YEPKiT

YKUSH Yepkit USB Switchable Hub

User Manual

Version 1.1

INDEX

1. PRODUCT OVERVIEW

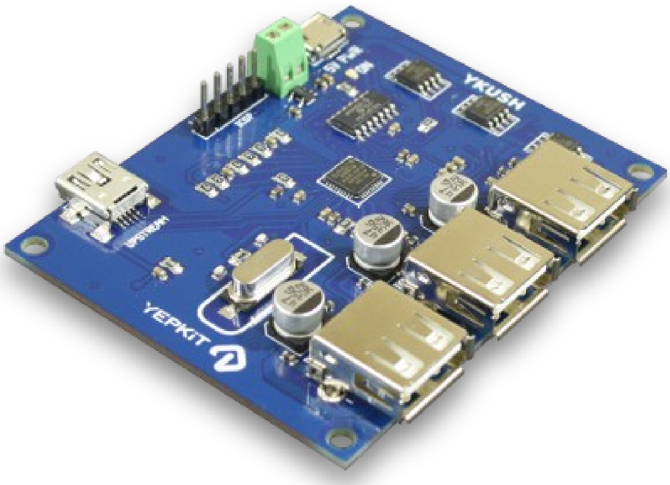
2. CONNECTING AND SETTING-UP

3. USING IT

4. CONTROL PROTOCOL

PRODUCT OVERVIEW

CONTROL YOUR USB DEVICES



Yepkit USB Switchable Hub

Linux boards, like Raspberry Pi, are widely used as home servers. One of the main functions of these home servers is to perform automatic periodic data backups being the typical storage medium used USB external disk drives (mechanical). Many of these drives tend to stay ON and "spinning" even when not being used (even if unmounted in the file system), which is not power efficient and may impact the lifetime of the drive.

To address this issue, YKUSH provides to the user the capability, through software, to selectively turn ON and OFF each of the USB devices connected to the hub.

The selective control is achieved by sending ON/OFF commands to the on-board microcontroller, visible to the host system as a HID USB device. The commands are based in a very simple protocol detailed in Control Protocol section of this manual.

Key Features:

- Three downstream 2.0 high-speed USB ports
- 2A maximum continuous current per downstream port
- Individual power drivers per downstream port
- 5V externally powered USB hub
- On-board microcontroller configured as a HID USB device

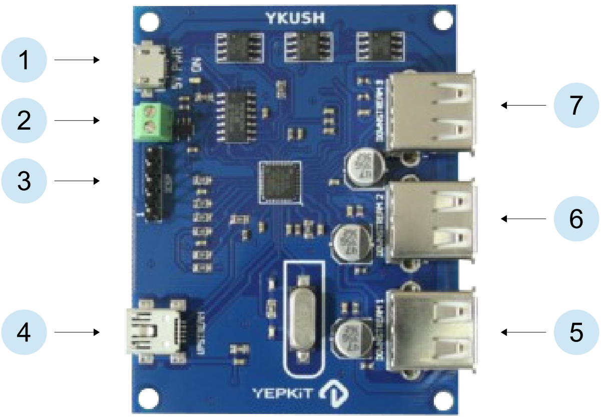
Key Benefits:

- Full control over your USB devices ON/OFF
- Open Hardware board
- Open Source drivers and Firmware you can alter to serves your specific requirements in full
- Low cost flexible USB Hub

CONNECTING AND SETTING-UP

YKUSH board terminals

1 5V PWR – Power input from 5 Volt power supply
2 Alternative 5 Volt power supply input terminal
3 ICSP Programming interface
4 Upstream USB Port
5 Downstream USB Port 1
6 Downstream USB Port 2
7 Downstream USB Port 3



The hardware setup for using YKUSH is quite simple, it's connected exactly as any other externally powered USB Hub. The upstream port is connected to the host system and an external 5 Volt power supply must be connected to the YKUSH Power In.

At this point YKUSH is a working high-speed USB Hub which can have up to three USB devices connected to it.

To unlock the enhanced functionalities a software component must be installed in the host system (or custom developed by the user).

This software component in the host system implements the communication protocol with the YKUSH in-board microcontroller that controls the downstream ports ON/OFF individually.

We provide a code that can be built in Linux systems than provides a command utility to send control commands to YKUSH.
You can get this software source code in the YKUSH product page at www.yepkit.com.

The software provided sends commands from the host system to YKUSH to perform the following actions:

- Power-up a specific downstream port
- Power-down a specific downstream port
- Power-up all downstream ports
- Power-down all downstream ports

To make these commands available in the host system, the source code provided by us must be built and installed.

CONNECTING AND SETTING-UP

Building the software

Before building the YKUSH software ensure that your Linux system has the following installed, available and working:

- Make utility
- g++
- libusb library

If the above is satisfied you can proceed to build and install the software.

Start by downloading the source code from the product page at www.yepkit.com. Then extract the source code files from the compressed archive.
To do this, navigate in a shell console to the directory where the compressed package was downloaded to and run the following command.

```
tar -xzvf ykush_v1.2.tar.gz
```

To compile and build the software enter the YkUSH folder and use the make command.

```
make
```

If everything worked normally there will be an executable file named *ykush* in the directory and you can start issuing commands to YKUSH.

Installing the software

There are several ways you may use to install the software. We will present two of them.

If this command will only be used only by you, it can be made accessible to you from anywhere using the *.bashrc* script from your user home environment.

A set-up we typically use is to have a */home/your_user_name/bin* directory to where our local executables are installed. If we consider such a set up you could copy or move the just built ykush executable to this directory.

```
cp ykush /home/your_user_name/bin/
```

Now edit the *.bashrc* file.

```
vi ~/.bashrc
```

Add the following line to this file.

```
alias ykush=/home/your_user_name/bin/ykush
```

Open a new terminal and you should be able to call the ykush command from anywhere.

