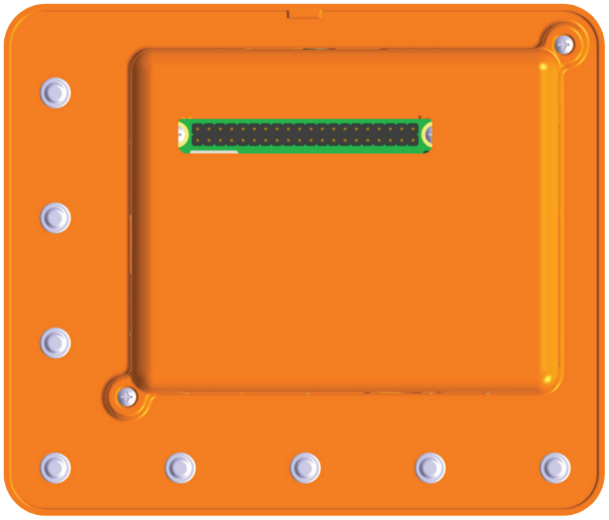
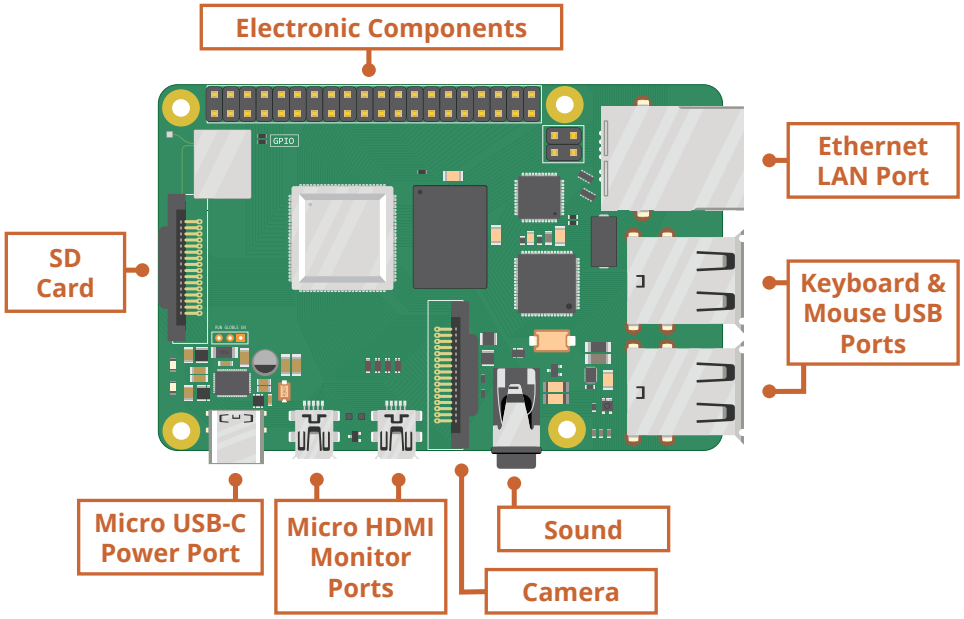
Introduction to Raspberry Pi

Inside your **Smart Module** is a **Raspberry Pi**. The Raspberry Pi is a fully functioning microcomputer that plugs into a monitor and uses a standard **keyboard** and **mouse**. Like your computer, it runs on an operating system (OS) called the Raspberry Pi OS. It can do everything your desktop computer can and can also interact with a variety of different products—including the Smart Rover. By programming the Pi, we can enable the Rover and its components to do a variety of things.





The Smart Module houses a Raspberry Pi 4. This Pi comes pre-configured with an SD card installed with the Raspberry Pi OS. **To use the Raspberry Pi, you will need an external keyboard and monitor with USB connectors.**

