAM FOR THE TOUCH SENSOR ROBOT

The touch sensor robot has two touch sensors which tell the robot if it has come into contact with an object. This program shows how important sensors are to robots. Point the touch sensor robot toward a flat wall, start the program, and see how the robot responds. This program is preloaded in the app in the Load Demo Code menu.

The green loop is called a **while loop**. This loop will always repeat as long as the knob sensor data is greater than 0. This ensures that the robot is always checking to see if its touch sensors have been pushed.

Inside the green loop is a block called an if-else-if-else statement. This loop has four different conditions that the robot can respond to. The program is read by the robot from top to bottom.



```
repeat while   get knob sensor data > 0
do  if   get button 1 sensor data = 1   and   get button 2 sensor data = 1
    do   motor id 1
         direction clockwise
         speed 270°
         motor id 2
         direction counter-clockwise
         speed 270°
    else if   get button 1 sensor data = 1
    do   motor id 1
         direction clockwise
         speed 270°
         motor id 2
         direction counter-clockwise
         speed 0°
    else if   get button 2 sensor data = 1
    do   motor id 1
         direction clockwise
         speed 0°
         motor id 2
         direction counter-clockwise
         speed 270°
    else motor id 1
         direction counter-clockwise
         speed 270°
         motor id 2
         direction clockwise
         speed 270°
```

When both touch sensors (buttons 1 and 2) are pressed the robot turns on both motors so that it will drive backward.

If only touch sensor 1 is pressed, then the robot will turn off motor 2 and turn on motor 1. This will turn the sensor robot to the left, moving it away from the object.

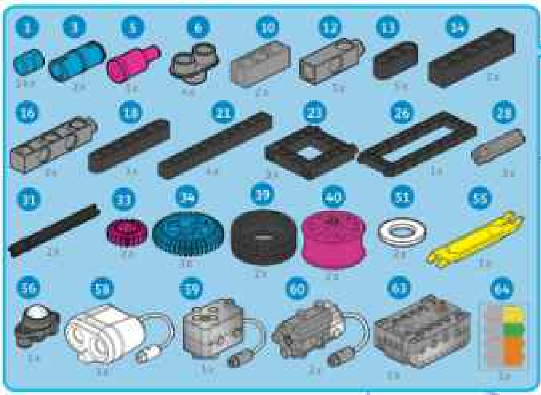
If only touch sensor 2 is pressed, then the robot will turn on motor 2 and turn off motor 1. This will cause the sensor robot to turn to the right to avoid the obstacle.

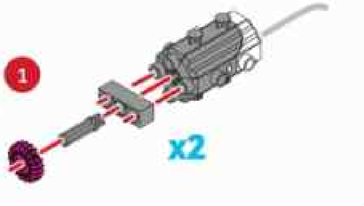
If neither of the touch sensors are activated, then the robot will carry out the else condition. This causes both motors to turn on, moving the touch sensor robot forward.

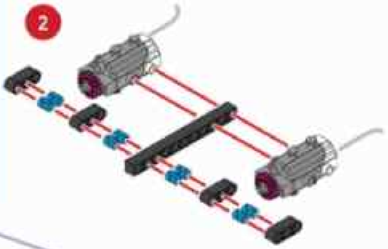
TIP!

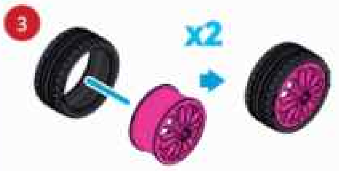
If the touch sensor robot gets stuck on an object, nudge the robot towards the object as the touch sensor may not be fully pushed.

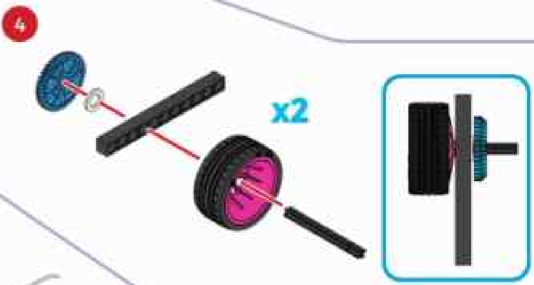
If the robot does not turn in the correct direction, check that the wires are connected correctly.

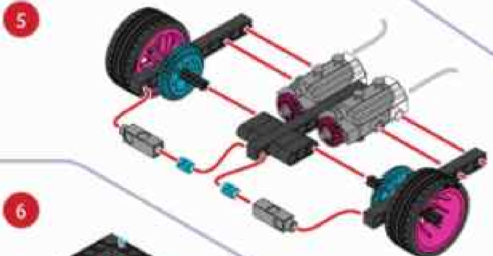


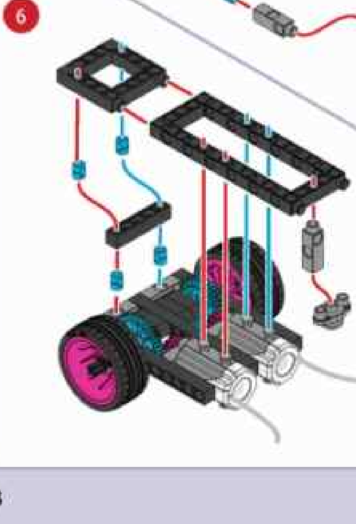

1
x2

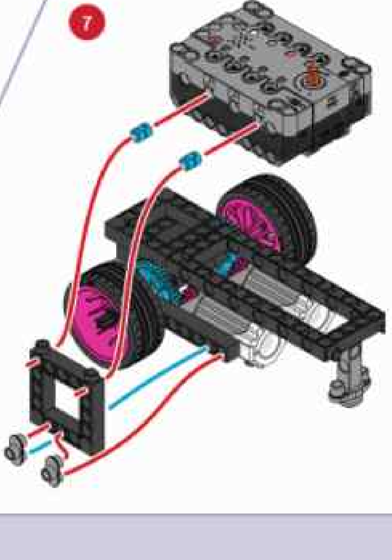

2

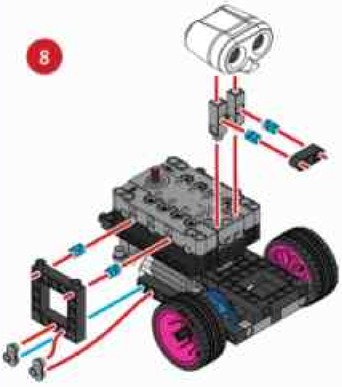

3
x2

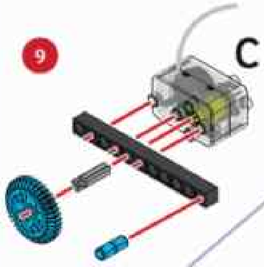

4
x2

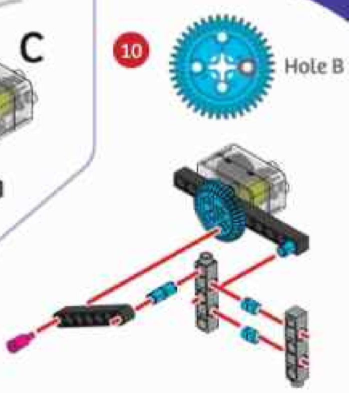

5

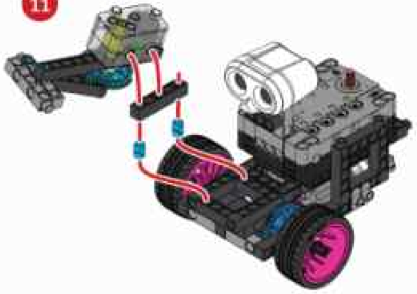

6

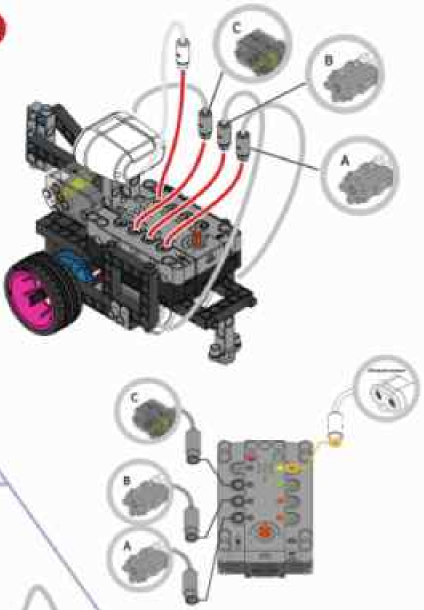

7

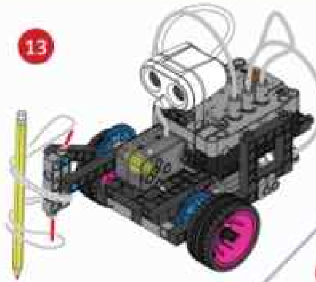
8


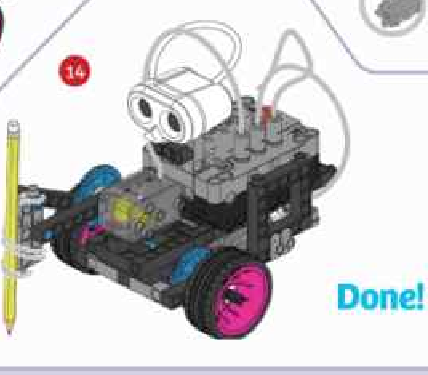
9


10


11


12


13


14


Done!

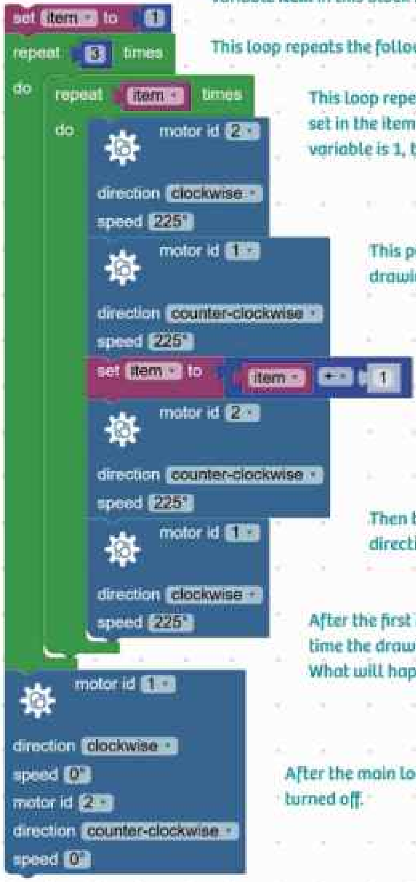
SAMPLE PROGRAM FOR THE DRAWING ROBOT

Arrange four sheets of paper in a large square and tape them down to a smooth floor or tabletop. Attach a pencil or marker to the drawing robot, and place the robot near one edge of the paper with the pencil touching the paper. Test out the program below and see what pattern the drawing robot makes. This program is preloaded in the app in the Load Demo Code menu.

This program shows how loops can be placed or "nested" within other loops.

Variables

Just like loops, **variables** are another important part of coding. A variable contains a value that can change depending on the information that the program provides. Initially the variable **item** in this block is set to a value of 3.



```
set item to 1
repeat 3 times
do  repeat item times
    do  motor id 2
        direction clockwise
        speed 225
        motor id 1
        direction counter-clockwise
        speed 225
        set item to item + 1
        motor id 2
        direction counter-clockwise
        speed 225
        motor id 1
        direction clockwise
        speed 225
motor id 1
direction clockwise
speed 0
motor id 2
direction counter-clockwise
speed 0
```

This loop repeats the following steps 3 times.

This loop repeats the following blocks a number of times equal to the value set in the item variable. The first time the code is run, the value of the item variable is 1, but code will change the item variable later in the program.

This portion of code turns on motors 1 and 2, moving the drawing robot forward.

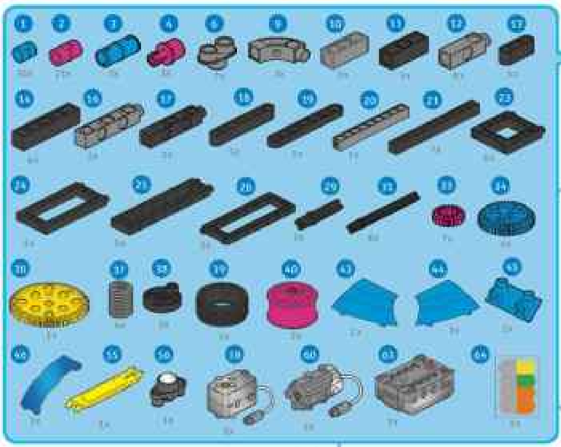
This portion of code adds 1 to the current value of the item variable and saves the new value as the item variable. For example, the first time that the program is run the value of the item variable is increased from 1 to 2.

Then this portion of code turns the motors in the opposite directions, so the robot drives backward.

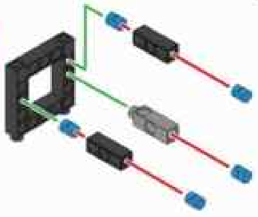
After the first run-through, the loop repeats again, but this time the drawing robot turns the motors on for twice as long. What will happen the third time the loop runs?

After the main loop has repeated 3 times, both motors are turned off.

