



**DesignNews**

## Constructing a Raspberry Pi RP2040 Low-Power Sensor Node

**Day 3:**

### RP2040 Sensor Node Power Subsystem Design

Sponsored by



## 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.

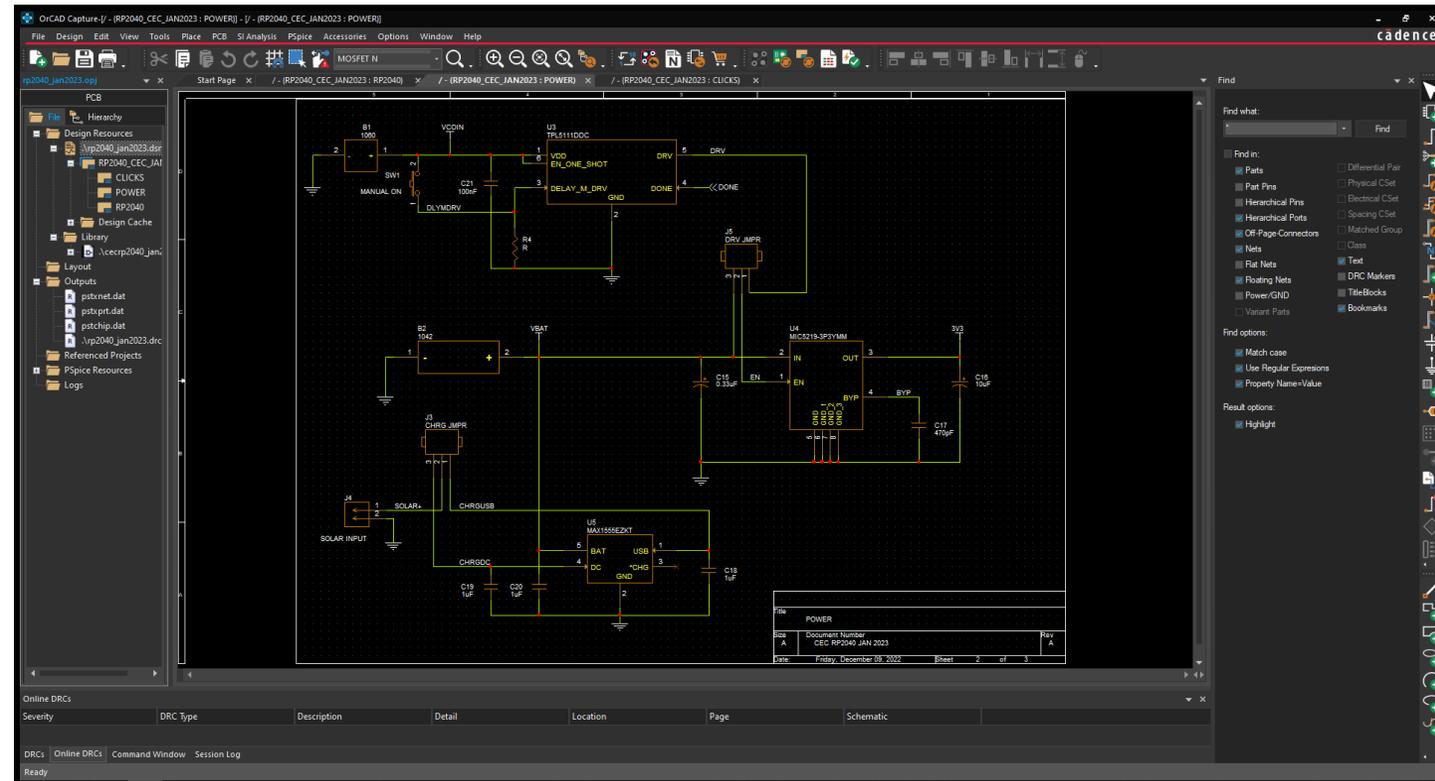


## Fred Eady

Visit 'Lecturer Profile' in your console for more details.

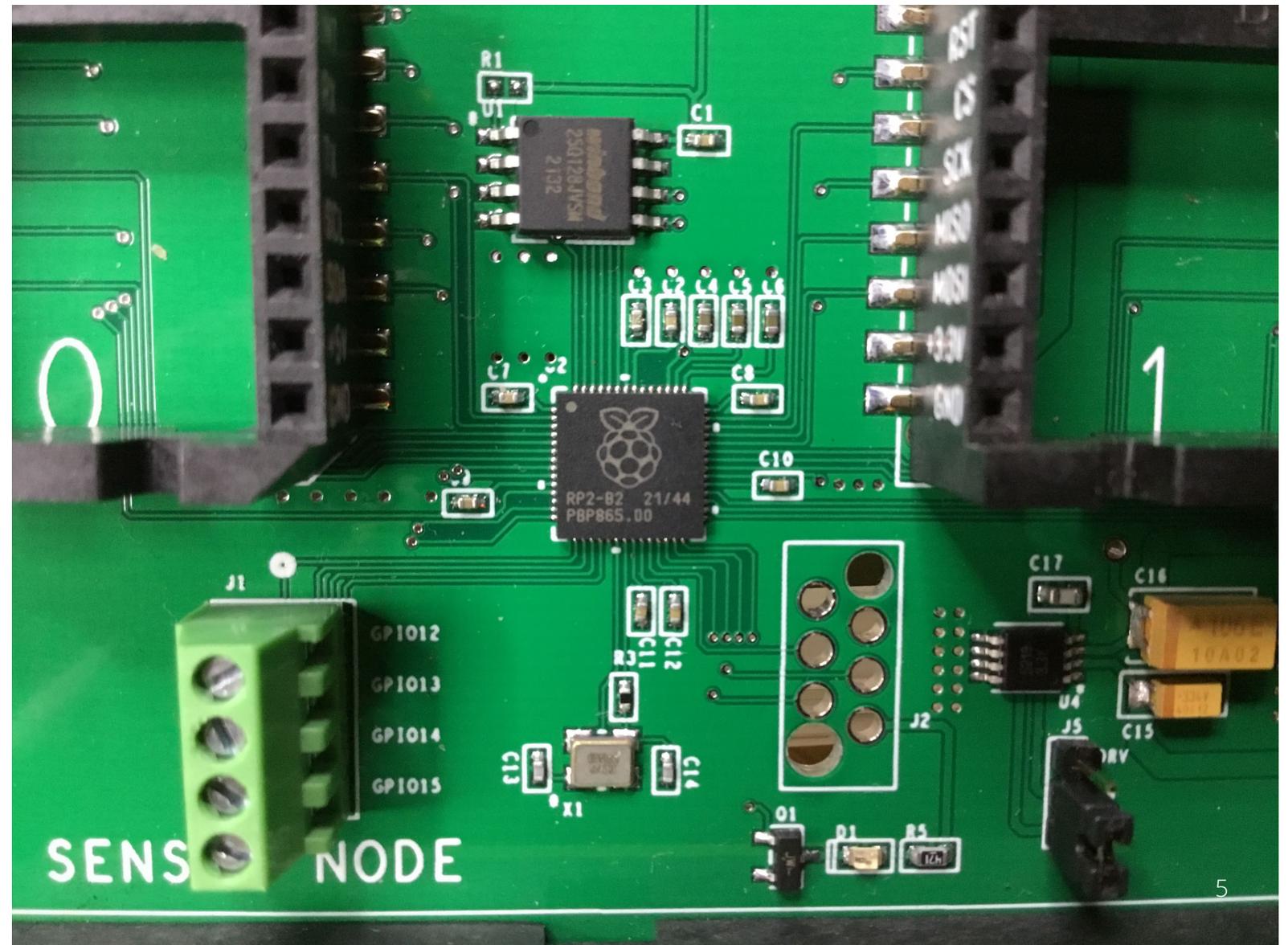
# AGENDA

- **Traditional Methods**
- **Hardware Solution**
  - **Main Power Regulator**
  - **Solar Battery Charging**
  - **Power Switching**



