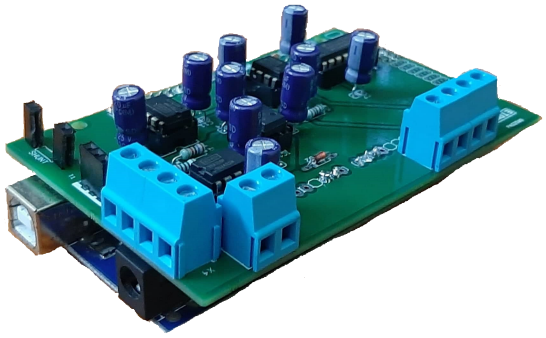
Ultra Fast Current-Voltage (I-V) Curve Scanner

User Manual



Contents

1 Introduction **1**

2 Hardware description **2**

3 Software description **3**

4 How to operate? **3**

5 Data manipulation **5**

1 Introduction

This circuit is employed for tracing ultra-fast I-V curves for power semiconductor devices. While tracing I-V curve traditionally, the temperature of device under test(DUT) rises due to supply of finite current in small duration of time. This leads to wrong I-V measurements at a given temperature for particular voltage/current values.
This circuit operates on the principle of pulsed I-V measurement technique and uses Arduino mega 2560 as a key development board. Further optimizations are executed and the standard circuit is improved to achieve a minimum pulse width of 56s with current measuring capability till 20A. The entire circuit rests

on a PCB which could be easily mounted on top of Arduino, making it easy to operate. This module would be deployed for accelerated testing of solar panels, monitoring the health of by-pass diodes, in manufacturing lines and as testing measurement equipment for various purposes. Further sections gives an insight on the module and methods explaining how to operate it.

2 Hardware description

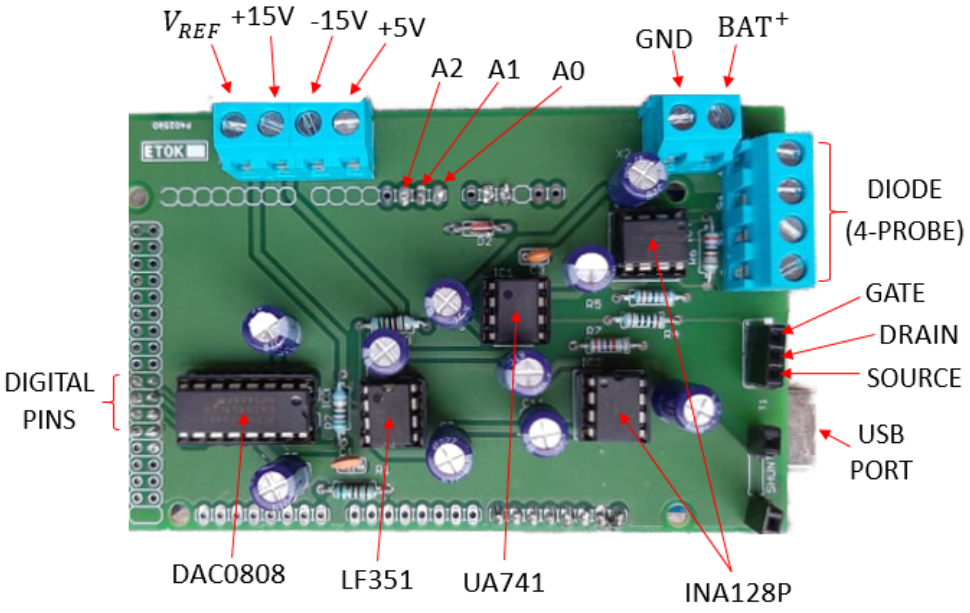


Figure 1: Connections and components

- It has four power supplies, +15V, -15V, +5V and a Reference Voltage (for DAC). All these supply pins are connected to decoupling capacitors (220μF) near every IC, for avoiding voltage transients.
- The USB port shown is of the Arduino, which could then be used to communicate with a computer.
- A0, A1 and A2 are the analog read pins of Arduino. A0 reads Voltage(amplified) across diode, A1 reads Current(amplified) through diode and A2 reads Current(non-amplified) through diode.
- There are 8 digital pins for DAC, connected to pins 30-37 in Arduino Mega 2560 board.

