BREADPI: Multipurpose Hat for Raspberry Pi

User Manual

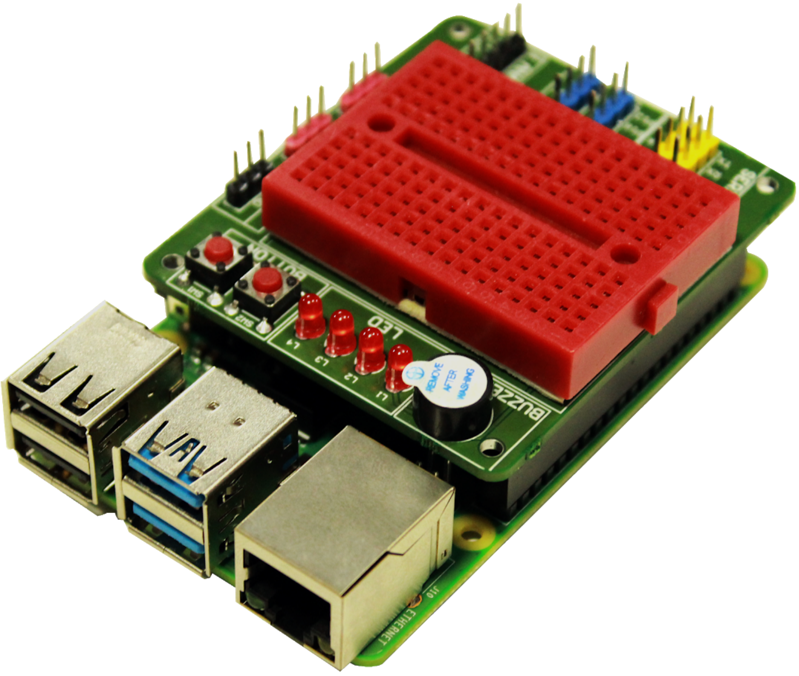


Table Of Content

Summary

BreadPi is a Raspberry Pi hat to provide added features to the Raspberry Pi. This shield provides digital and analog I/O functionalities with a breadboard to add hardware connections for testing, learning, and training purposes.

Features and Benefits

- 8-bit Analog Read/Write,
- 5 and 3v3 tolerant GPIOs,
- Breadboard for temporary prototyping,
- Raspberry Pi Zero/ Zero W/ Zero WH /2B /3B /3B+ /4B Compatible
- Easy GPIO pin access,
- Buzzer, LED, and Buttons for I/O operations,
- Easy access to power pins,
- Dimensions: 65mm × 56mm

Applications

- Analog to Digital Conversion,
- Digital to Analog Conversion,
- IoT,
- Training and Development

Hardware

BreadPi is an easily stackable Raspberry Pi Hat with dimensions of around 65mm x 56 mm. Stacked Breadpi covers the Raspberry pi from the top giving access to all the ports except GPIO header. Almost all the GPIOs can be accessed through the BreadPi.

Pinout

BreadPi	Raspberry Pi Board Pin	Functions
Buzzer	37	Audio signalling
LED L1	32	Light-up LED when pin turned high
LED L2	36	
LED L3	38	
LED L4	40	
Button SW1	29	Take digital input at Raspberry Pi pins
Button SW2	31	
Analog Input(AIN*)	I2C connection	Take analog input from the desired pin
Analog Output(AOUT)	I2C connection	Analog output from AOUT

Note: The other pins are connected with the indicated pin number/name.

