

RPi and MicroPython 64-Bit

NORMAN MCENTIRE

References

- <https://micropython.org/>

Example Project: MicroPython

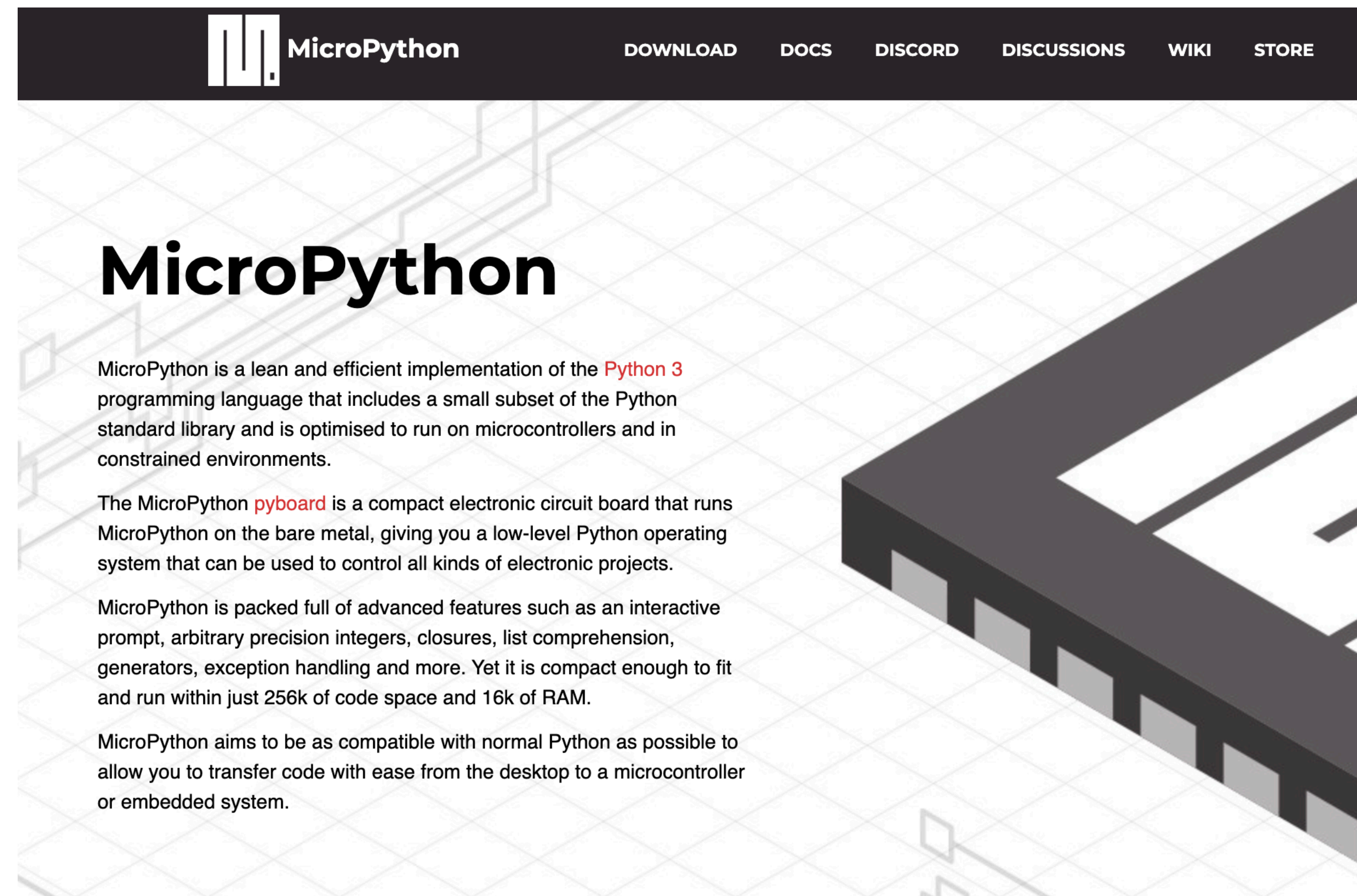
- Suppose your task is to build MicroPython
- You want to build MicroPython for both your 64-bit Host
- And for your 32-bit Target

About MicroPython

- MicroPython is Python3 for Embedded Systems
- Compatible with Python3 but optimized for embedded systems
 - Does not include ALL the Python3 standard library...but a lot of it!
- This will be covered in another slide deck/video

