

ANAVI Infrared pHAT

Convert Your Raspberry Pi to Powerful Remote Control

DISCLAIMER

Raspberry Pi and the Raspberry Pi logo are registered trademarks of the Raspberry Pi Foundation. ANAVI, the ANAVI 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 Anavi 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 Anavi products.

This work is licensed under the Creative Commons Attribution-ShareAlike 3.0 Unported License. To view a copy of this license, visit <http://www.creativecommons.org/licenses/by-sa/3.0/>.

ANAVI Infrared pHAT hardware design is licensed under a Creative Commons Attribution-ShareAlike 3.0 Unported License.

The software examples are released under MIT and the rest of the software is available under GPLv3.

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 Anavi 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. Anavi 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 Anavi 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 ANAVI Infrared pHAT. They are considered suitable only for ANAVI Infrared pHAT.

CHAPTER 1: Overview

Introduction

ANAVI Infrared pHAT is an open source hardware Raspberry Pi add-on board with IR receiver, transmitter, UART and 3 I2C slots for sensors. ANAVI Infrared pHAT was started as a hobby project by Leon Anavi in January 2017. The project allows you to convert your Raspberry Pi into a smart remote control using the open source software LIRC.

ANAVI Infrared pHAT is designed with the free and open source electronics design automation suite [KICAD](#). No soldering is required. You can assemble ANAVI Infrared pHAT to your Raspberry Pi with your bare hands.

Anavi is fully compatible with the Raspbian GNU/Linux distribution and open source sample applications are provided.

Features

ANAVI Infrared pHAT Raspberry Pi HAT includes:

- IR LED
- IR photo sensor
- Slots for up to 3 plug and play I2C sensors

Supported Raspberry Pi Versions and Models

ANAVI Infrared pHAT is compatible with the following Raspberry Pi versions and models:

- [Raspberry Pi 4 Model B](#)
- [Raspberry Pi 3 Model B](#)
- [Raspberry Pi 2 Model B](#)
- [Raspberry Pi 0](#)
- [Raspberry Pi 0 W](#)
- [Raspberry Pi Model B+](#)
- [Raspberry Pi Model A+](#)

ANAVI Infrared pHAT is NOT compatible with the earlier 26-pin models of Raspberry Pi 1 Model B & A's.

Target Market

ANAVI Infrared pHAT 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 and controlling IR devices without the urge of understanding perfectly the hardware.

Board Version

Revision 1.0 of ANAVI Infrared pHAT 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

ANAVI Infrared pHAT 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 ANAVI Infrared pHAT 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

ANAVI Infrared pHAT Raspberry Pi HAT has an infrared receiver and an infrared transmitter. Up to 3 I2C sensors can be also easily attached to the pHAT.

Sensors

The officially supported I2C sensor modules by ANAVI Infrared pHAT 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 ANAVI Infrared pHAT 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 3 I2C slots on ANAVI Infrared pHAT as follows:

BMP180 ANAVI Infrared pHAT

VIN	3.3V
GND	GND
SCL	SCL
SDA	SDA

Humidity Sensor

The official humidity temperature for ANAVI Infrared pHAT 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 3 I2C slots on ANAVI Infrared pHAT as follows:

HTU21 ANAVI Infrared pHAT

VIN	3.3V
GND	GND
SCL	SCL
SDA	SDA

Light Sensor

The official light I2C sensor for ANAVI Infrared pHAT is BH1750.

Using 4 Dupont jumper wires connect BH1750 to one of the 3 I2C slots on ANAVI Infrared pHAT as follows:

BH1750 ANAVI Infrared pHAT

VCC	3.3V
GND	GND
SCL	SCL
SDA	SDA

Assembly

You can assemble ANAVI Infrared pHAT and mount it on your Raspberry Pi with your bare hands following the steps below:

- Ensure that you Raspberry Pi is compatible with ANAVI Infrared pHAT.
- Power off your Raspberry Pi.
- Gently mount ANAVI Infrared pHAT on the 40 pin header of your Raspberry Pi.
- Add sensors to your ANAVI Infrared pHAT.
- Optionally, you may also mount two or four brass M2.5 standoffs to keep your Pi HAT snug on your Raspberry Pi while also keeping the two boards separated.
- That's all, now you are ready to go!

