

The TECO LED Matrix Protocol Specification (v1)

Contents

1 Introduction.....	1
2 Server states.....	2
3 Communication.....	4
4 Known Issues.....	5
5 Schedule.....	6

1 Introduction

This document describes a simple communications protocol used to send data to the TECO 24x24 LED display matrix built into the Connection Machine¹ in the Infobau². The display is controlled by a Raspberry Pi and communication is done via Bluetooth.

The display consists of four 12x12 LED matrices that act as one big 24x24 LED matrix (Figure 1). They have been built using multiple RGB LED strips purchased from Adafruit³.

The Raspberry Pi used is a Model B with 512 MB of RAM with a low-cost Bluetooth dongle⁴.

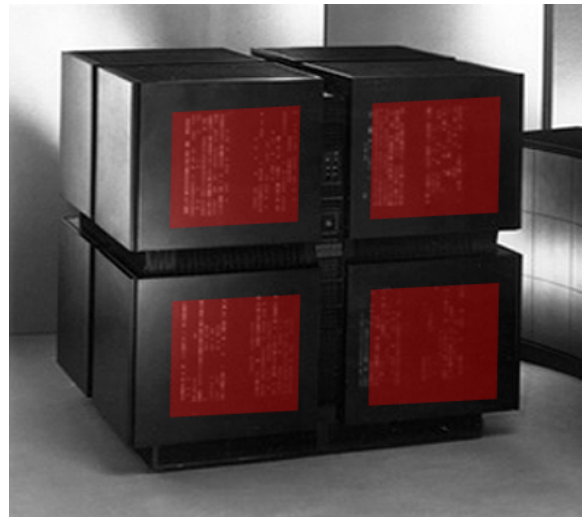


Figure 1: The Connection Machine. The position of the 12x12 LED matrices has been marked in red.

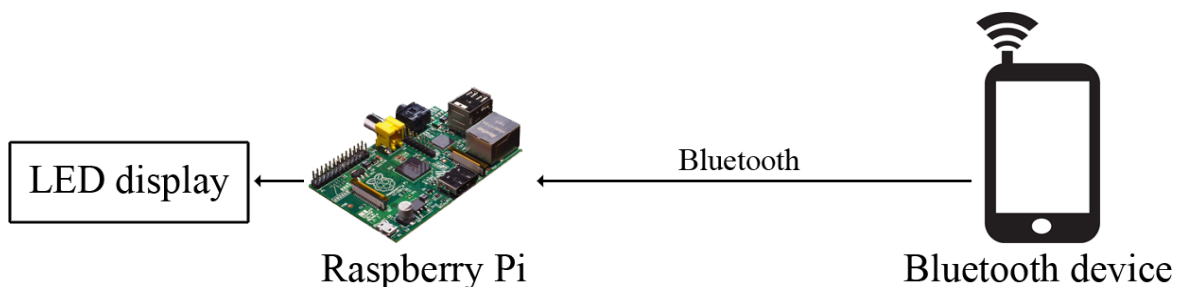


Figure 2: Schematics of the set-up.

1 https://en.wikipedia.org/wiki/Connection_machine

2 Building 50.34, 2nd floor

3 <http://www.adafruit.com/products/306>

4 BIGtec Nano micro Bluetooth Mini USB Adapter Stick Dongle Class2 EDR V2.0

The protocol was developed for use by students who were beginners in Android development for the most part, so the focus is on ease of use rather than speed and efficiency.

2 Server states

The server running on the Raspberry Pi has three states: **Idle**, **Connected** and **Running**.

When the server is in the **idle** state, random red patterns will be shown on the display. This is to mimic the behaviour of the Connection Machine⁵. A client can now connect to the server.

Upon connection, the server will be in the **connected** state. The server is then waiting for the client to initiate the handshake by sending the following packet:

1 byte	1 byte	1 byte	1 byte	1 byte	n byte
Version	X size	Y size	Color mode	Name length	App name

Version: Protocol version number. This document describes version 1, so the value should be *0x01*.

X/Y size: The size of the display. Should be 24 (*0x18*) for both values when using the TECO LED Matrix in the Connection Machine.

Color mode: Set to *0x00* to use only red, *0x01* for green and *0x02* for blue. Full RGB is currently not supported.

Name length: The length of the app name in bytes/ascii characters.

App name: The name of the app. This will be logged by the server along with the connection time and client MAC.

After receiving this packet, the server will send a two byte response:

1 byte	1 byte
Response code	Speed

The response code can have the following values:

0x00: No errors. Connection established.

0x01: X size not supported.

0x02: Y size not supported.

0x03: Color mode not supported.

0x04: Invalid name (e.g. length zero)

0x05: Other (unspecified) error.

