## **Lesson Plan**

| **Lesson 2 - Input with buttons** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Learning Aims:** In this lesson students will be introduced to breadboards, a tool used for prototyping with microcontrollers. They will then learn how to create simple circuits to input digital data via push buttons.  **Learning Objectives:**  * How breadboards are used as a tool for prototyping with microcontrollers. * How to create simple circuits to input digital data using a push button. * How to create a sequence of instructions to control the button input | | | | | |
| Key Words | | Input device, debounce, breadboard, jumper wires, resistors | | | |
| **Preparation before the lesson:**  Print required handouts **Prepare physical resources**  1 push button per student + spares for extension challenges  2x M2M jumper wires per button 2x M2M jumper wires per breadboard  **Consider** adding ground and power rails to breadboards prior to the lesson using U shaped Jumper wires for a semi permanent set up  In addition it's **highly recommended** that teachers undertake these activities prior to the lessons.  **Slides contain speaker notes where applicable**  Pins are also associated with protocols such as digital or analogue. The **Teacher Guide gives more details on PICO OBJECTs it is recommended you read this prior to the lesson.** | | | | | |
| **Assessment opportunities** | | PRIMM activity and discussions  Correct implementation of coding tasks Completion of Challenges activities Correct answers to plenary questions | | | |
| **Lesson Resources** | | [*Lesson 2 Starter activity*](https://docs.google.com/document/d/1QsfNdlk9kt4P1cwPT9cViL56qgLP189X/edit?usp=drive_link&ouid=114120863087517164066&rtpof=true&sd=true)  [*Lesson 2 Activity worksheet*](https://docs.google.com/document/d/1dQY31McRji4-zNbSuHMQ1771lGGs3DnA/edit?usp=drive_link&ouid=114120863087517164066&rtpof=true&sd=true)  [*Lesson 2 Wiring Diagram*](https://docs.google.com/document/d/1Np_twBgoFbcOx-v5v7Fg2zCCH7j_5hUp/edit?usp=drive_link&ouid=114120863087517164066&rtpof=true&sd=true)  [*Lesson 2 ppt*](https://docs.google.com/presentation/d/1NAWQentgVkcEHPsQ5pa7JU1dhMg7OLFG/edit?usp=sharing&ouid=114120863087517164066&rtpof=true&sd=true) | | | |
| **Teachers Guides:** | | | [Introduction to Objects](https://docs.google.com/document/d/1csX7IE7e4YacsG5z-crn9UIUiE5kY61z/edit?usp=drive_link&ouid=114120863087517164066&rtpof=true&sd=true)  [Handling Button Debounce](https://docs.google.com/document/d/19GkYqudG334g6eoQrewqmb3ORIJIOUfE/edit?usp=drive_link&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; | | |