Powering ANAVI Infrared pHAT

ANAVI Infrared pHAT 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, ANAVI Infrared pHAT 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 ANAVI Infrared pHAT

```
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 ANAVI Infrared pHAT

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

```
sudo raspi-config
```

- Select Interfacing 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 Interfacing 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 UART on ANAVI Infrared pHAT. The TX line of the cable should go to the RX line of UART on ANAVI

Infrared pHAT. The GND wire should go to the GND line of UART on ANAVI Infrared pHAT.

- Type in the following command on your computer if you are using GNU/Linux distribution to access ANAVI Infrared pHAT. Alternatively if your OS is Microsoft Windows use [putty](#).

The recommended USB to UART serial modules for ANAVI Infrared pHAT are any with CP2102 or [Olimex USB-Serial-Cable-F](#). All Infrared kits include CP2102 which out of the box on GNU/Linux distributions. Drivers for MS Windows and Mac OS X are [available at silabs.com](#).

Examples

Sample applications written in Python and the C programming language are provided for ANAVI Infrared pHAT 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 examples for ANAVI Infrared pHAT

```
cd ~
git clone https://github.com/AnaviTechnology/anavi-examples.git
cd anavi-examples
```

Sensors

It is mandatory to enable I2C to use any of the sensors supported by ANAVI Infrared pHAT.

Temperature Sensor (BMP180)

Follow the steps below to use the BMP180 I2C temperature and barometric pressure sensor with ANAVI Infrared pHAT:

- Connect BMP180 to any of the I2C slots on ANAVI Infrared pHAT 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 ~/anavi-examples/sensors/BMP180/c/
make
./BMP180
```

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

```
pi@raspberrypi:~/anavi-examples/sensors/BMP180/c $ ./BMP180
BMP180 Sensor Module
Temperature 28.6 C
Pressure 991.57 hPa
```

Humidity Sensor (HTU21D)

Follow the steps below to use the HTU21D I2C temperature and humidity sensor with ANAVI Infrared pHAT:

- Connect HTU21D to any of the I2C slots on ANAVI Infrared pHAT 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 ~/anavi-examples/sensors/HTU21D/c/
make
./HTU21D
```

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

```
pi@raspberrypi:~/anavi-examples/sensors/HTU21D/c $ ./HTU21D
HTU21D Sensor Module
25.64C
118.99%rh
```

BH1750 Light Sensor Module

Follow the steps below to use the BH1750 I2C light sensor with ANAVI Infrared pHAT:

- Connect BH1750 to any of the I2C slots on ANAVI Infrared pHAT using male to female Dupont jumper wires.
- 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 ~/anavi-examples/sensors/BH1750/c/
make
./BH1750
```

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

```
pi@raspberrypi:~/anavi-examples/sensors/BH1750/c $ ./BH1750
BH1750 Sensor Module
Light: 418 Lux
```

Infrared and LIRC

ANAVI Infrared pHAT 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 ANAVI Infrared pHAT infrared receiver and transmitter on Raspbian and to use LIRC.

In 2019 `lirc_rpi`, the Linux kernel module provided with Raspbian before, was replaced with `gpio-ir` and `gpio-ir-tx`. This tutorial is updated for Raspbian Buster. If you need to setup ANAVI Infrared pHAT on an older version of Raspbian, for example from 2018-04-18, please [have a look at the old user's manual](#).

Setting up LIRC

Perform the steps below to build LIRC from source, to patch it and to enable the infrared receiver and transmitter on ANAVI Infrared pHAT:

- Install dependencies

```
sudo su -c "grep '^deb ' /etc/apt/sources.list | sed 's/^deb/deb-src/g' > /etc/apt/sources.list.d/deb-src.list"
sudo apt update
sudo apt install -y vim devscripts dh-exec doxygen expect libasound2-dev libftdi1-dev libsystemd-dev libudev-dev libusb-1.0-0-dev libusb-dev man2html-base portaudio19-dev socat xsltproc python3-yaml dh-python libx11-dev python3-dev
```

- Download LIRC source code

```
mkdir ~/lirc-src
cd ~/lirc-src
apt source lirc
```

