SYMBOLA

PureBasic setup on Raspberry Pi

Table of Contents

Disclaimer

This guide is provided for informational purposes only. Every effort has been made to ensure the accuracy and completeness of this guide, but it is provided “as is” without warranty of any kind, express or implied. The author shall not be held liable for any direct, indirect, incidental, or consequential damages or losses arising from the use of this guide.

The procedures and software described in this guide are subject to change and may not be up-to-date. Users are advised to exercise caution and consider their specific circumstances when following the instructions.

This guide may contain links to external websites. The author is not responsible for the content or accuracy of any external site.

Please use this guide responsibly and at your own risk.

Contact the Author

If you've spotted an error or simply wish to make contact, feel free to leave a message at:

symbola.co.uk/contact

Your feedback and inquiries are always welcome!

Preface

Introduction to PureBasic

PureBasic is a modern programming language that combines the simplicity of BASIC with the efficiency and capabilities of contemporary programming tools. It stands out for its ease of use, making it accessible to beginners, yet it's powerful enough for experienced programmers due to its advanced features like multithreading, Unicode support, and cross-platform compatibility.

Developed to be procedural and highly readable, PureBasic allows for rapid software development across Windows, Linux, and macOS without altering code. Its compiler generates optimized machine code, ensuring applications run fast, which is ideal for everything from utility scripts to complex desktop applications. The language also includes a comprehensive standard library for GUI, graphics, sound, and more, plus it supports inline assembly for performance-critical tasks.

With an active community and extensive documentation, PureBasic lowers the learning curve for newcomers and provides a flexible, efficient environment for developing a wide range of applications.

Introduction to the Raspberry Pi 5 (8GB model)

The Raspberry Pi is a compact, affordable computer that has transformed digital learning and hobbyist projects worldwide. Originally designed to teach computer science basics, it has become a favorite for a wide range of users from educators and students to DIY enthusiasts and professional developers. Its small size, low cost, and considerable computing power make it ideal for various applications, including educational projects, home automation, and even commercial product development.

Featuring an ARM-based CPU, RAM, USB ports, GPIO pins for hardware interfacing, HDMI output, and network connectivity, all on a board roughly the size of a credit card, the Raspberry Pi stands out for its versatility. Its GPIO pins, in particular, enable connections to external devices and sensors, making it perfect for robotics and Internet of Things (IoT) applications.

Raspberry Pi OS, the official operating system, offers a user-friendly desktop experience with preloaded software for programming and multimedia, making it accessible to beginners. With numerous versions improving on processing power and connectivity, the Raspberry Pi continues to support a mission of democratizing access to computer science education, providing a platform for users to explore computing and realize creative projects.

What to Expect from This Guide

This guide covers the essentials of setting up your Raspberry Pi, from hardware assembly to software installation, and introduces you to programming with PureBasic. Expect straightforward steps to get your Raspberry Pi up and running, followed by a simple setup of PureBasic for your programming projects.

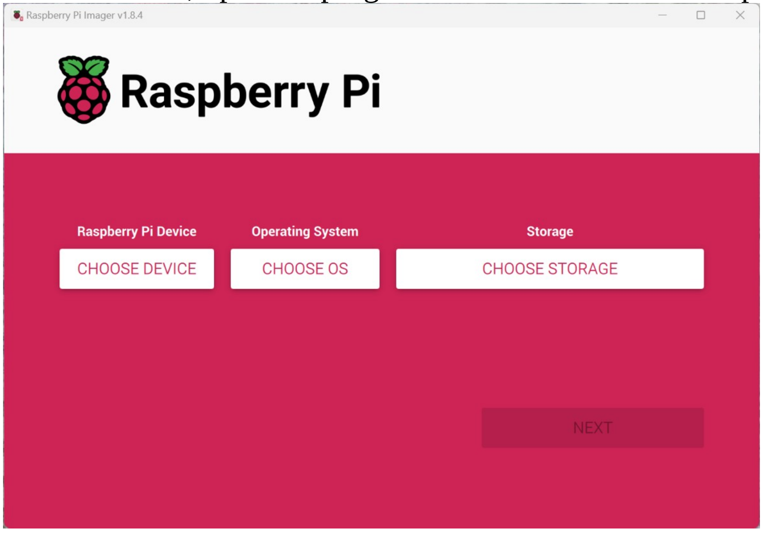
Note: This guide specifically utilizes a Raspberry Pi 5 (8GB) model with Raspberry Pi OS 64-bit, ensuring a smooth and optimized experience for both setup and programming with PureBasic.

