CEC Continuing Education Center

DesignNews

Raspberry Pi Pico/Pico2 Development Using Visual Studio Code

Day 4:

Coding Pico 2 Networking With VS Code

Sponsored by

DigiKey

informa markets

Webinar Logistics

- Turn on your system sound to hear the streaming presentation.
- If you have technical problems, click “Help” or submit a question asking for assistance.
- Participate in ‘Attendee Chat’ by maximizing the chat widget in your dock.

Fred Eady

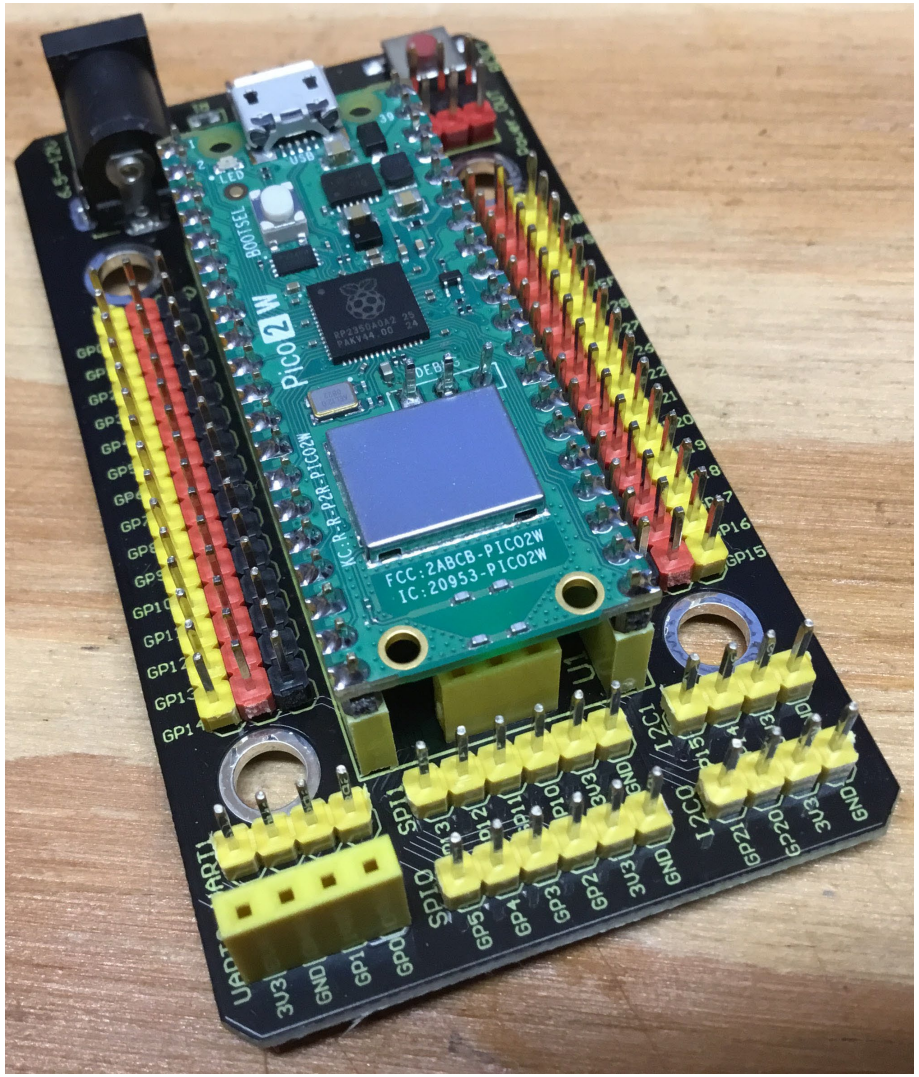
Visit ‘Lecturer Profile’ in your console for more details.

CEC Continuing Education Center

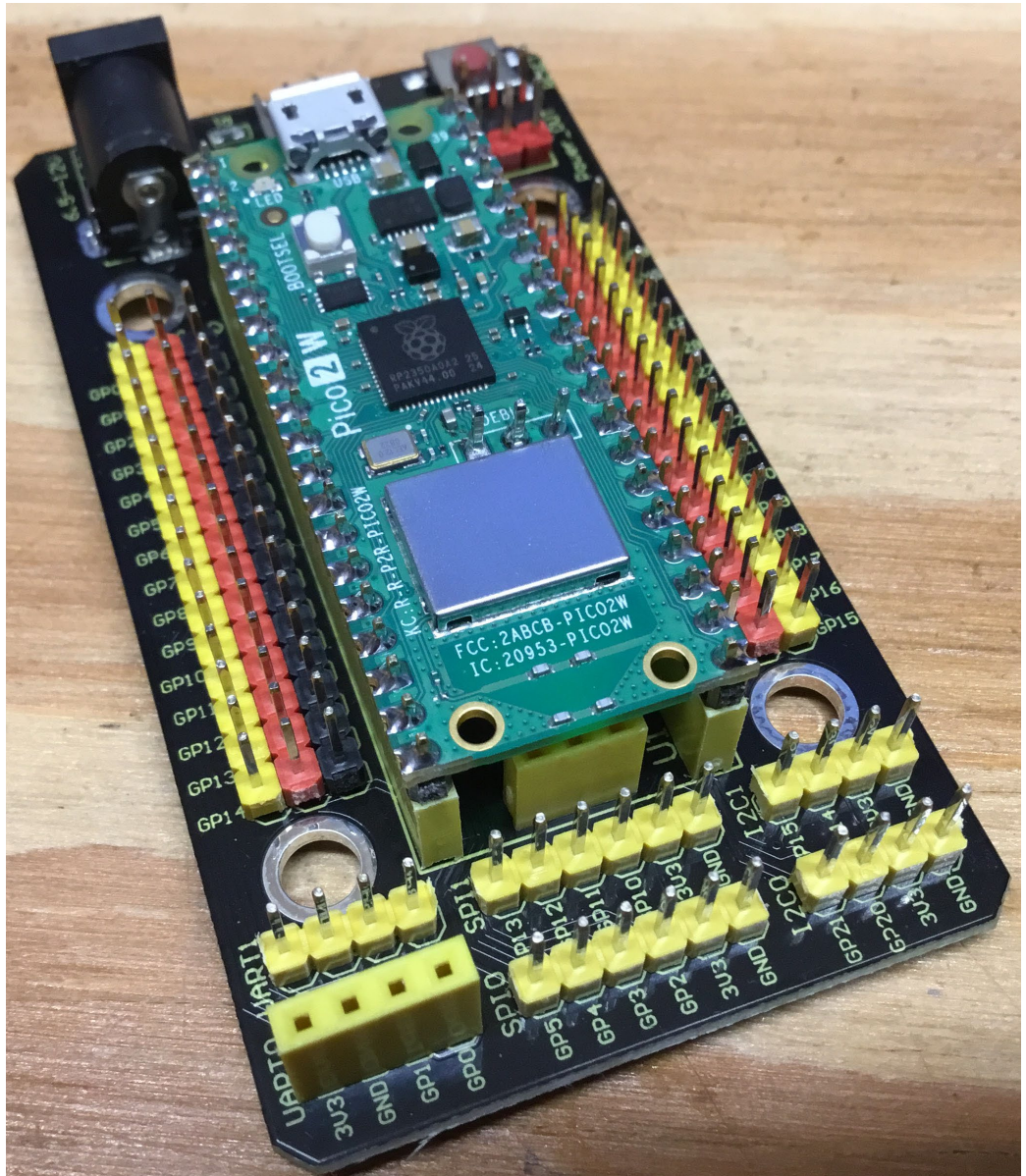
AGENDA

DigiKey

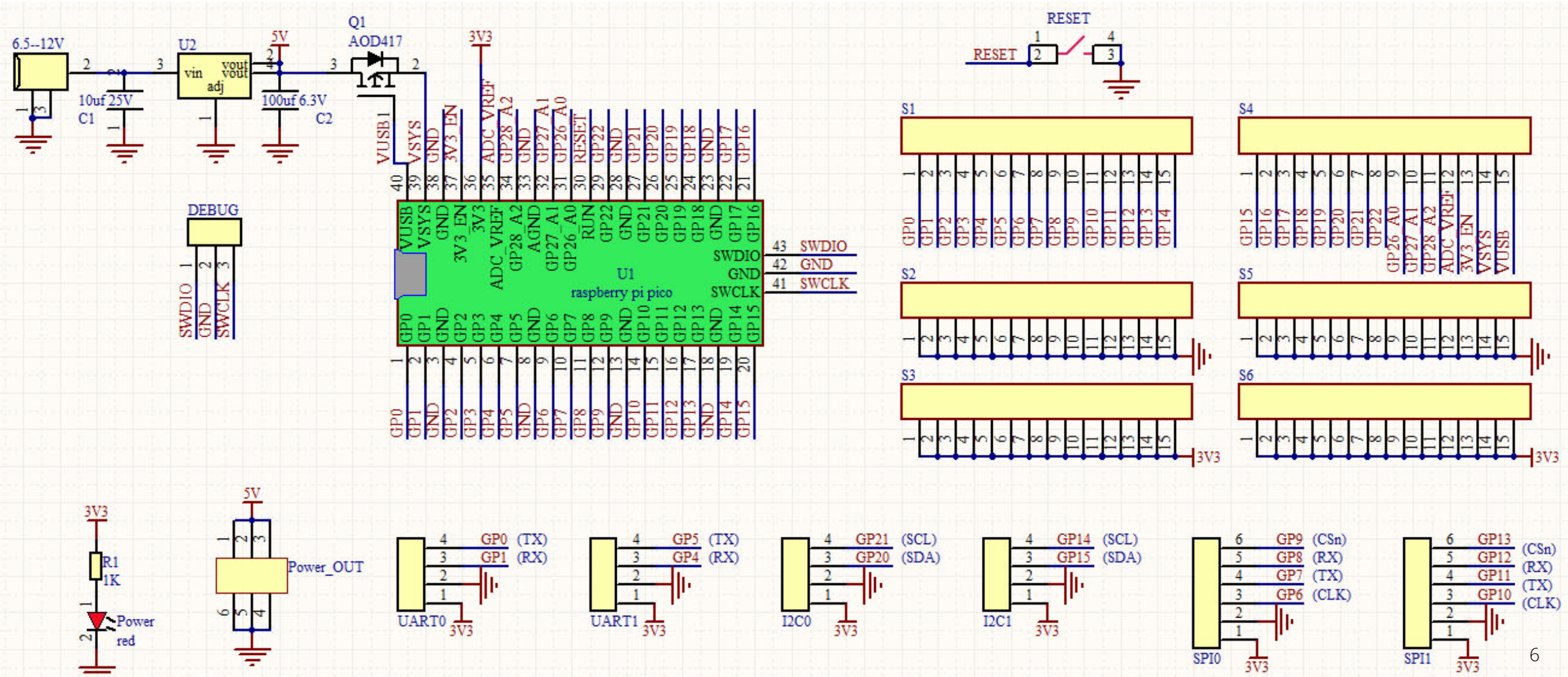
- Pico 2 W Development Hardware
- Raspberry Pi Debug Probe
  - The Hook Up
- Code a Pico 2 W Wi-Fi Application
  - Connect to the EDTP LAN Using Wi-Fi
  - Code a Pico 2 UDP Application



