Telegram

User's Manual

1.0.0

Table of Contents

Introduction

Telegram is specifically designed for Telegraph displays (museums and the like). It runs full-screen and presents a (simulated) telegram form. Messages received from the wire are sounded and decoded and displayed in the body of the form. At the end of the message, or after a configurable amount of idle time, a new form is shown, ready to display the next message. In addition to receiving messages from the wire, messages can also be entered on a local key or with the keyboard. These are sounded, displayed, and sent out on the wire.

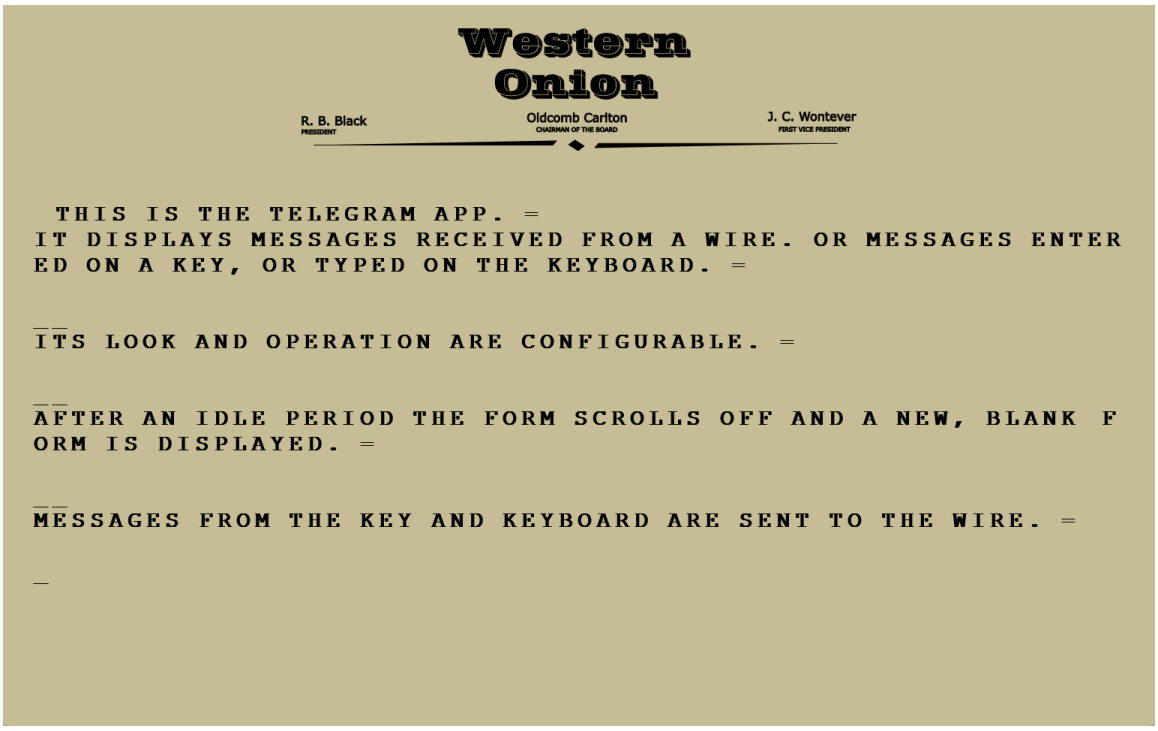

Figure 1. Telegram app screen

Preface

Telegram is a member of the MKOB Suite and uses the same configuration for the Morse, hardware, and other options, either Global or named, as MKOB, MRT, and the utilities do.

This user's manual covers the key and unique features of Telegram, while referring to other documentation for details on common functionality like creating and modifying configurations, selecting interfaces for the key and sounder, etc.

Related Components

pykob Package

Telegram uses the **pykob** package of modules. The pykob package consists of a number of modules that provide different aspects of handling code, the physical key and sounder, internet connection to a KOBServer instance, system audio to simulate a sounder or provide tone, configuration and logging capabilities, and more. When updating Telegram, it is advisable to get the latest (associated) release of pykob as well.

MKOB (Morse Key on Board)

MKOB is the full, graphical, user application. For most interactive use, MKOB is probably the application to use.

There is a full, illustrated, User's Guide that describes MKOB and its operation. Please refer to the **MKOB User's Guide** for descriptions of the core functionality, creating and using configurations, interfacing a key and sounder, using the *Virtual Closer*, and more.

MRT (Morse Receive & Transmit - Mr T)

MRT is a command-line/terminal application that is similar to MKOB in its ability to receive and send code. However, it is completely text based, allowing it to be run via a remote terminal connection to a system, or even automatically in the background (headless). It uses the same pykob configurations as MKOB (and the utility applications). In addition, it has a few options that allow it to support displays and similar applications.