Hardware Requirements for Setting Up Your Raspberry Pi

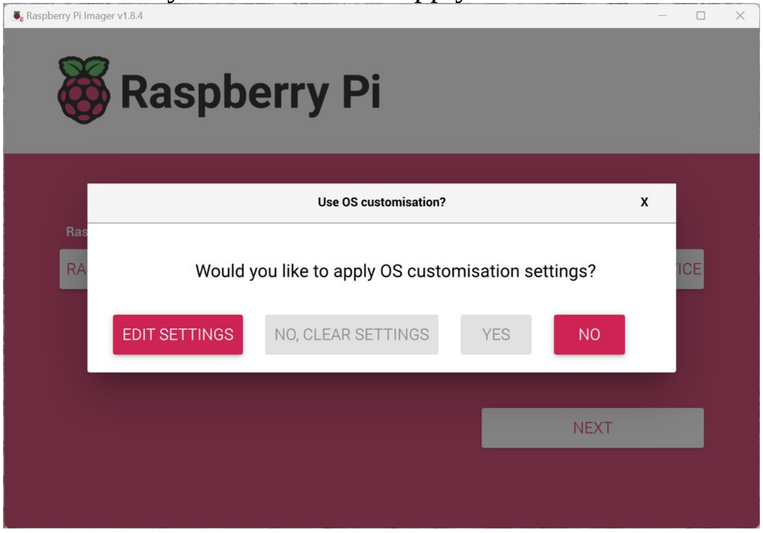
1. **Raspberry Pi (Raspberry Pi 5 8GB used in this guide)**
   This guide uses the Raspberry Pi 5 (8GB) for its optimal performance and features. However, older models such as the Raspberry Pi 4 or 3 are also compatible but may result in slower performance. The Raspberry Pi 5 with 8GB RAM is recommended for the best experience, though models with 4GB or 2GB of RAM can also be used, with an understanding that performance may be impacted.
2. **Power Supply**
   An appropriate and reliable power supply for the Raspberry Pi 5 is crucial for stable operation. The official power supply for this model is highly recommended. Make sure to purchase the correct version for your country!
3. **MicroSD Card**
   At least 16GB recommended, with a preference for 32GB or more for better storage and performance. Use a class 10 microSD card for improved read/write speed.
4. **Keyboard and Mouse**
   Essential for the initial setup and configuration.
5. **Monitor/Display**
   A monitor with an HDMI interface is required for the Raspberry Pi.
6. **Micro-HDMI to HDMI Cable**
   This cable is needed to connect the Raspberry Pi to the display.
7. **Ethernet Cable or Wi-Fi Connectivity**
   Internet access is crucial for downloading necessary software and updates.
8. **Cooling Fan**
   To ensure the best performance, a cooling fan is essential for this model.
9. **Case for Raspberry Pi (Optional)**
   A case is recommended to protect the hardware.
10. **MicroSD Card Reader**
    Either a built-in or an external USB reader will suffice.
11. **Access to Another Computer**
    This is necessary to install the operating system files onto the MicroSD card. This can be a Windows or Mac computer.

Writing the Operating System to the MicroSD card

1. **Install Raspberry Pi Imager on Your Computer**
   Download it from: raspberrypi.com/software.
2. **Insert the MicroSD Card into the Reader**
   Place your MicroSD card into the card reader, then connect it to your computer.
3. **Install and Run the Raspberry Pi Imager**
   Once installed, open the program. You should see the Raspberry Pi Imager interface.



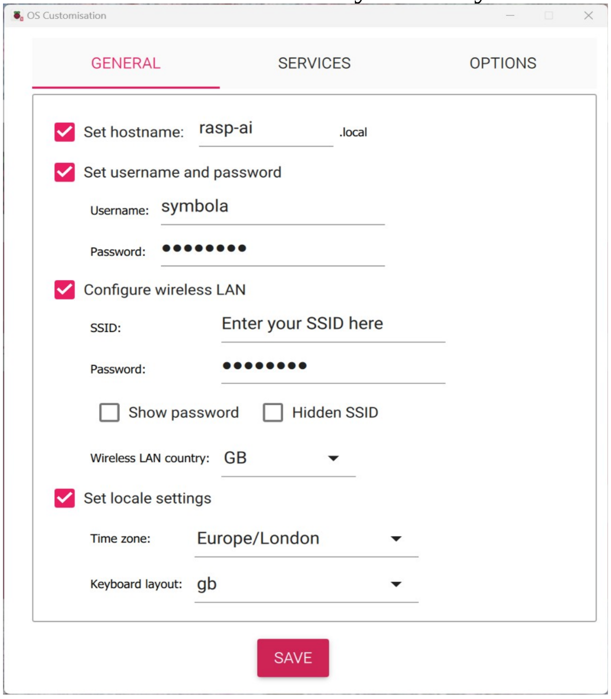
4. **Choose Raspberry Pi Device**
   Select "Raspberry Pi 5."
5. **Choose Operating System**
   Select "Raspberry Pi OS (64-BIT)."
6. **Choose Storage (The MicroSD Card to Write To)**
   Note: All data on the MicroSD card will be deleted!
   Choose your MicroSD card from the list.
7. **Click "Next"**
8. **A Pop-Up Message Box Will Appear**
   It will ask if you would like to apply OS customization settings.



