robo™ wunderkind

Curriculum

Block #1: Introduction to Robotics and Computer Science.
Theme topic: Robo's first steps.

© 2019 by Robo Technologies GmbH, Vienna, Austria

This work is licensed under the Creative Commons Attribution 4.0 International License. To view a copy of this license, visit http://creativecommons.org/licenses/by/4.0/ or send a letter to Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.

Welcome to the Robo Wunderkind Curriculum!

We are happy to introduce you to the first block of lessons from Robo Wunderkind robotics and coding program. **Block #1: Robo's first steps** consists of **10 lessons** which will introduce your students to the **basics of robotics and computer science** through personal experiences and cooperative play with Robo Wunderkind modules and Robo Code App. Our ready-to-use curriculum is made to fully support you during the preparation and entire course of teaching with RW. You can join your students in playing with Robo: learning, coding, and imagining together!

Included in this ready-to-use curriculum:

- All the **key information** and **details** to organize the lessons;
- **Concepts Overview** as well as formulated **Learning Outcomes**;
- **10 Lesson Plans** – well-constructed and easy to follow, with **Additional Activities** that allow you to adjust the complexity level of lessons to specific student needs;
- **Supporting materials** such as **teacher slides, worksheets, key vocabulary, and printable modules' images** to make your teaching comfortable.

Table of Contents:

Chapter	Page
1. Key Information	3
2. Robotics and Computer Science Concepts Overview	4
3. Learning Outcomes	6
4. Suggested Structure for a Robo Wunderkind Lesson	8
5. Detailed Lesson Plans: Lessons 1-10	10
6. Supporting Materials:	
• Key Vocabulary for Teachers	40
• Coding Buttons and Icons to Print	45

Key Information

Topic: STEAM subjects **Grades:** 1–4 **Group size:** 6-12 students

All the lessons in Block #1 are **story based** and linked by one topic: **Robo's first steps.** In this story, Robo is a small robot that visits the classroom. Students will help Robo to learn new skills and discover the world around it. By doing so, they will learn new concepts in robotics and computer science, build their own robots and discover the **Robo Code App** with its unique **Visual Based Programming language** to make their first State-Machine Based Programs.

Complexity: Each lesson includes the basic level as well as a possible modification – additional activity. This opens the possibility of adapting the complexity level of the lessons to the specific needs of your students.

Recommended Prior Knowledge: Students do not need any prior knowledge, as they will learn the basic terminology and the principles of robotics & controlling robots using Robo Wunderkind robotics kit and Robo Code App.

Materials Required:

- Robo Wunderkind robotics kit(s);
- Tablet(s);
- Some materials to customize robots and create an environment: Lego™ bricks, colored paper, cardboard etc.;
- Supporting materials: Printable Module images and "Robo's Diary" — student working sheets.

Robotics and Computer Science Concepts Covered in Block #1

robo™ wunderkind

Concepts	Lesson 1	Lesson 2	Lesson 3	Lesson 4	Lesson 5	Lesson 6	Lesson 7	Lesson 8	Lesson 9	Lesson 10
Robotics										
1. Robotics, Engineering	+	+	+	+	+	+	+	+	+	+
2. Electricity:										
• Electrical Power	+	+	+	+	+	+	+	+	+	+
• Local Communication	+	+	+	+	+	+	+	+	+	+
3. Wireless Communication	+	+	+	+	+	+	+	+	+	+
4. Remote Control	+									
5. Design Thinking Process:										
• Controls Design	+									
• (Mechanical) Design			+	+	+	+	+	+	+	+
• Code Design		+	+	+	+	+	+	+	+	+
6. User Input: Using the Controls (Sounds, Light, Motors, Servo Controls)	+									
7. Outputs: Functions of Modules										
• Outputs: Sounds		+				+		+	+	+
• Outputs: (RGB) Light			+			+		+	+	+
• Outputs: DC Motors				+	+	+	+	+	+	+
• Outputs: Servo Motor							+	+	+	+

Robotics and Computer Science Concepts Covered in Block #1

robo™ wunderkind

Concepts	Lesson 1	Lesson 2	Lesson 3	Lesson 4	Lesson 5	Lesson 6	Lesson 7	Lesson 8	Lesson 9	Lesson 10
Computer Science										
1. Programming, Code		+	+	+	+	+	+	+	+	+
2. State-Machine Based Programming:										
• Action		+	+	+	+	+	+	+	+	+
• Connection		+	+	+	+	+	+	+	+	+
• State						+		+	+	+
3. Sequential Logic		+	+	+	+	+	+	+	+	+
4. Loop		+	+	+	+		+			+
5. Parallel Execution						+		+	+	+
6. User Input: Parameters			+	+	+	+	+	+	+	+
7. Digital Literacy	+	+	+	+	+	+	+	+	+	+
8. Engineering Design Process									+	+

Learning Outcomes Related to Robotics and Computer Science with RW Robotics Kit:

I. Robotics:

- Understand and follow the classroom rules of using technology – **Digital literacy**;
- Understand what **robots** are and their functions in everyday life; what **robotics** is;
- Understand what **electrical power** and **wireless communication** (Bluetooth) is and why robots need it;
- Understand how people **control robots:** the difference between remote control and programming;
- Understand what **Mechanical Design** and **Code Design** is and can use it to create a Robo-project;

Robo Wunderkind Robotics Kit:

- Know some of **Robo's Modules** and **Connectors**, and understand their functions – **Outputs**;
- Know what the **Main Block** is, understand its functions; can explain why it is necessary for every project;
- Understand and take into account the general **logic of building robots** with Robo Wunderkind robotics kit;
- Can combine modules to consider the functions required in order to **build a robot for concrete purposes** – use Mechanical Design;

Robo Live App:

- Know and can use the **Robo Live App:** can connect the Main Block, create a new project, add controllers and manage them;

II. Computer Science:

- Understand what **programming and program** is;
- Understand the terminology of **State-Machine Based Programming** such as **Action, Connection, State, Loop** and can use **Visual Based Programming language** of Robo Code App to create a simple program;
- Understand the difference between a **sequential logic** program and a program with **Parallel execution**; can create both;
- Understand what parameters of Actions – **User Input** are and their function for the robot's performance;
- Understand and can follow the **Engineering Design Process** in order to create a Robo-project;

Robo Code App:

- Know and can use the **Robo Code App:** can connect the Main Block, create a new project; use the Menu;
- Recognize and can use the **Visual Based Programming language** in Robo Code App to create a simple program in order to **solve the set challenge(s)**.

Cognitive and Behavioral:

- Can pay **attention** to the information needed to complete certain tasks (Sustained, Selective, Alternating, and Divided attention);
- Can maintain **concentration** during the time allotted for completing tasks / projects;
- Can **memorize** information for short-term tasks, as well as for long-term periods;
- Can use **spatial thinking** in order to assemble robots which work in each certain way;
- Can use the **logical and algorithmic thinking** in order to control and code the robot in a specific way;
- Can **solve the problem** using critical thinking skills: set the goal, plan, action, reflect, recreate, evaluate, accept criticism;
- Can use **imagination and creativity** to create own project;
- Can **work in pairs / small groups** and use **social skills** to create a common project;
- Can **present** the created project to the class; can receive and give **constructive feedback** on the project.

Suggested Structure for a Project with Robo Wunderkind Robotics Kit

robo™ wunderkind

To Plan:

- Focus on the particular concepts in relation to RW Modules, Robo Live or Robo Code App;
- Objectives and Learning Outcomes; Key vocabulary;
- Printable supporting materials.

Activity Stages: **8 Steps**

Lead-in *7 – 10 min*

1. **Activate students:** draw on previous knowledge and personal experiences;
2. **Analyze:** tell Robo's Story to connect with the emotions of the students, **identify** the problem situation, discuss it and **come up with** a theoretical solution together.

