# USB2PPM

## User Manual

Version 2.4a

dated 04/02/23

Gregor Schlechtriem

webmaster@pikoder.com

www.pikoder.com

# Content

# 1

## *Overview*

The USB2PPM is a true single chip solution using a modern PIC controller to generate an PPM-Stream with a resolution of 1 µs controlled from a Windows Computer or an Android Smart device. This stream can be used by many RC transmitters for flying with a joystick or a gamepad by simply connecting the USB2PPM output to the trainer connector of the RC transmitter.

This User's Guide covers the features, the programming, and the command interface of the USB2PPM as well as common applications.

In section 2 you will find a brief description of key features, the overall function and an overview of the interfaces supported. It is recommended to carefully read this section to get a good basic understanding of the USB2PPM.

The next two sections 3 and 4 deal with the controller hardware. Section 3 provides the pinning and section 4 describes the reference schematic for the USB2PPM. **Please note that the applications and interfaces described in the following sections assume that the USB2PPM is setup in line with this reference schematic.**

Your next step is most likely to commission and test your USB2PPM. Rather than using the "bits and bytes"-serial interfaces directly, which are laid out in section 6, you may consider using the more sophisticated PCCpro (PiKoder Control Center) Windows 10 software with a graphical user interface.

Section 7 demonstrates how you would interface your USB2PPM to a Windows computer with a joystick or gamepad and Section 8 focusses on connecting the USB2PPM to an Android smart phone.

This User's Guide is based on the most recent hard- and firmware version 2.4 available for the USB2PPM and the related PCC programming software. Please check for updated information and new software releases on www.pikoder.com.

Hyperlinks were integrated into the text for convenience. You would also find all downloads referenced on the USB2PPM webpage.

Please share with me any comments, improvement ideas or errors you will find or encounter in working with your USB2PPM. I can be reached at webmaster@pikoder.com. Thank you very much!

# 2

## *Features*

This section will familiarize you with the feature set and provide a high-level overview of the intended use of the USB2PPM allowing you to customize the controller to your specific needs and requirements.

You can connect the USB2PPM directly to the USB port of a computer or via USB-OTG to an Android smart device. Especially the combination of the mobile device provides for a high degree of mobility.

The PPM stream generated by the USB2PPM is controlled via a command protocol. More specific, the USB2PPM features a two-way ASCII-Protocol designed to be used in combination with standard terminal programs such as (but not limited to) Tera Term and TTY.

In addition, various application specific parameters such as startup position after powering up, PPM polarity and number of channels can be adjusted to your specific needs and are maintained in a non-volatile memory.

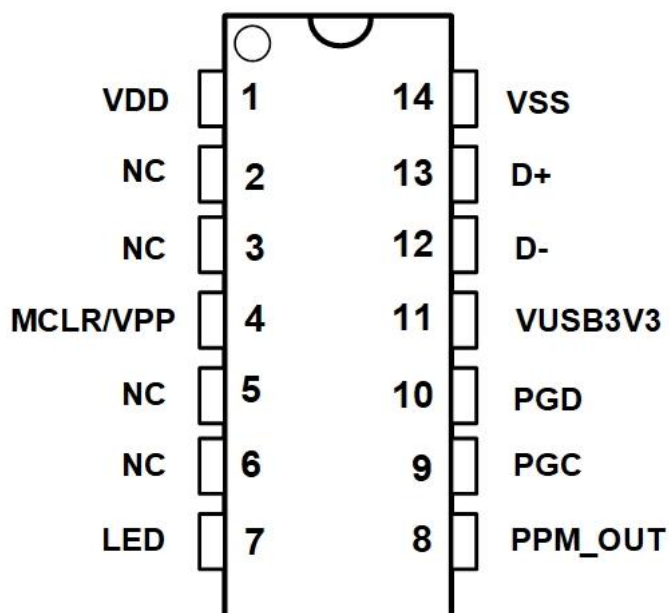The power for the USB2PPM is usually provided through the USB port.