- Apply a patch to fix LIRC for Raspberry Pi

```
wget https://raw.githubusercontent.com/neuralassembly/raspi/master/lirc-gpio-ir-0.10.patch
patch -p0 -i lirc-gpio-ir-0.10.patch
cd lirc-0.10.1
debuild -uc -us -b
```

- Install LIRC (built on the previous step)

```
cd ~/lirc-src
sudo apt install ./liblirc0_0.10.1-5.2_armhf.deb ./liblircclient0_0.10.1-5.2_armhf.deb ./lirc_0.10.1-5.2_armhf.deb
```

NOTE: the installation is expected to fail the first time when you run it. After applying changes to some configurations LIRC will be installed again in the next steps.

- Deploy LIRC configurations:

```
sudo cp /etc/lirc/options.conf.dist /etc/lirc/options.conf
sudo cp /etc/lirc/lircd.conf.dist /etc/lirc/lircd.conf
```

- Edit `/etc/lirc/lirc_options.conf` and make sure that driver and device are set as:

```
driver = default
device = /dev/lirc1
```

NOTE: Device `/dev/lirc1` is the receiver and device `/dev/lirc0` is the transmitter. Initially `/dev/lirc1` is used to scan a remote control. After that the configuration has to be updated to `/dev/lirc0` in order to send infrared commands.

- Edit `/boot/config.txt` (with `sudo` or as `root`) and configure kernel extensions by adding the following line to the end of the file:

```
dtoverlay=gpio-ir,gpio_pin=18
dtoverlay=gpio-ir-tx,gpio_pin=17
```

- Run the LIRC installation again:

```
cd ~/lirc-src
sudo apt install -y --allow-downgrades ./liblirc0_0.10.1-5.2_armhf.deb ./liblircclient0_0.10.1-5.2_armhf.deb ./lirc_0.10.1-5.2_armhf.deb
```

- 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 lircd
```

- Start outputting raw data from the IR receiver

```
mode2 -d /dev/lirc1
```

- Point a remote control at the IR receiver on ANAVI Infrared pHAT and press its buttons. If the IR receiver 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 lircd
```

- List all available names for buttons supported by LIRC:

```
irrecord --list-namespaces
```

- 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/lirc1 ~/lircd.conf
```

Example configuration output with name hifi:

```
Using driver default on device /dev/lirc1
```

```
irrecord - application for recording IR-codes for usage with lirc
Copyright (C) 1998,1999 Christoph Bartelmus(lirc@bartelmus.de)
```

```
This program will record the signals from your remote control
and create a config file for lircd.
```

```
A proper config file for lircd is maybe the most vital part of this
package, so you should invest some time to create a working config
file. Although I put a good deal of effort in this program it is often
not possible to automatically recognize all features of a remote
control. Often short-comings of the receiver hardware make it nearly
impossible. If you have problems to create a config file READ THE
DOCUMENTATION at https://sf.net/p/lirc-remotes/wiki
```

```
If there already is a remote control of the same brand available at
http://sf.net/p/lirc-remotes you might want to try using such a
remote as a template. The config files already contains all
parameters of the protocol used by remotes of a certain brand and
knowing these parameters makes the job of this program much
easier. There are also template files for the most common protocols
available. Templates can be downloaded using irdb-get(1). You use a
template file by providing the path of the file as a command line
parameter.
```

```
Please take the time to finish the file as described in
https://sourceforge.net/p/lirc-remotes/wiki/Checklist/ and send it
to <lirc@bartelmus.de> so it can be made available to others.
```

```
Press RETURN to continue.
```

```
Checking for ambient light creating too much disturbances.
Please don't press any buttons, just wait a few seconds...
```

```
No significant noise (received 0 bytes)
```

```
Enter name of remote (only ascii, no spaces) :hifi
Using hifi.lircd.conf as output filename
```

```
Now start pressing buttons on your remote control.
```

```
It is very important that you press many different buttons randomly
and hold them down for approximately one second. Each button should
generate at least one dot but never more than ten dots of output.
Don't stop pressing buttons until two lines of dots (2x80) have
been generated.
```

