Lab Entry – 2026-01-11

Metadata

- Date: 2026-01-11
- Project: Offgrid solar battery charger
- Board / Rev:
- Scope: HIL Test INA219 Breakout board

Objective

Verify that the INA219 Breakout Board works by seeing if it ACKS a request to write to it.

Setup

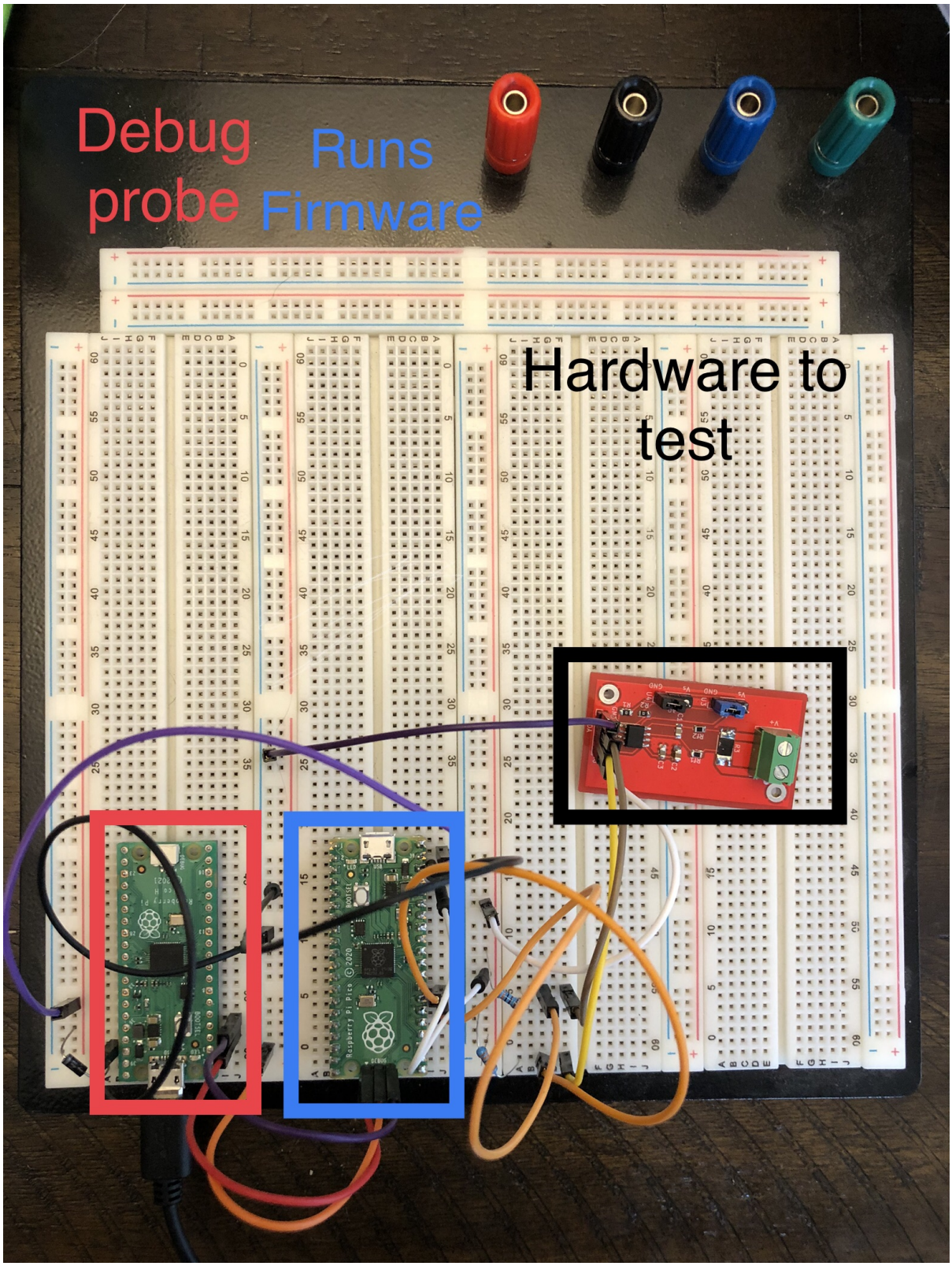

Figure 1: HIL Setup

We will use the i2c api provided by Raspberry pi to write to the hardware to be tested. The specific function used will be i2c_write_blocking, which returns the number of bytes written, or PICO_ERROR_GENERIC if address not acknowledged, no device present.

We set the addrees of the PCB to be 0x40 (A0 and A1 = GND).