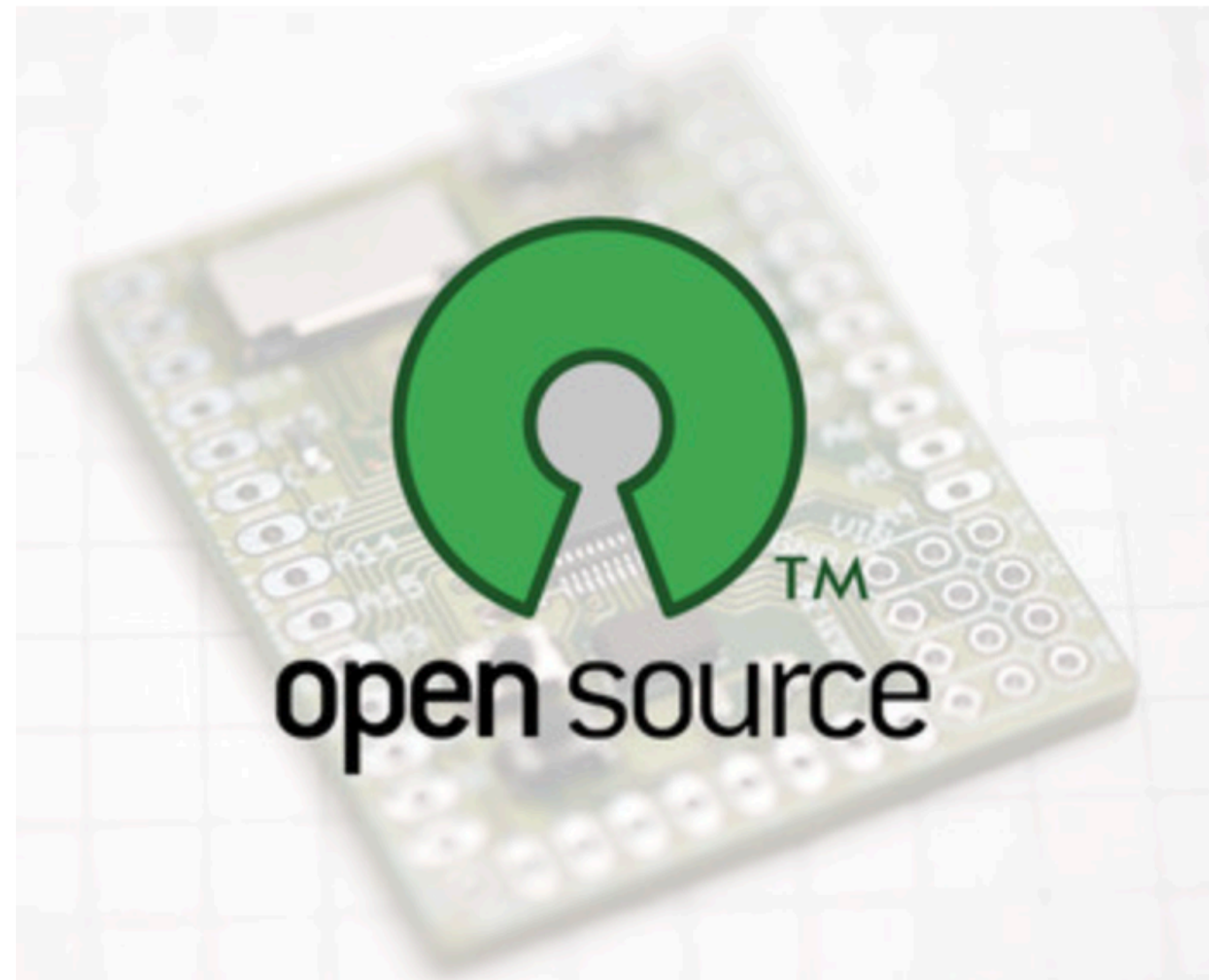
<https://micropython.org/>

Compact enough to fit within
256K of code space
And
16K of RAM



Open Source

Completely free, open source software



MicroPython is written in C99 and the entire MicroPython core is available for general use under the very liberal **MIT license**. Most libraries and extension modules (some of which are from a third party) are also available under MIT or similar licenses.

You can freely use and adapt MicroPython for personal use, in education, and in commercial products.

MicroPython is developed in the open on GitHub and the source code is available at **the GitHub page**, and on the **download page**. Everyone is welcome to contribute to the project.

Downloads

MicroPython downloads

MicroPython is developed using git for source code management, and the master repository can be found on GitHub at github.com/micropython/micropython.