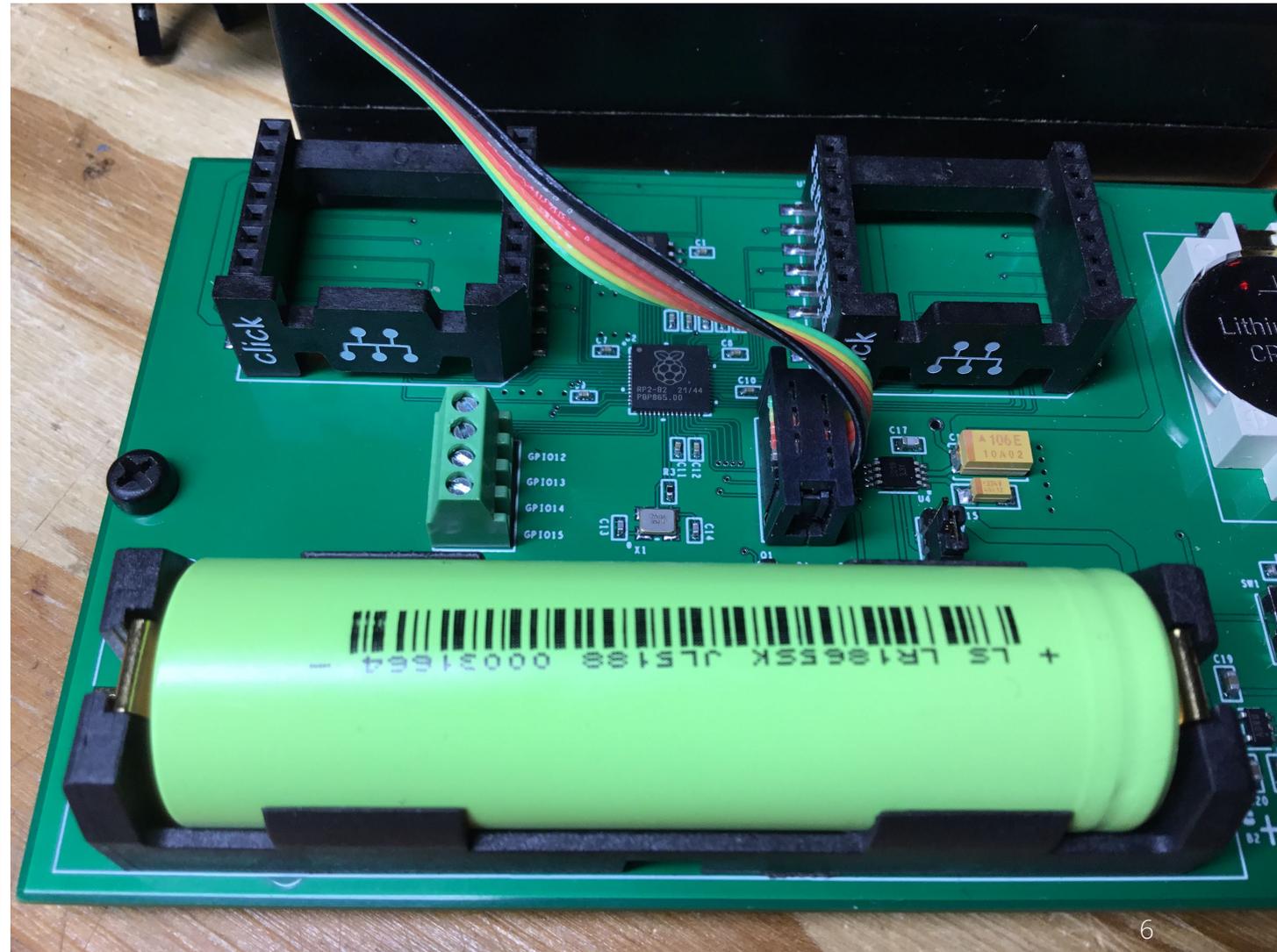
## Sleep Mode

```
CMakeLists.txt  C main.c  x
C main.c
7
8 static void sleep_callback(void) {
9     printf("RTC woke us up\n");
10    awake = true;
11 }
12
13 static void rtc_sleep(void) {
14     // Start on Friday 5th of June 2020 15:45:00
15     datetime_t t = {
16         .year = 2020,
17         .month = 06,
18         .day = 05,
19         .dotw = 5, // 0 is Sunday, so 5 is Friday
20         .hour = 15,
21         .min = 45,
22         .sec = 00
23     };
24
25     // Alarm 10 seconds later
26     datetime_t t_alarm = {
27         .year = 2020,
28         .month = 06,
29         .day = 05,
30         .dotw = 5, // 0 is Sunday, so 5 is Friday
31         .hour = 15,
32         .min = 45,
33         .sec = 10
34     };
35
36     // Start the RTC
37     rtc_init();
38     rtc_set_datetime(&t);
39
40     printf("Sleeping for 10 seconds\n");
41     uart_default_tx_wait_blocking();
42
43     sleep_goto_sleep_until(&t_alarm, &sleep_callback);
44 }
```



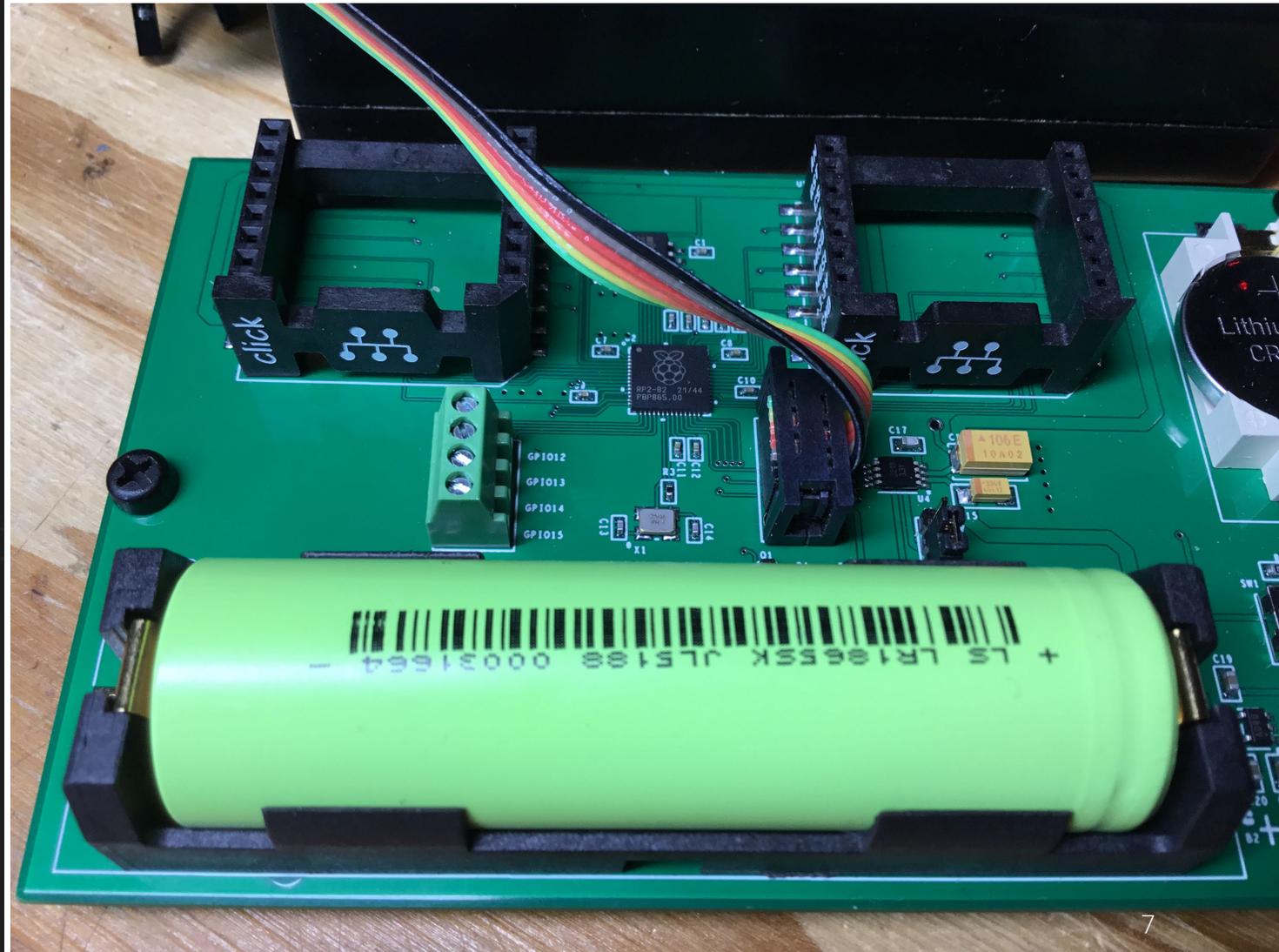
## Sleep Mode

```
46 int main() {
47     stdio_init_all();
48     printf("Hello Sleep!\n");
49
50     printf("Switching to XOSC\n");
51
52     // Wait for the fifo to be drained so we get reliable
53     uart_default_tx_wait_blocking();
54
55     // UART will be reconfigured by sleep_run_from_xosc
56     sleep_run_from_xosc();
57
58     printf("Switched to XOSC\n");
59
60     awake = false;
61
62     rtc_sleep();
63
64     // Make sure we don't wake
65     while (!awake) {
66         printf("Should be sleeping\n");
67     }
68
69     return 0;
70 }
```

