

MISSION 1 (Advanced)

The Reaction Tracker!

Mission Aim:

Explore how the SPIKE Prime Reaction Tracker works and test the speed at which you can react to lights and sounds

Success Criteria:

- Identify and use some of the SPIKE Prime hardware
- Explore the Python code that brings the SPIKE Prime hardware to life
- Explore how your brain, focus, and reactions work together as you respond quickly to lights and sounds

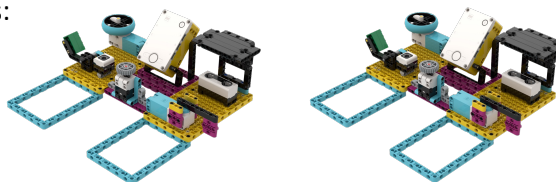
Key words:

- **Colour Sensor:** *Can detect 8 colours and reflected light. Can also be used as a light output.*
- **Distance Sensor:** *Uses ultrasonic sound waves to measure how far away an object is (5cm – 200cm)*
- **Force Sensor:** *Can measure how hard something is pressed or touched (up to 10 newtons)*
- **Hardware:** *Programmable elements such as the Hub, sensors and motors*

Resources for Mission 1:

Pre-build Models: (Build Guides in the Build Guide Folder):

- Build **2 SPIKE Prime Reaction Trackers** using the [ReactionTrackerBuildGuide.pdf](#) OR have pupils build together (4 pupils per build) using the Optional [FourPersonReactionTrackerBuildGuide.pdf](#)
- The Reaction Trackers look like this:



Mission 1 Summary:

- In Mission 1, pupils will be introduced to the theme of human reaction time.
- They will explore the hardware and code used in the SPIKE Prime Reaction Tracker, testing how fast their reactions are as they race to react to visual and audio prompts.

Mission 1: FLEXIBLE LEARNING PLAN

ENGAGE:

10 mins approx.

- Introduce pupils to the context of Coding Success 5 by showing **SLIDES 1-2** of the Mission 1 Presentation ([CS5_AM1_Presentation.pptx](#)).
- Show **SLIDE 3** to introduce upcoming Missions 1-4.
- Show **SLIDES 4-5** to outline the context and aims of Mission 1.
- Briefly elicit any prior learning of Python coding - knowledge and skills that will be used and developed in Coding Success 5.

SUPPORT AND/OR CHALLENGE:

Teacher can edit the Mission Aim and Presentation as required

Presentation has additional notes to support delivery.

EXPLAIN:

10 mins approx.

- Show **SLIDE 6** for pupils to see the construction of the SPIKE Prime Reaction Tracker and then reveal the two pre-built Reaction Tracker models.
- Show **SLIDE 7** to explain the hardware used and identify the key elements in the build.

SUPPORT AND/OR CHALLENGE:

Teacher might add a glossary of new terminology to the board to support literacy skills.

EXPLORE

35 mins approx.

- Play **SLIDE 8** to explain the key Python code used in the Reaction Tracker (3 min 24 sec video).
- Show **SLIDE 9** and explain the 3 rules of the Reaction Tracker game.
- **Important:** SPIKE Prime Python can't play customised sounds - so sounds from within the SPIKE App have had to be used. If you want to play a version with customised sounds, use the Word Block version (available in the Beginner and Intermediate resources).
- Show **SLIDE 10**. Run the Python code on the 2 SPIKE Prime hubs, invite the first pupils to play.
- Celebrate pupils' efforts – perhaps record pupils' reaction times on a leaderboard (**SLIDE 11**).
- After playing, show **SLIDE 12** and explain what happened in terms of the speed of human reactions.
- Show **SLIDES 13 - 15** and play **SLIDE 14** to explore the speed of robotic reaction compared to human reaction.

SUPPORT AND/OR CHALLENGE:

Teacher might select key parts of the video (**SLIDE 8**) to highlight key Python coding learning points that are most relevant to pupils. Some pupils might explore the code in greater detail.

Some pupils might benefit from a quick practise before playing. Some pupils might enjoy the challenge of working as a pair as they play the Reaction Tracker game.

EVALUATE AND CELEBRATE:

5 mins approx.

- Show **SLIDE 16** to review the Mission Aims.
- **OPTIONAL:** Show **SLIDE 17** for an opportunity to discuss how Mission 1 links to 6 STEM careers.
- Show **SLIDE 18** to signpost Mission 2.

SUPPORT AND/OR CHALLENGE:

Can pupils think of additional STEM jobs that require fast reactions?

As a homework activity, pupils might research 'Why is having quick human reactions important in one STEM job?' which builds on the discussion (**SLIDE 17**).

Discover more:

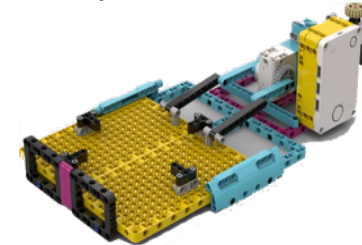
For more stand-alone Coding Success lessons that demonstrate the key components of SPIKE Prime, why not try...

