

# Lab Entry – 2026-01-31

## Metadata

- Date: 2026-01-31
- Project: Off Grid Solar Battery Charger
- Board / Rev: Pico 1
- Scope: Setup the PWM software side. Measure output with Multimeter. Use interrupt on positive edge of PWM to calculate Frequency.

## Objective

Implement a 100 KHz frequency on the raspberry pi pico.

## Setup

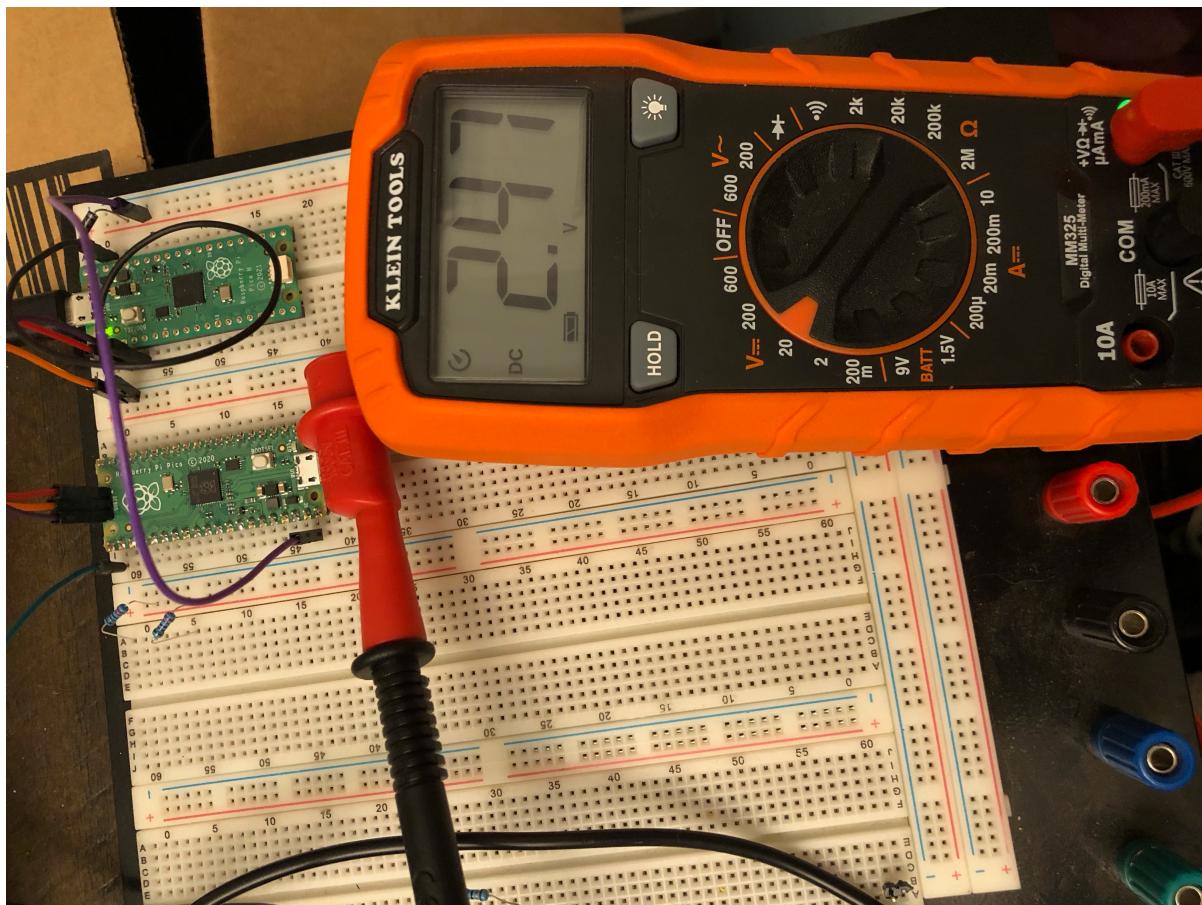


Figure 1: HIL Setup for 75% duty Cycle

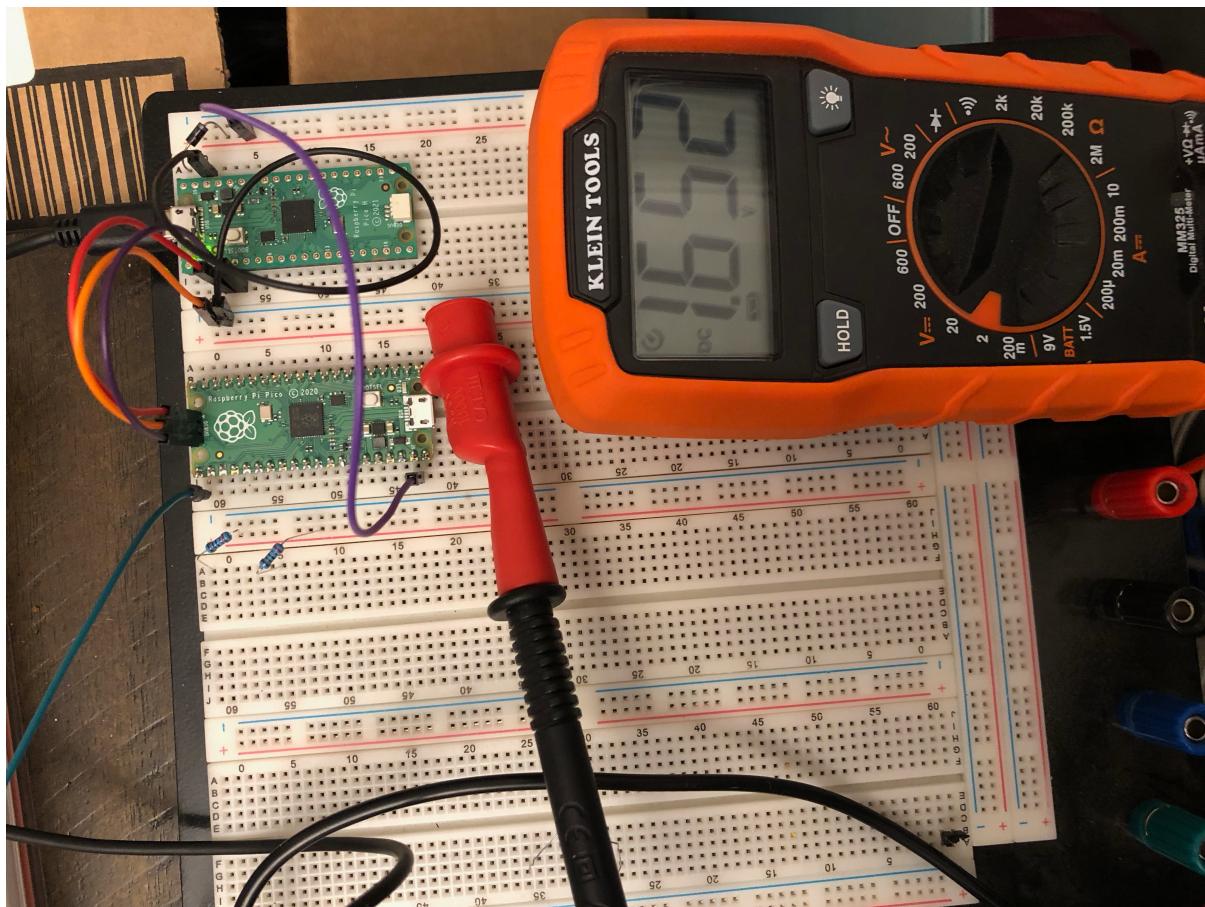


Figure 2: HIL Setup for 50% duty Cycle

**VARIABLES**

- ✓ Local
- > Global
- > Static: ../../Projects/Code/Embedded/
- > Registers

**WATCH**

```
Freq = 1000000
```

**CALL STACK**

- ✓ rp... PAUSED ON USER REQUEST
  - irq\_handler\_chain\_first\_slot@0x100003be <signal handler called>@0xfffff
  - main@0x100003be /Users... [57]
- > rp2040.core1 (Name... PAUSED

```
C PWM_Test.c > ...
1  /*
2  Author: Josh Benner
3  Date: 1/31/26
4
5  Overview: Test the PWM output. Feed PWM output into another pin.
6  Set interrupt to occur on the positive edge of the PWM signal.
7  Get the time stamp when interrupt occurs. Measure the difference
8  between two interrupt events to get the period. Convert period to
9  Frequency. Suspend code with debuffer and look at the Freq Variable.
10
11 We are expecting a Frequency value of 100 Khz. |
12 */
13 #include <stdio.h>
14 #include "pico/stdlib.h"
15 #include "hardware/gpio.h"
