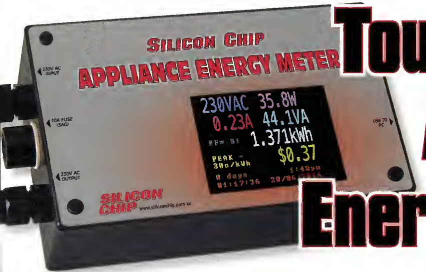

Touchscreen Appliance Energy Meter

Part 3 – Calibrating and Using it!

By JIM ROWE & NICHOLAS VINEN

In the last two months, we've described how our new Touchscreen Appliance Energy Meter works and how to put it together. Having finished assembling the unit, all that's left to do is to calibrate it and start using it.

You will need to perform several calibration steps. These allow the unit to compensate for variations in the transformer and divider resistors used to monitor the mains voltage and the isolated current sensor used to measure the instantaneous current drawn by the load.

In a little more detail, as shown in Fig.2 on pages 30-31 of the August issue, the AC-coupled output of the transformer used to monitor the mains voltage is DC biased to around 2.5V by two 56kΩ resistors across the 5V supply rail.

However, the 5V rail from the AC/DC converter may not be exactly 5V and the resistors may not be exactly the same value, so we can't assume that the DC bias level is exactly 2.5V.

During the calibration procedure, the unit measures the average DC level of this signal and stores it so that it can be subtracted from future readings, to give a pure AC signal.

Note that while the mains waveform could have a slight DC offset due to asymmetrical current flow and improperly balanced phases, as we're measuring via a transformer, we have to ignore it.

Mains current calibration

The output of the ACS712 isolated current sensor (IC4) has its own separate half-supply DC bias, obtained from a voltage divider inside the chip. So, calibration is performed with no load to allow the unit to measure the zero-current voltage level. This too is stored and subtracted from subsequent readings.

This bias exists because current can flow in either direction through the sensor and thus its output can swing above or below the zero level, to indicate both the magnitude and polarity of the current.

This is important since we need to be able to distinguish in-phase current, which indicates power flowing from the mains into the load, and out-of-phase current, which indicates power flowing from the load back into the mains.

To calculate the true power drawn by the load, we subtract one from the other. Note that for purely reactive loads, such as capacitors connected across Active and Neutral, the result of this subtraction is zero, indicating that the power is purely reactive.

While measuring the current sensor's zero level voltage, the unit also determines its RMS noise output, so that it can subtract this from future readings. Otherwise, it would look like current was flowing even with no load.

Calibration procedure

First, power the

Firmware update required

We have discovered a few bugs in the original version of the firmware (v1.0) supplied. The most serious causes it to run out of memory if you try to change the time or date. Other bugs fixed include a factor-of-ten error in the cost computations, incorrect mains frequency readout and lost logging data while updating graphs.

As a result of these bug fixes, we recommend upgrading to v1.01 immediately. You can easily do this via the unit's USB port.

Essentially, all you need to do is download the new BASIC source code (available in a zip on our website) and load it into the PIC32 over the USB serial interface. This will wipe the unit's settings so it should ideally be done before doing any calibration or setting up.

The procedure was explained in the panel on page 91 of the September issue, although you can skip uploading the Library BASIC file into the chip if it has already been programmed. The library file hasn't changed.

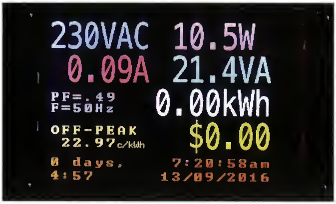

Fig.7: the main screen which has been improved slightly since the prototype was revealed in the August issue. The main differences are the addition of the frequency read-out below the power factor and support for fractional cents in the tariff, plus seconds display for the current time.

Fig.8: the logging status screen has also been improved since the first article. The same information is shown but there are now buttons to access the diagnostics screen and to perform automatic calibration. The button to dump logged data is not visible because you need to pause logging first.

unit up and wait at least 30 seconds for everything to settle (coupling capacitors to charge, etc).