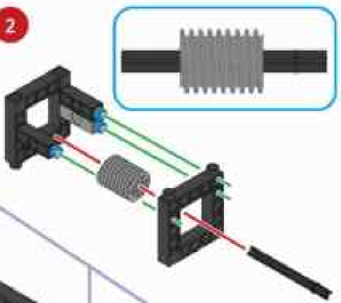
CLAW-ARM ROBOT



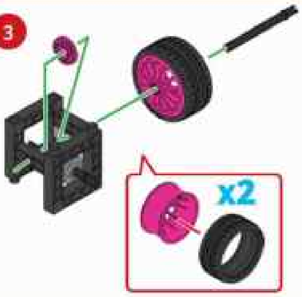
1



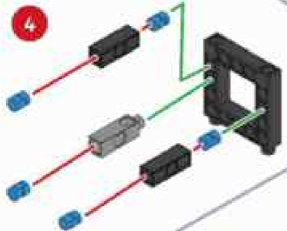
2



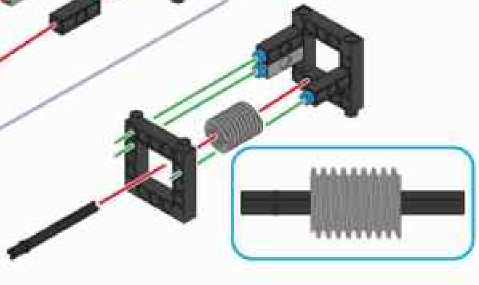
3



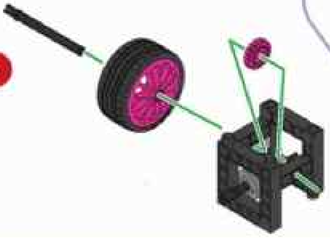
4



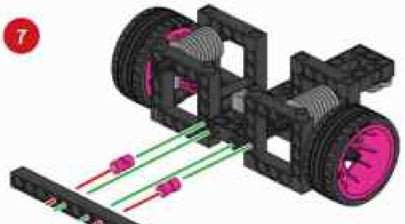
5



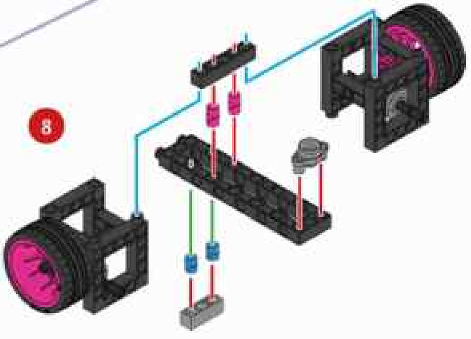
6



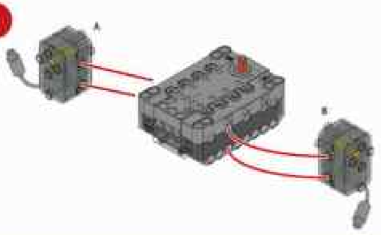
7



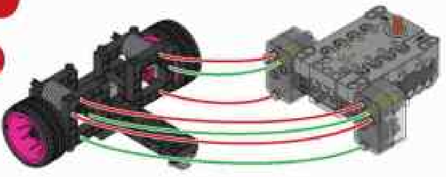
8



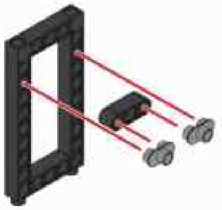
9



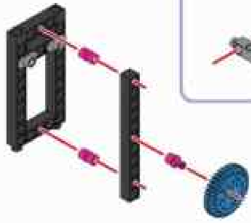
10



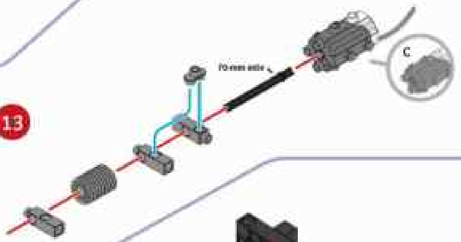
11



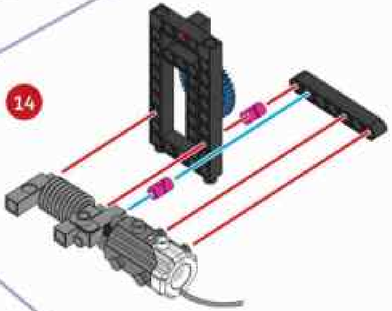
12



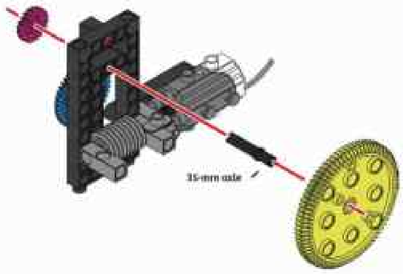
13



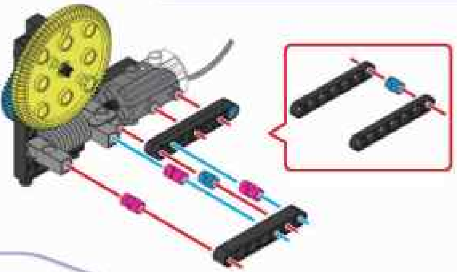
14



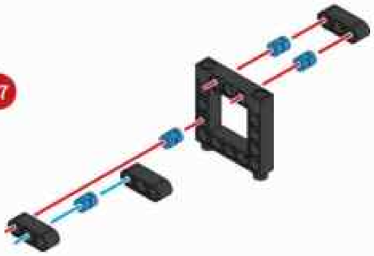
15



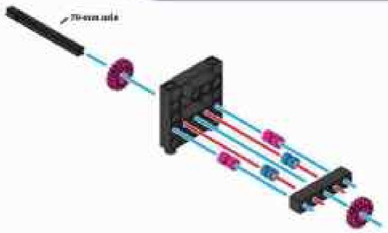
16



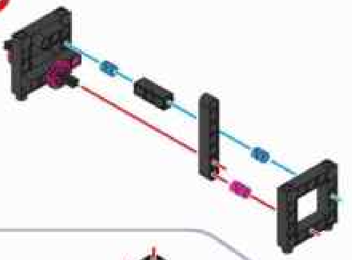
17



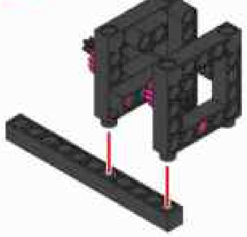
18



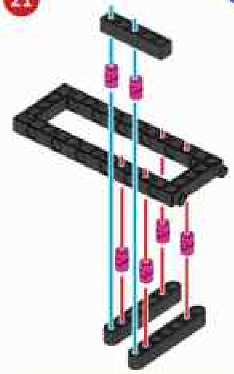
19



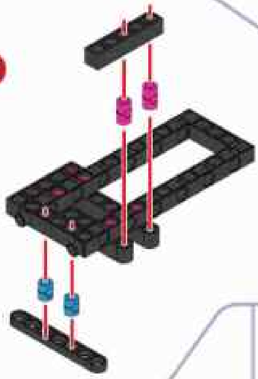
20



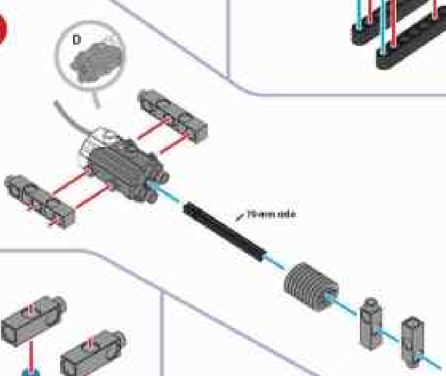
21



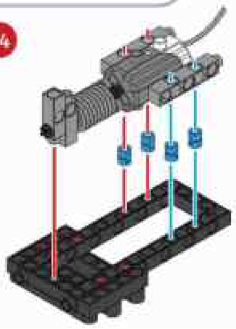
22



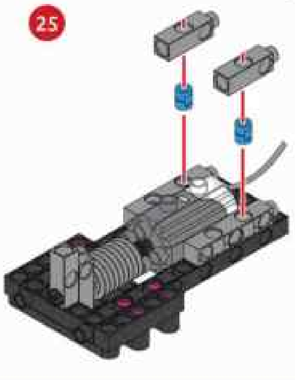
23



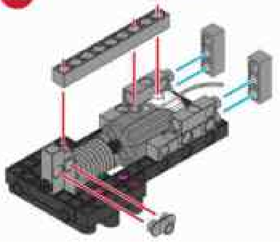
24



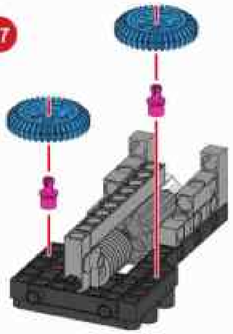
25



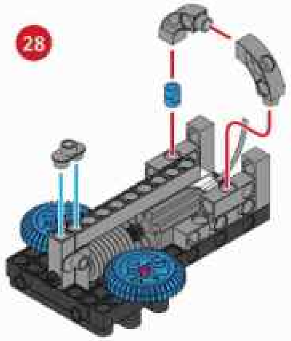
26



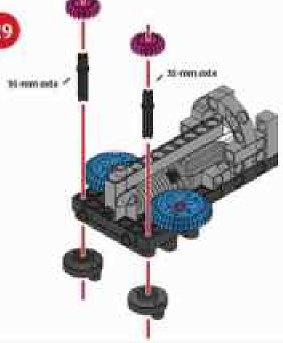
27



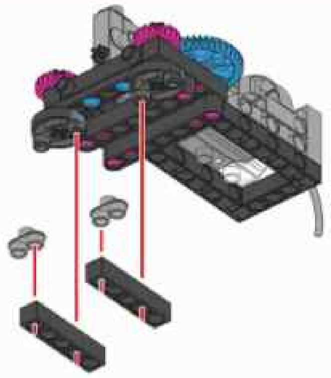
28



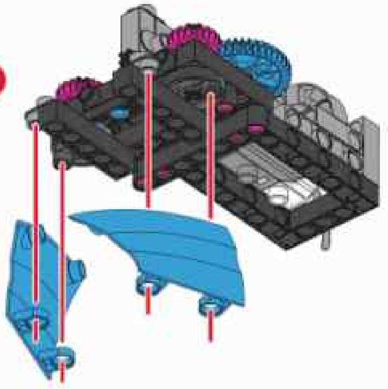
29



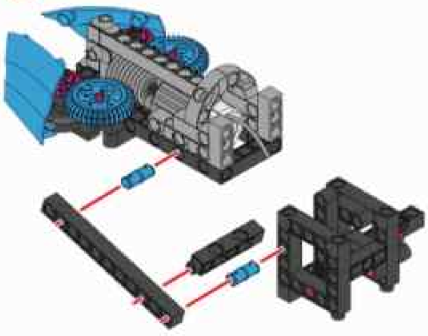
30



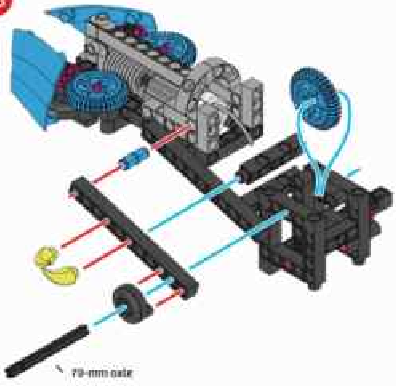
31



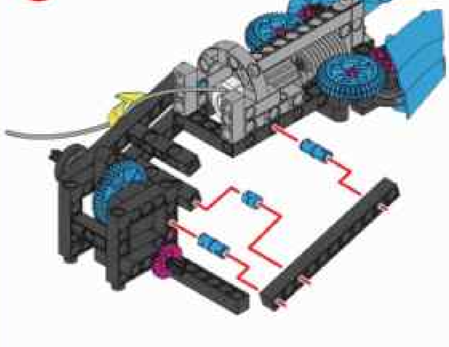
32



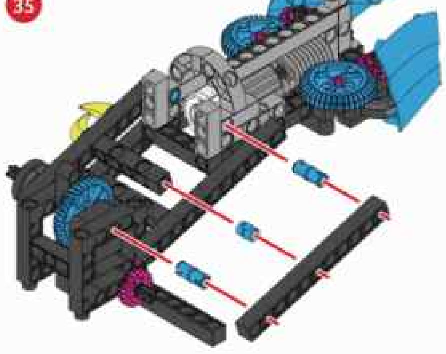
33



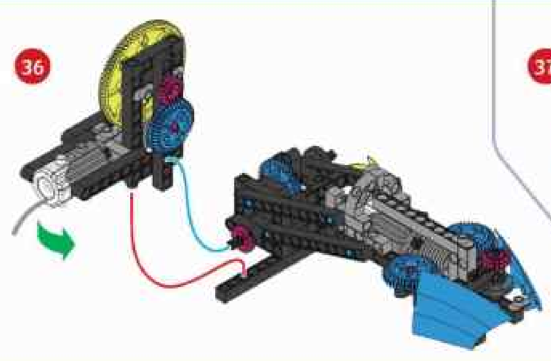
34



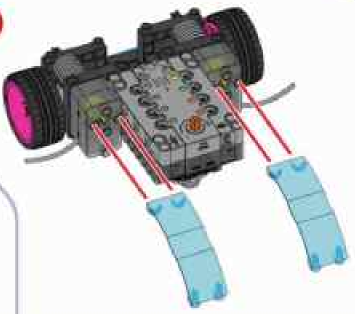
35



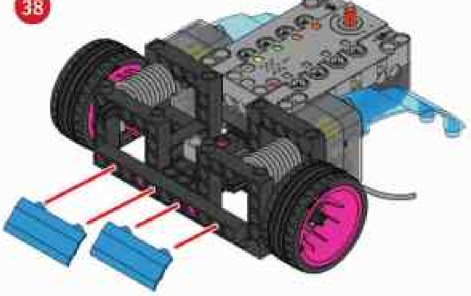
36



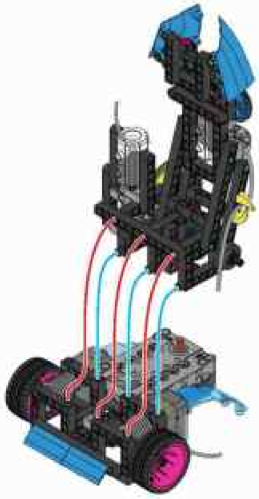
37



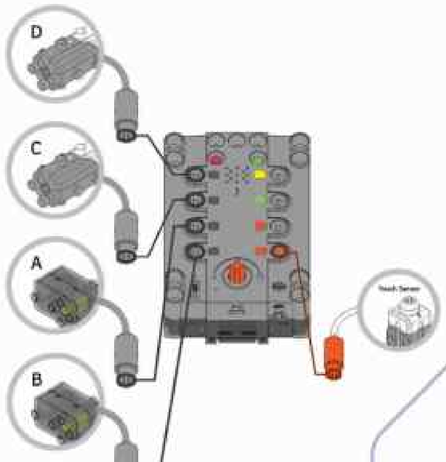
38



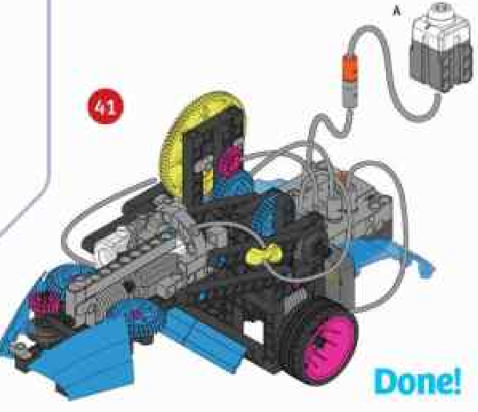
39



40



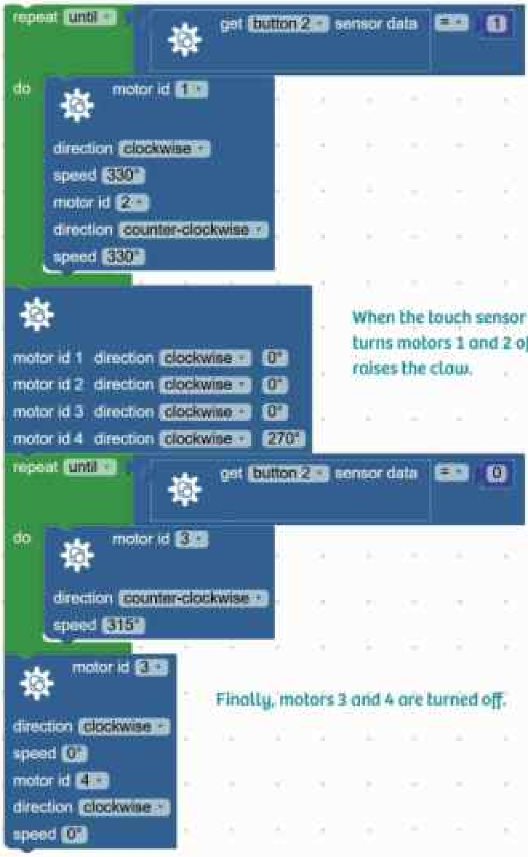
41



Done!

SAMPLE PROGRAM FOR THE CLAW-ARM ROBOT

When this program is run, the claw-arm robot moves straight toward an object, such as a ball, until the touch sensor is pressed. When the robot is close enough to the object, press the touch sensor. The robot will stop and then raise and open its claw around the object. This program is preloaded in the app in the Load Demo Code menu.



```
repeat until  get button 2 sensor data = 1
do   motor id 1
     direction clockwise
     speed 330°
     motor id 2
     direction counter-clockwise
     speed 330°

motor id 1  direction clockwise  0°
motor id 2  direction clockwise  0°
motor id 3  direction clockwise  0°
motor id 4  direction clockwise  270°

repeat until  get button 2 sensor data = 0
do   motor id 3
     direction counter-clockwise
     speed 315°

motor id 3
direction clockwise
speed 0°
motor id 4
direction clockwise
speed 0°
```

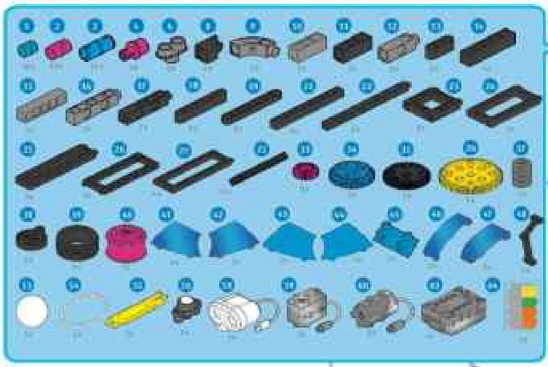
This program contains two loops. The first loop repeats until the touch sensor is pressed. Before it is pressed, motors 1 and 2 will turn, moving the robot forward.

When the touch sensor is pressed, this block of code turns motors 1 and 2 off, and turns motor 4 on, which raises the claw.

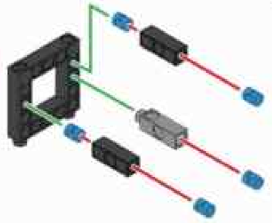
This block of code then repeats until the touch sensor is released. Motor 3 opens the claw of the robot.

Finally, motors 3 and 4 are turned off.