Pin Description

BreadPi has 6 sections-

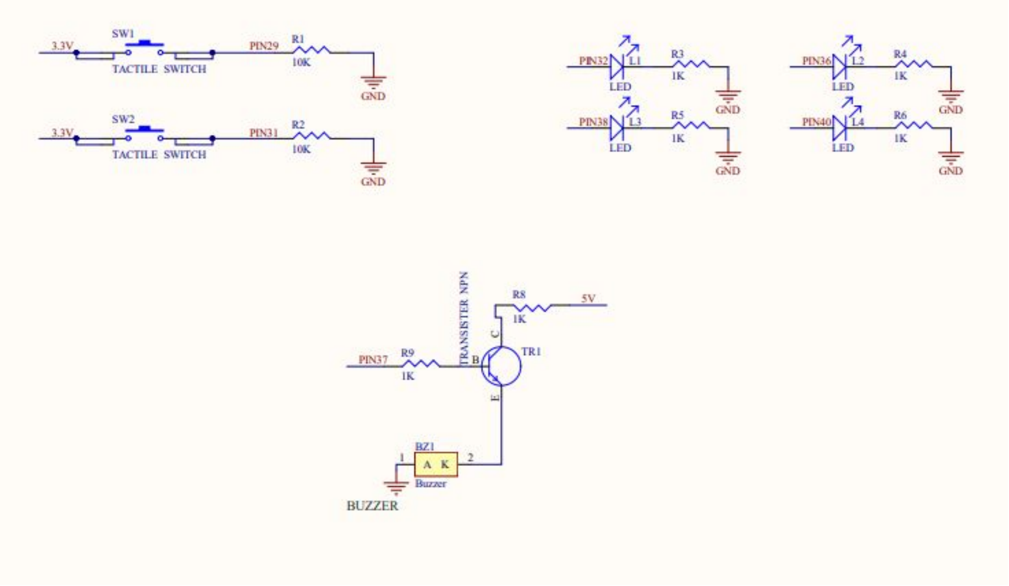
1. Digital I/O,
2. Analog I/O,
3. Power Pins,
4. GPIO Pins,
5. Serial Communication Pins,
6. Breadboard

All these sections provide 1 Buzzer, 4 LEDs, 2 tactical switches, 4 Analog input, 1 Analog output, 3 x (ground, 3v3, and 5v) power pins, 4 x (3v3, 5v) tolerant GPIO pins, and 8 serial communications pins for UART, I2C, and SPI.

Digital I/O

Although it is very easy to interface any Digital I/O device with Raspberry Pi, for beginners it can be really hard and confusing. Everyone makes mistakes of connecting wrong pins and expecting a write output. All the confusion arises from the 40 pin GPIO header of Raspberry Pi. The BreadPi resolves this problem by grouping similar pins together.

There is 1 buzzer, 4 LEDs, and 2 tactical switches on the BreadPi for digital input and output functionalities. These I/O components are mounted on the BreadPi to reduce our effort in making direct connections through jumper wires. Also, the connections have required protection circulatory.



Buzzer- An output polar beeping device. The buzzer used in BreadPi operates at 5 volts. A switching circulatory is used to provide the proper voltage to the buzzer. The pin 37 of Raspberry Pi is used to switch the transistor TR1 which ultimately beeps the buzzer with supply voltage at the collector(5v).

Use *buzzer_on* and *buzzer_off* functions from breadpi module to turn on and off the buzzer. These functions do not take any argument

LED- An another output device widely used in displays and indicators. The LEDs L1, L2, L3, and L4 are at pin 32, 36, 38, and 40 of Raspberry Pi respectively. All the LEDs are in series with a 1 kOhm resistor to limit current and prevent it from burning.

Switch- Switch is an input device. The circuit used for switches is a pull-down resistor circuit which gives a high input when the switch is pressed.

These digital I/O devices are placed for learning and protecting the Raspberry Pi at the same time.

Analog I/O

Like Raspberry Pi, doesn’t have an inbuilt ADC or DAC functionality the BreadPi is made to overcome this problem. The BreadPi uses 8-bit PCF8591 ADC DAC IC.
It has 4 analog output and 1 analog input. It has an I2C-bus serial interface. It has 3 hardware address pins, allowing to connect 8 PCF5891 together.

Hardware address for PCF8591 on BreadPi is ***0x48***. You can make your own program to use its ADC DAC functions or use breadpi module.
The reference voltage is 3.3volts. On BreadPi AIN0 - AIN3 is the analog input pins, while AOUT is the analog output pin.
You can directly use read and write operations on PCF data from breadpi library.
Use

```
pip3 install breadpi
```

or directly use PCF8591 class from https://github.com/sbcshop/BreadPi.
Two functions ‘read_analog’ and ‘write_analog’ are available for reading and write operation. The ‘read_analog’ function takes the register(0-3) parameter which is pin number to read from, ‘write_analog’ takes two parameters- the register(0-3) to write at and the digital equivalent of value(0-255) to write.