- The shunt resistor should be connected with two separate berg pins, near the USB port as shown in the figure.
- For more details of the amplifier/voltmeter, refer to datasheet of INA128P

3 Software description

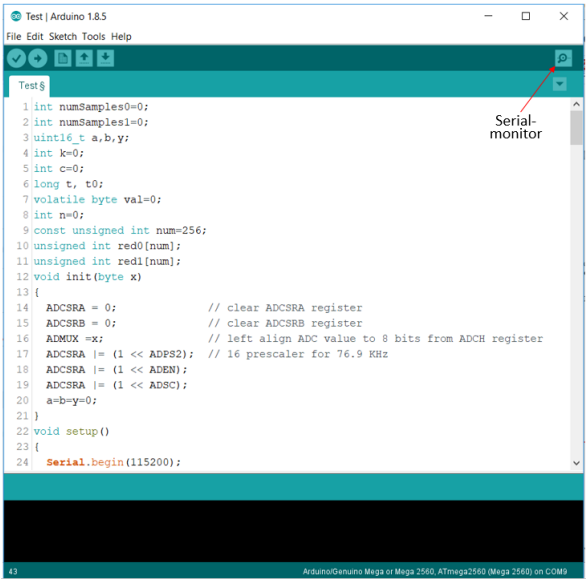


Figure 2: Code snippet

- The baud rate is kept at 115200
- The program contains an Init() which initializes the ADC ports of Arduino and starts the digital conversion
- For initialization of port A0, use **0x60** as argument for Init(), **0x61** for A1 and so on.
- The duty cycle of pulses (both width of pulses and delay between adjacent pulses) could be adjusted.
- A maximum of 256 samples could be taken as we are using 8-bit DAC, and hence 256 different pulses.

4 How to operate?

1. Make sure that the battery is charged and has enough capacity to supply higher currents

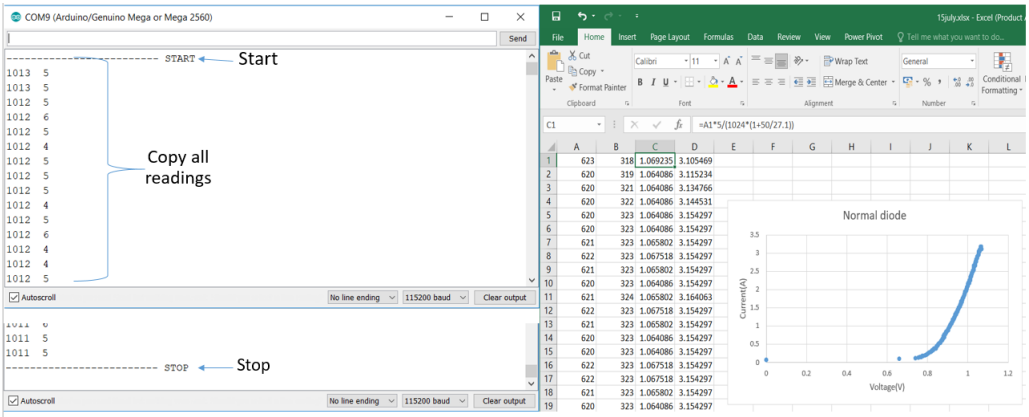
2. Insert an appropriate shunt resistor in the berg pin slots near USB port. The shunt resistor should be chosen according to the maximum required current

3. Currently, this module uses **PSMN7R0-100PS** Mosfet; the gate, drain and source could be located from datasheet

4. Make connections to all power supplies as shown in the connection diagram

5. Once all connections are made, go to the code and switch on the power supply. **As soon as** you switch on the supplies, go to the **Serial monitor** for viewing the data

6. All the readings the displayed after the word **"START"** and before the word **"STOP"**. Disconnect the USB port as soon as measurement is completed (to avoid beginning of new measurement)



7. Copy all these readings and paste them in a workbook in excel. Do suitable manipulation and adjust the amplification factor in the readings to obtain the correct analog values (*Refer to next section).

8. The corresponding I-V curve is plotted and further analysis is done. Steps 5-7 are repeated for new measurements.

Some cautions:

1. Keep a check on the temperature of various components on-board, any ambiguity refers to some damage.

2. Make sure current should not pass for long duration of time, switch off the supply if not needed.

3. Obvious to say, do not short-circuit anything, this could also damage the Arduino board beside it.

5 Data manipulation

For voltage (amplified) and current (amplified/non-amplified) readings, arduino returns readings in range 0-1023. These readings have to be appropriately converted in Volts(V) and Current(A) respectively. The following conversion gives the desired readings:

$$Voltage(V) = Arduino\_reading * \frac{5}{1024 * 2.9468}$$

,where 2.9468 is the gain-factor of amplifier.

Also for current readings(Ampere):

$$Current(A) = Arduino\_reading * \frac{5}{1024 * R_{shunt} * gain\_factor}$$

If you are reading current without amplification(Port A2), gain_factor=1; however for amplified reading(Port A1), the gain_factor is same as before(2.9468).

By default, the system uses port A0 and A2 for voltage and current measurements respectively.