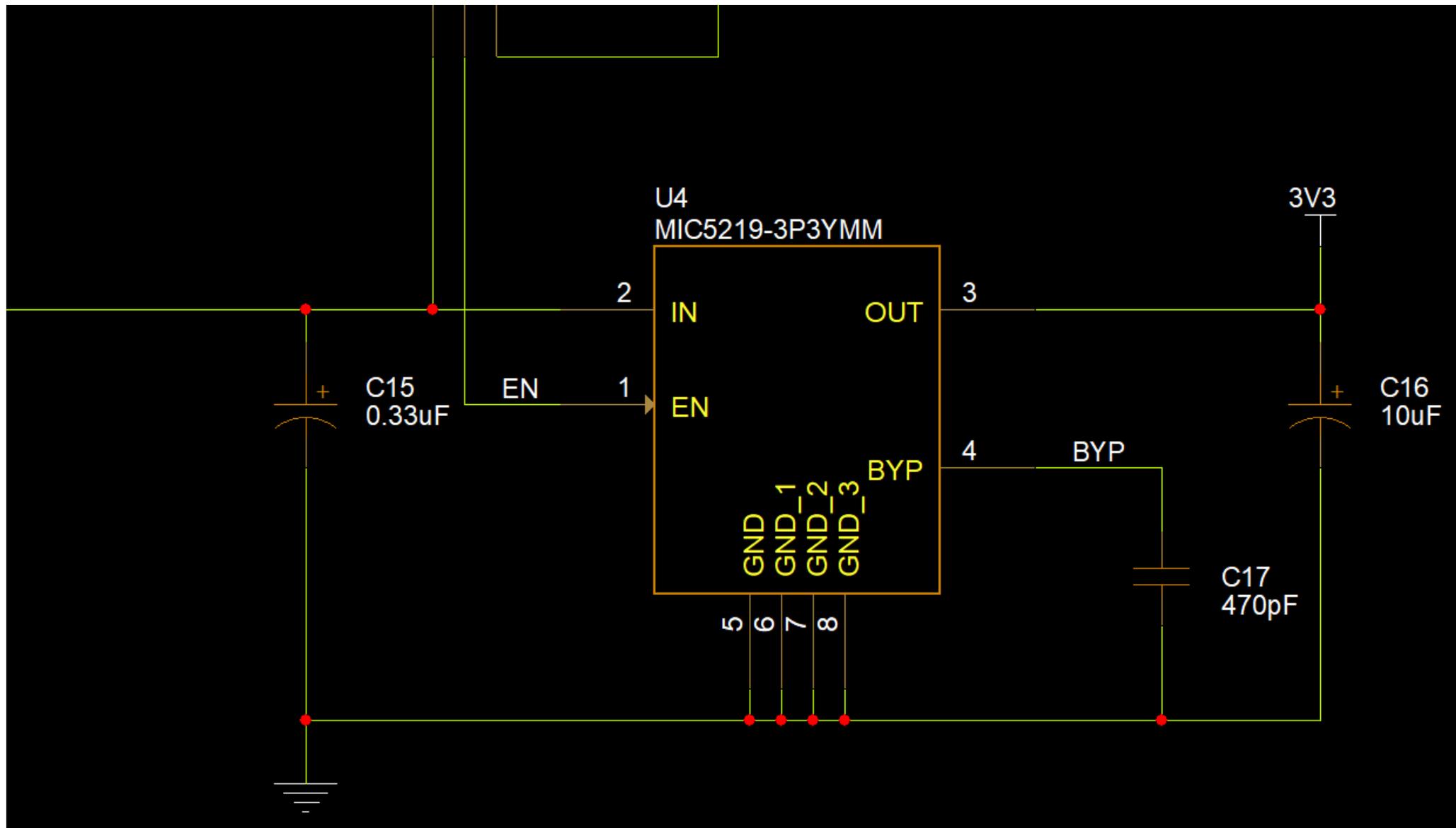


## Dormant Mode

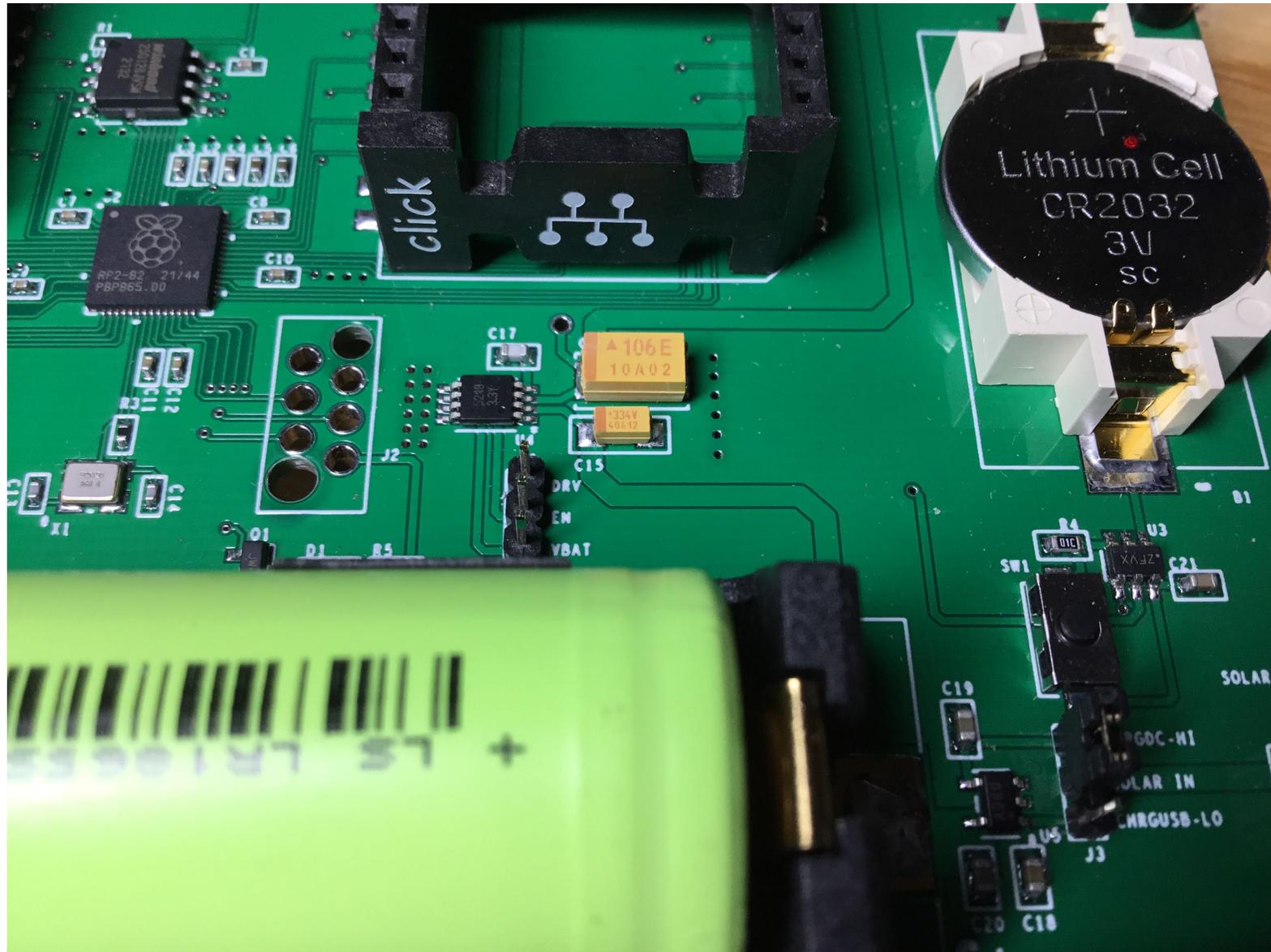
```
C hello_dormant.c
1  #include <stdio.h>
2  #include "pico/stdlib.h"
3  #include "pico/sleep.h"
4
5  int main() {
6      stdio_init_all();
7      printf("Hello Dormant!\n");
8
9      printf("Switching to XOSC\n");
10     uart_default_tx_wait_blocking();
11
12     // UART will be reconfigured by sleep_run_from_xosc
13     sleep_run_from_xosc();
14
15     printf("Running from XOSC\n");
16     uart_default_tx_wait_blocking();
17
18     printf("XOSC going dormant\n");
19     uart_default_tx_wait_blocking();
20
21     // Go to sleep until we see a high edge on GPIO 10
22     sleep_goto_dormant_until_edge_high(10);
23
24     uint i = 0;
25     while (1) {
26         printf("XOSC awake %d\n", i++);
27     }
28
29     return 0;
30 }
```