Would you like to apply OS customisation settings?

9. **Select "Edit Settings"**

10. **Set Hostname**
    Enter "rasp-ai" as the hostname.
11. **Set Username and Password**
    Don’t forget these credentials!
12. **Configure Wireless LAN**
    If you’re unsure of your Wi-Fi network's SSID, check on your phone, laptop, or Wi-Fi router.
13. **Set Locale Settings**
    Select the time zone and keyboard layout that suits your setup.



☑ Set hostname: rasp-ai .local

☑ Set username and password

Username: symbola

Password: ••••••••

☑ Configure wireless LAN

SSID: Enter your SSID here

Password: ••••••••

☐ Show password ☐ Hidden SSID

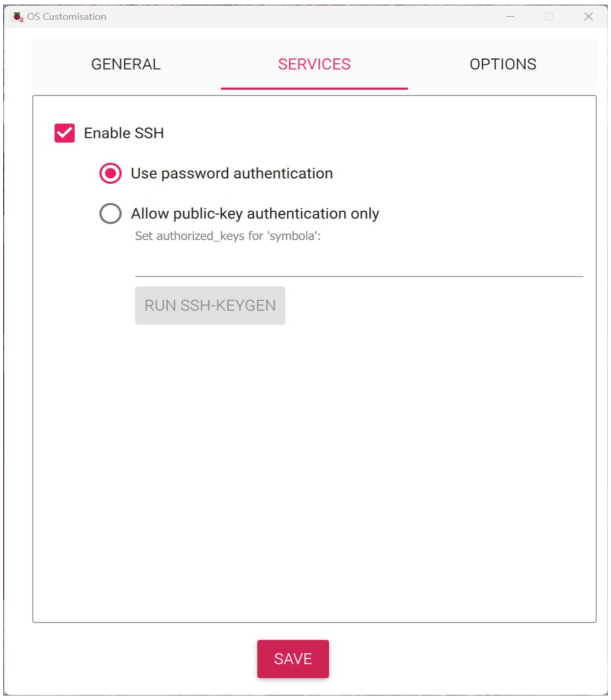
Wireless LAN country: GB

☑ Set locale settings

Time zone: Europe/London

Keyboard layout: gb

14. **Enable SSH in the 'Services' Tab**



☑ Enable SSH

● Use password authentication

○ Allow public-key authentication only

Set authorized_keys for 'symbola':

15. **Leave Options in the 'Options' Tab As Is**
16. **Click on “Save”**

17. **Back to the Pop-Up Message Box**
    When asked again about applying OS customization settings, select "YES".
18. **Read the Notice About Data Erasure**
    Select "YES" (only if you are okay with all data being erased on the MicroSD card).
19. **Wait for the Operating System to be Written to the MicroSD Card**

Setting up the hardware

1. **Insert the Prepared MicroSD Card into Your Raspberry Pi**
2. **Fit the Cooling Fan** (if purchased)
   Before powering up your Raspberry Pi, attach the cooling fan. This step is crucial to maintain optimal operating temperatures, especially important when running demanding applications. Follow the instructions provided with your cooling fan for proper installation.
3. **Place the Raspberry Pi in a Case** (if purchased)
   If you have a case for your Raspberry Pi, now is the time to secure the board inside it. A case helps protect the Raspberry Pi from physical damage and dust. Make sure that the case is compatible with the cooling fan setup and allows proper ventilation.
4. **Connect to a Display**
   Use the Micro-HDMI to HDMI cable to connect your Raspberry Pi to the monitor or display.
5. **Attach Keyboard and Mouse**
   Connect these peripherals to the USB ports on the Raspberry Pi.
6. **Ethernet Connection (Optional)**
   If you are not using Wi-Fi, connect your Raspberry Pi to the network using an Ethernet cable. This will provide a more stable internet connection, which is crucial for downloading updates and software.
7. **Connect to Power Supply**
   Plug in the power supply to your Raspberry Pi. Ensure you are using the recommended power adapter to prevent any power issues.
8. **Ensure Secure Connections**
   Before powering up, double-check that all connections (display, keyboard, mouse, Ethernet, and power supply) are secure. A loose connection could lead to problems during the setup process.
9. **Power On the Raspberry Pi**
   Once everything is connected and secured, power on your Raspberry Pi. Proceed with the initial setup of the Raspberry Pi OS.

