Tanishka Chakraborty, 2026, 14:5
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology
An Open Access Journal

Implementation of A Microcontroller-Based Power Factor and Energy Monitoring System

Tanishka Chakraborty[1] Prof. Thamilmaran A[2], Pallavi Kumari[3]
School Of Electrical Engineering Vellore Institute Of Technology Vellore, Tamil Nadu, India
School Of Electrical Engineering Vellore Institute Of Technology Vellore, Tamil Nadu, India
[3]Assistant Professor (Senior) Department Of Electrical Engineering School Of Electrical Engineering Vellore Institute Of Technology Vellore, Tamil Nadu, India

Abstract: Household electricity meters usually report total energy, which tells a user little about how individual appliances behave. This paper describes a low-cost single-phase monitoring unit built around an ESP32 microcontroller, a ZMPT101B voltage transformer module and an ACS712 Hall-effect current sensor. The firmware samples both channels over a 400 ms window, which spans 20 cycles of a 50 Hz supply, and computes RMS voltage, RMS current, real power, apparent power, power factor, accumulated energy and running cost from the sampled data. Results appear on a 16x2 I2C LCD, and a push button selects between three screens. Idle-channel records show mid-scale offsets near 1959 counts (voltage) and 1979 counts (current), with noise standard deviations of 19.4 and 48.5 counts. The current-channel noise alone corresponds to about 0.59 A, which led to a start-up noise-zeroing routine and a 0.08 A dead band. A first live voltage reading of 116 V against a nominal 230 V supply was corrected by rescaling the voltage constant to 0.595 V per count. Accuracy against a reference meter has not yet been measured. The limits of the present design are discussed.

Index Terms: energy monitoring, power factor, ESP32, ZMPT101B, ACS712, RMS measurement, embedded systems.

I. INTRODUCTION

Energy meters installed by utilities record active energy and little else [4], [5]. A household sees that number once a month, as a bill, after the consumption that produced it. Reviews of feedback studies suggest the delay matters. Darby [2] reports savings of 5 to 15 percent where users could see consumption directly from the meter, and Fischer [3] finds that feedback works best when it is frequent and tied to individual appliances. Smart metering programmes address part of this gap, although their deployment brings technical and cost issues of its own [1].

Power factor is a quantity that ordinary billing hides from the consumer. For a given real power, a load with a low power factor draws more current, and the extra current adds to losses in the wiring that supplies it [17]. Measuring it needs access to voltage and current, as in the classical wattmeter and power factor meter [6].

The same two measurements, real power and apparent power, are also the starting point of methods that identify appliances from an aggregate load signal [18], [19]. Low-cost microcontrollers now carry analog-to-digital converters (ADCs) that are good enough to do this work in software. Open hardware projects have shown that an Arduino-class board, a voltage transformer and a current sensor can compute RMS values and power from waveforms [20]. The ESP32 [13] adds a 12-bit converter, two processor cores and Wi-Fi, which makes it a reasonable base for a meter that may later be networked.

This paper describes the design and implementation of a single-phase monitoring unit built on an ESP32 module, a voltage transformer module [12] and an ACS712 Hall-effect current sensor [11].

The unit shows RMS voltage, RMS current, real power, apparent power, power factor, accumulated energy and cost on a 16x2 liquid crystal display (LCD). We describe the circuit, the computation, the calibration steps and the behaviour seen during bench testing. Where the data are incomplete, particularly the accuracy against a reference meter, we say so.

© 2026 Tanishka Chakraborty, This is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly credited.