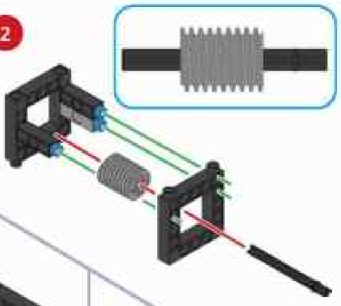
BALL-SHOOTING ROBOT



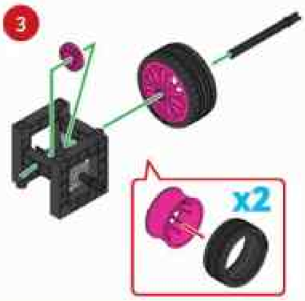
1



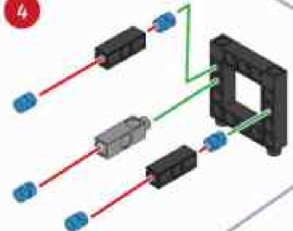
2



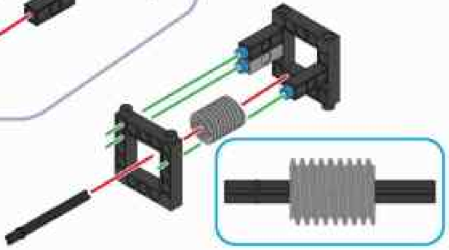
3



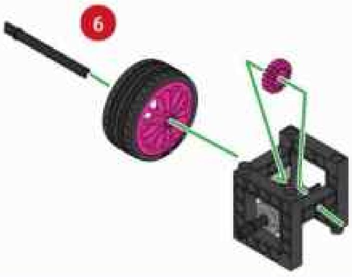
4



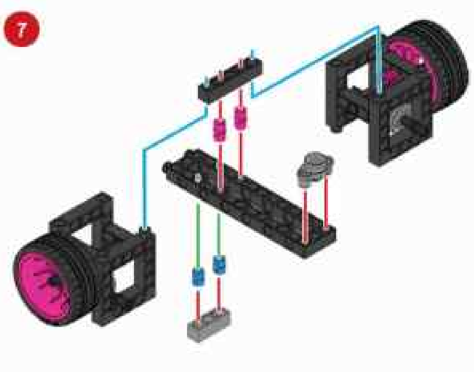
5



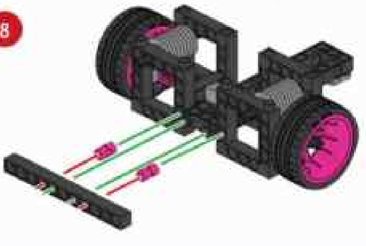
6

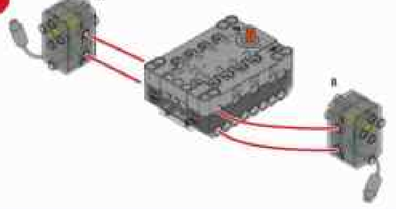


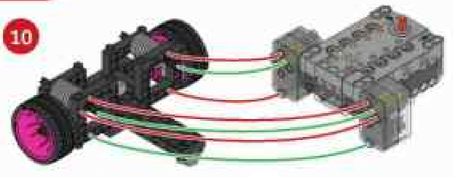
7

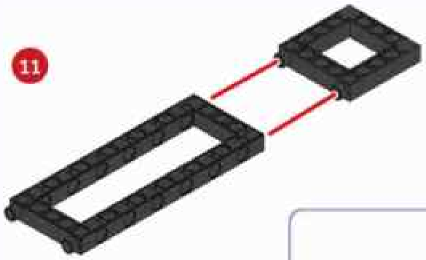


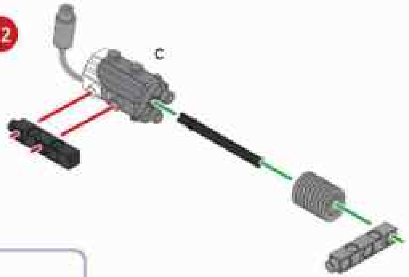
8

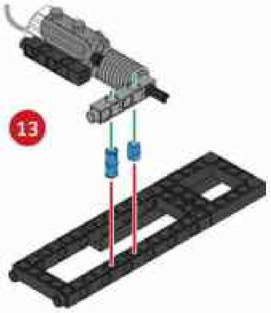


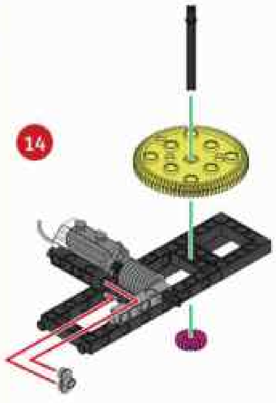
9


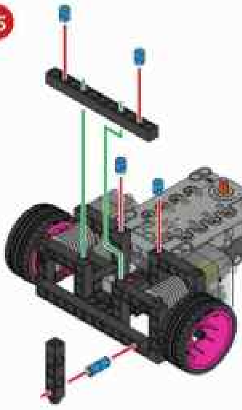
10


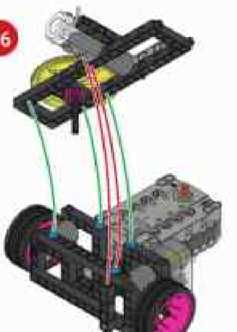
11


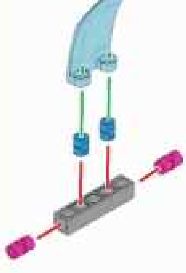
12


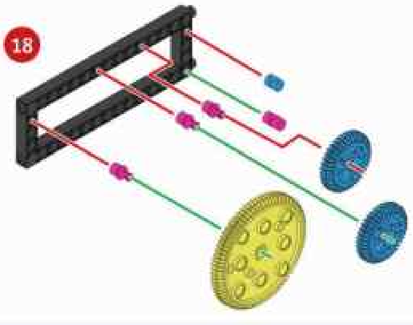
13


14


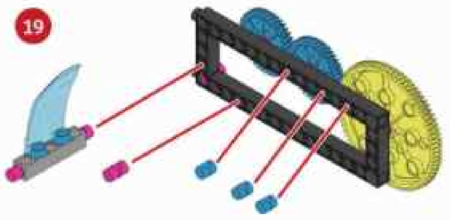
15


16


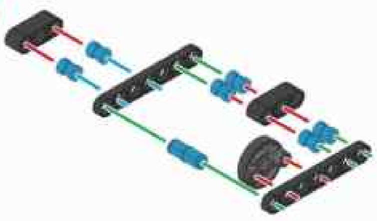
17


18


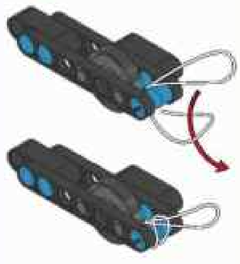
19



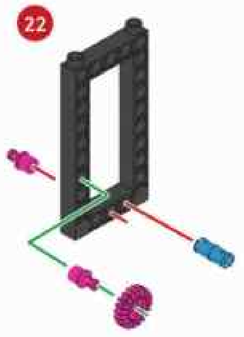
20



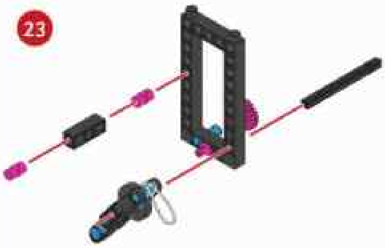
21



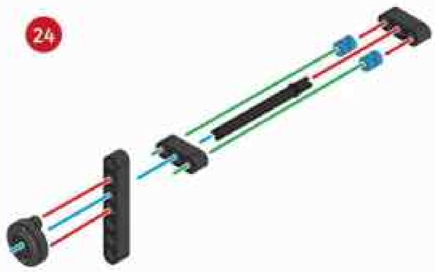
22



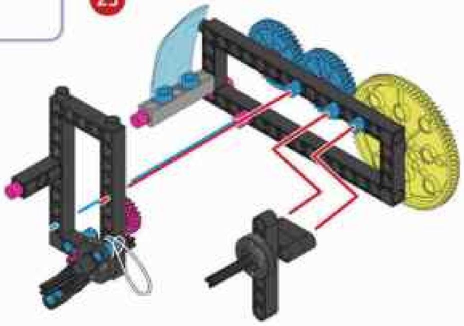
23



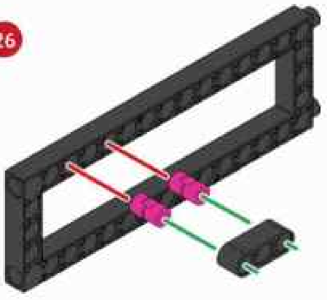
24



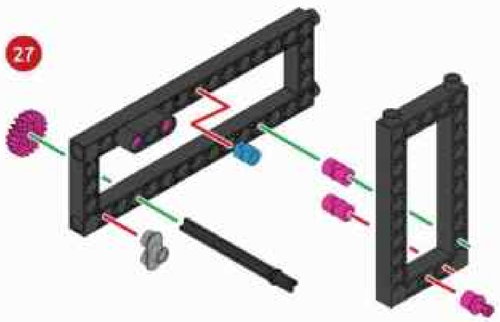
25



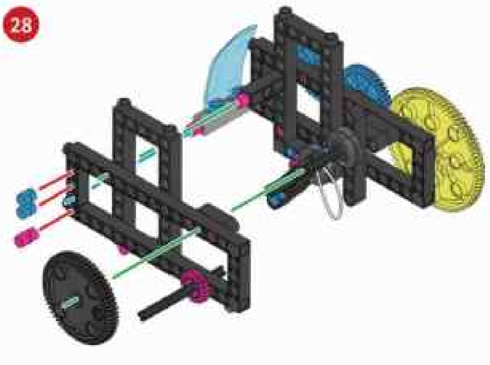
26



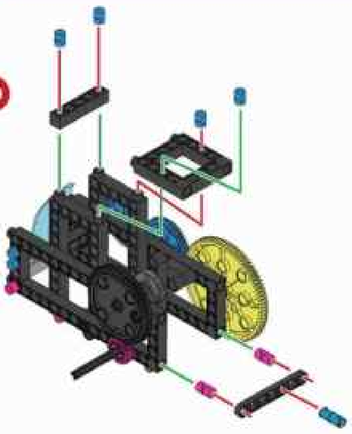
27



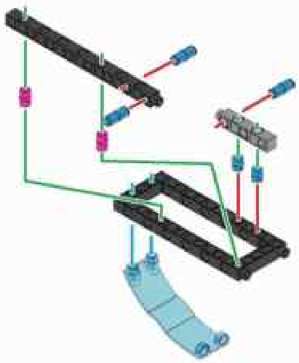
28



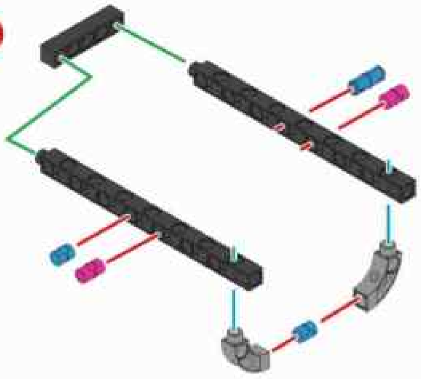
29



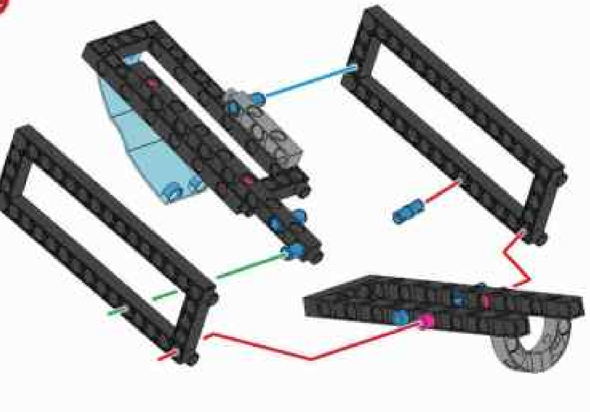
30



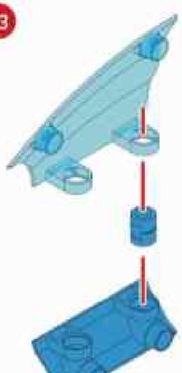
31



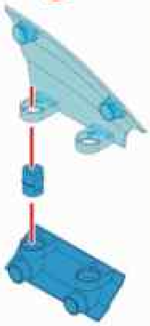
32



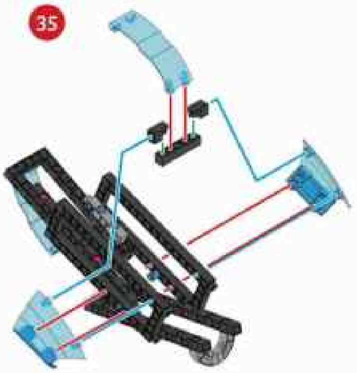
33



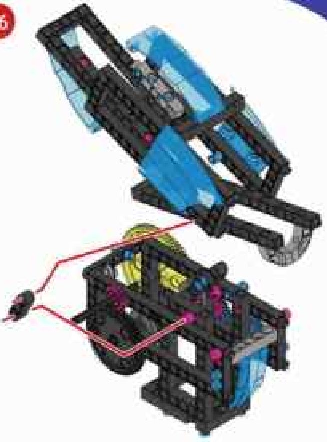
34



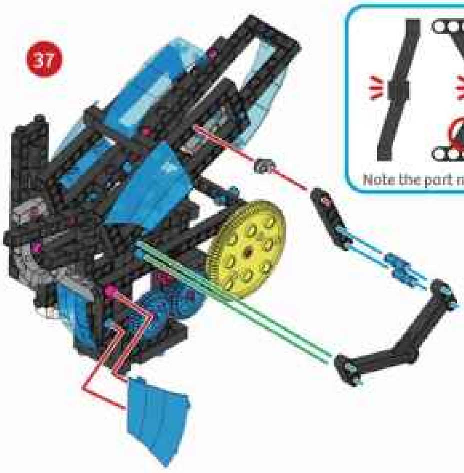
35

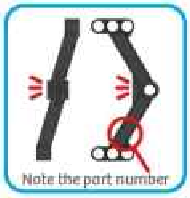


36

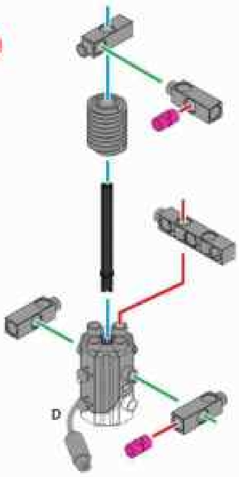


37

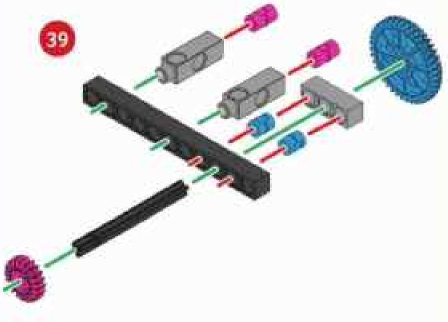




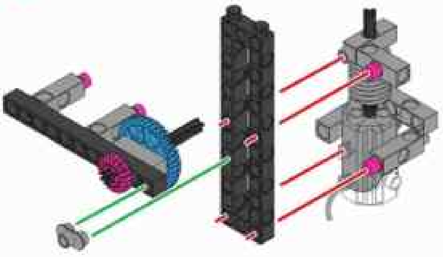
38



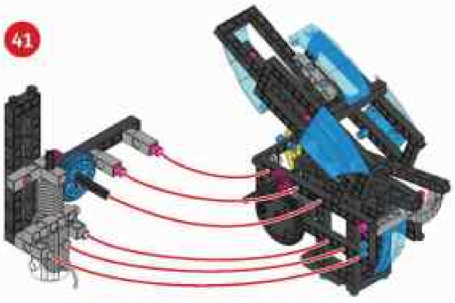
39



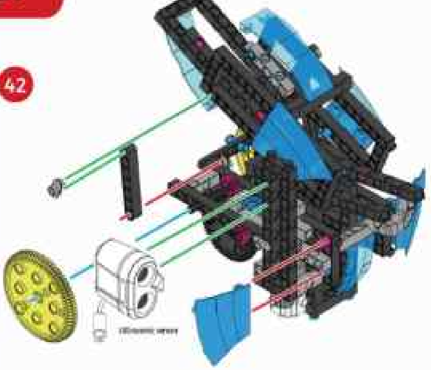
40



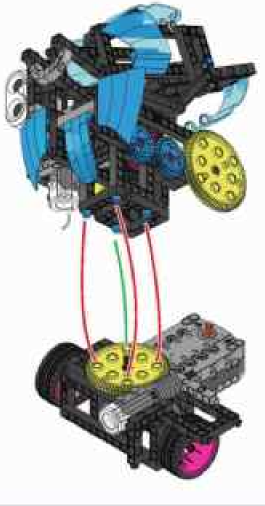
41



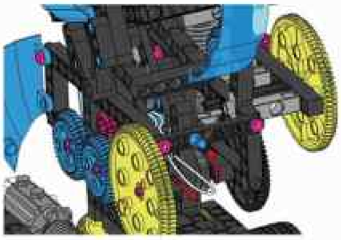
42



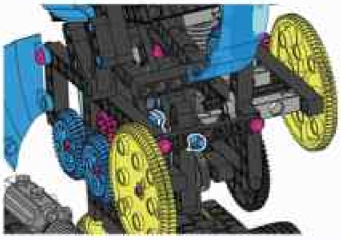
43



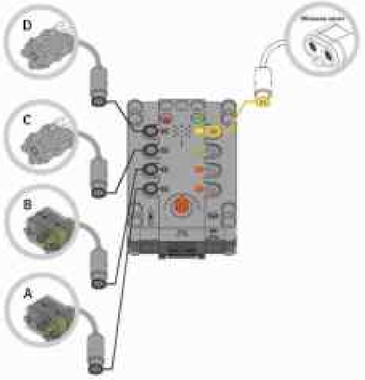
44



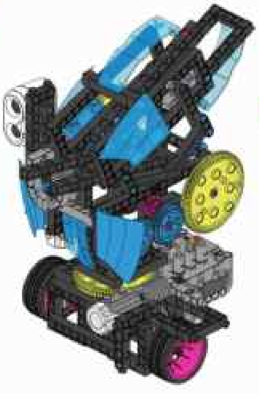
45



46



47

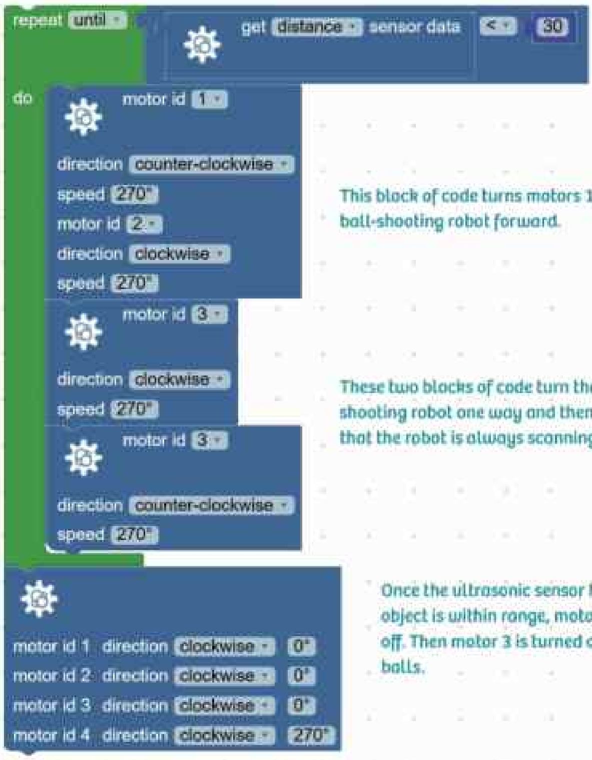


Done!

PROGRAMMING

SAMPLE PROGRAM FOR THE BALL-SHOOTING ROBOT

Load the polystyrene foam balls into the ball-shooting robot. Then place an object such as a cereal box a few feet in front of the robot. Run this program and the ball-shooting robot will move forward toward the object. Once the ultrasonic sensor has detected that the object is within range, the robot will shoot the balls at it. This program is preloaded in the app in the Load Demo Code menu.



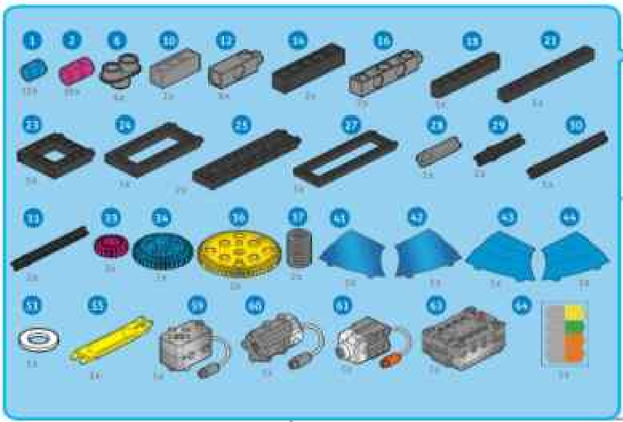
```
repeat until  get distance sensor data  <  30
do
    motor id 1
    direction counter-clockwise
    speed 270°
    motor id 2
    direction clockwise
    speed 270°

    motor id 3
    direction clockwise
    speed 270°

    motor id 3
    direction counter-clockwise
    speed 270°

motor id 1  direction clockwise  0°
motor id 2  direction clockwise  0°
motor id 3  direction clockwise  0°
motor id 4  direction clockwise  270°
```

This portion of code is placed within a loop so that the robot keeps checking whether the value from the ultrasonic sensor (distance) is less than 30. When that value is less than 30, the loop ends.

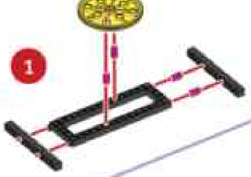
This block of code turns motors 1 and 2 on, which move the ball-shooting robot forward.

These two blocks of code turn the upper half of the ball-shooting robot one way and then the other. This makes sure that the robot is always scanning its field of view.

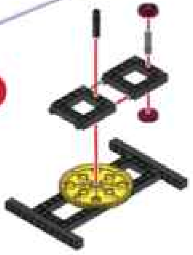
Once the ultrasonic sensor has detected that the object is within range, motors 1 and 2 are turned off. Then motor 3 is turned on, which shoots the balls.


