

Controlling the R31JP Kit from a PC Web Browser Using a Serial Link

Veloria Pannell, Dylan Brooks, Eric Ponce

August 2024

1 Overview

This tutorial will show how to configure the R31JP Kit to respond to user input from an HTML webpage using Chrome's Web Serial API. Refer to the "Controlling a PSoC Big Board from a PC Web Browser Using a Serial Link" manual for the complete context and documentation for the Web-Serial interfacing. The provided webpage allows students to input ASCII values on a PC to change the value shown on the R31JP's Port 1 LEDs. Similar to using Tera Term, any character typed into the text box will be sent to the peripheral. The assembly code for the R31JP kit updates Port 1 and returns the characters to be echoed in the webpage's Device Log. The assembly and webpage files can be found on the course website.

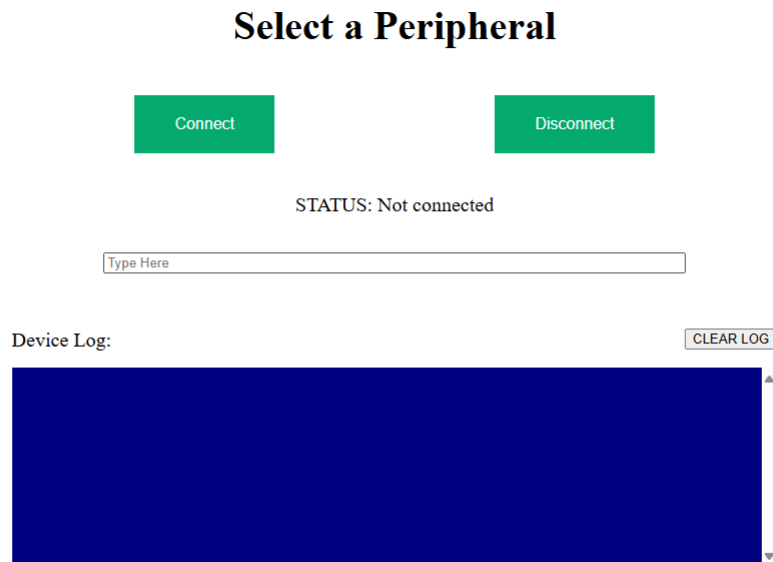


Figure 1: The webpage

Use the two buttons at the top to Connect or Disconnect with the R31JP kit. Type into the text box to send data; watch the LEDs on the R31JP kit to see the successively received ASCII value on the kit's P1 LEDs. Returned characters appear in the webpage's Device Log; clear the Device Log with the 'CLEAR LOG' button.

2 The Assembly

The assembly file consists of an initialization sequence in which the timer and serial registers are configured to communicate using serial communication at a baud rate of 9600. After the initialization sequence, the program consists of a single main loop in which it awaits a received character on the serial line. After this character is received, it moves the value to the P1 register to display the binary value on the R31JP's LEDs. This repeats indefinitely.

3 The Webpage

Once you have uploaded your assembly to the kit, you can connect to the webpage. Click on the 'Connect' button and select the correct device to establish a serial connection. For greater detail on webpage interactions, refer to the Big Board Demo's manual.

Note: you must be disconnected from Tera Term to connect to the webpage



Figure 2: Connecting to the Kit: 1) click on 'Connect' 2) Select the device port

Now, you can type into the input text box and watch the LEDs on the R31JP change. Echoes returned from the kit will appear in the Device Log beneath it. The LEDs will light up with the binary value corresponding to the user-entered ASCII character.



Figure 3: Type chars to send them to the R31JP; see echoes in the Device Log

For example, figure 4 below shows sending 'a,' 'b,' and 'c' to the R31JP. The JavaScript uses Chrome's Web Serial API to handle user input (explained in depth in the Big Board Demo). Upon entry of any character, the PC sends that ASCII value to the R31JP.

