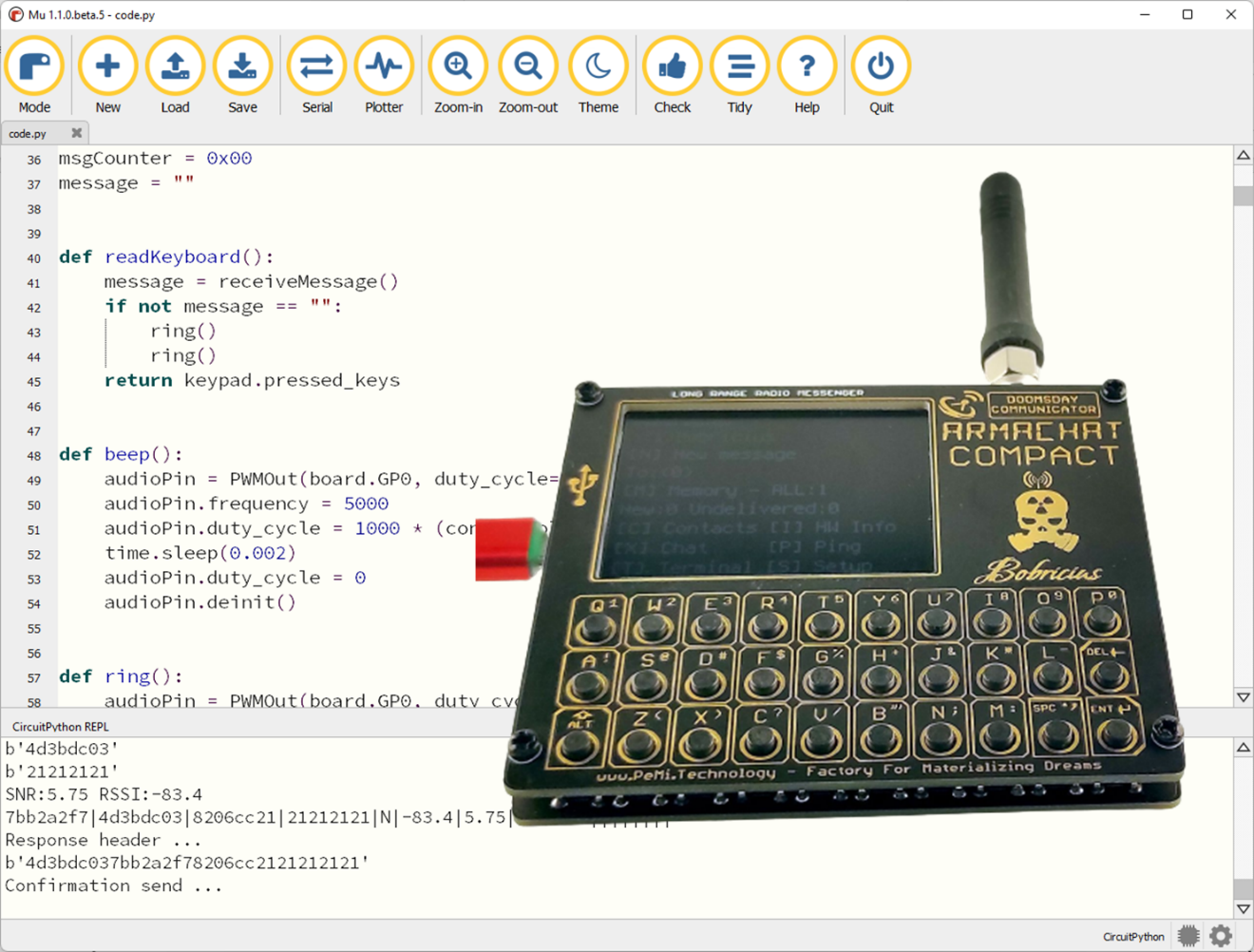
ARMACHAT Long Range Radio Messenger Pico Version - Developer Guide



Contributions By:

* Peter Misenko / GitHub: bobricius
* Kayto / GitHub: Kayto
* Richard Teel / GitHub: richteel

Contents

[Introduction 3](#_Toc97839479)

[Installing CircuitPython 3](#_Toc97839480)

[Prerequisites 3](#_Toc97839481)

[Steps 3](#_Toc97839482)

[Python Editor 3](#_Toc97839483)

[Obtaining and loading the ARMACHAT software and documentation 4](#_Toc97839484)

[Steps to install the code to your Raspberry Pi Pico 4](#_Toc97839485)

[boot.py 5](#_Toc97839486)

[Options provided in the boot.py file 6](#_Toc97839487)

# Introduction

The Pico version of ARMACHAT utilizes a Raspberry Pi Pico running CircuitPython.