1
2
6
10
12
14
16
19
21
23
24
25
27
28
29
30
33
33
34
36
37
41
42
43
44
51
55
59
60
61
63
64

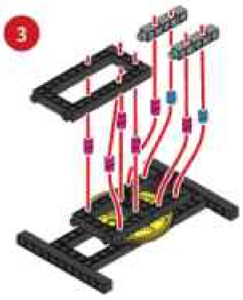
1



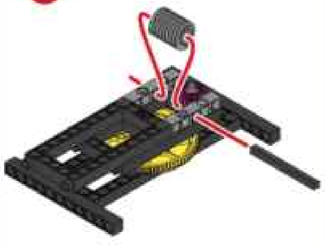
2



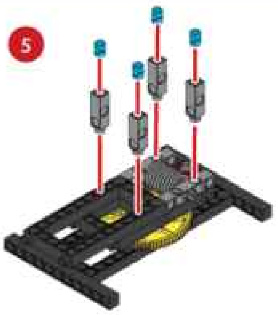
3



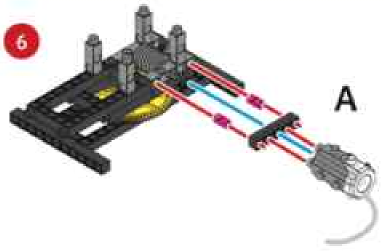
4



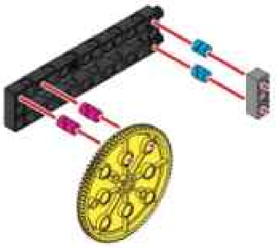
5



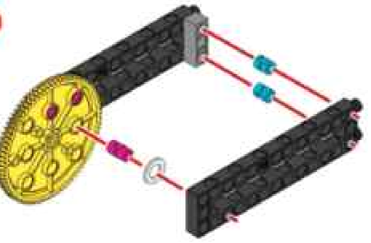
6


A

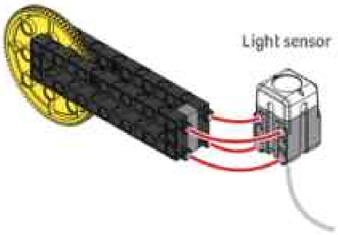
7

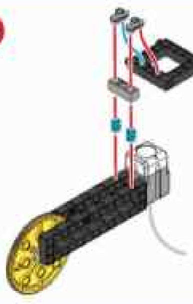


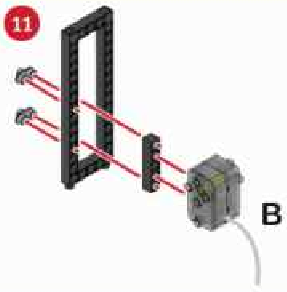
8

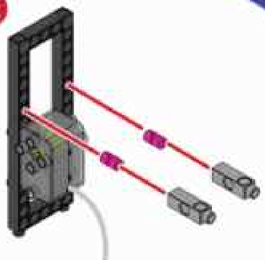


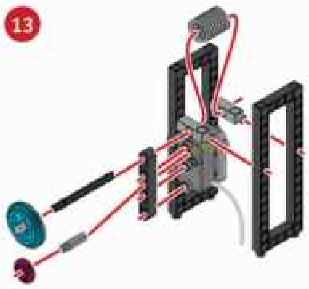
9

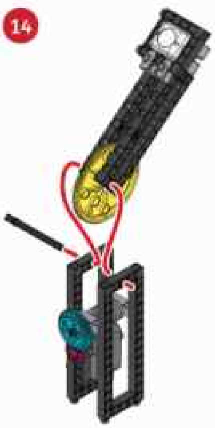

Light sensor

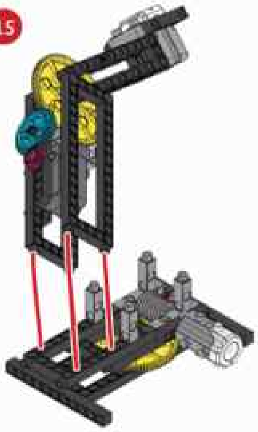
10


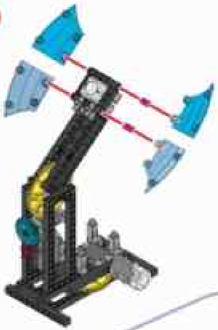
11


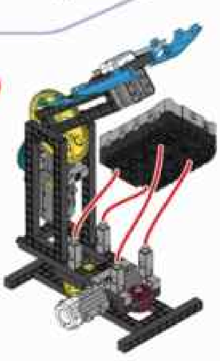
12


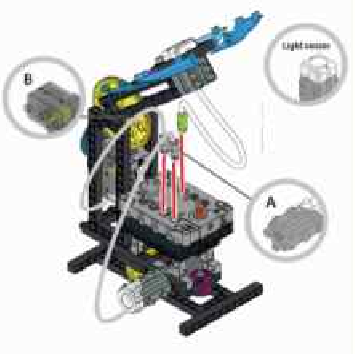
13


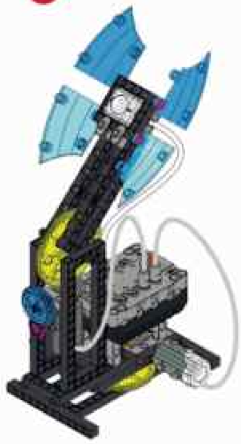
14


15


16


17


18


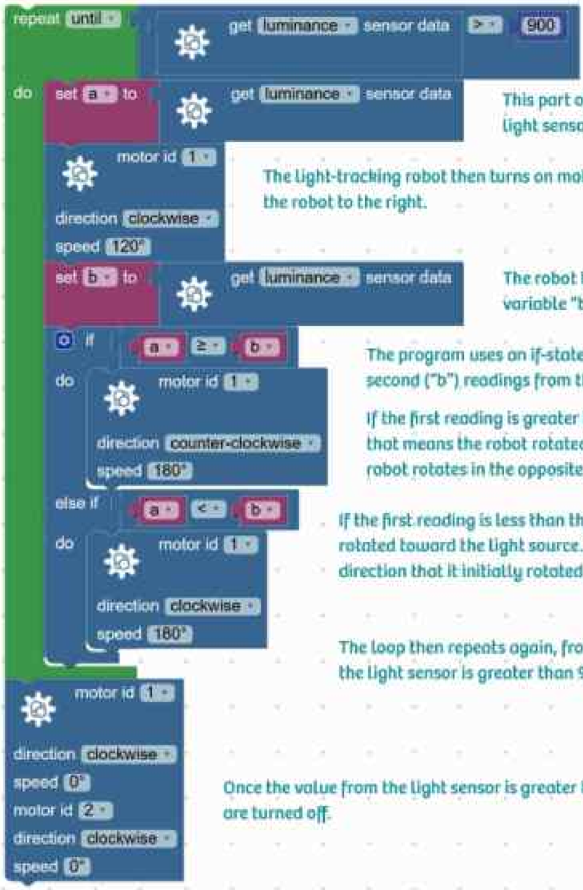
19


Done!

SAMPLE PROGRAM FOR THE LIGHT-TRACKING ROBOT

The light-tracking robot uses a light sensor and motor to rotate towards a light source such as a flashlight. Test out the program by holding a light source a few inches to the right or left of the light sensor. Watch as the robot rotates toward the light source then stops once it is pointing toward it. This program is preloaded in the app in the Load Demo Code menu. After testing out this program, try modifying it so that the light-tracking robot rotates away from the source of light.

The first part of the code is placed in a loop so that the light-tracking robot is always checking to see if the reading from the light sensor is greater than 900. Once that value is greater than 900, the program exits the loop.



```
repeat until  get luminance sensor data  >  900
do  set a to  get luminance sensor data
    motor id 1
    direction clockwise
    speed 120°
    set b to  get luminance sensor data
    if  a ≥ b
    do  motor id 1
        direction counter-clockwise
        speed 180°
    else if  a < b
    do  motor id 1
        direction clockwise
        speed 180°
motor id 1
direction clockwise
speed 0°
motor id 2
direction clockwise
speed 0°
```

This part of the code saves the initial value from the light sensor as the variable "a."

The light-tracking robot then turns on motor 1, rotating the robot to the right.

The robot then saves the second value as the variable "b."

The program uses an if-statement to compare the first ("a") and second ("b") readings from the light sensor.

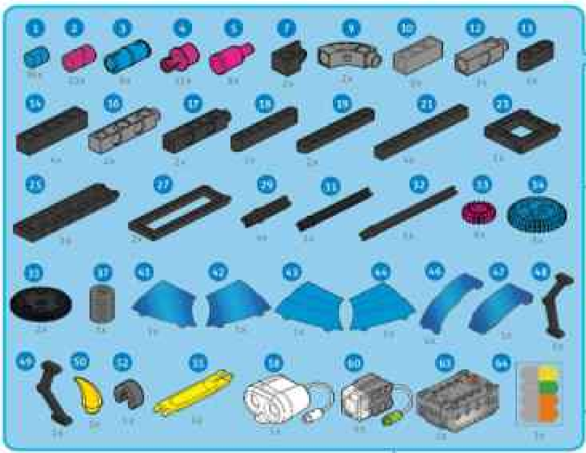
If the first reading is greater than or equal to the second reading, that means the robot rotated away from the light source. So, the robot rotates in the opposite direction (counterclockwise).

If the first reading is less than the second reading, that means the robot rotated toward the light source. So, the robot rotates again in the same direction that it initially rotated (clockwise).

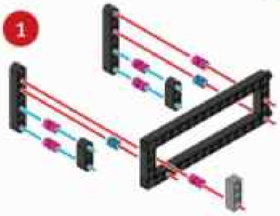
The loop then repeats again, from the beginning, until the value from the light sensor is greater than 900.

Once the value from the light sensor is greater than 900, motors 1 and 2 are turned off.

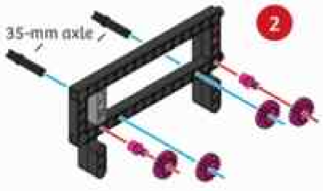
ULTRASONIC WALKING ROBOT



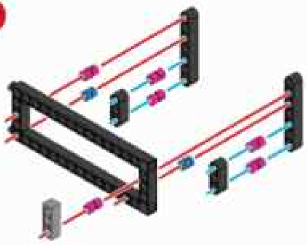
1



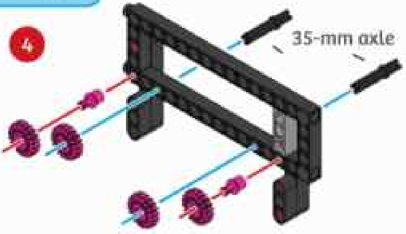
2



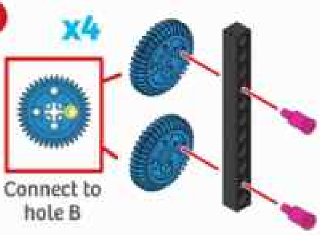
3



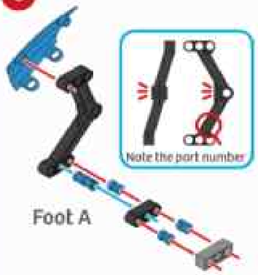
4



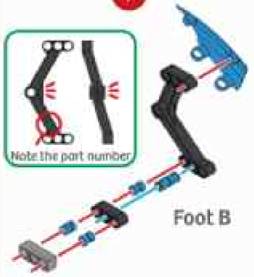
5



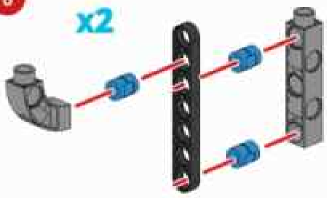
6



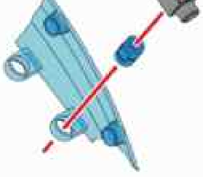
7



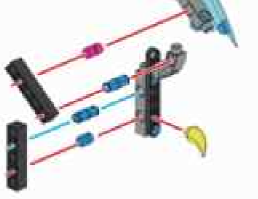
8



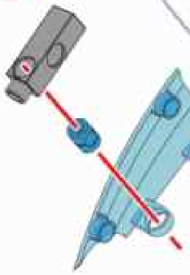
9



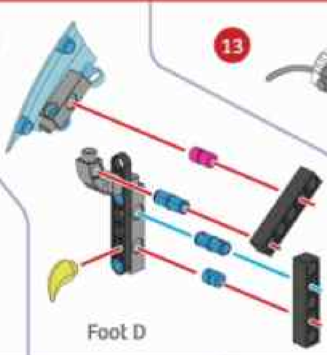
10



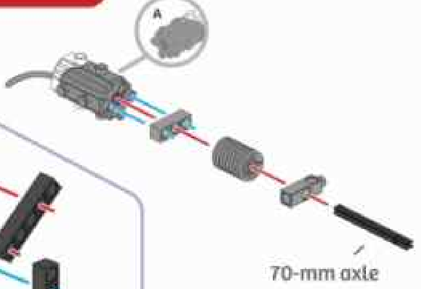
11



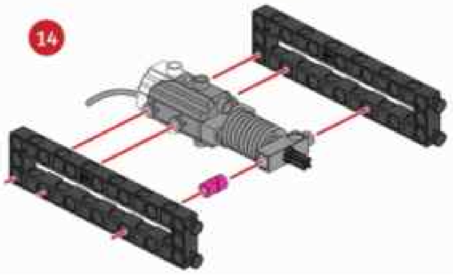
12



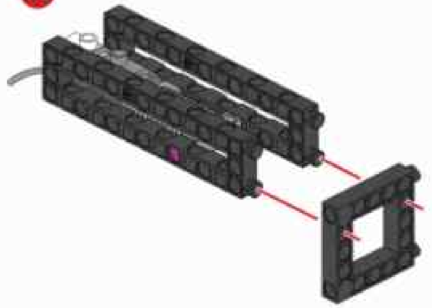
13



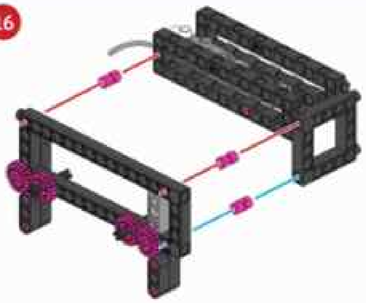
14



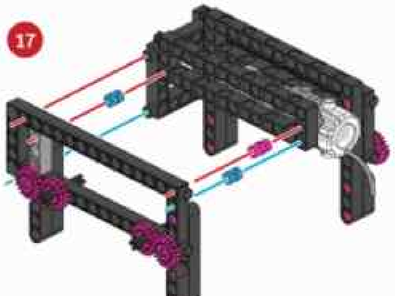
15



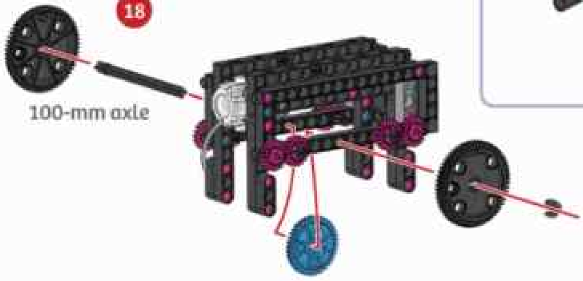
16



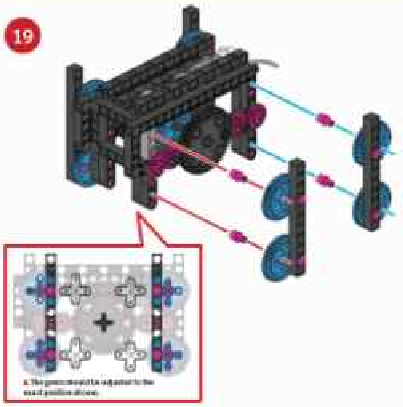
17



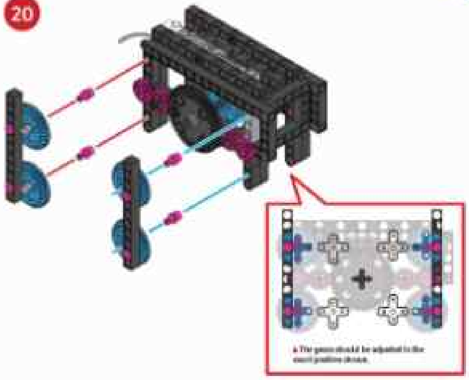
18



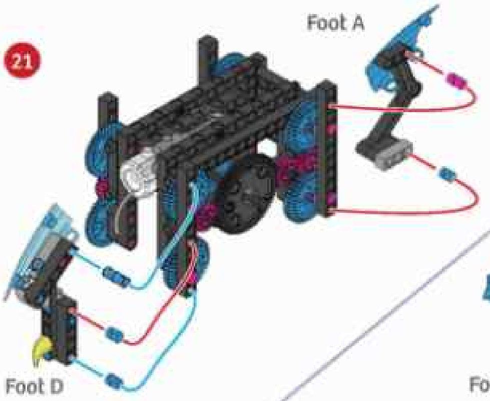
19



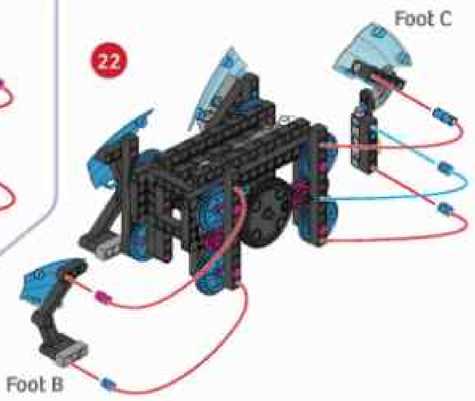
20



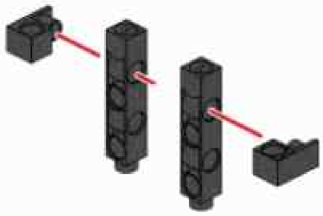
21



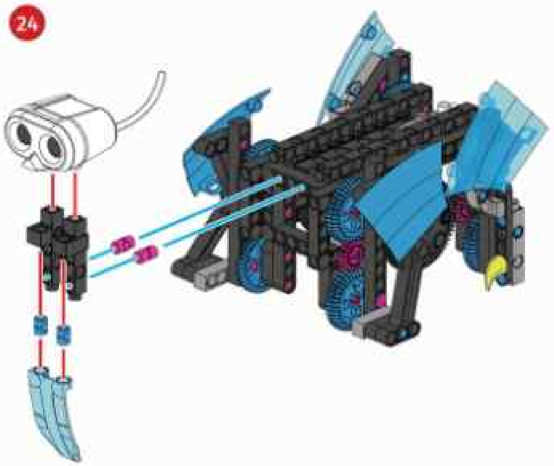
22



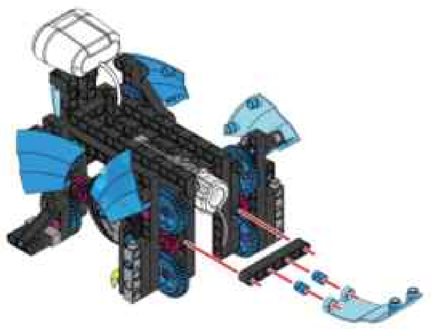
23



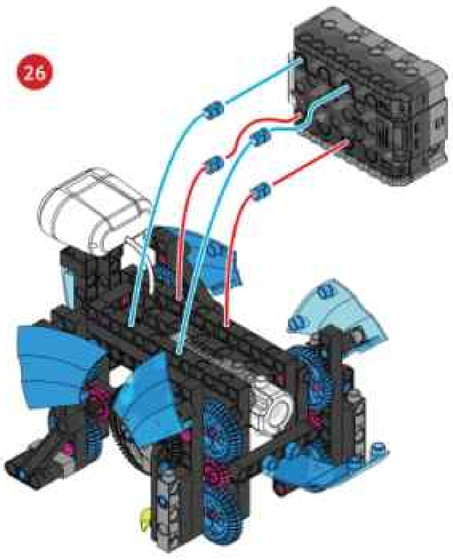
24



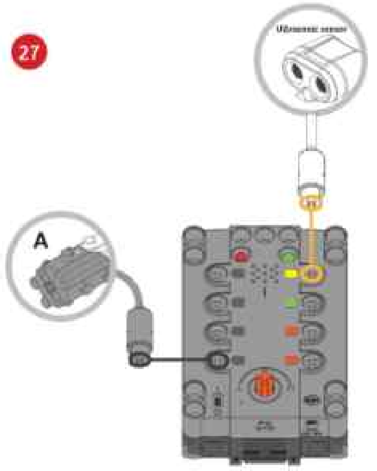
25



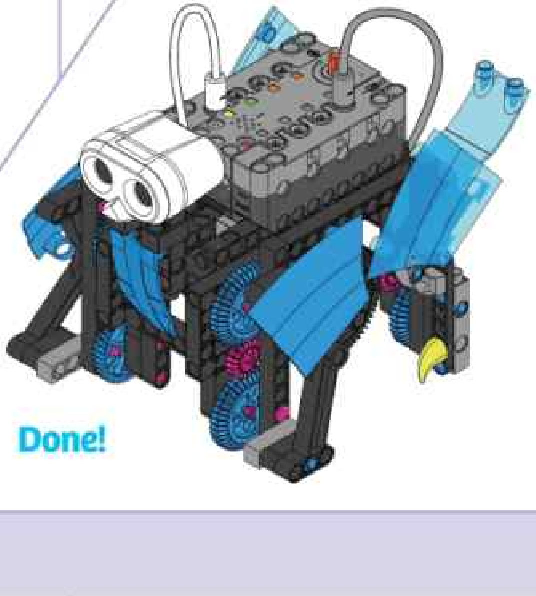
26



27



28

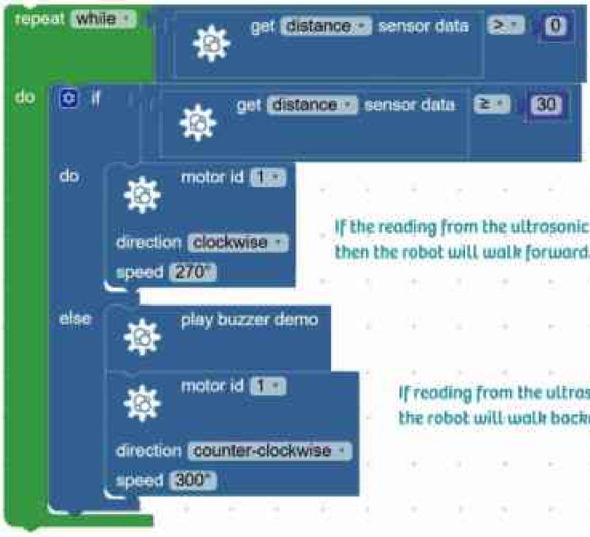


Done!

