EE 49
Electronics for IoT

Microcontroller

1st Project: characterize solar cell
Goal

Solar Cell Power versus Load Resistance

- Current [mA]
- Voltage [V*100]
- Power [mW]

Resistance [Ohm]
Plan
Measure Solar I, V with MCU
Web Search

Image of a web search for "microcontroller current sensor" showing sponsored results with images of different current sensors and their prices. The search results include links to various websites and diagrams.
INA 219 Manufacturer Website

INA219 (ACTIVE)
26-V, Bidirectional, Zero-Drift, High-Side, I2C Out Current/Power Monitor

Description
The INA219 is a current shunt and power monitor with an I2C- or SMBUS-compatible interface. The device monitors both shunt voltage drop and bus supply voltage, with programmable conversion times and filtering. A programmable calibration value, combined with an internal multiplier, enables direct readouts of current in amperes. An additional multiplying register calculates power in watts. The I2C- or SMBUS-
INA219 Datasheet

1 Features

- Senses Bus Voltages from 0 to 26 V
- Reports Current, Voltage, and Power
- 16 Programmable Addresses
- High Accuracy: 0.5% (Maximum) Over Temperature (INA219B)
- Filtering Options
- Calibration Registers
- SOT23-8 and SOIC-8 Packages

2 Applications

- Servers
- Telecom Equipment
- Notebook Computers
- Power Management
- Battery Chargers
- Welding Equipment
- Power Supplies
- Test Equipment

3 Description

The INA219 is a current shunt and power monitor with an I²C- or SMBUS-compatible interface. The device monitors both shunt voltage drop and bus supply voltage, with programmable conversion times and filtering. A programmable calibration value, combined with an internal multiplier, enables direct readouts of current in amperes. An additional multiplying register calculates power in watts. The I²C- or SMBUS-compatible interface features 16 programmable addresses.

The INA219 is available in two grades: A and B. The B grade version has higher accuracy and higher precision specifications.

The INA219 senses across shunts on buses that can vary from 0 to 26 V. The device uses a single 3- to 5.5-V supply, drawing a maximum of 1 mA of supply current. The INA219 operates from –40°C to 125°C.

Device Information(1)

<table>
<thead>
<tr>
<th>PART NUMBER</th>
<th>PACKAGE</th>
<th>BODY SIZE (NOM)</th>
</tr>
</thead>
<tbody>
<tr>
<td>INA219</td>
<td>SOIC (8)</td>
<td>3.91 mm × 4.90 mm</td>
</tr>
<tr>
<td></td>
<td>SOT-23 (8)</td>
<td>1.63 mm × 2.90 mm</td>
</tr>
</tbody>
</table>

(1) For all available packages, see the orderable addendum at the end of the data sheet.
INA219 “Simplified Schematic”
INA219 Configured

Figure 13. INA219 Configured for Shunt and Bus Voltage Measurement
Let’s Redraw this a little …
Real INA219

- This thing is tiny
- “big” pepper corn
Hone your soldering skills …
Or get a breakout board …

- https://www.adafruit.com/product/904
- In your “goodies” bag …
Connect to MCU

- \( \text{I}^2\text{C}: \) just 4 wires:
  - Data (SDA)
  - Clock (SCL)
  - 3.3V supply (Huzzah32 generates this)
  - GND
Huzzah32

GPIO  ALT  µPy

RESET  3.3V

GND

26 DAC2  A0
25 DAC1  A1
34 ADC6  A2
39 ADC3  A3
36 ADC0  A4

4  A5
5  SCK  A16
18 MOSI A17
19 MISO  
16  A19
17  A20
21  A21

µPy  ALT  GPIO

28 VBAT
27 EN 3.3V
26 VUSB

25 A12  LED  13
24 A11  BOOT  12
23 A10
22 A9  ADC5  33
21 A8
20 A7  ADC4  32
19 A6
18 A15  SCL  22
17 A14  SDA  23
I²C
I2C Communication

![I2C Communication Diagram](image)
### INA219 I²C Commands

4. Complement the binary result: 000 0010 1111 1111
5. Add 1 to the Complement to create the Two’s Complement formatted result → 000 0011 0000 0000
6. Extend the sign and create the 16-bit word: 1000 0011 0000 0000 = 8300h (Remember to extend the sign to all sign-bits, as necessary based on the PGA setting.)

At PGA = /8, full-scale range = ±320 mV (decimal = 32000). For $V_{\text{SHUNT}} = +320$ mV, Value = 7D00h; For $V_{\text{SHUNT}} = -320$ mV, Value = 8300h; and LSB = 10μV.

![Figure 20. Shunt Voltage Register at PGA = /8](image)

At PGA = /4, full-scale range = ±160 mV (decimal = 16000). For $V_{\text{SHUNT}} = +160$ mV, Value = 3E80h; For $V_{\text{SHUNT}} = -160$ mV, Value = C180h; and LSB = 10μV.

![Figure 21. Shunt Voltage Register at PGA = /4](image)

At PGA = /2, full-scale range = ±80 mV (decimal = 8000). For $V_{\text{SHUNT}} = +80$ mV, Value = 1F40h; For $V_{\text{SHUNT}} = -80$ mV; Value = E0C0h; and LSB = 10μV.
Someone has already done the work!

Micropython library for the TI INA219 voltage/current sensor ... https://forum.micropython.org › ... › Drivers for External Components
May 16, 2017 - https://github.com/chrisb2/pyb_ina219. I have written this library based on one I wrote for the Raspberry Pi. It supports the INA219 voltage, current and power monitor sensor from Texas Instruments. The intent of the library is to make it easy to use the quite complex functionality of this sensor. Its currently ...

