YKUR - Yepkit USB Relay

Data Sheet

March, 2018

Contents

1. Overview

YKUR is a relay board, with one on-board relay and four additional relay driver ports, which can be controlled from USB or I^{2}C.

The two control interfaces, USB and I^{2}C, provide flexibility to the user which can select the interface most appropriate for the application. For example, in a use case where a single board computer (e.g. Raspberry Pi) is used to implement the control logic, that “needs” to switch something On/Off, the USB control interface may be more appropriate and easy to set-up. On the other hand, if the use case is an embedded system where the control logic is implemented in a microcontroller (MCU) then the I^{2}C control interface may be a better fit. The two control interfaces, USB and I^{2}C, can be used concurrently.

Regarding the USB control interface, the board is seen by a computer (or other USB host) as an HID device not requiring the installation of drivers. This feature facilitates the usage by making it “plug-and-play” on the USB interface.

When you connect it through USB, to a computer, it becomes available to be controlled by the *ykurcmd*[1] command line application. For details on the command line application please check the *ykurcmd user manual* available at the product page[2].

For I^{2}C control interface the board is configured as a slave device with 7 bit address `0b1010001`. By sending data bytes with a specific value through I^{2}C to a YKUR board you are able to command the actions of the board, as will be detailed in the sections further on.

The four relay driver ports, which can switch both mechanical and solid state relays (SSR), facilitate the scale-up of switching capability in your projects without requiring additional boards and controls.

1.1. Main features

- USB bus powered or externally powered using a 5Vdc power supply.
- One on-board relay which can switch voltages up to 277Vac and 30Vdc supporting currents up to 10A.
- Four ports for driving the control coils of external relays.
- Individual control of the on-board relay and each of the four external relay driver ports.

1.2. Applications

Target applications include the remote automatic control of connecting and disconnecting electrical signals and devices.

[1] Get the latest version of ykurcmd on the github.com/Yepkit/ykurcmd repository

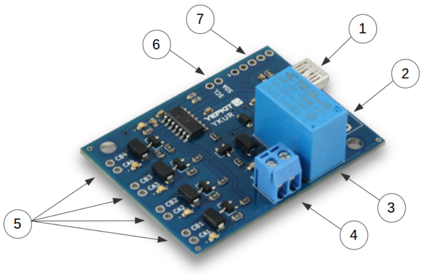
[2] www.yepkit.com/product/300106/YKUR

2. Description

YKUR is a USB and I^{2}C controllable relay board. It provides one on-board relay and four additional ports to drive the coils of external relays. The on-board relay can switch voltages up to 277Vac and 30Vdc being rated for current up to 10A. The four external relay driver ports can each support current up to 360mA at 5V, but at any given time the total board current is limited to 500mA.

2.1. Board layout

The board has one relay in-board and four additional relay driver outputs available to the user for controlling four external relays.



1. Mini USB port to connect to Host.
2. External power supply input.
3. On-board relay.
4. On-board relay contact terminal.
5. Driver ports for external relays.
6. I^{2}C pins.
7. ICSP pins.

When using the external relays care must be taken to ensure that the board power input is enough to support the additional current demanded by the coils of these external relays.

2.2. Board operation

The board is USB bus powered or externally powered by a 5V power supply. On power-up the on-board relay and all four external relay driver ports are in the off state by default. When powered the power a power-on indicator LED will be lighted on.

The on-board relay and external relay driver ports are controlled through the USB interface. An application on the USB host system can send commands to the YKUR HID USB device to control/switch the on-board relay and external relay driver ports individually. Check the *ykurcmd* application for an example of a control software.

The *ykurcmd* application source code is available for Linux and Windows systems. This code can be ported to other systems without much effort as it uses cross-platform USB libraries.

3. On-board relay and additional switching ports

3.1. On-board relay

The on-board high endurance mechanical relay can switch up to 30Vdc or 250Vac voltages, making it suitable to switch the power of typical consumer electronics, electrical systems and machinery. The main characteristics are the following.

Contact Data	
Rated voltage	30Vdc, 250Vac
Min. recommended contact load	100mA, 5Vdc
Electrical endurance	100x10^3 ops.
Mechanical endurance, DC coil	10x10^6 operations

3.2. Additional relay driver ports

4. USB control interface

The on-board relay and the four external relay driver ports are controlled through the USB interface. The board appears to the host as a HID device with:

- VID = 0x04D8
- PID = 0xF1CB

All the boards have unique serial numbers defined in the USB device descriptor iSerialNumber field. This can be used by the host to address a specific board when several YKUR boards are attached simultaneously.

The communication protocol uses a master/slave scheme where the USB host is the master and the YKUR board the slave. The communication is always initiated by the master by sending a command to the slave. The slave will then execute the command and respond/acknowledge to the master. The control protocol is based in USB HID 64 byte messages (HID reports).

