Protocol Reverse Engineering
PREPERATION

Connect the Raspberry Pi Pico to the computer and copy the sshs.uf2 file to the drive. Open putty and select serial and set SerialLine to port where the device was connected and set the speed to 115200 and click on open.

1.UART Identification

Connect logic analyzer to the computer and perform the following connections:
1. GND on microcontroller to GND on logic analyzer
2. CH1 to GP0.

Open logic2 and select async serial, set parity bit to odd parity and start capture. Open putty and input ‘1’ for UART and press enter. This will send the UART signal from microcontroller and is captured in logic2. Expand the captured data and we get the flag.

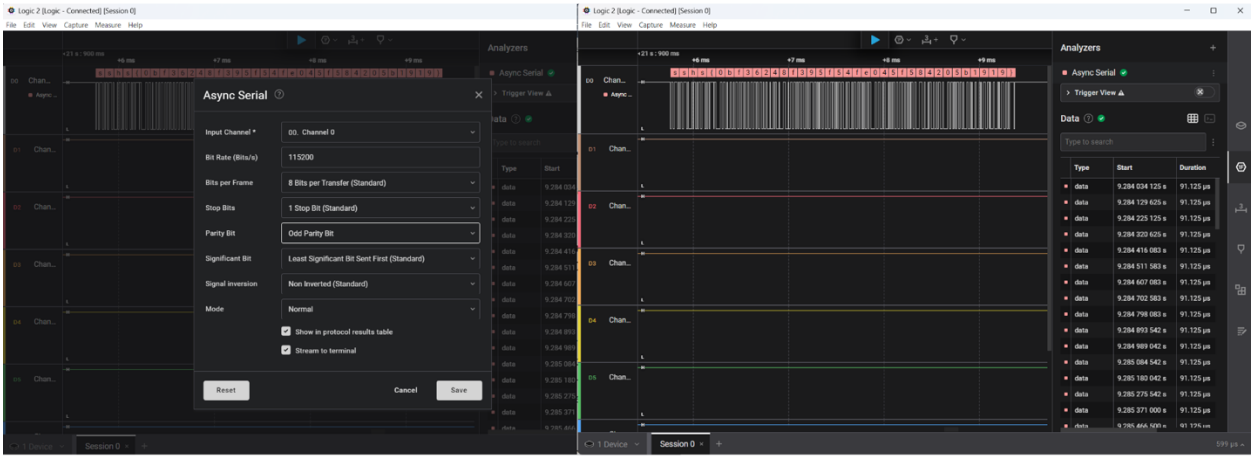


Fig1. UART settings

Flag1 = sshs{0bf362481395154fe045f584205b1919}

2.SPI Identification

We used the documentation of Raspberry pi Pico to identify the pins and connected the following pins.

Connections:

Ch1=MISO pin 6

Ch2=enable/cs pin 7

Ch3=clk pin 4

Ch4=MOSI pin 5

Open logic analyzer and select SPI and start capture. Open PUTTY input ‘2’ and press enter. The microcontroller will send an SPI signal which is captured in logic 2. Open the capture data in logic 2 and we will get the flag.

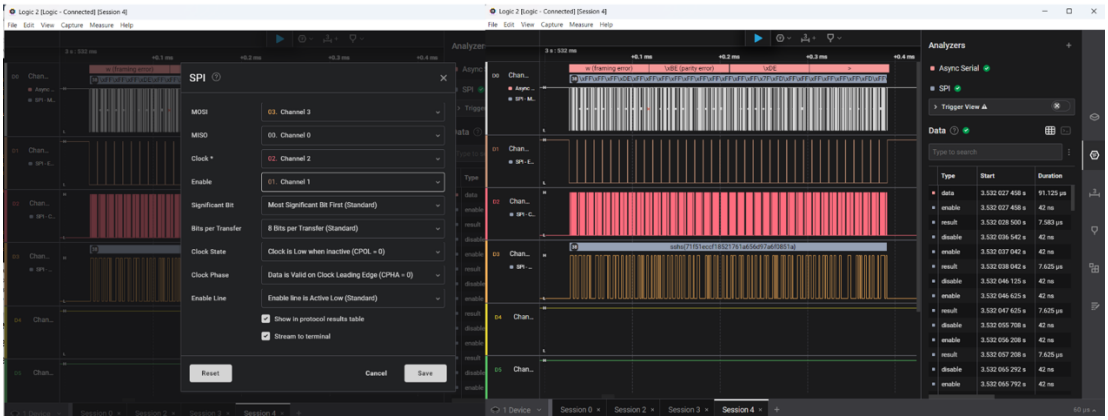


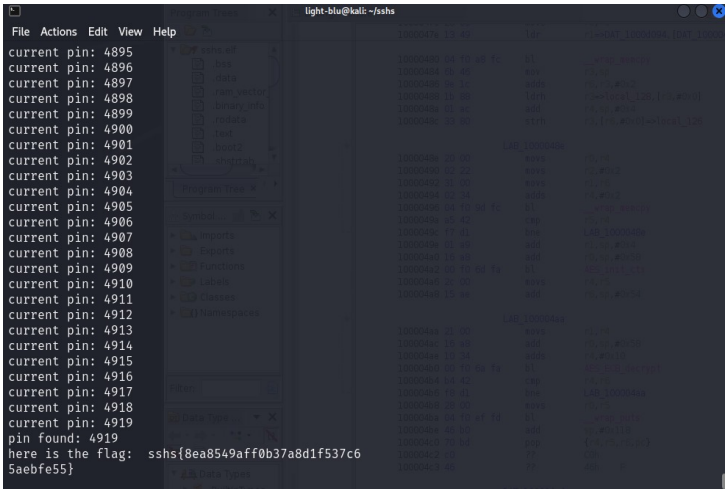
Fig2: SPI setting

Flag2= sshs{71f51eccf18521761a65d97a6f0851a}

3.Rehosting

Read ‘fw.bin’,’ rom.bin’, ‘sram.bin’ into the script. Create a list with all the registers and values. Loop through all values till 9999 and inside the loop perform memory mapping and write memory addresses of ‘firmware’, ‘rom’, and ‘sram’. Write to the registers from the list created. Hook ‘_wrap_printf’, ‘scanf’ , ‘sleep_ms’, and ‘_wrap_puts’. Skip ‘ _wrap_printf’ and ‘sleep_ms’ and for ‘scanf’ give current guess ‘i’ as user data and return when ‘_wrap_puts’ is called. Read the memory for the output string and check if it contains ‘sshs’ string in it. If it does then print the flag and break from the loop or else continue for next key guess.

.



Flag 3: sshs{8ea8549aff0b37a8d1f537c65aebfe55}