PropKeyer Construction and Usage Notes

This document presents a few notes on construction and usage of the PropKeyer. These should be useful to anyone who plans to build up their own version of this keyer.

1. Construction

Since it will more than likely occupy space at your operating position, picking the right enclosure can make the PropKeyer more convenient to use. You should select one that lets you spread out the controls in a convenient layout and, in particular, gives easy access to the speed control knob (which should be a fairly good size). I have found that I'm constantly adjusting the code speed to match the operator I'm communicating with, so I like a large knob which I can easily grasp. This is a good argument in favor of an analog-type control for the code speed adjustment, as opposed to typical all-digital systems where the speed is entered as a numerical value on a menu.

The enclosure pictured on this web site has a sloping front panel and measures 4 by 6 by 2 ½ inches. The major operating controls (there are only two: the mode selector switch and the speed control) were mounted on a piece of perforated circuit board, along with the power and “CW” LED’s and affixed to the front panel. This is the Top Panel Assembly – the circuit diagram is included on this web site as a .PDF file.

The loudspeaker and its volume control potentiometer were also mounted on a separate piece of circuit board and attached to the side of the cabinet. This is the Side Panel Assembly and its circuit diagram is also included on this web site.

The main electronics assembly is a Propeller Development Board, manufactured by Parallax, Inc. This board measures approximately 3 by 4 inches and is a good fit for the bottom section of the enclosure. The board is located so as to allow access to the DC power coax jack and the serial/USB programming pins through an opening cut into the rear apron of the cabinet.

The development board comes in two flavors – the only difference being that one has the serial-to-USB interface circuitry and USB mini-B connector included on the board, while the other, slightly less expensive version provides a simple 4-pin serial data interface to the outside world. A serial-to-USB converter for programming and data communication, called the Prop Plug, is also available from Parallax. A serial data interface is necessary to program the Propeller chip, and a USB connection is the easiest way to go. The same port can come in handy for displaying debugging data on a terminal or PC if the software to support this is added in the future.

The complete circuit diagram for the main electronics assembly is also included on this web site. Note that the Propeller chip is the only integrated circuit in the PropKeyer. All other components are passive devices that easily fit on the Propeller Development Board, or on one of the other sub-assemblies. The only modification to this board is to wire an external rocker switch to control DC power. A Schottky diode is also included in series with this switch to prevent accidentally applying power with the wrong polarity .

The following I/O connectors are located on the rear panel of the cabinet.

Function	**Connector Type**
Iambic paddle key	3.5 mm stereo audio jack
Straight key	3.5 mm mono audio jack
Keyer output	¼ inch mono 'phone jack
PS/2 keyboard	6-pin mini-DIN, female

These connect to the main electronics board via single-row multi-pin connectors. The cable from the iambic paddle key should be wired so that the dot paddle connects to the tip of the stereo audio jack. The dash paddle goes to the ring on the stereo connector and the common terminal to ground. Inside the PropKeyer, on the main electronics board, the connector for the iambic paddle key is non-polarized and is wired so that the dot and dash terminals to the iambic key can be reversed simply by reversing the connector where it plugs into the main board.

2. Usage

2.1 First Time Start-up Procedure

2.1.1 Loading the Software

The complete .SPIN code package for the PropKeyer is available on this web site as a .ZIP file. After downloading to your computer and un-zipping, it can be programmed directly into the Propeller chip. Notice that the .ZIP file contains four .SPIN objects:

FullDuplexSerial.spin,
Basic_I2C_Driver.spin,
Keyboard.spin, and the main object,
PROPKEY_100213.spin.

All four of these objects must be in the same folder on your computer. Then loading the *PROPKEY* file into the Propeller will automatically load the others.

2.1.2 Speed Control Calibration

Before the unit can be used, the speed control limits must be set up and stored in the on-board EEPROM. This is done very simply by first rotating the speed control pot to its minimum setting (i.e., 5 words per minute) and then pressing the pushbutton on the main board labeled "SC5". The "CW" LED on the Top Panel flashes briefly to indicate that the information has been read and successfully stored in the EEPROM. The pot is next rotated to the 40 wpm position and the button labeled "SC40" is pressed. Again, the LED flashes when the information has been read and saved. This only needs to be done once, when the unit is first placed in service, unless some of the hardware components which make up the speed control system are changed, or unless the EEPROM contents are erased. Normally, this cannot happen.

After this has been done, the unit is ready to use.

2.2 Using the Keyboard

The PropKeyer accepts a PS/2 computer keyboard via a standard 6-pin mini-DIN female connector mounted on the rear panel . When keyboard mode is selected characters are fully formed and transmitted by typing on the keyboard. The internal buffer only holds 16 characters, so a fast typist should be careful not to get too far ahead of the output stream. The dash-to-dot ratio, or “dash weight” is controlled by software running within the PropKeyer. Standard dash weight is 3:1, but this can be changed by the operator over a range from 2:1 to 6:1 by typing Control-w on the keyboard, followed by a single digit number between 2 and 6.

Inter-character spacing is also under software control and is normally set equal to seven dot periods. This can be changed in increments of 3 dot periods by typing Control-f and a single digit number from 0 to 9. This is known as “Farnsworth” spacing and is often used to increase intelligibility when slow code speeds are being used. Beginning code trainees who are trying to increase their copying ability may also find this spacing beneficial. With longer spaces between characters, individual characters can be sent at high speed, while the overall words-per-minute rate remains low.

A few keys on the keyboard have been re-interpreted to send common Morse code abbreviations as follows:

ESC key:	sends 8 dots (code for “error”)
Apostrophe key:	sends KN (go ahead, specific station)
] key (right square bracket):	sends SK (end of contact)
Control-a (or Control-A):	sends AS (standby or wait)