4.1. Relay and port switching commands

The switching commands are sent from the USB host to the YKUR board and once is executed the YKUR will send a response message confirming the switching.

4.1.1. Switch On/Off command

Command from host:

Switching command is a 64 byte message where only the first two bytes are relevant, as detailed bellow.

Byte 1	Byte 2	...	Byte 64
action	target	...	don’t care

action - Action to be executed.

0x01 - Switch **On**

0x02 - Switch **Off**

target - Command target port.

0x01 - Port 1

0x02 - Port 2

0x03 - Port 3

0x04 - Port 4

0x11 - Relay

0x0A - Port 1, 2, 3 and 4

0xAA - All ports and Relay

Response from YKUR:

Following a command from the host application, YKUR board will respond with execution confirmation message. The response is the "echo" of the sent command plus a status byte.

Byte 1	Byte 2	Byte 3	...	Byte 64
action	target	status	...	don't care

action - Action executed.

0x01 - Switch **On**

0x02 - Switch **Off**

target - Command target port.

0x01 - Port 1

0x02 - Port 2

0x03 - Port 3

0x04 - Port 4

0x11 - Relay

status - Switching status.

0xFF - Ok

0xAA - Error

Note: A response message with value 0x00 in the first byte indicates that an unknown error occurred. In this case only the first byte is relevant.

4.1.2. Port switching status

Command to get the current port or relay state (On/Off).

Command from host:

Byte 1	Byte 2	...	Byte 64
0x03	target	...	don't care

Where:

target - Command target port.

0x01 - Port 1

0x02 - Port 2

0x03 - Port 3

0x04 - Port 4

0x11 - Relay

Response from YKUR:

YKUR board will respond with port status or an error.

Byte 1	Byte 2	Byte 3	...	Byte 64
0x03	target	status	...	don't care

Where:

target - Command target port.

0x01 - Port 1

0x02 - Port 2

0x03 - Port 3

0x04 - Port 4

0x11 - Relay

status - Switching status.

0x00 - Off

0x01 - On

0xAA - Error

Note: A response message with value 0x00 in the first byte indicates that an unknown error occurred. In this case only the first byte is relevant.

4.2. Status And Information Commands

4.2.1. Get Firmware Revision

Get the firmware version/revision command structure.

Byte 1	Byte 2	...	Byte 64
0x20	0x01	...	don't care

YKUR will respond to this command message with the following response message.

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	...	Byte 64
0x20	0x01	major	minor	patch	...	don't care

Where:

major - Major number of the revision

minor - Minor number of the revision

patch - Patch number of the revision

5. I^{2}C control interface

In addition to USB, I^{2}C interface can be used to control the YKUR. This is useful for use in a microcontroller based environment where typically I^{2}C bus is available and, being a two wire interface, of easy integration.

Sending a command to YKUR using this interface is accomplished by transmitting a single byte code addressed to `0b1010001`, which is the device 7-bit address for YKUR boards, with R_nW[3] bit cleared.

The step by step process of I^{2}C transmission of a command to YKUR, the slave device, is the following.

1. Master generates a Start condition.
2. Master sends 7 bit address followed with the R_nW bit cleared (0).
3. Master sends data byte with the code associated to a specific action/command.
4. Master generates Stop condition.

The following table shows the mapping between action and byte code value.

Command/Action	Byte Code
Switch ON relay	0x01
Switch OFF relay	0x02
Switch ON port 1	0x03
Switch ON port 2	0x04
Switch ON port 3	0x05
Switch ON port 4	0x06
Switch OFF port 1	0x07
Switch OFF port 2	0x08
Switch OFF port 3	0x09
Switch OFF port 4	0x0A
Switch ON all ports	0x0B
Switch OFF all ports	0x0C
Switch ON all ports and relay	0x0D
Switch OFF all ports and relay	0x0E
Get relay switching state	0x0F
Get port 1 switching state	0x1F
Get port 2 switching state	0x2F
Get port 3 switching state	0x3F
Get port 4 switching state	0x4F

An example of the control of the YKUR board through the I^{2}C interface can be found in the YKAMELEAN24 firmware available for download at the product page *www.yepkit.com/product/300105/YKAMELEAN24*. The control YKUR library can be found in the source code files `ykamelean24_firmware/src/modules/ykur/`.

[3] Refer to PIC16F1454 datasheet on “I2C Mode Overview”.

A. Development Support

The latest version of the ykurcmd utility source code is available at GitHub in ykurcmd Yepkit repository. It's distributed as open source under MIT license.

B. Revision History

1.2.1 I^2C control interface.
New control interface using I^2C for easier integration into microcontroller based environments.

1.2.0 Get relay and port switching state.

1.1.0 Configurable default states for Relay and Ports. NOTE: This functionality has been deprecated.

1.0.0 Initial release.