16 #include "hardware/pwm.h"
17
18 #define WRAP 642 // count up to
19 #define PWM_GPIO 16 // PWM output pin
20 #define READ_PWM 15 // Read PWM input pin
21 #define EVENT_MASK GPIO0_IRQ_EDGE_RISE // rising edge
22
23 float Duty = 0.5;
24
25 absolute_time_t Current_timeStamp;
26 absolute_time_t Prev_timeStamp;
27 int64_t delta_Time = 0;
28 int64_t Freq = 0;
29 bool first_edge = true;
30
31 // GPIO interrupt callback
32 // Calculate the period of the pwm rising edge and convert that to Frequency
33 void callback(uint gpio, uint32_t events) {
34     Current_timeStamp = get_absolute_time();
35
36     if (!first_edge) {
37         delta_Time = absolute_time_diff_us(Prev_timeStamp, Current_timeStamp);
38         if (delta_Time > 0) {
39             Freq = 10000000 / delta_Time; // Frequency in Hz
40         }
41     } else {
42         first_edge = false; // skip first edge
43     }
44
45     Prev_timeStamp = Current_timeStamp;
46 }
47
48 int main() {
49     stdio_init_all();
50
51     // --- Configure PWM output ---
52     uint Slice_Num = pwm_gpio_to_slice_num(PWM_GPIO);
53     uint Channel = pwm_gpio_to_channel(PWM_GPIO);
54
55     gpio_set_function(PWM_GPIO, GPIO_FUNC_PWM);
56     pwm_set_wrap(Slice_Num, WRAP);
57     pwm_set_chan_level(Slice_Num, Channel, (uint)(WRAP * Duty));
58     pwm_set_phase_correct(Slice_Num, true);
59     pwm_set_enabled(Slice_Num, true);