The full source-code distribution of the latest version is available for download here:

- [micropython-1.21.0.tar.xz](#) (78MiB)
- [micropython-1.21.0.zip](#) (165MiB)

Downloads

<https://github.com/micropython/micropython>

MicroPython downloads

MicroPython is developed using git for source code management, and the master repository can be found on GitHub at github.com/micropython/micropython.

The full source-code distribution of the latest version is available for download here:

- [micropython-1.21.0.tar.xz](#) (78MiB)
- [micropython-1.21.0.zip](#) (165MiB)

Cloning the code from GitHub

<https://github.com/micropython/micropython>

```
$ git clone --depth=1 https://github.com/micropython/micropython
Cloning into 'micropython'...
remote: Enumerating objects: 5612, done.
remote: Counting objects: 100% (5612/5612), done.
remote: Compressing objects: 100% (4561/4561), done.
remote: Total 5612 (delta 1340), reused 3114 (delta 726), pack-reused 0
Receiving objects: 100% (5612/5612), 7.97 MiB | 6.08 MiB/s, done.
Resolving deltas: 100% (1340/1340), done.
```

```
$ cd micropython/
```

```
$ ls
```

ACKNOWLEDGEMENTS	CONTRIBUTING.md	examples	LICENSE	ports	README.md	tools
CODECONVENTIONS.md	docs	extmod	logo	py	shared	
CODEOFCONDUCT.md	drivers	lib	mpy-cross	pyproject.toml	tests	

Building MicroPython for Linux

```
$ cd ports
```

```
$ ls
```

```
bare-arm  embed  esp8266  minimal  pic16bit  qemu-arm  rp2  stm32  unix  windows  
cc3200    esp32  mimaxrt  nrf       powerpc   renesas-ra  samd  teensy  webassembly  zephyr
```

```
$ cd unix
```

```
$ ls
```

```
alloc.c          input.h          modmachine.c    mpbtstackport_common.c  mphalport.h      README.md  
coverage.c       main.c           modos.c         mpbtstackport.h         mpnimbleport.c   unix_mphal.c  
coveragecpp.cpp  Makefile        modsocket.c     mpbtstackport_h4.c      mpnimbleport.h   variants  
fatfs_port.c     mbedtls         modtermios.c    mpbtstackport_usb.c     mpthreadport.c  
gccollect.c      modffi.c        modtime.c       mpconfigport.h          mpthreadport.h  
input.c          modjni.c        mpbthciport.c   mpconfigport.mk        qstrdefsport.h
```

```
$ make submodules
```

```
. . .
```

```
$ make
```

```
. . .
```

make submodules

```
$ make submodules
```

```
Use make V=1 or set BUILD_VERBOSE in your environment to increase build verbosity.
```

```
Package libffi was not found in the pkg-config search path.
```

```
Perhaps you should add the directory containing `libffi.pc'  
to the PKG_CONFIG_PATH environment variable
```

```
Package 'libffi', required by 'virtual:world', not found
```

```
Package libffi was not found in the pkg-config search path.
```

```
Perhaps you should add the directory containing `libffi.pc'  
to the PKG_CONFIG_PATH environment variable
```

```
Package 'libffi', required by 'virtual:world', not found
```

```
Updating submodules: lib/mbedtls lib/berkeley-db-1.xx lib/micropython-lib
```

```
Submodule 'lib/berkeley-db-1.xx' (https://github.com/pfalcon/berkeley-db-1.xx) registered for path '../..lib/  
berkeley-db-1.xx'
```

```
Submodule 'lib/mbedtls' (https://github.com/ARMmbed/mbedtls.git) registered for path '../..lib/mbedtls'
```

```
Submodule 'lib/micropython-lib' (https://github.com/micropython/micropython-lib.git)
```

```
registered for path '../..lib/micropython-lib'
```

```
Cloning into '/home/nmcentire/micropython/lib/berkeley-db-1.xx'...
```

```
Cloning into '/home/nmcentire/micropython/lib/mbedtls'...
```

```
Cloning into '/home/nmcentire/micropython/lib/micropython-lib'...
```

```
Submodule path '../..lib/berkeley-db-1.xx': checked out '35aaec4418ad78628a3b935885dd189d41ce779b'
```

```
Submodule path '../..lib/mbedtls': checked out '981743de6fcdbe672e482b6fd724d31d0a0d2476'
```

```
Submodule path '../..lib/micropython-lib': checked out 'e025c843b60e93689f0f991d753010bb5bd6a722'
```

