

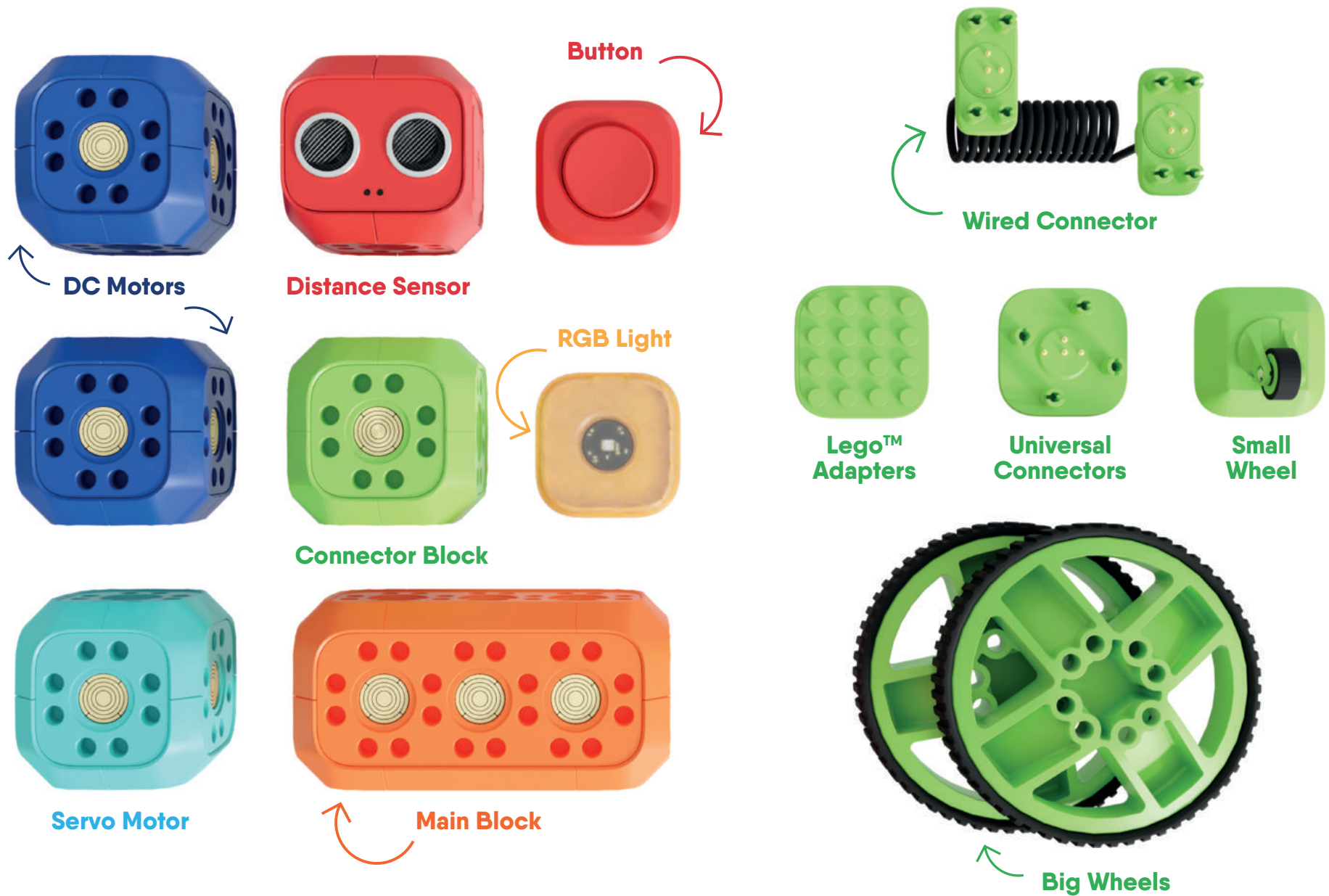


# Teacher's Guide

# Table of Contents:

<b>Chapter</b>	<b>Page</b>
1. Robo Wunderkind Robotics Kit Key Information:	
• What's in the Box	2
• How to Get Started	3
<hr/>	
2. Robo Live and Code Apps: What's on the Screen?	4
<hr/>	
3. Robo Live App: How to Control Your Robo	5
<hr/>	
4. Robo Code App: How to Program Your Robo	
• Robo Code App: Programming Screen	6
• Robo Code App: Actions and Conditions	7
• Robo Code App: Programming Logic	8–10
<hr/>	
5. Robo Wunderkind Modules:	11
• Main Block	12
• Connectors: Universal, Wired, Connector Block, Disconnecting Tool	13–14
• RGB Light	15
• DC Motors	16
• Servo Motor	17
• Button	18
• Distance Sensor	19
• Technical Details: Main Block	20
<hr/>	
6. Lessons with RW Robotics Kit: Key Information	21–22
<hr/>	
7. Supporting Materials:	
• Key Vocabulary for Teachers	23–25

# 1. What's in the Box



# 1. How to Get Started

## Device requirements:

- An iOS 8 / Android 4.4.2 or newer with Bluetooth 4.0. / Windows version higher than 8.1

