Trilby HAT – Getting Started

Kinetic Avionics Ltd

Version 1.03

28 April 2016

Introduction

Thank you for purchasing the Trilby HAT. It adds Software Defined Radio, a Field Programmable Gate Array (FPGA), and a Real Time Clock chip to the Raspberry Pi.

It is designed to work with the Raspberry Pi models 1B+, 2B or 3B. These models have a 40-pin HAT connector.

In normal operation the Trilby HAT is powered via the Raspberry Pi and the HAT connector.

Note - the Trilby HAT can also be used for stand-alone operation without a Raspberry Pi, in which case the HAT is powered via its mini-USB socket. This mode of operation requires specialised firmware and is beyond the scope of this document.

To try out the provided Software Defined Radio sample, you will need an antenna with an SMA connector, either a VHF/UHF antenna connected to J2 and/or a HF antenna connected to J4.

You will also need an audio output device connected to the Raspberry Pi 3.5mm jack socket or HDMI port. Alternatively you may connect an audio device directly to the Trilby HAT using the three pin header J457 near the mini-USB connector (the centre pin is ground); however external amplification will be required when using this option.

Installing the Trilby HAT Hardware

1. If you intend to use the Real Time Clock chip's battery back-up feature, you may wish to install a type CR1220 3V lithium battery before installing the board. This must be installed with the +ve terminal facing up, away from the surface of the Trilby HAT board. Alternatively the board can be unplugged at a later date to install the battery.
2. With the Raspberry Pi powered down, carefully plug the Trilby HAT onto the Raspberry Pi board, ensuring that the pins on the 40-pin header of the Raspberry Pi line up correctly with the connector on the Trilby HAT, and that the 4 mounting holes on both boards line up. It is recommended that the correct spacers and screws be used to ensure a stable mounting of the HAT board.

3. Check that the power mode selection jumper (J456, in the "bottom left" corner of the Trilby HAT board when the 40-pin HAT connector is at the top edge), adjacent to the mini USB socket, is set to link pins 1 and 2 (the two pins furthest from the mini USB socket).
4. In order to allow upload of new firmware versions from the Raspberry Pi, Both J450 and J451 jumpers should be installed. These are at the "top-left" corner of the Trilby HAT board when the 40-pin connector is at the top edge. To obtain an extra jumper you may "borrow" the jumper on J26 (this is the write enable for the HAT ID EEPROM, and is adjacent to the 24-pin expansion header on the top of the Trilby board).
5. It is also worth checking at this point that the remaining jumpers are correctly set:
   J30 – installed
   J31 – not installed
   J32 – installed
   J33 – not installed
6. With the monitor, keyboard and/or mouse connected to the Raspberry Pi as required, power up the Raspberry Pi as normal. The Trilby HAT should be powered up automatically through the HAT connector.
7. After a few seconds, if a red light is blinking once per second on the Trilby HAT then you have successfully completed the hardware installation. If there are no LEDs at all lit on the Trilby HAT when the Raspberry Pi is powered up, then check that the power jumper is correctly installed as described above.

Software Installation

1. It is assumed that an up to date version of the Raspbian operating system is installed on your Raspberry Pi, and that you are now logged in and have a command prompt. To get any updates you can use the following commands:
   ```
   sudo apt-get update
   sudo apt-get -y upgrade
   ```
2. Run raspi-config. If you are not logged in as "root" or other super-user you will need to prefix the command with "sudo":
   ```
   sudo raspi-config
   ```
3. Select the advanced options and use the SPI option to ensure that the SPI interface and SPI kernel are enabled. These are used by the ttune utility. Press Yes to enable the interface, then OK and Yes again to load the kernel by default. Likewise use the advanced options-I2C menu option to enable the I2C interface and I2C kernel (which are required for the real time clock). Then use the advanced options-Audio option to ensure that the required audio output device, HDMI or 3.5mm jack, is enabled. When you are returned to the main menu, navigate to the Finish button to exit.
4. Start the Raspberry Pi GUI by typing
   ```
   startx
   ```

   Then run the web browser, which can be found on the start menu of the GUI, under "Internet".

