|  |  |  |  |
| --- | --- | --- | --- |
| **Programming the robot buggy** | | | |
|  |  | |  |
| Work as a team to program the robot buggy so that it can navigate a maze path | | | |
| **Subjects:** Design & Technology, Computing  **Approx. time:** 60 - 120 minutes depending on ability and prior experience of learners |  | | **Key words / Topics**   * BBC micro:bit * system and sub-system * input, process, output * mechatronics * robotics * programmable components * embedded intelligence |
| **Suggested Learning Outcomes** |  | |  |
| * To program the robot buggy system to successfully navigate a maze layout. * To understand how to integrate motors and motor drivers into a programmable system. | | | |
| **Introduction** |  | |  |
| This is one of a series of resources to support the use of the BBC micro:bit in Design and Technology lessons.  Programmable robotic systems are becoming an important part of industrial developments in Design and Technology. Robots are now being developed that can sense changes in their surroundings and respond accordingly.  In this unit of learning, learners will use the BBC micro:bit to develop a robotic buggy that can successfully navigate a maze or path. | | | |
| **Purpose of this activity**  In this activity, learners will work as a team to program the robot buggy so that it can navigate a maze path.  This could be used as a main lesson activity with ‘Navigating the Maze’ as the starter activity. It is an ideal exercise for learners to develop their understanding of programming a system, including the use of motors and motor drivers. | | | |
| **Activity** |  | | **Teacher notes** |
| **1. Assigning groups**  Introduce the activity to learners – they will be programming a robot buggy to navigate a maze. They will be doing this in small groups.  Spit the class into teams (suggested size of three learners per team). Teacher to introduce the maze or pathway that the robot buggy must follow.  **2. Programming and testing the robot buggy**  Learners to work in their teams to create a program that will navigate the maze given to them. They could use the example program as a starting point for this.  The program should successfully control the operation of the motors and wheels, thus controlling the movement and direction of the buggy. If using a 3D maze use should be made of collision detection sensors.  **3. Testing**  At various intervals each team should have the opportunity to test the success of their program on the maze layout prior to the final run through. Teams download and test their programs, refining and improving them as necessary.  **4. Progress review**  Review learner progress part way through the activity and recap key points as necessary. Refer to the examples where appropriate to help guide learners.  **5. Final run through**  Each team has the opportunity to conduct the final run through of their buggy through the maze layout. Class to comment on the success of each run thorough and give constructive feedback on their performance. |  | | The IET TV video – Robot Buggy <https://tv.theiet.org/?videoid=7825> can be shown as an introduction or starter for activities in this unit of work.  This activity should be attempted by learners as individuals in small groups.  **Motors and motor drivers**  Motors and a motor driver circuit will be required to provide the wheels for the robot buggy to move. This can be achieved but using a geared motor for each wheel and constructing a suitable chassis for the buggy, for example by laser cutting. This approach would also require suitable motor driver circuitry as part of the output stage of the system, such as a circuit based on the [L293D motor driver IC.](http://www.engineersgarage.com/electronic-components/l293d-motor-driver-ic)  Alternatively a kit complete with everything needed to integrate the BBC micro:bit into a robot buggy is available from [Kitronik.](https://www.kitronik.co.uk/)  Teachers are advised to refer to the BBC micro:bit’s specification when selecting additional devices and circuitry to use with it.  **The maze**  The maze that the buggy has to navigate can either be a 2D layout (e.g. marked out on the classroom floor using masking tape) or a 3D layout. The Kitronik based kit has the capability to sense when the buggy collides with an object, so this could be used in the case of the latter.  Depending on facilities and space available the teacher could provide a single maze that each team attempts to progress through in turns, or set up different maze layouts for each different team.  **Programming the robot buggy and example**  An example program is available as JavaScript Blocks Editor Powered by Microsoft MakeCode (microbit-robot-jsb.hex) and Python Editor (robot.py) code.  For this program a Kitronik motor board was used at the back of the buggy, and the button\_b input from the motor controller board was wired to a microswitch on the front of the robot. If a microswitch is not available, then button\_b can be used to stop the buggy.  This program drives the robot forward until it hits an obstacle. It then backs off a small amount, turns left a quarter turn and then repeats. It therefore acts as a simple 3D maze solver.  You may need to experiment with which way the motors are wired for the program to work as intended.  This program could form a good starting point for developing an original program for the system.  **To open the example program:**   * Go to <http://microbit.org/code/> * Find the appropriate programming editor. * Click on ‘Let’s Code’. * Drag the relevant program file from the folder where it is stored to the work area on screen. * Visit <http://support.microbit.org/support/home> for help and further support if needed. |
| **Differentiation** |  | |  |
| **Basic** |  | | **Extension** |
| Learners can write the example program directly into the programming software and download it as a starting point. They can then develop this further as required. The example program is included in the accompanying teacher PowerPoint.  Teacher may wish to produce further example programs or a programming framework (e.g. a flowchart showing what the main stages need to be for the program) for groups of learners unable to produce their own code from scratch to cross the maze. |  | | Learners could refine their programs to reduce the time required to move round the maze.  They could also discuss the meaning of the term ‘mechatronics’ – the integration of electronic and mechanical systems.  What products are already available or could be developed that could make use of mechatronics? What are the benefits of integrated electronic and mechanical systems? |
|  |  | |  |
| **Resources** |  | | **Required files** icon-docicon-ppticon-pdficon-ppticon-pdf |
| * Projector/Whiteboard * BBC micro:bit system and online programming software * Internet (to access programming software) * Robot buggy kit (either pre-purchased or assembled using suitable motors and motor driver circuit) |  | | Programming the Buggy presentation  Example BBC micro:bit JavaScript Blocks Editor and Python Editor programs (also provided as .hex and .py files). |
| **Additional websites** |  | |  |
| The following websites can be used for additional technical information or to provide a wider context to the development of the system.   * **IET TV – Robot Buggy:** Supporting IET TV video - ideal for use as part of a starter or introductory activity to support this resource. <https://tv.theiet.org/?videoid=7825> * **BBC micro:bit website:** Website containing everything needed to get started with using the BBC micro:bit! <https://microbit.org/> * **YouTube Video – Programmable Robots:** Video showing examples of programmable robots. A look at an application of programming and what it can be used to achieve. <http://www.youtube.com/watch?v=Io4HiGrFkUw> * **YouTube Video – Robot Buggy:** A robot buggy developed by students at the University of Manchester. <https://www.youtube.com/watch?v=ysR_qFDpPlA> * **Engineers Garage – Motor Driver:** Technical information on choosing the right motor <https://www.engineersgarage.com/choosing-motor-for-robots/> * **Wikipedia – Mechatronics:** introduction to the field of mechatronics. <https://en.wikipedia.org/wiki/Mechatronics> | | | |
|  |  | |  |
| **Related activities (to build a full lesson)** |  | |  |
| **Starters**   * ACTIVITY: Navigating a Maze   **Main**   * ACTIVITY: Programming the robot buggy | | **Plenary**   * ACTIVITY: Future Robots * Opportunities within activity for presentations, peer/self assessment * Reflection on Objectives and PLTS skills used | |
|  |  | |  |