II. MEASUREMENT BACKGROUND

A. *Power quantities*

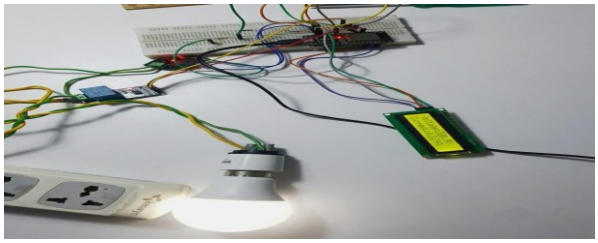
IEEE Std 1459 [16] defines active power as the average of instantaneous power over an integer number of cycles, apparent power as the product of RMS voltage and RMS current, and power factor as the ratio of the two. For sinusoidal waveforms the ratio equals the cosine of the phase angle between voltage and current. With distorted waveforms it does not, and the difference between displacement and total power factor starts to matter [16], [17]. The meter described here divides power by the product of the RMS values, so it reports the total power factor of the sampled waveforms, within the accuracy of the sampling chain.

A. *Sensors*

The ACS712 passes the load current through an internal copper conductor and senses the resulting magnetic field with a Hall element, a principle covered in [10]. The datasheet [11] lists versions rated 5 A, 20 A and 30 A, with sensitivities of 185, 100 and 66 mV/A. With no current the output sits at half the supply voltage, and it moves linearly with current from there. The package isolates the conductor from the signal pins to 2.1 kVRMS. The ZMPT101B [12] is a 2 mA : 2 mA instrument transformer. A series resistor on the module limits the primary current, and the module electronics turn the secondary current into a voltage centred near mid-supply. Both sensors therefore keep the microcontroller galvanically separate from the mains.

B. *Line current and power factor*

Power factor matters because it sets the current needed to deliver a given real power. At 230 V, a 1 kW load draws 4.35 A at unity power factor, 5.43 A at 0.8 and 8.70 A at 0.5. Resistive loss in the supply wiring grows with the square of the current, so these three cases lose 1, 1.56 and 4 times as much, with the same real power delivered [17].



A domestic bill does not show any of this. The meter in this paper reports powe factor next to real power so that the two can be read together for one appliance at a time.The reading also depends on waveform shape. Loads with an inductive winding, such as a fan motor or a magnetic ballast, shift the current behind the voltage. Loads with a rectifier and a smoothing capacitor, such as small power supplies, draw current in short pulses, and distortion lowers their power factor even when the pulses are centred on the voltage peak [16], [17]. Because the meter divides real power by the product of the RMS values, both effects are included in the single number it shows.

III. SYSTEM DESIGN

A. *Hardware*

Fig. 1 shows the arrangement. The supply cord feeds a socket through the ACS712, which sits in series with one conductor. The ZMPT101B primary is connected across the supply. The sensor outputs go to GPIO34 and GPIO35 of the ESP32, which are input-only pins on the first ADC unit; the second unit shares hardware with Wi-Fi and was avoided for that reason [13], [14]. The module used was an ESP32-WROOM-DA, as identified by the programming environment. A 16x2 I2C LCD at address 0x27 shows the results, and a push button on GPIO27 with the internal pull-up enabled selects the screen. Table I lists the connections. The ACS712 variant is the 20 A version (100 mV/A), and this value is used in the current calibration constant of Section IV. The assembled bench prototype, with a lamp as the test load, is shown in Fig. 2.

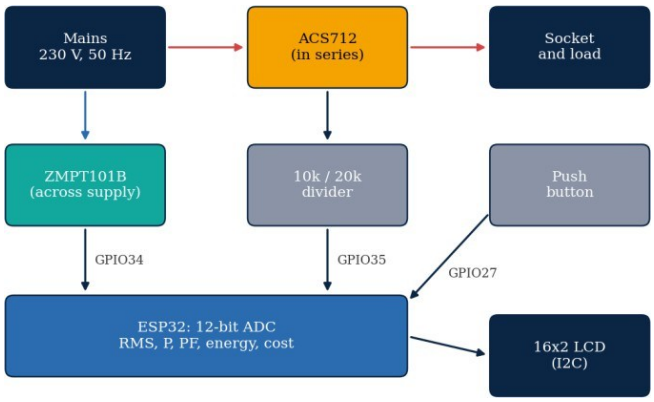


Fig. 1. Block diagram of the monitoring unit. Mains-side connections are shown in red.

