

## Medium Term Plan: Supporting Implementation of LTP/Progression Grid

|  |                              |
|--|------------------------------|
| Subject: Computing – Programming: Sensing  | Year: UKS2 – Year A – Summer |
| NC/PoS: <ul style="list-style-type: none"><li>• Design, write, and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts</li><li>• Use sequence, selection, and repetition in programs; work with variables and various forms of input and output</li><li>• Use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs</li><li>• Select, use and combine a variety of software (including internet services) on a range of digital devices to design and create a range of programs, systems and content that accomplish given goals, including collecting, analysing, evaluating and presenting data and information</li></ul> |                              |
| Prior Learning (what pupils already know and can do)<br>Understanding giving and following instructions, using floor robots to create and debug programs, creating a sequence of commands to follow a routed, using Scratch Jnr to create a program using blocks, how to add music and link to motion, how to use the pen tool within programmes, how to use repeat and create count-controlled loops, how to create a programme in a text-based language, how to use conditions and selection to ask questions,   |                              |
| End Points (what pupils MUST know and remember) <ul style="list-style-type: none"><li>• To create a program to run on a controllable device</li><li>• To explain that selection can control the flow of a program</li><li>• To update a variable with a user input</li><li>• To design and create a project that uses inputs and outputs on a controllable device</li></ul>  |                              |
| Key Vocabulary<br>Micro:bit, MakeCode, input, process, output, flashing, USB, trace, selection, condition, if then else, variable, random, sensing, accelerometer, value, design, task, algorithm, step counter, plan, create, code, test, debug   |                              |
| Recommended Resources:<br><a href="https://tinyurl.com/UKS2-Sensing">https://tinyurl.com/UKS2-Sensing</a><br><a href="https://tinyurl.com/UKS2-Sensing-OA">https://tinyurl.com/UKS2-Sensing-OA</a>   |                              |
| Session 1: Micro:bits<br><br>What is a micro:bit? How can the device be an input, process and output? How is the device programmed? How does MakeCode compare to Scratch? Can we create a basic program to control lights and sound on the device? How do we send the program to the device from the laptop?<br><br>Vocabulary: Micro:bit, MakeCode, input, process, output, flashing, USB, trace  |                              |
| Session 2: Conditional Inputs<br><br>What inputs does the micro:bit have? How can these inputs be used to control the flow of the program? How can 'if...then...else...' statements be used to influence the program? Can different inputs operate different outputs? Can we create a simple program to suit a real-world situation? Can we use the accelerometer to sense motion? Can this input work in the same way as others?<br><br>Vocabulary: Selection, condition, if then else, variable, random, sensing, accelerometer, value   |                              |
| Session 3: Step Counter<br><br>How does a step counter work? What features are similar to the micro:bit? What will be the input, process and output functions be for this purpose? Can we design an algorithm and program to fit this purpose? Can we test the code using Make Code an emulator to debug before loading the program to   |                              |

## Medium Term Plan: Supporting Implementation of LTP/Progression Grid

the physical device? How can our peers help to evaluate this project? How can we use feedback to make improvements?

Vocabulary: Micro:bit, design, task, algorithm, variable, step counter, plan, create, code, test, debug

Future learning this content supports:

The content of this unit will support other units on developing programs for a micro:bit