## Hardware Solution : Main Regulator



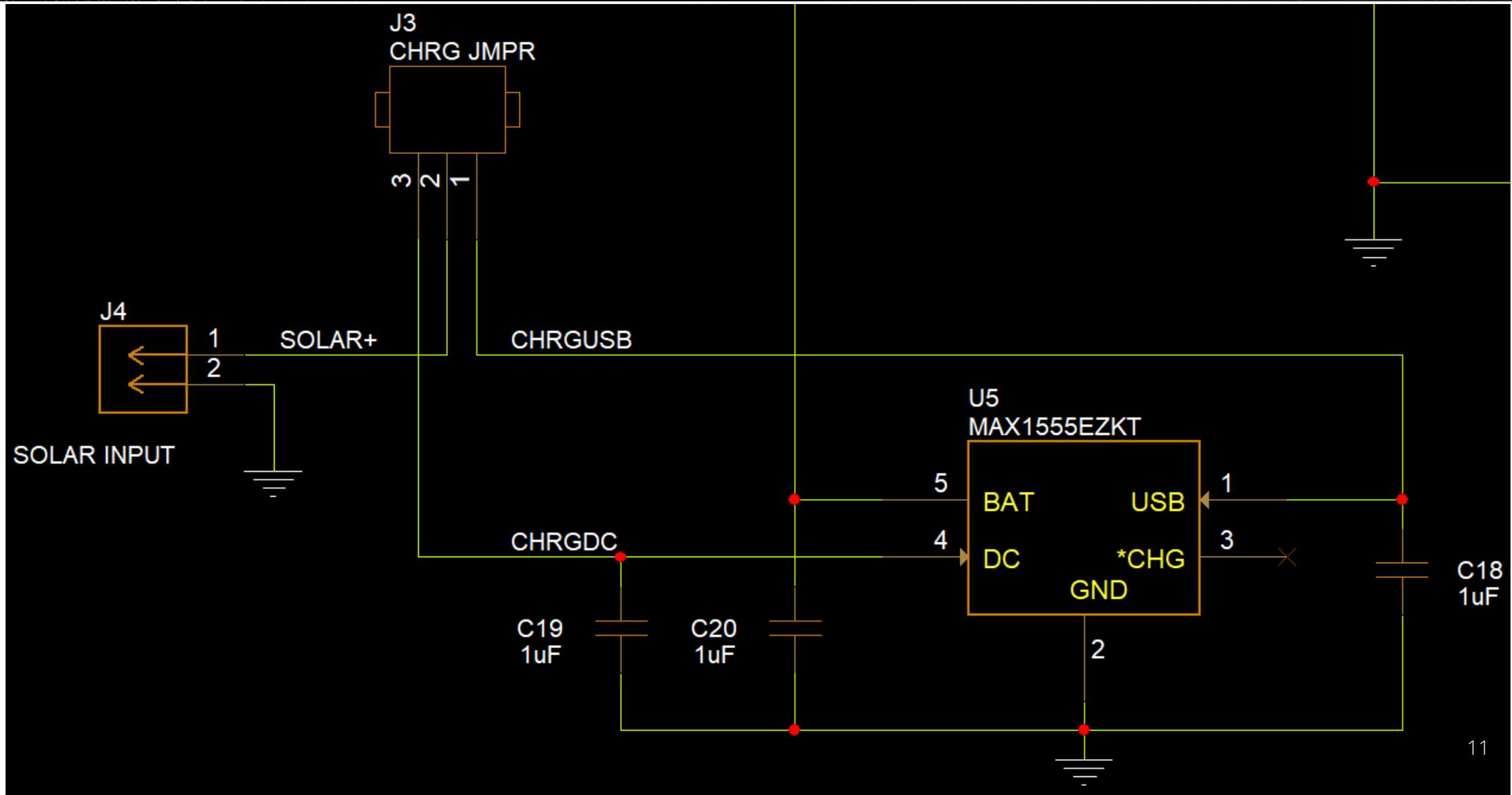
## Hardware Solution : Main Regulator



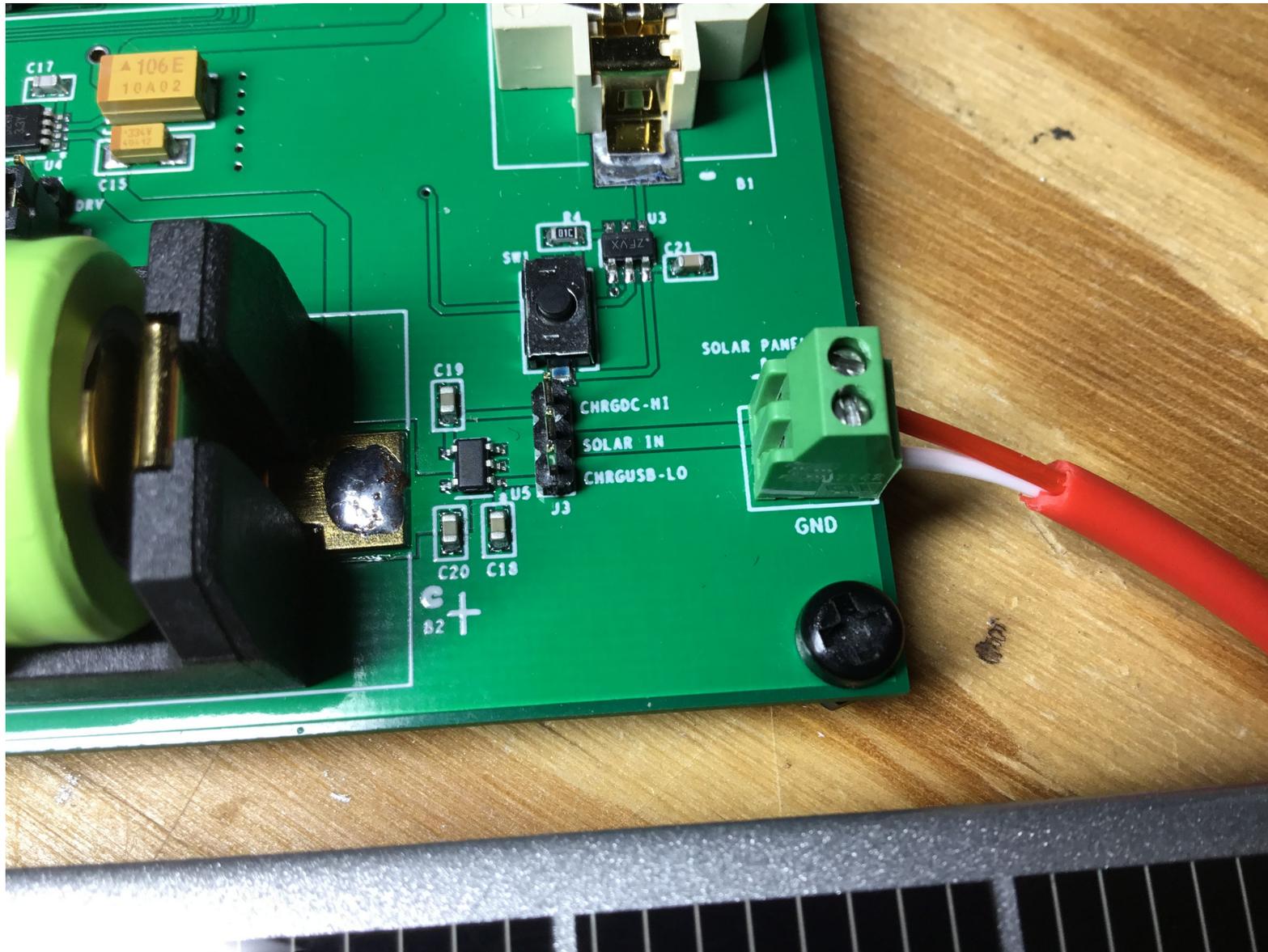
## Hardware Solution : Solar Battery Charging