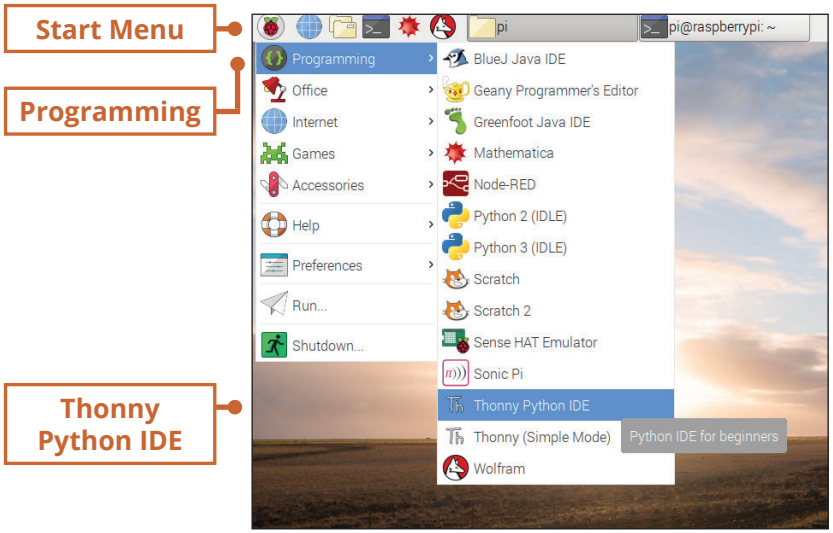
Connecting your keyboard, mouse, and monitor: To get started, you will need to connect your keyboard and mouse USB connectors to the two USB ports on the Pi (either port is fine). You will also need to connect the power cord to the Pi and to an outlet. Your monitor will be connected to the Pi through the left-hand HDMI port. An optional second screen can be connected to the right-hand port.

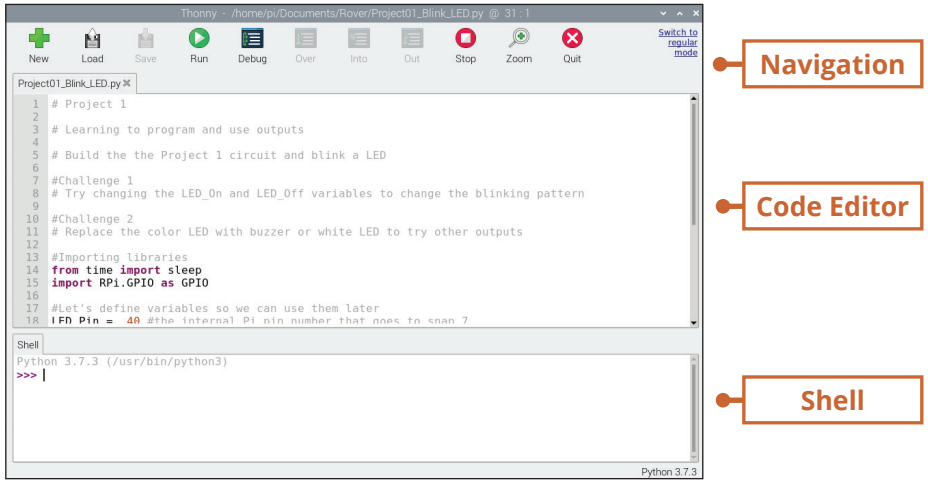
The camera will be used in later projects, while the network, sound, and electrical components ports will not be used in this manual. The Pi does not have a power switch; once the power port is connected, it will turn on. After a few moments, the Raspberry Pi OS desktop will appear.

When you start the Pi for the first time, the Welcome to Raspberry Pi application may pop up and guide you through setting up your country, language, time zone, password, and wireless network. There may also be software updates to be installed.

Programming the Raspberry Pi

Writing and running programs allows us to "speak" to computers and tell them the tasks we want them to do. Different programming languages are used for different types of tasks. Like spoken languages, programming languages each have their own types of “grammar” and punctuation, but learning one helps you understand the underlying logic of all of them. For the Smart Rover projects, we will be programming in **Python** using the **Thonny Python IDE** application.





Your Raspberry Pi comes with a variety of programming applications, which you can access under **Programming** in the menu, and click on **Thonny Python IDE.** IDE stands for Integrated Development Environment. IDE applications help us organize and manage our programs.

When you load up Thonny, across the top bar there are icons to create a **New** program, as well as **Save, Load, Run,** and **Stop** icons. Each project will be its own program, saved in its own file. In the middle of the page is the code editor, where you will write your code. At the bottom is the **Shell**, where you will see outputs from your code.

Once Thonny is open, click **Load** in the top left to select a program from the Projects folder. Once you have loaded a program and are ready to run it, click the green **Run** arrow button at the top. You’ll know your code was successfully loaded when it shows the **>>> %Run** command in the bottom in the **Shell** window. If you need to make changes to your code and want to run it again or run a different script, press the red **Stop** button at top. It’s important to stop running your code before you attempt to run something else so the Pi can reset itself.