## Our Apps – free to download:

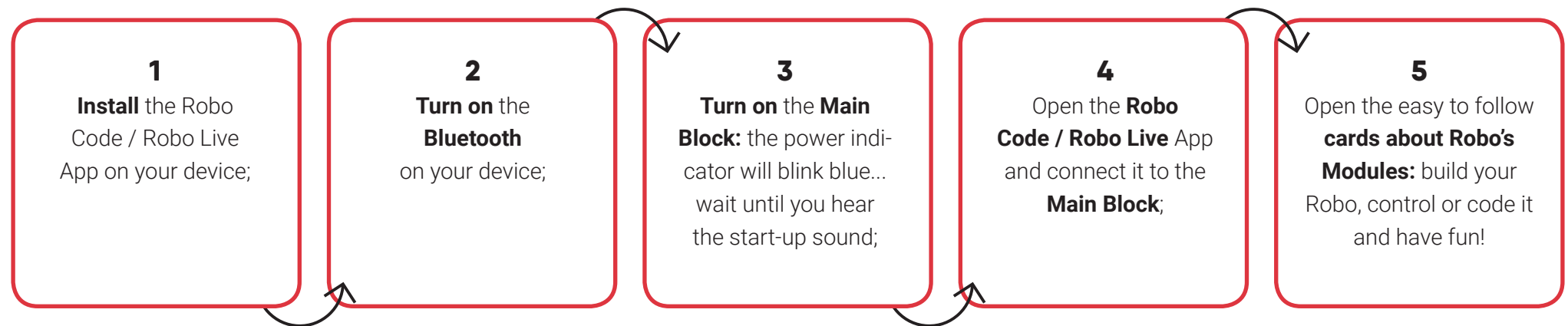


Robo Code App



Robo Live App

## First steps:



## DO:

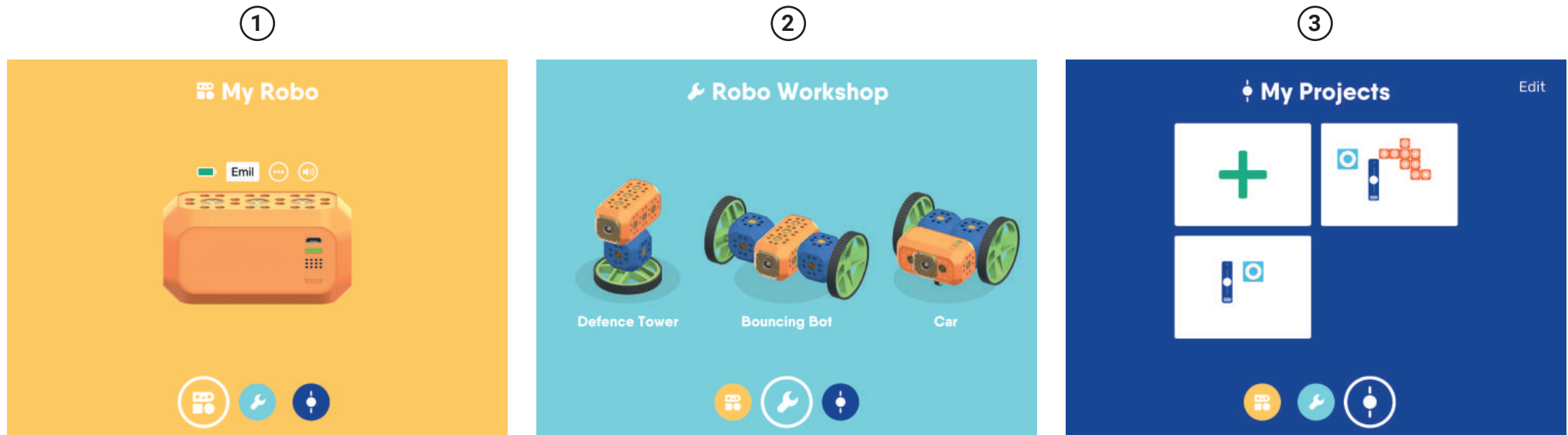
- Use the included Connectors to attach the modules;
- Use the included Disconnecting Tool to detach the connectors;
- Use Apps to control and program the modules;
- Use Lego™ bricks to customize your Robo.



## DON'T:

- Use metal or sharp objects to detach the Connectors;
- Allow any contact with water;
- Try to disassemble the modules;
- Throw or drop the modules;
- Rotate DC or Servo motors by hand.

## 2. Robo Live and Code Apps: What's on the Screen?



### ① My Robo Screen

Find all the information about your Robo here – its name, battery level, firmware update, and the Module(s) connected.

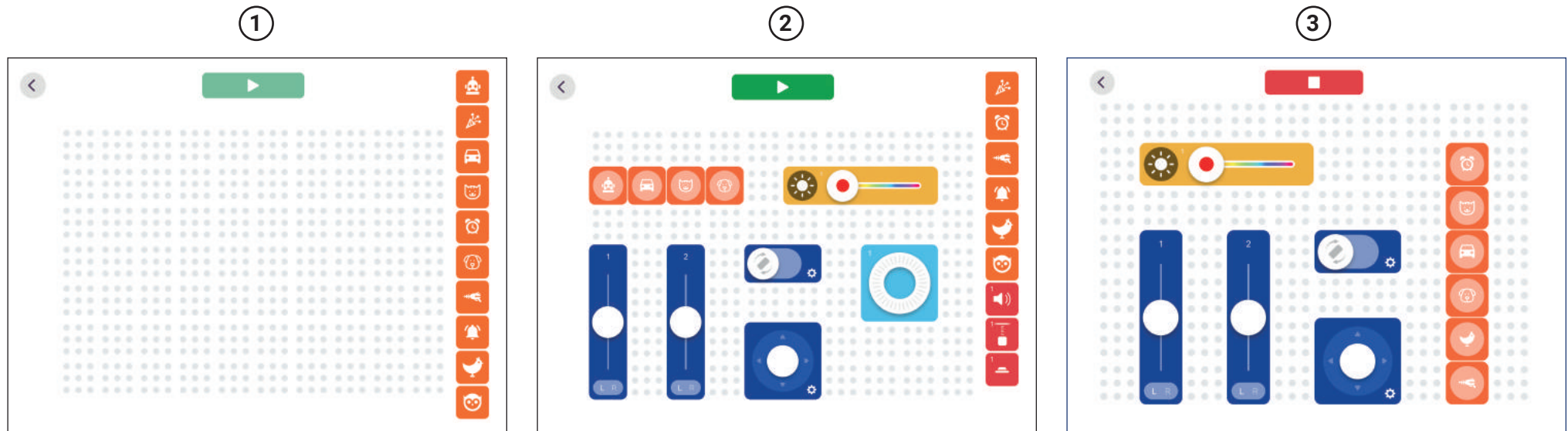
### ② Robo Workshop

Try fun projects from Robo Team.

### ③ My Projects

Here you can see all the projects you create. **New Project button:** Click it to start a new project.

# 3. Robo Live App: How to Control Your Robo



## ① Controlling screen

Here you will see the Controls added.