Guided Activity *15 – 20 min*

3. **Get ready: recall** some of the previous knowledge about the RW robotics kit and the Apps; **Connect** the Main Block to the tablet and create a new project;
4. **Learn by doing – build and program together:** let students solve different challenges and gather knowledge through cooperative play and discussions;
5. **Sum up** new information before the independent activity.

Independent Activity *15 – 20 min*

6. **Make your own project:** students work individually / in pairs / in small groups to create their own project.
   The teacher provides struggling students with one-on-one assistance. They can use different materials in order to create an environment and customize robots. Students can also cooperate to make a shared project with two or more robots.
   Presentation (optional): Students present their projects to the class and give constructive feedback to each other.

(*) ***Additional activity:*** *additional task to increase the complexity of the lesson for more advanced students.*

Reflection & Feedback *7 – 10 min*

7. **Sum up** the learned information. **Receive feedback** about the complexity of tasks and activities.
8. **Clean up:** 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

Projects' Overview

Projects	Concepts	Complexity	Page
1. Meet Robo!	Robotics, Engineering; Mechanical Design	☆	10
2. Program Your First Robo!	Programming, a Code; Sequential Logic, Action, Connection, Loop	☆	13
3. Robo Lights Up the Classroom	Sequential Logic, Action, Connection, Loop	☆	16
4. Robo-car Drives Around	Drive and Turn Actions (Settings: Distance, Angle, Speed)	☆	19
5. Robo's First Journey	Parallel Execution, State (+Action's Lifespan), Mechanical design	☆☆	22
6. Robo Travels to Toytown	Mechanical Design and DC Motors Use	☆☆	25
7. Robo Looks Around	Motors: Servo vs. DC Motor – Distance vs. Angle; Mechanical Design	☆	28
8. Robo Meets Friends	Parallel Execution, State (+Action's Lifespan), Mechanical Design	☆☆	31
9. Robo Makes a Surprise	Engineering Design Process, Mechanical and Code Design	☆☆☆	34
10. What Does Your Robo Do?	Engineering Design Process, Mechanical and Code Design	☆☆	37

Project 1: Meet Robo!

Concepts: Robotics, Engineering, Mechanical Design

Complexity: ★☆☆

Lesson #1 is an introduction to the Robo Wunderkind Robotics Kit. It is perfect for students who do not know Robo and for those who already went through the Elementary Projects. Use different approaches: 1) If the topic is new in your class, guide students through the lesson in order to grasp a new knowledge; 2) If your students are already familiar with Robo, let them work more independently to master their knowledge through practice and cooperative play.

Robo's Story:

Today, we have a special guest in our class! This is Robo, a smart robot that has come to our class to be our friend and learn with us!

Lesson Goal:

We will become engineers in order to assemble and control our first robot!

Project 1: Meet Robo!

Modules:

Main Block

DC Motors

Servo

RGB LED

Button

Distance Sensor

Wheels

Small Wheel

Connectors

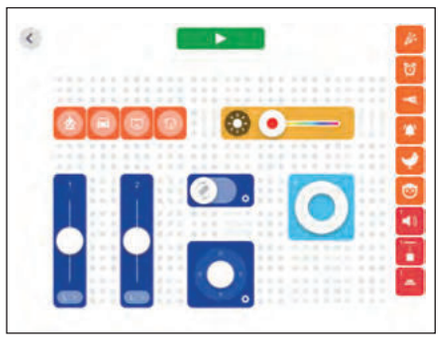
Connector Flat Lego™

Connector Wire

Connector Block

Control:

Use Robo Live App to control modules and see how they work.



Focus:

- **Robotics:** robots and their functions in everyday life, remote control as a way of controlling robots;
- **RW robotics kit:** Modules, Connectors, and their functions.

Objectives:

- To assemble the robot using different Modules and Connectors, and considering their functions;

Learning Outcomes:

- I can explain why every robot needs the Main Block;
- I know some of Robo's modules and their functions;
- I can comprehend how to combine modules and build a robot;
- I can use Robo Live App to control different modules.

Key Vocabulary:

- Robotics, Engineering, Robot, Main Block, Connectors, Disconnecting tool.

Additional Materials:

- Printable Modules' images;
- Worksheet #1;
- *Optional: Lego™ bricks, colored paper, and/or other materials.*

Project 1: Meet Robo!

Activity Stages:

Lead-in *7 – 10 min*

1. **Ask:** Do you know what a robot is? Why do people create robots? Who has experience with using robots? How do people control robots? Who studies robots? **Discuss** the terminology: **Engineering, robotics, robot.**
2. **Analyze:** Tell Robo's Story, **identify** the problem situation and **come up** with a theoretical solution.

Guided Activity *15 – 20 min*

3. **Open** and **explore the boxes with Robo:** briefly examine the modules.
4. **Discover the Robo Live App:** **Use the worksheet #1.**
   - **Discuss** the need of tablet to control Robo; hand out the tablets and discuss the **rules for tablets use;**
   - Switch on the Main Block and connect it to the **Robo Live App;** discuss it's interface;
   - **Connect Modules** to the Main Block and see how they appear in the App; disconnect them; discuss the Connectors and Disconnecting tool;
   - Use **Controls** to control Modules and discuss how they work;
   - **Discuss** the term **design** and how different builds of Robo influence its functions.
5. **Sum up** new information before the independent activity.
   Note: Students do not need to know all the modules by the end of the lesson. Rather, allow them to explore the Modules by touching, connecting, and controlling them to see how they work. **Only pay attention to the Main Block and its function.**

Independent Activity *15 – 20 min*

6. **Make an own project:** Build any variation of Robo and control it using the Robo Live App. Share the story of why you created this particular robot and how it will help you in your life. Use some materials to customize your Robo.

(*) ***Additional activity:*** *Explore* ***technical details*** *and* ***design*** *of Robo's Modules. Discuss embedded* ***computer*** *and the* ***battery*** *inside the Main Block,* ***Pogo-Pins*** *on Connectors and their functions; DC Motors and Servo Motor, their rotating parts without Pogo-Pins. Try to connect Motors differently and see if they appear in the Robo Live App or not.*