There is a separate User's Guide for MRT.

Utilities

The pykob package also includes a number of utility applications. The README file included in the package contains a short description of each of the utility applications.

Quick Start

Get started with Telegram by using **MKOB** or **Configure** to set up your configuration. Copy the example `tg_config.tgc` Telegram configuration file from the Telegram Documentation folder to the directory that you are running Telegram from. Copy the `WesternOnion-Masthead.svg` file from the Telegram Documentation folder to the directory that you are running Telegram from. These two files configure the look and operation of the telegram form and provide the form masthead content.

Now launch Telegram.

Telegram accepts a number of command-line options. Use the `--help` option to get a description of the available options.

When run, Telegram will display incoming code as text. It accepts text typed on the keyboard and converts it to code, which is sent to the wire (if connected). It also listens to an attached key and sends the code to the wire, as well as decoding and displaying it. The Telegram configuration file **(tg_config.tgc)** is described in a following section.

User’s Guide

Telegram automatically connects to the wire when it is started and it remains connected while it is running. There is no option to connect/disconnect during operation.

The exception to this is if wire number 0 is specified. In that case, Telegram does not connect to a wire, and can be used for local (only) displays.

It monitors the wire and the local key. If the local key is opened, keyed code will be sent to the wire. While the local key is closed, Telegram will sound, as well as decode and display, code received from the wire.

Keyboard Commands

Telegram automatically connects to the wire, either from the configuration or specified on the command line. There are four keyboard commands:

1. '~' - Open the key.
2. '+' - Close the key.
3. ESC - Toggle the key closed state
4. Ctrl-X - Clear the form (scroll the current form off)
5. Ctrl-Q - Quit/Exit
6. Ctrl-C - Quit/Exit

If the key is open (either physically or by using '~'), other text typed will be sounded, displayed, and encoded and sent to the wire if connected.

(Simple, right!)

Exiting

Telegram runs continuously once started. To stop, enter either Ctrl-Q (^Q) or Ctrl-C (^C) on the keyboard.

Command-Line Options

Telegram is controlled by command-line options in addition to the MKOB/PyKOB Configuration being used (a command-line option allows specifying which configuration to use, rather than using the global configuration). It is suggested that the `--help` command-line option be used to get the up-to-date description of the options available. The following list is provided for convenience.

```
usage: Telegram.py [-h] [-a sound] [-A sounder] [-g gpio] [-P serial]
                   [-p portname] [-S station] [-c wpm] [-t wpm]
                   [--config config-file] [--logging-level logging-level]
                   [wire]

Telegram - Display a telegram form with local and received messages.
Telegram
specific configuration is in the 'tg_config.tgc' file.

positional arguments:
  wire                  Wire to connect to. If specified, this is used
rather
                        than the configured wire. Use 0 to not connect to
a
                        wire (local only).

options:
  -h, --help            show this help message and exit
  -a sound, --sound sound
                        'ON' or 'OFF' to indicate whether computer audio
                        should be used to sound code.
  -A sounder, --sounder sounder
                        'ON' or 'OFF' to indicate whether to use sounder
if
                        'gpio' or `port` is configured.
  -g gpio, --gpio gpio  'ON' or 'OFF' to indicate whether GPIO (Raspberry
Pi)
                        key/sounder interface should be used. GPIO takes
                        priority over the serial interface if both are
                        specified.
  -P serial, --serial serial
                        'ON' or 'OFF' to indicate whether a Serial
key/sounder
                        interface should be used. GPIO takes priority over
the
                        Serial interface if both are specified.
  -p portname, --port portname
                        The name/ID of the serial port to use, or the
special
```

```
                        value 'SDIF' to try to find a SilkyDESIGN-
Interface,
                        or 'NONE'.
  -S station, --station station
                        The Station ID to use (or 'NONE').
  -c wpm, --charspeed wpm
                        The minimum character speed to use in words per
                        minute.
  -t wpm, --textspeed wpm
                        The morse text speed in words per minute. Used for
                        Farnsworth timing. Spacing must not be 'NONE' to
                        enable Farnsworth.
  --config config-file  Configuration file to use. The special value
'GLOBAL'
                        will use the global (un-named) configuration. The
                        special value 'NEW' will use a new (defaults)
                        configuration.
  --logging-level logging-level
                        Logging level. A value of '0' disables DEBUG
output,
                        '-1' disables INFO, '-2' disables WARN, '-3'
disables
                        ERROR. Higher values above '0' enable more DEBUG
                        output.
```

Telegram Configuration File (tg_config.tgc)

While the configuration that you create and set up using **MKOB** or **Configure**, that is used by **Telegram**, controls the Morse encoding and decoding, what hardware you have and how it's connected, your Office/Station name, and other aspects of the operations, a separate configuration file, specific to **Telegram** controls the look and operation of the form.