**Controls Menu:** See all the Controls available at the moment, drag and drop a Control to the Controlling screen to try it out.

**Play / Edit button:** When you are ready to play, press the Play button and control your Robo; press Edit button to go back to the editing mode.

**Save button:** Click it to return to the project screen; don't worry, your project will be saved automatically.

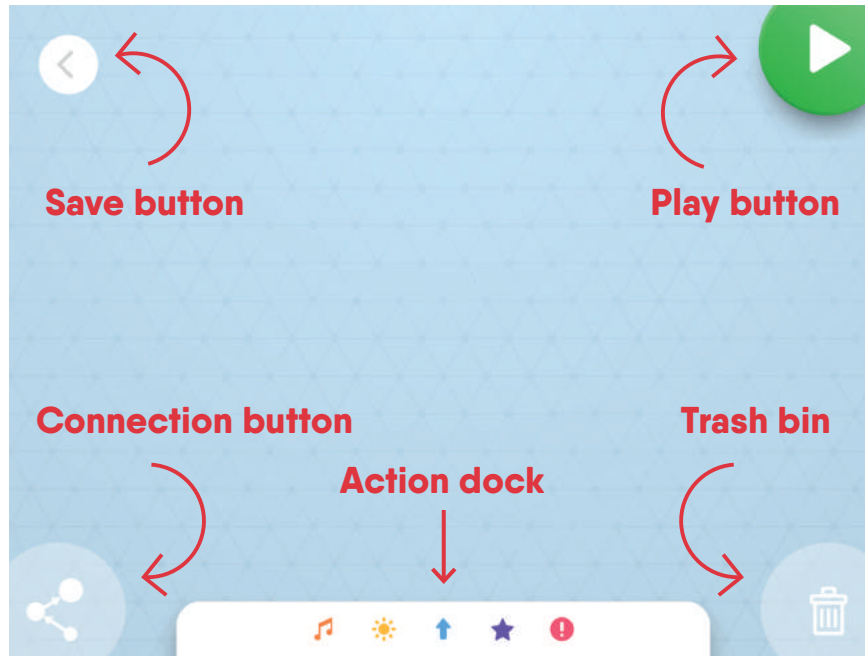
## ② Controls

- When you create a new project, the Controlling screen is empty;
- Using only the Main Block, you can already add Sound control(s); simply drag and drop them onto the screen;
- When you attach the Module to the Main Block, the Control(s) available for this module appears in Controls Menu automatically;
- You can delete the Control from the Controlling screen if needed.

## ③ Play Mode

When you are ready to play, press the Play button and control your Robo; press Edit button to go back to the editing mode.

# 4. Robo Code App: Programming Screen



## Programming screen

The unlimited space for your future program.

**Actions dock:** Here you will find all the Actions and Conditions to program.

**Connection button:** Use it to connect the Actions into the code.

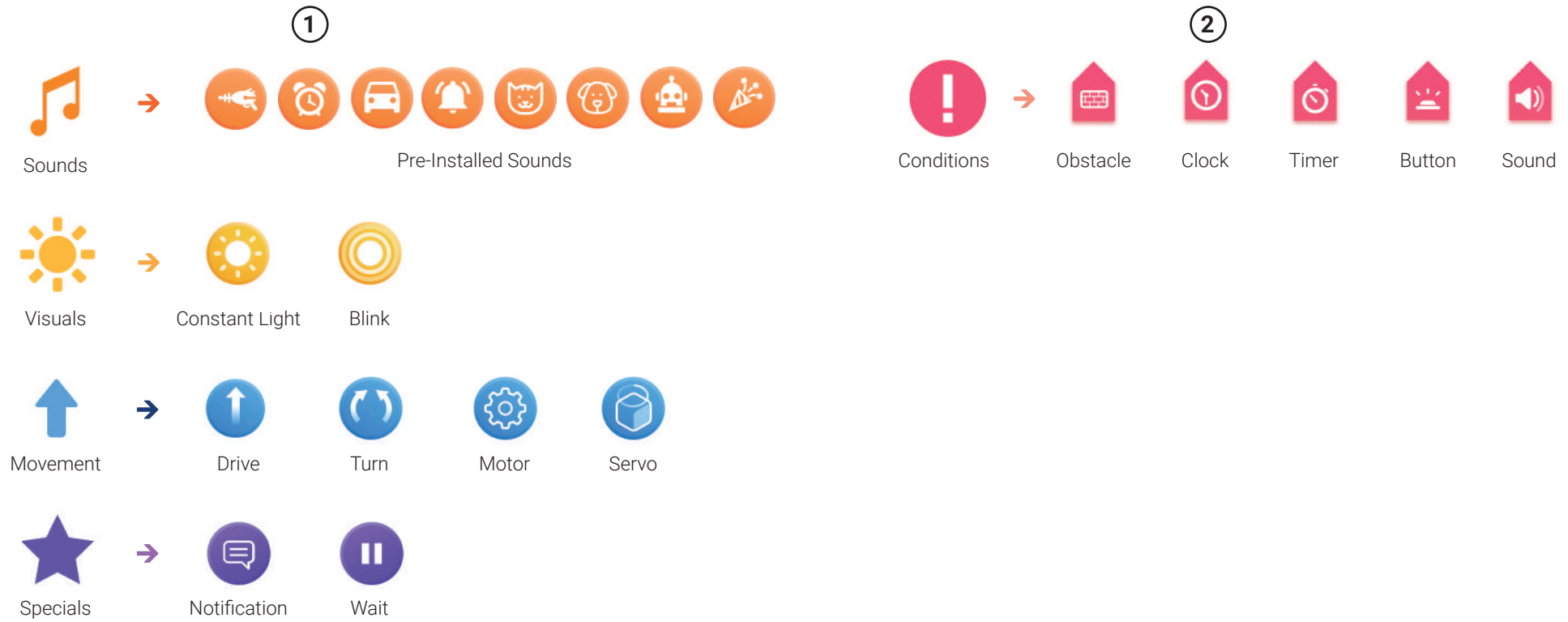
**Trash bin:** Use it to delete the Actions or Transitions / Connections.

