## **Lesson Plan**

| **Lesson 1- Getting started** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Learning Aims:** In this first lesson students will gain an understanding of the features of the Raspberry Pi Pico as a microcontroller. They will recall from prior learning of how microcontrollers play an important part in our everyday lives and how all computers have things in common.**Learning Objectives:**That a Raspberry Pi Pico is a microcontroller that has GPIO Pins that allow connections to external devices  * How to connect, set up and communicate directly with the Pico using Thonny and Micro-python * How to create a sequence of instructions in Micro-Python to control a GPIO Pin on the Pico | | | | | |
| Key Words | | General-Purpose Input/Output (GPIO)Pins, microcontroller, embedded systems, peripheral devices, firmware | | | |
| **Preparation before the lesson:**   * Print required Handouts Ensure sufficient Pico’s and Micro USB Cables are available Ensure Programming devices allow USB access Install Thonny IDE * In addition it's **highly recommended** that teachers undertake the programming activities prior to the lessons. * **Slides contain speaker notes where applicable**   **Optional**: Read the teachers guides for a deeper understanding of the topics covered during the lesson. | | | | | |
| **Assessment opportunities** | | Correct implementation of coding tasks Completion of Challenges activities Correct answers to plenary questions | | | |
| **Lesson Resources** | | [*Lesson 1 Starter Activity*](https://docs.google.com/document/u/0/d/1T_hhWYQ0-HQX99y2J0CGg-veBVHLGT6e/edit)  [*Primm Activity Worksheet*](https://docs.google.com/document/d/1KO1q3nRgHO_120P17NM6mvL9vr7npaug/edit?usp=sharing&ouid=114120863087517164066&rtpof=true&sd=true)  [*Lesson1\_Introduction\_to \_the\_pico.docx – question/info sheet*](https://docs.google.com/document/d/1oBU8lqr3G8craFgcXhLhRVgMV44NZV-C/edit?usp=drive_link&ouid=114120863087517164066&rtpof=true&sd=true)  [*Setting up the pico and Thonny*](https://docs.google.com/document/d/19LGAEQ1gJcNccbiF3c9ZYqR5cKSe_y8Q/edit?usp=drive_link&ouid=114120863087517164066&rtpof=true&sd=true)  [*Programming the LED*](https://docs.google.com/document/d/1H4OsJXfUrevHXXvWEkSlKdVSfz1CVPIr/edit?usp=drive_link&ouid=114120863087517164066&rtpof=true&sd=true)  [*Lesson 1 ppt*](https://docs.google.com/presentation/d/1kiNJqf8vuV-wlWt7rdhwG5Uu23nRj6yc/edit?usp=drive_link&ouid=114120863087517164066&rtpof=true&sd=true) | | | |
| **Teachers Guides:** | | | [Intro to the pico teacher guide](https://docs.google.com/document/d/1TUoFmw04BkGhh1rsQga3S5n1-RW7K_sq/edit?usp=drive_link&ouid=114120863087517164066&rtpof=true&sd=true)  [Teacher guide to the Pico Pins](https://docs.google.com/document/d/1TUoFmw04BkGhh1rsQga3S5n1-RW7K_sq/edit?usp=sharing&ouid=114120863087517164066&rtpof=true&sd=true) | | |
|
| **NC LInks** | | | * design, use and evaluate computational abstractions that model the state and behaviour of real-world problems and physical systems * understand the hardware and software components that make up computer systems, and how they communicate with one another and with other systems * use two or more programming languages, at least one of which is textual, to solve a variety of computational problems; * Understand how computers represent data | | |

| Starter activity (Slide 2&3)  5 mins | **Microcontrollers**  **Slide 2:** Learners are re-introduced to the term *microcontroller and should be encouraged to think about types of embedded systems they know.*  **Slide 3** Using the starter worksheet sheet, learners think about input and output devices on embedded systems. Specifically they are challenged to consider the different inputs and outputs of a home assistant such as the Alexa or google home assistant.  **Here's a general overview of the inputs and outputs commonly associated with smart home assistants:**  **Inputs:**  **Voice Commands:** The primary input method for interacting with Smart assistants is through voice commands. Users can activate Smart assistants by using a wake word , followed by a command or question. Smart assistants then process the voice input to understand and respond accordingly.  **Touch Controls:** Some Smart assistants devices, such as the Echo Show or Echo Spot, feature touchscreens that allow users to interact with visual content and controls in addition to voice commands.  Remote Controls: Certain Smart assistants-enabled devices come with remote controls, which provide an alternative input method for users who prefer physical buttons over voice commands.  