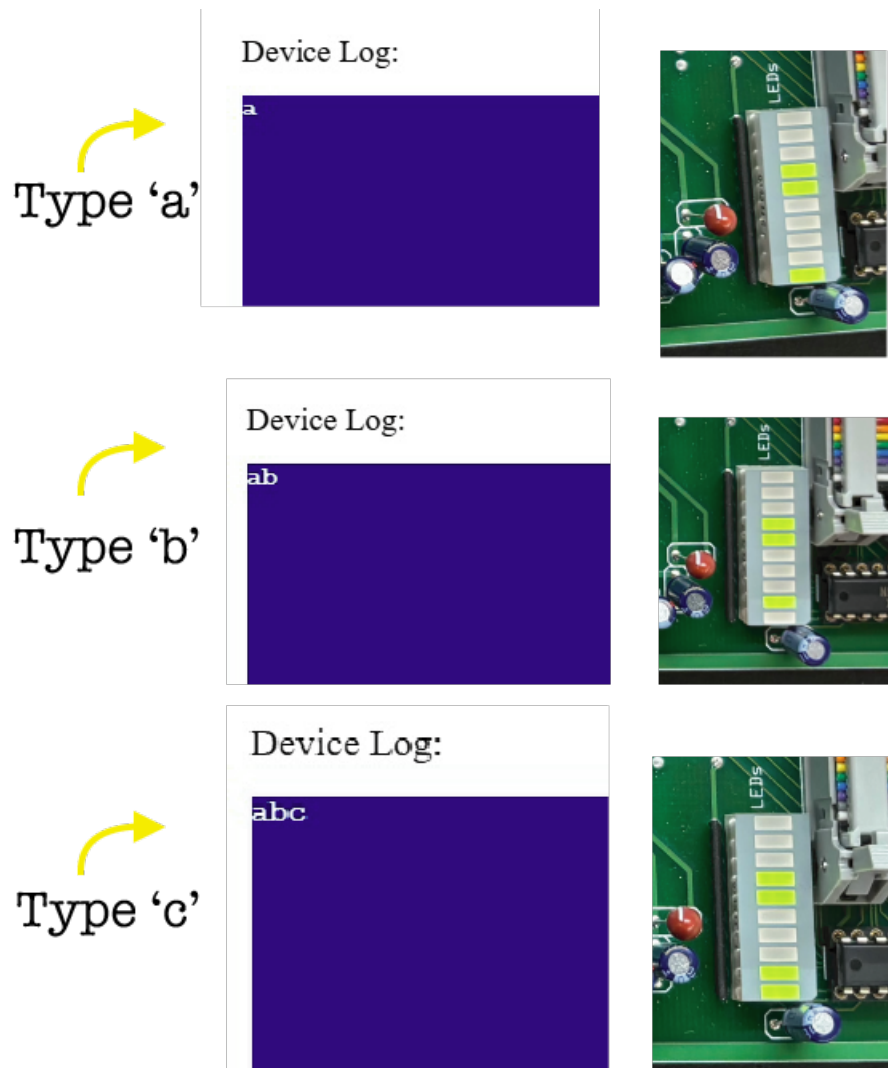


Figure 4: Sending 'a', followed by 'b', then 'c'

In the JS code, the API constantly reads returned ASCII values and displays them in the Device Log, where the characters accumulate; empty it using the 'Clear Log' Button.



Figure 5: Clearing the Device Log

4 Appendix

4.1 Lazerdillo

This appendix describes how the Lazerdillo webpage variation communicates with the R31JP kit; the webpage and assembly files are available on the course website. In this version, the webpage sends chars a-g depending on the the interaction with the webpage to light up specific values on the LEDs, indicative of various lissajous figures and rotations.

For this demo, the LEDs will light up with a certain pattern depending on the shape and rotation sent from the webpage. The set of LEDs is divided into an upper and lower nibble; the upper 'bits' will hold the shape, and the lower 'bits' will hold the rotation.

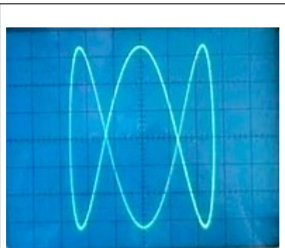
Lazerdillo

Connect


STATUS: Not connected

Disconnect

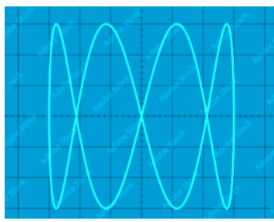
Select a Shape:



TIE Fighter



Circle



$a = 1, b = 4, \phi = 0$

Rotation Speed (Hz):