**Play button:** Try out your program!

**Stop button:** When you run your code, the 'Play' button changes to the 'Stop' button; you can stop your Robo any time you want.

**Save button:** Click it to return to the project screen; don't worry, your project will be saved automatically.

# 4. Robo Code App: Actions and Conditions



## ① **Actions:**

Coding icons which look like bubbles and perform an output task. There are 4 types of Actions in the Actions Dock – Sounds, Visuals, Movement and Specials.

## ② **Conditions:**

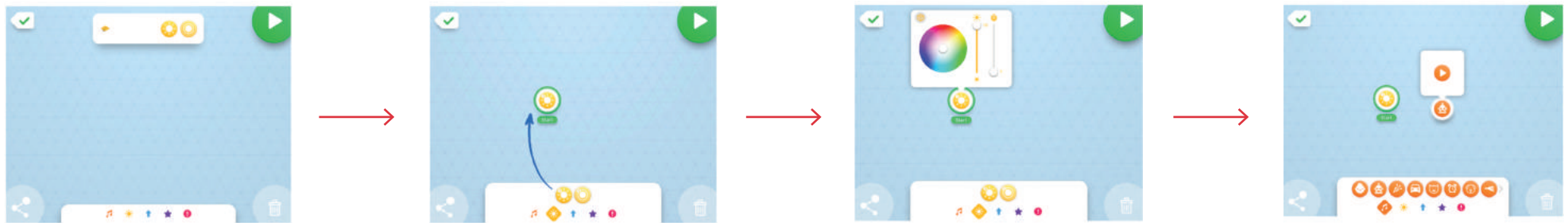
Coding icons which look like pink stickers and have to be placed on the Connection between 2 Actions. Conditions compare two numbers and determines the results to be true or false, and tells if true – Connection/Transition happens or false – nothing happens.



# 4. Robo Code App: Programming Logic

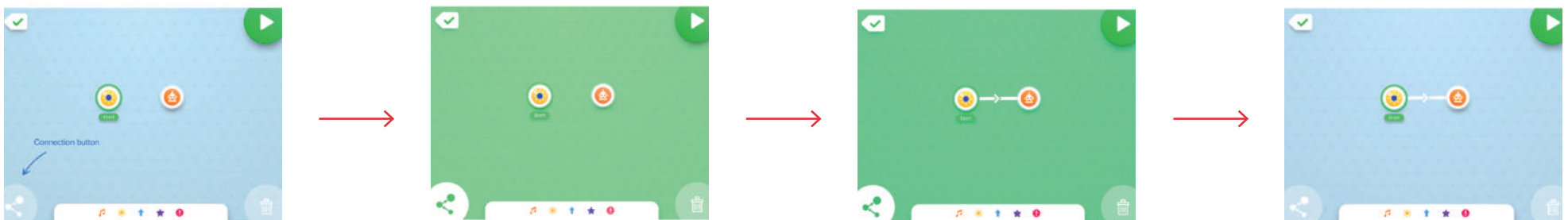
## 1. Program a certain Action:

- You can program a certain **Action** only if the module **which corresponds to this Action is assembled** to your Robo's Main Block; when you attach the module you see the push notification about new Actions available to program;
- **Drag and drop** an Action from the Action dock onto the Programming Screen; you can place it anywhere you like;
- **Action's settings:** Click on it one more time to open the settings; click once again to hide it; not all of the Actions have the setting option.



## 2. Connect the Actions:

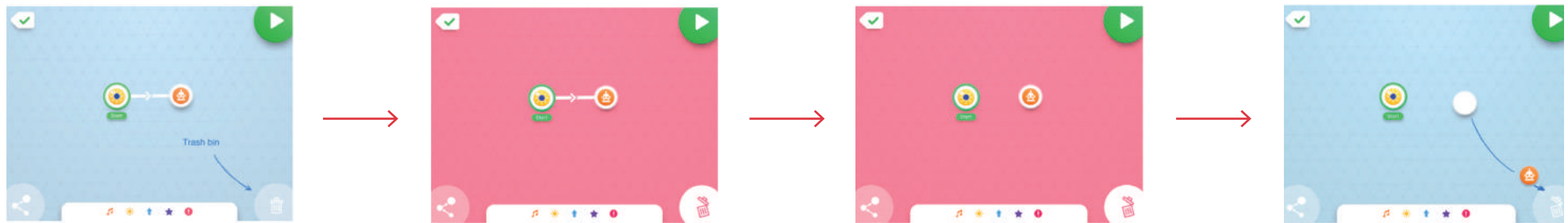
- **Turn the Connection mode on:** Click on the Connection button, the Programming screen will turn green;
- **Draw the Transition / Connection** From the center of one Actions to the centre of another Action; pay attention to the direction of the arrow you draw;
- **Turn the Connection mode off:** Click on the Connection button once again or simply tap the Programming screen, the Programming screen will turn blue again.



## 4. Robo Code App: Programming Logic

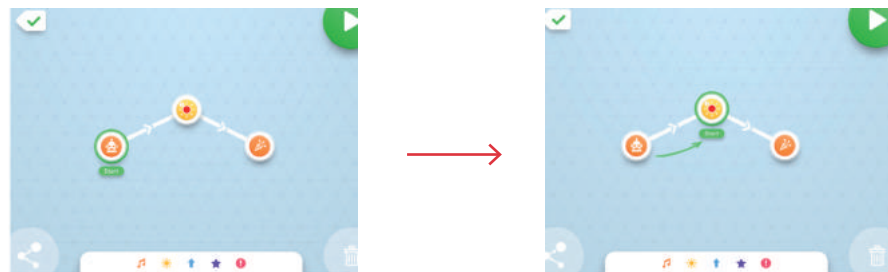
### 3. Delete the Action or Connection / Transition:

- **Trash bin button:** Click on it; the Programming screen will turn red; this means that you are in the **Delete mode**; delete the Action or Connection/Transition you want. Don't forget to turn the Delete mode off! Or:
- Simply **drag and drop** the Action you want to delete **into the Trash bin**.