5. Navigate to www.kinetic.co.uk and go to the Trilby HAT downloads page. Click on the button to download the Trilby HAT utilities. These are contained in a ZIP file which the browser will download to the Downloads directory on the Pi.
6. Also click on the button to download the current Trilby HAT firmware file, which will have the file type .vme
7. Now run the File Manager (under Accessories on the start menu, or by clicking on the filing cabinet icon on the menu bar). In the File Manager window, open the Downloads folder by double clicking on it. Then right click on the ZIP file and select "extract here". The contents of the file ZIP file should then appear in the folder.
8. Change the permissions of the vme and ttune utilities to make them executable. To do this, right click on the icon "vme" and select "properties", and then go to the permissions tab. Under "execute" select "anyone" and then click on "Save". Repeat this procedure for the ttune icon.
9. Upload the latest version of the firmware to the FPGA's flash memory in the Trilby HAT so that you get the latest and features and bug fixes. To do this, identify the firmware file name (which will end in .vme) from the file manager window. Now close the X GUI by selecting "shutdown" and "log out" from the start menu, or alternatively open a terminal window ("accessories" – "terminal" on the start menu).
10. Change to the Downloads folder by typing:
    cd Downloads
11. Upload the firmware by typing the vme command with the correct versioned file name, for example:
    sudo ./vme trilby_1_03.vme
    if all is well, the utility will display the word "PASS" before returning to the command prompt. If not, check that the file name is correct, the Trilby HAT is powered up and that the jumpers J450 and J451 are both installed.
12. Now you need to power-cycle the Trilby HAT to ensure that the new firmware is actually loaded into the FPGA. You can do this without powering down the Raspberry Pi by temporarily removing and then reinstalling the power jumper (J456). After a few seconds the red light should flash again.
13. If you have connected a VHF antenna to J2, and an audio output device to the Raspberry Pi, why not try tuning the Software Defined Radio to a local FM broadcast station? The ttune utility needs the frequency in kHz, for example for 93.5 MHz you could type:
    sudo ./ttune –ws 93500
    The –w option tells it to tune to a wideband FM station, and the –s option tells it to send the sound to the audio device of the Raspberry Pi. If you connected your audio device directly to the Trilby HAT then omit the "s". Notice that as well as its own version number, ttune also reports the version number of the firmware in the Trilby HAT FPGA.
    The various options for ttune are described elsewhere or you can run ttune without any arguments to get a list.

Configuring the Real Time Clock

1. Install the battery if this has not already been done, as described in the section "Installing the Trilby HAT Hardware" above. Without the battery the time will not be retained when the power is removed.
2. Edit the modules file. To do this, type the command:
   sudo nano /etc/modules
3. add the following line to the file:
   rtc-ds1307
   Then exit from nano and save the file using CTRL-X followed by Y and ENTER.
4. Edit the startup file. To do this, type the command:
   sudo nano /etc/rc.local
   add the following two lines before the "exit 0" line (if you are using a version 1 Raspberry Pi then you need to replace the i2c-1 with i2c-0):
   echo ds1307 0x68 > /sys/class/i2c-adapter/i2c-1/new_device
   sudo hwclock -s
   Then exit from nano and save the file using CTRL-X followed by Y and ENTER.
5. Restart the system by typing
   sudo reboot
6. Set the time in the hardware (real time) clock. Do this while the Raspberry Pi is connected to the network so that the system time is correct, or if necessary set the system time manually. The command to write the system time to the hardware clock is:
   sudo hwclock -w
7. Power down the Raspberry Pi (for example using sudo halt), disconnect the network and wait several minutes. When you restore the power and the system restarts, the time should still be correct, even with the network disconnected.
8. You can display the time from the hardware clock at any time by using the command:
   sudo hwclock -r
   you can set the system time from the hardware clock by typing:
   sudo hwclock -s
   (this is what we did in the startup file).
9. You can install a handy i2c diagnostic tool from the network by typing:
   sudo apt-get install -y i2c-tools
   Running the i2cdetect tool shows what I2C slave devices are connected to the I2C bus:
   sudo i2cdetect -y 1
   The result should look like this. Address 44 is an example I2C device provided within the sample FPGA firmware. Address 68 is the Real Time Clock, which shows as UU to indicate that it is in use:

```
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
```

```
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- 44 -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- UU -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
```

Raspberry Pi is a trademark of the Raspberry Pi Foundation.