The key features are:

- PPM streams for 1 to 8 channels with a resolution of 1 µs
- Android app as well as Windows app for joystick or gamepad control available
- operating voltage 5.0 V (provided by USB)
- non-volatile memory for application specific parameters
- bi-directional ASCII protocol confirming successful command execution
- support of the *CompuFly USB PPM converter*-protocol
- Programming application *PCCpro* provided for easy customization

# 3

# *Pin Description and Packaging*

The USB2PPM comes in a 14 pin DIP package (see below). The device operates from 3.3 – 5 Volts. Please refer to the PIC 16F1455 data sheet from Microchip (www.microchip.com) for complete electrical and physical specifications.

```
         _____∪_____
VDD     ⊏  1            14  ⊐   VSS
NC      ⊏  2            13  ⊐   D+
NC      ⊏  3            12  ⊐   D-
MCLR/VPP⊏  4            11  ⊐   VUSB3V3
NC      ⊏  5            10  ⊐   PGD
NC      ⊏  6             9  ⊐   PGC
LED     ⊏  7             8  ⊐   PPM_OUT
```

A complete description of the pins is given on the following page. If a different package would be required for your application, then please contact sales@pikoder.com for more information.

## Description of pins

| Pin | Symbol | Description |
|-----|--------|-------------|
| 1 | VDD | Supply voltage. Connect to 3,3 – 5 V DC |
| 2 | NC | Not connected (reserved for later use) |
| 3 | NC | Not connected (reserved for later use) |
| 4 | MCLR/VPP | Reset pin, active low. Connects directly to Vss for automatic reset at power up. |
| 5 | NC | Not connected (reserved for later use) |
| 6 | NC | Not connected (reserved for later use) |
| 7 | LED | LED indicator output (please refer to Appendix B for more information) |
| 8 | PPM_OUT | Output pin with PPM signal |
| 9 | PGC | Clock pin for In-Circuit-Serial-Programming |
| 10 | PGD | Data pin for In-Circuit-Serial-Programming |
| 11 | VUSB3V3 | Positive supply for USB transceiver; 470n capacitor required |
| 12 | D- | USB data line |
| 13 | D+ | USD data line |
| 14 | VSS | Ground connection |

# 4

## Standard application

The following schematic shows the standard application of the USB2PPM. Please note that the bi-color LED indicates the USB status and is optional.
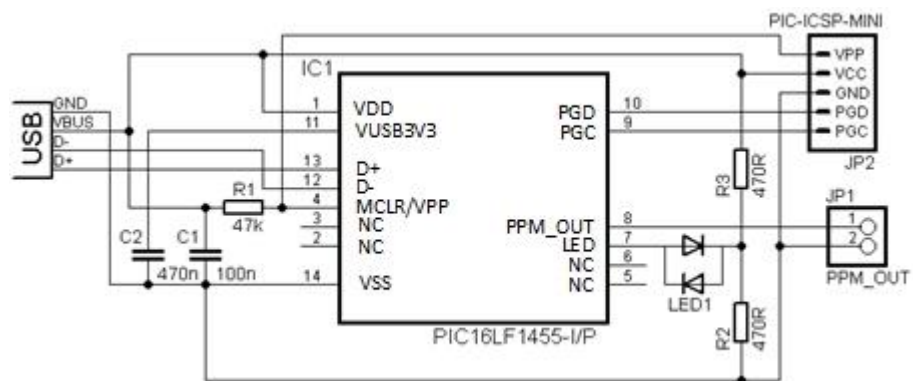


## LED status indication

This following table shows the status of the USB connection ("DEVICE STATE") based on the LED signaling state ("LED RESPONSE").

| DEVICE STATE | LED RESPONSE |
|---|---|
| DETACHED | OFF |
| ATTACHED | RED |
| POWERED | RED |
| DEFAULT | RED & GREEN |
| ADRESS | RED & GREEN |
| CONFIGURED | GREEN |

Given the simplicity of the schematic, the USB2PPM can easily be evaluated on a prototype board. If you are looking for a more permanent prototype then please consider the evaluation board which is available as a kit on **www.pikoder.com.**

**5**

# *The PiKoder Control Center PCCpro*

The USB2PPM's USB interface provides access to configuration options as well as support for real time control. The *PCCpro PiKoder Control Center* is a Windows 10 based graphical tool that makes it easy for you to use this interface. For almost any project you will start by using the *PCCpro PiKoder Contol Center* to set up and test your PiKoder. This section explains the features of the *PCCpro PiKoder Control Center*.

