

## User Manual – Trigger circuit

**1) Goals:** to quantify the amount of energy consumed by a computing device while processing a specific part of a program.

**2) General Operation:** The measurement is done when the power meter measure the voltage drops in a known shunt resistor inserted between the source and the board and the CMeasure software do the power calculations and, from the power, calculates the energy. To control the measurement interval, the power meter works using two channels: one to measure the voltage drops as already said; end the other is a trigger channel, that's control the beginning and the end of the measurements. CMeasure, in the configuration usable in this project, works in the following way: when there is a pulse signal in the trigger channel, will begin another measurement interval. Thus, when a output port of the microcontroller device (in this project, the Jetson board TK1) is linked to the power meter trigger channel, a pulse can be generate to control the measurement.

Note: a pulse, in this case, means the change of logic state. The circuit begin a new measurement when the logic state changes from '0' to '1'. Thus, it said pulse, because this is a signal that's in low state, changes to high and quickly go back to low.

**3) Send a signal in a GPIO Jetson TK1 port:** According to <http://elinux.org/Jetson/GPIO> there are lots of ports that can be used as GPIO. In this project, we have used the gpio165 port, which corresponds to the pin 55 into the J3A2 group (see Figure 1).

- To send the signal, first the GPIO port must be enabled. Then this port must be defined as output and, finally, its logic level must be defined. In Linux, these procedures are performed by the following commands:

```
echo 165 > /sys/class/gpio/export
echo out > /sys/class/gpio/gpio165/direction
echo 1 > /sys/class/gpio/gpio165/value
```

The first line enables the gpio165 port to be used; the second line defines the same port as output; and the third line defines the port logic level as high.

Note: the library used in this project was implemented in C++. Thus the commands above must be written as C++ functions. Such functions are invoked from within the target program, i.e., the program that must be analyzed. To know more about this C++ API, see <https://android.googlesource.com/kernel/tegra/+android-tegra3-grouper-3.1-kitkat-mr1/Documentation/gpio.txt>. An example of code is seen below:

Function that activates GPIO	Code that we want to analyze
<pre>int gpioSetValue (unsigned int gpio, unsigned int value) {   int fileDescriptor;   char commandBuffer[64];   snprintf(commandBuffer, sizeof(commandBuffer),     "/sys/class/gpio/gpio%d/value", gpio);   fileDescriptor = open(commandBuffer, O1);   if (fileDescriptor &lt; 0) {</pre>	<pre>int main(int argc, char *argv[]) {   // Do something without measuring energy   // We are not measuring energy consumption   // in foo().   foo();</pre>

<pre> char errorBuffer[128] ; sprintf(errorBuffer,sizeof(errorBuffer),         "gpioSetValue unable to open gpio%d",gpio) ; perror(errorBuffer); return fileDescriptor; } if (value &amp;&amp; write(fileDescriptor, "1", 2) != 2) {     perror("gpioSetValue");     return fileDescriptor; } else if (write(fileDescriptor, "0", 2) != 2) {     perror("gpioSetValue");     return fileDescriptor; } close(fileDescriptor); return 0; } </pre>	<pre> // Start measuring energy: gpioSetValue(165, 1); // Do something measuring energy, e.g.: // Measure the energy consumed by bar(); bar();  // Stop measuring energy. gpioSetValue(165, 0);  // Function baz() runs without energy // measurement baz(); return 0; } </pre>
---	---