If the INA219 PCB is working as expected, we should see i2c_write_blocking return a 0 (this is equivalent to ACKing a write request of 0 bytes).

```c
/*
*   Author: Josh Benner
*   INA219 HIL test
*
*   If ACK_Test returns 0 the HIL test is considered passed at this point.
*   If ACK_Test return -1 then the request was NACKed and the test is failed.
*
*   TODO:   Create a more robust test, Come up with a simple circuit and calculate
*           expected values in each register. Verify results.
*
*/
#include <stdio.h>
#include "pico/stdlib.h"
#include "hardware/i2c.h"

// I2C defines
// This example will use I2C0 on GPIO8 (SDA) and GPIO9 (SCL) running at 400KHz.
// Pins can be changed, see the GPIO function select table in the datasheet for information on GPIO assignments
#define I2C_PORT i2c0
#define I2C_SDA 4
#define I2C_SCL 5
#define CALIBRATION 0xFFFE
#define CURRENT_LSB

//PCB Deice
#define ADDRESS 0x40

typedef enum{
 CONFIG_REG = 0x00,
 SHUNT_VOL_REG,
 BUS_VOL_REG,
 POWER_REG,
 CURRENT_REG,
 CALIBRATION_REG
} INA219_REGS_E;









int main()
{
    uint8_t datarx;
    stdio_init_all();
    int ACK_Test;;
    // I2C Initialisation. Using it at 400Khz.
    i2c_init(I2C_PORT, 400*1000);

    gpio_set_function(I2C_SDA, GPIO_FUNC_I2C);
    gpio_set_function(I2C_SCL, GPIO_FUNC_I2C);
    gpio_pull_up(I2C_SDA);
    gpio_pull_up(I2C_SCL);
    // For more examples of I2C use see https://github.com/raspberrypi/pico-examples/tree/master/i2c
    ACK_Test=i2c_write_blocking(I2C_PORT, ADDRESS, NULL, 0, false);

    if(ACK_Test==0){
        printf("Address: %u was ACKed\n",ADDRESS);
    }else{
        printf("Address: %u was NACKED",ADDRESS);
    }
}
```

Figure 2: Firmware Runing The Test

Observations

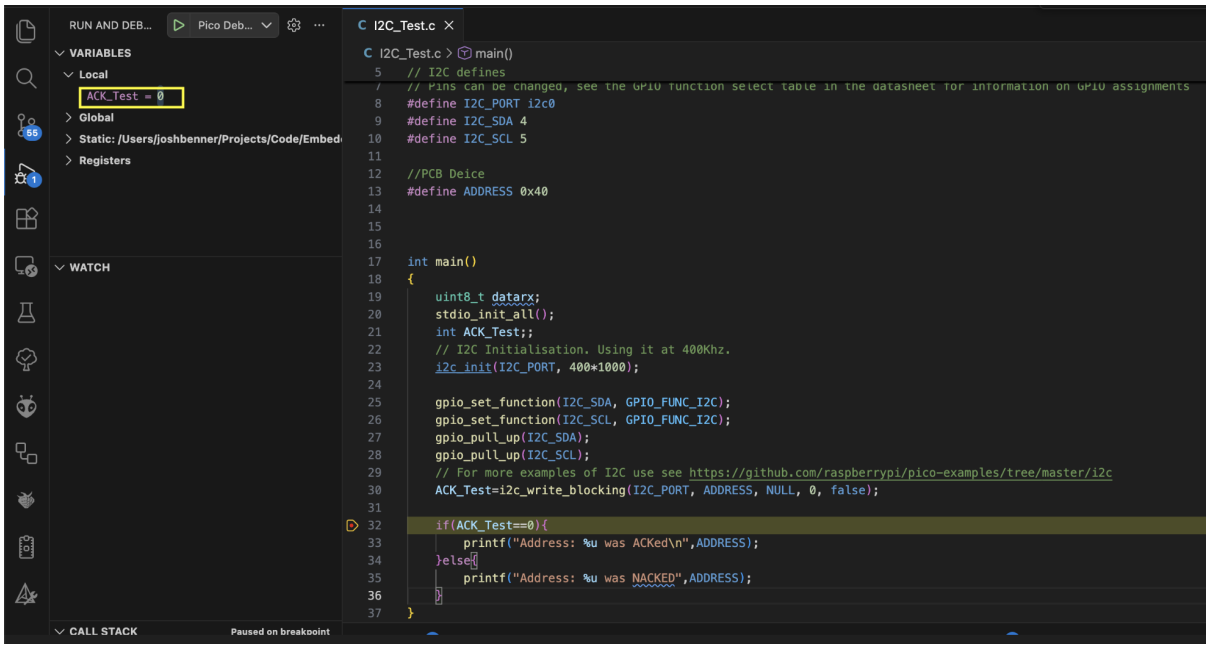

Figure 3: Test Result

Figure 3 is a snap shot of the debug process of running the firmware test. We have a breakpoint set right after ACK_Test has been assigned by i2c_write_blocking function. If we look at the yellow box at the top left corn, it shows that ACK_Test reult is 0 and thus the INA219 PCB acknowleged the request to write 0 bytes to it. Therefore This pcb has passed this simple test.

Next Steps

Provide a more robust test. Use a 9 V battery and 100 ohm resistor. Calculate expected values in registers and then compare the expected result to the actual result. Considere error tollerances and if the results are within reason, the PCB will be considered functional.