Fig. 2. Bench prototype of the monitoring unit: ZMPT101B voltage module, ESP32 on a breadboard, 16x2 LCD and a lamp as the test load.

TABLE I

Components and connections	Components and connections	Components and connections
Component	Supply	Connection
ZMPT101B module	3.3 V	OUT to GPIO34
ACS712 (20 A)	5 V (VIN)	OUT to GPIO35 through a 10 kΩ / 20 kΩ divider
16x2 I2C LCD	5 V (VIN)	SDA to GPIO21, SCL to GPIO22
Push button	none	GPIO27 to GND, internal pull-up

A. *Signal conditioning*

The ACS712 is powered from 5 V, so its output can swing well above the 3.3 V that the ESP32 pins tolerate. A divider of 10 kΩ and 20 kΩ scales the output by α = 0.667 and loads the sensor with 30 kΩ. For an ideal 2.5 V zero-current output the divided level would be 1.67 V. The observed idle mid-point was 1979 counts, which is about 1.60 V for an ideal converter and a little below that figure. The 5 V USB rail is often slightly low and the ESP32 converter has its own offset [14], so the difference was not pursued; the firmware removes the mean of every window in any case. The ZMPT101B is powered from

1. V So That Its Output Stays Within The Adc Range

Its idle mid-point of 1959 counts is close to half scale.

A. Firmware

The firmware uses the Arduino-ESP32 framework. The ADC is set to 12-bit resolution with 11 dB attenuation, the setting intended for the widest input range. Each pass of the loop reads both channels continuously for 400 ms. At every sampling instant each channel is read four times and averaged (M = 4), and the pair is added to running sums of v, i, v^2, i^2 and vi held in 64-bit integers. No sample array is stored, so memory use does not depend on how many samples fit in the window. A 400 ms window holds 20 cycles of a 50 Hz supply, or 24 cycles at 60 Hz, so it always covers a whole number of cycles. At the end of the window the means, variances and covariance are computed from the sums.

The choice of 64-bit sums is a matter of headroom. With the mid-scale offset near 1959 counts, each squared sample is about

$$P = KVKI|\langle vi\rangle - \langle v\rangle\langle i\rangle| \quad (3)$$

Noise that is uncorrelated between the two channels contributes little to $\langle vi\rangle - \langle v\rangle \langle i\rangle$, so no correction of the type used in (2) is applied to power. The absolute value is taken because a two-pin plug does not tell which wire is the line, and the sign of the covariance has no reliable meaning. Because of that, the meter cannot show power flow or tell a leading load from a lagging one. Apparent power and power factor follow as

$$S = Vrms\ Irms \quad (4)$$

2. × 106, So A 32-Bit Unsigned Sum Would Overflow After Roughly 1100 Samples

A 64-bit sum holds values up to about

$$PF = \frac{P}{S} \quad (5)$$

× 10^{19}, which no window the ADC can fill in 400 ms comes close to. Keeping the sums as integers also means that the subtractions in (1) and (2) work on exact values until the final conversion to volts and amperes.

Three routines deal with problems. At start-up, with the load with power factor limited to a maximum of 1. Energy and cost are accumulated from the window power Pj and the time Δtj of each pass, in hours:

$$EWh = Pj\ \Delta tj \quad (6)$$

unplugged, the firmware records eight windows and stores the mean current-channel variance as the noise variance σn2.

$$C = \frac{EWh}{1000}$$

× T (7)

Holding the button for two seconds repeats this step. Voltages below 20 V and currents below 0.08 A are displayed as zero, and the displayed current passes through a first-order smoother,

where T is the tariff in rupees per kWh. The current constant comes from the sensor sensitivity S_s, the divider ratio α and the ADC step $q = V_{ref}/4095$:

$I_s \leftarrow 0.6 I_s + 0.4 I$. A short press cycles through three screens: voltage and current; power and power factor; energy and cost.

$$K_I = q / (\alpha S_s)$$

(8)

The button is polled inside the sampling loop with a 50 ms debounce, so a press is not lost during a measurement window. Energy is accumulated in watt-hours using the elapsed time of each pass, and cost uses a tariff of Rs 8 per kWh that can be edited in the code. Table II collects the parameters.

