

Internal Assessment Resource

Achievement standard: 91881

Standard title: Develop an electronics outcome

Credits: 6

Resource title: Data Logging System

Resource reference: Digital Technologies & Hangarau Matihiko
1.5B Version 1

Student/Ākonga instructions

Introduction/Kupu Arataki

This assessment activity requires you to develop a data logging system and computer program that can automatically respond to environmental data.

You are going to be assessed on the quality of the refined system that you develop and on how well you structure and refine your program. A well-structured, refined program will have code that is easy to follow, update and maintain, and is free from bugs.

This is an individual assessment task.

Due Date: 8th June 2018

Task/Hei Mahi

You are going to create a data logging system using basic electronic components and a microcontroller. The system and embedded software must perform to the specifications outlined in the task.

You must:

- choose appropriate component types and values for your system
- modify, test and debug a functional model of your system
- use data sheets or calculations to assist in choosing appropriate component types and values for your system
- write well-structured, clearly commented, and readily understandable software.

As part of developing your outcome you will need to describe the interfaces and functions of the components of the systems used, explain the behaviour and function of the electronics outcome and justify your choice of the components and systems you have used. Examples of concepts that you can refer to when demonstrating this requirement are:

- a circuit as a complete path
- voltage as an energy path
- current as rate of flow of charge
- distribution of voltage and current through a circuit (series and parallel circuits)
- conduction (limited to the macroscopic behaviour of conductors, insulators, and semiconductors)

- circuit subsystems
- symbolic conventions and schematics
- hardware (for example components and combinations of components)
- embedded systems as software subject to hardware constraints.

As you complete the task, gather evidence to include in your portfolio on Google Classroom. This will be handed in with your completed data logging system.

In your portfolio, you must include:

- circuit diagrams
- layouts
- screenshots of embedded software programs for each step of the task
- evidence of iterative improvement of the outcome and software program you developed.

You may also want to include:

- annotated diagrams, screenshots and photographs, or videos
- journal entries with tables that describe your trialling and testing of circuits and programs
- your responses to teacher questions
- interview notes from a scheduled teacher consultation
- test results.

Electronic System Specifications

Your specifications must be agreed with your teacher prior to construction, and include sensors and a method to read environmental data, respond to data and provide output (this may be physical, auditory or visual).

The system will be a microcontroller-based circuit, and may use the following components:

- Microcontroller (BBC Micro:bit) or other suitable system
- sensors e.g. temperature, light, humidity, sound, level detectors, motion.
- displays, LEDs and/or buzzers/speakers as indicators
- actuators/motors to control relays, levers etc.

After the hardware has been selected:

- attach inputs or outputs to microcontroller and modify embedded programs that test each microcontroller input and output to ensure they function as expected.
- integrate embedded software programs to develop the outcome to meet specifications.
- the embedded software program needs to be well structured, readily understandable, and clearly annotated. The program may be based upon commonly available program stubs, sketches, examples, etc., but needs to be modified appropriately to meet the context. This means your program should:
 - be clearly set out and correctly indented
 - include comments that explain exactly what the program is doing at each step
 - use labels so that it is easy to read and understand the program.
- You must create your code in an iterative manner. For example, break down the program into smaller achievable parts then code, test, debug and evaluate each part before moving on the next progressive improvement. It is advisable to save each

version of the code as you make progress.

Note: To test a program in a comprehensive way, you should think about how you will test the program for various cases such as expected, boundary and unexpected input, what happens when data is not what is expected or there is a fault in a sensor. It is often useful to note down what you want to test and what you expect to happen, as well as what actually happened.

- Ensure that you comment your code appropriately as you develop it and use variable names and comments that describe code function and behaviour.
- Ensure that you have followed conventions for the programming language of your choice and that you have chosen a well-structured, logical response to the task. You should ensure that your code is robust and that it handles expected, boundary and invalid values.
- Wherever possible you should try to ensure that your code has a flexible structure to allow for continued development.

Note: Your program must include variables of at least two different types, and sequence, selection, and iteration control structures. The program must also include one or more of: data stored in collections (e.g. lists, arrays, dictionaries); user-defined methods, functions or procedures.

Completing the task

As you perform the task, make notes and gather evidence for inclusion in your portfolio. For example:

- show your use of appropriate resources and techniques used in developing your data logger.
- document the testing and iterative improvement you have made to your data logger and program throughout the development and testing process. For example:
 - selecting the best type and value of components
 - selecting the best arrangement of components
 - adjusting hardware input and/or output parameters
 - adjusting software parameters
 - using a multimeter to measure and report voltage and/or current levels at indicated points.
- include diagrams, annotated photographs, written descriptions or video evidence to show your understanding of the interfaces and functions of the components you have selected for your system. This evidence should explain the behaviour and function of the outcome and show reasons (justify) for your choice of components and systems. For example:
 - explain your choice of basic components to build your circuit, for example resistor to limit current and/or transistor to amplify current
 - explain any calculations and/or research, for example manufacturer data sheets, that you used to determine the best components for your circuit
 - describe how your circuit behaves in terms of the basic concepts, for example, a voltage divider, or the effect of a low battery
 - explain the operation, function, and calculation of the electronic components you used in your circuit.
- show evidence of your testing procedures to debug and diagnose the electronic system and how you have modified and debugged the embedded software program to ensure it is fit for purpose and to demonstrate how the reliability of the system has been improved.

- show evidence as to how you have addressed any implications relevant to your outcome.

Resources

Useful websites:

<https://makecode.microbit.org/>

[SparkFun Tutorials](#)

[Inventor's Kit Tutorials](#)

[Khan Academy Electronics Tutorials](#)