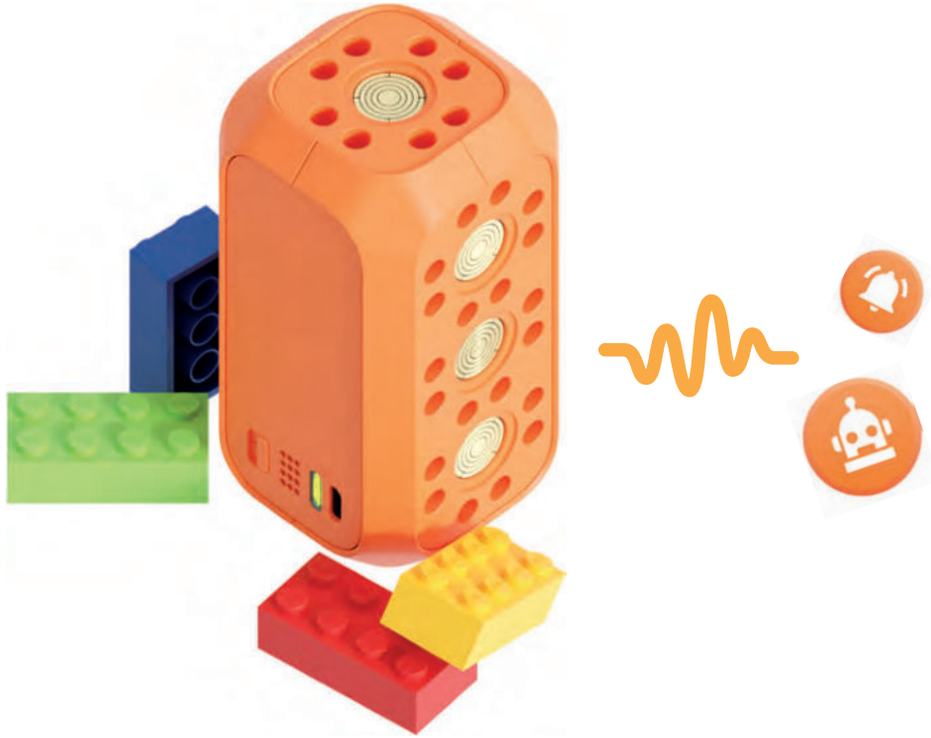
Reflexion & Feedback *5 – 7 min*

7. **Sum up:** some of the Robo's Modules and their functions, the projects students have created using them. **Receive feedback:** Were the tasks easy or complex? Interesting or boring? Which part of the lesson was the most interesting? Why?
8. **Clean up:** Teach students to take care of the devices they use: RW Modules and the tablets.

Project 2: Program Your First Robo!

Concepts: Programming, Code; Sequential Logic;
RW Visual Based Programming: Action, Connection, Loop

Complexity: ★☆☆



Robo's Story:

Our new friend Robo is very excited to meet us and wants to say "Hi" to us. But can robots do something like this by themselves? In this special world, our Robo Code App lets us create programs for our robot to perform.

Lesson Goal:

To help Robo make sounds, we will become programmers and create our first program code in the Robo Code App.

Project 2: Program Your First Robo!

Modules:

Main Block

Program:

Sounds

Connect

Focus:

- **Computer Science:** programming as a way of controlling robots; the importance of coding in everyday life;
- **Robo Code App:** Action, Connection, Loop.

Objectives:

To create a simple sequential logic program that includes Sounds and Connections between them; then modify it to form a loop.

Learning Outcomes:

- I understand what a program is;
- I recognize the coding buttons and Actions – Sounds in the Robo Code App;
- I can connect two or more Sound actions together to make a simple sequential logic program;
- I can modify a code to form a loop;

Key Vocabulary:

- Programming, code;
- Project screen, Programming screen, Action, Start point, Play button, Connection button, Connection, Trash bin, Loop;

Additional Materials:

- Worksheet #2;
- Printable images: Main Block, coding Actions and buttons;
- *Optional: Lego™ bricks, colored paper or other materials.*

Project 2: Program Your First Robo!

Activity Stages:

Lead-in *7 – 10 min*

1. **Ask:** Who is our new friend? What did we create? How can people control robots? What is the difference between remote control and programming? Have you tried to program before? What is programming? What is a program? **Discuss** the terminology: **Programming, code.**
2. **Analyze:** Tell Robo's Story, **identify** the problem situation and **come up** with a theoretical solution.

Guided Activity *15 – 20 min*

3. **Open the boxes** and **recall** the function of the Main Block and its importance.
4. Explore the **Robo Code App** and program **Sound actions:** Teach your Robo to speak to you! **Use the worksheet #2.**
   - **Connect** the Main Block to the App and **discuss:** My Robo menu, Coding Lab screen and New Project button, Programming screen, Actions Dock;
   - Find Sounds in the **Action Dock:** discuss the color of the Actions;
   - Program the **first Sound** and discuss: **Start point, Play button;**
   - Program the **second Sound:** discuss how and why to change a **Start point;**
   - **Connect two Sounds:** learn about **Connection button,** connection mode, and **Connections** between the Actions;
   - **Delete** one Sound or Connection: learn about **Trash bin** and delete mode;
   - **Program a loop:** Modify a code to form a loop and discuss the term.
5. **Sum up:** Main Block and its functions, Robo Code App interface; the role of robots in everyday life.

Independent Activity *15 – 20 min*

6. **Make an own project:** think about how Robo which makes Sounds can help you in your life; build any variation of Robo and program Sounds for this particular situation. Use some materials to customize your Robo.

(*) ***Additional activity:*** *Learn about* ***programming languages.*** *Ask: What is programming language? Do you know any of the programming languages? What does Robo Code App's programming language look like? Discuss the terminology: Action, Connection, Transition, Loop.*

Reflexion & Feedback *5 – 7 min*

7. **Sum up:** What is programming, code; Main Block and its function; Sounds actions and Connections, Sequential code, Loop; and the projects students have created using them. **Receive feedback** about the complexity of tasks and activities.
8. **Clean up:** Teach students to take care of the devices they use: RW Modules and the tablets.

Project 3: Robo Lights Up the Classroom

Concepts: Sequential Logic; RW Visual Based Programming: Action, Action's Settings

Complexity: ★☆☆

Robo's Story:

Robo is very happy to be your friend and wants to throw a small party with colorful lights! Can we program our Robo to shine and color up the classroom?

Lesson Goal:

To help Robo to throw a party, we will build a shining Robo and program it to light up in different colors.

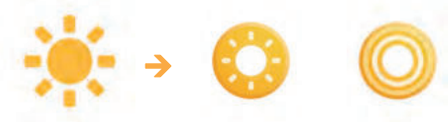
Project 3: Robo Lights Up the Classroom

Modules:

Main Block

RGB LED

Program:



Visuals → Constant Light → Blink

Focus:

- **Robotics:** Output – RGB Light and its function;
- **Computer Science:** User input – parameters of Actions.

Objectives:

- To create a simple sequential logic program that includes two types of Visuals with different settings and Connections between them.

Learning Outcomes:

- I can combine modules to build the Robo which emits lights;
- I can use the Robo Code App to create a new project;
- I can create a simple sequential logic program that includes Visuals: Constant Light and Blink actions;
- I can change settings for Constant Light and Blink actions.

Key Vocabulary:

- Programming screen, Actions Dock, Action, Connection; Visuals: Constant Light action, Blink action.

Additional Materials:

- Worksheet #3;
- Printable images: Main Block, RGB Light, Visuals;
- *Optional: Lego™ bricks, colored paper or other materials.*

Project 3: Robo Lights Up the Classroom

Activity Stages:

Lead-in *7 – 10 min*

1. **Ask:** What did we create last time? What program is? What did we program last time and why? Which module did we use and why?
2. **Analyze:** Tell Robo's Story, **identify** the problem situation and **come up** with a theoretical solution.

Guided Activity *15 – 20 min*

3. **Connect** the Main Block to the tablet and **recall** the Robo Code App interface: My Robo menu, Coding lab, Programming screen, Action Dock.
4. **Build and program:** Teach your Robo to emits different lights! **Use the worksheet #3.**
   - Find and attach **RGB Light** to the Main Block;
   - Find **Visuals** in the **Action Dock:** discuss the color of the Actions in relation to the RGB Light module;
   - Program the **Constant Light action;** try different **settings:** the color, the time, the brightness;
   - Program the **Blink action;** try different **settings:** the color, the number of times, the speed;
   - Compare: the **Blink action** and the **Constant Light action;**
   - **Play around:** Program the **Visuals** with different **settings** and **Connections** between them. **Discuss:** Talk about the action's **settings** and how they influence Robo's performance. Turn the light in the classroom off and throw a small **colorful party** together!
5. **Sum up:** Constant Light and Blink actions and their settings; how Robo with RGB Light can help in your life.

Independent Activity *15 – 20 min*

6. **Make an own project:** think about how Robo with RGB Light can help you in your life; build any variation of Robo and program Visuals for these particular situations. Use some materials to customize your Robo.

