Measuring voltage, current and power with INA226

By jean-claude.feltes@education.lu

1. Purpose

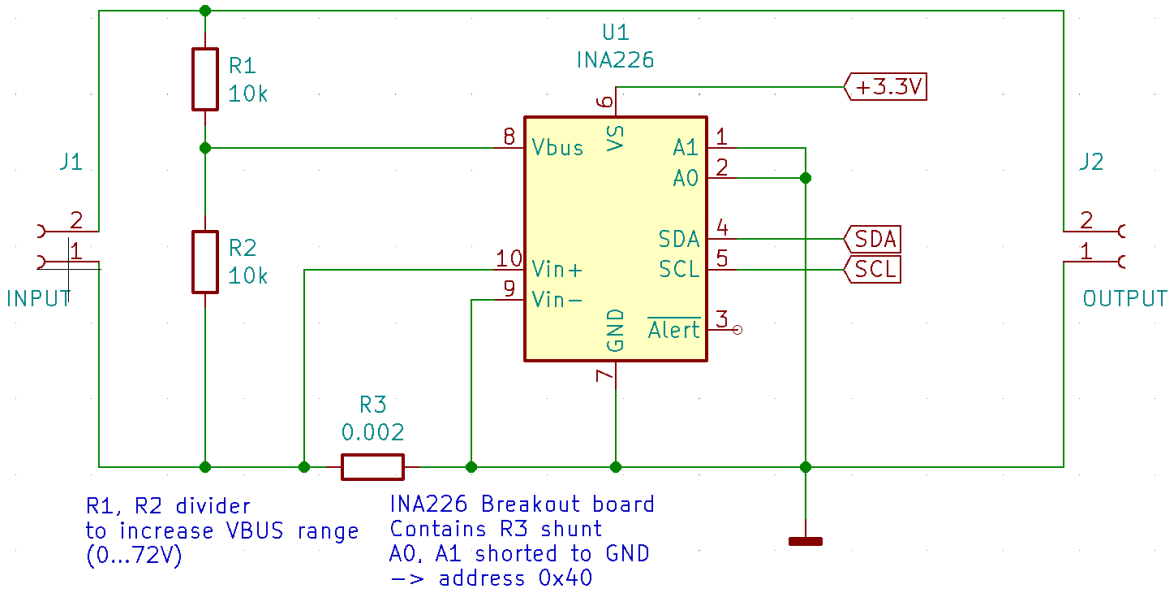
This device shall measure voltage, current and power

- for 2 solar panels connected in series, or
- for an 48V battery charged by a stepup power supply feeded by these panels

From this, the range for current and voltage is derived:

Vmax = min. 60V, Imax = 15A

2. Circuit



Fortunately there is a breakout board for the INA226 that is only 3mm x 3mm in size.
The schematic shows a simplified circuit for the brakout board.

I added a voltage divider to get a larger voltage range 0...72V.

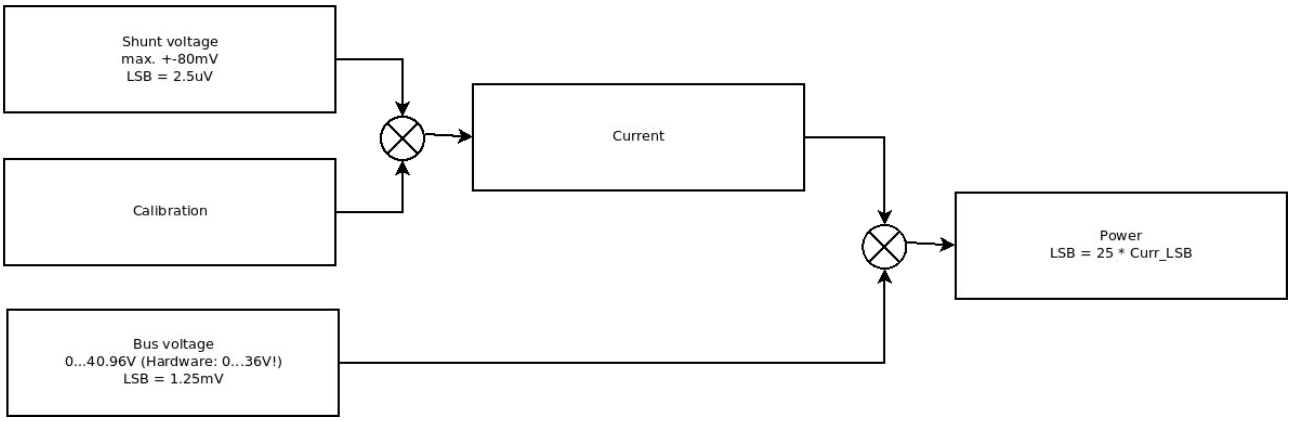
Will the shunt survive at Imax?