### 4. Start point:

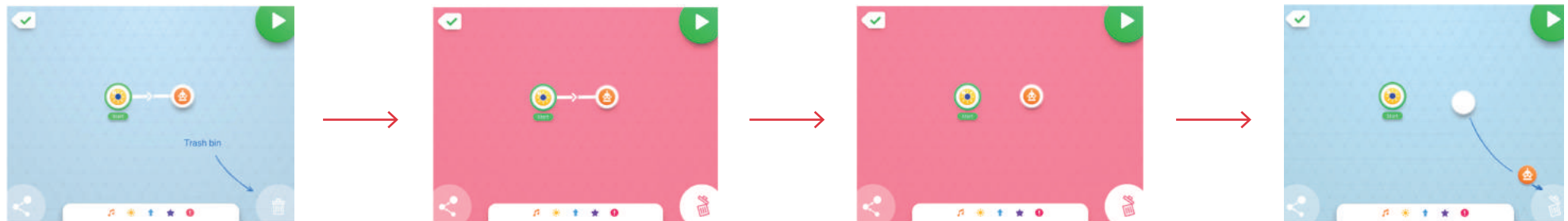
- The green circle around one of the Actions means that the **code starts here**;
- You can **move the Start point** from Action to Action in order to start your program from different Actions.



## 4. Robo Code App: Programming Logic

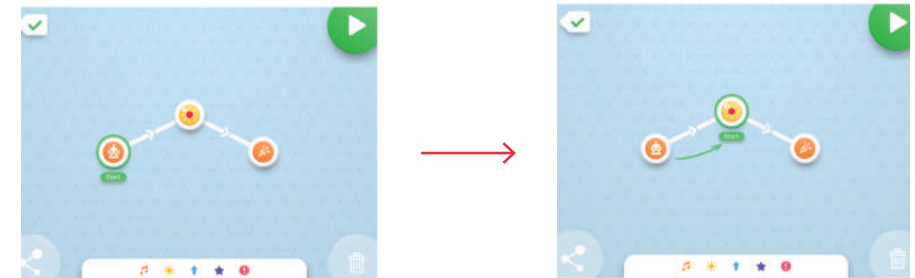
### 5. State or Parallel Actions

- Drag one Action onto the other and they will **combine** into one big bubble, or **State**; the Actions will take place at the same time, parallel to one another;
- **Note:** only 5 different Actions can be put in one State; two actions of the same type (for example, two Sounds) cannot be parallel.



### 6. Conditions:

- Conditions look like pink stickers; you need to place them on the Condition/Transition between 2 Actions;
- The **Conditions** are special clauses; each of them **makes the Connection/Transition between Actions happen IF / BECAUSE** a certain condition causes it.



**7. Use Play button** to try out your program and **Stop button** to stop it.

- **Note:** you can not make any changes in your program when it's running, stop it before making changes in the code!

**8. Save button:** Press it to **save your project**; you will be automatically returned to the Project screen.

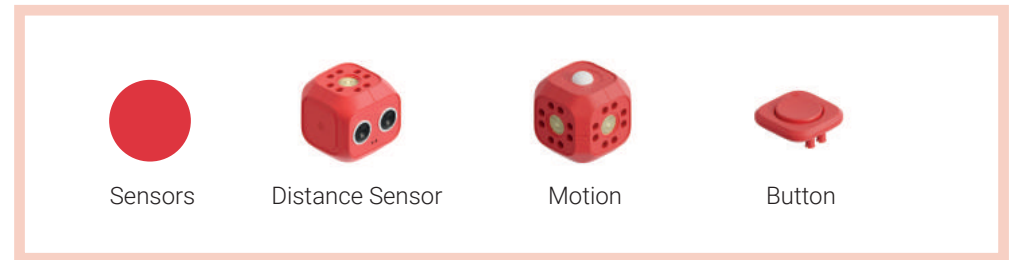
# 5. Robo Wunderkind Modules

**Robo Wunderkind Modules** are **different colors**: Each expresses a block's functions and is related to the colors of Actions in the Robo Code App, and the controller in the Robo Live App; makes building, coding, and controlling Robo more intuitive for students.

## RW Modules' Overview:



System + Sound      Main Block



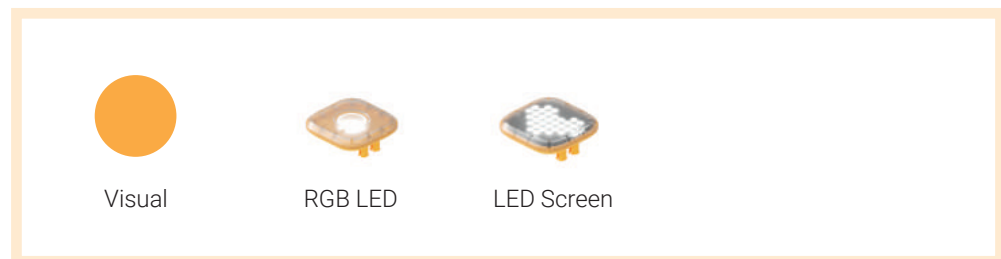
Sensors      Distance Sensor      Motion      Button