## Getting started

The hardware setup is simple and straight forward with a USB2PPM engineering board or your own prototype. You must connect your USB2PPM with the USB port of your Windows PC using a suitable cable. This cable will provide also for the power supply of the USB2PPM.

When you connect your USB2PPM for the first time, Windows will automatically install the drivers needed. On the engineering board the LED would be blinking red in fast mode during the installation process. Please refer to section 4 for more information regarding the LED indicator.

The PCCpro app is available free of charge in the Windows app store. It is highly recommended that you install the latest version of the program to enjoy the complete feature set of your USB2PPM. The PCCpro software is Open Source and released under a Apache License Version 2.0. The complete source is available through github.

After starting the PCC, you will see the window as shown on the following page.
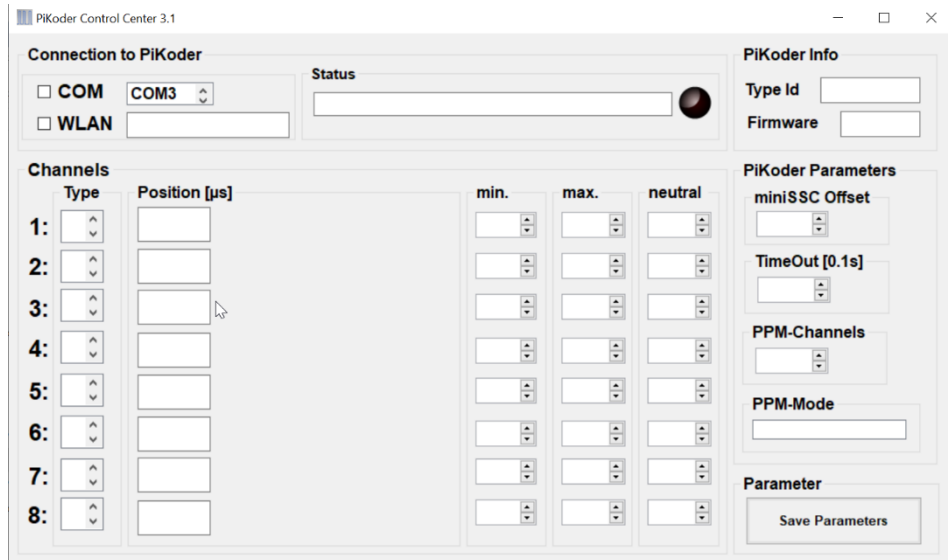
The *PCCpro* application would show COM3 to be the first available port in your computer. Select the COM port your USB2PPM is connected to (in this example COM6, see below) and then check the box in front of the COM port to connect.

The Status-LED on the screen would change to green for an online indication and the current parameters and settings of the USB2PPM would be loaded and displayed as shown on the following page. The firmware version of your USB2PPM would be displayed in the respective field.
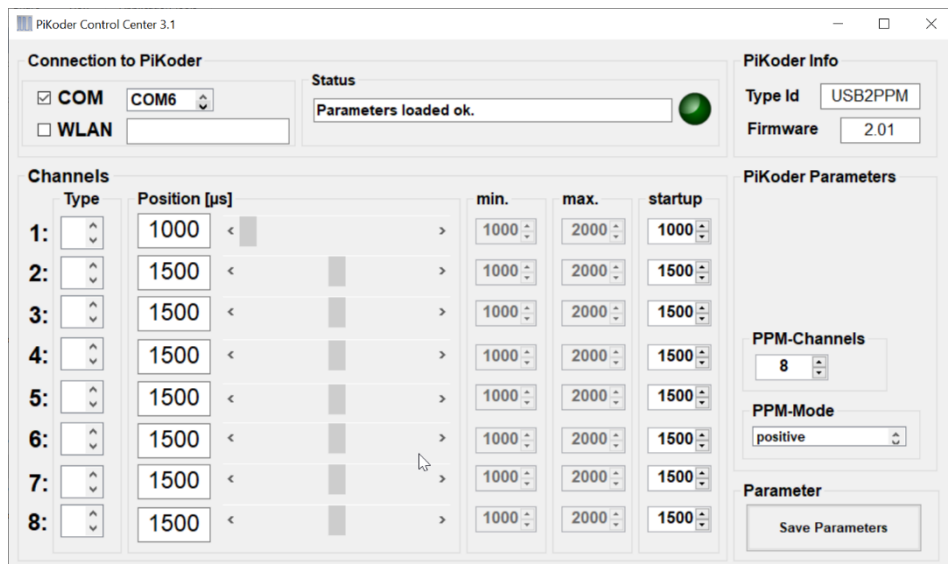