(*) ***Additional activity:*** *Learn about the Action's* ***lifespan*** *and the* ***infinity*** *symbol. Ask: What is a lifespan? What is the difference between finite and infinite lifespan? How does it influence the code and Robo's performance?*

Reflexion & Feedback *5 – 7 min*

7. **Sum up:** Visuals – Constant Light and Blink actions and their settings; Sequential code, Loop; and the projects students have created using them. **Receive feedback** about the complexity of tasks and activities.
8. **Clean up:** Teach students to take care of the devices they use: RW Modules and the tablets.

Project 4: Robo-car Drives Around

Concepts: Mechanical Design; Sequential Logic;
RW Visual Based Programming: Action, Connection, Loop

Complexity: ★☆☆

Robo's Story:

Robo is a very curious creature who likes to discover the world around! Robo wants to travel, but first Robo needs to learn how to drive. As we know, robots can do many things but only if we program them!

Lesson Goal:

To help Robo to drive around, we will build and program a Robo-car.

Project 4: Robo-car Drives Around

Modules:

Main Block

DC Motors

Wheels

Small Wheel

Connectors

Program:

Movement

Drive

Turn

Focus:

- **Robotics:** Output – DC Motors and their function;
- **Computer Science:** User input – parameters of Actions.

Objectives:

- To create a simple sequential logic program that includes some of the Movement: Drive and Turn actions and Connections between them; then modify it to form a loop.

Learning Outcomes:

- I can combine modules to build the Robo-car which can drive;
- I can create a simple sequential logic program that includes some of the Movement: Drive and Turn actions;
- I can change settings for Drive and Turn actions;
- I can modify a code to form a loop.

Key Vocabulary:

- Actions Dock, Action, Connection; Movement: Drive and Turn actions, Loop; Design.

Additional Materials:

- Worksheet #4;
- Printable images: Main Block, DC Motors, Movement: Drive and Turn actions;
- *Optional: Lego™ bricks, colored paper or other materials.*

Project 4: Robo-car Drives Around

Activity Stages:

Lead-in *7 – 10 min*

1. **Ask:** Which modules did we already use? What is our Robo capable of? How can we use this in our everyday life? **Recall** the terminology: robotics, programming, code, wireless communication, design.
2. **Analyze:** Tell Robo's Story, **identify** the problem situation and **come up** with a theoretical solution.

Guided Activity *15 – 20 min*

3. **Connect** the Main Block to the tablet and **recall** the Robo Code App interface: My Robo menu, Coding lab, Programming screen, Action Dock.
4. **Build and program:** Teach your Robo to drive around! **Use the worksheet #4.**
   - Find and attach **DC Motors** to the **Main Block.** Discuss their mechanical details: **the rotating part** without Pogo-Pins, compare it to the **Connector.** Let students attach the DC Motors differently and see if they appear in the App or not. Ask: What is the right way to attach the DC Motors and why? **Conclude:** DC Motors should be attached with the help of the Connectors in order to work properly. The rotating part should be attached to the Wheels.
   - Find **Movement** in the **Action Dock:** discuss the color of the Actions in relation to DC Motors;
   - Program the **Drive action;** try different **settings:** distance and speed; reverse button;
   - Program the **Turn action;** try different **settings:** angle and speed;
   - **Play around:** program **Drive and Turn actions** with different **settings** and **Connections** between them. **Discuss:** Action's settings and how they influence Robo's performance.
5. **Sum up:** how to attach the DC Motors and why; types of Movement; their settings; which other robots in real life have motors and what their functions are.

(*) ***Additional activity:*** *Learn about* ***technical design.*** *Ask students to build some variations of Robo-car with DC Motors attached differently and try the same code for them.* ***Discuss:*** *How does the build – technical design of Robo influence its performance? What does it mean to make Robo's design more efficient?*

Independent Activity *15 – 20 min*

6. **Make an own project:** think about how Robo with DC Motors can help you in your life; build any variation of Robo and program Movement for this particular situation. Use some materials to customize your Robo.

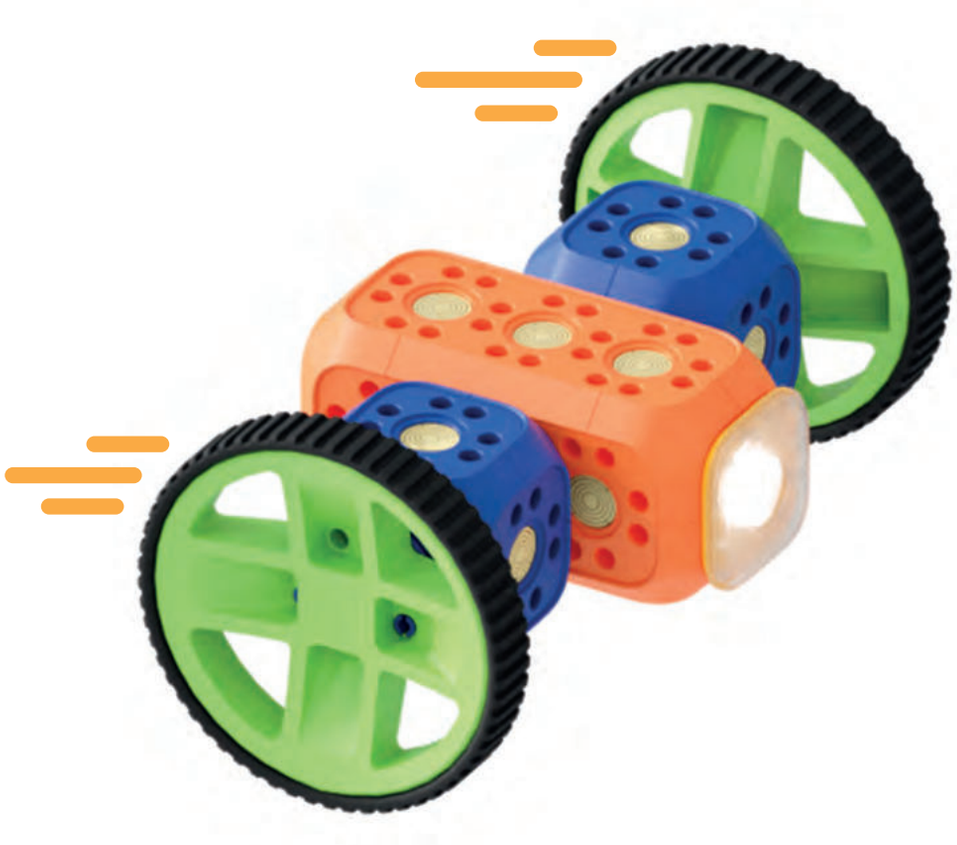
Reflexion & Feedback *5 – 7 min*

7. **Sum up:** Technical details of DC Motors; Drive and Turn actions and their settings; Sequential code, Loop; the projects students have created using them. **Receive feedback** about the complexity of tasks and activities.
8. **Clean up:** Teach students to take care of the devices they use: RW Modules and the tablets.

Project 5: Robo's First Journey

Concepts: Parallel Execution; RW Visual Based Programming: State; Mechanical and Code Design.

Complexity: ★★☆



Robo's Story:

Now it's time for Robo's first journey! Where will your Robo travel to, and why? Who will it meet? Which Actions will it need in this adventure?

Lesson Goal:

To help Robo to go to its first journey, we will build and program a Robo-traveller and combine all the Actions we had learned before.

Project 5: Robo's First Journey

Modules:

Main Block

DC Motors

RGB LED

Wheels

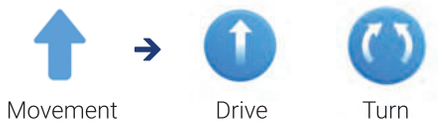
Small Wheel

Connectors

Program:

Sounds

Visuals Constant Light Blink


Movement Drive Turn

Focus:

- **Robotics:** Outputs – Sounds, RGB Light, DC Motors;
- **Computer Science:** User input – parameters of Actions; Parallel Execution.