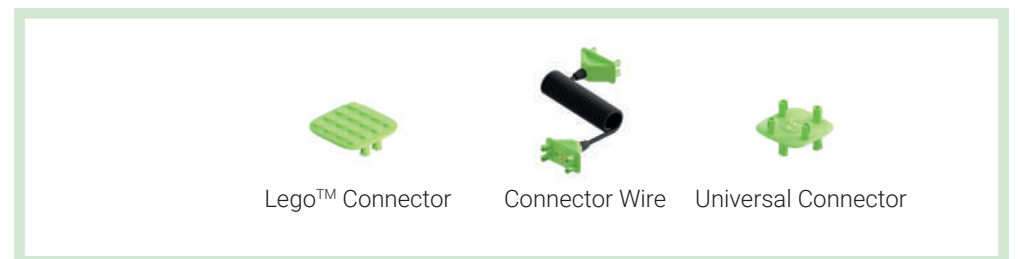
Motors      DC Motor      Servo Motor



Functional      Connector Cube      Big Wheel      Small Wheel

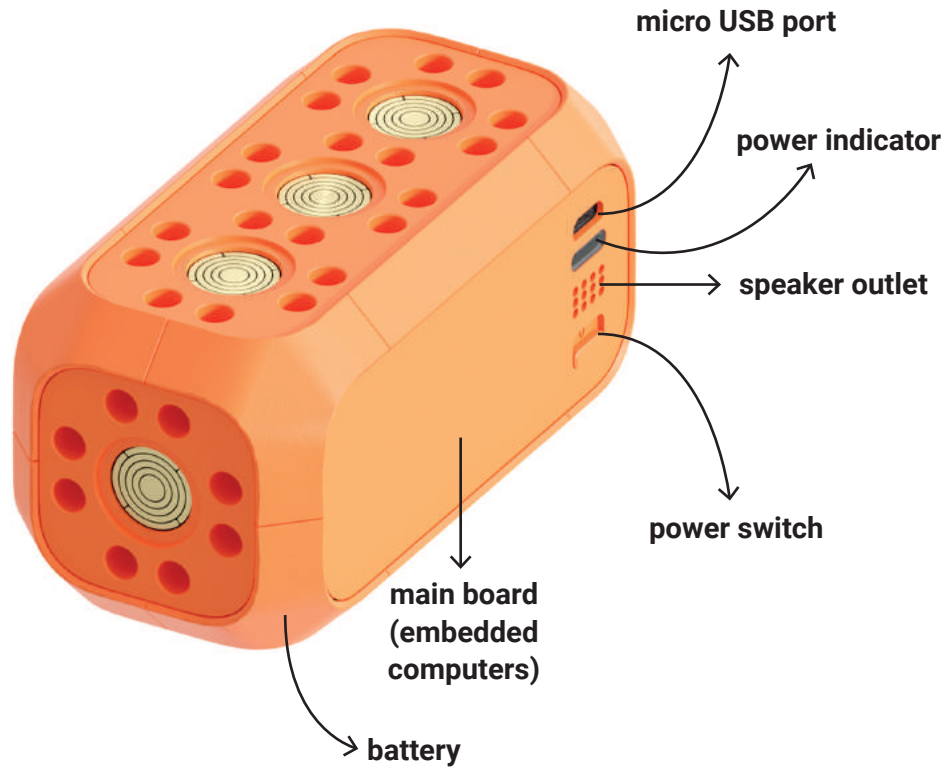


Visual      RGB LED      LED Screen



Lego™ Connector      Connector Wire      Universal Connector

# 5. Main Block



## Function:

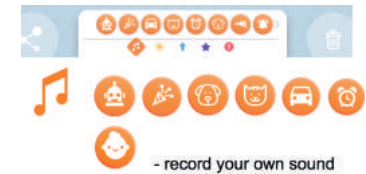
- The “brain” of Robo: contains an embedded computer, battery, accelerometer and speaker outlet inside;

## How to build:

- Always start to build your robot with the Main Block in order for all other modules to work;

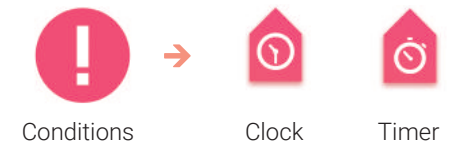
## Robo Code App:

- Sounds

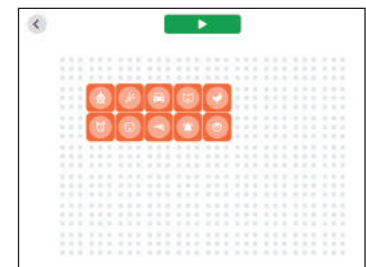


## Robo Live App:

- Conditions

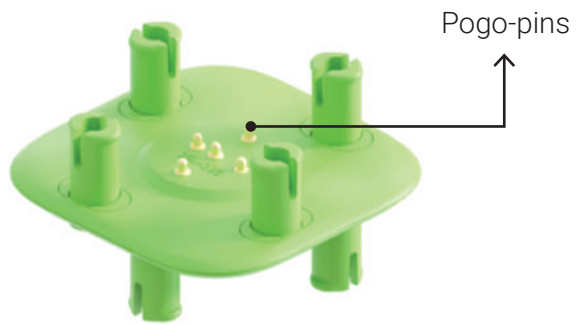


## Robo Live App:



# 5. Connectors

## Universal Connectors:



## Connector Block:



## Function:

- **To make all the modules work,** you need the electricity and the code signals from the Main Block, which passes through the small **Pogo-pins** on **Connectors** to **Modules' faces**.

## Wired Connector:

- To connect Modules on a distance or in a special case.



## Disconnecting Tool:

- Use the Disconnecting Tool in order to disconnect the modules safely.



## 5. Connectors

### How to assemble the connectors

1. The connectors need to be assembled properly, with the **sides parallel to the modules' sides;**

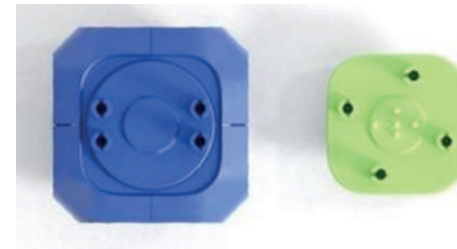


2. Some of the modules, such as two DC Motors and the Servo Motor, **do not have Pogo pins;**

One side of these modules consists of **a rotating part** that is made out of plastic, so the electricity doesn't flow through it;



3. The module that is attached to the moving part of the Motor **doesn't provide electricity**, therefore it **will not communicate with the Main Block;**



4. Use the **Wired Connector** to connect the module you need to work with the Main Block directly. Can be attached to any active connection face (any block).



# 5. RGB Light



### Function:

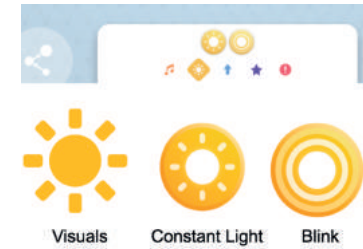
- Program or control different light signals.