60
61     // --- Configure input pin with IRQ ---
62     gpio_init(READ_PWM);
63     gpio_set_dir(READ_PWM, GPIO_IN);
64     gpio_pull_down(READ_PWM); // avoid floating input
65
66     gpio_set_irq_enabled_with_callback(READ_PWM, EVENT_MASK, true, &callback);
67
68     // --- Main loop ---
69     while (true) {
70         if (delta_Time > 0) {
71             printf("PWM period: %lld us, Frequency: %lld Hz\n", delta_Time, Freq);
72             sleep_ms(500); // slow down prints
73         }
74         tight_loop_contents();
75     }
76 }
```

Figure 3: Software running the HIL Test- Verifies 100 KHz frequency

## Observations

Looking at Figure 1, we can see that the multimeter reads 2.47. The duty is set to 75%. We expect  $3.3 \cdot 0.75 = 2.475V$ .

Looking at Figure 2, we can see that the multimeter is reading 1.652 V. The duty cycle is set to 50%, so we expect  $3.3 \cdot 0.5 = 1.65V$ .

Looking at Figure 3, we see that the program is stopped on line 78, and Freq is 1000000.

## Conclusions / Next Steps

We have successfully verified the operation and control of the PWM on the Raspberry Pi Pico. We now need to set up the Gate Drive circuit, and verify our PWM output works on that by using a multimeter. Verifying the frequency will require an oscilloscope, due to running the gate drive circuit at a higher voltage than the Pico can tolerate.