RabbitMax Flex

Open source Raspberry Pi HAT for Internet of Things (IoT)

DISCLAIMER

Raspberry Pi and the Raspberry Pi logo are registered trademarks of the Raspberry Pi Foundation. RabbitMax, the RabbitMax logo and combinations thereof, are registered trademarks of Leon Anavi. Other product names may be trademarks of others and the rights belong to their respective owners.

The information in this document is provided in connection with RabbitMax products. No license, express or implied or otherwise, to any intellectual property right is granted by this document or in connection with the sale of RabbitMax products.

It is possible that the pictures in this manual differ from the latest revision of the board.

The product described in this document is subject to continuous development and improvements. All particulars of the product and its use contained in this document are given by RabbitMax in good faith. However all warranties implied or expressed including but not limited to implied warranties of merchantability or fitness for purpose are excluded. This document is intended only to assist the reader in the use of the product. RabbitMax shall not be liable for any loss or damage arising from the use of any information in this document or any error or omission in such information or any incorrect use of the product.

This evaluation board/kit is intended for use for engineering development, demonstration, or evaluation purposes only and is not considered by RabbitMax to be a finished end-product fit for general consumer use. People handling the product must have electronics training and observe good engineering practice standards. As such, the goods being provided are not intended to be complete in terms of required design-, marketing-, and/or manufacturing-related protective considerations, including product safety and environmental measures typically found in end products that incorporate such semiconductor components or circuit boards.

There is no warranty for the design materials and the components used to create RabbitMax Flex.

There are considered suitable only for RabbitMax Flex.

CHAPTER 1: Overview

Introduction

RabbitMax Flex is an open source hardware Raspberry Pi HAT for Internet of Things (IoT). RabbitMax Flex was started as a hobby project by Leon Anavi in 2016. It is suitable for do it yourself (DIY) weather station, automated desk assistant or rapid prototyping of Internet of Things (IoT).

RabbitMax Flex is designed with the free and open source electronics design automation suite KiCAD. No soldering is required. You can assemble RabbitMax Flex to your Raspberry Pi with your bare hands.

RabbitMax is fully compatible with the Raspbian GNU/Linux distribution and open source sample applications are provided. RabbitMax Flex also supports its own RabbitMax IoT GNU/Linux distribution which is based on the Yocto Project and Openembedded and features open source daemon for communicating with other Internet of Things (IoT) through the lightweight machine-to-machine connectivity protocol MQTT.

Features

RabbitMax Flex Raspberry Pi HAT includes:

- Relay
- IR LED
- IR photo sensor
- Piezoelectric speaker (buzzer)
- Button
- RGB LED
- Slot for modular LCD character display
- Slots for up to 5 plug and play I2C sensors

Supported Raspberry Pi Versions and Models

RabbitMax Flex is compatible with the following Raspberry Pi versions and models:

- Raspberry Pi 3 Model B
- Raspberry Pi 2 Model B
- Raspberry Pi 0
- Raspberry Pi Model B+
- Raspberry Pi Model A+

RabbitMax Flex is **NOT** compatible with the earlier 26-pin models of Raspberry Pi 1 Model B & A's.

Target Market

RabbitMax Flex is a Raspberry Pi HAT suitable for existing Raspberry Pi customers interested in home automation, software development and Internet of Things. The board is appropriate for embedded programming enthusiasts, GNU/Linux gadget fans, students as well as web and/or mobile app developers. The main usage of the board is embedded software development without the urge of understanding perfectly the hardware.

Board Version

Revision 1.1 of RabbitMax Flex was used while writing this document. It is possible that it is outdated so it is always recommended to check the latest sources from the GitHub page of the board.

CHAPTER 2: Getting Started

Electrostatic Warning

RabbitMax Flex is shipped in a protective antistatic bag. The HAT as well as the Raspberry Pi must **NOT** be exposed to high electrostatic potentials. A grounding strap or similar protective device should be worn when handling the board. Avoid touching the component pins or any other metallic element.