You can judge this using the elapsed time in the lower-left corner of the device's display. Then touch this elapsed time display at the lower-left corner of the screen and you should see a "Calib" button appear at the bottom (centre) of the screen (see Fig.8).

Press this and the calibration screen will be displayed for a few seconds. It will then return to the main screen and after a second or two, the amps reading should drop to zero (power should be zero, too). This indicates that the unit has correctly calibrated the DC offset and base noise level from the current sensor.

Next, you need to manually adjust the voltage scale to give a correct mains voltage reading. All you need to do this is a mains-rated DMM.

Set it to AC volts mode and if it isn't auto-ranging, set it to a suitable range for measuring mains (eg, up to 260VAC). After ensuring that you have suitably rated leads, push its prongs into the Active and Neutral sockets of a mains outlet (GPO). Make sure that there's no exposed metal that you could touch and also check that the probes won't fall out.

Now touch the lower-left corner of the screen again (the elapsed time display) and this time press the "Diag" button. You should get voltage and current readings at the top of the screen, with + and – buttons to the right of each (see Fig.9).

Use these buttons to adjust the displayed voltage reading so that it matches the voltage on the DMM as closely as possible. You can now unplug the DMM from the GPO.

Current scale calibration

Now connect a device which will draw a small, fixed and easily determined amount of real power; for example, a small incandescent or halogen lamp. In a pinch, you could also use a desk fan or fluorescent lamp but make sure it has a power consumption figure printed on it so you know what to expect. If you already have an accurate mains power meter, that's even better – use it to measure the power so that you have a calibration target for the new unit.

Now connect your test load to the Energy Meter and switch it on, then let it stabilise (it may need to warm up etc) and check the power reading. It will probably be close to the rated power, but maybe a little off. As you did when adjusting the mains voltage, use the + and - buttons next to the current reading in the diagnostic screen to make small adjustments to the current reading, then go back to the main screen and check the power reading. Continue adjusting until the power reading is very close to what you would expect.

If you'd like, you can now disconnect your test load and connect another small load, and verify that you get a reasonably accurate reading. Note that loads which draw very little power (eg, under 5W) could have a quite substantial measurement error and some loads such as plugpacks may even read zero when they are in fact drawing a watt or two. This is down to the limited resolution of the ADC and current sensor and there isn't a lot we can do about it.

You may also get some slightly inaccurate readings from loads with very low power factors. But generally, the unit should be quite accurate, within 1% or so of the actual reading, plus or minus a couple of watts.

Setting up tariffs

That's all you need to do to measure power consumption but if you want to see how much an appliance is costing you to run, you will also need to program in your tariff(s) and if your home has a smart meter, the peak, off-peak and shoulder times. You will also need to set the current time and date. These all contribute to the unit being able to calculate the cost of power at any given time.

First, set the time and date by touching on the time/date display in the lower-right corner of the main screen. Type in the time, in 24-hour notation, with colons separating the hours, minutes and seconds. The seconds value is optional and the time will be set as soon as you press "OK", so once you have entered the time value, you can wait until your clock rolls over to the next minute and then press that button. The value entered will be red if it is invalid or incomplete, or black if it is valid and complete.

Having set the time, enter the date in the same manner, in DD/MM/YY format. You can just press OK if you just want to update the time and keep the current date.

Now that the time and date are set, press on the yellow tariff data to the left of the screen (initially, it will read

Fig.9: the diagnostic screen which shows the voltage and current readings with extra decimal places and allows fine adjustment of the scaling factors for both. It also displays the automatically calibrated calibration constants below, plus the sampling rate, measured frequency and pre-processing VA figures.