The Telegram Configuration controls:

- Form (page) color
- Text font
- Text font size
- Text color
- Text bold (true/false)
- Text italic (true/false)
- Side (left/right) margin
- Top margin
- Masthead (top banner/graphic) file
- Masthead text (used if the masthead file isn't found)
- Masthead font (used for the text)
- Masthead font size
- Masthead text color
- Page (form) clear idle time
- Page (form) scroll/clear speed

The example `tg_config.tgc` file is in the Telegram Documentation folder

JSON Format

The Telegram Configuration file is a JSON format file, as used by JavaScript, Python, and many other programming languages. Familiarity with JSON can be helpful, but is not required to be able to create or modify the configuration to meet your needs. It is suggested that you start with the provided file, as that will give you everything you need, and you should be able to simply modify some of the values.

File is JSON Format Map/Dictionary

Below is an example file. The following sections will describe each of the value options. Although the example is provided here, it is recommended that you use the file provided as a starting point, as it is used for testing.

```
{
    "font":"courier",
    "font_bold":true,
    "font_italic":false,
    "font_size":30,
    "text_color":"black",
    "page_color":"(198,189,150)",
    "masthead_file":"WesternOnion-Masthead.svg",
    "masthead_font":"Vineta,Century725,Times New Roman,Times",
    "masthead_font_size":40,
    "masthead_text":"Western Onion Telegraph Co",
    "masthead_text_color":"(0,0,10)",
    "top_margin":30,
    "side_margin":40,
    "page_clear_idle_time":15.5,
    "page_clear_speed":1.2
}
```

font

Specifies the font to be used for the messages received or entered locally. This is a font name as found on the system. Many text editors and graphics applications provide a mechanism to see the various fonts available on the system. This entry can also be a comma separated list of font names. An example of that is used in the `masthead_font` entry in the file. When a list is used, the list is scanned from left to right and the first font found on the system will be used.

font_bold

Specifies whether the text used for messages should be bold. The value must be `true` or `false`.

font_italic

Specifies whether the text used for messages should be italic. The value must be `true` or `false`.

It is valid to specify both bold and italic as `true`.

font_size

Specifies the size of the text used for messages. The size is the font *point size*.

text_color

Specifies the color of the text used for messages. This can either be a standard system color name, or a RGB (Red,Green,Blue) value. For example, the values "black" and "(0,0,0)" result in the same black text.

page_color

Specifies the color of the page (form) background. See **text_color** for a description of how to specify a color value.

masthead_file

Specifies a graphic file to be used for the Masthead of the form. Testing is is done with SVG and PNG format files. If SVG is used, any text must be converted to paths in order to be displayed. If PNG is used, you will probably want to use a transparent background. Other graphics formats may work, but are not tested.

masthead_font

If a Masthead file isn't specified, or the file can't be found or loaded, it is possible to specify Masthead text to be used instead. This property specifies the font to be used. See font above for valid values.

masthead_font_size

Specifies the font size to use for the Masthead text. Not used if a Masthead file is used. See font_size above for valid values.

masthead_text

Specifies the Masthead text (content) to use if a Masthead file isn't specified, or the file can't be found or loaded.

masthead_text_color

Specifies the color of the Masthead text. See **text_color** for a description of how to specify a color value.

top_margin

Specifies the space, in pixels, between the top of the page (form) and the Masthead.

side_margin

The space, in pixels, between the left and right sides of the page (form) and the text.

page_clear_idle_time

Specified the idle time, in seconds, before the page (form) is cleared (a new form, including Masthead, is displayed). A fractional value is allowed.

page_clear_speed

Specifies the amount of time, in seconds, for the current page (form) to be scrolled off and a new page to be displayed. A fractional value is allowed. A value of 0 will simply replace the current page with a fresh one. A negative value will cause the current content to scroll down off of the screen rather than up.

Index

Colophon

This document reflects the features of Telegram Version 1.0.0 Edit date: 12-10-2024

This document is authored using **asciidoc**, and the pdf version rendered through **asciidoctor-pdf**, to create a manual that is more readable and more pleasing to the eye.

Dedication

The team would like to thank everybody who through comments, criticism and suggestions has helped to make the Telegram application and this document better and more usable.

Special thanks go out to Les Kerr, who’s vision and effort created the original MorseKOB application, and who initiated the Python version.