Programming Basics

Learning to write code is a little like learning a new language, as each programming language has its own terminology and syntax. Python, the program we will be using, was designed for readability. The projects will walk you through how to learn writing Python step-by-step.

Loading & Saving Programs: For all programming languages, it is good to continously save your programs. When loading each project file in this manual, we recommend you save a new file immediately, in case you want to reference the original instructions or you accidentally delete a line of code.

Commands: All programming languages use commands. A command is a unique word to perform a specfic operation. For example, "print" is a command used to display text on the screen. Try writing **print "Hello World!"** in the Shell of Thonny IDE. Python uses new lines to complete a command. Other languages may require a command to end with a semicolon or parentheses to run.

Blocks: A Python program is constructed from blocks of code. A block is a piece of program text that is executed as a unit. A single command line is a block, as are multiple lines grouped under a function, loop, or conditional statement, all of which will be introduced throughout the projects.

Indents: Indentation refers to the spaces at the beginning of a code line, usually created using the tab key. While in other programming languages, indentation is used for readability only, the indentation in Python is very important. Python relies on indentation to indicate where a block of code starts and ends. You have to use the same number of spaces in the same block of code, otherwise Python will give an error.

Comments: A hashtag symbol (#) is used to create a comment in the program. Comments have no effect on your code but are helpful to explain programs or provide instructions. You can also 'comment out' parts of code if you don't want that part of the program to run but would like to have it for reference. For multi-line comments, you can use """ before and after your comment, to comment out multiple lines of code without starting each line with a #.

Keywords: Keywords are some predefined and specially reserved words that have specific meanings and purposes and can only be used for those specific purposes. Keywords are used to define the syntax of the coding, and will appear in a different color than the rest of the code text to signal their usage.

Examples of keywords include: and, if, for, while, return, import, else, and def. All the keywords in Python are written in lower case except True and False and are ready to be used without importing a library.

Punctuation: Parentheses, brackets, and braces are all utilized in Python but have different use cases and syntax to be aware of when writing your code. Similar to writing a normal sentence, proper punctuation matters in order to express what you mean. Parentheses, like (and), are most commonly used when passing arguments into functions, both custom and predefined. Square brackets, like [and], are used to define lists, arrays, and strings and can also index elements of those storage types to get a specific value at a certain location. Curly braces, like { and }, can separate a block of code and be used with classes but are not necessary for most Python applications.

Debugging Code

If your code isn't working as expected on the first try, don't worry! This is completely normal and to be expected whenever programming complex logic, which is why we have debugging tools to help us identify and fix errors. Next to the green **Run** button in Thonny IDE is **Debug Current Script**, which allows the code to run more slowly and present additional information about its performance. In debugging mode, there's a lot we can do to get back on track.