5 <https://www.youtube.com/watch?v=pdO712Cshz4> (A video explaining the idea of the Connection Machine)

If there were no errors, the server will now be in the **running** state. If there were errors, the server will stay in the connected state.

The second byte („speed“) is the maximum framerate supported by the display. For example, if the value is `0x0B`, you may send up to 12 data-packets per second.

Once one client is connected, the server will refuse all other connection attempts. After the connection is closed, a new client may connect. The server will also terminate the connection if the client remains inactive for 10 seconds.

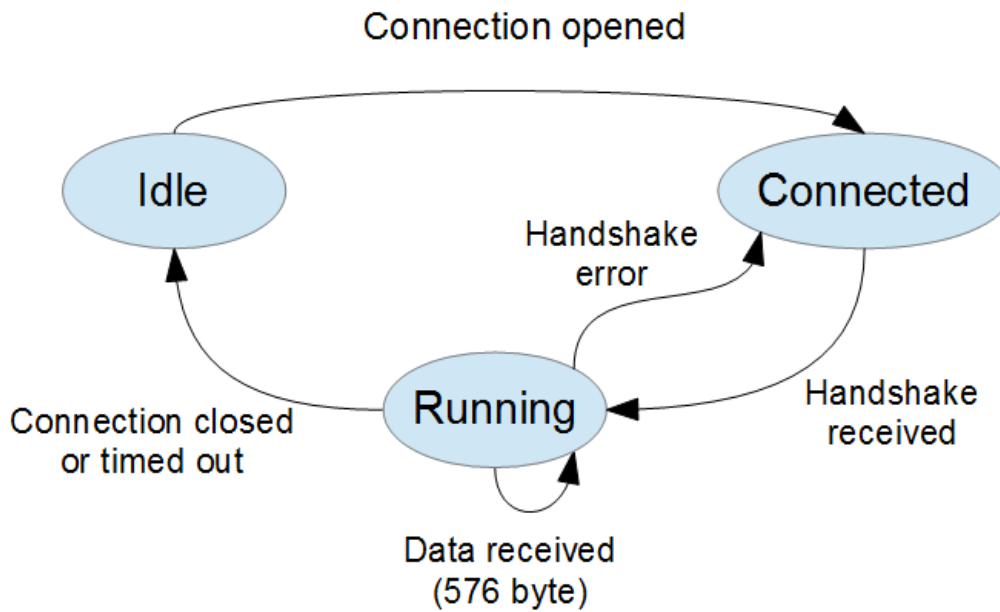


Figure 3: The state diagram of the server.

3 Communication

Communication with the LED matrix consists of 5 steps:

1. Find MAC address

You may only connect if the device is already Bluetooth-paired with the server. To make sure you are paired, check the list of paired devices for the entry „ledpi-teco“. If it's not there, ask your user to pair with the device „ledpi-teco“ before starting the app. Even if we change the Bluetooth dongle at some point in the future, this name will not change.

Once the devices are paired, you can get the MAC address of the server from the Bluetooth-paired-devices list from within your app.

2. Opening the socket

Open up a RFCOMM socket to the server⁶. The server listens on RFCOMM channel/port 16.

3. Initiating handshake

Send the initial packet as described above. If you receive a non-zero response code (error), the server will stay connected and you may try to send the packet again with different parameters or terminate the connection.

If the response is zero, you may start sending data.

4. Sending data

You may now send data to the server. Packets consist of 576 byte, each byte representing the brightness of one LED, ranging from 0 (LED off) to 255 (max. brightness). Please refer to Figure 4 to see how the data should be formatted.

When the packet is received, the data is almost immediately (~15 ms delay) shown on the display.

You may send up to X packets per second, where X is the speed-value received from the server. If you send any faster than this, the server will not be able to display your data on time and start buffering. When the buffers are full, the connection will be terminated.

When not sending a packet but still keeping the connection open, the display will show the last received data until a new packet arrives.

5. Closing the socket

After sending the data, close the socket so a new client can connect. The server will then go back into the idle state and listen for new connections.

The server may also close the connection if no new packet is received for 10 seconds.

⁶ If you're writing an App for an Android device, please note that the Bluetooth API does not appear to be functional in recent versions so a small hack has to be used to open the connection. Please refer to the provided reference implementation for more details.