Requirements

In order to setup RabbitMax Flex the following items are required:

- Compatible Raspberry Pi
- microSD card with compatible image
- USB power supply

Additionally you may attach USB mouse, keyboard, HDMI monitor or addition peripheral devices to your Raspberry Pi. It is recommended to use 2.5A (2500mA) power supply.

Supported Peripherals

RabbitMax Flex Raspberry Pi HAT has a relay, a button, a piezoelectric speaker (buzzer), a RGB LED, an infrared receiver and an infrared transmitter. Up to 5 I2C sensors and a 1602 character LCD display module can be also easily attached to the HAT.

Sensors

The officially supported sensors by RabbitMax Flex are: * Temperature and barometric pressure

(BMP180) * Temperature and humidity (HTU21) * Light (BH1750)

You may also attach any other I2C sensors but you will have to take care of their software integration.

Temperature Sensor

The official temperature sensor for RabbitMax Flex is BMP180. This is I2C sensor capable of measuring both temperature and barometric pressure.

Using 4 Dupont jumper wires connect BMP180 to one of the 5 I2C slots on RabbitMax Flex as follows:

BMP180	RabbitMax Flex
VIN	3.3V
GND	GND
SCL	SCL
SDA	SDA

Humidity Sensor

The official humidity temperature for RabbitMax Flex is HTU21 (SHT21). This is I2C sensor capable of measuring both humidity and temperature.

Using 4 Dupont jumper wires connect HTU21 to one of the 5 I2C slots on RabbitMax Flex as follows:

HTU21	RabbitMax Flex
VIN	3.3V
GND	GND
SCL	SCL
SDA	SDA

Light Sensor

The official light I2C sensor for RabbitMax Flex is BH1750.

Using 4 Dupont jumper wires connect BH1750 to one of the 5 I2C slots on RabbitMax Flex as follows:

BH1750	RabbitMax Flex
VCC	3.3V
GND	GND
SCL	SCL
SDA	SDA

LCD Display Modules

RabbitMax Flex supports monochrome 1602 character LDC alphanumeric display screen with 16

characters on two rows. Plug the display in the 16 female header pins above the logo of RabbitMax Flex. Use a screwdriver to adjust the brightness of the backlight of the display through the potentiometer.

Assembly

You can assemble RabbitMax Flex and mount it on your Raspberry Pi with your bare hands following the steps below:

- Ensure that you Raspberry Pi is compatible with RabbitMax Flex.
- Power off your Raspberry Pi.
- Gently mount RabbitMax Flex on the 40 pin header of your Raspberry Pi.
- Add sensors and LCD display module to your RabbitMax Flex.
- Optionally, you may also assemble two brass M2.5 standoffs to keep your Pi HAT snug on your Raspberry Pi while also keeping the two boards separated, in particular the HDMI port.
- Optionally, adjust the LED backlight of the LCD display module using a screwdriver.
- That's all, now you are ready to go!

Powering RabbitMax Flex

RabbitMax Flex is Raspberry Pi HAT therefore it is powered through Raspberry Pi. It is recommended to use the official Raspberry Pi Power Supply or another 2.5A USB power supply from reputable retailer.

CHAPTER 3: Software

Installation

In order to work correctly, RabbitMax Flex HAT requires an up-to-date kernel, I2C to be enabled, and a few libraries to get started. After booting microSD card with **Raspbian**, open a terminal and follow the steps below:

- Ensure your APT package list is up-to-date:

```
sudo apt-get update
```

- Install additional applications, libraries and other tools needed by RabbitMax Flex

```
sudo apt-get install -y git i2c-tools vim
```

Enable I2C

Follow the steps below to enable I2C which is required by the sensors for RabbitMax Flex

- Open a terminal or login remotely via SSH to your Raspberry Pi and type in the following command:

```
sudo raspi-config
```

- Select **Advanced Options > I2C** and enable it.
- Reboot the board.