Objectives:

- To create a program with a Parallel execution that includes Sounds, Visuals and some of the Movement: Drive and Turn actions and Connections between them.

Learning Outcomes:

- I can combine modules to build the Robo-vehicle which can drive;
- I can create a program with a Parallel execution that includes Sounds, Visuals and some of the Movement: Drive and Turn actions and Connections between them;
- I can change settings for Constant Light, Blink, Drive and Turn actions considering their place in a State.

Key Vocabulary:

- Actions Dock; Sounds, Visuals, Movement;
- RW Visual Based Programming: State, Action's lifespan.

Additional Materials:

- Worksheet #5;
- Printable images: Main Block, RGB Light, DC Motors; Sounds, Visuals, Movement;
- *Optional: Lego™ bricks, colored paper or other materials.*

Project 5: Robo's First Journey

Activity Stages:

Lead-in *7 – 10 min*

1. **Ask:** Which modules did we already use? What is our Robo capable of? How can we use this in our everyday life?
2. **Analyze: tell** Robo's story, **identify** the problem situation and **come up** with a theoretical solution.

Guided Activity *15 – 20 min*

3. **Connect** the Main Block to the tablet and **recall:** Main Block, RGB Light, DC Motors and RW Visual Based Programming language.
4. **Build and program:** prepare Robo for the first journey! **Use the worksheet #5.**
   - **Build a Robo-traveller** using the Main Block, RGB Light, and DC Motors;
   - Program a **Sequential code:** Robo-traveller drives, turns around and then makes a sound one;
   - Program a **Parallel execution:** ask students to program a specific code – Robo makes a sound each time it drives forward – simultaneously.
   - Let students come up with an idea how to make it: combine **Drive and Sound actions** in one big bubble – State. **Discuss** the terminology: **Parallel execution, State;**
   - **Add Visuals** to the State; try to add other Sounds or Movement and discuss which **types of Actions** can or cannot be used in one State and why;
   - **Program two States** and a connection between them: discuss how the transition from one State to another happens;
   - **Play around:** program different **States** with **Sounds, Visuals, Movement,** and **Connections** between them.
5. **Sum up:** Analyze the difference between the Sequential code and Parallel execution; how all these Actions can help Robo-traveler in its journey.

(*) ***Additional activity:*** *Learn about* ***code design.*** *Recall what design is, why people design things differently. Ask: can we design a code, what does it mean to design a code?* ***Conclude:*** *Design a code means to program it efficiently, so it consists of all the Actions, States, Connections we need to achieve the project's goal(s) – not less, not more; the code is clear and easy to read by others.*

Independent Activity *15 – 20 min*

6. Make your own project: decide where your Robo-traveler go to; create an environment and program Robo-traveler for this particular situation using all known Actions to create a code with the Parallel execution.

Reflexion & Feedback *5 – 7 min*

7. **Sum up:** Parallel execution, State, which Action can form a State. **Receive feedback** about the complexity of tasks and activities.
8. **Clean up:** Teach students to take care of the devices they use: RW Modules and the tablets.

Project 6: Robo Travels to Toytown

Concepts: Parallel Execution; RW Visual Based Programming: State; Mechanical and Code Design

Complexity: ★★★

Robo's Story:

Robo is invited to visit Toytown! To get there, Robo has to travel a long way avoiding obstacles and making tricky turns. Are we ready for serious programming to help Robo?

Lesson Goal:

To help Robo to get to Toytown, we will program Robo-vehicle to avoid obstacles and cope with some challenges.

Project 6: Robo Travels to Toytown

Modules:

Main Block

DC Motors

RGB LED

Wheels

Small Wheel

Connectors

Program:

Sounds

Visuals → Constant Light, Blink

Movement → Drive, Turn

Focus:

- **Robotics:** Outputs – Sounds, RGB Light, DC Motors;
- **Computer Science:** User input; Parallel execution.

Objectives:

- To create a program with a Parallel execution that includes Sounds, Visuals and some of the Movement: Drive and Turn actions and Connections between them to solve this particular challenge.

Learning Outcomes:

- I can combine modules to build the Robo-vehicle which can drive;
- I can analyze a challenge and create a specific program to solve it;
- I can create a program with a Parallel execution that includes Sounds, Visuals and some of the Movement: Drive and Turn actions and Connections between them;
- I can change settings for Constant Light, Blink, Drive and Turn actions considering their place in a State.

Key Vocabulary:

- Actions Dock; Sounds, Visuals, Movement;
- RW Visual Based Programming: State, Action's lifespan.

Additional Materials:

- Worksheet #6
- *Optional: Lego™ bricks, colored paper or other materials.*

Project 6: Robo Travels to Toytown

Activity Stages:

Lead-in *7 – 10 min*

1. **Ask:** Which modules did we already use? What types of code did we program? How are they different, and how did this influence Robo's performance?
2. **Analyze: tell** Robo's story, **identify** the problem situation and **come up** with a theoretical solution.

Guided Activity *15 – 20 min*

3. **Create an environment:** Use different materials to build the road to Toytown with various obstacles – trees, rocks, turns. Connect the Main Block to the tablet in Robo Code App.
4. **Build** a Robo-vehicle and program: help Robo to get to Toytown! **Use the worksheet #6.**
   - Program a **Sequential code:** Movement – Drive and Turn actions to avoid certain obstacles and get to Toytown.

   Note: It's best to start with simple concrete tasks and increase in complexity as the students become more proficient. In this case, students will create a simple code and improve it step by step. For some students, it could be easier to start with the code that imitates real Robo movements or even try all the movements by themselves before coding;
   - **Modify** the **Sequential code:** add Visuals and Sounds. For example: Robo lights up the road each time before the Turn action. Robo makes a sound before each Drive action;
   - Program a **Parallel execution:** add Visuals and Sounds to the Movement to form the States. For example: Robo lights up the road each time it makes Turn action – simultaneously; Robo makes a sound each time it performs Drive action – simultaneously.
5. **Sum up:** Analyze the difference between the Sequential code and Parallel execution; how all these Actions can help Robo to travel to Toytown.

(*) ***Additional activity:*** *Explore* ***Motor 1 and Motor 2 actions.*** *Program a* ***Sequential code*** *with Motor 1 and Motor 2 actions and discuss how they are different from the Drive and Turn actions. Try different settings and see how they affect Robo's movements. Modify the code to the p****arallel execution*** *with Motor 1 and Motor 2 actions performed simultaneously. Discuss: why it is possible to add two Movement actions in one State. Drive all the way to Toytown using only Motor 1 and Motor 2 actions.*

Independent Activity *15 – 20 min*

6. **Group activity:** create a road to Toytown with various challenges and program your Robo to solve them and get to Toytown. Present the project to the class.

Reflexion & Feedback *5 – 7 min*

7. **Receive feedback** about the complexity of tasks and activities; about students' projects.
8. **Clean up:** Teach students to take care of the devices they use: RW Modules and the tablets.

Project 7: Robo looks around

Concepts: Sequential Logic, RW Visual Based Programming: Action and its Settings, Loop. Mechanical Design

Complexity: ★★☆

Robo's Story:

Last time we helped Robo to travel to Toytown. Now it's time to meet other creatures – the toys! Robo wants to learn how to look around in order to communicate with the fun toys.

Lesson Goal:

To help Robo to look around, we will build a Robo with a head and program turn it around using different motors.