$$P_{Smax} = R_S \cdot I^2_{max} = 0.45W$$

Could be too much for a longer time, let's hope the copper areas on the board will help with the cooling!

The INA226 is connected to the I2C bus of a Raspi Pico.

3. Internal function of the INA226



The shunt voltage is read with a resolution of 2.5µV, with a bipolar range of 80mV.

From this the currentregister value is deduced by multiplying with the calibration register and then dividing by 2048. So the current register value is CAL * Vshunt.

The bus voltage is read with a resolution of 1.25mV and a theoretical range of 40.96V.
Note that the hardware only supports a range of 36V!

Example:

With a shunt resistor Rs = 2mΩ and a current I = 1A, we have a shunt voltage Us = Rs * I = 2mV. This gives us a shunt register value of Us / 2.5µV = 2000/2.5 = 800.

We could also say that for this shunt register the shunt register has a characteristic of 800 counts per Ampère.

In general, the S**hunt register value** is $\boxed{SR = \frac{I \cdot R_S}{2.5\,\mu V}}$ (1)

With this equation we can calculate the theoretical limits of operation: maximum current and bit resolution for the shunt register:

$$I = \frac{SR \cdot 2.5\,\mu V}{Rs} \quad \text{so} \quad I_{max} = \frac{32768 \cdot 2.5\,\mu V}{2\,m\Omega} = 40.96\,A \quad \text{(from what fits into the register)}$$

$$\text{and} \quad I_{1\,bit} = \frac{1 \cdot 2.5\,\mu V}{2\,m\Omega} = 1.25\,mA$$

The **Current register value** is calculated from the Shunt register value as $\boxed{CR = \frac{SR \cdot CAL}{2048}}$ (2)

If we want to know the CAL register value that gives us a current register value in mA, we can use for the example with I = 1A = 1000mA (2):

$$CAL = \frac{CR \cdot 2048}{SR} = \frac{1000 \cdot 2048}{800} = 2560$$

With this calibration the Current register would directly give the current in mA.

The current_LSB value would be 1mA.
Thus the maximum value the current register could hold would be 32.768A.

The power_LSB is always 25 times the current_LSB, so 25mW for our example.
This resolution is not quite as good as the one resulting from using the values of shunt register and bus voltage register directly, as that would give 1.25mV * 1.25mA = 2.25mW.

Is it better to use only the shunt and bus voltage registers?

When a powerful microcontroller is reading the INA226, and when timing is not a problem it could be wiser to use only those values and let the controller calculate current and power values.

The power resolution would be better, and there would be no fuzzing with the calibration values. There could be no overflow problem for these values.

On the other hand, the internal calculations of the INA226 seem to be very fast, so using the current and power register would be advantageous for time critical measurements.

And using the alert pin could be of interest.

4. Micropython code

My Micropython library

https://github.com/jean-claudeF/INA226

offers both possibilities.

Example code:

```
import ina226_jcf as ina226
from machine import Pin, I2C
import time

i2c = I2C(0,scl=Pin(9), sda=Pin(8), freq=100000)
ina = ina226.INA226(i2c, 0x40, Rs = 0.002, voltfactor = 2)

i=0
while True:

    # Get voltage, current and power by using only Bus and Shunt voltage
registers
    # This is easier, should be with best accuracy and no register overflow
problems:
    v, i, p = ina.get_VIP()
    V = '%2.3f' % v
    I = '%2.3f' % i
    P = '%2.3f' % p
    print(i, '\t', V, '\t', I, '\t',  P)
```