0 Hz

1.5 Hz

Value: 0 Hz

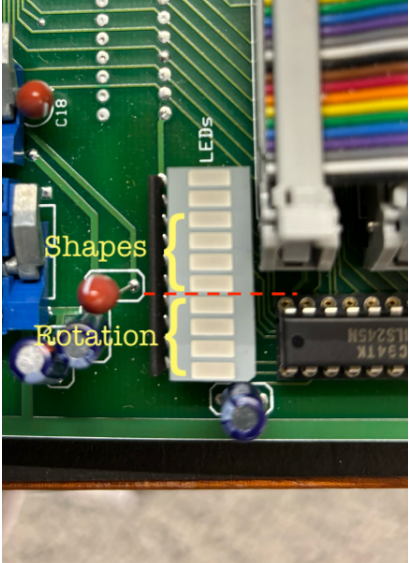
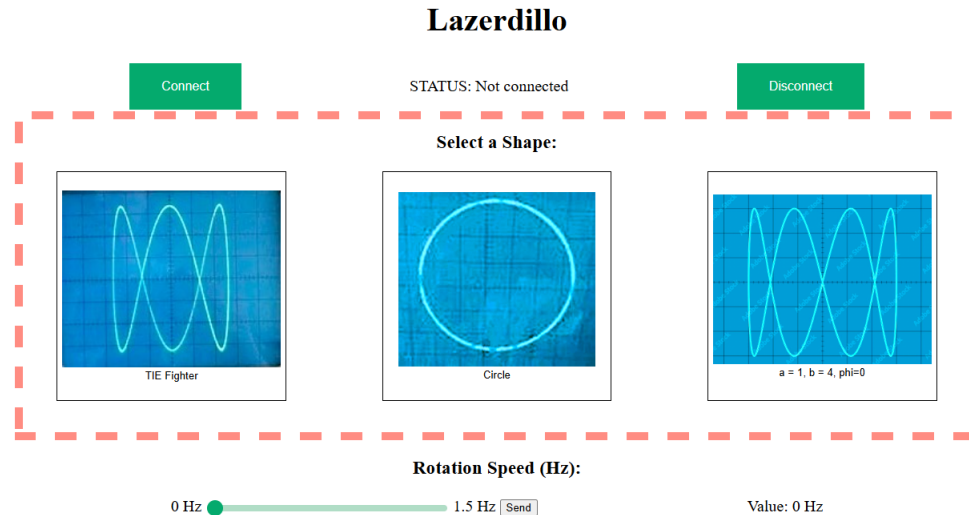


Figure 6: Upper Nibble for Shape; Lower Nibble for Rotation

1. Sending a shape:

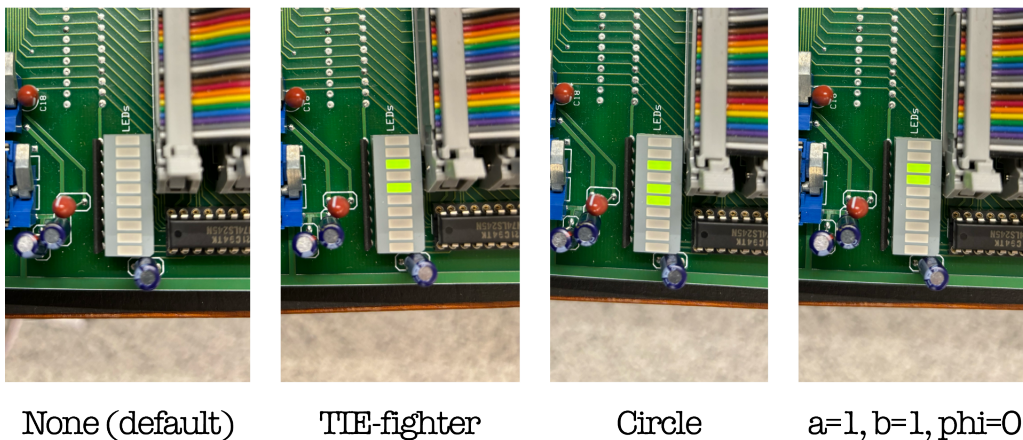
By default, all the LEDs are off to show no selected shape or rotation; each of the three large buttons will send a different char to the R31JP kit: the TIE Fighter will send “a,” the Circle will send ‘b,’ and the figure on the right will send “c.”