### How to build:

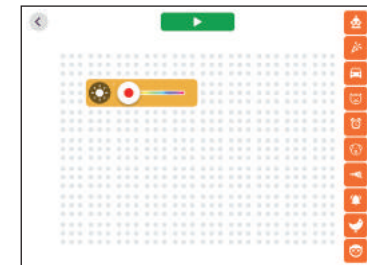
- Attach the GRB Light to any part of Main Block or any other Module.



### Robo Code App:



### Robo Live App:





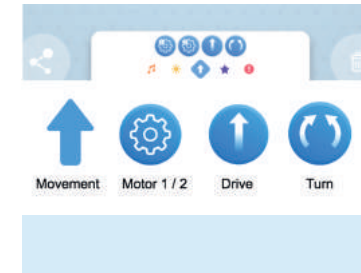
# 5. DC Motors



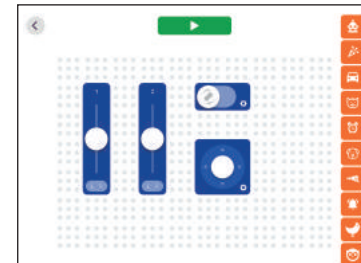
### Function:

- The DC Motors **rotate around themselves**, so you can control or program a robot to drive around;
- You can control or program each of the DC Motors independently, or both of them in the same time.

### Robo Code App:



### Robo Live App:



### How to build:

1. Attach the **rotating part** of the **DC Motor** (without Pogo pins) to the Wheel or a module you want to be rotated;
2. Attach the **DC Motor** to the other module by the **Connector** with the **Pogo-pins**.



# 5. Servo Motor



## Ideas how to build:

1. Attach the **Servo Motor** to the Main Block or ther module by the **Connector** with the **Pogo-pins** in order to work;
2. If you attach the **rotating part** of the **Servo Motor** (without Pogo pins) to the module, use the **Wired Connector** in order to make this module work, too.



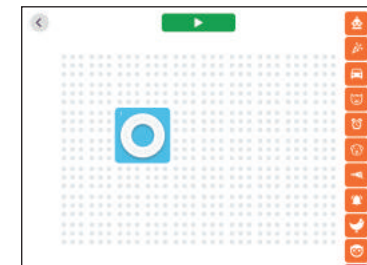
## Function:

- **Servo Motor** turns to the **exact angle** from the **zero position**;
- When you attach the Servo Motor to the other module, you **set the zero-position** for it.

## Robo Code App:



## Robo Live App:



# 5. Button



## Function:

- Detects, clicks and presses.

## How to build:

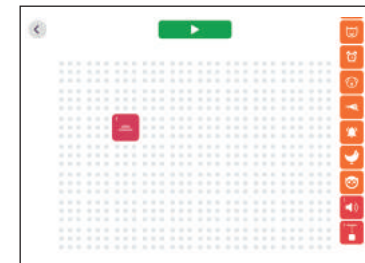
- Attach the Button to any part of Main Block, or any other Module.



## Robo Code App:



## Robo Live App:



# 5. Distance Sensor



### Function:

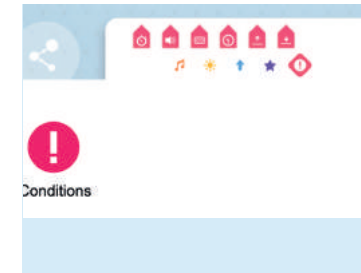
- **Distance Sensor** detects the distance before an obstacle;
- **Sound Sensor** detects the loudness of the sound.

### How to build:

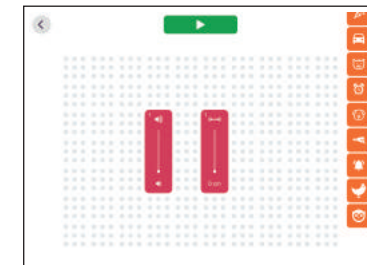
- Use the Universal connector to attach the Distance Sensor.



### Robo Code App:



### Robo Live App:



# 5. Technical Details: Main Block

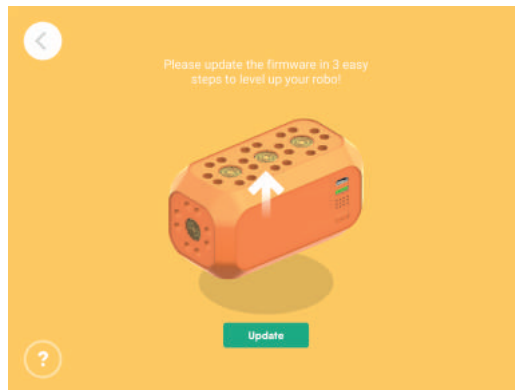
## The battery life:

- Lasts for a total of 3 hours of intensive work;
- Best to keep the charging level of the battery at 30% or higher.

## Charging:

- Plug the micro USB cable into the port at the bottom of the Main Block. The charging sound will be played once you connect the other side of the cable to a power adapter;
- Check the charging status in the App.

## Firmware Update:



## Important:

Make sure you have a stable WiFi connection and the Bluetooth is on. For a safe update process, the main block of your Robo must be plugged in. Please note that the battery percentage of your Robo should be no less than 30%, otherwise the update won't be possible.

Connect your Main Block to Robo Live or Robo Code App and follow 3 easy steps to update your Robo.

# 6. Lessons with RW Robotics Kit: Key Information

---

**Topic:** STEAM subjects; RW robotics kit can be also used as a supporting tool for the other subjects

**Student's age:** 6-12

**Group size:** 8-12 students

---



**Recommended time:**

45 – 60 min



**Recommended prior knowledge:**

Students do not need any prior knowledge, as they will learn the basic terminology and the principles of robotics and programming using Robo Wunderkind.



**Materials required:**

- Robo Wunderkind robotic kit(s)
- Tablet(s)



**Optional materials:**

- *Lego™ bricks*
- *Other materials such as colored paper, cardboard, markers etc*
- *Printed cards with the Robo Wunderkind Modules, coding Button, Actions, Conditions from the Robo Code App.*

## 6. Suggested Structure for a Lesson with Robo Wunderkind

### To Plan:

- Objective and Learning Outcomes;
- Focus on the particular module(s) and Robo Live or Robo Code App function(s);
- Key Vocabulary.