```
Press RETURN now to start recording.
```

```
.....
Got gap (45034 us)
```

```
Please keep on pressing buttons like described above.
.....
```

```
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)
```

```
Checking for toggle bit mask.
Please press an arbitrary button repeatedly as fast as possible.
Make sure you keep pressing the SAME button and that you DON'T HOLD
the button down!
If you can't see any dots appear, wait a bit between button presses.
```

```
Press RETURN to continue.
```

```
.....Cannot find any toggle mask.
You have only recorded one button in a non-raw configuration file.
This file doesn't really make much sense, you should record at
least two or three buttons to get meaningful results. You can add
more buttons next time you run irrecord.
```

```
Successfully written config file hifi.lircd.conf
```

- Backup the original LIRC configuration file:

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

- Load the new configuration file, for example:

NOTE: The name of configuration depends on the selected name of remote. Please adapt the command below depending on your name of remote!

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

- Switch LIRC configurations to the device for transmitting. Edit again /etc/lirc/lirc_options.conf and make sure that driver and device are set as:

```
driver = default
device = /dev/lirc0
```

NOTE: In this case the device is /dev/lirc0.

- Launch LIRC systemd service again:

```
sudo systemctl start lircd
```

- List all saved keys, for example:

NOTE: Please adapt the command below depending on your name of remote!

```
irsend LIST hifi ""
```

- ◦ 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):

NOTE: Please adapt the command below depending on your name of remote!

```
irsend SEND_ONCE hifi 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 ANAVI Infrared pHAT contains DT overlay with description of the peripheral devices on the HAT. After adding Anavi 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 Feb 15 00:27 name
-r--r--r-- 1 root root 20 Feb 15 00:27 product
-r--r--r-- 1 root root 7 Feb 15 00:27 product_id
-r--r--r-- 1 root root 7 Feb 15 00:27 product_ver
-r--r--r-- 1 root root 37 Feb 15 00:27 uuid
-r--r--r-- 1 root root 6 Feb 15 00:27 vendor
```

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

```
pi@raspberrypi:~$ cat /proc/device-tree/hat/product
ANAVI Infrared pHAT

pi@raspberrypi:~$ cat /proc/device-tree/hat/vendor
ANAVI
```

More information about device trees, overlays and parameters are available at [the official Raspberry Pi documentation](#).

CHAPTER 4: ANAVI IoT GNU/Linux Distribution

Anavi 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:

- [anaviflexd](#)
- [meta-anavi](#)
- [anavi-api](#)
- [anavi-ui](#)

CHAPTER 5: Schematics

Pinout

The components of ANAVI Infrared pHAT utilize the following pins on Raspberry Pi:

Component	Pins
I2C	3, 5
EEPROM	27, 28
IR LED	11
IR receiver	12
UART	1, 8, 9, 10

I2C

The sensors that can be connected to ANAVI Infrared pHAT communicate with a host microcontroller via a communications standard called I2C (Inter-Integrated-Circuit). 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 ANAVI Infrared pHAT includes two 4.7kohm resistors labelled as R6 and R7. If for one reason or another you need to disable the I2C pullup resistors remove R6 and R7.

CHAPTER 6: Frequently Asked Questions (FAQ)

May I use ANAVI Infrared pHAT with other operating systems?

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

May I use other I2C sensors with ANAVI Infrared pHAT?

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

May I use non-I2C sensors with ANAVI Infrared pHAT?

No.

Can I remotely control ANAVI Infrared pHAT from a web browser on my smartphone, tablet, or laptop?

Yes, you may install popular open source software for home automation on your Raspberry Pi such as [Home Assistant](#) or [OpenHAB 2](#) and integrate ANAVI Infrared pHAT.

Is ANAVI Infrared pHAT software free and open source?

Yes, the official ANAVI Infrared pHAT 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
15 February 2017	Initial manual release	All	Leon Anavi
16 September 2017	Update for Raspbian Stretch	All	Leon Anavi
08 July 2019	Update for Raspbian Buster	All	Leon Anavi

ANAVI Infrared pHAT Revision

Revision Notable changes

1.0 First version

See Also

For more information please visit [anavi.technology](#) and our [GitHub repositories](#). If you have any questions or enquiries please contact us through [Facebook](#), [Twitter](#) or [email](#).