Example- If you want to output a voltage of 3.3 volts, the value to write function will be 255. If you want to give an output voltage of 2.2 volts-

Required Voltage = 2.2 volts
Vref = 3.3 volts
Samples = 2^^{8} = 255

Digital Write Value = Samples x Required Voltage / Vref

Digital Write Value = 255 x 2.2 / 3.3 = 170
Which means you have to write a value of 170 to provide analog output voltage of 2.2volts.

The datasheet for PCF8591 is available at-
https://www.nxp.com/docs/en/data-sheet/PCF8591.pdf

Power Pins

Raspberry Pi has 8 ground pins, 2x 3V, and 5V pins. While connecting jumper wires, we have to count the number of pins on the header to reach the desired pin. BreadPi provides resolves this problem by providing these power pins in a section of 3 x Ground, 3v3, and 5v pins. So with a BreadPi, you don’t need to remember the pin number and power pins are handy to use.

GPIO Pins

Almost all the GPIO pins of the Raspberry Pi are on BreadPi for I/O or serial communication. The GPIO part of BreadPi is important as it uses TXB0104 bidirectional voltage-level translator. Which means the GPIO pins 7, 11, 13, and 15 gives 5 volts at the output side while at input end 3v3 is read. This is helpful for protecting GPIO from high voltages and ESD. You can use these pins for 5volt input-output operations.
You can use pin 12, 16, 18, and 22 for 3v3 I/O operations. All the pins on GPIO sections can be used for digital input-output functionalities.
The datasheet for TXB0104 is available at http://www.ti.com/lit/ds/symlink/txb0104.pdf**.**

Serial Communication Pins

As Raspberry Pi has UART, I2C, and SPI serial protocol support, there are dedicated pins for these operations on the Raspberry Pi board. UART pin Tx and Rx are pin 8 and 10, I2C pin SDA and SCL are pin 3 and 5, and SPI pins MOSI, MISO, SCLK, and CE1 are pin 19, 21, 23, and 24 of Raspberry Pi. All these pins are given in the serial section of BreadPi.
It reduces the user’s effort of counting all the pins and focus on functioning and programming.

Breadboard

For training and development, the breadboard is very useful for making connections and simplify the air hanging complicated circuits. BreadPi’s name is derived from the small

breadboard placed in the centre. BreadPi was made keeping in mind the need for breadboard for beginners, trainers, and developers.
The pins on the breadboard are shortened horizontally(through the shorter side of the breadboard). You can place any components or IC and easily jump a connection to the Raspberry Pi board pins.