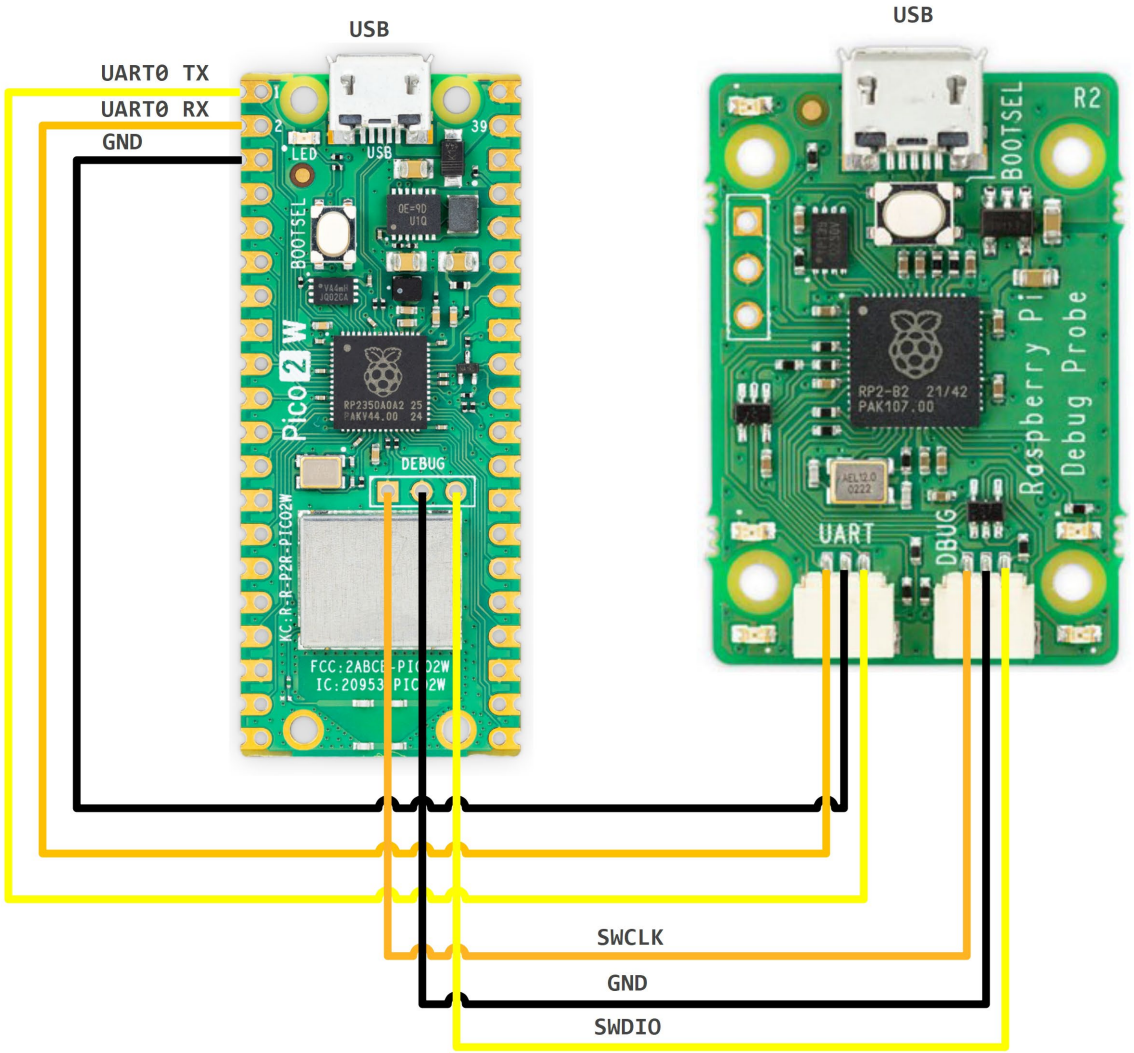
Pico 2 W Carrier Board

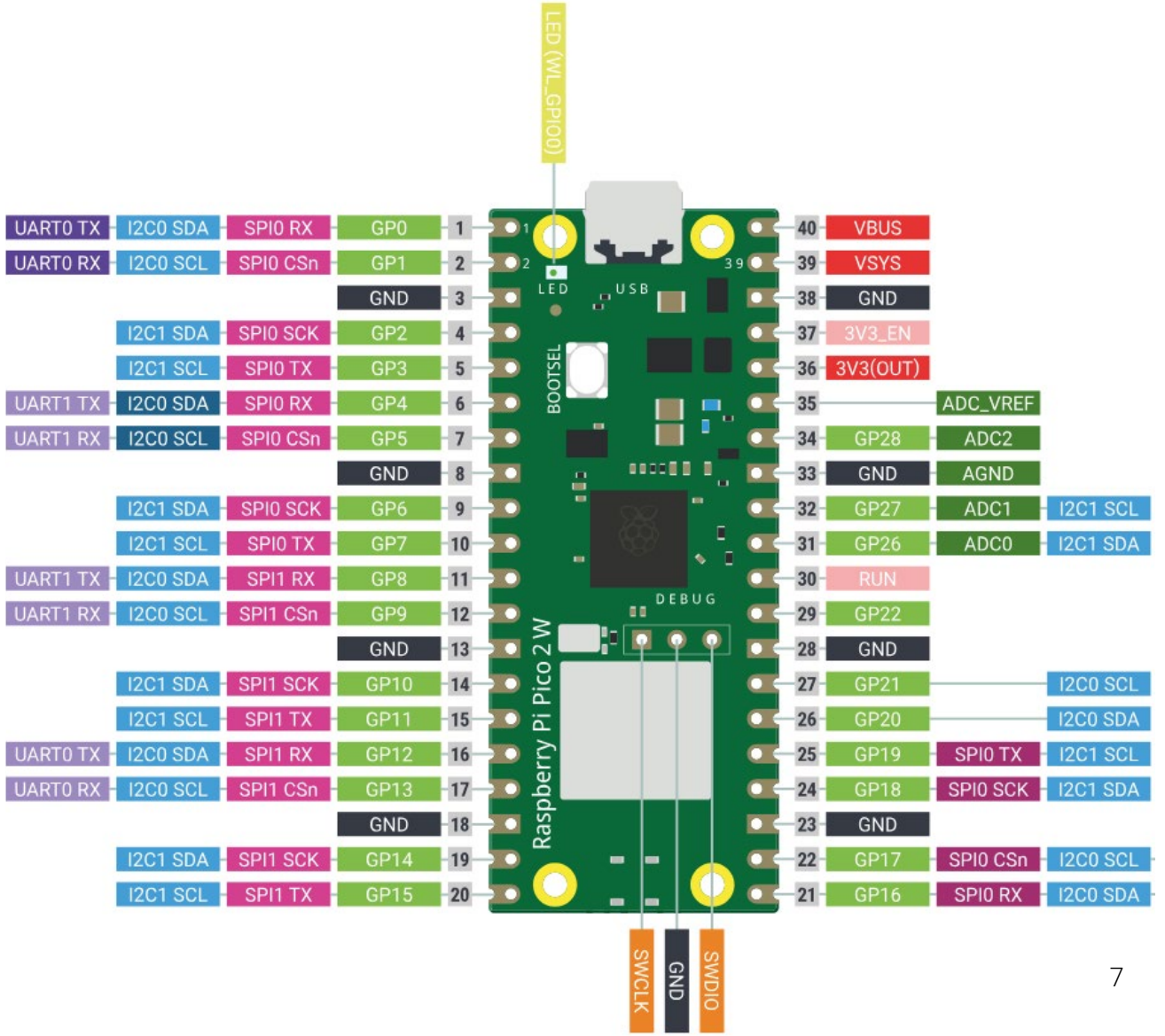


Pico 2 Carrier Board



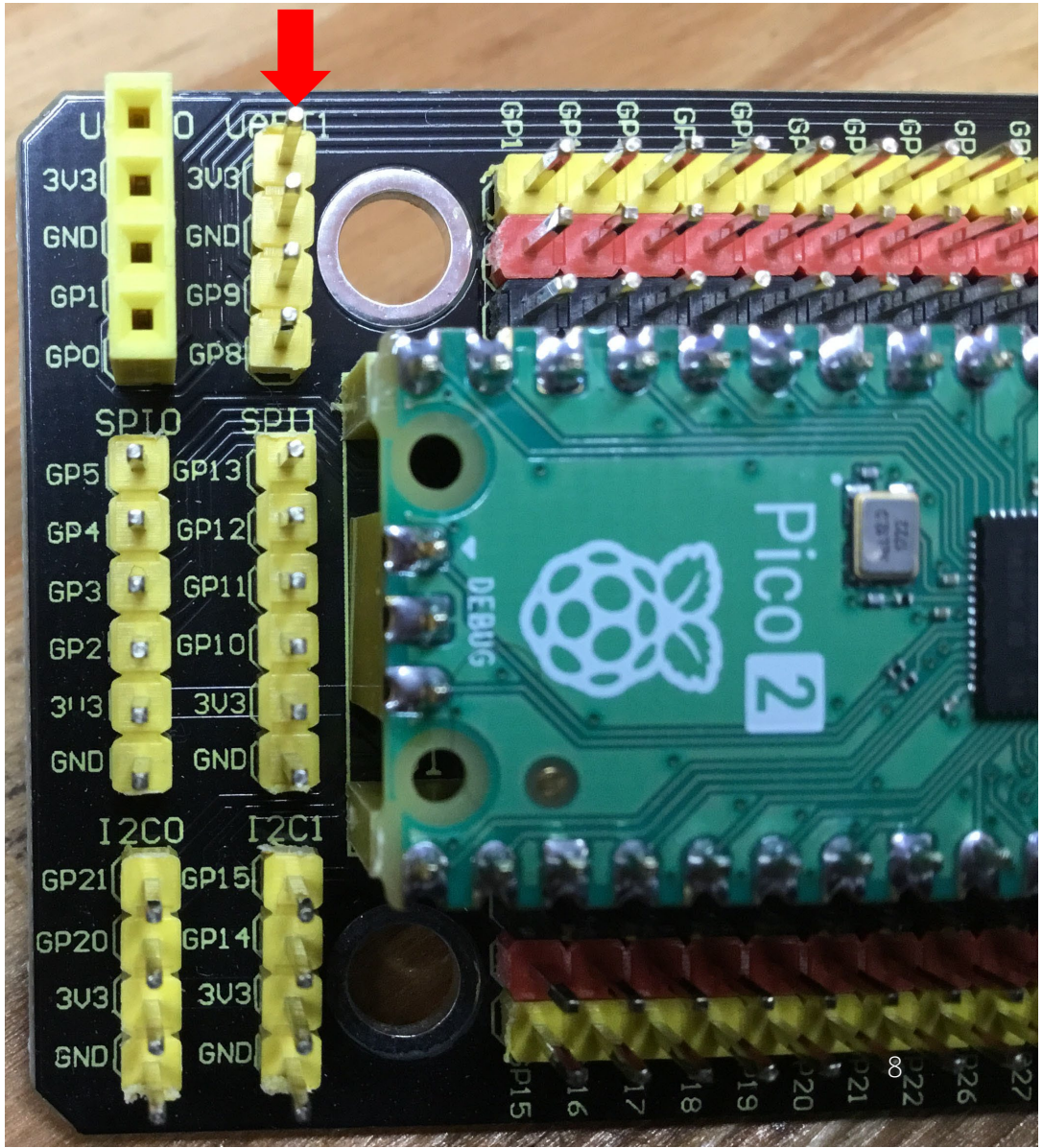
The Hook Up


USB
USB
UART0 TX
UART0 RX
GND
SWCLK
GND
SWDIO
LED
USB
BOOTSEL
DEBUG
UART
DBUG
R2
Raspberry Pi
Debug Probe
Pico 2 W

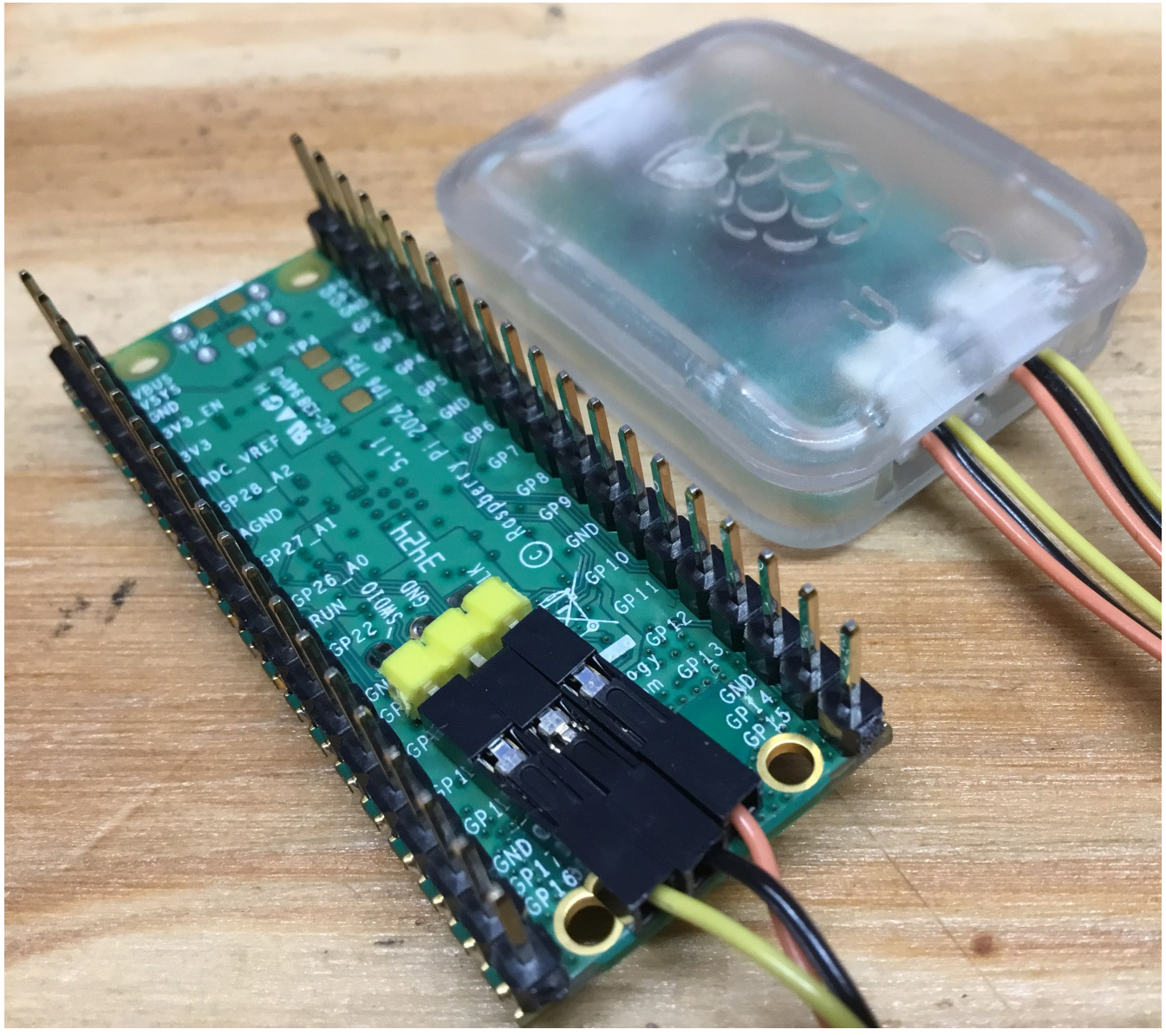

LED (WL_GPIO0)
UART0 TX I2C0 SDA SPI0 RX GP0 1
UART0 RX I2C0 SCL SPI0 CSn GP1 2
GND 3
I2C1 SDA SPI0 SCK GP2 4
I2C1 SCL SPI0 TX GP3 5
UART1 TX I2C0 SDA SPI0 RX GP4 6
UART1 RX I2C0 SCL SPI0 CSn GP5 7
GND 8
I2C1 SDA SPI0 SCK GP6 9
I2C1 SCL SPI0 TX GP7 10
UART1 TX I2C0 SDA SPI1 RX GP8 11
UART1 RX I2C0 SCL SPI1 CSn GP9 12
GND 13
I2C1 SDA SPI1 SCK GP10 14
I2C1 SCL SPI1 TX GP11 15
UART0 TX I2C0 SDA SPI1 RX GP12 16
UART0 RX I2C0 SCL SPI1 CSn GP13 17
GND 18
I2C1 SDA SPI1 SCK GP14 19
I2C1 SCL SPI1 TX GP15 20
40 VBUS
39 VSYS
38 GND
37 3V3_EN
36 3V3(OUT)
35 ADC_VREF
34 GP28 ADC2
33 GND AGND
32 GP27 ADC1 I2C1 SCL
31 GP26 ADC0 I2C1 SDA
30 RUN
29 GP22
28 GND
27 GP21 I2C0 SCL
26 GP20 I2C0 SDA
25 GP19 SPI0 TX I2C1 SCL
24 GP18 SPI0 SCK I2C1 SDA
23 GND
22 GP17 SPI0 CSn I2C0 SCL
21 GP16 SPI0 RX I2C0 SDA
LED
USB
BOOTSEL
DEBUG
Raspberry Pi Pico 2 W
SWCLK
GND
SWDIO