Serial Debugging

Follow the steps below to enable serial debugging through USB to serial cable:

- Open a terminal or login remotely via SSH to your Raspberry Pi and type in the following command:

```
sudo raspi-config
```

- Select **Advanced Options > Serial** and enable it.
- Reboot the board.
- Plug the USB to serial cable and connect your Raspberry Pi to your PC. The **RX** line of the cable should go to **TX** line of **UART1** on RabbitMax Flex. The **TX** line of the cable should go to the **RX** line of **UART1** on RabbitMax Flex. The **GND** wire should go to the **GND** line of **UART1** on RabbitMax Flex.
- Type in the following command on your computer if you are using GNU/Linux distribution to access RabbitMax Flex. Alternatively if your OS is Microsoft Windows use putty.

The recommended USB to serial cable for RabbitMax Flex is Olimex USB-Serial-Cable-F.

Examples

Sample applications written in Python and the C programming language are provided for RabbitMax Raspberry Pi HAT under MIT license in GitHub. All examples have been tested on **Raspbian**.

Open a terminal and execute the follow the steps by step instructions to install all dependencies and to get the source code:

- Install dependencies:

```
sudo apt-get update
sudo apt-get install -y git git-core vim python-dev python-rpi.gpio
```

- Install the GPIO interface library for Raspberry Pi called wiringPi:

```
cd ~
git clone git://git.drogon.net/wiringPi
cd wiringPi
./build
```

- Download the example for RabbitMax Flex Raspberry Pi HAT

```
cd ~
git clone https://github.com/RabbitMax/rabbitmax-examples.git
cd rabbitmax-examples
```

Relay

There is SONGLE SRD-05VDC-SL-C relay on RabbitMax Flex Raspberry Pi HAT. Please refer to the technical documentation of the relay for its specification and wiring guidelines. Couple of examples how to control the relay are provided. Both examples have exactly the same behavior but are written in different programming languages. The examples turn on the relay and turn it of after 3 seconds.

The first example is written in the C programming language and uses the popular library wiring Pi. Execute the commands below to build and run the example.

```
cd ~/rabbitmax-examples/flex/relay/c/
make
sudo ./relay
```

The second example is written in Python. Type the following command to run it:

```
cd ~/rabbitmax-examples/flex/relay/python/
sudo python relay.py
```

Buzzer

There is a built-in piezoelectric speaker (buzzer) in RabbitMax Flex. Three examples are provided to demonstrate how to use it. Two of them are written in the C programming language and the third is written in Python.

Type in the commands below to build and run the first example which is a simple beep application written in C:

```
cd ~/rabbitmax-examples/flex/buzzer/c/
make
sudo ./beep
```

Type in the commands below to build and run the second example which is written in C and plays

sounds similar to the imperial march, Darth Vader's theme, from Star Wars:

```
cd ~/rabbitmax-examples/flex/buzzer/c/
make starwars
sudo ./starwars
```

Execute the command below to run the third example which is written in Python:

```
cd ~/rabbitmax-examples/flex/buzzer/python/
sudo python beep.py
```

Press *Ctrl+C* to terminate any of the example.

Button

There are two examples for handling the button on RabbitMax Flex. The first example is written in the C programming language. Execute the commands below to build and run it:

```
cd ~/rabbitmax-examples/flex/button/c/
make
sudo ./button
```

The second example is written in Python. Execute the following commands to run it:

```
cd ~/rabbitmax-examples/flex/button/python/
sudo python button.py
```

After launching each example press the button on RabbitMax Flex and release it after at least a second and the following message will be printed on the screen "Button released!".

RGB LED

There are two examples that demonstrate the usage of the RGB LED on RabbitMax Flex Raspberry Pi HAT. The first example is written in the C programming language and depends on library wiringPi. Execute the commands below to built and run it:

```
cd ~/rabbitmax-examples/flex/rgb-led/c/
make
sudo ./rgb-led
```

The second example is written in Python. Type in the following commands to run it:

```
cd ~/rabbitmax-examples/flex/rgb-led/python/
sudo python rgb-led.py
```

After launching each one of the examples the RGB LED will start blinking in three different colors: red, green and blue. Press *Ctrl+C* to terminate the example.

