KS4 - Physical computing programming project

**Enjoyed teaching these lessons? Found a mistake? Share feedback at** [**the-cc.io/feedback**](http://the-cc.io/feedback)**.**

## Unit introduction

This unit introduces learners to physical computing through six lessons and culminates in a finished working robotic buggy. Learners use a Raspberry Pi Pico microcontroller to explore inputs and outputs, and utilise a range of hardware components, including motors, reflective optical sensors, LEDs, and an ultrasonic sensor. This unit of work allows learners to practically experience the use of embedded systems in support of the GCSE computer science specifications.

## Overview of lessons

| **Lesson** | **Brief overview** | **Learning objectives** |
| --- | --- | --- |
| Lesson 1: Introduction to physical computing | In this unit, learners use a Raspberry Pi Pico microcontroller to design, create, and program a motorised buggy. This unit is divided into six individual lessons where learners connect and program different components to increase the features of their buggy. In this first lesson, learners are introduced to physical computing with Raspberry Pi Pico. This lesson focuses on how to set up Raspberry Pi Pico and use the GPIO (General-Purpose Input/Output) pins to interact with physical objects, such as LEDs and buttons. | * Define the term physical computing * Explain the term embedded systems * Create and test a working circuit |
| Lesson 2: Working with motors | The focus of this lesson is to understand how to use a separate motor controller to get a motor to work. Learners discover why a separate board is needed and what the additional requirements of this are in relation to creating a working prototype. They then connect two DC motors to a motor controller and Raspberry Pi Pico, before they experiment with code to determine correct usage for specific directions and actions, such as forwards, backwards, spinning, and turning. | * Explore how to add functionality using a motor controller * Interact with real-world objects using code and additional hardware |
| Lesson 3: Chassis design and build | In this lesson, learners consider and experiment with materials to produce a chassis for their robotic buggy. They need to test that their material of choice can withstand not only the weight of the motors, wires, board, etc, but also the additional weight from a power bank, to make it independent of a connection to the computer/mains. Their prototype must also maintain structural integrity when it moves. | * Use basic materials and tools to create a prototype |
| Lesson 4: Going ultrasonic! | Lesson 4 starts with an introduction to ultrasonic sound waves. This is then expanded upon to show how robotic devices can make use of sensors that include this technology, to avoid bumping into things. Learners connect an ultrasonic sensor to their buggy, code its functionality, and test and iterate, if needed, to refine their implementation. | * Understand how ultrasonic sound waves work * Combine inputs and outputs to solve a problem |
| Lesson 5: Follow me! | In this lesson, learners discover reflective optical sensors and determine why they are useful. Learners have access to two sensors and are shown where these should be located on their buggy to have maximum benefit when they implement a line-following program that they have written. How successful learners are in this task could be assessed through a class competition where all buggies attempt to follow the same course. | * Understand how reflective optical sensors work * Process input data to monitor and react to the environment |
| Lesson 6: Time to shine! | By now, learners should be very familiar with the use of GPIO pins on Raspberry Pi Pico. It is time for them to add some lights to the buggy. In this lesson, learners have to turn the lights on at certain times, depending on the inputs of the sensors fitted to the buggy. For instance, the headlights switch on before the buggy moves off, the buggy performs a movement sequence with brake lights on when it is at a standstill, and finally all lights turn off when the buggy has finished its sequence. | * Synchronise the behaviour of physical hardware components for a given situation |

## Progression

This unit progresses learners’ knowledge and understanding of physical computing. It starts with the creation of basic circuits and builds up their usage and confidence each lesson, as it incorporates programming a new piece of hardware with different inputs and outputs. By the end of the unit, learners should have a good understanding of microcontrollers and embedded systems in general, as well as conceptual and practical experience of physically connecting input/output devices to GPIO pins, obtaining data from sensors, and using this data to make decisions.

This unit provides an opportunity for learners to apply knowledge and skills from their studies in a practical and tangible project, which spans multiple lessons.

## Curriculum links

### GCSE computer science specifications links

* [AQA](https://filestore.aqa.org.uk/resources/computing/specifications/AQA-8525-SP-2020.PDF)
  + 3.2 Programming; 5 Programming skills
  + 3.4.5 Systems architecture - Embedded systems
* [Edexcel](https://qualifications.pearson.com/content/dam/pdf/GCSE/Computer%20Science/2020/specification-and-sample-assessments/GCSE_L1_L2_Computer_Science_2020_Specification.pdf)
  + Practical Programming Statement (PPS)
  + 3.1 Hardware - 3.1.3 Embedded systems
* [Eduqas](https://www.eduqas.co.uk/media/tf3bvmxe/eduqas-gcse-computer-science-specification-10-02-2020.pdf)
  + Unit 1 - Hardware - Embedded systems
  + Practical programming statement
* [OCR](https://ocr.org.uk/Images/558027-specification-gcse-computer-science-j277.pdf)
  + 1.1 Systems architecture - 1.1.3 Embedded systems
  + 2d Practical programming; 2.2 Programming fundamentals

### [National curriculum links](https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study/national-curriculum-in-england-computing-programmes-of-study)

* Develop their capability, creativity and knowledge in computer science, digital media and information technology
* Develop and apply their analytic, problem-solving, design, and computational thinking skills

## Assessment

### Summative assessment

* Assessment rubric

## Subject knowledge

Enhance your subject knowledge to teach this unit through the following training opportunities:

### Online training courses

* [Teaching Physical Computing with Raspberry Pi and Python](http://rpf.io/physicalcomputing)
* [Robotics With Raspberry Pi: Build and Program Your First Robot Buggy](http://rpf.io/robotics)
* [Getting Started with Your Raspberry Pi](https://rpf.io/get-started-course)

Resources are updated regularly - the latest version is available at: [the-cc.io/curriculum](http://the-cc.io/curriculum).



This resource is licensed by the [Raspberry Pi Foundation](https://www.raspberrypi.org/) under a Creative Commons At

tribution-NonCommercial-ShareAlike 4.0 International license. To view a copy of this license, visit, see [creativecommons.org/licenses/by-nc-sa/4.0/](https://creativecommons.org/licenses/by-nc-sa/4.0/).