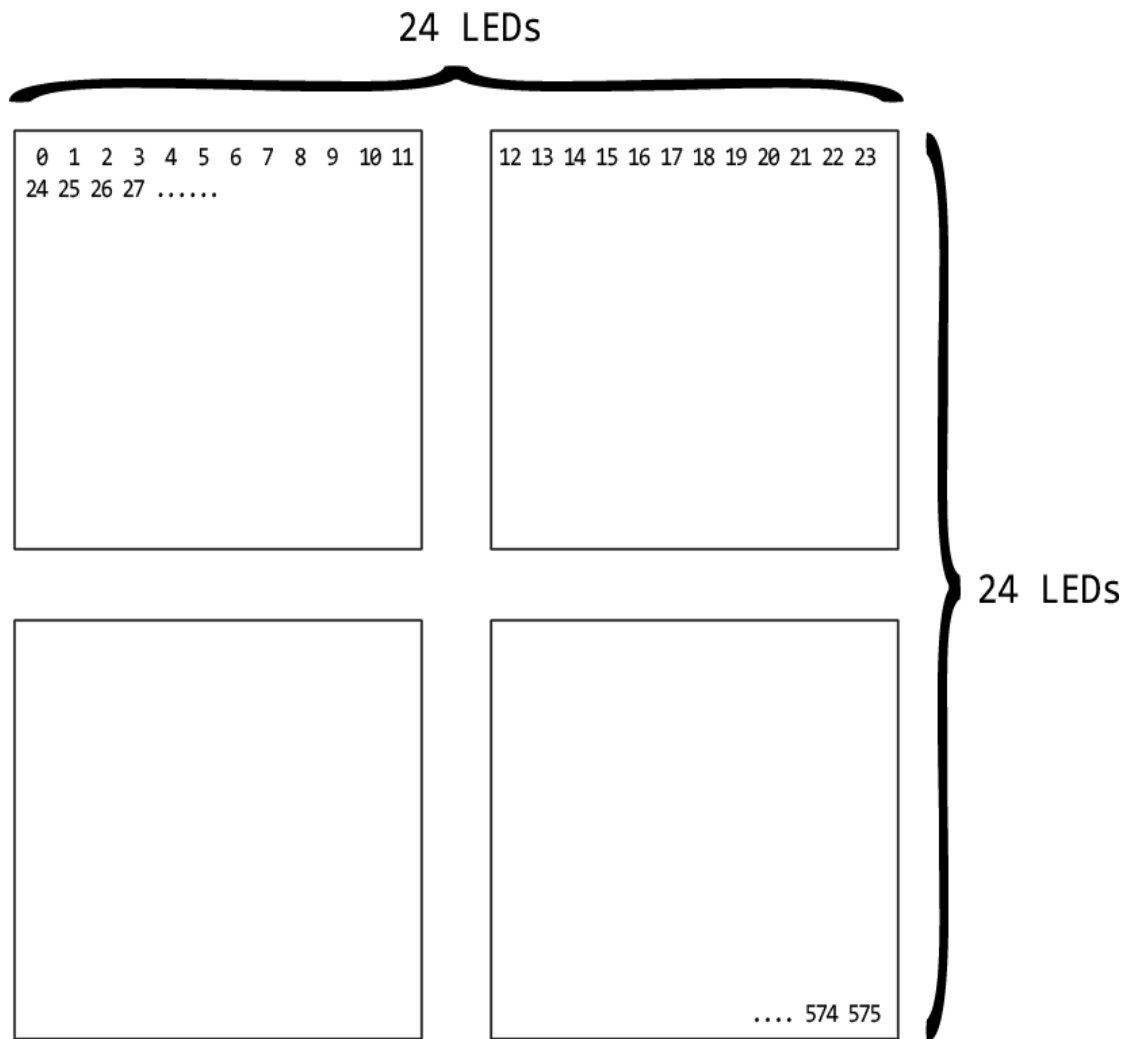


Figure 4: The packet should consist of 576 bytes. The figure shows where each byte in the packet will be shown on the display (physical location). For example, the first byte in the packet will define the brightness of the top left LED and the byte at offset 575 will define the brightness of the bottom right LED. Data after offset 575 is discarded by the server.

4 Known Issues

- Testing has shown that the connection cannot handle more than 10 packets per second. This is likely to be caused by hardware limitations of the very low-cost Bluetooth adapter. In theory, it should be possible to send much more data⁷.

⁷ http://en.wikipedia.org/wiki/Bluetooth#Bluetooth_v2.0_.2B_EDR

5 Schedule

- 4.6. - 5.7. **Testing phase.** The LED display matrix will stay at the TECO for a month. You may use this time to test your app. If there are any problems with the server, please contact diener@teco.edu.

Do not turn off the Raspberry Pi! It should stay online at all times.

From 7 AM to 9 PM, random patterns are shown on the display. After that, those patterns aren't shown but you can still connect to the display and send data. If the server crashes, it will automatically be restarted (might take up to one minute).

- 1.10. **Installation.** The LED display matrix has been installed into the Connection Machine.

We will be happy to publish your Apps on the TECO website. If you have written an App and want us to publish it, please send us the code, a short description and a couple screenshots.

- 20.10 The LED display matrix is now inside the Connection Machine and open to the public. Apps can be downloaded from teco.edu/cm.