LCD Display Module

Mount 1602 LCD display module to RabbitMax Flex. Adjust the brightness of the back light of the LCD display module through the potentiometer using a screwdriver.

Execute the following commands to build and run sample application written in the C programming language to write text on the display:

```
cd ~/rabbitmax-examples/flex/lcd/c/
make
sudo ./lcd
```

Sensors

It is mandatory to enable **I2C** to use any of the sensors supported by RabbitMax Flex Raspberry Pi HAT.

Temperature Sensor (BMP180)

Follow the steps below to use the BMP180 I2C temperature and barometric pressure sensor with RabbitMax Flex:

- Connect BMP180 to any of the I2C slots on RabbitMax Flex using male to female Duport jumper wire.
- Type in the following command and verify that the address of the sensor is listed:

```
sudo i2cdetect -y 1
```

- Type in the following commands to build and run the sample application that display temperature and barometric pressure:

```
cd ~/rabbitmax-examples/flex/sensor-temperature/c/
make
sudo ./rabbitmax-sensor-temperature
```

- Verify that the output is similar to (the exact values depend on the atmospheric condition):

```
pi@raspberrypi:~/rabbitmax-examples/flex/sensor-temperature/c $ sudo
./rabbitmax-sensor-temperature
RabbitMax Temperature and Barometric Pressure Sensor
Temperature 25.0 C
Pressure    995.49 hPa
```

Humidity Sensor (HTU21D)

Follow the steps below to use the HTU21D I2C temperature and humidity sensor with RabbitMax Flex:

- Connect HTU21D to any of the I2C slots on RabbitMax Flex using male to female Duport jumper wire.
- Type in the following command and verify that the address of the sensor is listed:

```
sudo i2cdetect -y 1
```

- Type in the following commands to build and run the sample application that display temperature and humidity:

```
cd ~/rabbitmax-examples/flex/sensor-humidity/c/
make
sudo ./rabbitmax-sensor-humidity
```

- Verify that the output is similar to (the exact values depend on the atmospheric condition):

```
pi@raspberrypi:~/rabbitmax-examples/flex/sensor-humidity/c $ sudo
./rabbitmax-sensor-humidity
RabbitMax Humidity and Temperature Sensor
24.54C
37.78%rh
```

Light Sensor (BH1750)

Follow the steps below to use the BH1750 I2C light sensor with RabbitMax Flex:

- Connect BH1750 to any of the I2C slots on RabbitMax Flex using male to female Duport jumper wire.
- Type in the following command and verify that the address of the sensor is listed:

```
sudo i2cdetect -y 1
```

- Type in the following commands to build and run the sample application that display luminous emittance:

```
cd ~/rabbitmax-examples/flex/sensor-light/c/
make
./rabbitmax-light-sensor
```

- Verify that the output is similar to (the exact values depend on the atmospheric condition):

```
pi@raspberrypi:~/rabbitmax-examples/flex/sensor-light/c $ ./rabbitmax-light-sensor
RabbitMax Light Sensor
Light: 43 Lux
```

Infrared and LIRC

RabbitMax Flex Raspberry Pi HAT has built-in infrared receiver and transmitter. LIRC (Linux Infrared Remote Control) is popular open source application for sending and receiving data over infrared on GNU/Linux distributions. This chapter provides guidelines how to enable RabbitMax Flex infrared receiver and transmitter on **Raspbian** and to use LIRC.