apt-cache search libffi

apt-cache search libffi

libffiindex0 - library for simple index/database for huge amounts of small files
libffiindex0-dev - library for simple index/database for huge amounts of small files (development)
libgirepository-1.0-1 - Library for handling GObject introspection data (runtime library)
libghc-libffi-dev - A binding to libffi
libghc-libffi-doc - A binding to libffi; documentation
libghc-libffi-prof - A binding to libffi; profiling libraries
libjffi-java - Java Foreign Function Interface
libjffi-jni - Java Foreign Function Interface (JNI library)
libffi-dev - Foreign Function Interface library (development files)
libffi8 - Foreign Function Interface library runtime
libffi-c-perl - C data types for FFI
libffi-checklib-perl - module to check availability of a library for FFI
libffi-platypus-perl - module to create Perl bindings to non-Perl libraries with FFI
libffi-platypus-type-enum-perl - custom platypus type for dealing with C enumerated types

apt show libffi-dev

```
$ apt show libffi-dev
Package: libffi-dev
Version: 3.4.4-1
Priority: optional
Section: libdevel
Source: libffi
Maintainer: Debian GCC Maintainers <debian-gcc@lists.debian.org>
Installed-Size: 293 kB
Depends: libffi8 (= 3.4.4-1)
Conflicts: libffi4-dev
Homepage: https://sourceware.org/libffi/
Tag: devel::library, role::devel-lib
Download-Size: 56.0 kB
APT-Sources: http://deb.debian.org/debian bookworm/main arm64 Packages
Description: Foreign Function Interface library (development files)
 This package contains the headers and static library files necessary for
 building programs which use libffi.
▪
A foreign function interface is the popular name for the interface that
allows code written in one language to call code written in another
language.
```

sudo apt install libffi-dev

```
sudo apt install libffi-dev
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  libffi-dev
. . .
```

dpkg -L libffi-dev

```
dpkg -L libffi-dev
```

```
/usr
/usr/include
/usr/include/aarch64-linux-gnu
/usr/include/aarch64-linux-gnu/ffi.h
/usr/include/aarch64-linux-gnu/ffitarget.h
/usr/lib
/usr/lib/aarch64-linux-gnu
/usr/lib/aarch64-linux-gnu/libffi.a
/usr/lib/aarch64-linux-gnu/libffi_pic.a
/usr/lib/aarch64-linux-gnu/pkgconfig
/usr/lib/aarch64-linux-gnu/pkgconfig/libffi.pc
/usr/share
. . .
```

NOTE! These include
Files are for aarch64-linux-gnu

Also install the armhf versions

```
$ sudo apt install libffi-dev:armhf
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libffi8:armhf
The following NEW packages will be installed:
  libffi-dev:armhf libffi8:armhf
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.
Need to get 74.5 kB of archives.
After this operation, 353 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

dpkg -L libffi-dev

Notice the arm-linux-gnueabi
Versions of header files

```
$ dpkg -L libffi-dev:armhf
/.
/usr
/usr/include
/usr/include/arm-linux-gnueabi
/usr/include/arm-linux-gnueabi/ffi.h
/usr/include/arm-linux-gnueabi/ffitarget.h
/usr/lib
/usr/lib/arm-linux-gnueabi
/usr/lib/arm-linux-gnueabi/libffi.a
/usr/lib/arm-linux-gnueabi/libffi_pic.a
/usr/lib/arm-linux-gnueabi/pkgconfig
/usr/lib/arm-linux-gnueabi/pkgconfig/libffi.pc
. . .
```

make

```
$ make
```

```
Use make V=1 or set BUILD_VERBOSE in your environment to increase build verbosity.
```

```
GEN build-standard/genhdr/qstr.i.last
```

```
GEN build-standard/genhdr/qstr.split
```

```
GEN build-standard/genhdr/qstrdefs.collected.h
```

```
QSTR updated
```

```
GEN build-standard/genhdr/qstrdefs.generated.h
```

```
GEN build-standard/genhdr/moduledefs.split
```

```
GEN build-standard/genhdr/moduledefs.collected
```

```
Module registrations updated
```

```
. . .
```

```
CC modsocket.c
```

```
CC modffi.c
```

```
CC modjni.c
```

```
CC ../../shared/runtime/gchelper_generic.c
```

```
CC ../../shared/timeutils/timeutils.c
```

```
CC ../../shared/readline/readline.c
```

```
LINK build-standard/micropython
```

text	data	bss	dec	hex	filename
685353	59144	6976	751473	b7771	build-standard/micropython

Running 64-bit micro python

```
$ ./build-standard/micropython  
MicroPython 91a3f18 on 2023-10-21; linux [GCC 12.2.0] version  
Use Ctrl-D to exit, Ctrl-E for paste mode  
>>> print("hello micropython")  
hello micropython  
>>>
```