For the 20 A sensor, q = 0.806 mV, α = 0.667 and S_s = 0.1 V/A, which gives K_I = 0.01208 A per count. The voltage constant depends on the module gain and its trimmer setting, so it is set by testing:

TABLE II Firmware parameters

Parameter	Value
Sampling window	400 ms
Oversampling per channel	$M = 4$
ADC setting	12 bit, 11 dB attenuation
Noise-zeroing windows	8
Voltage cutoff	20 V
Current dead band	0.08 A
Current smoothing	0.6 old + 0.4 new
Long-press time	2 s
Tariff	Rs 8 per kWh

$$K_{V,new} = K_{V,old} \times V_{nom} / V_{disp}$$

(9) Two error factors are worth mentioning. The channels are read
one after the other, so the current sample is behind the voltage sample by one conversion time Δt_s, which looks like a phase error

$$\Delta\varphi = 360° \times f \times \Delta t_s \quad (10)$$

The conversion time was not measured in this work; for example, a delay of 50 µs is 0.9 degrees at 50 Hz. Its effect on power factor depends on sin φ, so it is small for a resistive load and bigger for strongly reactive ones. The second factor is noise. Averaging M readings reduces uncorrelated noise as [15]

IV. COMPUTATION

Let v_k and i_k be the oversampled counts at the k-th sampling instant of a window, with N samples. Let $\langle x \rangle = (1/N) \Sigma x_k$ denote the window average. The RMS voltage is

$$V_{rms} = K_V \sqrt{\langle v^2 \rangle - \langle v \rangle^2} \quad (1)$$

Subtracting $\langle v \rangle$ removes the DC bias of the sensor module. The offset does not have to be known. K_V converts counts to volts. For current, the corresponding quantity is

$$I_{rms} = K_I \sqrt{\sigma_I^2 - \sigma_n^2} \quad (2)$$

where $\sigma^2 = \langle I^2 \rangle - \langle i \rangle^2$ is the variance of the current channel. Noise that has no relation to the load current adds to the variance, so taking away the stored noise variance σ_n^2 removes it from the estimate. If the difference is negative, the current is taken as zero. Real power uses the covariance form:

$$\sigma_{avg} = \frac{\sigma}{\sqrt{N}}$$

so M = 4 halves such noise, while correlated disturbances like supply ripple are not reduced in the same way.

V. OBSERVATIONS AND CALIBRATION

A. *Idle-channel behaviour*

The first firmware version used *K*V = 0.30 and *K*I = 0.01208, with no oversampling and no noise subtraction. With the supply off and no load connected, it printed the mean and standard deviation of each channel to the Serial Monitor at 115200 baud. Table III lists seven windows.

TABLE III

Idle readings, first firmware revision (counts)

Window	V mean	V σ	I mean	I σ
1	1959	19.0	1979	48.1
2	1958	19.5	1979	48.6
3	1959	19.7	1978	48.1
4	1959	19.3	1978	48.7
5	1959	19.5	1979	48.3
6	1959	19.3	1979	49.1
7	1959	19.6	1980	48.5
Mean	1958.9	19.41	1978.9	48.49
Sample s	0.38	0.23	0.69	0.36

The mid-points are steady to within two counts. The standard deviation of the AC component, which should be zero with the supply off, averaged 19.4 counts on the voltage channel and counts on the current channel. With *K*I = 0.01208 A per count, 48.5 counts equals 0.586 A, which matches the 0.58 to

A shown on the display with nothing connected. On the voltage channel, 19.4 counts at *K*V = 0.30 equals 5.8 V, and the 20 V cutoff hid it. The display read 0.0 V. At the later value of *K*V, the same noise would read 11.5 V, still under the cutoff.