Setting up LIRC

Perform the steps below to enable the infrared receiver and transmitter:

• Install LIRC

```
sudo apt-get update
sudo apt-get install -y lirc
```

• Edit */etc/modules* and add the IR pins by adding the following line to the end of the file:

```
lirc_dev
lirc_rpi gpio_in_pin=18 gpio_out_pin=17
```

• Configure */etc/lirc/hardware.conf* in a way to match:

```
# /etc/lirc/hardware.conf
#
# Arguments which will be used when launching lircd
LIRCD_ARGS="--uinput"

#Don't start lircmd even if there seems to be a good config file
#START_LIRCMD=false

#Don't start irexec, even if a good config file seems to exist.
#START_IREXEC=false

#Try to load appropriate kernel modules
LOAD_MODULES=true

# Run "lircd --driver=help" for a list of supported drivers.
#DRIVER="UNCONFIGURED"
DRIVER="default"
# usually /dev/lirc0 is the correct setting for systems using udev
DEVICE="/dev/lirc0"
MODULES="lirc_rpi"

# Default configuration files for your hardware if any
```

```
LIRCD_CONF=""
LIRCMD_CONF=""
```

- Edit */boot/config.txt* and configure kernel extensions by adding the following line to the end of the file:

```
dtoverlay=lirc-rpi,gpio_in_pin=18,gpio_out_pin=17
```

- Reboot Raspberry Pi:

```
sudo shutdown -r 0
```

Using IR Receiver

Follow the steps below to verify that the IR receiver is working as expected:

- Stop LIRC systemd service:

```
sudo systemctl stop lirc
```

- Start outputting raw data from the IR receiver

```
mode2 -d /dev/lirc0
```

- Point a remote control at the IR receiver on RabbitMax Flex and press its buttons. If the IR receive is configured successfully you will see similar output:

```
space 3662230
pulse 2428
space 594
pulse 1201
space 596
pulse 1230
space 595
pulse 1209
space 590
pulse 1204
```

Using IR LED

Follow the steps below to create LIRC configuration file and test the infrared transmitter:

- Stop LIRC systemd service

```
sudo systemctl stop lirc
```

- Type in the following command to create new LIRC control configuration file and follow the on screen instructions to scan a remote control:

```
irrecord -d /dev/lirc0 ~/lircd.conf
```

List all available names for buttons supported by LIRC:

```
irrecord --list-namespace
```

Example configuration output:

```
Now enter the names for the buttons.

Please enter the name for the next button (press <ENTER> to finish recording)
KEY_POWER

Now hold down button "KEY_POWER".

Please enter the name for the next button (press <ENTER> to finish recording)
KEY_VOLUMEUP

Now hold down button "KEY_VOLUMEUP".

Please enter the name for the next button (press <ENTER> to finish recording)
KEY_VOLUMEDOWN

Now hold down button "KEY_VOLUMEDOWN".

Please enter the name for the next button (press <ENTER> to finish recording)

Successfully written config file.
```

- Backup the original LIRC configuration file:

```
sudo mv /etc/lirc/lircd.conf /etc/lirc/lircd-backup.conf
```

- Load the new configuration file:

```
sudo mv ~/lircd.conf /etc/lirc/lircd.conf
```

- Launch LIRC systemd service again:

```
sudo systemctl start lirc
```

- List all saved keys:

```
irsend LIST /home/pi/lircd.conf ""
```

- Test the configuration file by sending recorded IR command, for example POWER (please note the exact command may vary for different LIRC configuration files, IR devices and IR remote controls):

```
irsend SEND_ONCE /home/pi/lircd.conf KEY_POWER
```

Device Tree Overlays

Device Tree (DT) in Linux is a description of the hardware in a system. The DT overlay adds a number of optional elements.

The EEPROM of RabbitMax Flex contains DT overlay with description of the peripheral devices on the HAT. After adding RabbitMax to your Raspberry Pi and booting it you should have some new filesystem nodes at */proc/device-tree/hat*:

```
pi@raspberrypi:~ $ ls -l /proc/device-tree/hat/
total 0
-r--r--r-- 1 root root  4 Sep 18 23:29 name
-r--r--r-- 1 root root 15 Sep 18 23:27 product
-r--r--r-- 1 root root  7 Sep 18 23:27 product_id
-r--r--r-- 1 root root  7 Sep 18 23:27 product_ver
-r--r--r-- 1 root root 37 Sep 18 23:29 uuid
-r--r--r-- 1 root root 24 Sep 18 23:27 vendor
```

The information provided in these filesystem nodes helps you to identify RabbitMax Flex vendor, version, product name, etc. For example:

```
pi@raspberrypi:~ $ cat /proc/device-tree/hat/product
RabbitMax Flex

pi@raspberrypi:~ $ cat /proc/device-tree/hat/vendor
RabbitMax by Leon Anavi
```

More information about device trees, overlays and parameters are available at the official Raspberry Pi documentation.