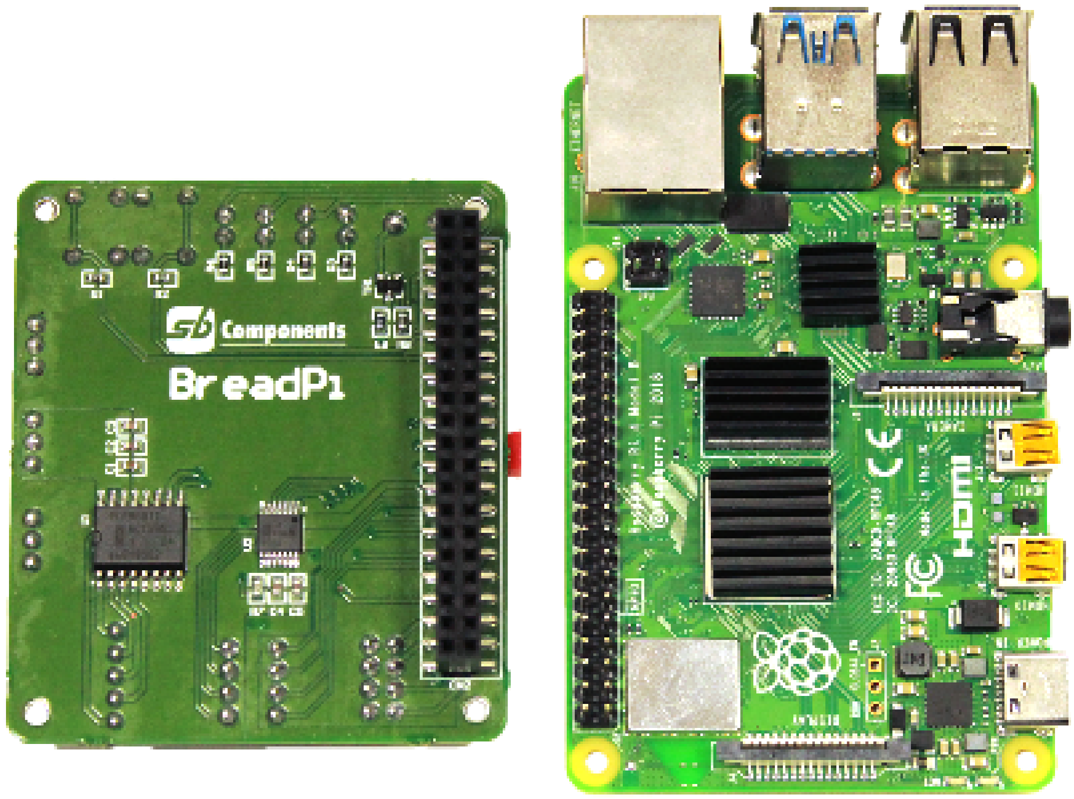
How To Use

BreadPi is a Raspberry Pi hat which extends the Raspberry Pi capabilities. BreadPi doesn't need any external power supply. Using BreadPi is very easy, like other hats. It is stackable over all Raspberry Pi 40 pins models. After stacking you have access to header pins on the top of the board.

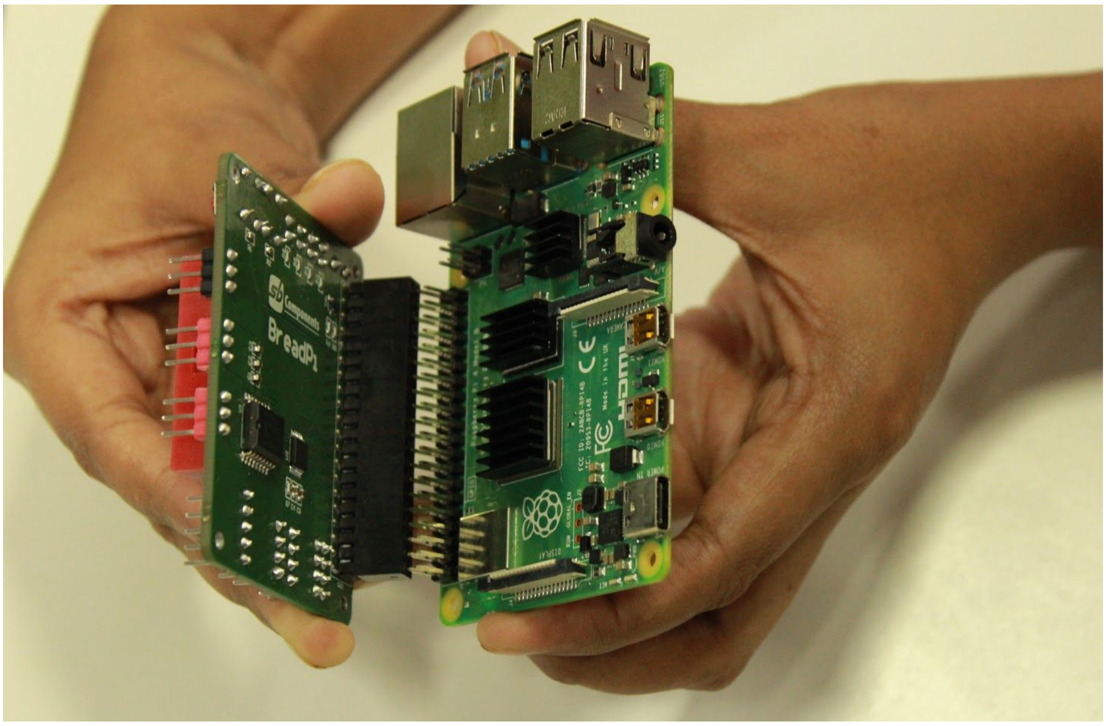
For using Breadpi, stacking it over the Raspberry Pi and using its features from the program are necessary.