DEBUG CURRENT SCRIPT

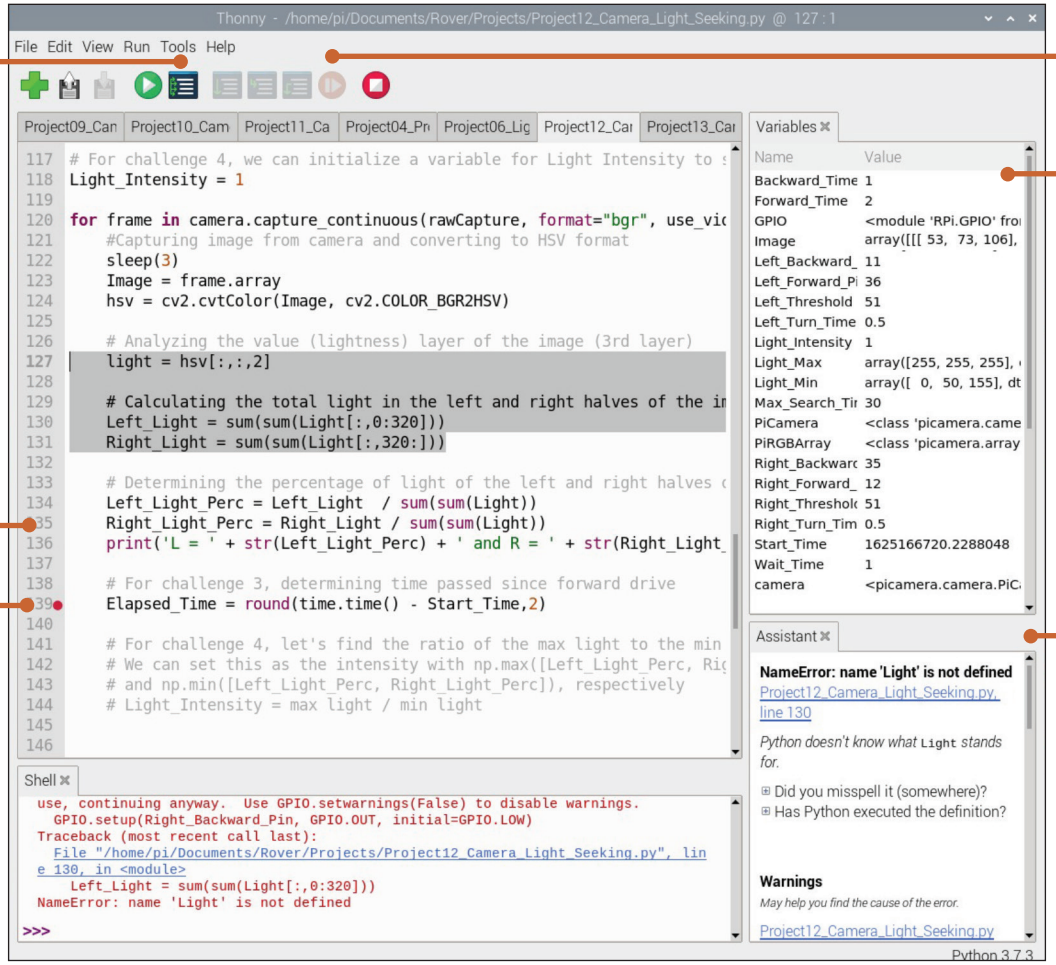
Runs your current code in debugging mode.

PRINT STATEMENT

To see how the code is working while running, use print statements for variables of interest. You can print strings alongside the variables for extra information.

BREAKPOINT

Click to the right of the number of a line of code to add a breakpoint, which appears as a small red circle. Now, when debugging, the code will stop here and allow you to view all variables.



RESUME

After hitting a breakpoint, press Resume to continue running the code.

VARIABLES

To see how your code is performing, looking at the variables is helpful. At a breakpoint, see if they are taking on the expected values and shapes/types. In the top menu bar, click on View, then select Variables.

ASSISTANT

Thonny has a built-in coding assistant that can help spot oddities within the code. Here it's clarifying the error the "Light" is not defined because it's "light" in line 127. It can also show warnings for stylistic or functional issues and always gives a line number. In the top menu bar, click on View, then select Assistant.

CONFORMS TO ALL APPLICABLE U.S. GOVERNMENT REQUIREMENTS

WARNING: Always check your wiring before turning on a circuit. Never leave a circuit unattended while the batteries are installed. Never connect additional batteries or any other power sources to your circuits. Discard any cracked or broken parts.

ADULT SUPERVISION: Because children's abilities vary so much, even with age groups, adults should exercise discretion as to which experiments are suitable and safe (the instructions should enable supervising adults to establish the experiment's suitability for the child). Make sure your child reads and follows all of the relevant instructions and safety procedures and keeps them on hand for reference.

This product is intended for use by adults and children who have attained sufficient maturity to read and follow directions and warnings.

Never modify your parts, as doing so may disable important safety features in them and could put your child at risk of injury. The packaging has to be kept since it contains important information.

BATTERIES:

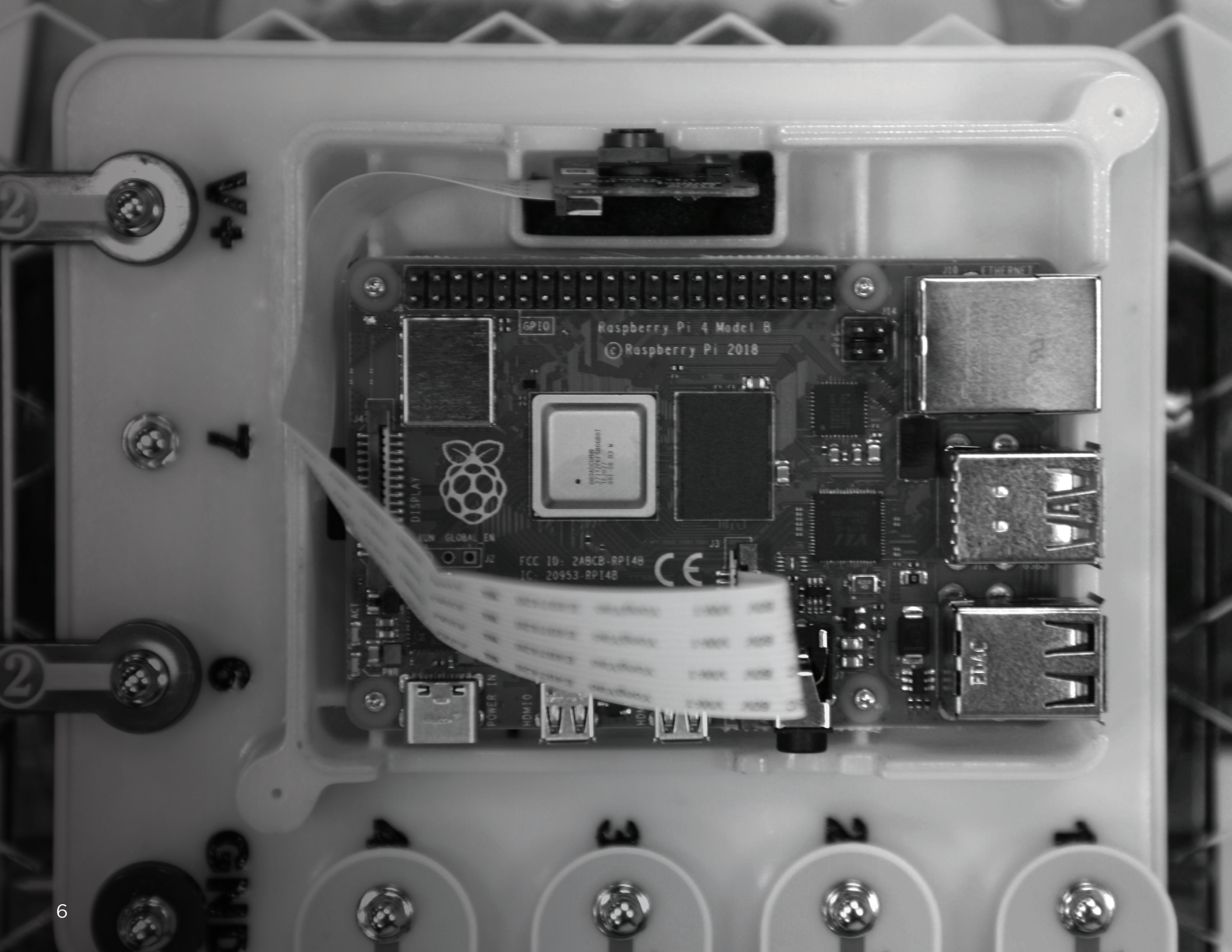
- Use only 1.5V AA type in the rover body.
- Insert batteries with correct polarity.
- Non-rechargeable batteries should not be recharged. Rechargable batteries should only be charged under adult supervision and should not be recharged while in the product.
- Remove batteries when they are used up.
- Do not mix alkaline, standard (carbon-zinc), or rechargeable (nickel–cadmium) batteries.
- Do not mix old and new batteries.
- Do not connect batteries in parallel.
- Do not short circuit the battery terminals.
- Never throw batteries in a fire or attempt to open their outer casing.
- Batteries are harmful if swallowed, so keep away from small children.

WARNING: the Smart Rover should ONLY be powered using AA batteries. The Smart Module should ONLY be powered using the power cord provided. Neither should ever be used with Snap Circuits® battery holders or other power sources!

WARNING: ELECTRIC TOY
The Smart Rover is rated for ages 10 and older, and is not recommended for children under 10 years of age.

WARNING: SHOCK HAZARD
The Smart Module should ONLY be powered using the power cord provided.

Warning to Snap Circuits® Owners
Do not use parts from other Snap Circuits® sets with this kit. The Smart Rover uses higher voltage, which could damage those parts.


GPIO
Raspberry Pi 4 Model B
©Raspberry Pi 2018
DISPLAY
GLOBAL_EN
FCC ID: 2ABCB-RPI4B
IC: 20953-RPI4B
POWER IN
HDMI0
ETHERNET
EDAC