CODING SUCCESS 1

EARTHQUAKE... RESCUE

In **Lesson 1**, pupils learn about the SPIKE Prime hardware and they are introduced to coding the key components. They build a structure and test it on a SPIKE Prime Earthquake simulator.

CS1 Lesson 1: Earthquake Simulator

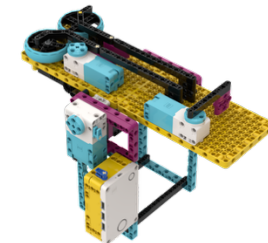


CODING SUCCESS 2



In **Mission 1**, pupils use SPIKE Prime to launch a rocket! They will explore the code that brings the rocket launcher to life and create their rocket from paper. They will set the angle and speed of launch on the SPIKE Prime rocket launcher, then test and amend their rocket design and code to see which rocket travels the furthest (or most accurately!) Available at 3 different levels: Beginner and Intermediate (using Word Blocks) and Advanced (using Python).

CS2 Mission 1: Rocket Launch!



CODING SUCCESS 3



In **Mission 1**, pupils explore the hardware used in the SPIKE Prime Top Thumb! robotic machine. Working in teams of 4, pupils will play 'Top Thumb' and compete to generate the most 'power'. Pupils will then explore how power is calculated and how power can be generated by humans. Available at 3 different levels: Beginner and Intermediate (using Word Blocks) and Advanced using Python.

CS3 Mission 1: Top Thumb!






CODING SUCCESS 4




In **Mission 1**, pupils will explore hardware and code used in the SPIKE Prime Robodog build. Working in teams of 2, pupils will play Robodog Racing, using Coding Sticks to have Robodog travel as quickly as possible on the racetrack. Pupils will explore how robotic dogs are used in the real world and will be introduced to the concept of Robot Academy! Available at 3 different levels: Beginner and Intermediate (using Word Blocks) and Advanced using Python.

CS4 Mission 1: Robodog Race!



Computing/Computer Science Links		
	Key Stage 2 <ul style="list-style-type: none"> Design programs that accomplish specific goals. Debug programs that accomplish specific goals. Use repetition in programs. Control or simulate physical systems. Use logical reasoning to detect and correct errors in programs. Work with various forms of input. Work with various forms of output. 	Key Stage 3 <ul style="list-style-type: none"> Design, use and evaluate computational abstractions that model the state and behaviour of real-world problems and physical systems. Understand several key algorithms that reflect computational thinking Use logical reasoning to compare the utility of alternative algorithms for the same problem Understand simple Boolean logic and its uses in programming Use two or more programming languages
	Second Experience & Outcome (to end of P7) <ul style="list-style-type: none"> TCH 2-14a: I can explain core programming language concepts in appropriate technical language. TCH 2-15a: I can create, develop and evaluate computing solutions in response to a design challenge. 	Third & Fourth Experiences & Outcomes (S1 – S2) <ul style="list-style-type: none"> TCH 3-13b: I am developing my understanding of information and can use an information model to describe particular aspects of a real-world system. TCH 3-15a: I can select appropriate development tools to design, build, evaluate and refine computing solutions based on requirements. TCH 4-13a: I can describe in detail the processes used in real world solutions, compare these processes against alternative solutions and justify which is the most appropriate. TCH 4-13b: I can informally compare algorithms for correctness and efficiency. TCH 4-15a: I can select appropriate development tools to design, build, evaluate and refine computing solutions to process and present information whilst making reasoned arguments to justify my decisions.
	Progression Step 3 <ul style="list-style-type: none"> I can use conditional statements to add control and decision-making to algorithms. I can identify repeating patterns and use loops to make my algorithms more concise. I can use sensors and actuators in systems that gather and process data about the systems' environment. I can explain and debug algorithms. 	Progression Step 4 <ul style="list-style-type: none"> I can plan and implement test strategies to identify errors in programs. I can apply design principles in order to design a range of efficient user interactions.

Skills Builder – Universal Framework 	<p>During each lesson, you might wish to highlight one or more of the essential skills that students build over their lifetime. You might give students the opportunity to reflect on how successful they have been in developing these skills. You can download resource M1_SkillsBuilder for further details of how the 'Universal Skills Builder Framework' links to the Coding Success 5 project. Further details of the Skills Builder Framework and assessment opportunities can be found at www.skillsbuilder.org</p>
---	---

Gatsby Framework	<p>The Gatsby Career Benchmarks is a framework of eight guidelines about what makes the best careers provision in schools and colleges. The resource M1_Gatsby (in Mission 1 resources) provides further details of how the 'Gatsby Framework' links to the Coding Success 5 project.</p>
-------------------------	---