The current channel is noisier in counts. Each count is worth mA, so the equivalent noise at the load is large for a 20 A sensor. Possible reasons are the sensor output noise, the 6.67 kΩ source impedance that the divider presents to the ADC, and ripple on the USB supply [7], [9]. These were not tested separately. The records come from the first firmware version and therefore show the noise floor before any improvement. Oversampling, start-up noise subtraction, the dead band and the smoothing filter were added in response. Their effect on the no-load display was not kept, so no improvement is claimed here.

A. *Voltage calibration*

With the mains connected and *K*V = 0.30, the display read 116 V. Taking the 230 V supply as the reference gives an error of (116 − 230)/230 = −49.6 percent. Equation (9) then gives *K*V

= 0.30 × 230/116 = 0.595, the value in the firmware (Table IV). Because the standard voltage was the reference, the constant should be refined using a multimeter reading, since real mains voltage is not always at its standard value.

Constant	Value	Basis
KV, initial	0.30 V/count	starting value
KV, current	0.595 V/count	Eq. (9), standard 230 V
KI	0.01208 A/count	Eq. (8), 20 A sensor

TABLE IV Calibration constants

A. *Expected values for a lamp load*

For a 100 W lamp on 230 V, the expected current is 100/230

= 0.43 A and the power factor is close to 1. Ten hours of use should add 1 kWh, which costs Rs 8 at the tariff in the code. These are expected values for checking and are not actual measurements.

The energy accumulator can be checked in the same way. At 100 W, one 400 ms window adds 100 × 0.4/3600 = 0.0111 Wh, so 1 kWh takes 90000 windows, which is 10 hours of continuous use. The power of each window is applied over the whole pass, including the time spent on computation and the LCD update, and that interval was not measured. For a steady load the effect should be small. For a load that switches often it

A.

Signal size and expected effect of noise subtraction

The constants in Table IV allow the size of each signal in ADC counts to be estimated, and the idle noise of Table III can then be set against it. At 230 V and *K*V = 0.595 V per count, the RMS voltage is 387 counts and the peak is 547 counts, which fits easily within the converter range around the 1959-count mid-point. Against the 19.4-count voltage noise this is a ratio of

about 20, or 26 dB. The current channel is in a weaker position. Table V lists the current signal for several loads at 230 V and unity power factor, using *K*I = 0.01208 A per count and the 48.5-count noise of Table III. TABLE V Estimated current signal and noise effect (calculated)

Load	I (A)	Signal (counts RMS)	Error without subtraction (%)	Error with subtraction (%)
40 W	0.17	14.4	251	8.4
100 W	0.43	36.0	68	1.3
500 W	2.17	180	4	0.05
1000 W	4.35	360	1	0.01
2000 W	8.70	720	0	0.00

The last two columns need explanation. The first is the overestimate that Eq. (2) would give if σ^2 were not subtracted, which is the square root of (S^2 + σn2) divided by *S*, minus 1, where *S* is the signal in counts RMS. For a 100 W lamp this is 68 percent, and for a 40 W lamp it is 251 percent. The second is what remains when the noise variance is subtracted but the window-to-window scatter of the noise is not. Table III gives a sample standard deviation of 0.36 counts for σI, which is about 35 counts^2 in variance; dividing by twice the signal variance gives the entry shown. These figures use the first firmware revision, so they indicate scale only. They are not a prediction for the present firmware, where oversampling should change the noise.Table V supports the point made in Section VI. Below about 100 W the noise correction carries most of the result, and the 40 W case leaves an uncertainty near 8 percent even when the subtraction works as intended. From 500 W upward the noise has little effect on current, and the limit moves to the other items listed in Section VI.