Please note that the layout of the form is adapted to the parameters and features available for the PiKoder firmware version found.

You would now have full control of your PiKoder: either for real-time control by the sliders or for changing the settings with respect to PPM polarity, number of channels and startup-value.

## Real-time Control

The sliders are used for controlling the USB2PPM's outputs and the respective numerical fields monitor the status in real time displaying the current channel value in μs. A separate row of controls is displayed for each of the active channels.

The key parameters for each channel such as min. and max. pulse width and neutral position can be set individually in the respective row. The PCCpro PiKoder Control Center will limit the slider value to the min. and max. value shown. This feature however is implemented in the Control Center. The PiKoder itself does not perform a parameter check and would therefore accept channel values outside of the shown boundaries.

## PPM Settings

The USB2PPM features either a negative or positive PPM stream. The polarity is related to the output of the PPM_OUT pin. The following picture presents the signals. Also, you can select the number of channels encoded in the stream. A range from 1 to 8 channels is supported; more than 8 channels can technically not be encoded in a standard 50 Hz PPM frame.
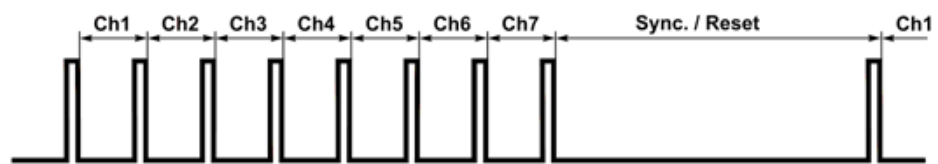


Above: "positive" PPM Frame at PPM_OUT (7 channels selected)
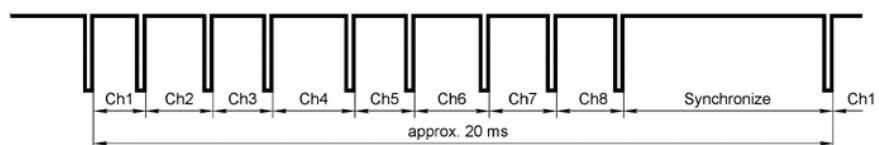


Above: "negative" PPM Frame at PPM_OUT (8 channels selected)

## Startup

In many applications the pulse width for the neutral value would be 1.5 ms. However, some ESCs (Electronic Speed Controllers) for drones might need 1.0 ms as a start-up value to make sure that they are not spinning immediately after turning the USM2PPM on. This configuration is shown in the screen print above with channel 1 representing *Throttle*. To address these

applications, you would set the initial value and safe the parameters by hitting the respective button on the lower right of the screen.

## Save Parameters

Changes made while using the *PCCpro* will not be permanent unless you select to save the parameters. Hitting this button transfers all settings into the non-volatile memory of the USB2PPM to be retrieved when started up the next time.

Please note that saving new values may take some time and requires disabling some internal interrupt logic. This may result in erroneous servo behavior.

# 6

## *Serial Interface*

The USB2PPM supports a two-way ASCII-Protocol named Ascii Command Interface (ACI) designed to support controlling PiKoders with standard terminal programs such as (but not limited to) Tera Term.