MicroPython on ESP32 with SPIRAM support - Page 25
Lolin32 Battery State
boot.py not executed
Micropython driver for TI INA219?
More results from forum.micropython.org

GitHub - chrisb2/pyb_ina219: This library for the MicroPython makes it ... https://github.com/chrisb2/pyb_ina219
pyb_ina219 - This library for the MicroPython makes it easy to leverage the complex functionality of the Texas Instruments INA219 sensor to measure voltage, current and power.
You've visited this page 4 times. Last visit: 1/12/18
INA219 on Github

This library for the MicroPython makes it easy to leverage the complex functionality of the Texas Instruments INA219 sensor to measure voltage, current and power.

- micropython
- ina-219
- pyboard
- esp8266
- esp32

- 30 commits
- 1 branch
- 0 releases
- 1 contributor
- MIT
INA219 Driver Usage Instructions

Usage

If you want to give it a try then copy `ina219.py` onto the flash drive of your pyboard, connect the sensor to the I2C(1) or I2C(2) interfaces on the pyboard, then from a REPL prompt execute:

```python
from ina219 import INA219
from machine import I2C

I2C_INTERFACE_NO = 2
SHUNT_OHMS = 0.1

ina = INA219(SHUNT_OHMS, I2C(I2C_INTERFACE_NO))
ina.configure()
print("Bus Voltage: %.3f V" % ina.voltage())
print("Current: %.3f mA" % ina.current())
print("Power: %.3f mW" % ina.power())
```

See next page

https://github.com/chrisb2/pyb_ina219

Depends on MicroPython port
INA219 Example

```python
from ina219 import INA219
from machine import I2C, Pin
from board import SDA, SCL
import time

i2c = I2C(id=0, scl=Pin(SCL), sda=Pin(SDA), freq=100000)

# optional: detect all devices connected to I2C bus
print("scanning I2C bus ...")
print("I2C:", i2c.scan())

# initialize INA219
SHUNT_RESISTOR_OHMS = 0.1
ina = INA219(SHUNT_RESISTOR_OHMS, i2c)
ina.configure()

# read measurements
while True:
    v = ina.voltage()
    i = ina.current()
    p = ina.power()
    print("V = {:.6f}, I = {:.6f}, P = {:.6f}".format(v, i, p))
time.sleep(0.5)
```
Circuit

- Solar cell characterization
- Components:
  1.
  2.
  3.
  4.
Approach

- Symbols
- Circuit diagram
- Optional: wiring diagram
INA219 Breakout Board
Huzzah32
Wiring Diagram
Complete Circuit
## Testing

<table>
<thead>
<tr>
<th>V</th>
<th>I</th>
<th>P</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.13V</td>
<td>1.97mA</td>
<td>10mW</td>
<td>2606.30hm</td>
</tr>
<tr>
<td>5.16V</td>
<td>2.10mA</td>
<td>11mW</td>
<td>2461.90hm</td>
</tr>
<tr>
<td>5.18V</td>
<td>2.60mA</td>
<td>13mW</td>
<td>1989.50hm</td>
</tr>
<tr>
<td>5.14V</td>
<td>3.54mA</td>
<td>17mW</td>
<td>1454.50hm</td>
</tr>
<tr>
<td>5.11V</td>
<td>5.16mA</td>
<td>26mW</td>
<td>990.10hm</td>
</tr>
<tr>
<td>5.12V</td>
<td>9.43mA</td>
<td>49mW</td>
<td>542.80hm</td>
</tr>
<tr>
<td>5.13V</td>
<td>13.40mA</td>
<td>69mW</td>
<td>382.70hm</td>
</tr>
<tr>
<td>5.13V</td>
<td>16.16mA</td>
<td>82mW</td>
<td>317.50hm</td>
</tr>
<tr>
<td>5.08V</td>
<td>19.63mA</td>
<td>98mW</td>
<td>258.70hm</td>
</tr>
<tr>
<td>5.05V</td>
<td>24.76mA</td>
<td>126mW</td>
<td>204.00hm</td>
</tr>
<tr>
<td>5.07V</td>
<td>31.23mA</td>
<td>160mW</td>
<td>162.20hm</td>
</tr>
<tr>
<td>5.04V</td>
<td>42.50mA</td>
<td>216mW</td>
<td>118.60hm</td>
</tr>
<tr>
<td>4.38V</td>
<td>110.48mA</td>
<td>446mW</td>
<td>39.60hm</td>
</tr>
<tr>
<td>0.16V</td>
<td>135.02mA</td>
<td>19mW</td>
<td>1.20hm</td>
</tr>
<tr>
<td>0.23V</td>
<td>140.01mA</td>
<td>26mW</td>
<td>1.60hm</td>
</tr>
<tr>
<td>0.13V</td>
<td>144.61mA</td>
<td>15mW</td>
<td>0.90hm</td>
</tr>
<tr>
<td>0.12V</td>
<td>149.01mA</td>
<td>18mW</td>
<td>0.80hm</td>
</tr>
</tbody>
</table>
How can we plot the result?

- Connect plotter to ESP32?
  - And run Matlab or Excel?

- Better solution
  - Do the plotting on a computer that’s made for this (e.g. laptop)
  - How do we get the data there?
Summary

• IoT Application
  – Circuits
  – Python
  – Internet …

• Sensors
  – INA219
  – I2C
  – Driver

• Prototyping
  – Wiring
  – Electrical and software testing