---

### Activity Stages:

#### Lead-in

7 – 10 min

- **Activate students:** Draw on previous knowledge and personal experiences of children;
- **Tell the story of Robo** to connect to the emotions of the students;
- **Set each lesson goals:** Together with your students, indicate the problem situation and motivate your students to solve it.

---

#### Guided Activity

15 – 20 min

- **Recall** some of the previous knowledge about the Robo Wunderkind robotics kit and the Apps;
- **Learn by doing:** Let students solve different challenges and gather knowledge through cooperative coding and discussions;
- **Sum up** new information before the independent activity.
- \* **Modification** (optional for advanced students) to make the lesson more challenging: 2 levels of complexity.

---

#### Independent Activity

15 – 20 min

Own project created on the basis of the learned skills.

- **Set the concrete tasks** for the own project;
- **Independent work:** Students work individually / in pairs / in small groups;
- **Presentation** of own projects for the class.

---

#### Final Reflexion & Feedback

5 – 7 min

Ask your students about the tasks they have solved, the projects they have created, and the attitudes and emotions they experienced during the lesson.

---

#### Clean up

1 – 3 min

Teach students to take care of devices they use: Turn the power off on the orange Main Block, take it apart, and put all modules of Robo back into the boxes, lock, and carefully collect all the tablets.

**Expected time of each lesson: 45 – 60 min**

# 7. Key Vocabulary for Teachers

---

<b>Robot –</b>	a machine capable of carrying out a complex series of actions automatically. Robots can be guided by an external control device – such as a remote controller – or they can be programmed previously. Robots are created by people to help with many different tasks which may sometimes be too complex, too dangerous, or simply too boring to do ourselves.
<b>Robotics –</b>	an interdisciplinary branch of engineering and science utilized in order to deal with the design, construction, operation, and use of robots, as well the control, sensory feedback, and information processing of computer systems.
<b>Engineering –</b>	the process of creating and building technological solutions and products by using math and science. An engineer is a person who does the engineering. <b>Engineers</b> solve problems with their inventions. There are several branches of engineering.
<b>Electricity –</b>	a type of energy that can build up in one place or flow from one place to another; it is used to make many different electronic devices work. <b>Robo Wunderkind</b> modules are powered by a battery inside the Main Block. Power is passed from module to module through the <b>Connectors (Pogo-Pins</b> on them) and <b>Faces</b> on each module.
<b>Pogo-pins –</b>	small metallic pins on the Connectors which help to establish a connection between two Robo Wunderkind Modules;
<b>Algorithm –</b>	tep-by-step solution of one complex task; each step is a clear instruction. A simple example of an algorithm is a cooking recipe, where you have one by one instructions in order to cook one final dish.
<b>Program code –</b>	a set of instructions which tells a computer what to do; a sequence of short commands, one after another or parallel one to another.
<b>Programming – language</b>	In order for you to communicate with a computer (and to get it to execute your instructions) you must speak its language. There are a number of different programming languages, some are very complicated while others are similar to spoken English. In the <b>Robo Code App</b> , we use a special visual coding language. There are different <b>Actions</b> that look like small bubbles, while bigger bubbles – <b>States</b> or parallel Actions, the arrows – Connections / Transitions between them, and Conditions, which look like stickers to put on Connections / Transitions.

---



## 7. Key Vocabulary for Teachers

These 3 terms (**Algorithm, program, programming language**) are interconnected. To help students understand them better, we can say that:

- When we have one complex task, we can break up it into a set of smaller, individual instructions – create an **algorithm**;
- We can use a **programming language** to write those instructions in the language computer understands – thus, we create a code.

---

### Condition –

a conditional statement tells a program to execute different actions depending on whether a condition is true or false; In **Robo Wunderkind visual programming language** Conditions look like pink stickers to place them on the Condition/Transition between 2 Actions. Each of them makes the Connection/Transition between Actions happen IF / BECAUSE a certain condition causes it. Example: “If you click the Button, Connection/Transition happens”.

---

### Computer –

a device for working with information. The information can be numbers, words, pictures, movies, or sounds. Computer information is also called data. Computers can process huge amounts of data very quickly. They also store and display data. People use computers every day: at work, at school, and at home. Computers are used in factories to control how things are made and in offices to keep records, for example.

There is a small computer inside the **Robo’s Main Block**, which sends and receives the signals from the tablet and processes the information so that we can control it. This is why the Main Block always has to be in every project in order for all other modules to work!

---

### Signal –

a type of electrical communication between electronic devices. In **Robo Wunderkind robotics kit** the Main Block sends signals to other modules using wired communication from the center of each cube face. The Robo communicates with the smart device using **Bluetooth** wireless communication.

---

### Bluetooth signals – intelligence

is a wireless technology standard for exchanging data over short distances.

## 7. Key Vocabulary for Teachers

---

<b>Sensor –</b>	a device that receives a physical signal or stimulus (as heat or pressure or light or motion etc.) and responds to it in a distinctive manner.
<hr/>	
<b>Smart device –</b>	an electronic device, generally connected to other devices or networks (via Bluetooth for example) and can operate interactively and autonomously to some extent.
<hr/>	
<b>Accelerometer –</b>	an instrument for measuring acceleration (process of moving) or for detecting and measuring vibrations. In <b>Robo Wunderkind robotics kit</b> , there is the accelerometer inside the <b>Main Block</b> which allows it to react in certain situations, such as if you take the Main Block up or put it down (program Pick Up and Put Down conditions).
<hr/>	
<b>Decibel –</b>	a unit of sound intensity which is used to measure how loud a sound or signal is. In <b>Robo Wunderkind robotics kit</b> , the Distance Sensor has also the sound sensor inside that allows it to react to the sounds you make (program Sound condition).
<hr/>	
<b>Artificial – intelligence</b>	the capability of a machine to imitate intelligent human behavior (can think, solve problems and learn new things)