

Signals Guide

Introduction

Signals is an Android app that runs on phones and tablets and aims to automate starting signals for sailing races.

Signals has the ability to connect via USB to a microcontroller (eg. Arduino) or relay to control an electrical horn (or other electrical device).

The good news for sailing clubs is that the Signals app is free and the equipment needed to make the sound box to connect it to are inexpensive and simple to put together. The code for both the app and the microcontroller are open source and available at Bitbucket (see link in Technical details).

Technical Details

USB Serial Relays

The HW-667 USB Serial (one channel) relay can be chosen in the Setup screen under USB Connection. The HW-667 is a cheap and widely available USB relay. It may also be called the CH340 USB Relay (the CH340 is the controller chip on the relay).

Relays using the CH340 chip that have more than one channel should also work, but the app will only operate the first channel.

This type of relay recognises the commands

A0 01 01 A2 (hex) to open the relay and

A0 01 00 A1 (hex) to close it

Microcontrollers/Arduino

To use an Arduino, select "Generic One" in the Setup screen. The app can send simple commands over the serial USB connection. It sends "1" when a signal should be started and "0" to turn it off.

For automated signals, the gap between on and off again is 1.5 seconds. The app also has a manual “hoot” button for shorten course, over-the-line, general recall etc which keeps the signal “on” for as long as the button is pressed.

On the microcontroller a simple program is required to respond to the USB commands. The code for this program (called **main.cpp**) is open-source and available, along with the code for the app, at Bitbucket:

<https://bitbucket.org/Rotor-Rig/signals-mobile>

Sound Equipment

At the other end of the USB cable is the physical equipment making the sound signal. The app takes care of the signal automation and relies on the equipment to respond to the commands sent. Typically the equipment consists of:

- Arduino connected to a relay or standalone USB relay
- Car horn or other device (to alert sailors to the signal)
- Power supply or battery
- Box and wiring to contain the equipment

The power supply varies in cost - a mains supply is inexpensive (around £10) whereas batteries cost around £20 (depending on capacity).

The relay and horn are often £10 or less each. The microcontroller is between £10 and £25 depending which one you buy. The box to contain the equipment varies in cost from very little to £20 - £30 if it is waterproof.

NB The clone Arduinos are normally cheaper than the originals. This is because they use a different USB to serial chip called a CH340. The app should work with both original and clone Arduinos.

Timing Accuracy

The app uses a timer that checks the time every 300ms and will only make automated signals if the time is within -150ms and +1000ms of the requested signal time. In practice this means the signals are usually within a quarter of a second of the set time and that countdowns can be stopped and started reliably in the middle of a sequence.

On the countdown screen the app records both the actual start time (to the ms) and the latency between sending a request to begin a signal and the response to that command, which is usually between 50 and 100ms.

Timing Sequences

There are three sequences that can be chosen - and the choice is remembered by the app. They are:

- 5-4-1-Go (5 minutes between starts)
- 10-5-Go (5 minutes between starts)
- 3-2-1-Go (1 minute between starts)

The first two are typically used by sailing clubs for starting races and the third option is for training or special events when signals every minute are required.

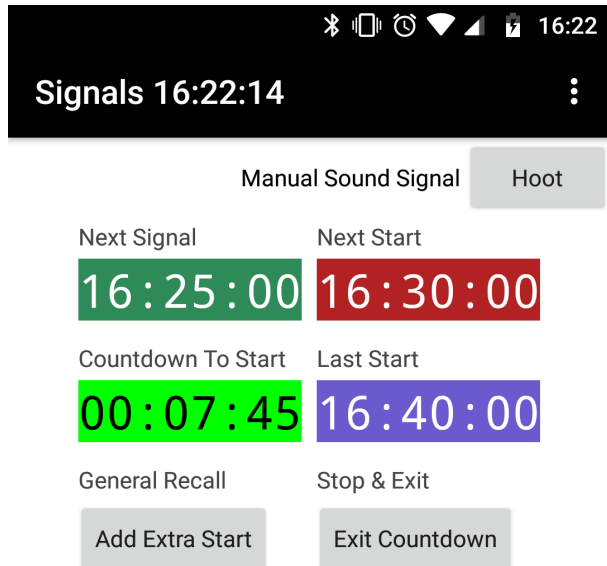
To set up Signals you need to specify the time of the first start, the sequence you are using and the number of starts required. You also need to connect the sound equipment and make sure it is working.

The setup screen looks like this:



If the USB connection is not active, the Signals app will make the sound of an air horn by playing an mp3 file - this is for your testing purposes. For use in a race, the equipment must be connected via the USB cable and the Connect button used to set the status to "Connected".

Once **Start Countdown** is pressed the countdown screen is shown:



USB: Waiting to connect...

USB Settings:

Sequence: 5-4-1-Go

Actual Signal Time:

Latency:



At the top right is the **Hoot** button to make extra sounds for shorten course etc. This can also be used to test if the system is working.

The **Add Extra Start** button increases the number of starts used in the countdown without having to go back to the setup screen. This is most likely to be used for a general recall. You can use this button either during the countdown or after the countdown has finished. If a general recall is required but the new start is not following on in the current sequence, simply return to the setup screen and begin a new sequence of signals as required.

Acknowledgements

[Bitbucket](#) - repository service

[App Inventor](#) - app design system

[Arduino](#) - microcontroller code and designs

License

This code is open-source under an [MIT License](#) and can be copied and used for free by anyone. Here is the official license wording:

Copyright 2021 Ian Cherrill

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

support@rotor-rig.com

Ian Cherrill

Rotor-Rig.com

2021