PROGRAMMING

SAMPLE PROGRAM FOR THE WALKING ROBOT

The ultrasonic walking robot uses a motor to walk on four legs and the ultrasonic sensor to detect objects. This program is preloaded in the app in the Load Demo Code menu.



```
repeat while  get distance sensor data  >  0
do
    if  get distance sensor data  ≥  30
    do
        motor id 1
        direction clockwise
        speed 270°
    else
        play buzzer demo
        motor id 1
        direction counter-clockwise
        speed 300°
```

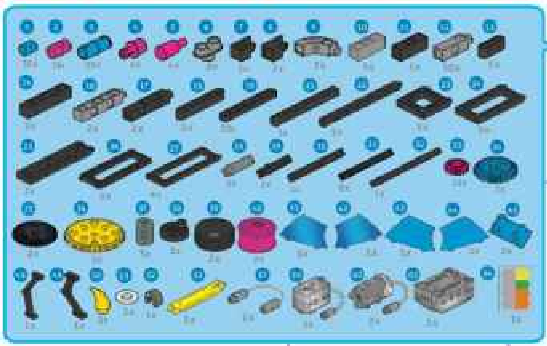
The program is placed in a loop which will turn the robot on if the reading from the ultrasonic sensor is greater than 0.

Within the loop there is an if-else statement, which checks whether the reading from the ultrasonic sensor is greater than or equal to 30.

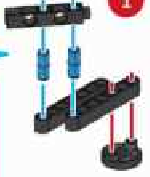
If the reading from the ultrasonic sensor is greater than 30, then the robot will walk forward.

If reading from the ultrasonic sensor is less than 30, then the robot will walk backward.