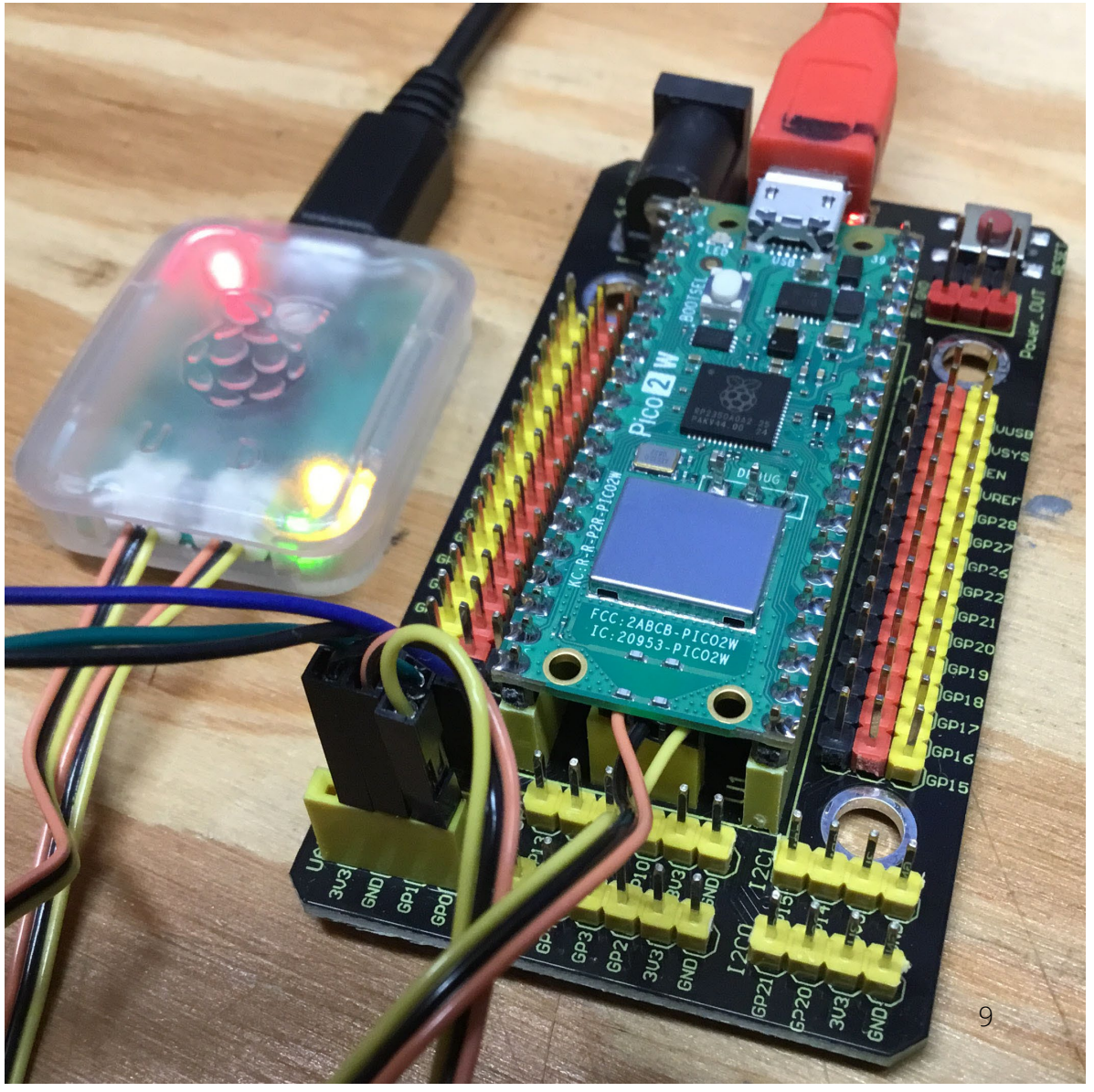
The Hook Up


USB TO TTL (D)
PWR
TXD
RXD
USB
5V
3V3
VCC
GND
TXD
RXD

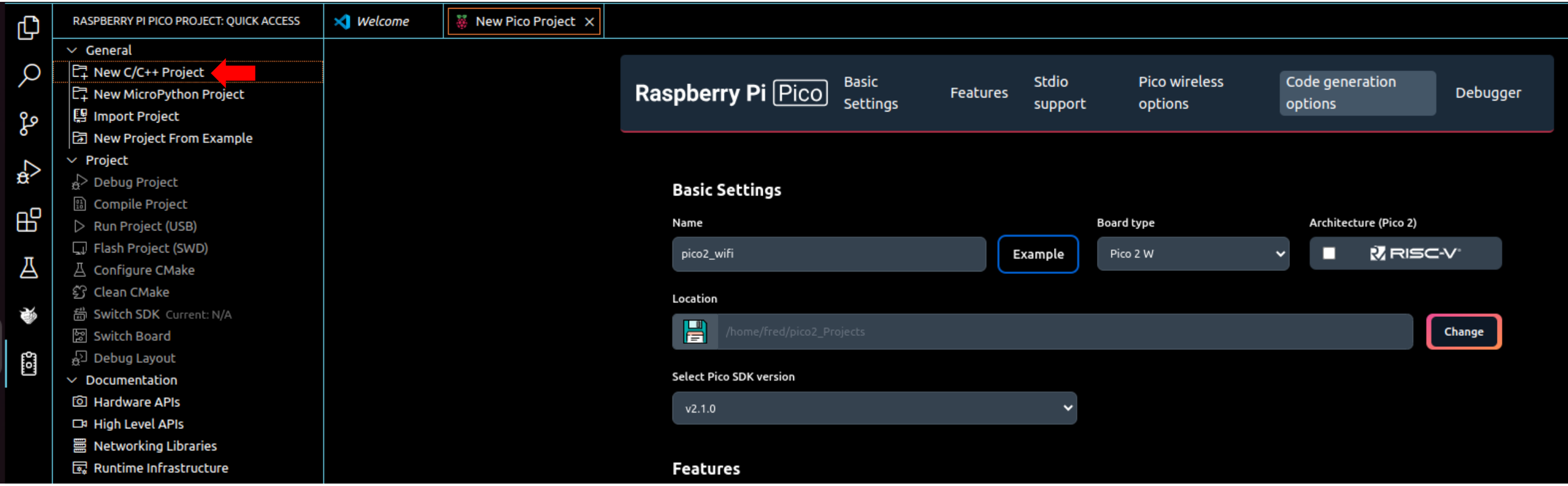

3V3
GND
GP1
GP0
GP9
GP8
SPI0
SPI1
GP5
GP4
GP3
GP2
GP13
GP12
GP11
GP10
3V3
GND
I2C0
I2C1
GP21
GP20
GP15
GP14
Pico 2
DEBUG

The Hook Up

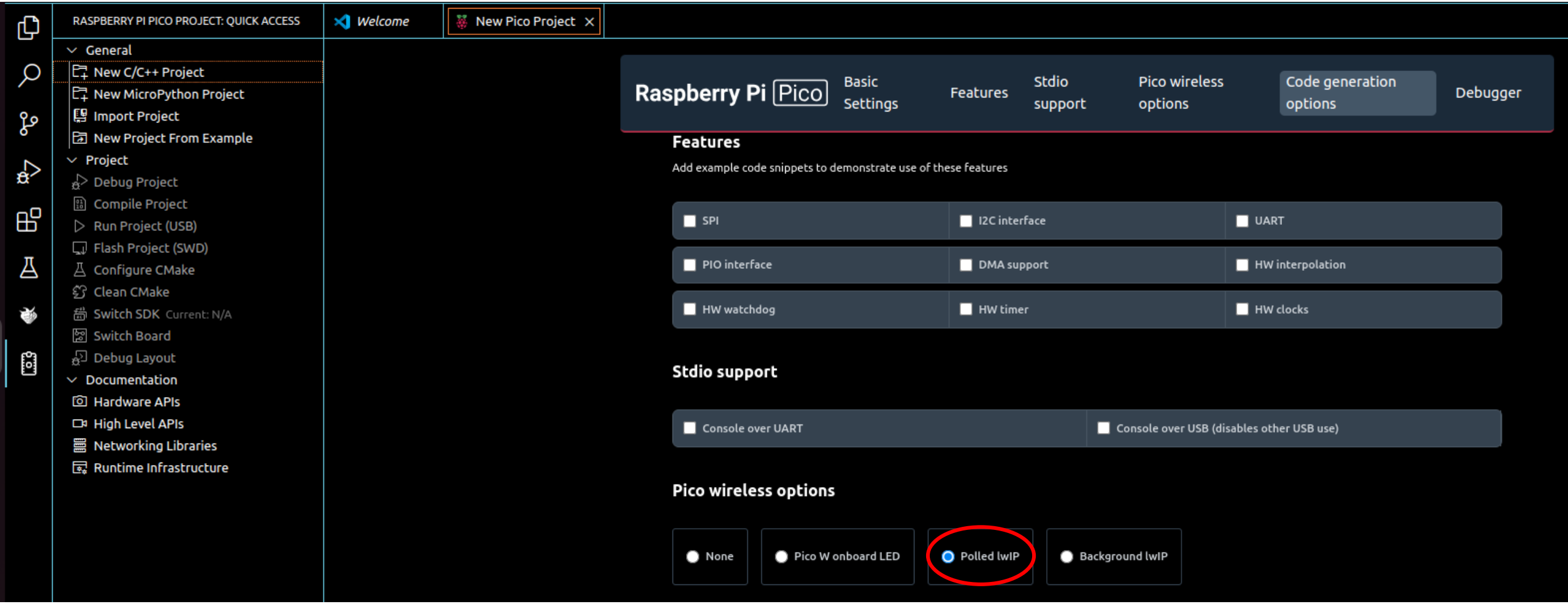




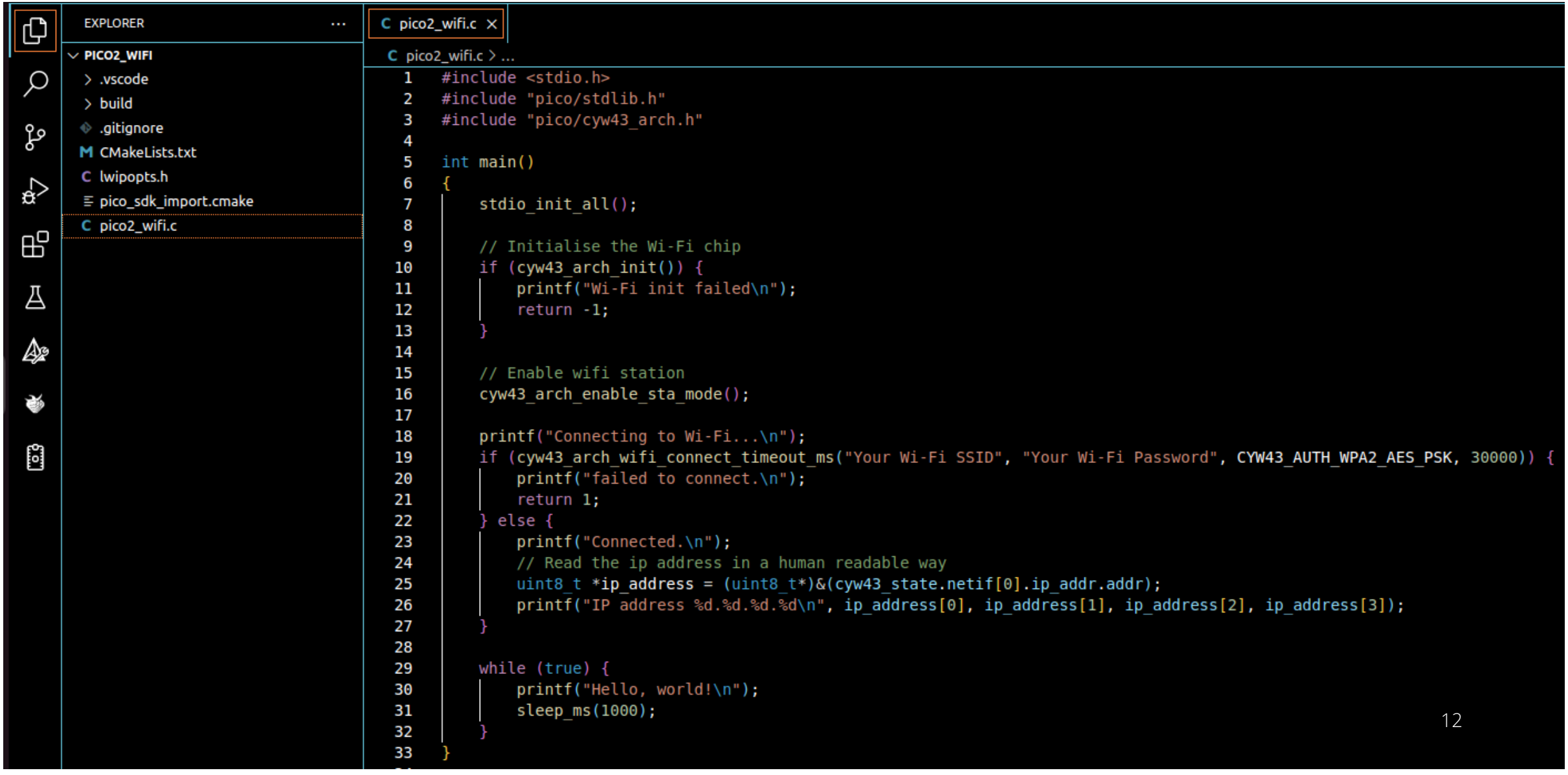
Connect to the EDTP Shop LAN



Connect to the EDTP Shop LAN



Connect to the EDTP Shop LAN



Connect to the EDTP Shop LAN

C pico2_wifi.c > main()

```c
#include <stdio.h>
#include "pico/stdlib.h"
#include "pico/cyw43_arch.h"
#include "hardware/uart.h"

#define UART1_ID uart1
#define BAUD_RATE 115200
#define DATA_BITS 8
#define STOP_BITS 1
#define PARITY    UART_PARITY_NONE

#define UART1_TX_PIN 8
#define UART1_RX_PIN 9
```

Connect to the EDTP Shop LAN

pico2_wifi.c > main()

```c
#include <stdio.h>
#include "pico/stdlib.h"
#include "pico/cyw43_arch.h"
#include "hardware/uart.h"

#define UART1_ID uart1
#define BAUD_RATE 115200
#define DATA_BITS 8
#define STOP_BITS 1
#define PARITY    UART_PARITY_NONE

#define UART1_TX_PIN 8
#define UART1_RX_PIN 9

int main()
{
    gpio_set_function(UART1_TX_PIN, UART_FUNCSEL_NUM(UART1_ID, UART1_TX_PIN));