Project 7: Robo looks around

Modules:

Main Block

DC Motors

Servo

Distance Sensor

Connectors

Program:

Movement → Motor → Servo

Focus:

- **Robotics:** Outputs: Servo and DC Motors;
- **Computer Science:** User input – parameters of Actions.

Objectives:

- To create a simple sequential logic program that includes some of the Movement: Servo OR Motor 1 actions and Connections between them; to understand the difference between Servo and DC Motors.

Learning Outcomes:

- I understand the difference between Servo and DC Motor and their functions;
- I can combine modules to build the Robo using either Servo or DC Motor regarding its function;
- I can create a simple sequential logic program that includes some of the Movement: either Servo or Motor 1 action;
- I can change settings for Servo and Motor 1 actions.

Key Vocabulary:

- Movement: Servo and Motor 1 actions; distance, angle, zero position;
- Mechanical and Code Design.

Additional Materials:

- Worksheet #7;
- Printable images: Main Block, DC and Servo Motors;
- *Optional: Lego™ bricks, colored paper or other materials.*

Project 7: Robo looks around

Activity Stages:

Lead-in *7 – 10 min*

1. **Ask:** What were the last projects? Which Modules did we use? What was the function of the DC Motors? In which devices are motors used in real life? Are they all the same?
2. **Analyze: tell** Robo's story, **identify** the problem situation and **come up** with a theoretical solution.

Guided Activity *15 – 20 min*

3. **Get ready:** connect the Main Block to the tablet and create a new project.
4. **Build and program:** teach your Robo to look around! **Use the worksheet #7.**
   - Find **DC and Servo Motors** and compare their color and mechanical design: are they the same or different?
   - Attach and program one **DC Motor** – Motor 1 action, and discuss how Robo's movement: it turns its head around several times;
   - Attach and program **Servo Motor** – Servo action, and discuss the Robo's movement is different from the previous robot: it turns its head to the exact angle;
   - Play the **same Servo action** (with the same settings) once again and ask: Does Robo turns its head now, why not?
   - Change a **degree** in settings of the **Servo action** and discuss the terminology: **degree, angle, zero position;** What does it mean to set a zero position?
   - Try the **random function** of Servo action and discuss: What does it mean "random"? Who generates the setting if it is random? How can we control the range of possible random settings?
   - **Play around:** program different **Servo actions** and Connections between them, so Robo turns or shakes its head. Change the Servo motor to DC Motor and compare the difference between Robo's movement.
5. **Sum up:** the difference between DC and Servo Motors, their settings, and Robo's performance.

Independent Activity *15 – 20 min*

6. **Make your own project:** think about how Robo with the Servo Motor can help you in your life; build any variation of Robo and program Servo actions for this particular situation. Use other Modules and some materials to customize your Robo.

(*) ***Additional Activity:*** *Explore* ***technical and code design*** *using Motors. Add both DC and Servo Motors to the build and program them accordingly their functions. Discuss: Which type of Motors is more suitable for which tasks? How are the Motor and Servo actions same or different? Is it possible to form a State with both Movement actions and why?*

Reflexion & Feedback *5 – 7 min*

7. **Sum up:** Technical details of DC and Servo Motors, Servo action, and its settings such as degree, random function, zero position.
   Receive feedback about the complexity of tasks and activities.
8. **Clean up:** Teach students to take care of the devices they use: RW Modules and the tablets.

Project 8: Robo Meets Friends

Concepts: Sequential Logic and Parallel Execution; RW Visual Based Programming; Code Design

Complexity: ★★★

Robo's Story:

Today is a significant day for our Robo: we are going to help Robo meet and interact with other toy creatures! Which Modules and Actions will Robo need to do it?

Lesson Goal:

To help Robo to communicate with the other toys, we will build Robo using all of the learned Modules, and program different Actions.

Project 8: Robo Meets Friends

Modules:

Main Block

RGB LED

Distance Sensor

Servo

x2
DC Motors

x2
Wheels

Small Wheel

x4
Connectors

Connector Block

Program:

Sounds

Visuals → Constant Light, Blink

Movement → Drive, Turn, Servo

Focus:

- **Robotics:** Outputs – Sounds, RGB Light, DC Motors, Servo;
- **Computer Science:** User input – parameters of Actions.

Objectives:

- To create a simple program with both sequential logic and a Parallel execution that includes Sounds, Visuals and Movement and Connections between them.

Learning Outcomes:

- I can combine modules to build the Robo which can perform the set tasks;
- I can create a program with both sequential logic and a Parallel execution that includes Sounds, Visuals and Movement and Connections between them;
- I can change settings for different actions: Constant Light, Blink, Drive, Turn, and Servo actions considering their place in a State.

Key Vocabulary:

- Action, State, Connection, Transition, Action's lifespan;
- Code Design.

Additional Materials:

- Worksheet #8;
- *Optional: Lego™ bricks, colored paper or other materials.*

Project 8: Robo Meets Friends

Activity Stages:

Lead-in *7 – 10 min*

1. **Ask:** Which modules did we already use? What did we program? What is our Robo capable of? How can it help our Robo?
2. **Analyze: tell** Robo's story, **identify** the problem situation and **come up** with a theoretical solution.

Guided Activity *15 – 20 min*

3. **Create an environment:** Use different materials to build Toytown and its inhabitants. **Connect** the Main Block to the tablet in Robo Code App.
4. **Build and program:** teach your Robo to look around! **Use the worksheet #8.**
   - Program a **Sequential code:** Sounds – Robo speaks to toys; Visuals – Robo emits different lights to greet new friends; and Movement – Robo goes from one toy to another to meet them all; Robo turns its head to look at the new friends.
   - **Modify** the **Sequential code:** change the settings for some Actions; try out random functions for Visuals – Constant Light and Blink actions, Servo action.
   - Program a **Parallel execution:** combine Actions to form the States – Robo performs some Actions simultaneously.
   - **Play around:** Program robots to perform different tasks and to interact with toys or with each other.
5. **Sum up:** Analyze the difference between the Sequential code and Parallel execution; how all these Actions can help Robo to communicate with different toys in Toytown.

(*) ***Additional Activity:*** *Explore the* ***random function.*** *Recall the term and remember how to program a random function. Find all Actions which have a random function in the settings – Constant Light, Blink, Turn and Servo actions and try them in different combinations: as Sequential code and Parallel execution. Discuss how the random function influences Robo's performance.*

Independent Activity *15 – 20 min*

6. **Group activity:** create Toytown with different inhabitants and their stories – some challenges for Robo; build and program your Robo to solve them. Present the project to the class.

Reflexion & Feedback *5 – 7 min*

7. **Receive and give feedback** about the complexity of tasks and activities; about students' projects.
8. **Clean up:** Teach students to take care of the devices they use – RW Modules and the tablets.

Project 9: Robo Makes a Surprise

Concepts: Engineering Design Process; Mechanical and Code Design

Complexity: ★★★

Robo's Story:

Robo is happy that it met some fun toys and wants to create a surprise for these new friends.

Lesson Goal:

To help Robo to design a surprise for his new friends, we will learn about the Engineering Design Process and then use it to build and program the Robo-project.

Project 9: Robo Makes a Surprise

Modules:

Main Block

DC Motors

Servo

RGB LED

Distance Sensor

Wheels

Small Wheel

Connector

Connector Flat Lego™

Connector Block

Program:

Sounds

Visuals → Constant Light, Blink

Movement → Drive, Turn, Servo

Focus:

- **Robotics:** Outputs – Sounds, RGB Light, DC Motors, Servo;
- **Computer Science:** User input regarding the set task.

Objectives:

- To create a simple program with both sequential logic program and Parallel execution that includes Sounds, Visuals, Movement and Connections between them accordingly to the set task.