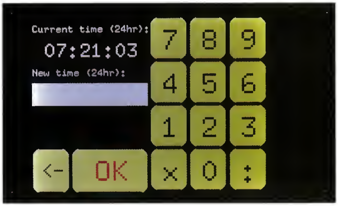

Fig.10: this keypad allows you to update the current time and date as well as set the tariffs and various other tariff-related settings. In this case we're setting the time and pressing OK without entering anything leaves it unchanged. The new time can be entered with or without seconds.

"OFF-PEAK 0.00c/kWh"). Now press on the "Off-peak" text towards the bottom of the screen, type in the cost of power, in cents per kilowatt-hour. You can use up to three decimal places. Press OK when finished, then press in the very upper-left corner to go back to the main screen.

If you don't have a smart meter, that is all you have to do because this tariff value is the default for situations where a conventional watt-hour meter is fitted. (Don't worry if you have an off-peak hotwater system as it is on a separate circuit in your house wiring).

Setting up time-of-day metering

Assuming you have a smart meter, you now need to set the peak and shoulder tariffs, using the same method. Then you will need to set the start and end times for the peak period during the week (ie, Monday through Friday). Refer to your electricity bill or electricity authority website if you don't have this information.

To set the peak times during the week, press on the text which says "Weekday: N/A", just under where the peak tariff is displayed, near the top of the screen. Then, enter the peak start time in 24-hour format, with the hours and minutes separated by a colon and press OK. You will immediately be prompted to enter the end time, in the same format.

The unit has support for two peak periods, however presently no Australian supplier has a separate morning and afternoon peak time. So you can simply press OK to go through the two following screens without entering any additional time values.

The peak time period should now be displayed below the peak tariff. If your supplier also has peak periods during the weekend, you can enter the start and end time by pressing on the line below which says "Weekend: N/A" and using the same procedure as above. Otherwise, move on to setting up the shoulder period.

Most suppliers which have a peak period also have a "shoulder" period before and after the peak period, where the cost of electricity is higher than it is off-peak but lower than during peak times. Assuming yours does too, you will need to set its start and end times just as you did for the peak period, but instead by pressing on the weekday and weekend lines below the Shoulder tariff.

Note that it's OK for the peak and shoulder periods to overlap; indeed, they should. The peak tariff will override the shoulder tariff during those times when they are both active.

That's it, you can now go back to the main screen. The tariff data is automatically stored in non-volatile flash memory and will survive a power outage (or simply unplugging and moving the unit).

Public holidays

While probably not critical, for the cost display to be truly accurate, we also need to take into account the fact that public holidays are charged the same as weekends. For the unit to take account of this, it must know the dates of public holidays and so you can program them in. If you don't, it won't normally make a big difference to cost calculations, so it's entirely up to you. But it only takes a few minutes.

To do this, acquire a list of the public holidays in your state for the next couple of years, then touch on the area at the bottom of the tariff settings screen. You can then press on each blank public holiday space and enter the date in dd/mm/yy format. Enter as many or as few as required. Whenever the date matches one of these days, weekend rates will be applied. Touch right at the top of the screen to go back to the main tariff settings display.

Accumulating & logging data

Logging and accumulation of energy usage and cost begin automatically when the unit is plugged in. However, you can pause or stop and reset this data at any time. To do this, press on the time elapsed in the lower-left corner of the screen. The logging screen displays the current logging status, such as how much memory has been used and the maximum time that logging can continue with the current interval, as well as some buttons to control it (see Fig.8).

Pressing the "pause" button will stop logging but retain all data so far. You can then resume or press the "stop" button to clear the cumulative energy usage, cost and voltage/current/power logs.

Note that you can log data for up to two hours and 40 minutes with a one-second interval, up to 24 hours with a ten-second interval and up to one week with a one-minute interval but you can only change the interval when logging is stopped (ie, no data is stored). To do so, simply press on the

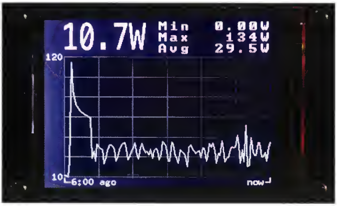

Fig.11: power usage plot for a soldering iron. The iron was switched on around five minutes ago and you can see the large power draw as it warms up initially, followed by the consumption jumping up and down as the element is switched on for brief periods to keep it warm.