An alternative installation manner, which will make the ykush command visible to everyone, is just copying or moving the ykush executable file to the */usr/bin/* folder.

USING IT

USING YOUR YKUSH

Now that everything is set-up you can start using the full capabilities of YKUSH instead of just using it as a mere USB hub.

If you are using the provided software, built as described in the previous section, the commands have the following structure and options.

The command structure

```
sudo ykush option port_number
```

option:

-u	Turns the port up
-d	Turns the pot down

port_number:

1	Downstream port 1
2	Downstream port 2
3	Downstream port 3
a	All downstream ports

Some examples

The following command turns the downstream port number 1 down.

```
sudo ykush -d 1
```

The following command turns the downstream port number 1 up.

```
sudo ykush -u 1
```

The following command turns all downstream ports down.

```
sudo ykush -d a
```

The following command turns all downstream ports up.

```
sudo ykush -u a
```

CONTROL PROTOCOL

UNDERSTANDING THE PROTOCOL

YKUSH board has an on-board microcontroller that handles the control communication with the host system and performs the switching of the downstream USB ports.

This in-board microcontroller is visible to the host system as a HID USB device with the following details:

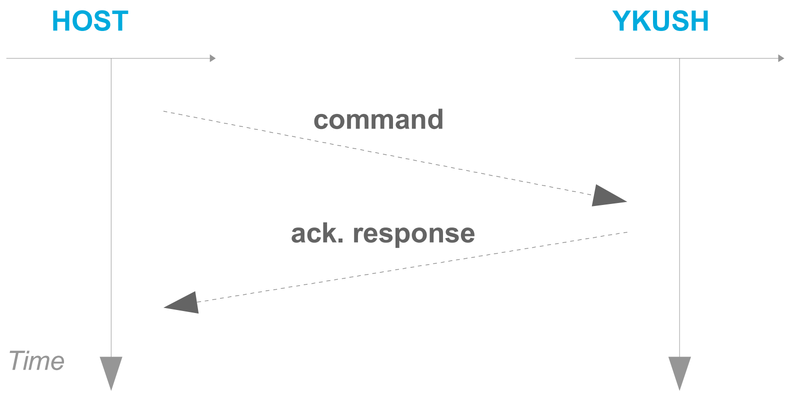
Vendor ID (VID):	0x04D8
Product ID (PID):	0xF2F7

IMPORTANT NOTE: Older versions of the YKUSH Board used a different PID, always check the PID of your board.

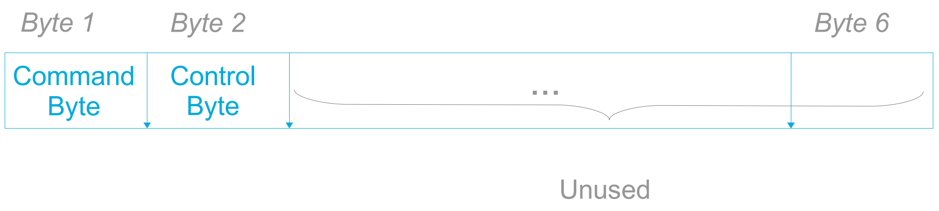
Using a library as libusb in Linux and with the Vendor and Product ID you can easily communicate with the HID device.

The communication protocol

The communication protocol between the host system and YKUSH on-board microcontroller is a simple six bytes packet base command/acknowledge interaction.



Both the command and the acknowledge response (ack. response) are six bytes packet with the following format:



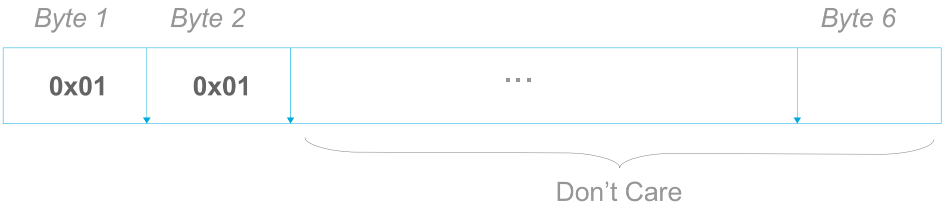
CONTROL PROTOCOL

The Command – From Host to YKUSH

As mentioned previously, the command sent from the host to the YKUSH microcontroller is transported in a six byte packet.

In this command packet, the **byte 1** carries the action code to be performed and the **byte 2** carries a control byte that is the repetition of the action code.

For example, the action code for powering down the Downstream Port 1 is 0x01. So the command packet sent by the host to YKUSH to switch off the Downstream Port 1 is the following:



All available action codes are listed in the table bellow.

Action	Byte 1	Byte 2
Power Down Downstream Port 1	0x01	0x01
Power Down Downstream Port 2	0x02	0x02
Power Down Downstream Port 3	0x03	0x03
Power Down All Downstream Ports	0x0A	0x0A
Power Up Downstream Port 1	0x11	0x11
Power Up Downstream Port 2	0x12	0x12
Power Up Downstream Port 3	0x13	0x13
Power Up All Downstream Ports	0x1A	0x1A

The Acknowledge Response – From YKUSH to Host

The acknowledge response message, sent from YKUSH to the host, is also a six byte packet with similar structure. In this packet the **byte 1** carries the 0x01 code, which signals a successful reception and processing of the command sent. In the **byte 2** follows the received and executed action code.

YEPKIT

YKUSH Yepkit USB Switchable Hub

Find more at www.yepkit.com

Version 1.1