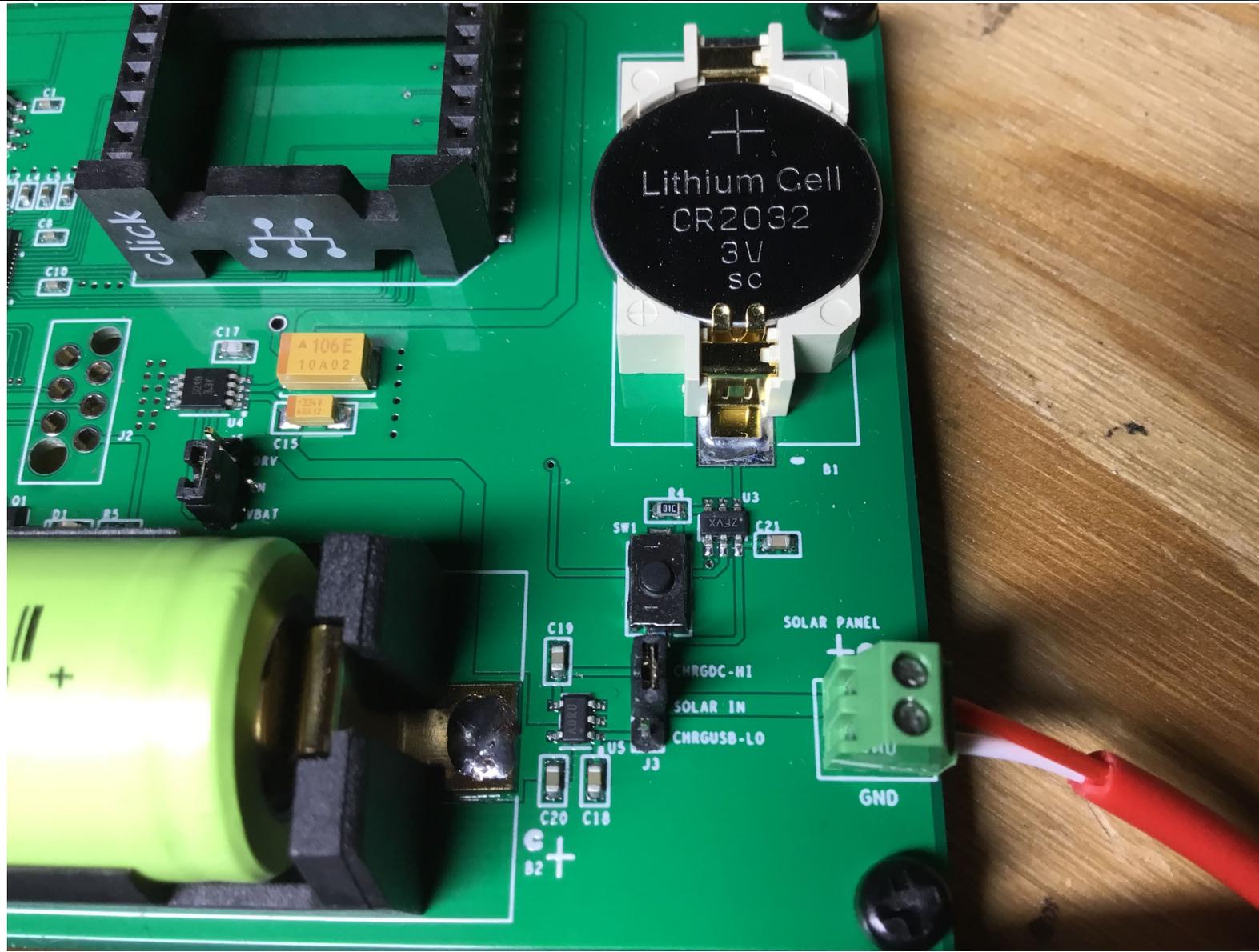


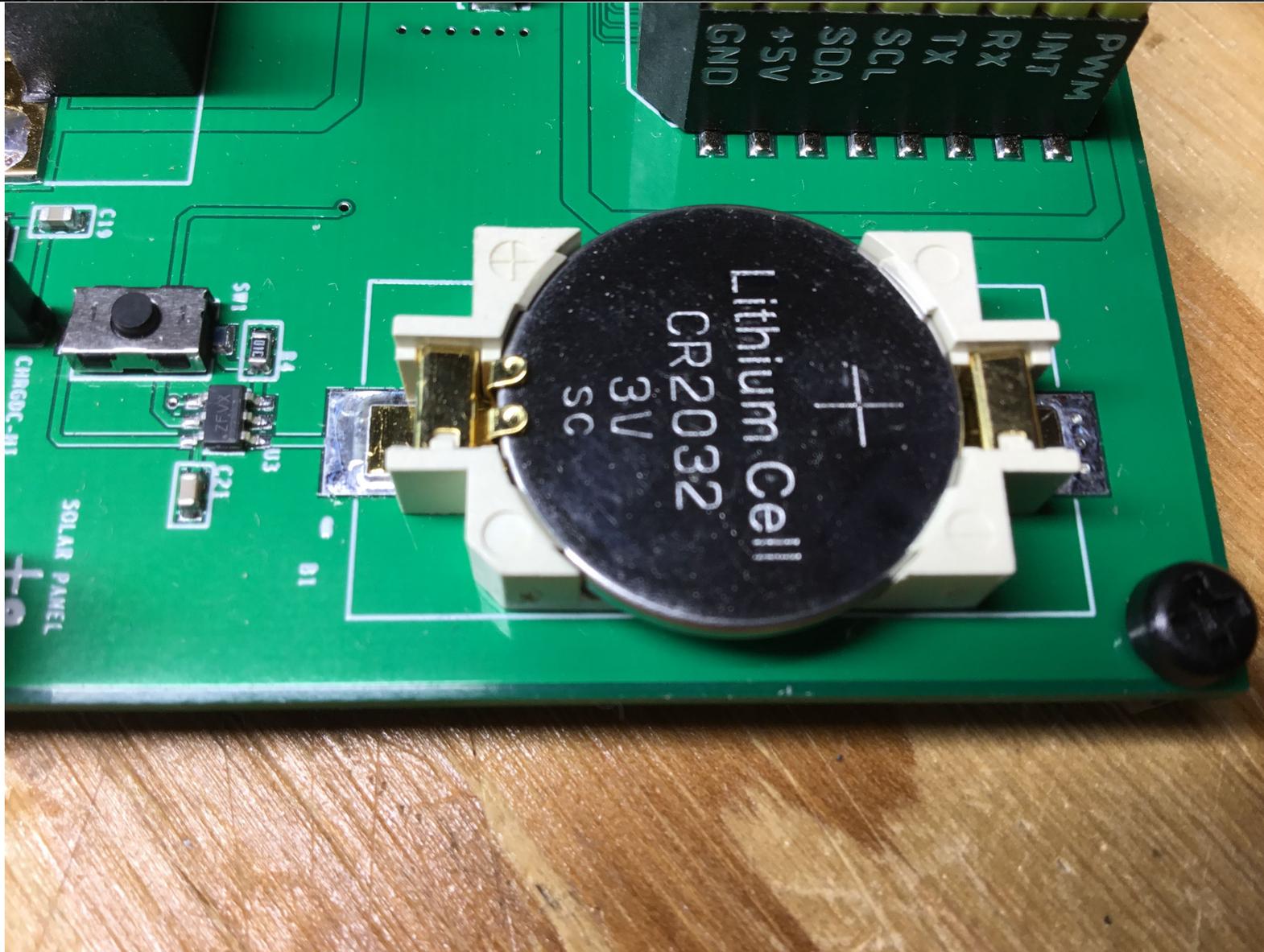
## Hardware Solution : Solar Battery Charging