First Boot

Upon the first boot, after the desktop screen loads, a pop-up window titled “Wi-Fi Network Authentication Required” may appear. If it does, select “Connect” to establish an internet connection.

If all has gone well, your Raspberry Pi should boot up to the desktop and be connected to the internet. At the top left of the screen, from left to right, you will see several icons:

1. **The Raspberry Pi Logo**
   This is the operating system's main "Start" menu, where you can access programs, settings, etc.
2. **The Globe Icon**
   This represents the Chromium web browser by default.
3. **Two Yellow Folders**
   These are icons for the file explorer, where you can manage and access your files.
4. **The Dark Box with “>_”**
   This is the terminal application, a powerful tool for running various commands and scripts.

On the right side at the top, you'll find icons for Bluetooth, Wi-Fi, volume control, and the time display. These are quick access buttons that allow you to manage your device's connectivity and settings easily.

Open a Terminal Window

Much of the installation process will be conducted using the terminal. Here's how to open it:

1. **Navigate to the Terminal**
   Look at the top left of the screen and find the icon that looks like a dark box with “>_” symbol. This is the terminal application.
2. **Open the Terminal**
   Click on the “>_” icon. A terminal window will open, displaying a command prompt, which is where you can enter your commands.
3. **Using the Terminal**
   Once the terminal window is open, you will see a command prompt waiting for commands to be entered.

Tip: If you are viewing this guide as a PDF document on your Raspberry Pi, you can conveniently copy and paste the commands directly into the terminal. This can save time and reduce the chance of typing errors.

Download PureBasic

For the demo version, visit:

- https://www.purebasic.com/download.php
- Select "**PureBasic free for Raspberry Pi OS (arm64 – 64-bit)**" under the Raspberry Pi OS section.
- This will download a file named "**purebasic-demo_arm64.tgz**".

For the full version (**requires purchase**), visit:

- https://www.purebasic.com/securedownload/administration.php
- Under "**Download LTS**" choose "**Download PureBasic 6.04 LTS for Raspberry Pi OS (arm64)**".
- This will download a file named "**PureBasic_Linux_ARM64_LTS_6.04_(bk7njn).tgz**".

Note: The filenames mentioned above will change over time. Please adjust accordingly!

Unpack the downloaded files

- Create a folder named "**Programming**" in the user's home directory. For example: **/home/symbola/Programming** (replace "symbola" with your username).
- Use the terminal command **mkdir -p ~/Programming** or utilize the desktop GUI file manager to create this folder.

Extract the "**.tgz**" files into the folder created above. The file structure should look like this:

- /home/symbola/Programming/purebasicHello
  or
- /home/symbola/Programming/purebasic_demo

Use the following terminal commands to extract the files, ensuring to adjust the file paths as needed:

- **tar -xzvf ~/Downloads/purebasic-demo_arm64.tgz -C ~/Programming/**
  or
- **tar -xzvf ~/Downloads/PureBasic_Linux_ARM64_LTS_6.04_\(bk7njn\).tgz -C ~/Programming/**

Note: Pay attention to the backslashes before the parentheses in the second command!

You can perform the extraction using the terminal commands provided above or through the desktop GUI file manager.

Check for system updates

Update the System. All these commands need to be typed (or copied and pasted) into an open Terminal window. (Commands shown in bold and green).