Learning Outcomes:

- I know what an Engineering Design Process is;
- I know and can follow the steps of the Engineering Design Process to deal with the set task;
- I can combine modules to build the Robo which can perform the set task;
- I can create a simple program with both sequential logic program and Parallel execution that includes Sounds, Visuals, Movement and Connections between them accordingly to the set task;
- I can change settings for different actions considering their place in a State and the set task.

Key Vocabulary:

- Engineering Design Process and its steps; Mechanical and code design;

Additional Materials:

- Printable cards with Engineering Design Process steps;
- Worksheet #9: number of challenges for students;
- *Optional: Lego™ bricks, colored paper or other materials.*

Project 9: Robo Makes a Surprise

Activity Stages:

Lead-in *7 – 10 min*

1. **Ask:** How many projects did we do with Robo? Each time we helped Robo with a different problem, what was the reason for the project? How did we do it? Have you ever created your own project in your school or daily life? How did you do it? Is this process similar to what we did for Robo? Do you think that there is a cohesive plan for creating a project?
2. **Analyze: tell** Robo's story, **identify** the problem situation and **come up** with a theoretical solution.

Guided Activity *15 – 20 min*

3. Learn about the **Engineering Design Process.** Ask students to describe how they would make a project for the set task; **define each step** together with students, hang the card for this step on the board. Discuss the terminology: **Technical and code design, Engineering Design Process.** **Use the worksheet #9 and printable cards with the steps.**
   - **Step 1. Identify a reason: What is the problem or idea?** Define the purpose of particular Robo-project. Reframe it, say it aloud or write down, be able to explain the reason.
   - **Step 2. Brainstorm: What are the solutions?** Brainstorm as many solutions as possible, but do not evaluate them. At this step, the solutions don't necessarily need to be good.
   - **Step 3. Evaluate and pick one: What would happen if...?** Think about the pros and cons of each solution, discus and range them. Pick the best solution.
   - **Step 4. Sketch and plan: What will I need?** Make a sketch and decide which Robo's Modules or other materials you will need for the project.
   - **Step 5. Work on a solution: Build and program, test, repeat!** Work on your idea and try it out. If the first solution doesn't work, discuss why and move on to another one. It is important to **keep trying** until the problem is solved. Do not lose the motivation, not all of the problems are easy to solve.
   - **Step 6. Finalize: Is everything ready?** As soon as you find the best solution for your project, finalize it: create an environment for your Robo or customize it, and check if everything is ready for a presentation.
   - **Step 7. Present the solution.** Show your project to the class, ask for feedback.
   - **Step 8. Reflect: How was it?** As soon as the problem is solved, reflect on the process and ask yourself: What worked? What didn't? What can you do differently next time?
4. **Practice the Engineering Design Process:** Discuss Robo's story and go through all the steps for this situation together with your students planning the individual activity.

Independent Activity *15 – 20 min*

5. **Individual or group activity:** Create the Robo-project accordingly set task using the Engineering Design Process and present your solution to the class.
6. **Presentation:** Students show their project to the other students, give and receive feedback.

Project 10: What is Your Robo?

Concepts: Engineering Design Process, Mechanical and Code Design

Complexity: ★★★

Robo's Story:

Now it is your turn to decide which device or character you would like your Robo to transform into! Each student needs to come up with a reason as to why we need this particular Robo-project or character, and be able to explain how it will help us!

Lesson Goal:

To transform your Robo into your own project, follow the steps of the Engineering Design Process.

Project 10: What is Your Robo?

Modules:

Main Block

DC Motors

Servo

RGB LED

Distance Sensor

Wheels

Small Wheel

Connector

Connector Flat Lego™

Connector Block

Program:

Sounds

Visuals

Constant Light

Blink

Movement

Drive

Turn

Servo

Focus:

- **Robotics:** Outputs – Sounds, RGB Light, DC Motors, Servo;
- **Computer Science:** User input regarding the set task.

Objectives:

- To create their own project with RW robotics set – build and program a robot accordingly its function using all the learned Modules and RW Visual Based Programming language;

Learning Outcomes:

- I can come up with a story and reason to build a particular robot;
- I know and can follow the steps of the Engineering Design Process to deal with the set task;
- I can build and program my own Robo-project using all the learned Modules and RW Visual Based Programming language;
- I can create an environment for it using different materials;
- I can present my project to the class and give constructive feedback to other students.

Key Vocabulary:

- Project, Engineering Design Process; Mechanical and code Design.

Additional Materials:

- Printable cards with Engineering Design Process steps;
- Worksheet #10: some ideas of projects for students;
- *Optional: Lego™ bricks, colored paper or other materials.*

Project 10: What is Your Robo?

Activity Stages:

Lead-in *7 – 10 min*

1. **Recall** the **project** you have done last time, including the Engineering Design Process and how it helped you with a project.
2. **Analyze: tell** Robo's Story and set the lesson's goal.

Guided Activity *15 – 20 min*

3. Plan the activity stages for an independent activity together with the students. Recall each step of the Engineering Design Process and hang the card for this step on the board.
   - **Step 1. Identify a reason: What is the problem or idea?**
   - **Step 2. Brainstorm: What are the solutions?**
   - **Step 3. Evaluate and pick one: What would happen if...?**
   - **Step 4. Sketch and plan: What will I need?**
   - **Step 5. Work on a solution: Build and program, test, repeat!**
   - **Step 6. Finalize: Is everything ready?**
   - **Step 7. Present the solution.**
   - **Step 8. Reflect: How was it?**

Independent Activity *15 – 20 min*

4. **Make your own project:** Students follow the steps on the board in order to create their own Robo-project. They can work independently, in pairs or in small groups and use different materials to create an environment and customize robots. Students can also cooperate to collaborate on a shared project with two or more robots.
5. **Presentation:** Students present their projects to the class and give constructive feedback to each other.

Reflexion & Feedback *5 – 7 min*

7. Receive and give feedback about the complexity of tasks and activities.
8. Clean up: Teach students to take care of the devices they use – RW Modules and the tablets.

Key vocabulary for teachers

I. Robotics

Robot – 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 pre-programmed to behave autonomously. Robots are created by people to help with many different tasks which may sometimes be too complex, too dangerous, or simply too repetitive to do ourselves.

Robotics – 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.

Engineering – The process of creating and building technological solutions and products by using math and science. **Engineers** solve problems with their inventions. There are several branches of engineering.

Electricity – A type of energy that can build up in one place or flow from one place to another; it is used to make all electrical devices function. **Robo Wunderkind** modules are powered by a battery inside the Main Block. Power is passed from module to module through the **Connectors** (**Pogo-Pins** on them) and **Faces** on each module.

Wired (Local) Communication – A type of electrical communication between electronic devices which transfers information over a wire-based communication technology – connectors. In **Robo Wunderkind robotics kit** there are two types of connectors: Universal Connectors and Wired Connector which transfer the signals – commands from the **Main Block** to other modules.

Wireless Communication – The type of communication which transfers information over a distance without the use of “hard wired” connection (like Pogo-pins on the Robo Wunderkind modules). The distances involved may be short (when using a television remote control) or very long (thousands or even millions of kilometers for radio communications). The **Robo Code and Robo Live Apps** send the information – commands for Robo to perform through wireless communication via Bluetooth.

Bluetooth signals – Is a wireless technology standard for exchanging data over short distances.

Key vocabulary for teachers

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.

Electronic device –

A device that achieves its purpose through electrical means. There is a wide range of electronic devices people use every day such as laptops, mobile phones, cameras, fans, ovens, washing machines, game consoles, printers, radios, and Robo, of course!

Smart device –

An electronic device, generally connected to other devices or networks (via Wireless Communication) and can operate interactively and autonomously to some extent.