    gpio_set_function(UART1_RX_PIN, UART_FUNCSEL_NUM(UART1_ID, UART1_RX_PIN));

    //uart_init(UART0_ID, 115200);
    uart_init(UART1_ID, 115200);
    //uart_set_hw_flow(UART0_ID, false, false);
    uart_set_hw_flow(UART1_ID, false, false);
    //uart_set_format(UART0_ID, DATA_BITS, STOP_BITS, PARITY);
    uart_set_format(UART1_ID, DATA_BITS, STOP_BITS, PARITY);
    //uart_set_fifo_enabled(UART0_ID, false);
    uart_set_fifo_enabled(UART1_ID, false);
```

Connect to the EDTP Shop LAN

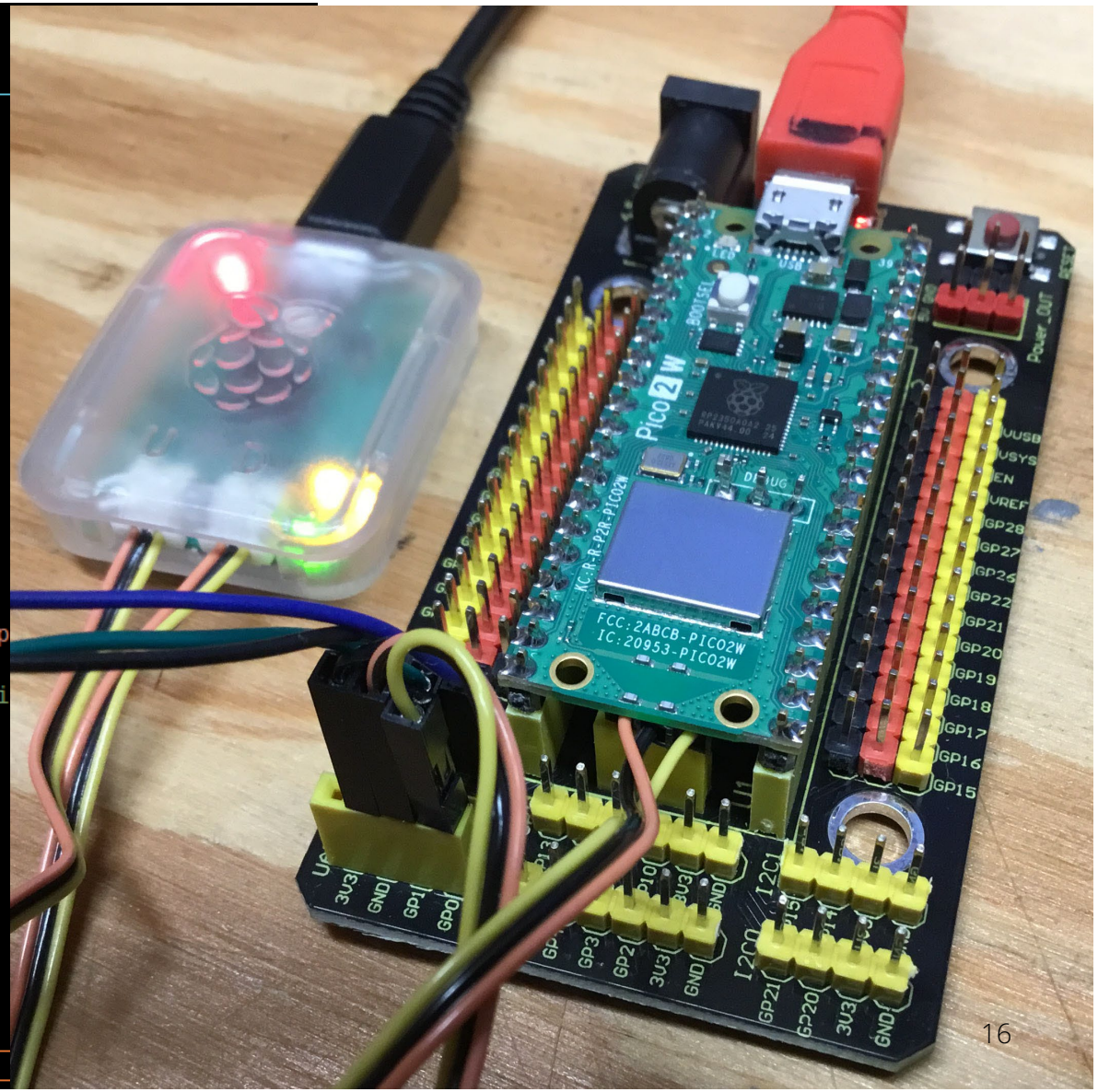
```
29      // Initialise the Wi-Fi chip
30      if (cyw43_arch_init()) {
31          uart_puts(UART1_ID,"Wi-Fi init failed\n");
32          return -1;
33      }
34
35      // Enable wifi station
36      cyw43_arch_enable_sta_mode();
37
38      uart_puts(UART1_ID,"Connecting to Wi-Fi...\n");
39      if (cyw43_arch_wifi_connect_timeout_ms("edtpnet2", "Your Password", CYW43_AUTH_WPA2_AES_PSK, 30000)) {
40          uart_puts(UART1_ID,"failed to connect.\n");
41          return 1;
42      } else {
43          uart_puts(UART1_ID,"Connected.\n");
44      }
45
46      while (true) {
47          uart_puts(UART1_ID,"Connected to edtpnet2!\n");
48          sleep_ms(1000);
49      }
50  }
```