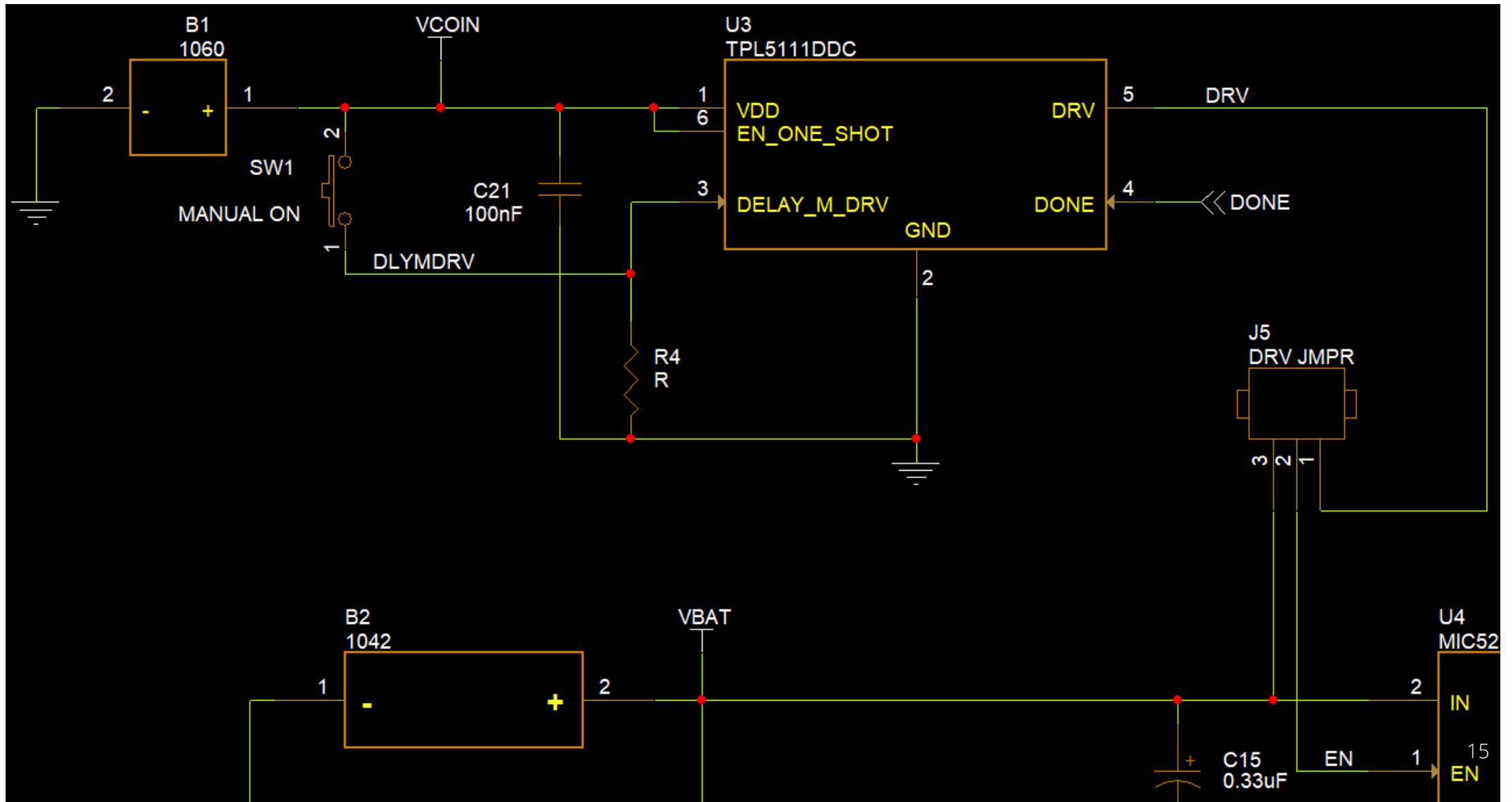
## Hardware Solution : Solar Battery Charging



**Hardware Solution : Solar Battery Charging**

**Hardware Solution : Power Switching**

## Hardware Solution : Power Switching



## Hardware Solution : Power Switching

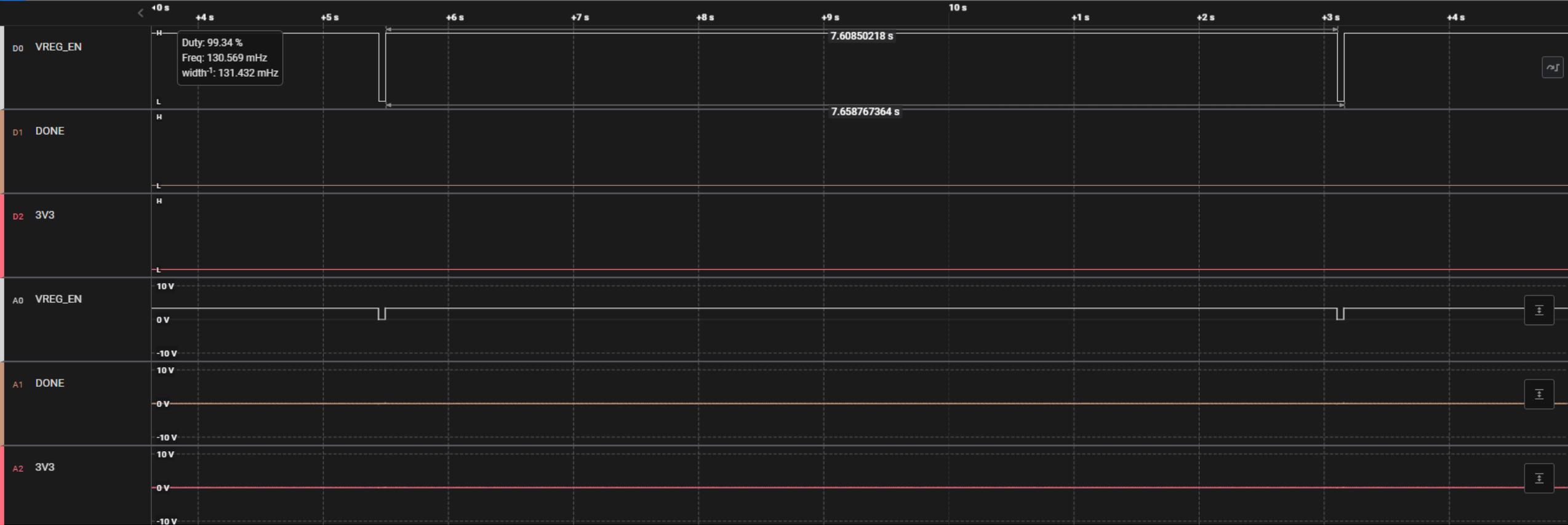
```
C main.c
1  #include <stdio.h>
2  #include "pico/stdlib.h"
3
4  int main()
5  {
6
7      const uint led_pin = 17;
8      const uint done_pin = 16;
9
10     // Initialize LED pin
11     gpio_init(led_pin);
12     gpio_set_dir(led_pin, GPIO_OUT);
13     gpio_init(done_pin);
14     gpio_set_dir(done_pin, GPIO_OUT);
15
16     gpio_put(done_pin, false);
17     gpio_put(led_pin, false);
18
19     // Loop every 8 seconds forever
20     while (true)
21     {
22         // Blink LED
23         gpio_put(led_pin, true);
24         sleep_ms(500);
25         gpio_put(done_pin, true);
26         while(1);
27     }
28 }
29 }
```

```
M CMakeLists.txt
1  # Set minimum required version of CMake
2  cmake_minimum_required(VERSION 3.12)
3
4  # Include build functions from Pico SDK
5  include($ENV{PICO_SDK_PATH}/external/pico_sdk_import.cmake)
6
7  # Set name of project (as PROJECT_NAME) and C/C++ standards
8  project(blinkdone C CXX ASM)
9  set(CMAKE_C_STANDARD 11)
10 set(CMAKE_CXX_STANDARD 17)
11
12 # Creates a pico-sdk subdirectory in our project for the libraries
13 pico_sdk_init()
14
15 # Tell CMake where to find the executable source file
16 add_executable(${PROJECT_NAME}
17     main.c
18 )
19
20 # Create map/bin/hex/uf2 files
21 pico_add_extra_outputs(${PROJECT_NAME})
22
23 # Link to pico_stdlib (gpio, time, etc. functions)
24 target_link_libraries(${PROJECT_NAME}
25     pico_stdlib
26 )
27
28 # Enable usb output, enable uart output
29 pico_enable_stdio_usb(${PROJECT_NAME} 0)
30 pico_enable_stdio_uart(${PROJECT_NAME} 1)
```