Stacking

Before stacking BreadPi over the Raspberry Pi keep the power unplugged. The female headers of the BreadPi will head over to male headers on the Raspberry Pi.



Align BreadPi over the Raspberry pi such that both male and female headers are over each other and Raspberry Pi is covered with BreadPi and only USB ports and ethernet ports are visible from the top.



Insert the BreadPi headers into the Raspberry Pi pins, if all the Raspberry Pi pins are covered give it a gentle push to make the connection tight. Now you can power up the Raspberry Pi.

Downloading and Installation

The resources for BreadPi are available in Python 3. However, you can choose any language to use breadpi.
For getting started with BreadPi you can simply use any GPIO module available for Raspberry Pi. The Pin description is given in this document. You may use any language of your preference. But if you want to make things a little easier you can use breadpi module.
Use pip to install breadpi, in terminal type-

```
pip3 install breadpi
```

Or

```
python3 -m pip install breadpi
```

The related example files can also be cloned to your Raspberry Pi, use git clone command. In terminal type-

```
git clone https://github.com/sbcshop/BreadPi.git
```

Once cloning is successful you can use the example codes and the library file breadpi.py

Programming BreadPi

After the installation is complete, create a python file and import the module with

```
from breadpi import BreadPi
```

For GPIO and Analog to Digital or Digital to Analog conversion, create an instance of the class

```
bread_pi = BreadPi()
```

Now you can access its functions of analog read, analog write, LEDs, buzzers, and switches with the class instance.

For ADC or analog data reading, connect the output of the analog sensor with any AIN pin. Use read_analog function with analog input channel number as an argument. If the AIN1 is used, use register 1-

```
data = bread_pi.read_analog(register=1) # Read Data from AIN1
```

It will return the digital value read from the register 1.

For writing an analog value at AOUT use write_analog function. This function takes register number and the digital value to write at AOUT. If you want output voltage of 2v2 at AOUT, pass the value 170.

```
bread_pi.write_analog(register=1, value=170)
```

For Turning On the buzzer use buzzer_on function with no arguments-

```
bread_pi.buzzer_on()
```

For Turning Off the buzzer use buzzer_off function with no arguments-

```
bread_pi.buzzer_off()
```

For using any LED use led_on() and led_off() functions. These functions take LED name 'L1', 'L2', 'L3', and 'L4' or Board pin numbers like 7, 11, 12, etc as arguments. Ex-

To turn on LEDs, or switch any pin to high-

```
bread_pi.led_on('L1')
bread_pi.led_on(11)
```

To turn off LEDs, or switch any pin to low

```
bread_pi.led_off('L1')
bread_pi.led_off(11)
```

For taking digital input from the button use button function which returns the status of the button at the given pin. The buttons on breadpi are pull down, so if this function returns 1(High) for SW1 or SW2 it means the button is pressed.
Use 'SW1', 'SW2' or board pin numbering 7, 11, 12, etc to read digital input from breadpi.

```
digital_input1 = bread_pi.button('SW1')
digital_input2 = bread_pi.button(11)
```

Example Codes

Running python example codes is very simple and easy. Find the example codes in the 'Examples' directory which you cloned in your Raspberry Pi. The directory will be at location /home/pi/BreadPi/Examples if you didn't change the directory while cloning.
After reaching the Examples directory, you will find various examples of using BreadPi. You can change functions and modify these examples according to your use.

To run a file from the terminal you can directly run the code as-

```
python3 adc_read.py
```

Or you can open the files in any IDE, and simple press run button.