Enable *liveWatch*

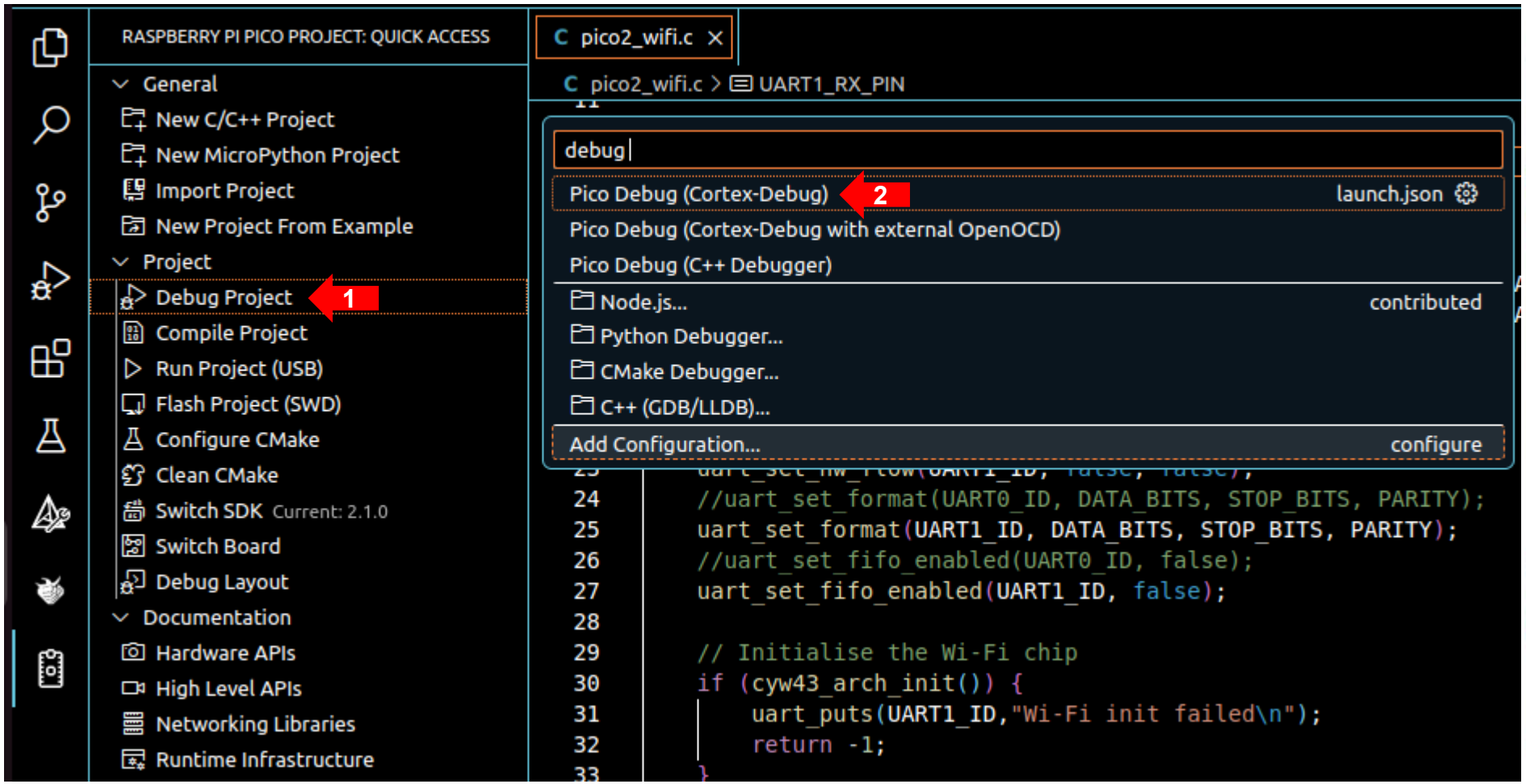
pico2_wifi.c | {} launch.json 1

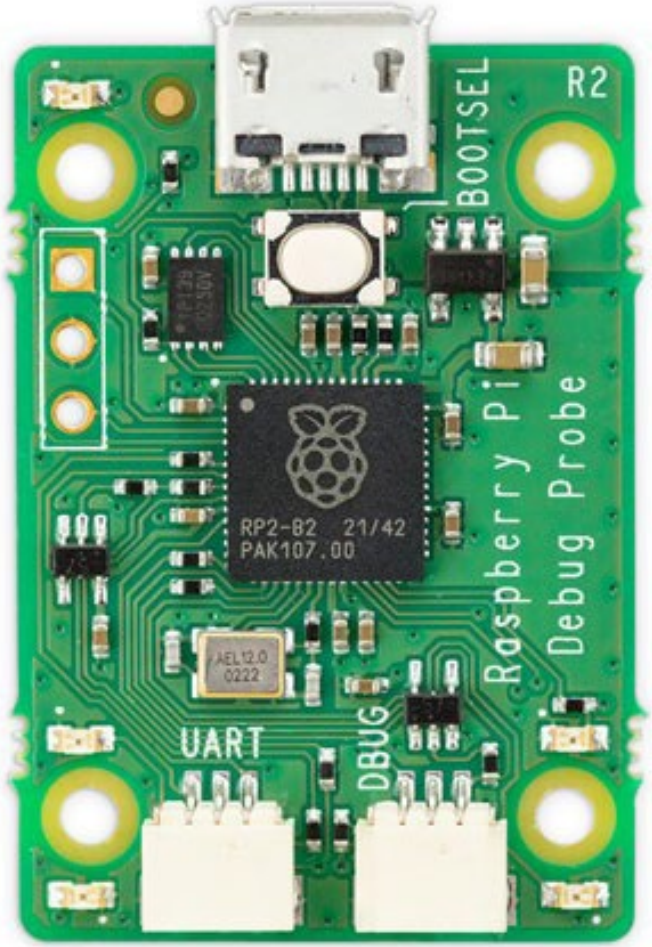
.vscode > {} launch.json > Launch Targets > {} Pico Debug (Cortex-Debug)

```
{
    "version": "0.2.0",
    "configurations": [
        {
            "name": "Pico Debug (Cortex-Debug)",
            "cwd": "${userHome}/.pico-sdk/openocd/0.12.0+dev/scripts",
            "executable": "${command:raspberry-pi-pico.launchTargetPath}",
            "request": "launch",
            "type": "cortex-debug",
            "servertype": "openocd",
            "serverpath": "${userHome}/.pico-sdk/openocd/0.12.0+dev/openocd.exe",
            "gdbPath": "${command:raspberry-pi-pico.getGDBPath}",
            "device": "${command:raspberry-pi-pico.getChipUppercase}",
            "configFiles": [
                "interface/cmsis-dap.cfg",
                "target/${command:raspberry-pi-pico.getTarget}.cfg"
            ],
            "svdFile": "${userHome}/.pico-sdk/sdk/2.1.0/src/${command:raspberry-pi-p
            "runToEntryPoint": "main",
            // Fix for no_flash binaries, where monitor reset halt doesn't do what i
            // Also works fine for flash binaries
            "overrideLaunchCommands": [
                "monitor reset init",
                "load \"${command:raspberry-pi-pico.launchTargetPath}\""
            ],
            "openOCDLaunchCommands": [
                "adapter speed 5000"
            ]
            "liveWatch": {
                "enabled": true,
                "samplesPerSecond": 4
            }
```