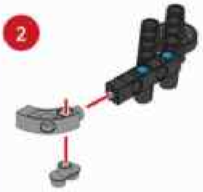
ROBOTIC ARM



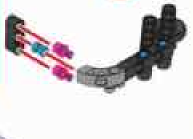
1



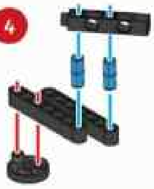
2



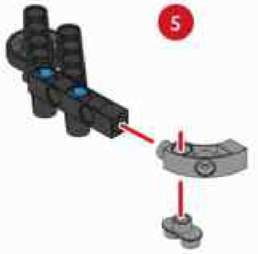
3



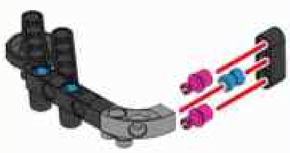
4



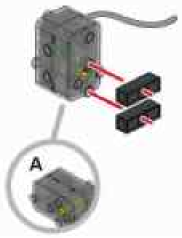
5



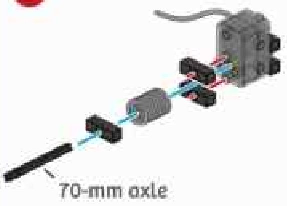
6



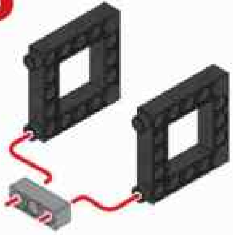
7



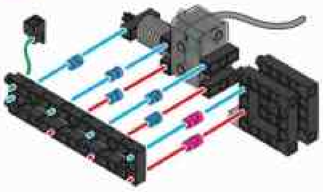
8



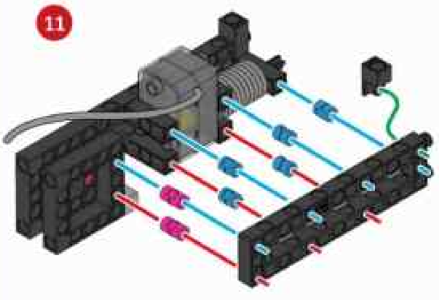
9



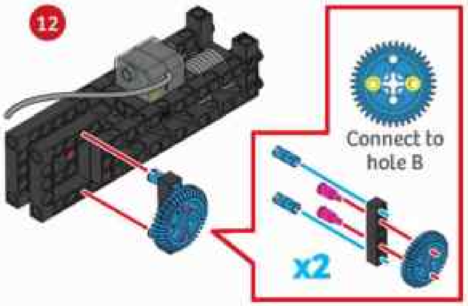
10



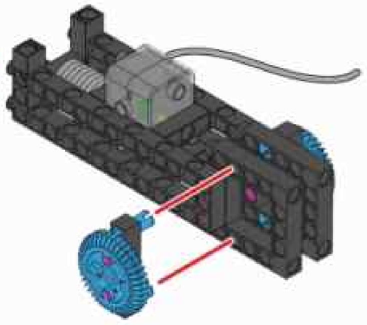
11



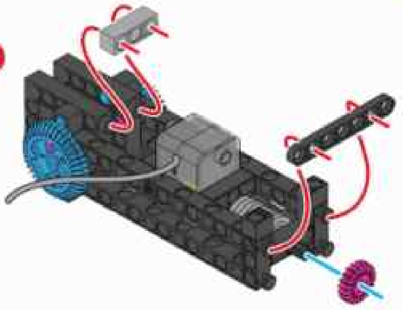
12



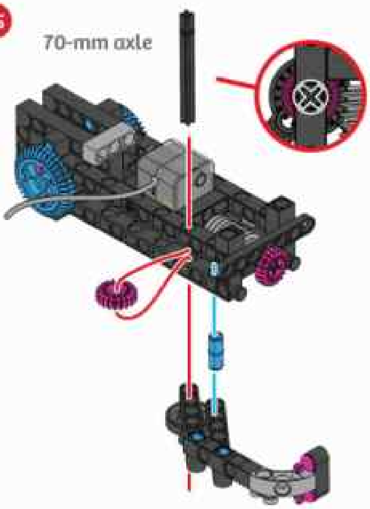
13



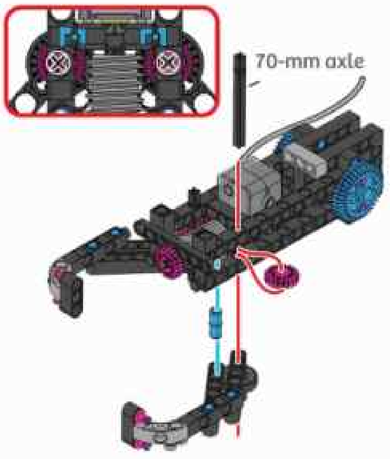
14



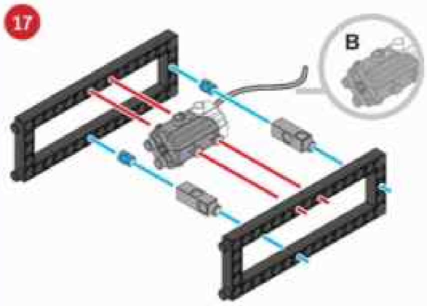
15



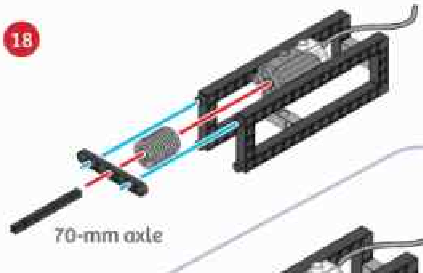
16



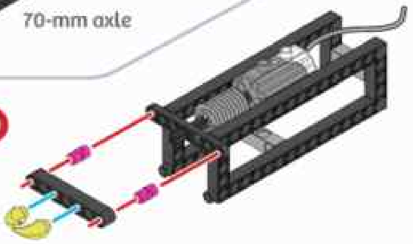
17



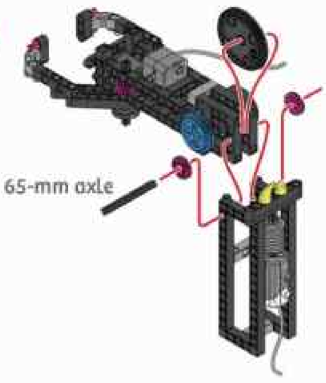
18



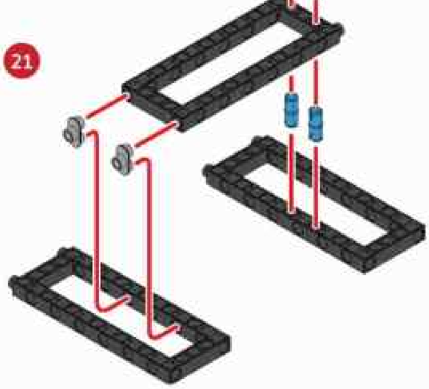
19



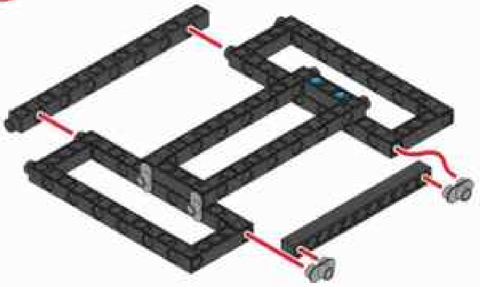
20



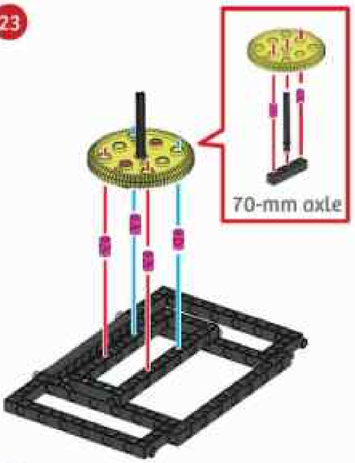
21



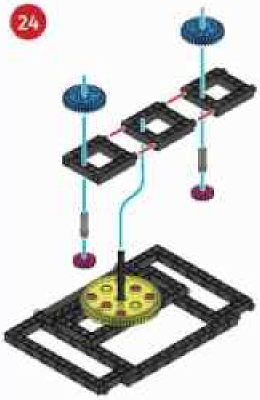
22



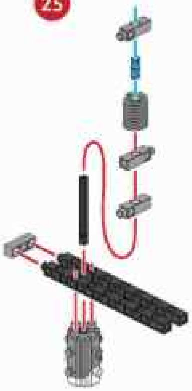
23



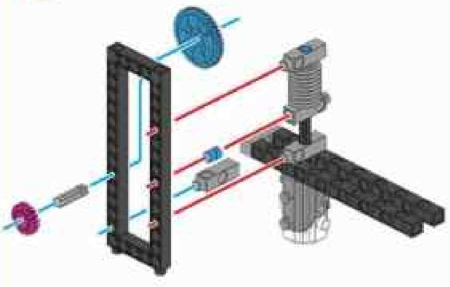
24



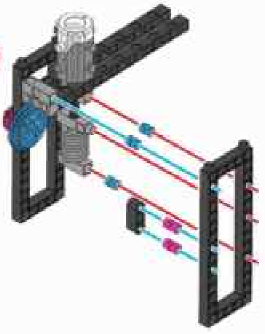
25



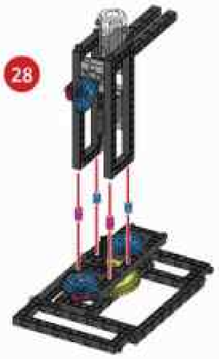
26



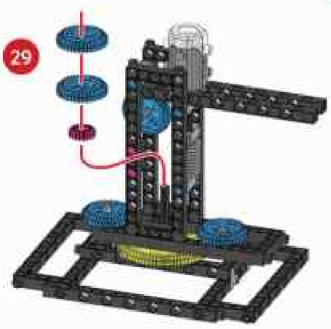
27



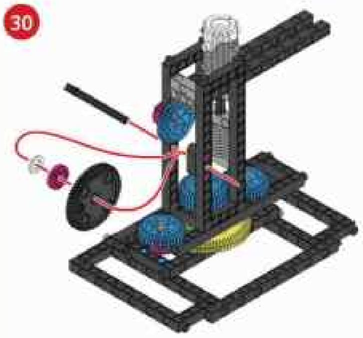
28



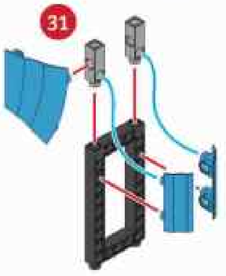
29



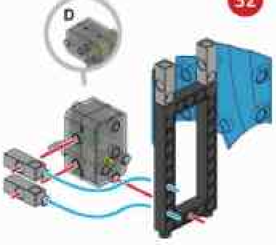
30



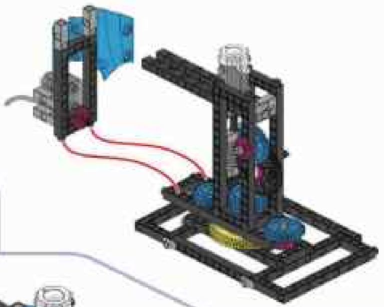
31



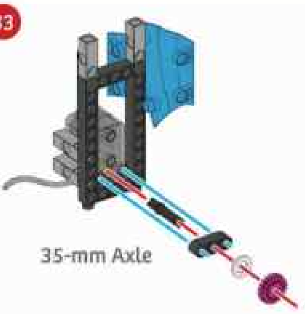
32



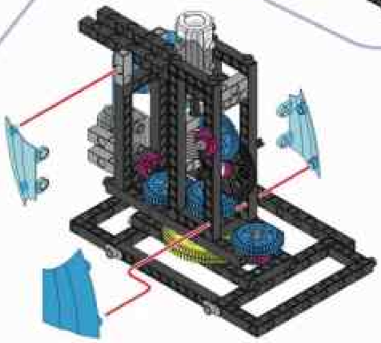
34



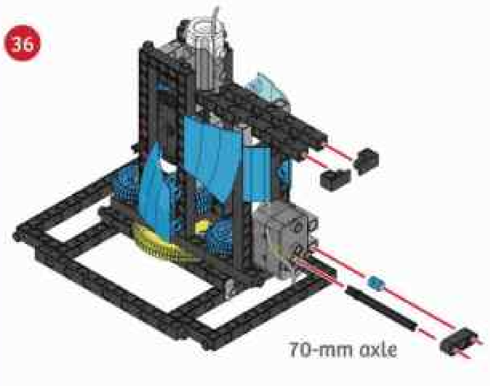
33



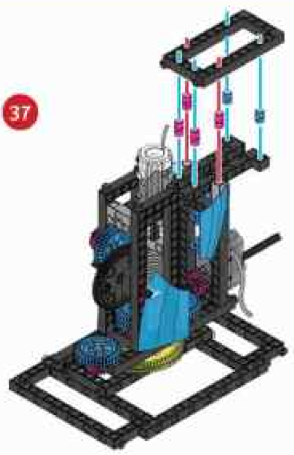
35



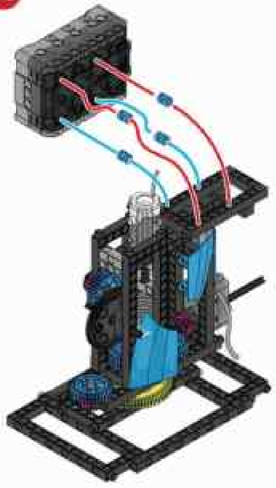
36


70-mm axle

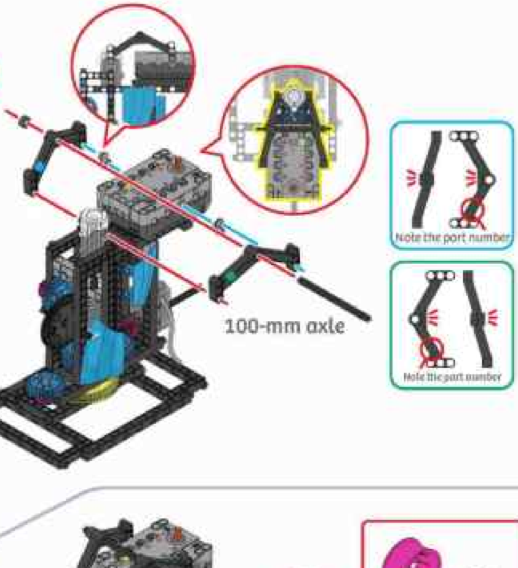
37



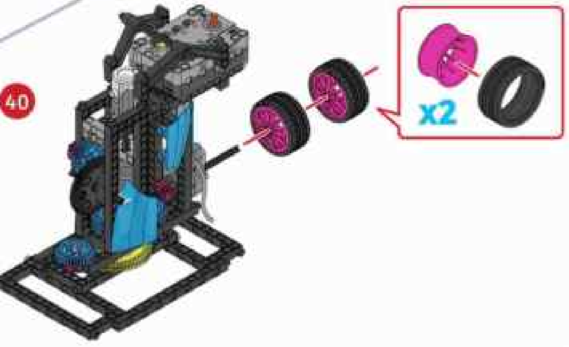
38



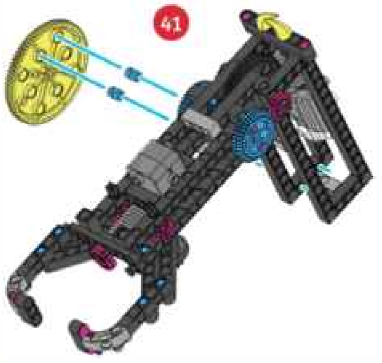
39


100-mm axle
Note the port number
Note the port number

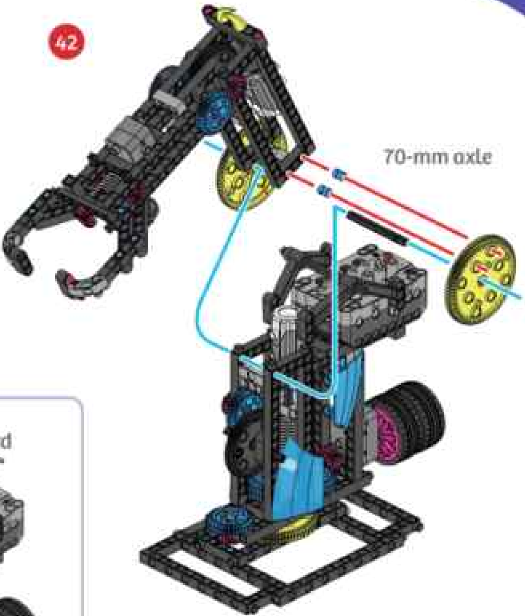
40


x2

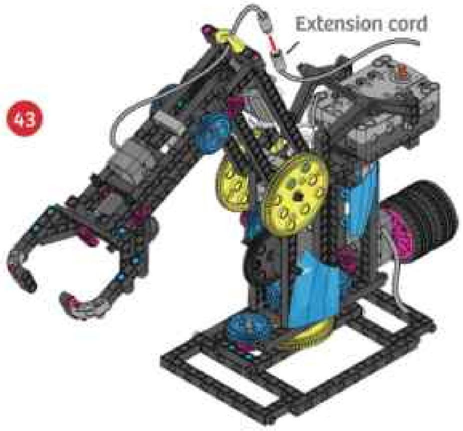
41



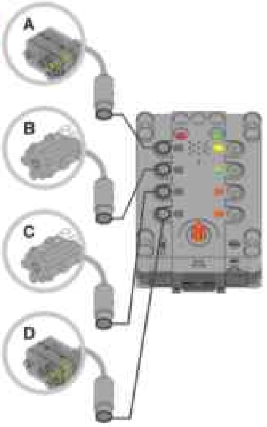
42



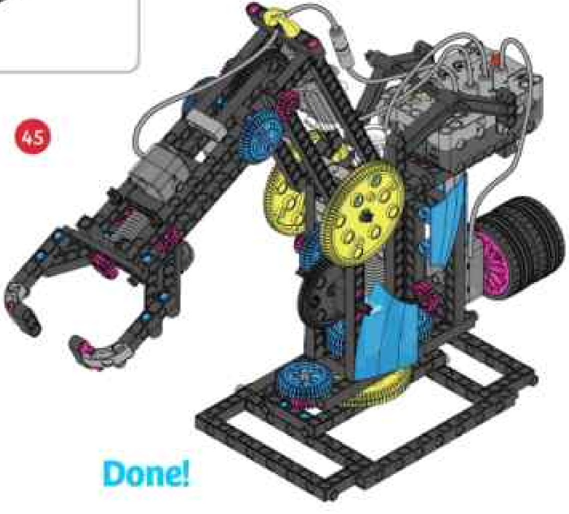
43



44



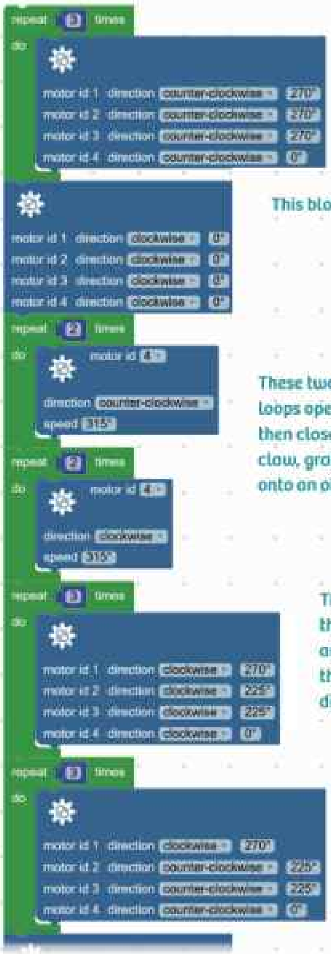
45



Done!

SAMPLE PROGRAM FOR THE ROBOTIC ARM

The robotic arm is able to move up and down, open and close its claw to grab objects, and rotate around. This allows the robotic arm to place objects in a new location, just like a robotic arm on a factory assembly line. This program is preloaded in the app in the Load Demo Code menu. Test the program out. Then place objects around the robotic arm and try to pick them up and move them around.



```
repeat 3 times
do
    motor id 1  direction counter-clockwise  270°
    motor id 2  direction counter-clockwise  270°
    motor id 3  direction counter-clockwise  270°
    motor id 4  direction counter-clockwise  0°
motor id 1  direction clockwise  0°
motor id 2  direction clockwise  0°
motor id 3  direction clockwise  0°
motor id 4  direction clockwise  0°
repeat 2 times
do
    motor id 4
    direction counter-clockwise
    speed 315°
repeat 2 times
do
    motor id 4
    direction clockwise
    speed 315°
repeat 3 times
do
    motor id 1  direction clockwise  270°
    motor id 2  direction clockwise  225°
    motor id 3  direction clockwise  225°
    motor id 4  direction clockwise  0°
repeat 3 times
do
    motor id 1  direction clockwise  270°
    motor id 2  direction counter-clockwise  225°
    motor id 3  direction counter-clockwise  225°
    motor id 4  direction counter-clockwise  0°
```

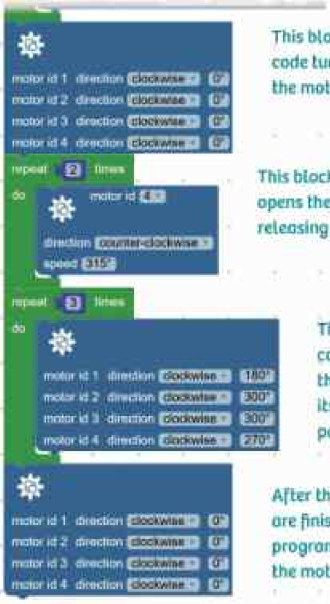
The first loop repeats the following three times: It turns motor 1 counterclockwise, rotating the robotic arm; and it turns motors 2 and 3 counterclockwise, lowering the arm.

This block of code turns off all the motors.

These two loops open and then close the claw, grabbing onto an object.

This loop raises the robotic arm and rotates it in the opposite direction.

This loop rotates the arm some more, but this time lowering the arm.



```
motor id 1  direction clockwise  0°
motor id 2  direction clockwise  0°
motor id 3  direction clockwise  0°
motor id 4  direction clockwise  0°
repeat 2 times
do
    motor id 4
    direction counter-clockwise
    speed 315°
repeat 3 times
do
    motor id 1  direction clockwise  180°
    motor id 2  direction clockwise  300°
    motor id 3  direction clockwise  300°
    motor id 4  direction clockwise  270°
motor id 1  direction clockwise  0°
motor id 2  direction clockwise  0°
motor id 3  direction clockwise  0°
motor id 4  direction clockwise  0°
```

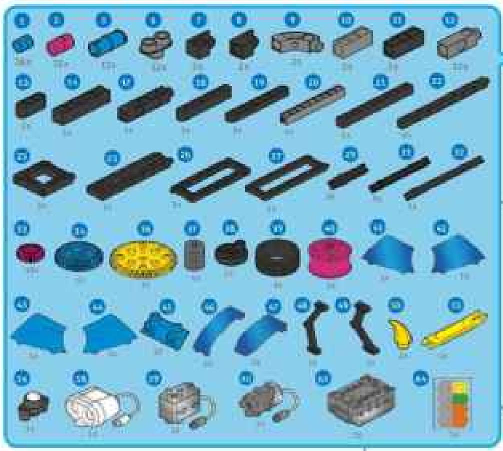
This block of code turns all the motors off.

This block of code opens the claw, releasing the object.

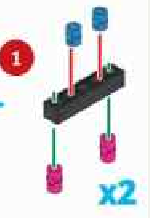
This block of code returns the claw to its starting position.

After three loops are finished, the program turns all the motors off.