## Installing CircuitPython

Adafruit has a great article on installing CircuitPython at <https://learn.adafruit.com/welcome-to-circuitpython/installing-circuitpython>. This section will not duplicate the documentation on the site but will provide a checklist for the steps needed to install CircuitPython on the Raspberry Pi Pico. Please see the Adafruit site for the latest information.

### Prerequisites

* A PC running Windows, MacOS, or Linux and connection to the internet or the CircuitPython installer for the Raspberry Pi Pico
* A Raspberry Pi Pico
* USB C Cable to connect the Raspberry Pi Pico to the PC

### Steps

1. Download CircuitPython for the Raspberry Pi Pico from <https://circuitpython.org/downloads>.
2. If you are running Windows 7 or 8.1, you may need to install drivers to use a CircuitPython board. (See the Adafruit tutorial if you are running an older version of Windows.)
3. Press and hold the BOOTSEL button on the Raspberry Pi Pico and plug the USB cable into the Raspberry Pi Pico and the PC.
4. Release the BOOTSEL button when the RPI-RP2 drive appears on the PC.
5. Drag/Copy the CircuitPython file you downloaded in the first step and copy it to the RPI-RP2 drive. Once the file is copied, the Raspberry Pi Pico will reboot and a CIRCUITPY drive will appear on the PC.

## Python Editor

There are several editors that you may use for CircuitPython. One of the ones recommended in the Adafruit article is Mu Editor. It may be downloaded from <https://codewith.mu/>.

The Mu editor provides several features that make it easier to code and debug CircuitPython. It includes a Check feature to make certain that coding standards are being used. Some of the things that are checked are:

* Unused libraries
* Code indentation
* Comment formatting (Space after hash mark and two spaces before inline comments.)
* Unused variables (Helpful when forgetting to use the Global keyword.)
* Omitting colon, “:”, at the end of if, elif, else, for, etc.
* General syntax errors

The Mu editor also contains a Tidy feature, which will do some code clean up for you such as switching single quotes used for string to double quotes. In order to use the Tidy feature, your code will need to be able to pass the Check feature.

The Serial feature brings up a serial console at the bottom of the editor for viewing the output of print statements or interacting with the microcontroller by running code.

# Obtaining and loading the ARMACHAT software and documentation

You will need to go to GitHub to get the latest code for the ARMACHAT. The official repository is owned by Peter, aka Bobricus, and is located at <https://github.com/bobricius/Armachat-circuitpython>.

There are other repositories, which are folks of Peter’s code. This documentation is being created and maintained by Kayto at <https://github.com/Kayto/Armachat-Compact-Documentation>, but may be pulled into the official repository later.

This documentation may be outdated but will contain content that will still be relevant to help understand the code structure of the code into the future. Currently the CircuitPython code is going through several revisions and is quite active. It is expect that once the code is mature and provides all of the expected functionality and is stable, the rate of code changes will diminish.

## Steps to install the code to your Raspberry Pi Pico

1. Download the code from the official repository at <https://github.com/bobricius/Armachat-circuitpython> by clicking the green code button and either downloading a zip file or copying the clone URL and cloning to your PC using git.  
   Graphical user interface, text, application, email

   Description automatically generated
2. Connect the Raspberry Pi Pico with CircuitPython installed to your PC and drag the following folders and files from the Software folder to the Pico:
   1. config
   2. fonts
   3. lib
   4. boot.py
   5. code.py

The folders and files on your Raspberry Pi Pico should look like the following.  
Graphical user interface, text, application

Description automatically generated

# boot.py

The boot.py file runs when the Raspberry Pi Pico is first powered on, or the reset switch is pressed if one is available. The boot.py file does several things on load. The flowchart below shows the program flow of the ARMACHAT boot.py file.

Diagram

Description automatically generated

If you edit code, and the Raspberry Pi Pico stops being recognized by the PC, you may need to reset the flash and reinstall CircuitPython. To reset the flash on your Pico, see the instructions in the “Resetting Flash Memory” at <https://www.raspberrypi.com/documentation/microcontrollers/raspberry-pi-pico.html>.

## Options provided in the boot.py file

When the Raspberry Pi Pico boots by initial power on or pressing the reset button if installed on the ARMACHAT, there are two options available.

* Start the Raspberry Pi Pico in safe mode the next time the Pico is rebooted.
* Mount the Raspberry Pi Pico so that the code running on the Pico can write to the filesystem. When this option is enabled, the PC will not be able to write to the Pico to update code. The file storage on the Pico can only allow one device to write to the storage. Either the PC may write to the file storage, or the device can write to the file storage but not at the same time.

# config/config.py

The config.py file contains information for your ARMACHAT and should be edited to allow the software to know which model of ARMACHAT you have, set the correct frequency for your LoRa module, default settings, and other options to make your ARMACHAT unique.