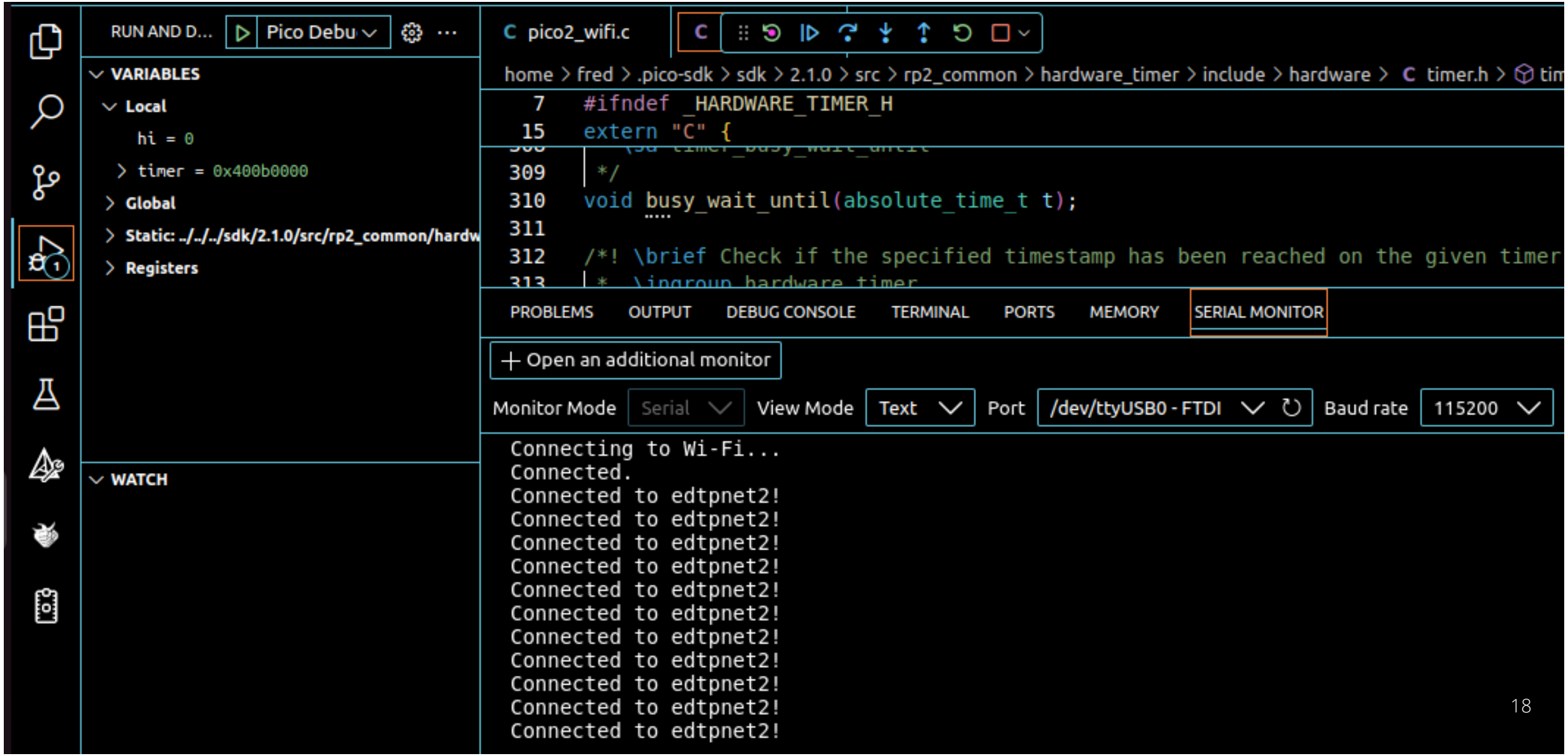


Debug the Pico 2 W Wi-Fi Connect Application





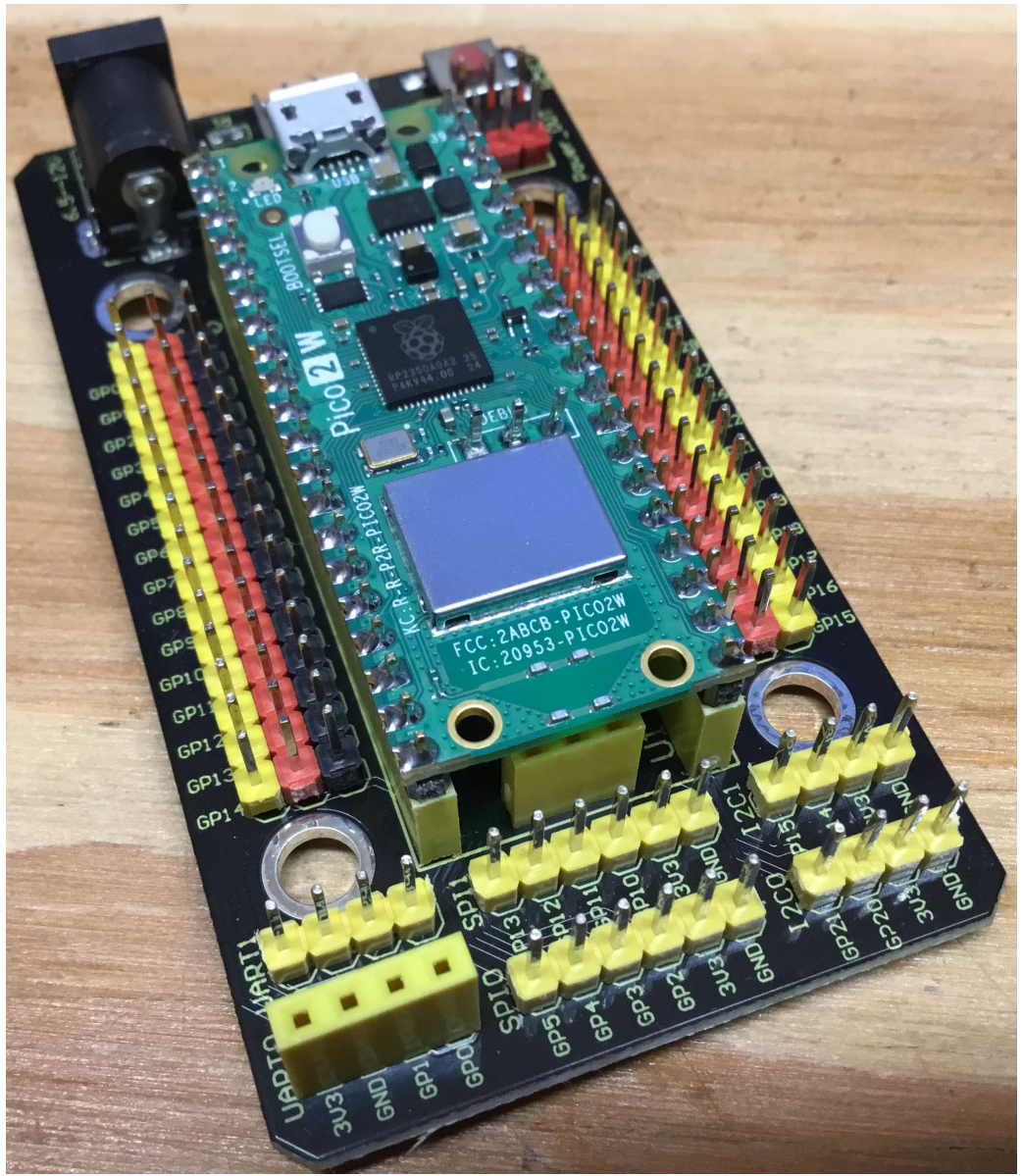
Debug the Pico 2 W Wi-Fi Connect Application



Define the UDP Application Ports and IP Addresses

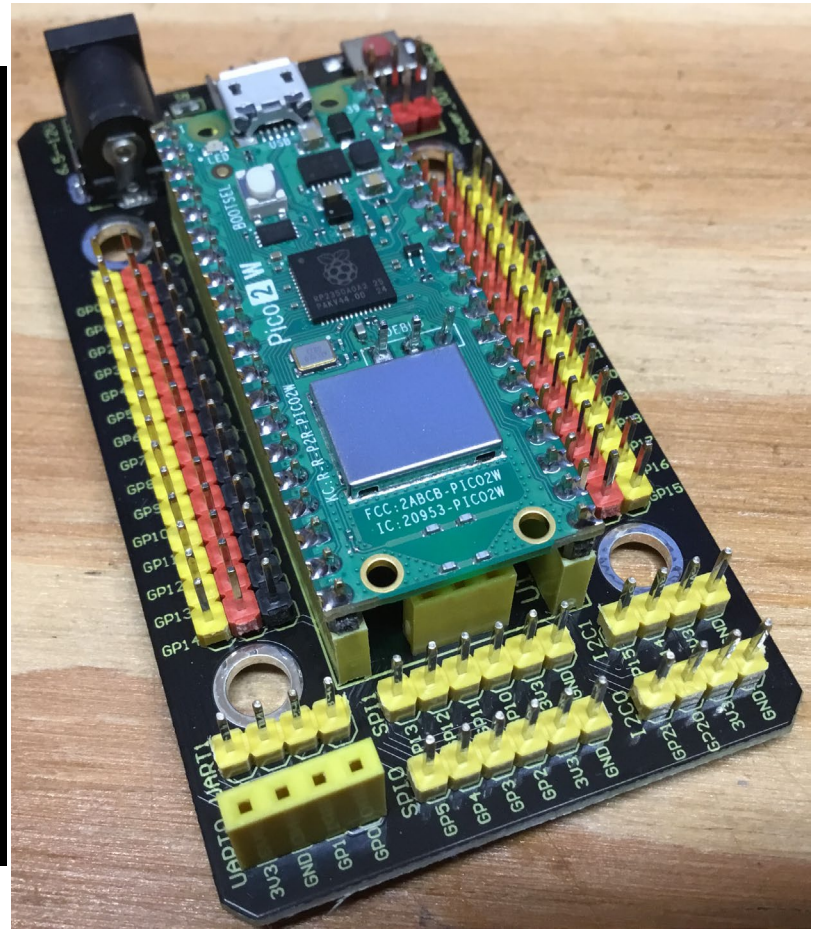
pico2w_udp.c

```c
#include <stdio.h>

#include "pico/cyw43_arch.h"

#include "lwip/pbuf.h"
#include "lwip/udp.h"
#include "hardware/uart.h"

#define UART1_ID uart1
#define BAUD_RATE 115200
#define DATA_BITS 8
#define STOP_BITS 1
#define PARITY    UART_PARITY_NONE

#define UART1_TX_PIN 8
#define UART1_RX_PIN 9

#define UDP_PORT 4444
#define UDP_MSG_LEN_MAX 127
#define UDP_TARGET "255.255.255.255"
#define UDP_TX_INTERVAL 1000

uint16_t sent_pkt_cnt = 0;
```



Code the UDP *send_udp_packet* Function

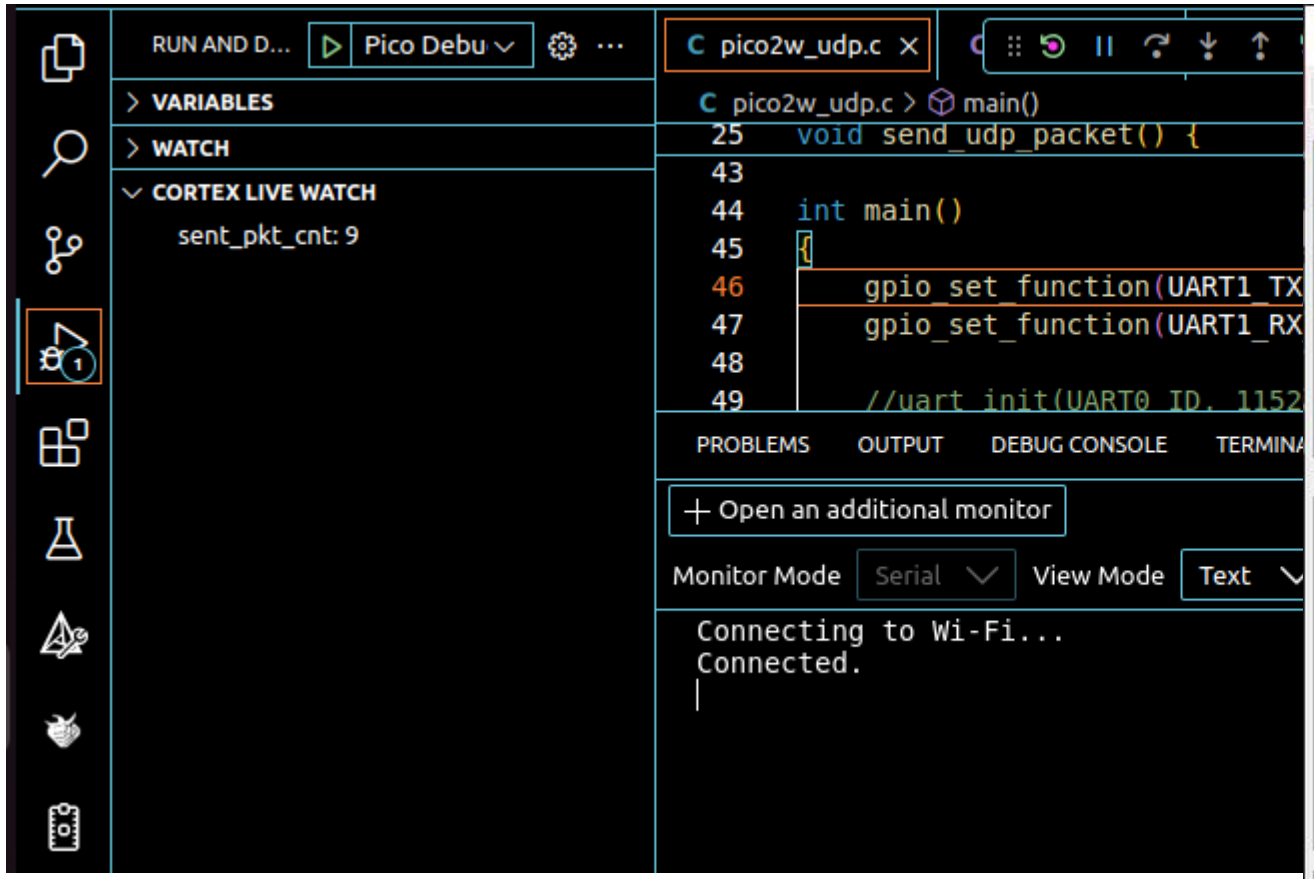
```
25  void send_udp_packet() {
26      struct udp_pcb* pcb = udp_new();
27
28      ip_addr_t addr;
29      ipaddr_aton(UDP_TARGET, &addr);
30
31      while (true) {
32          struct pbuf *p = pbuf_alloc(PBUF_TRANSPORT, UDP_MSG_LEN_MAX+1, PBUF_RAM);
33          char *pkt = (char *)p->payload;
34          memset(pkt, 0, UDP_MSG_LEN_MAX+1);
35          snprintf(pkt, UDP_MSG_LEN_MAX, "%d\n", sent_pkt_cnt);
36          err_t er = udp_sendto(pcb, p, &addr, UDP_PORT);
37          pbuf_free(p);
38          sent_pkt_cnt++;
39          cyw43_arch_poll();
40          sleep_ms(UDP_TX_INTERVAL);
41      }
42  }
```

