ADCTimer Demo – Energy Meter

# Overview

This demonstration uses the ADC and timer peridocially-triggered one-shot conversion of a multiple channels (ANI0-ANI3) to create an energy meter. Power for two loads is calculated based on voltage drop across the load and the voltage drop across a series resistor. Energy is calculated by multiplying power by sample period.

In this case the loads are a green LED and a red LED. The current sense resistors are 15 ohms, leading to a current measurement resolution of 3.3V/(1024\*15) = 0.215 mA. Note that the 22 and 150 ohm resistors are for current-limiting to protect the LEDs.

# Connections

|  |  |  |  |
| --- | --- | --- | --- |
| **Description** | **MCU Pin** | **Signal Name** | **Direction** |
| Load 1 Voltage | 56 | AN0 | Input |
| Load 1 Current Sense Resistor | 55 | AN1 | Input |
| Load 2 Voltage | 54 | AN2 | Input |
| Load 2 Current Sense Resistor | 53 | AN3 | Input |
| Timer ISR Active | 38 | P55 | Output |
| ADC ISR Active | 19 | P62 | Output |
| Main Thread Active | 35 | P52 | Output |
| 3.3 V Power | 15 or 16 | 3V3 | Common |
| Ground | 13 or 14 | GND | Common |

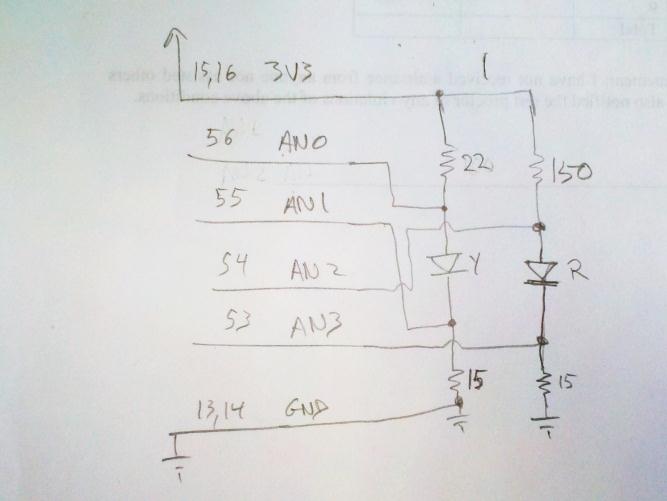


Figure 1. Schematic of energy & power meter circuit.