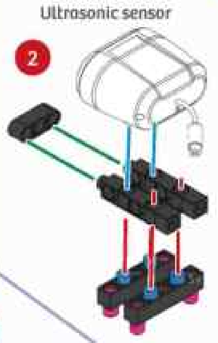
BUTLER ROBOT



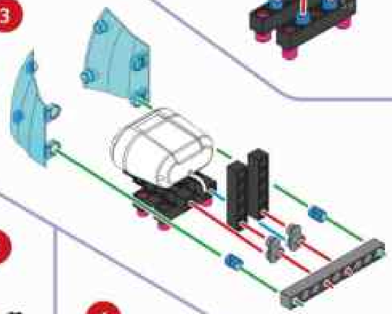
1



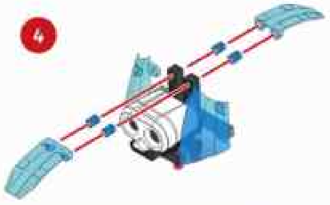
2



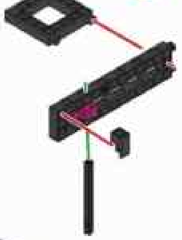
3



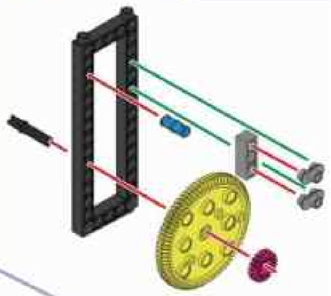
4



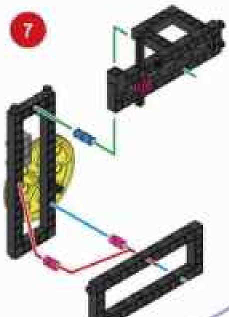
5



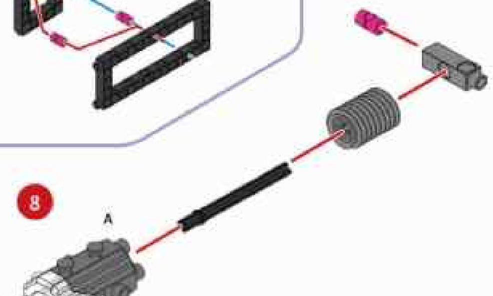
6



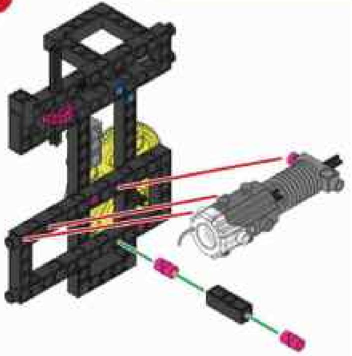
7



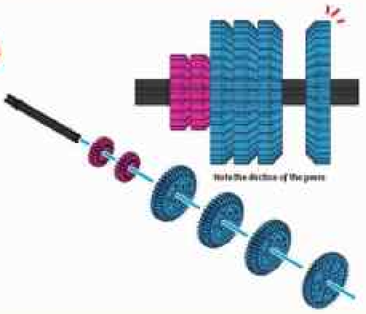
8



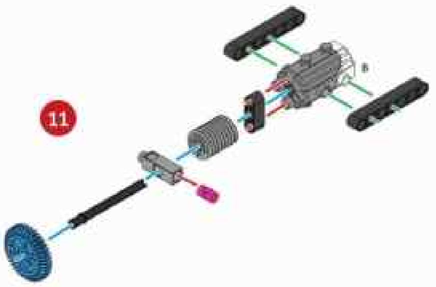
9



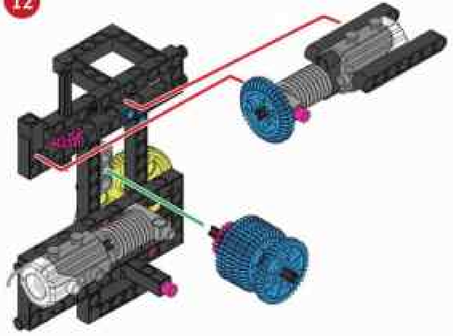
10



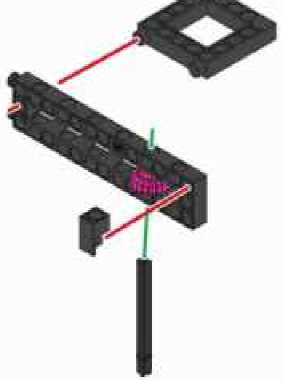
11



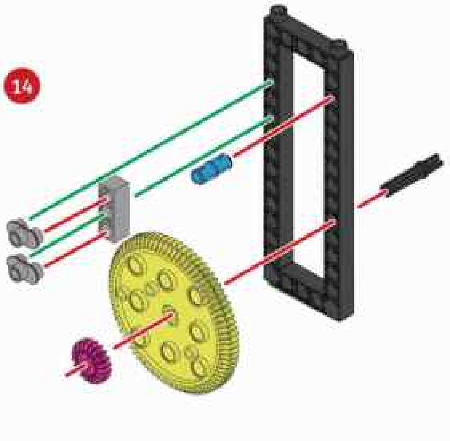
12



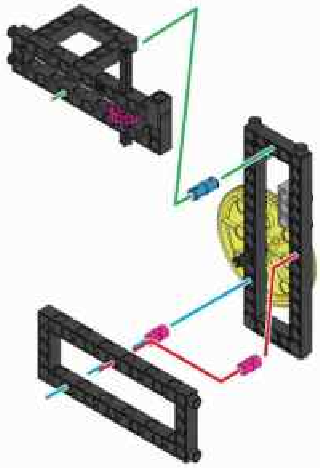
13



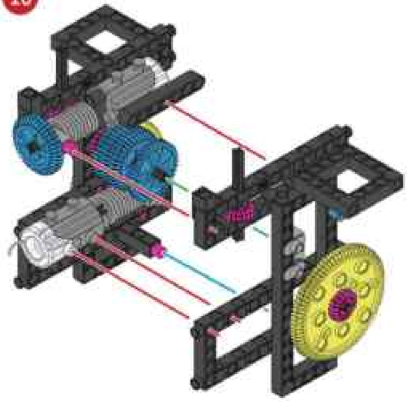
14



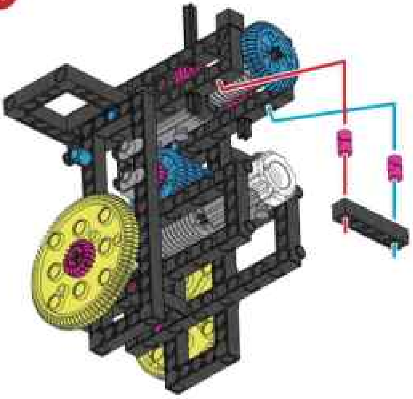
15



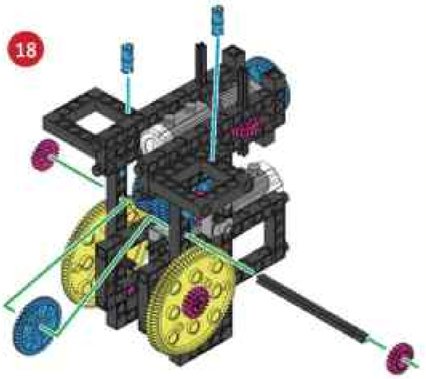
16



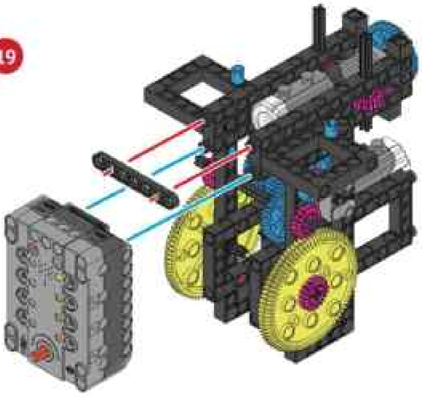
17



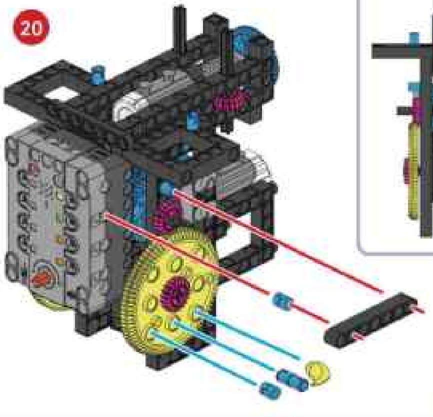
18

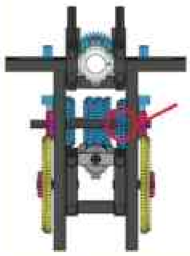


19

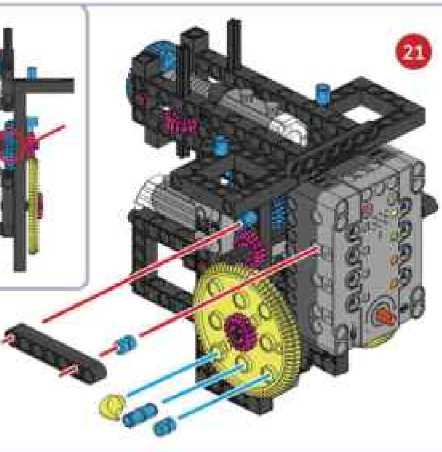


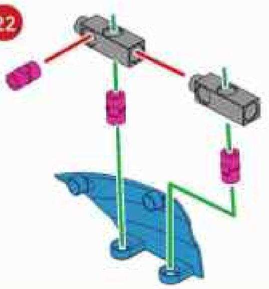
20

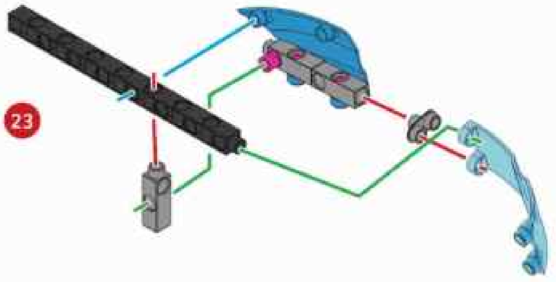


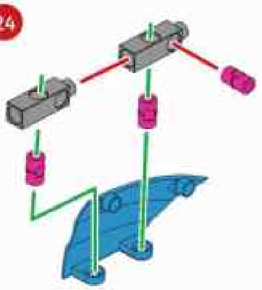


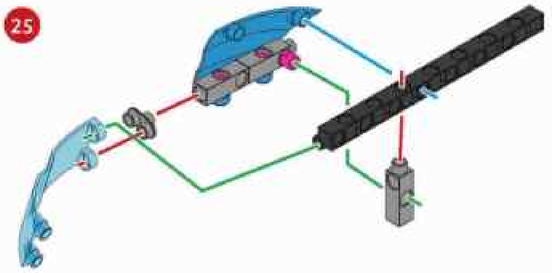
21

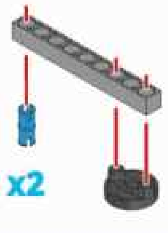


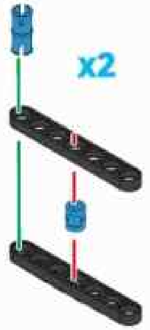
22


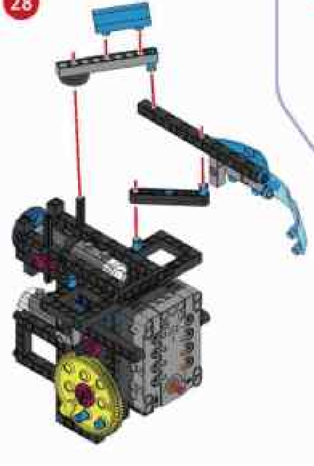
23


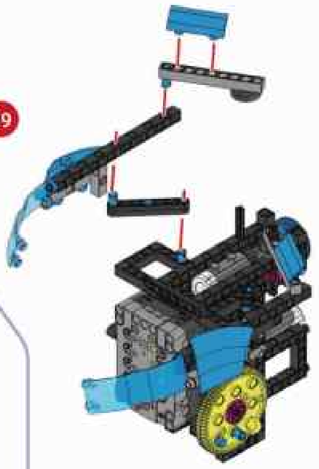
24


25


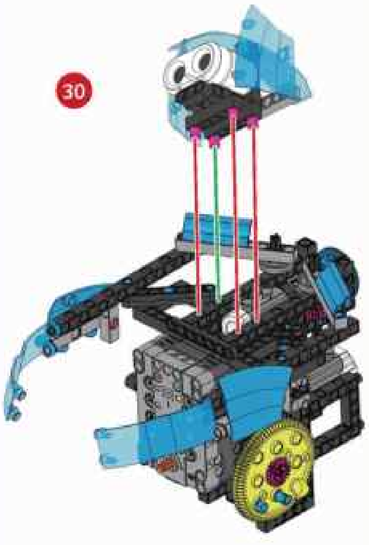
26


27


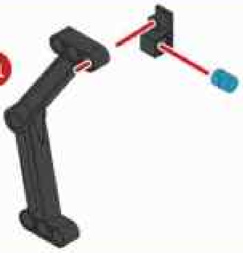
28


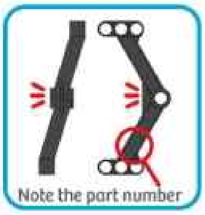
29


30

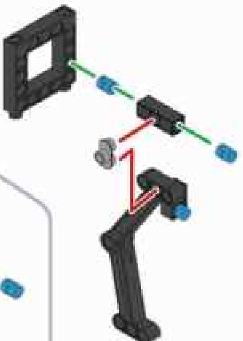


31

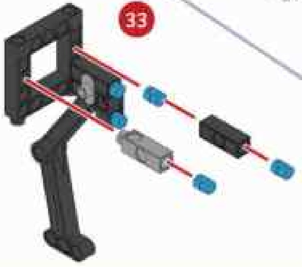




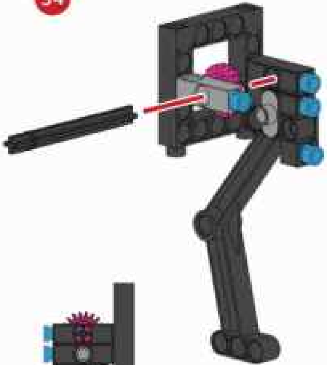
32

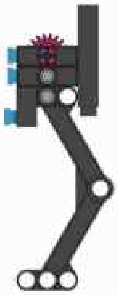


33

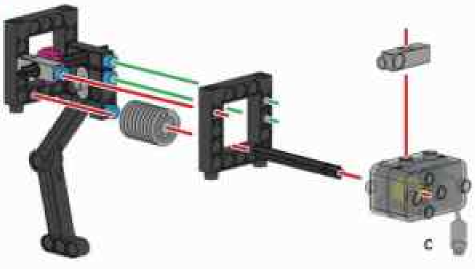


34

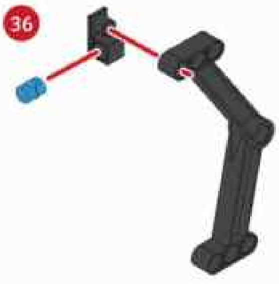


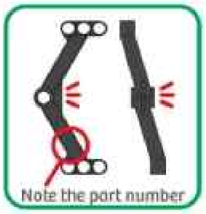


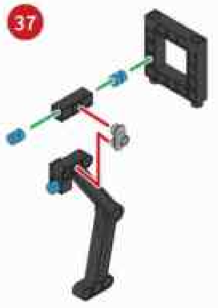
35

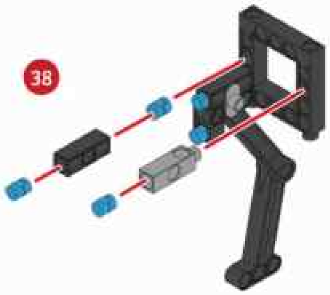


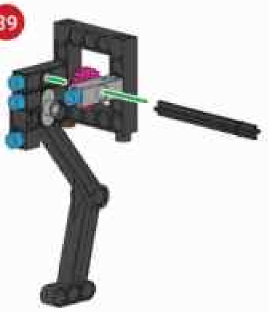
36

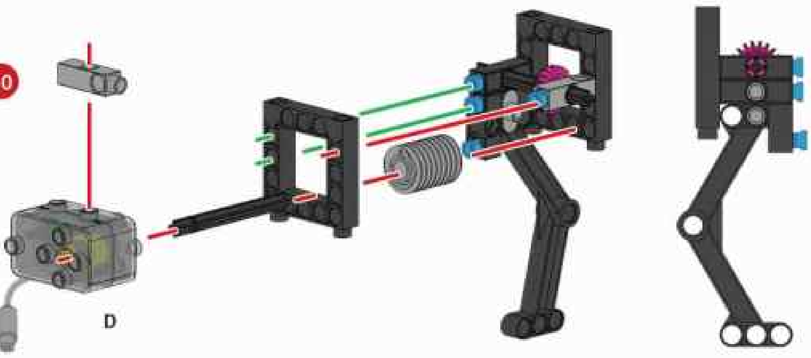


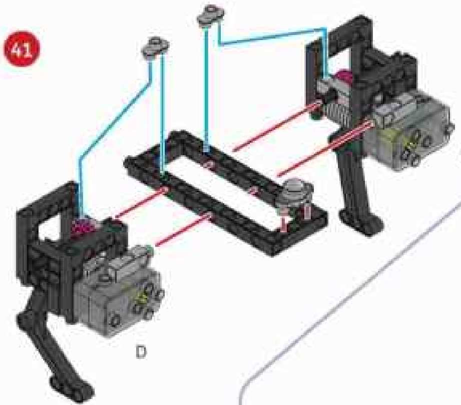


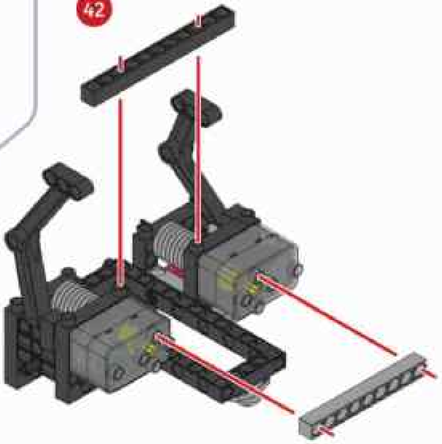
37


38


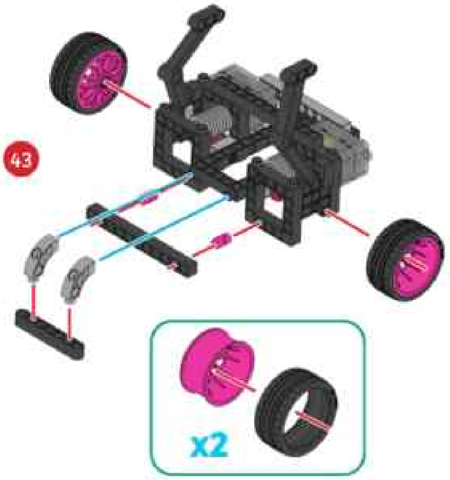
39


40


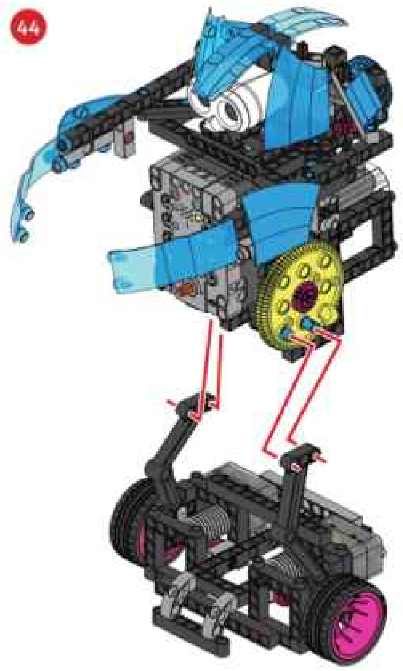
41


42


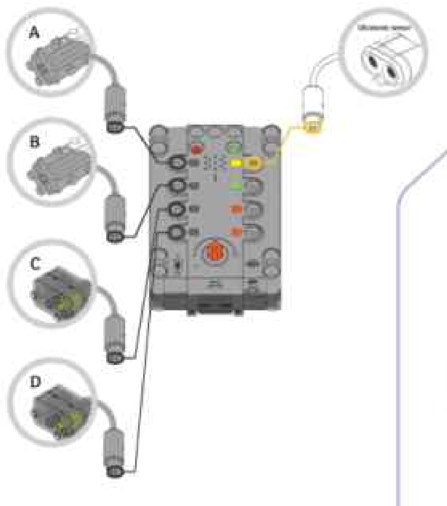
43



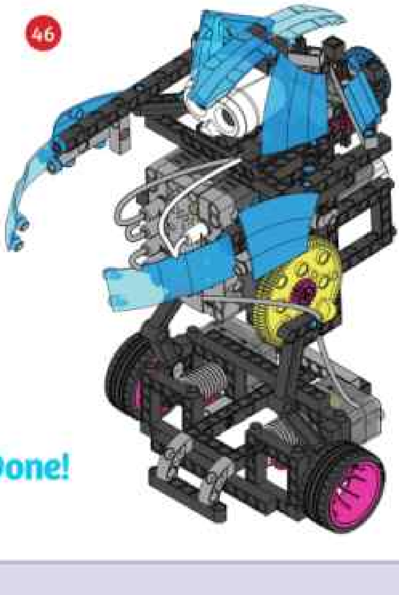
44



45



46

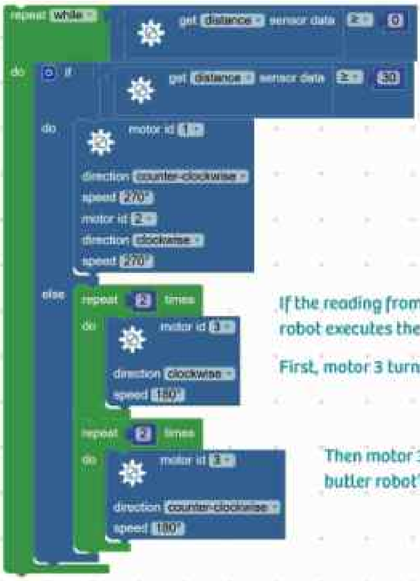


Done!

SAMPLE PROGRAM FOR THE BUTLER ROBOT

The butler robot is able to move around and sense objects using its ultrasonic sensor. When the butler robot senses that an object is close enough, it can close its arms and grab the object. Place an object, such as a paper towel roll, on top of book in front of the butler robot. Make sure that ultrasonic sensor detects the object. Then test the following program. This program is preloaded in the app in the Load Demo Code menu.

This program is placed in a loop so that the code will repeat over and over again. This loop will repeat as long as the value from the ultrasonic sensor is greater than or equal to 0.



```
repeat while   get distance sensor data ≥ 0
do   if   get distance sensor data ≥ 30
     do   motor id 1
          direction counter-clockwise
          speed 270°
          motor id 2
          direction clockwise
          speed 270°
     else repeat 2 times
          do   motor id 3
               direction clockwise
               speed 180°
          repeat 2 times
          do   motor id 3
               direction counter-clockwise
               speed 180°
```

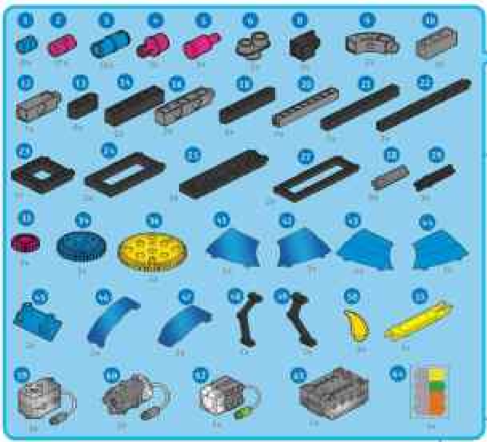
Within the loop there is an if-else statement. First the program checks whether the reading from ultrasonic sensor is greater than or equal to 30. If it is, then motors 1 and 2 are turned on, moving the butler robot forward.

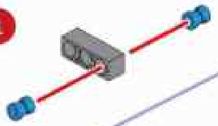
If the reading from the ultrasonic sensor is less than 30, then the robot executes the "else" portion of the program.

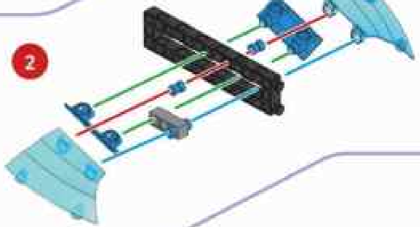
First, motor 3 turns clockwise, opening the butler robot's arms.

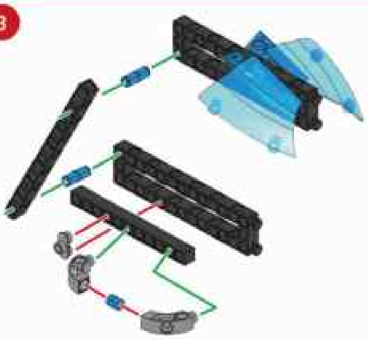
Then motor 3 turns counterclockwise, closing the butler robot's arms around the object.

