# Lesson plan - Lesson 4 Intelligent cooling fan part I

## Using the MonkMakes solar experimenters kit

## Introduction

In this lesson you have time to introduce the intelligent cooling fan program to the students. In addition you will ensure that they understand what an algorithm is and how pseudocode is written. The intelligent cooling fan is a quite involved circuit and it’s worth having enough time to charge the solar store to see how the program works. For this reason it is split over two lessons, with time allocated in the final lesson of the 6 lesson scheme to focus on modification and making.

Learning objectives

* To successfully set up a micro:bit/intelligent cooling fan circuit and flash code to the micro:bit
* To recall the definition of an algorithm
* To apply their knowledge of algorithms and pseudocode to a specific problem
* To be able to read and interpret a program containing user defined functions

## Keywords

Solar store, voltage, super-capacitor, terminals, bulb, harvested, circuit, GPIO, pins, iteration, loops, trace tables, testing, algorithm, functions

## Preparation

**Subject knowledge:**

This lesson is suitable for a class who already has a basic knowledge of programming using python. They should also have previously used algorithms and pseudocode as this lesson starts with a diagnostic question starter testing aspects of their knowledge of both. The coding for the micro:bit is written in a reduced version\* of python called micropython but for practical purposes the syntax will appear the same to the user. In addition the students should have already used the micro:bit, practising simple set up and smaller micro:bit specific programs so that they are used to building and flashing programs. Initially, building the circuit for the “intelligent cooling fan” is a question of copying a diagram so a practical demonstration of handling the components gently and clipping with alligator clips would suffice.

**Pedagogical approach:**

The lesson is planned using the PRIMM pedagogy which stands for:

P - Predict

R - Run

I - Investigate

M - Modify

M - Make

For this reason it is important that the class have quick and easy access to the coding of the program which can be accessed via: https://tinyurl.com/2wtkdh2d

The very first step is for the students to view the code and attempt to predict what the code does. We use a tool called a **code conversation** which provides the teacher with a conversation style **talkthrough** of the micropython code for you to support your students’ developing knowledge. Initially the lesson focuses on the predict, run, investigate sections of the pedagogy. Occasionally daggers (††) are used to clarify things in the code conversations, instead of asterisks, which could be confused with the multiply symbol.

**Practical set-up and development environment:**

We recommend either using the on-line Python editor (https://python.microbit.org) or Mu (https://codewith.mu). Or you could manage your coding in micro:bit’s own environment <https://classroom.microbit.org/> which enables you to push code to the whole class and manage their activities.

**Resources that you need:**

* Presentation
* Starter worksheet questions
* Starter worksheet answers
* Code conversation full solution
* Code conversation template (level 1 - basic)
* Code conversation template (level 2 - intermediate)
* Plenary worksheet
* Plenary worksheet answers
* Hardware Per pair - 1 micro:bit, 1 Solar panel, 1 Solar store, 1 USB/micro USB cable, 8 alligator clip leads, 1 motor, 1 fan
* Access to appropriate development environment (the on-line Python editor, Mu or micro:bit classroom - Mu is needed for the last lesson in the scheme)

## How the students’ progress is assessed

**Intelligent cooling fan I**

The lesson begins with a short quiz using diagnostic questions on the subject of algorithms. The lesson is interspersed with pair or whole class discussion opportunities. As this is predominantly a practical lesson, observing the students’ success in flashing the program, compiling the circuit and making small successful modifications to the program will also enable you to assess their progress. The learners also complete a worksheet at the end of the lesson which demonstrates their understanding of key principles conveyed during the lesson. Because of the complexity of the circuit and the need to leave some time for charging, the lesson is split into two with the code conversation started in part I and completed in part II. There is then time left for the modification/make part of the PRIMM methodology in the final lesson of the scheme.

## Plan (with approximate timings)

| Starter activity 8 mins | **Algorithm diagnostic questions**  The slide that greets the student is inviting them to complete the algorithm diagnostic questions worksheet. After the students have completed the worksheet, quickly go through the solutions and ensure that they correct any wrong answers on their starter worksheets. |
| --- | --- |
| Objectives & Homework4 mins | **Objectives and homework review**  Review the objectives and last week’s homework with the students to put the lesson in context. |
| Activity 1 8 mins | **Algorithms and pseudocode**  Using the slide show, have the students explain what their definition of an algorithm and pseudocode is. The students are then invited to take a brief explanation of a program and put it into some recognisable form of pseudocode. |
| Activity 2 5 mins | In order to put the lesson and the intelligent cooling fan in context the student  **Predict: students view code and attempt overview code conversation**  The students are shown the code and are asked to predict what the whole program does i.e. an overview in a couple of sentences. You should not expect accuracy here, just an attempt and some curiosity.  For example:  The intelligent cooling fan turns on if the ambient temperature is warm and the solar store is more than half full, then off if the temperature is cold or the solar store is not full enough. In addition the program puts a message on the micro:bit which says YES when the fan is on and NO when the fan is off. Buttons A and B are provided for an override that allows you to request the fan manually. B overrides and A cancels the override. |
| **Activity 3**  5 mins | **Run: students compile the circuit, flash and run the code**  Share the slide with instructions on the board to enable the students to build the circuit, flash and run the code. Ask them if the circuit behaves as they would have expected from the **predict** phase of the lesson. Depending on how bright the day is, or the proximity to bright indoor light the solar store may take some time to charge so once the circuit is compiled it is important that the students leave the fan off and allow it to build some charge so that they can see the bar chart increasing on the micro:bit. There is plenty to do with the code conversation while they wait. |
| Activity 4 19 mins | **Investigate (part I): teacher and students use correct terminology to identify the syntax and features of the program**  Using the A4 worksheet, the learners attempt to complete the first part of the code conversation matching the correct explanation to each line of code.  Using the slides talk through the different steps in the program. |
| **Plenary**  10 mins | **Intelligent fan plenary worksheet**  Students complete the plenary worksheet testing the application of pseudocode knowledge. |
| **Homework** | **None set for this lesson.** |

## The Author

This lesson plan and all its parts were created by Dr. Paula Beer of Beer Academic Consultancy in collaboration with Monk Makes Ltd.



Dr Paula Beer has taught Computer Science and IT education to new and established teachers since 2007. Her own research has focused on the use of play and collaboration in computer science. She enjoys supporting practicing teachers by designing accessible lesson planning materials to get students engaged in computer science through play and collaboration. Paula has also produced educational materials for The Raspberry Pi Foundation, been a secondary school teacher, written a successful book (Hello App Inventor!) and has previously worked in IT project management for a media company and for the NHS.