Fig.12: plot of the mains voltage which shows how it varies over a one-hour period. Depending on the location and time of day, the voltage can vary far more dramatically than this. Even so, we can see it varying by more than 1% (2.3V) in a relatively short period of just 30 seconds or so.

"Interval:" line on the logging screen.

While paused, you also have the option to dump the logged data to your PC via the USB interface. This can be done with the mains still connected. In fact, if the unit loses power, this logged data will be lost, so you will need to keep the mains power plugged in, at least until you've connected the USB interface.

Once the USB serial port has been recognised by your PC, fire up a terminal program and open that port with the correct baud rate (normally 38,400). Next, set up the terminal program to capture data from that serial port to a file. You can then press the "Dump" button on the screen and the data will be output in CSV format, as follows:

```
SILICON CHIP Appliance Energy Meter log at 11:04:37 09/09/2016
num,seconds,time,v,a,va,power,pf
1,0,00:00,237,0.221,52.4,12.3,0.235
2,10,00:10,235,0.219,51.5,12.7,0.247
...
```

It may take some time to off-load all this data at 38kbaud, depending on how long you have been logging. This data can be saved in a CSV file and opened in a spreadsheet program. The columns are as follows:

1) record number, starting at one for the first row of data.

2) number of seconds since logging began. Starts with zero and increments by one, 10 or 60 depending on the logging interval.

3) time since logging began, in mm:ss or hh:mm format, depending on how long logging has been going.

4) average mains RMS voltage for the logged interval.

5) average mains RMS current for the logged interval.

6) product of #4 & #5, ie, average VA for the logged interval.

7) average real power for the logged interval.

8) average power factor for the logged interval (ie, #7 divided by #6).

When finished, press the "Back" button to return to the main screen.

Note that while logged data is lost if the unit's power is removed, the accumulated power usage and cost information, shown on the main screen, is stored in the EEPROM once per minute and the last saved data is restored at power-on. This data is only reset when logging is stopped.

Plotting data on the unit

The data stored in RAM which can be exported to a PC can also be used to produce various plots on the Meter's touchscreen. However, due to limited screen space (and program space), you can only plot one measurement at a time.

Simply touch on one of the following items on the main screen to draw a graph of the data collected so far: voltage, current, power, VA or power factor. Initially, a line graph will be drawn, showing the variation in that parameter over time. You can change the plot duration between one hour, one day and one week by touching on the duration legend below the graph. Note that if the unit has insufficient data to show the selected duration, it will simply show what it has so far.

The vertical axis of the graph is automatically scaled to fit the data collected so far. The horizontal axis has the latest measured value at right and the oldest data at far left. Note that depending on how long the unit has been running, it can take some time for it to average all the data required to plot the graph, so be patient.

The unit can also display the same data in a histogram. Simply press in the middle of the graph to switch to histogram mode. The data is automatically allocated to ten "bins" which span the range of data collected and their height indicates the proportion of values measured which fit into those "bins" (see Fig.13). Press on the middle of the graph again to go back to the main screen. (This is the only way to get out of the graph display.)

Extrapolating power consumption and cost

During logging, the total power consumption and accumulated cost on the main screen are continuously updated (once per second). They will continue to increase even if the logging RAM is full, indefinitely.

If you want to see how much an appliance is costing you on average, or its average power usage, connect it to the Meter and let it run for a sufficient period for it to experience representative power usage. In some cases (eg, a refrigerator or air conditioner), this may take one or two days.