CHAPTER 4: RabbitMax IoT GNU/Linux Distribution

RabbitMax IoT GNU/Linux distribution is still *under construction*. It is built using the **Yocto Project** and **OpenEmbedded** and features daemon that sends data from the sensors and receives messages through the lightweight machine-to-machine communication protocol **MQTT**. There is also a responsive HTML5 web interface developed with jQuery Mobile and **Node.js** API. For more details visit:

- rabbitmaxflexd
- meta-rabbitmax
- rabbitmax-api
- rabbitmax-ui

CHAPTER 5: Schematics

Pinout

The components of RabbitMax Flex utilize the following pins on Raspberry Pi:

Component	Pins
I2C	3, 5
EEPROM	27, 28
Relay	29
Button	23
Piezo buzzer	31
RGB LED	33 (blue), 35 (green), 37 (red)
LCD display	7, 13, 15, 19, 21, 40
IR LED	11
IR receiver	12

I2C

The sensors that can be connected to RabbitMax Flex communicate with a host microcontroller via a communications standard called **I2C** (Inter-Integrated-Circut). I2C uses two wires, labelled SDA (Serial Data) and SCL (Serial Clock). To function properly, I2C requires a pullup resistor on each of those lines therefore RabbitMax Flex includes two 4.7kohm resistors labelled as R9 and R10. If for one reason or another you need to disable the I2C pullup resistors remove R9 and R10.

LCD Module

1602 LCD character display module can be attached to RabbitMax Flex. The HAT supports modules with 16 characters on 2 rows which are compatible with the Hitachi HD44780 LCD controller. The LCD module requires 8 data lines to provide data to bits 0-7 but it can be also set to a 4 bit mode which allows sending data in two chunks. Each chuck is 4 bits. The 4 bit mode is convenient as it

reduces the number of used GPIO pins on Raspberry Pi. RabbitMax Flex uses the following wiring for the connector for 1602 LCD modules:

LCD	Function	Raspberry Pi Pin
1	GND	6 (GND)
2	5V	2
3	Contrast	6 (GND)
4	RS	7
5	RW	6 (GND)
6	E	40
7	Data 0	13
8	Data 1	15
9	Data 2	19
10	Data 3	21
11	Data 4	-
12	Data 5	-
13	Data 6	-
14	Data 7	-
15	5V via potentiometer	2
16	GND	6 (GND)

LED backlight of the LCD display module can be manually adjusted through the potentiometer POT1 using a screwdriver.

CHAPTER 6: Frequently Asked Questions (FAQ)

May I use RabbitMax Flex with other operating systems?

Yes, you can use RabbitMax Flex with other GNU/Linux distributions and even other operating systems but some porting efforts might be required.

May I use other I2C sensors with RabbitMax Flex?

Yes, you can use other I2C sensors with RabbitMax Flex but you should take care for their drivers and software support.

May I use non-I2C sensors with RabbitMax Flex?

No.

Is RabbitMax Flex software free and open source?

Yes, the official RabbitMax Flex software is free and open source. The examples are available under MIT license and the rest is available under GPLv3. Please contact us if you are working on a

commercial product and you would like to use the software under alternative commercial license.

CHAPTER 7: Revision History

Document Revision

Date	Changes	Modified pages	Author
14 September 2016	Initial manual release	All	Leon Anavi

RabbitMax Flex HAT Revision

Revision	Notable changes
1.0	First prototype
1.1	Piezo buzzer and button changed, potentiometer added

See Also

For more information please visit rabbitmax.com and our GitHub repositories. If you have any questions or enquiries please contact us through Facebook, Twitter or email.