Loch Lochy Block Instruments & Auto Ken

My layout is a branch line terminus which can be operated by two or more people but most of the time it is just myself. I wanted to employ block instruments and token exchange equipment for realism, but I needed a way to operate them when I was by myself.

The solution was to create a device that uses Arduinos (a small microcontroller) to listen for incoming block bells, recognise the pattern of beats, and respond accordingly, either by responding with bells or issuing a token. The device was christened Auto Ken after a railway signalling engineer friend.

The first requirement to make this work was to create a web-based application, with an MS SQL Server database back-end, to store routes, timings between stations, timetables and bell codes. Further information was required to make this work, such as train classes, so timetables could be built up from individual routes. A further complication was that I needed this to work either when following a timetable or when I was just running trains "spontaneously". Therefore, there are two modes – Timetable Mode is based upon timings that trains are attempted to be run against, whereas Spontaneous Mode uses received bell codes to respond with appropriate trains chosen from what is available within the routes.

It then occurred to me that a further enhancement would be to add train describers as all the information required was in the database. A web page was created to resemble the old-style information systems of a black screen with yellow text. Raspberry Pis acting as web browsers in kiosk mode with small 7" LCD touch screens are used as dedicated displays at each station.

The next relatively easy step was to add automatic audio train announcements at the stations. Because the station information for the route was in the database, along with the scheduled time, audio announcements could be created by combining individual audio files of stations, times, and phrases. This amounted to over 200 separate audio files. Announcements can be chosen to be made before a train departs, when a train is shortly arriving, after a train has arrived and when a train has been delayed or cancelled. In addition, random audio announcements can be made to remind people to not drop litter or leave luggage unattended etc. Reason for delays announcements are chosen at random and if a delay lasts more than a few minutes then the reason for the delays will randomly change!

A version of electric token block instruments were provided at each station as that was prototypical for the West Highland Line. Again, Arduinos are used along with ethernet shields and small LCD displays. Each block instrument retrieves train and timetable information from the database via their ethernet connections, but bell rings are transmitted via dedicated wires to the next block instrument.

To issue a token a request is made by a block instrument by sending a bell code, e.g., 3 -1. The receiving box would then repeat the code 3-1 but hold the bell down for the last beat. This flashes the Token Issue button in the originating box and by pressing the button they can accept the token. This also allows the signals to be cleared if interlocking is enabled. Holding down the plunger for the last beat is prototypical but physical tokens are not issued, rather they are electronic and shown by indications on each block instrument.

A 2-2 bell code is sent either manually (or automatically from Auto Ken) when the train is entering the next section, and this is indicated on the display and by the illumination of the appropriate lamps on the block instrument.

When the train arrives the box sends a 2-1 bell, holding down the last beat, which allows the Token Return button to flash. Pressing the button replaces the token and completes the move.

The bell sounds are mp3 files played out by the Arduinos which allows different bell sounds to be used and for the volume to be changed.

All the bell codes and token exchanges are logged by the database which is useful when it comes to diagnosing any issues which may occur.

When the layout is operated by one person Auto Ken acts as a block instrument. It understands bell codes and, because it is also connected to the database via the network, it knows whether it is working in Timetable or Spontaneous Mode. In Timetable Mode it will act autonomously to initiate bell sequences and token requests based upon time information. In Spontaneous Mode it can either initiate a bell sequence when a route is set, when the Train Down button is pressed, or it can automatically respond to an Up Train request when call attention (1 bell) is sent from the other box. In this situation Auto Ken will identify a suitable route based upon information in the database and the bell code, and this will trigger the train describer displays and the audio announcements. In spontaneous mode the only intervention required for Auto Ken is to choose the class of Down Train. This is done via a rotary encoder on its front panel.

The audio announcements system works via a small application that runs on a PC. The layout is controlled using TrainController software, so a PC is always on when operating the railway. The application, which sits in the system tray, is written in C# and also connects to the SQL database to retrieve train information. Individual audio files are stitched together “on the fly” to create complete announcements – in much the same way as can be heard today on real stations.

This system has increased the realism of the layout by adding prototypical bell and token exchanges even when operating solo. It has also provided train information describers and audio station announcements which are generated simply by initiating a bell exchange. I believe it demonstrates how modern technology can greatly add enjoyment to the hobby.

Please watch and listen to the video https://www.youtube.com/watch?v=yQx-xu10t68