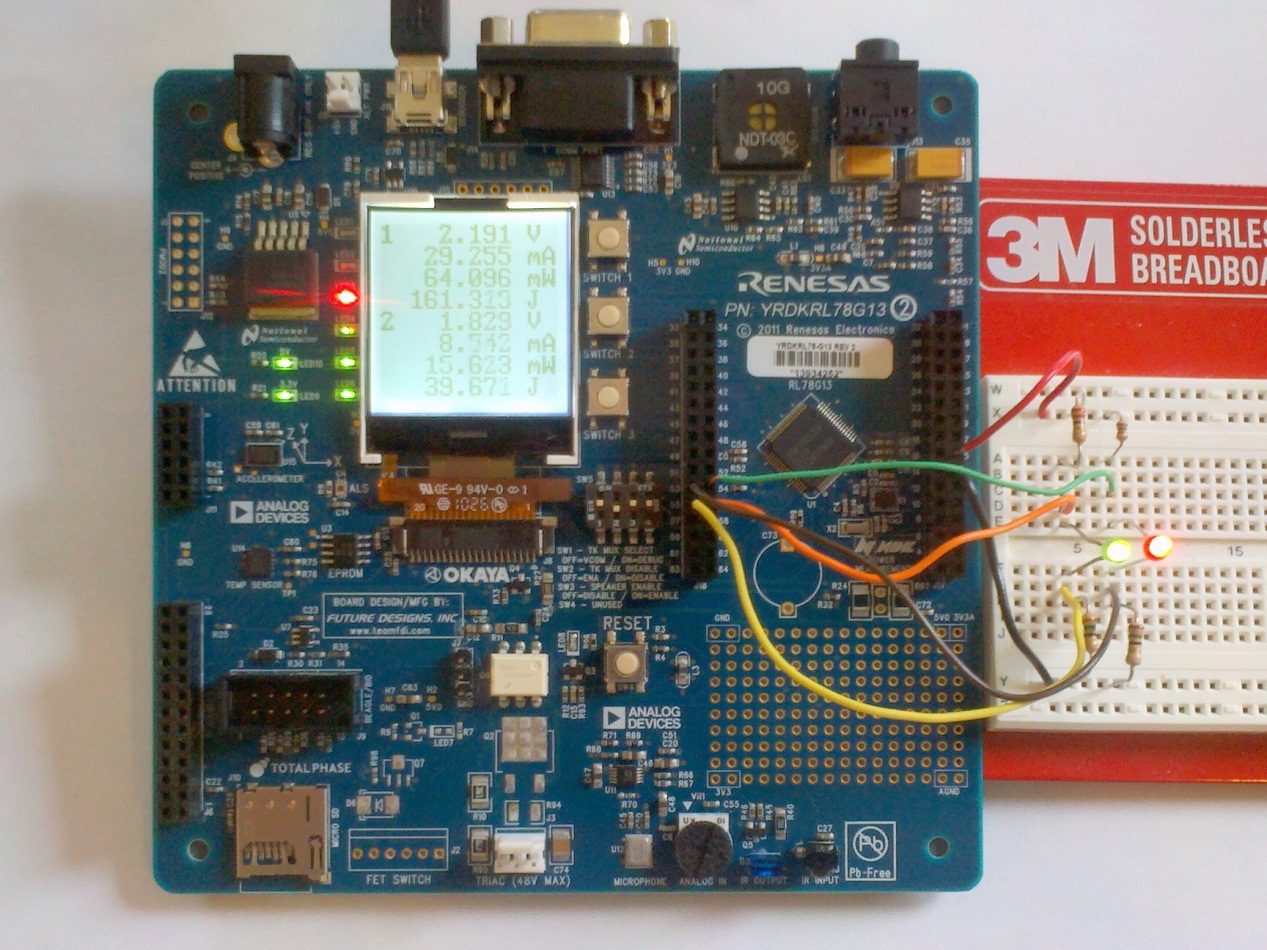


Figure 2. Energy & power meter circuit implemented on a solderless breadboard.

Two digital outputs (connected to LEDs on the RDK) are used to indicate program activity. These can be monitored with an oscilloscope or logic analyzer to see timing relationships between the signals and hence program components.

* LED1 (red) indicates when the timer ISR executes.
* LED2 (red) indicates when the ADC ISR executes.
* LED4 (green) indicates when the main loop is executing useful code, rather than waiting for the results of the ADC conversions.

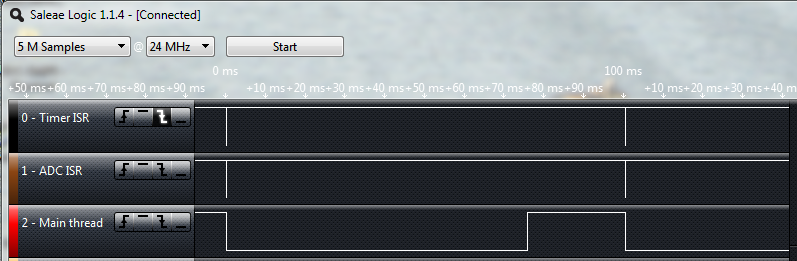


Figure 3. Logic analyzer screenshot of program activities.

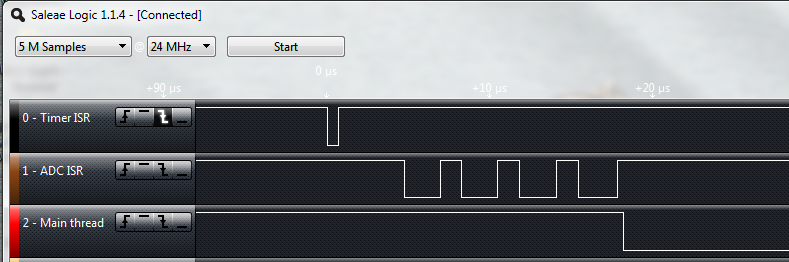


Figure 4. Detail of program activities showing Timer ISR, then four ADC ISRs followed by main code activity.

# Comments and Suggestions

* Try measuring the voltage with a multimeter and evaluate the accuracy of the ADC and software. Don’t forget that the current-sensing resistors induce a voltage drop which is subtracted out in the power calculations.
* The maximum safe input voltage is 3.3 V (the analog supply voltage for the MCU). If you need to measure a higher voltage, use a resistive voltage divider to lower it below 3.3 V and adjust the code accordingly.
* Evaluate the power used by different colors of LEDs. Blue and white LEDs will likely need more than 3.3 V to turn on, so you might want to drive them from 5V and divide down the voltages supplied to AN0 and AN2 using a 2:1 resistive divider.
* Use an oscilloscope to monitor program activity via LEDs 1, 2 and 4 (MCU pins 38, 19 and 35). How long does each ISR take to execute? How long does the main loop body take? How much faster than 10 Hz can we run the timer?