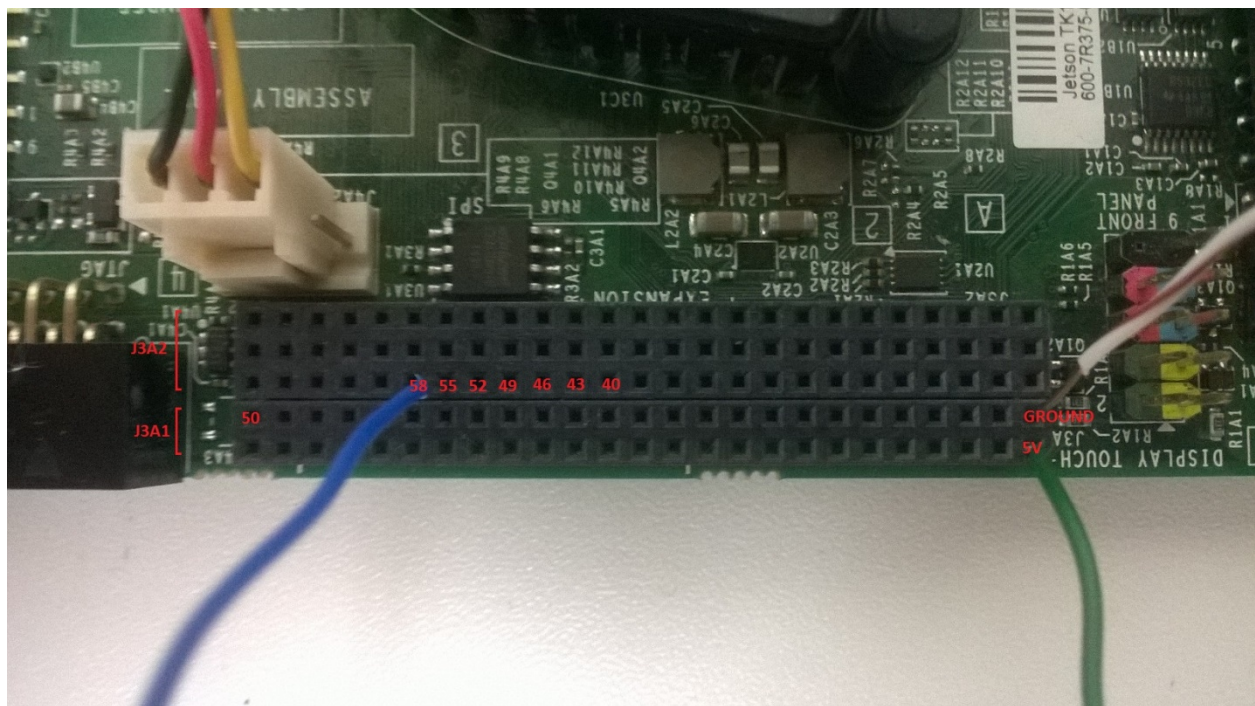


Figure 1: The Jetson TK1 ports

**4) System Assembly:** The schematic of the physical circuit is shown in the Figure 2. This is a very simple circuit, where there are only one electronic component: a 0.1Ω 5W resistor. Analyzing this figure, we can notice that: SUPPLY\_0V must be linked to BOARD\_0V; SUPPLY\_12V and POWER\_METER\_PIN\_2 must be linked to one edge of the resistor; and in the other edge, must be linked BOARD\_12V and POWER\_METER\_PIN\_3. Besides, to control the pulse signal, the power meter channel 2 must be linked to the board: POWER\_METER\_PIN\_5 must be linked to GPIO\_SIGNAL\_1.8V (J3A2, pin 55); and POWER\_METER\_PIN\_6 must be linked to BOARD\_GROUND (J3A1, pin 2).

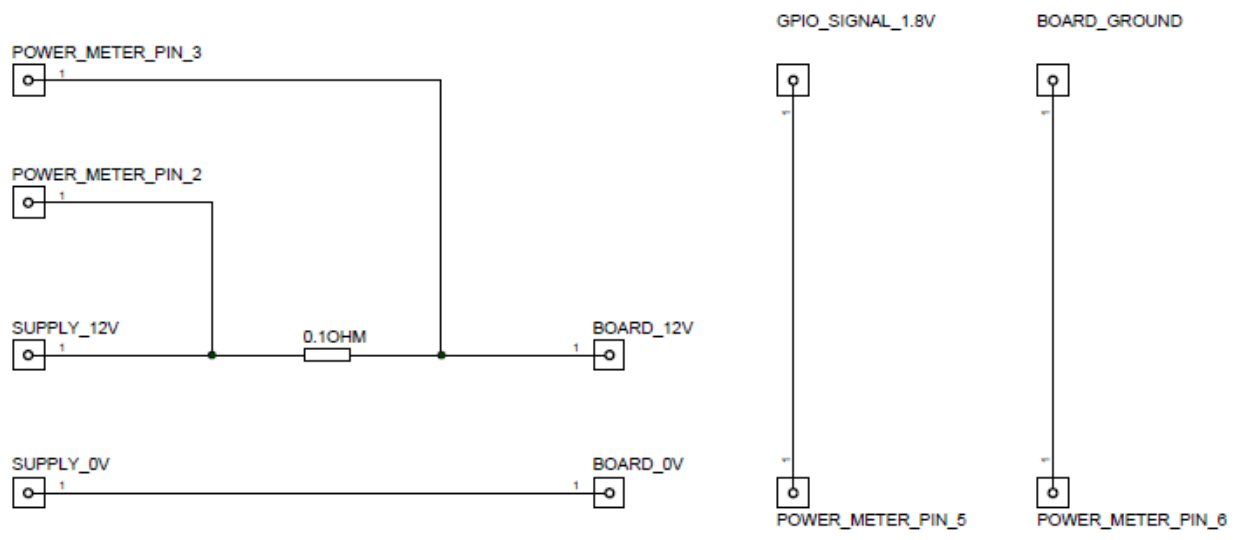


Figure 2: High level view of the circuit used to measure energy