### ASCII Command Interface

The ASCII Command Interface (ACI) is probably the most versatile way to program any PiKoder without any specific host software such as the *PCCpro PiKoder Control Center*. All commands are simple ASCII and are sent using a Windows based terminal program such as Tera Term. The commands can be typed in right away and the response of the controller is readable without referring to any specific code tables. Please note that neither 'CR' nor 'LF' is needed to send the command to the controller.

There are two basic types of commands: commands for querying parameters and for setting parameters.

If a parameter is read the PiKoder will provide for proper formatting by sending a "CRLF" prior to sending the parameter value and support readability by sending another "CRLF" after the parameter value.

If a parameter is set the PiKoder will acknowledge the proper execution by sending an "!" framed by "CRLF".

If a command could not be interpreted at all then a question mark '?' framed by 'CR' 'LF' would be echoed. Please note that protocol syntax checking is extremely limited now.

The following ACI commands are available:

- '?': query the PiKoder type information; USB2PPM will respond in a format 'T=USB2PPM' framed by 'CR' 'LF'

- '0': query the firmware version; PiKoders will respond in a format 'n.nn' framed by 'CR' 'LF'

- 'i?': query the current pulse width for channel i (i = 1..8); PiKoders will respond 'CR' 'LF' 'xxxx' 'CR' 'LF' with xxxx representing the pulse width in μs

- 'i=xxxx': set the pulse width for channel i to xxxx μs (xxxx in decimal format, i = 1..8); PiKoders will acknowledge execution of the program by sending an 'CR' 'LF' '!' 'CR' 'LF'

**Additional commands as off release 2.1:**

- 'Nk?': query the pulse width for channel k after power up which might differ per application. PiKoders will respond 'xxxx' with xxxx representing the pulse width in μs (xxxx in decimal format, k = 1..8) - the command is not case sensitive

- 'Nk=xxxx': set the pulse width for the power up position for channel k to xxxx μs (xxxx in decimal format, k = 1..8); - the command is not case sensitive and the PiKoders will acknowledge execution with a '!' framed by 'CR' 'LF' - the command is not case sensitive

- 'P?': query the current PPM settings. USB2PPM will respond 'xy' with x representing the number of channels and y indicating the polarity with 'P' for positive and 'N' for negative - the command is not case sensitive

- 'P=xy': set the PPM parameters with x representing the number of channels and y representing the polarity of the PPM signal ('P' or 'N'). Please note that you always must enter both parameters. The command is not case sensitive and the USB2PPM will acknowledge execution with a '!' framed by 'CR' 'LF'

- SU]U], sU]U]: will save the current parameters to the controller's non-volatile memory making the current servo positions the start up positions after powering up; returns a '!' upon successful completion framed by 'CR' 'LF'

## CompuFly USB PPM converter interface

In addition, the USB2PPM supports the *Compufly v2* protocol. This command interface is comprised of three messages:

- 'S' d hhhh: set channel value for channel d (0x1.. 0x8) to '0xhhhh' (hex format with a range from 2000 … 4000 (actual channel value can be computed by dividing by 2.

- 'S' 0x15 0x00 i: set the number of channels for the ppm frame to i (default = 8). The USB2PPM supports from 1 ... 8 channels. Please note that a ppm frame with more than eight channels does not allow for a repetition rate of 50 Hz and is thus not supported. If the user would try to set the number of channels for the frame to a number > 8 then this request will result in a frame with eight channels.

- 'S' 0x16 0x00 i: set the polarity of the ppm signal (0 = negative value (default), 1 = positive value)