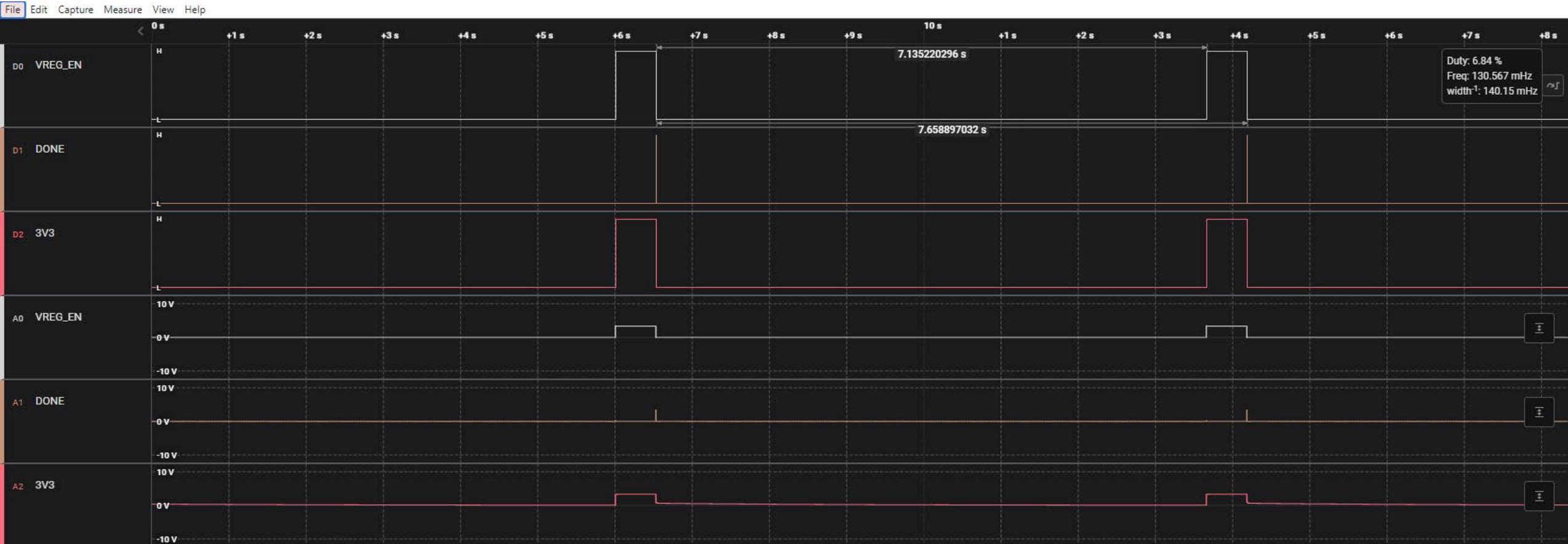
# Hardware Solution : Power Switching

Logic 2 [Logic Pro 16 - Connected] [Session 0]