## **Lesson 2- Buttons Outline plan**

| Starter activity (Slide 2)  5 mins | Ask students to recall the core components of a computer system and the difference between an embedded system and a General Purpose Computer.  This should be *retrieval of prior learning from earlier key stage 3 learning*  **Answer Key:** Students should be able to build on prior learning to identify Input, Process, Output and Storage as common components of computing systems and that embedded systems have a single function a general purpose computer can offer a wider range of functions. |
| --- | --- |
| Slide 3 & 4 | The lesson aims are identified these should be presented to the students  Lesson 4 recaps the last lesson  Remind students they are building up to a complete embedded system. This will require connecting input and output devices to the Pico. Last lesson they programmed an output device via the onboard LED. Can students recall how the output was controlled?  **Answer Key:** Turning on and off the voltage of the Pin setting the Pin value to 1 & 0. .value() and .toggle() where the methods used |
| **Activity 1**  (Slides 5-6)  5 mins | **PRIMM – Predict activity and Run.**  Provide students with a copy of the PRIMMworksheet containing the starter code and allow them time to read and annotate the code. Encourage them to make a prediction on what will happen during execution.  Demonstrate the program execution (**code provided)** then walk through the code using **slide 6** have students **annotate code** on the worksheet filling in any gaps they may have. Draw students' attention to **character case** and **indentation** to help reduce errors during inputting.  **Discuss** how GPIO pins are objects that have an identifier(name) type either input or output. Last lesson the pin was used as an output pin where the values could be set. this lesson a pin will also be set as an input pin where values can be read (get methods)  **Code Key:**  To instantiate the Pin object as an output pin, allocate it to a Pin number and set as either an input PULL\_UP or PULL\_DOWN this pulls the voltage up to 3v or down to 0.  Pin values can then be read for either 0 or 1. The pins can float with a voltage anywhere between the 0 and 3v which can result in miss reads so whilst the PULL\_UP or DOWN parameter is optional it is recommended to use this parameter. The default is PULL\_DOWN. This means the pin is always set to 0v and awaiting an interrupt in signal. The interrupt is read as 1.  Using a pull down requires the button to be wired to the 3v, whereas using a pull up pin requires the button to be wired to the ground pin. In this example the pin down is used.  The **debounce teacher guide** provides a more detailed explanation of why sleep is required to prevent false readings of pins when a button press is made. Essentially the pin can be read so quickly that it is read twice in a single manual press. |
| **Activity 2** (Slide 6 -8)  15 mins | **Introduce the breadboard** walk through the anatomy of the breadboard so that students fully understand how the pins will be connected on the rails of the board.  It is a good idea to give students a breadboard to handle before explaining the anatomy. Taking the bottom sticky pad off a breadboard will allow students to see inside the breadboard and allows them to understand how the rails are connected.  A breadboard is a fundamental tool used in electronics prototyping to create temporary circuits without the need for soldering. It allows engineers, hobbyists, and students to quickly build and test electronic circuits. Here's a breakdown of the anatomy of a typical breadboard:  **Board Base**  The main body of the breadboard is often made of plastic. It provides the structure for the board and holds the metal contact strips in place.  **Terminal Strips:**  The long strips running along the sides of the breadboard are called terminal strips. These strips are typically used for power and ground connections.  **Bus Strips:**  The shorter strips running perpendicular to the terminal strips are called bus strips. Each set of bus strips is usually used to connect related points in a circuit.  **Rows and Columns:**  The holes on the breadboard are organised into rows and columns. Each row typically consists of five holes, and there are usually multiple rows and columns.  **Power Rails:**  The two vertical columns on the sides of the breadboard are known as power rails. One side is often used for the positive supply voltage (Vcc), and the other side is used for ground (GND). These rails are connected internally, providing power throughout the corresponding rows.  **Tie Points:**  The holes in the rows and columns where components are inserted are often referred to as tie points. Components, such as resistors, LEDs, and integrated circuits, are inserted into these holes to make electrical connections.  **Jumper Wires:**  Jumper wires are used to create electrical connections between various points on the breadboard. They can be straight or flexible wires with connectors that fit into the holes. |
| **Slide 8/9** | Distribute the circuit diagram and the physical components before tackling this slide.  **Slide 8 Briefly Introduce the Button**  Push buttons come in a variety of different shapes and sizes  Momentary buttons alter state only while pressed there are other types of buttons where once depressed remain depressed until they are pressed again  Push Buttons are an effective way to receive external input they are easy to program and easy to use.  Push buttons are used on everyday devices such as keyboards, mice, keypads on games consoles and simple on/off buttons  Explain how buttons have **4 legs** and that these are in pairs. The slide image demonstrates how these legs are grouped  students should be able to identify the inward facing legs on their buttons before adding the button to the breadboard.  **Introducing the circuit diagram**, Ask students to first install the Picoon the breadboard so that it straddles the centre of the breadboard with the USB port at the top of the board as shown on the slide. Aligning the pins carefully to prevent them from getting bent.  Creating a ground and power rail at this stage will make later lessons set up simpler.  The ground used in the circuit diagram is the 3rd pin down on the right hand side and the 3v is the 5th pin down. **It is really important that students do not put the ground and 3v in the same rail** as this will create a circuit and the Pico will get extremely hot very quickly. With less confidence students consider setting up the breadboard with the power rail and ground rail using U shape Jumper wires as a permanent set up, reducing the likelihood of error.  In the diagram GPIO 14 is used as the signal pin the second from the bottom on the left side of the Pico is used. Have students annotate the diagram to identify the Pin numbers on the diagram  The process of wiring the board can be daunting for students in the 1st lessons. Setting up the ground and the live connections before the lesson removes some issues in this first breadboard lesson.  **Group students in small teams to wire the circuits** and to check each other’s wiring against the diagram. This will help to reduce errors and provide students with confidence in completing the activity.  It is also advisable to have a completed circuit for students to be able to check against. This also provides a quick swap out if a student’s circuit does not work during testing. This can help greatly with debugging issues with wiring or code to be able to quickly test a known working circuit.  Allow time to complete the circuits encouraging students to help each other. This gets much quicker with experience. |
| **Activity 3**  Slide 9-10  25 mins | **The Programming activity worksheets** guide students through a number of activities with the button as an input. Students can complete these working either individually or in pairs.  Having students in buddy groups of two pairs or 3 or 4 students ‘ will help greatly with troubleshooting.  The starter code is provided and can be distributed to the students as python scripts to less confident students. Alternatively the handout guides students through the challenges.  Students attempt independent challenges as appropriate, while some students will need the whole lesson to complete the tasks, some students will finish quickly and be able to attempt the optional games activities worksheet. These are motivational for more confident students and serve as a great homework activity. |
| **Plenary**  (Slides  5 mins | **The activity sheet has a single plenary question for students to answer which is also on the final slide**. Students can answer on the sheet or in books or on the whiteboard.  A quick verbal recap of some key terms is also recommended |
| **Homework** | **Optional** Research the following different types of buttons and switches that can be used in electronic circuits. Find an image of each and explain how they work  **Capacitive Touch Button**   * **Description**: Detects touch using changes in capacitance. * **Applications**: Modern touch screens, touch-sensitive devices.   **Toggle Switch**   * **Description**: Manually operated switch that stays in its position (on or off) after being toggled. * **Types**: Single Pole Single Throw (SPST), Single Pole Double Throw (SPDT), Double Pole Single Throw (DPST), Double Pole Double Throw (DPDT). |