Sensor –

A device that receives a physical signal or stimulus – Physical Input (such as sound, pressure or light) and responds to it in a distinctive manner. There are Sound and Distance sensors as well as Button in Robo Wunderkind Education Kit and Motion and Light sensors in Advanced Kit.

Button (Push-button) –

Is a simple switch mechanism for controlling some aspect of a machine or a process. In Robo Wunderkind robotics kit there is a red **Button** which can be programmed as **Button condition** in the **Robo Code App**.

Remote control –

A method for controlling a machine from a distance by using wireless signals, it happens in real time. For example, the remote control for television when you press buttons to change channels.

Digital literacy –

An ability to find, evaluate, compose and create clear information through using various digital platforms. It includes both the **practical software skills** and **critical thinking** which helps to stay safe online. Digital literacy is evaluated by an individual's grammar, composition, typing skills and ability to produce writings, images, audio and designs using technology. In **Robo Wunderkind Curriculum** the **numerous skills** related to the digital literacy are covered such as rules for **safe use of devices, consuming information through digital media** and, most importantly, **ability to produce technology** through coding.

Key vocabulary for teachers

Problem-solving – Is the act of finding a solution for a problem. The steps for the problem solving are slightly different depending on the discipline and strategy but it always involves defining a problem, identifying, prioritizing and selecting alternatives for a solution, implementing a solution, and reflection on this solution.

In **Robo Wunderking Curriculum** students go through **4 steps** in order to solve a problem using Robo Wunderkind robotics kit and Robo Code or Robo Live App:
1) Identify a problem => 2) Plan the solution(s) => 3) Work on the solution: build and program => 4) Reflect

---

Engineering Design Process – A series of steps that engineers follow to come up with a solution to a problem. It involves cognitive, strategic and practical processes by which design concepts are developed.

In **Robo Wunderkind Curriculum** students learn about and go through 8 stages of **Engineering Design Process** in order to create their own Robo-project.

Design – Is the intentional creation of a plan in order to fashion, execute, or construct a certain object.
In **Robo Wunderkind Curriculum** we approach 3 types of design:

1) Mechanical Design – Robo Modules
- How to design a robot efficiently, so it performs the set task, doesn't crash or have accidents?
- How to make an efficient build – attach only those Modules which you need for your goals?
- How do different builds and configurations provide the same or different results?

2) Control Design – Robo Live App
- You have all the controls you need – not less;
- You have only those controls on the screen which you need (not more);
- You organize the Controls on the screen so it's convenient to use them in the Play modeю

3) Code Design – Robo Code App
- Your code consists of all the Actions, States, Connections you need to achieve the project's goal(s) – not less;
- Your code consists of all the Actions, States, Connections you need to achieve the project's goal(s) – not more;
- Your code is clear and easy to read by others.

Key vocabulary for teachers

II. Computer Science

Algorithm – Step-by-step solution of one task; each step is a clear instruction. A simple example of an algorithm is a cooking recipe, where you have one by one instruction in order to cook one final dish.

Program code – A set of instructions which tells a computer what to do; a sequence of short commands, one after another.

Programming language

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 **Robo Code App**, we use a special visual coding language.
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**.

Robo Wunderkind Visual Based Programming – A unique and intuitive programming interface designed for young children to build State-Machine Based Programs

---

State-Machine Based Programming –

A State performs Action(s) and Transition to another State based on events.

- **State** – set of Actions which can consist of one or more Actions;
- **Connection** – tells the possible Transition to other States (it may happen / may not);
- **Transition** – the act of changing from one State to another (= the act of happening);
- **Condition** – an icon which compares two numbers and determines the results be true or false, and tells if true – Transition happens or false – nothing happens;
- **Action** – an icon which performs an output task; and has different parameters: Lifespan, Speed, Distance etc.

Key vocabulary for teachers

Sequential logic – A sequence of States where one State leads to only one other State until a program is complete.

Loop – A sequence of States that lead one State to the next that does not terminate the program but the transition back to the starting State (repeats N-times).

Parallel Execution – The execution of several Actions at the same time within a State.

User Input – Is is data provided by the user to the device. It can be digital input such as text to display on a screen or it could be physical like a button click by the user or a key board press on a keyboard.
In the Robo Live App – using the appropriate Control to control the appropriate Module. **In the Robo Code App:**
1) Software Input
- Parameters of Actions such as Action lifespan, brightness, speed, angle or distance;
- Connections drawn between Actions;
- Conditions and Conditions parameters.

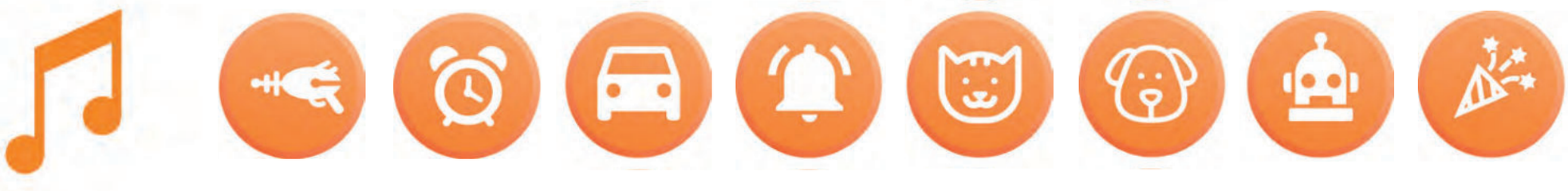
2) Sensor Input – data received from Sensors such as sound level, distance before an obstacle or pressure on the Button.

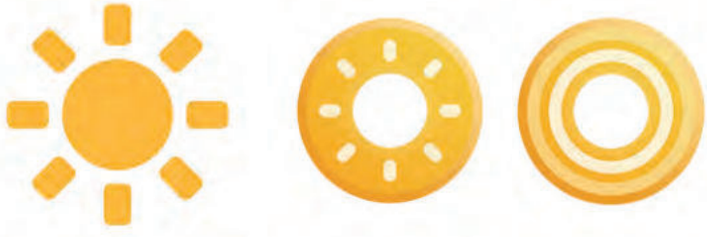
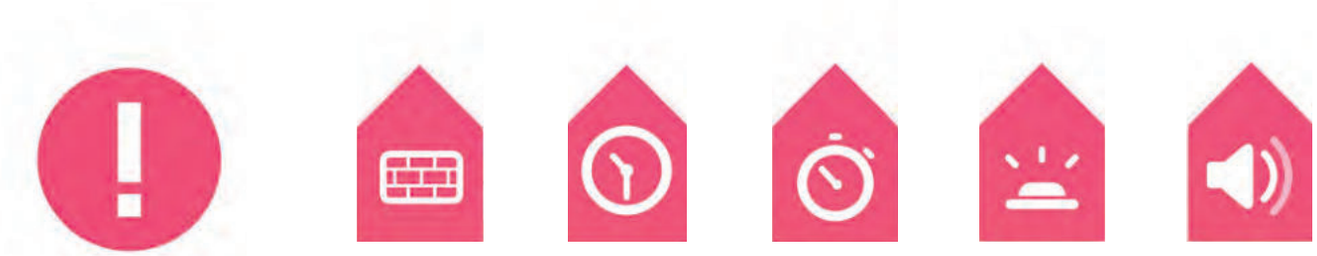
User Output – For Robo Wunderkind Modules the Outputs would be the answers for the questions:
What does the Module do?
- How are Modules different from each other?
- How do they work together?
- How to combine the Modules to achieve the project's goal(s)?

Modules, coding buttons, and icons to print

Modules, coding buttons, and icons to print

Modules, coding buttons, and icons to print





Modules, coding buttons, and icons to print