In the assembly code, each char is paired with a command to a separate subroutine (tfight, circle, and figure) that will mask OFF the upper nibble and add a specific value to the upper nibble:

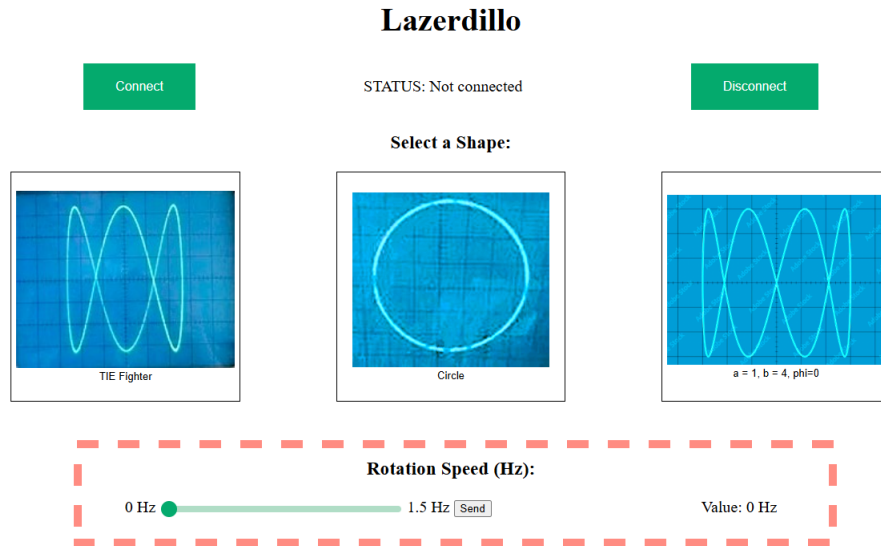
command ‘a’ —> adds a value of 10 (A in hexadecimal, 1010)
command ‘b’ —> adds a value of 11 (B in hexadecimal, 1011)
command ‘c’ —> adds a value of 12 (C in hexadecimal, 1100)

Key: SHAPES



2. Sending Rotation:

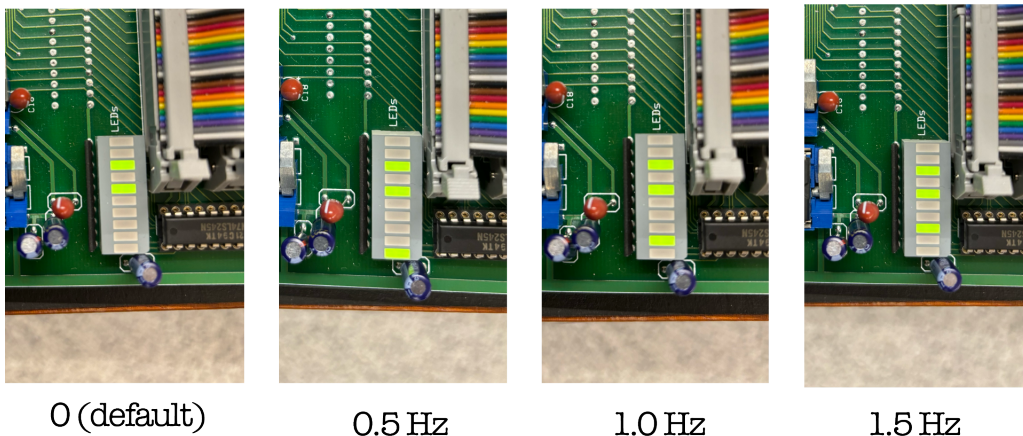
By default, the bottom nibble is off, and it will remain empty until a different rotation speed is sent. Each rotation speed (0.0, 0.5, 1.0, and 1.5 Hz) corresponds to a different char: 0.0 Hz will send “d,” 0.5 Hz will send “e,” 1.0 Hz will send “f,” and 1.5 Hz will send “g.” In the JS file, there is a comment in the slider `oninput` function instructing how to make the slider update the rotation speed in real time.



In the assembly code, each char is paired with a command to a separate subroutine (slorot, medrot, hghrot, and maxrot) that will mask OFF the lower nibble and add a specific value to the lower nibble:

- command 'd' —> adds no values (All OFF, 0000)
- command 'e' —> adds a value of 1 (First LED, 0001)
- command 'f' —> adds a value of 2 (Second LED, 0010)
- command 'g' —> adds a value of 4 (Third LED, 0100)

Key: Rotation



5 Helpful Links

- R31JP Lab Familiarization – <https://web.mit.edu/6.115/www/document/familiarization.pdf>
- Configuring Hypertext – <https://web.mit.edu/6.115/www/document/hyper.txt>
- as31 Manual – <https://web.mit.edu/6.115/www/document/as31.pdf>
- as31 notes – <https://web.mit.edu/6.115/www/document/as31-notes.txt>
- Chrome Developer Docs – <https://developer.chrome.com/docs/capabilities/serial>
- Web Serial API Guide – https://developer.mozilla.org/en-US/docs/Web/API/Web_Serial_API
- MDN Web Docs – <https://developer.mozilla.org>