- Room for your notes -

# 7

## *Joystick2PPM for Windows*

Joystick2PPM uses a joystick, or a gamepad connected to your PC to control your RC TX through feeding a PPM stream generated by the USB2PPM PiKoder into the student input. When started the app will automatically connect with the USB2PPM and the first Joystick found. Your configuration will be saved and reloaded when restarting.
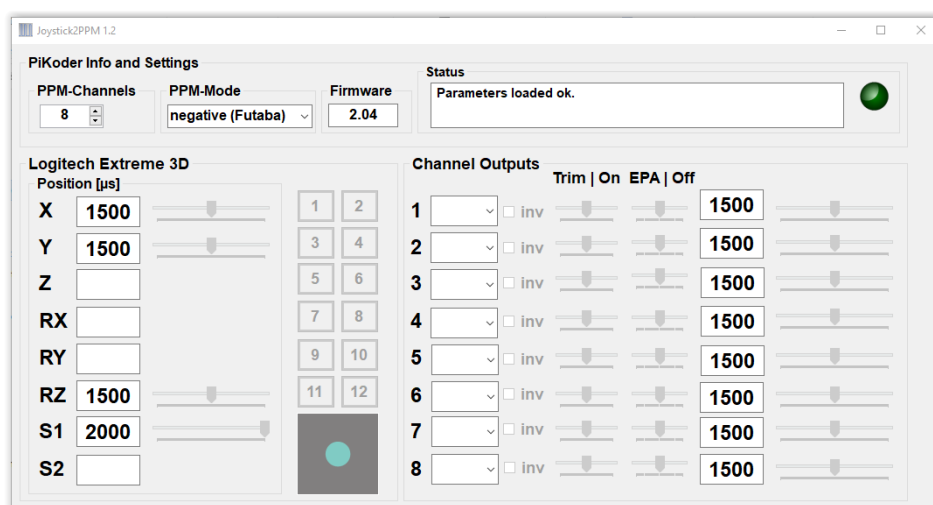


The user interface is intuitive and self-explanatory: simply associate joystick axes, sliders, or buttons to channel outputs. To accommodate the specifics of your application, channels connected to sliders and axis can be trimmed and the pulse range can be adjusted (EPA). Buttons can be turned into on/off switches by checking the "sw" box which replaces the inverse option after you assigned a button to an output channel.

The *Joystick2PPM* app is available in the Microsoft app store free of charge.

- Room for your notes -

# 8

## *Joystick2PPM for Android*

If you wanted to use a Joystick to control your model, then also an Android device could be deployed as a gateway. When using an Android device you would install the Joystick2PPM app which is offered free of charge in the Google Play Store. Additionally, you would need an USB hub. Please note that this app currently only supports a Logitech Extreme 3D PRO joystick and a USB2PPM.

## System setup



Setup for Joystick2PPM Android™ app

## User interface

The user interface is intuitive and allows for smooth operation.
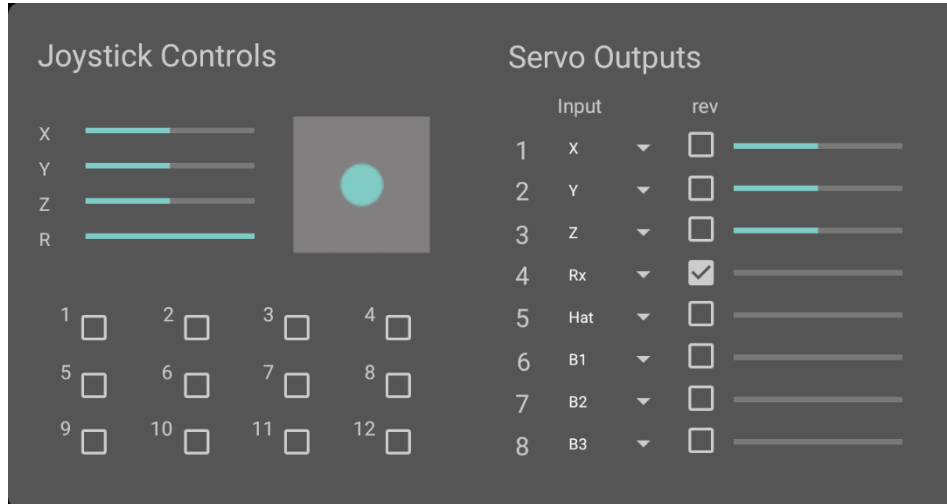


The joystick and the USB2PPM are automatically recognized when starting the app. You would have to grant access to the respective USB ports when running the app for the first time.