

MISSION 3 (Advanced)

Smart Sensors!



Mission Aim:

Learn how to Python code the SPIKE Prime colour and distance sensors

Success Criteria:

- Understand how sensors help my robot 'see' and react to the world
- Use Python coding to combine sensors with movement
- Develop our understanding of how to achieve Smart Speed

Key words:

Colour Sensor: Can detect 8 colours and reflected light. Can also be used as a light output.

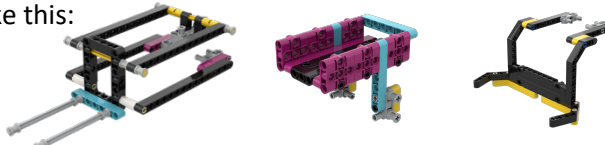
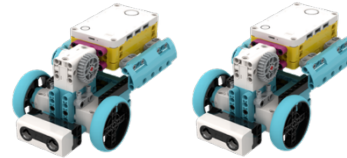
Debugging: Testing the outcome of programs and correcting any errors.

Distance Sensor: Uses ultrasonic sound waves to measure how far away an object is (5cm – 200cm).

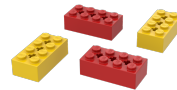
Resources for Mission 3:

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

- Build (or have pupils build) **TWO SPIKE Prime Multi-Movers** using the Build Guide. The 2 Multi-Movers look like this:
- Build (or have pupils build) **TWO fork**, **TWO tipper bed** and **TWO blade** attachments. They look like this:



- Build (or have pupils build) **TWO** of each of the mat objects using the build guides. They look like this:



Debris



Solar Panel



Radar Upgrade



Used Tyres

Worksheet: [CS5_AM3_Worksheet](#)

Additional resources: Set up the Mission Mat used in Mission 2 (PRINT/STICK/TAPE?)

Mission 3 Summary:

- Pupils will be introduced to the colour sensor and distance sensor used in the SPIKE Prime Multi-Mover.
- Pupils will tackle a number of Mission Mat challenges using different attachments as they learn how to code the colour and distance sensor to control the Multi-Mover's movements.

Mission 3: FLEXIBLE LEARNING PLAN

ENGAGE:

5 mins approx.

- Briefly introduce pupils to Mission 3 and the Mission Aim by showing **SLIDES 1-4** of the Mission 3 Learning Presentation ([CS5_M3A_Presentation.pptx](#)).

SUPPORT AND/OR CHALLENGE:

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

EXPLAIN:

10-15 mins approx.

- Play **SLIDE 5** to show the construction of the Multi-Mover and its attachments – point out where the colour and distance sensors are on the build.
- Show **SLIDE 6** to remind pupils about the pre-built attachments (blade and tipper bed) and introduce The Lifter.
- Show **SLIDE 7** - Play the video and explain how to add the fork attachment. Show the key motor positions.
- Show **SLIDE 8** and play the video to explain how to Python code the distance sensor (or demo this live).
- Show **SLIDE 9** and play the video to explain how to Python code the colour sensor (or demo this live).
- Show **SLIDE 10** to introduce the Mission Mat objects and the challenges.

SUPPORT AND/OR CHALLENGE:

Pupils might benefit from the teacher pausing the coding videos on **SLIDE 8** and **SLIDE 9** to reinforce the key learning points, or pupils can rewatch the videos independently to support them when coding.

EXPLORE

35 mins approx.

- Hand out the Worksheet [M3A_SmartSensors](#) and briefly outline Challenges A-C (**SLIDES 11-14**).
- Pupils use the SPIKE App to create the Python code needed to tackle Challenges A-C.
- Pupils should take it in turns, running their Python code on one of the 2 Multi-Movers and debug as required!

SUPPORT AND/OR CHALLENGE:

Some pupils might focus on Challenge A-B on the 'Smart Sensors' Worksheet. Teachers might also offer additional support using the sample programs – found in Program folder.

Some pupils might begin an optional Scratch Challenge activity using Word Blocks - the link is shown on **SLIDE 15**.

Some pupils might complete the Showcase challenge after successful completion of Challenges A-C.

EVALUATE AND CELEBRATE:

5-10 mins *approx.*

- Show **SLIDES 16-18** and discuss the power of technical advantage for the Multi-Mover and SMART cars/robots.
- Show **SLIDE 19** and review the Mission Aim .
- Show **SLIDE 20** and congratulate pupils on their coding success!

SUPPORT AND/OR CHALLENGE:

As an extra-curricular activity, pupils might research the science behind the colour sensor or distance sensor.

Pupils might enjoy extending their learning outside lesson time, by taking part in **Coding Success: CLUB (SLIDE 21)**

Discover more:

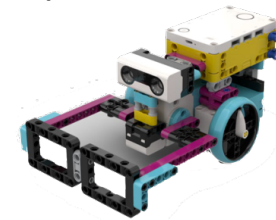
For more Coding Success lessons that demonstrate how to code the SPIKE Prime colour, distance or gyro sensors, why not try...

CODING SUCCESS 1



In **Lesson 3**, pupils learn how the colour sensor works and how to code the colour sensor. In **Lesson 4**, pupils learn how to code the distance sensor on the SPIKE Prime Rescue Vehicle supporting islanders affected by an earthquake. Available at 2 different levels: Beginner (Word Blocks) and Advanced (Python).

CS1 Lesson 3: Search and Rescue CS1 Lesson 4: Pipeline Problems

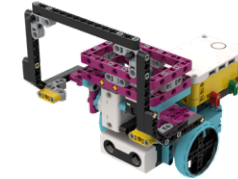


CODING SUCCESS 2



In **Mission 3**, pupils program their SPIKE Prime Support Vehicle to locate, grab or move 4 hydroponic pods. Each hydroponic pod is represented by 2 coloured LEGO DUPLO bricks. Pupils will learn to program the tipper/grabber, using the distance sensor and colour sensor. Available at 3 different levels: Beginner and Intermediate (using Word Blocks) and Advanced (using Python).

CS2 Mission 3: Satellites and Sensors



CODING SUCCESS 3



In **Mission 3**, pupils learn how to code the colour sensor to help develop greater accuracy and precision as their SPIKE Prime EV moves forwards, backwards, turns and moves 'green' infrastructure to different locations on the themed Mission Mat. At Intermediate level, they also learn to code the gyro sensor. Available at 3 different levels: Beginner and Intermediate (using Word Blocks) and Advanced using Python.

CS3 Mission 3: On The Move!

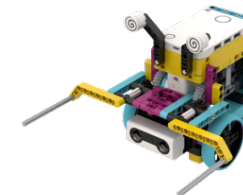





CODING SUCCESS 4










In **Mission 3**, pupils tackle a number of Coding Mat challenges as they learn how to code the colour and distance sensor to control Grabot's movements. At Intermediate level, pupils can also develop their understanding of the gyro sensor when they code it to move Grabot in a straight line. Available at 3 different levels: Beginner and Intermediate (using Word Blocks) and Advanced using Python.

CS4 Mission 2: Skills Boost!



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       	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
--	--

Gatsby Framework	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.
-------------------------	--