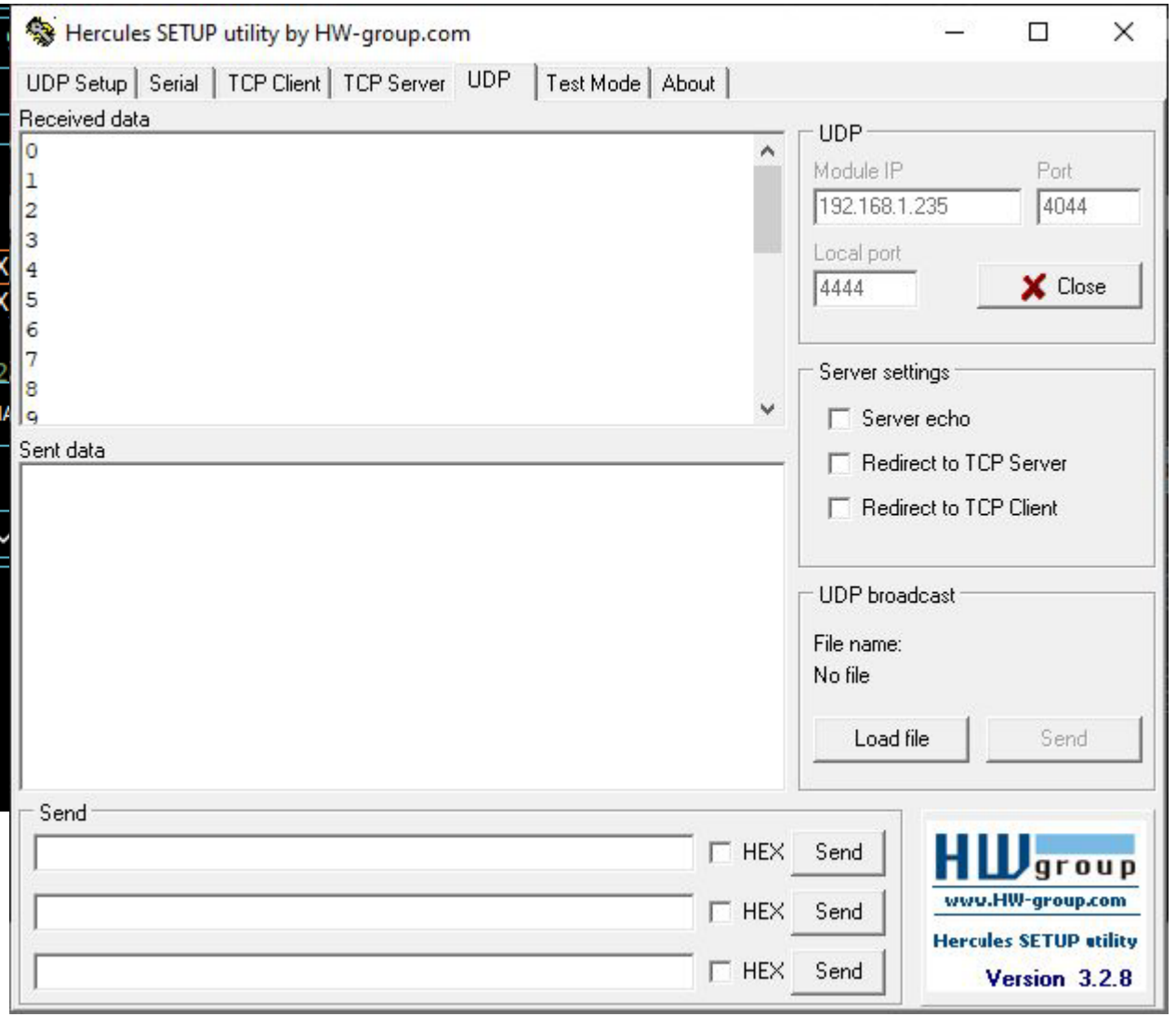


Add the *send_udp_packet* Function Call

```
58      // Initialise the Wi-Fi chip
59      if (cyw43_arch_init()) {
60          uart_puts(UART1_ID,"Wi-Fi init failed\n");
61          return -1;
62      }
63
64      // Enable wifi station
65      cyw43_arch_enable_sta_mode();
66
67      uart_puts(UART1_ID,"Connecting to Wi-Fi...\n");
68      if (cyw43_arch_wifi_connect_timeout_ms("edtpnet2", "Your Password", CYW43_AUTH_WPA2_AES_PSK, 30000)) {
69          uart_puts(UART1_ID,"failed to connect.\n");
70          return 1;
71      } else {
72          uart_puts(UART1_ID,"Connected.\n");
73      }
74
75      send_udp_packet();
76      return 0;
77  }
```

Debug the Pico 2 W UDP Application





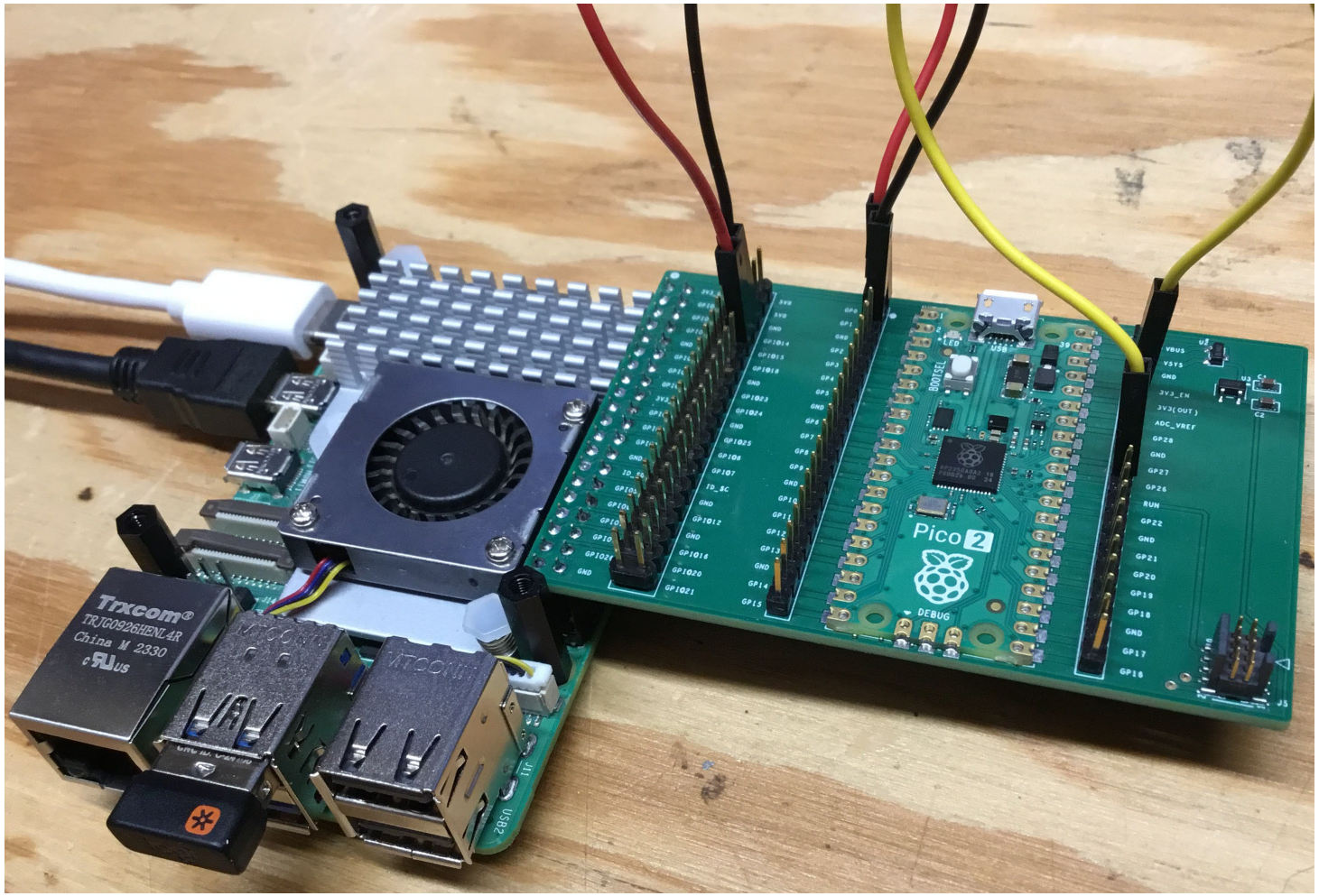
Next Time…

MORE TO COME..

Thank you for attending!!!

Please consider the resources below:

- Today's Download Package
- raspberrypi.org



CEC Continuing Education Center

DesignNews

Thank You

Sponsored by

DigiKey

informa markets