At the end of this period, simply touch on the power consumption or cost figure on the main screen. The unit will divide the figure by the amount of time it has spent monitoring that load, then extrapolate the energy usage/cost out to the following periods: one hour,

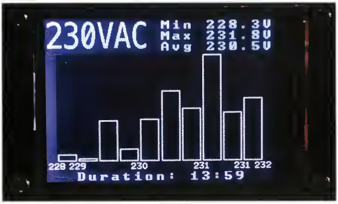

Fig.13: histogram plot of mains voltage. This gives you a good idea of which voltages the mains sits at most of the time relative to outliers. Note that the X-axis labels are rounded to the nearest volt while the data has sub-volt resolution.

Fig.14: extrapolated energy usage involved in running a temperature-controlled soldering iron, based on around eight minutes of data. You don't normally leave a soldering iron on all the time but if you did, this shows just how much power it would use.

one day, one week, one month and one year. This will tell you the energy usage/cost for running that appliance over those periods, assuming that the energy usage continues at the same rate (see Fig.14).

With something like an air conditioner, you will have to keep in mind that if you are measuring during summer or winter, the yearly usage will be overestimated (since you won't need the same amount of cooling or heating year-round). For heaters, the same is true in reverse. And refrigerator energy usage is likely to vary significantly with the season too.

Conclusion

The easiest way to become familiar with the functions of this device is probably to set it up and then "have a play".

For those constructors who may wish for features that we didn't have room for, feel free to download the BASIC source code and add your own features. However, keep in mind that you will probably need to remove some of the existing features to make room. SC

Developing the two critical CFUNCTIONs

While the GUI code is mostly written in BASIC, we had to write two sections of the program in C. The first is the part which queries the ADC and performs averaging, power calculations and zero crossing/frequency detection. This needed to be written in C both so that it was fast enough to be run thousands of times per second while still allowing enough free CPU resources to handle screen updates, and so that it could run constantly in the background to avoid missing any voltage, current or power samples.

The second is the part of the code which calculates the current tariff based on the time, date and configuration data. This was originally written in BASIC, however, it used too much RAM; this was especially problematic because the very inner-most function which reads and stores power data must call it in order to keep the running cost up to date (based on the current tariff). Re-writing this code in C caused it to use up more flash memory (due to the way CFUNCTIONs are stored) but significantly less RAM and solved a long-running problem with the unit crashing due to lack of memory. It's also a lot faster than the equivalent BASIC code.

Essentially, what this second function does is calculate the day of the week based on the date, then if it is a weekday, it checks to see if the date matches any of the public holidays programmed into the unit. Once it knows whether to use the weekday or weekend tariffs, it figures out the current tariff based on the time.

The other CFUNCTION is significantly more complex. While it's a single function, it performs multiple duties. The first one is to set up the hardware sampling timer (TIMER1) and the internal data structures used to keep track of the voltage, current, power, etc. As soon as TIMER1 is set up, the interrupt handler runs several thousand times per second and this alternately samples the voltage and current.

After each pair of samples has been completed, it then updates the internal RMS voltage, current, VA and power variables and checks to see if a zero crossing has occurred. If so, it increments the zero crossing count and transfers the accumulated data into a second area of RAM, so that all averages are performed on full multiples of half-cycles of data (to prevent readings from varying depending on which point in the half-cycle the data is read).

The BASIC software can then call the same CFUNCTION with a different set of parameters to read out these internal registers and get at the accumulated data. When this data is read, interrupts are disabled and it is cleared, so that the next ADC interrupt will start fresh, collecting the next set of data.

The number of zero crossings detected per time period are used to calculate the mains frequency along with the real time clock and the Micromite's internal millisecond timer.

Finally, this CFUNCTION also provides calibration functions, ie, the ability to read and write the registers which define the voltage and current DC offset levels as well as compute these levels when no load is connected. Once set, the calibration levels are used by the sampling code to improve the accuracy of the readings. Some calibration functions, specifically the relationship between measured voltage and actual mains voltage and current, as well as dealing with noise from the current sensor, are performed solely by the BASIC code.

Those who are curious can download both the BASIC and C source code from the SILICON CHIP website and see the full details.