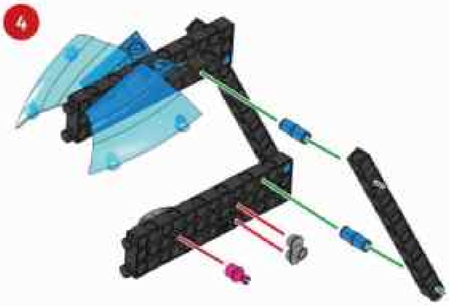
SCORPION ROBOT

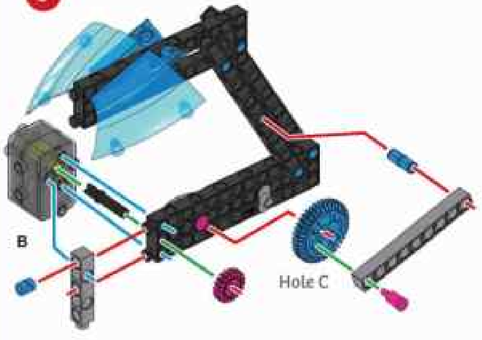


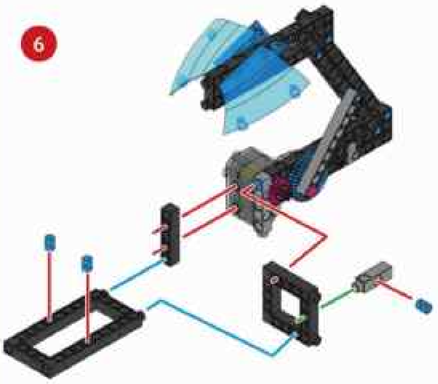
1


2


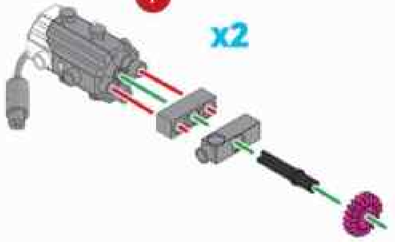
3


4


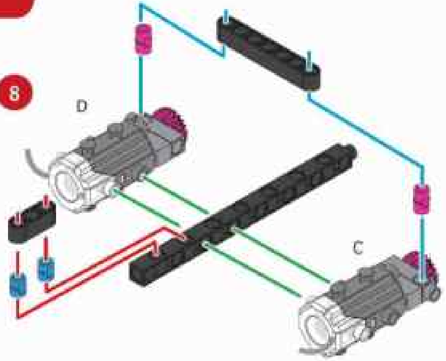
5


6


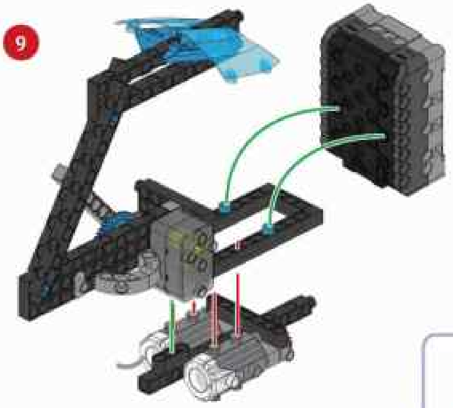
7



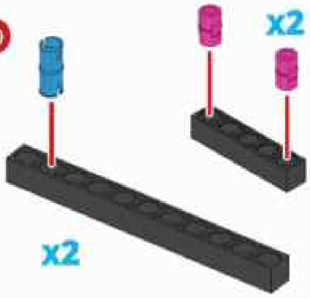
8



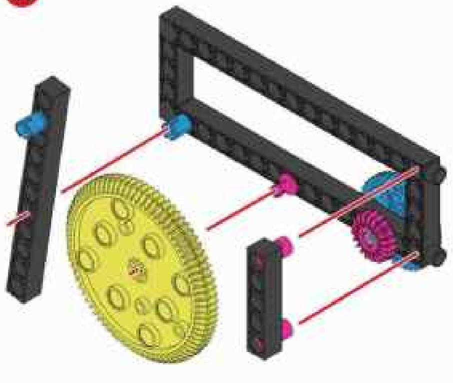
9



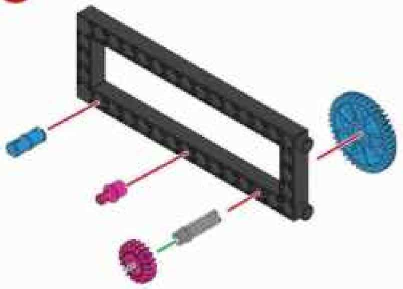
10



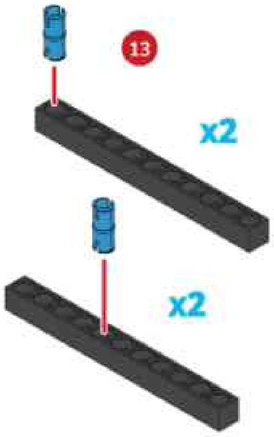
12



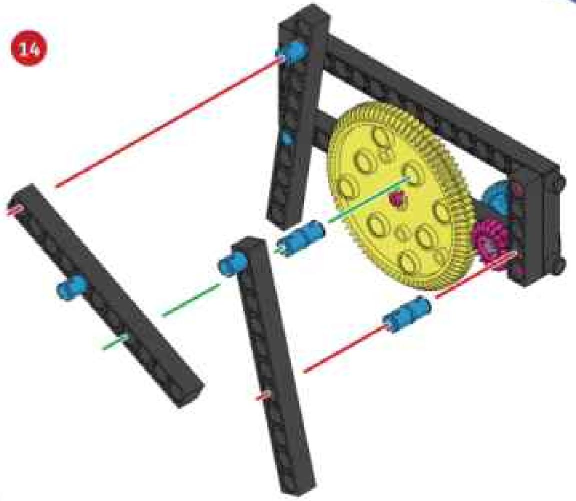
11

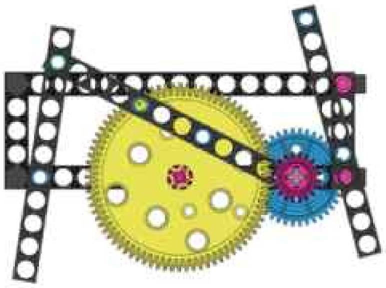


13

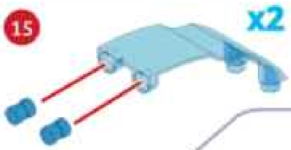


14

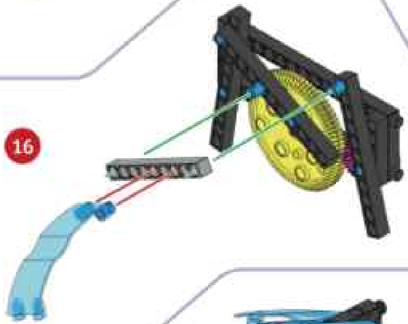




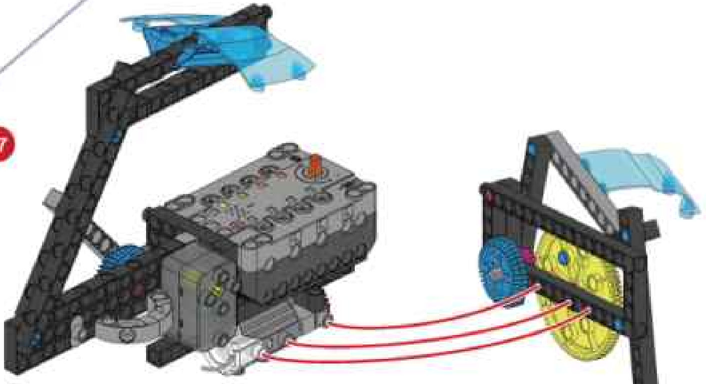
15



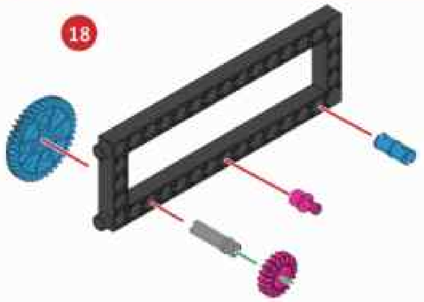
16



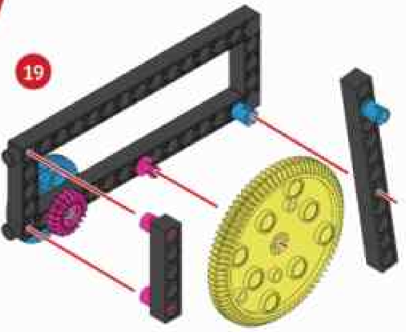
17



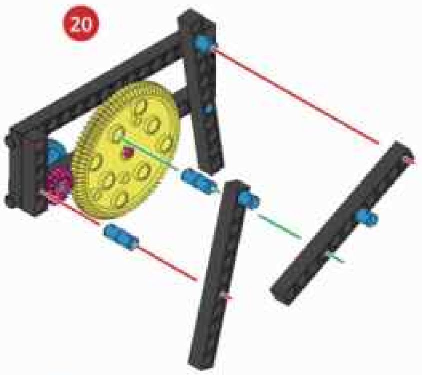
18

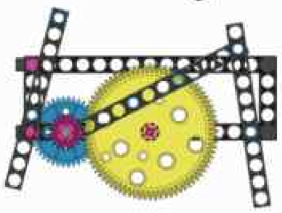


19

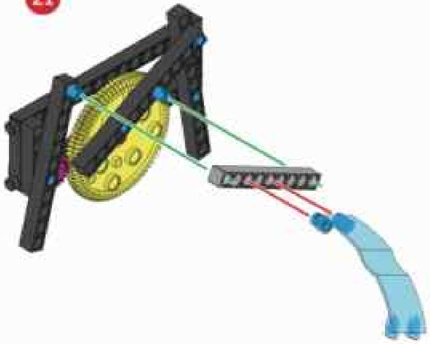


20

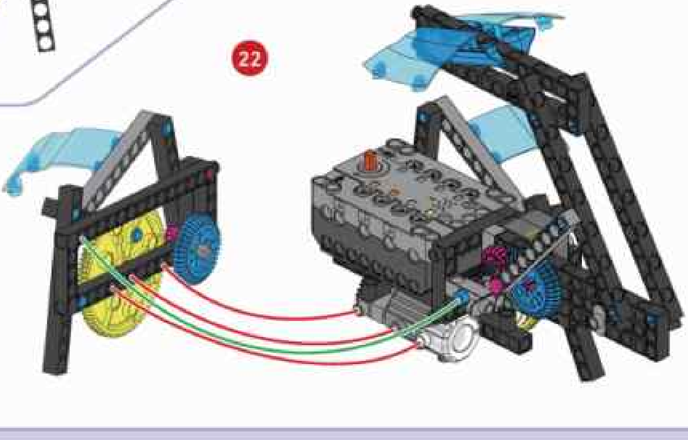


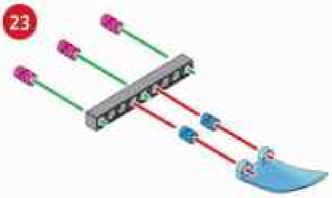


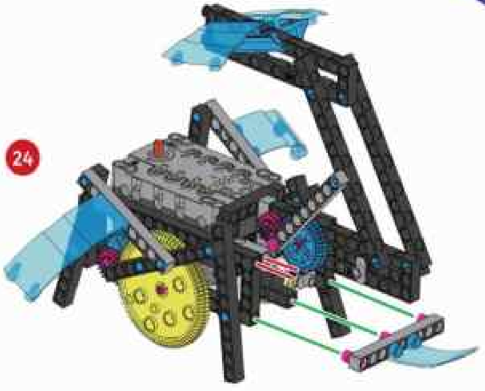
21

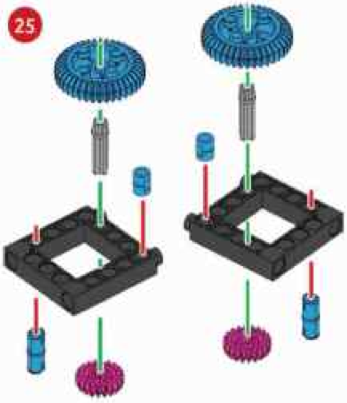


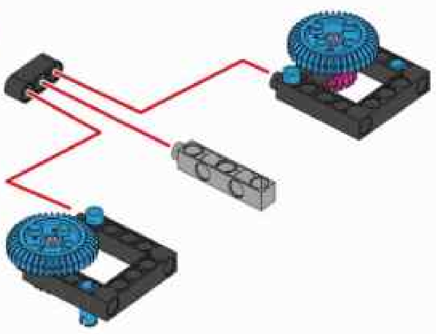
22

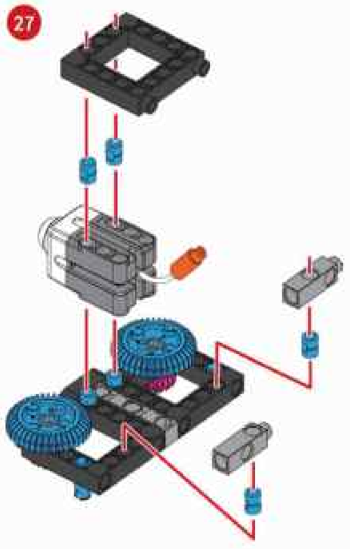


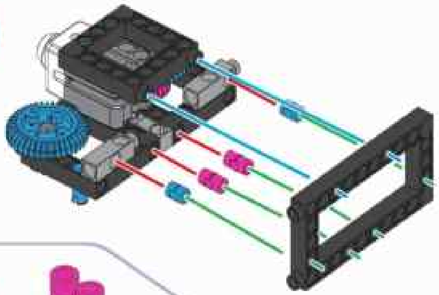
23


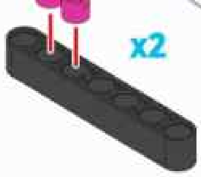
24


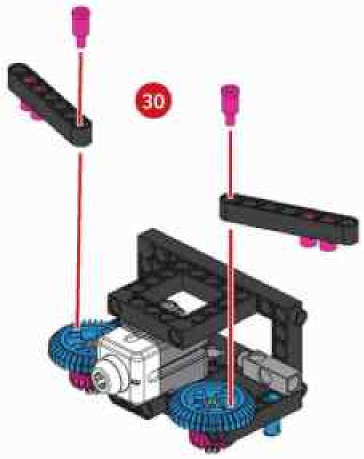
25


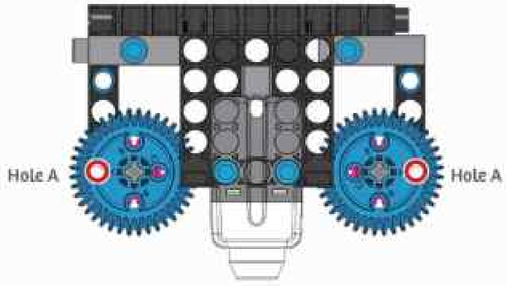
26


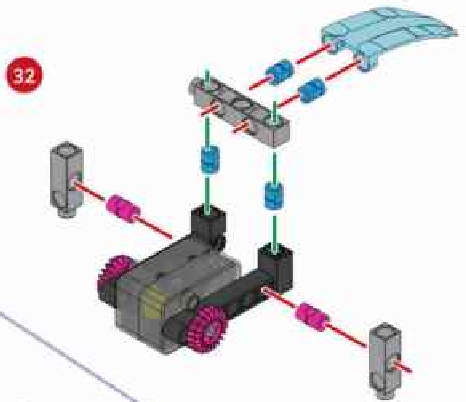
27


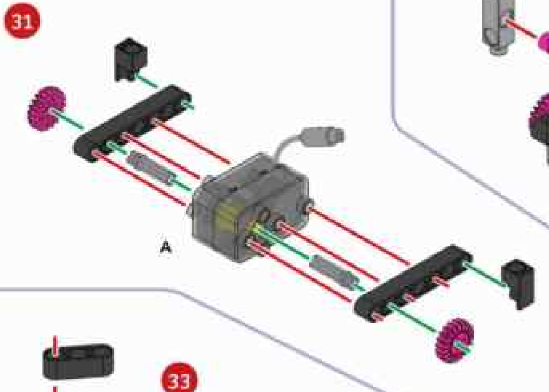
28


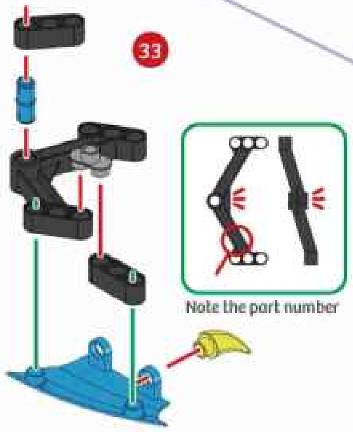
29


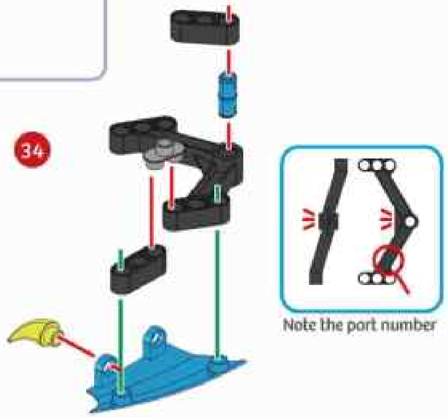

30


Hole A
Hole A

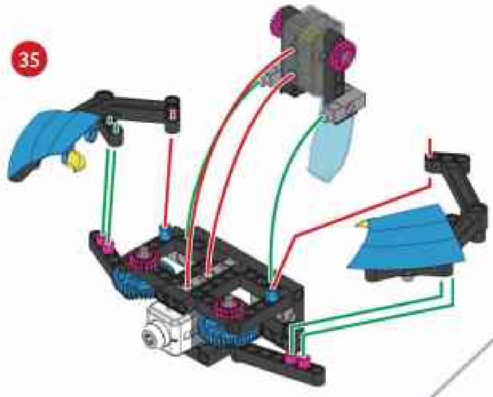

32


31
A

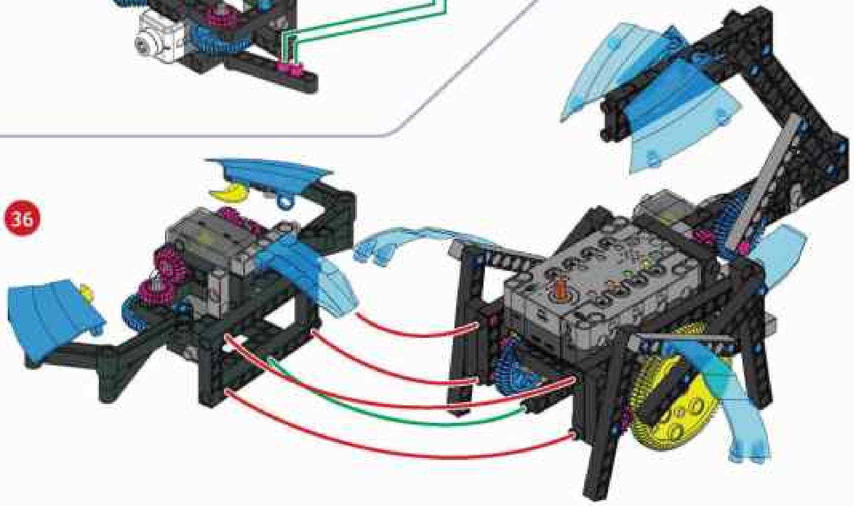

33
Note the part number


34
Note the part number

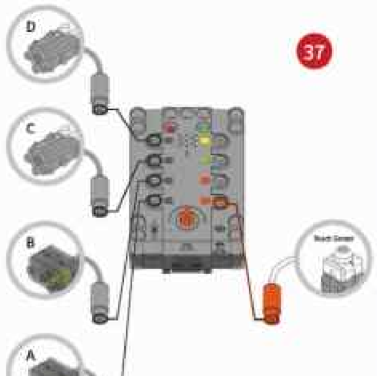
35



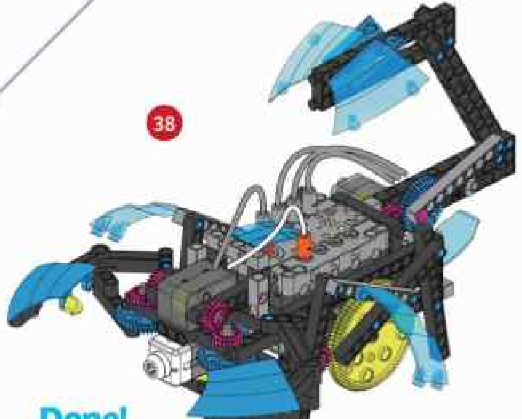
36



37



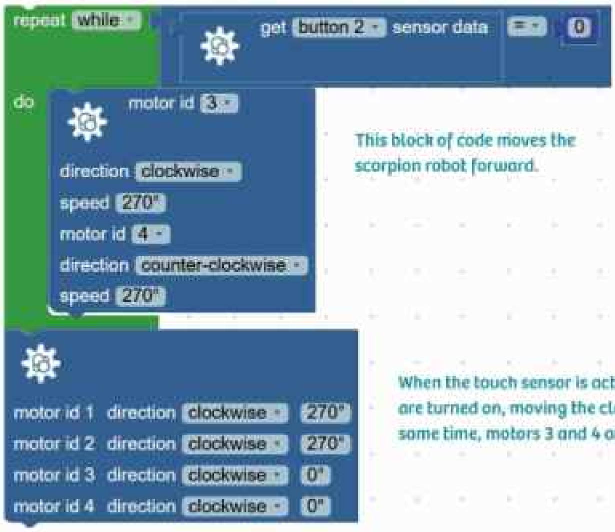
38



Done!

SAMPLE PROGRAM FOR THE SCORPION ROBOT

The scorpion robot has two motors which it uses to move around. But, when its touch sensor is activated, it can be programmed to strike with its claws and tail. This program is preloaded in the app in the Load Demo Code menu.



```
repeat while   get button 2 sensor data = 0
do
    motor id 3
    direction clockwise
    speed 270°
    motor id 4
    direction counter-clockwise
    speed 270°

motor id 1  direction clockwise  270°
motor id 2  direction clockwise  270°
motor id 3  direction clockwise  0°
motor id 4  direction clockwise  0°
```

The first part of the program is a loop which repeats as long as the touch sensor has not been activated.

This block of code moves the scorpion robot forward.

When the touch sensor is activated, motors 1 and 2 are turned on, moving the claws and tail. At the same time, motors 3 and 4 are turned off.