Many smart assistants will also have push buttons for manual control of set up or audio output  **Bluetooth Pairing:** Users can pair Bluetooth-enabled devices, such as smartphones or tablets, with Smart assistants-enabled devices to stream audio or perform other functions from external devices or sensors.  **Outputs:**  **Voice Responses:** Smart assistants respond to voice commands with spoken feedback. Responses can include answers to questions, confirmation of actions, or instructions for further interaction.  **Visual Display:** Some Smart assistants devices, like the Echo Show, Echo Spot, and Fire TV Cube, feature built-in displays that can show visual content, such as weather forecasts, calendars, news updates, song lyrics, or video streams.  **Audio Output:** All Smart assistants-enabled devices have built-in speakers that produce audio responses from Smart assistants, including music playback, spoken feedback, alarms, notifications, and more. Additionally, some devices allow users to connect external speakers for enhanced audio quality.  **Smart Home Control:** Smart assistants can control various smart home devices, such as lights, thermostats, door locks, and cameras, either through voice commands or via the Smart assistants app. Outputs in this context include actions performed on connected smart home devices, such as turning on/off lights or adjusting the thermostat.  **Notifications:** Smart assistants can provide notifications through audio alerts, visual indicators (e.g., light rings on Echo devices), or both, to inform users of incoming messages, reminders, calendar events, or other notifications from compatible apps and services. |
| --- | --- |
| **Activity 1**  (Slide 4-6  5 mins | **Introduction to physical computing and Raspberry Pi Pico** Unit aims and lesson objectives are shown on slide 4 and 5 respectively and should be presented to the students.  **Slide 6 Introduces the Pico**   * **Definition**: A microcontroller is a compact integrated circuit designed to govern a specific operation in an embedded system. It typically combines a processor, memory (RAM and flash), and peripherals on a single chip. * The PICO has 26 multipurpose GIO pins which allow us to attach peripheral devices * Using Micropython we can write programs to control peripheral devices attached to the PICO   The Pico is a microcomputer and can be used for a wide variety of uses. They are particularly good for creating embedded systems such as the amazon Alexa. This can be achieved by adding additional peripheral devices in the form of input and output devices such as speakers and microphones, sensors or push buttons.  The WWW is home to a wide range of projects built with the Pico and other similar microcontrollers. A simple web search will produce lots of results of projects built using Pico.  The Pico has 26 general purpose pins that have multiple uses. It is these pins that are used to attach additional peripheral devices to the Pico using a range of different protocols we can then control these devices.  A **peripheral** in computing refers to any external device that provides input to, or receives output from, a computer system or microcontroller. These devices are not part of the core architecture but extend the functionality of the system.  A **protocol** in computing refers to a set of rules and conventions for communication between devices or software systems. This is covered in more detail in a later lesson. |
| **Activity 2** (Slide7)  5 mins | Give an introduction to the GPIO Pins (**G**eneral **I**nput and **O**utput **P**ins.  Students should be provided with the PIN out diagram reference sheet. There are a total of 40 Pins on the Pico some are ground Pins that can be used for developing circuits to control peripheral devices, others are used to provide power to these devices. Different pins are able to draw on different protocols as different hardware requires different protocols in order to work with the Pico.. At this stage there is no need to go into a great deal of depth on the different types of pin protocols. Various types of pins are introduced in more detail in later lessons as students encounter them for different aspects of the project.  Use the slide Image of the RP2040 Pico and its Pin, point out the **boot select button** top left hand side for installing firmware and the **USB port** at the top for connecting to the computer for programming.  **Definition of Firmware** is a specialised type of software required by hardware components to control their functions. Firmware is closely integrated with the hardware, providing low-level control and managing essential device operations.  **The LED** is attached to a PIN 25 it is this that students will control in today's lesson **\*note** on a PICO W the led is no longer on PIN 25 but is attached directly to the CHIP and is referred to as “LED” in micro python see sample file  **Draw student’s attention** to the fact that the **physical pin numbers do not correlate to the GPIO numbers** and explain that this is because some physical PINs have other purposes such as voltage and ground etc to create circuits explain that Inputs and outs need voltage – power as well as instructions in order to work Draw students attention to the slightly different shape of the ground Pin on the actual Pico.  The **teacher guide** outlines the protocols available for each Pin for reference |
| **Activity 3**  Slide 8-9  25 mins  Slide 10 | Using the PRIMM model and the PRIMM worksheet, students predict what may happen when the code on slide 9 is executed. Students should be able to predict the outcome based on the earlier discussions but may need some prompting. give students a few minutes to make their predictions.  Demonstrate the program execution so that students are able to see if their predictions are accurate.  Give an overview of the code **using slide 9** challenge students to predict some changes such as sleep  **The Code is provided but it is recommended teachers work through this independently so they experience the tasks as students will.**  **Code overview:**  Explain to students that libraries help to simplify code by making use of code written by others. Manipulating hardware on the pico is simplified by the use of libraries. Students may already be aware of the purpose of sleep from the time library.  GPIO pins are objects. Objects have Methods that allow us to control the object and attributes that are its individual values.  To instantiate/create a pin object a variable is needed to store the object and then the object is allocated to a Pin number (attribute)and set as either an input or an output type (attribute).  In this case the variable **led\_onboard** is the **variable ID** of the Pin number 25 and it is set as an output pin.    The **pin.value()** method is called to either get or set the value of the pin. In this example the pin value is set by passing in the desired value. When the Pin value is set as 1 a current flows through the PIn turning on the LED. This current is set at around 3v when the Pin value is set to 0 the current is removed. The sleep provides a delay so that the change in states can be seen. This is the same action as turning on and off a light switch.  Having broken down the code it is time to execute the code and allow students to view the output to LED.  The worksheet takes students through challenges to modify the code prompting them to consider what would happen if style questions. This can be done verbally by questioning and demonstrating the results if preferred.  **Distribute the PIco devices** to students and provide students with the **worksheet setting up Thonny.** The pico must be plugged in to configure the interpreter.  Alternatively you can **demo to students** this simple process before they begin work on coding.  **Once students have configured the interpreter.**  Students can then work through the **activity worksheet** which guides them through a series of short exercises from turning on and off the LED, using the toggle function and changing the duration of the sleep to control the duration of the blink. Students then attempt independent challenges to consolidate their skills and knowledge.  **The starter code** is provided and can be distributed to the students as python scripts or as handouts depending on the level/experience of students.  **Draw on links to ASCII and Morse code** how each uses only two symbols to represent text.  While some students will need the whole lesson to complete the worksheet tasks, others will complete them with time to spare. The stretch **games activities worksheets** are very motivational for more confident students, and you may wish to give these as an optional homework challenge with the promise of testing time in future lessons. |
| **Plenary**  (Slides  5 mins | **Quick fire quiz**  This can be done as an exit ticket style or answers can be written in books or on the Primm worksheet |
| **Homework** | **Homework worksheet** provides and overview of the Pico including its specifications. This activity can be used to introduce the Pico or could be used as a homework sheet |