1. Update package list
   - **sudo apt update**
2. Upgrade installed packages
   - **sudo apt full-upgrade**

Install Developer Packages

Install the necessary developer packages: -

1. Update compiler and default libraries
   - **sudo apt-get install build-essential gcc g++**
2. Install essential development libraries
   - **sudo apt-get install libxxf86vm-dev libxine2-dev unixodbc-dev libsdl1.2-dev libsdl2-dev libssl-dev**
   - **sudo apt-get install libgtk2.0-dev libgtk-3-dev**
3. For WebKit (for web content in applications), use
   - **sudo apt-get install libwebkit2gtk-4.0-dev**
4. For movie library support (e.g., for video playback in applications), install VLC development files
   - **sudo apt-get install libvlc-dev**
5. For game development libraries
   - **sudo apt-get install libgl1-mesa-dev libgl1-mesa-glx**
6. Subsystem qt (for Qt-based applications)
   - **sudo apt-get install qtbase5-dev qttools5-dev qtmultimedia5-dev qtdeclarative5-dev libqt5svg5-dev libqt5webkit5-dev libqt5multimedia5-plugins (write as one line, keep spaces between packages)**
7. Install Missing Default Libraries for WebKitGTK
   - **sudo apt-get install libwebkit2gtk-4.0**

Create a PureBasic desktop icon

Open a text editor and paste the following, adjusting paths as necessary:

```
[Desktop Entry]
Type=Application
Name=PureBasic
Comment=PureBasic Integrated Development Environment
Icon=/home/symbola/Programming/purebasic/logo.png
Exec=/home/symbola/Programming/purebasic/compilers/purebasic
Path=/home/symbola/Programming/purebasic/compilers
Terminal=false
StartupNotify=false
```

Replace “**/home/symbola/Programming/purebasic**_demo” with the actual path to your PureBasic installation

Save this text as “**purebasic.desktop**” (or another name, but keep the “**.desktop**” extension) in your desktop folder (“**~/Desktop/**”).

You should now see a PureBasic icon on your desktop. When launching it from the desktop, if prompted with options such as execute, execute in terminal, open, or cancel, select "**execute**".

Adding PureBasic to the main program menu

Open a text editor and paste the following, adjusting paths as necessary:

```
[Desktop Entry]
Type=Application
Name=PureBasic
GenericName=PureBasic IDE
Comment=PureBasic Integrated Development Environment
Icon=/home/symbola/Programming/purebasic/logo.png
Exec=/home/symbola/Programming/purebasic/compilers/purebasic
Path=/home/symbola/Programming/purebasic/compilers
Terminal=false
Categories=Development;IDE;
```

Replace “**/home/symbola/Programming/purebasic**_demo” with the actual path to your PureBasic installation.

Save this text as “**purebasic.desktop**” (or another name, but keep the “**.desktop**” extension) in your applications folder (“**~/.local/share/applications/**”).

To refresh the menu, use the following command: “**lxpanelctl restart**”.

In the Raspberry Pi OS main menu, you should now be able to launch PureBasic from the Programming submenu.

Example programs

Example programs to try can be found at:

- /home/symbola/Programming/purebasic/examples

Replace “**/home/symbola/Programming/purebasic**” with the actual path to your PureBasic installation.

Conclusion

In conclusion, with your Raspberry Pi set up and PureBasic installed, you're well-equipped to dive into the world of programming. PureBasic offers a blend of simplicity and power, making it an ideal tool for both beginners and seasoned programmers. Here's a short guide to get started with using PureBasic:

1. Explore the IDE: Familiarize yourself with the PureBasic Integrated Development Environment (IDE). Check out the menus, settings, and the layout to understand where everything is.
2. Run Example Programs: Navigate to the example programs directory (/home/symbola/Programming/purebasic/examples or your respective path) to explore and run existing PureBasic code. This is a great way to see PureBasic in action and learn from real examples.
3. Write Your First Program: Start with something simple, like a "Hello, World!" program. Open the PureBasic IDE, type in the code, and use the compile/run button to execute your program. This step will give you a sense of accomplishment and a taste of what PureBasic can do.
4. Learn PureBasic Syntax: PureBasic's syntax is straightforward, but powerful. Spend some time learning the basics of variables, control structures (if statements, loops), and functions. The PureBasic documentation and online forums are great resources for learning.
5. Experiment with GUI Applications: PureBasic makes it easy to create applications with graphical user interfaces (GUIs). Try out the GUI commands to create windows, buttons, and other interface elements. Check the documentation for OpenWindow, ButtonGadget, and other related commands.
6. Dive Into More Complex Projects: As you become more comfortable with PureBasic, challenge yourself with more complex projects. Try creating a simple game, a utility tool, or anything else that interests you. The sky is the limit!
7. Join the Community: The PureBasic community is an invaluable resource. Join forums, participate in discussions, and don't hesitate to ask for help or share your projects.

By following this guide and experimenting on your own, you'll gain proficiency in PureBasic and unlock the potential to create a wide array of applications and projects. Happy coding, and enjoy the journey of exploration and creation with PureBasic on your Raspberry Pi.