File Edit Capture Measure View Help



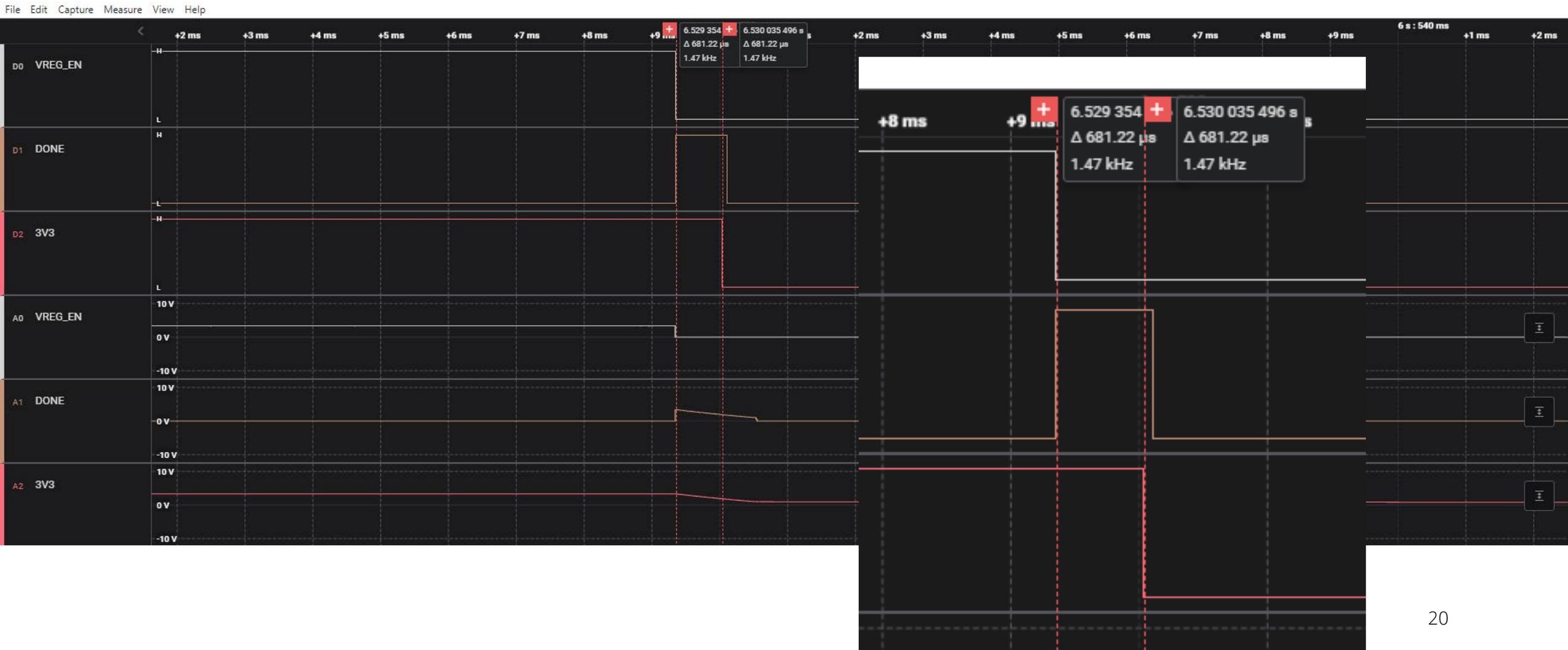
# Hardware Solution : Power Switching



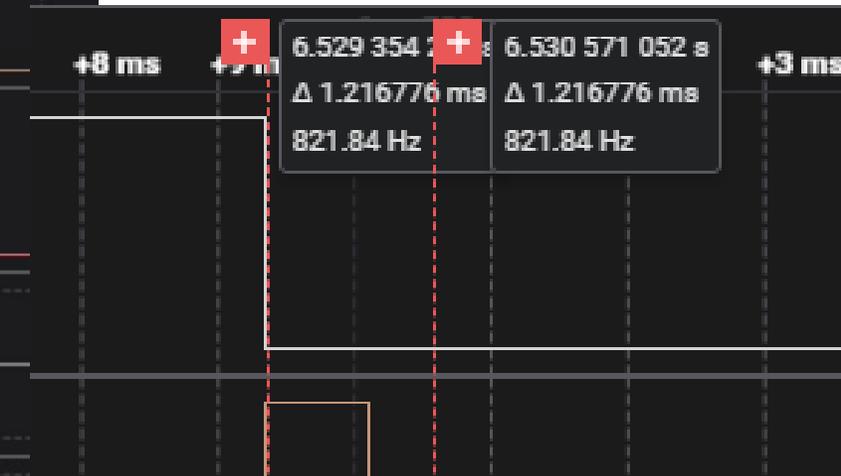
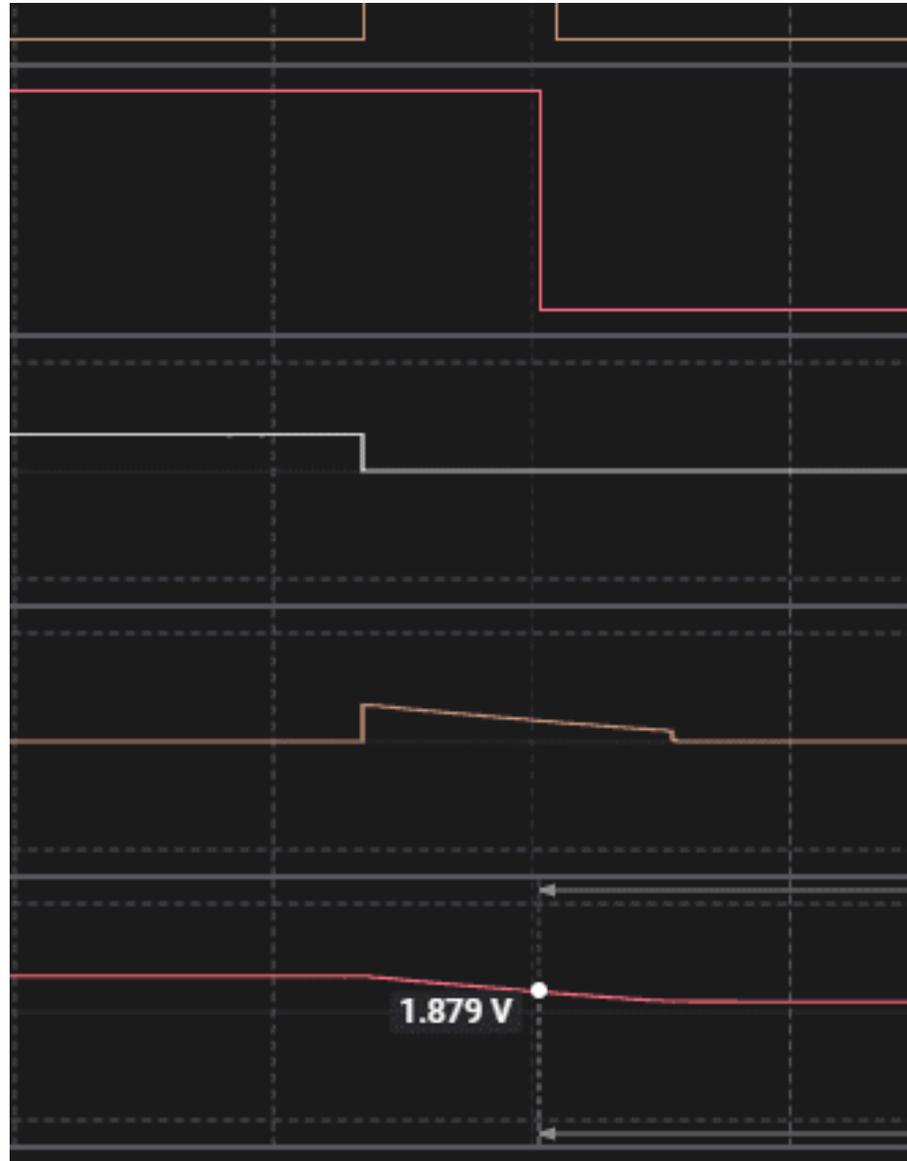
# Hardware Solution : Power Switching



# Hardware Solution : Power Switching



## Hardware Solution : Power Switching

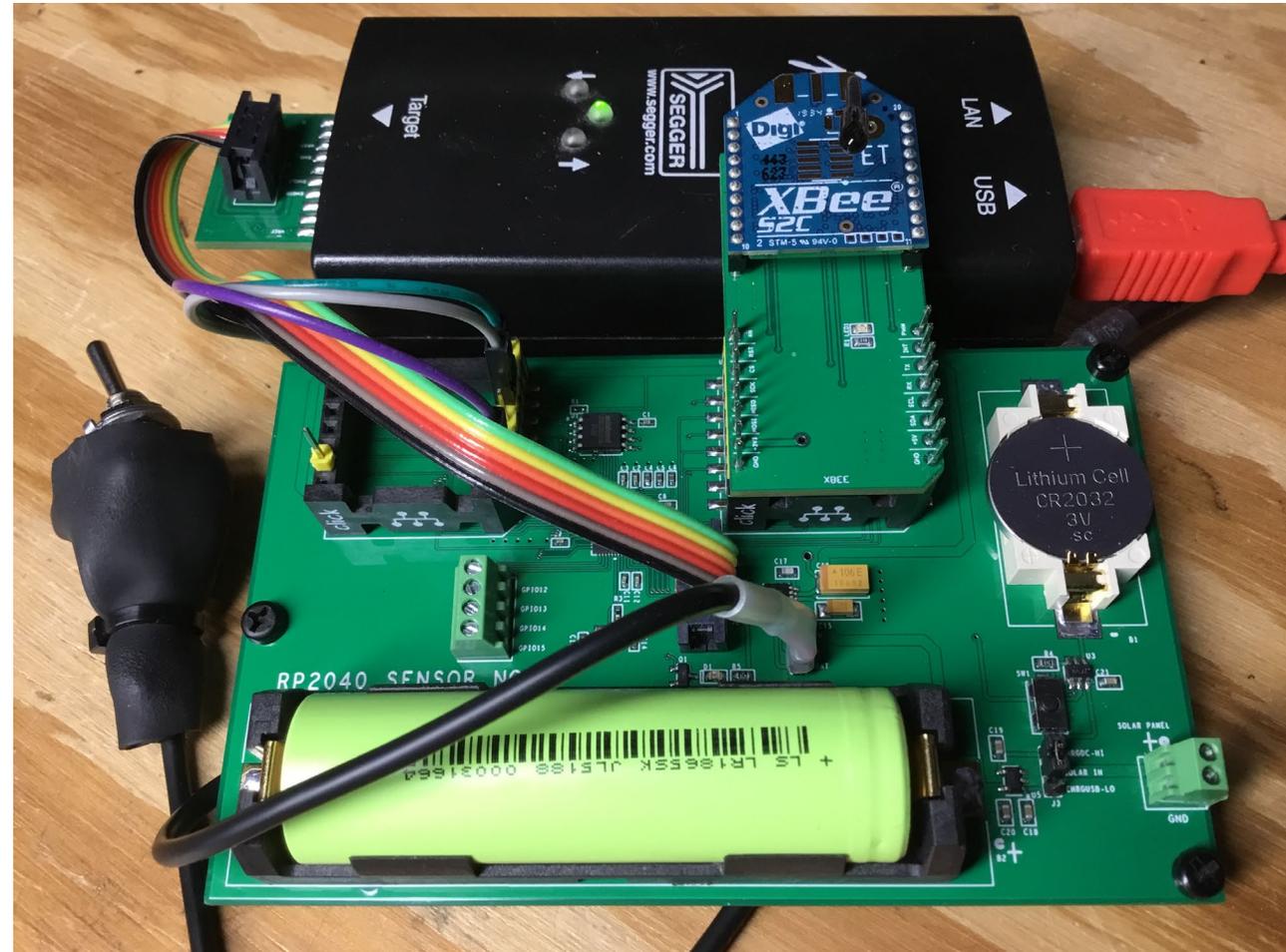


# Thank you for attending!!!

Please consider the resources below:

- [raspberrypi.org](http://raspberrypi.org)
- [RP2040 Data Sheet](#)
- [TLP5111 Data Sheet](#)
- [MIC5219 Data Sheet](#)
- [MAX1555 Data Sheet](#)

**MORE TO COME..**





**DesignNews**

Thank You

Sponsored by