A. *Effect of channel skew on power factor and energy*

The phase error of Eq. (10) can be turned into an effect on the displayed values. For a lagging load with true phase angle φ, a skew Δφ changes the measured power factor from cos φ to cos(φ − Δφ), because the later current sample makes the load look slightly less lagging. Real power and energy change by the same ratio. Table VI evaluates this for three assumed delays at 50 Hz. The delays are examples, since the conversion time was not measured here.

Delay (µs)	**Δφ (deg)**	**PF 0.8 reads**	**PF 0.5 reads Energy error at PF 0.5 (%)**
25	0.45	0.805	0.507 1.4

TABLE VI Effect of assumed channel delay on power factor (calculated)

adds an error that has not been quantified.	**50**	**0.90**	**0.809**	**0.514**	**2.7**
	100	1.80	0.819	0.527	5.4

Near unity power factor the error is negligible, and at PF 0.8 even a 100 µs delay shifts the reading by only 0.019. The skew becomes noticeable for strongly lagging loads, where a 50 µs delay already raises the energy total by 2.7 percent at PF 0.5. Measuring the actual conversion time, or shifting one sample stream by the known delay, would remove most of this.

I. Discussion and Limitations

Static meters of class 1 and class 2 are required by [4] and [5] to keep their errors within limits of 1 and 2 percent under standard conditions. The present unit has not been tested against those requirements, and the data above give reasons to expect issues at low current. A 40 W lamp draws 0.17 A, which is only 14 counts of RMS signal against a current-channel noise of 48 counts. The current estimate then depends on the difference between two large variances, and the uncertainty in the stored noise variance (a sample standard deviation of 0.36 counts in Table III) is not negligible for it. Loads of several hundred watts are a better case.

The converter nonlinearity reported by the manufacturer [14] is not corrected, because raw counts from the Arduino analog read function are used. The sequential sampling of the two channels leaves the

phase skew described in Section IV. The energy total is held in RAM, so it is lost when power is removed. The unit measures one phase, and as built it reports no sign for power. On the practical side, the low-voltage part was assembled on a breadboard, and an enclosure and mains-side protection such as a fuse are needed before the unit is used for more than bench testing.

I. Future Work

The first task is a comparison against a reference energy meter and a multimeter over inductive loads, reported as error against load current. A bypass capacitor at each ADC pin and a lower-impedance divider are hardware steps that may reduce the current-channel noise. Per-chip ADC calibration [14], storage of the energy total in non-volatile memory, and a zero-crossing phase measurement that recovers the sign of power would address several of the limits listed above. The Wi-Fi interface of the ESP32 could send the readings to a phone or a logging service, which would also allow the appliance-level analysis discussed in [18] and [19].

A proposed validation set follows from the sections above. It uses three load types: a resistive load such as a lamp, for which the power factor should be near 1; an inductive load such as a fan, for which the reading should be below 1; and a rectifier-type load such as a small power supply, which tests the total power factor of (5) on a distorted current. Each would be recorded at several load levels against a reference meter, with the error plotted against current as in Table V. Repeating the idle-channel records with the present firmware would show how much of the noise in Table III the added oversampling and subtraction actually remove.

VI. CONCLUSION

A microcontroller-based meter was built from an ESP32, a ZMPT101B module and an ACS712 sensor, computing RMS voltage, RMS current, real power, apparent power, power factor, energy and cost from samples of the two channels. The bench records identified the practical difficulty: the current channel carried noise equivalent to 0.59 A with the supply off, which led to noise subtraction and a dead band in the firmware. The voltage constant was corrected from 0.30 to 0.595 V per count using the supply as reference. The computation method is complete. The unit runs, but its accuracy for power, power factor and energy has not yet been established, and that comparison is the next step.

REFERENCES

1. S. S. S. R. Depuru, L. Wang, V. Devabhaktuni, and N. Gudi, “Smart meters for power grid: Challenges, issues, advantages and status,” Renew. Sustain. Energy Rev., vol. 15, no. 6, pp. 2736–2742, Aug. 2011.
2. S. Darby, “The effectiveness of feedback on energy consumption: A review for DEFRA of the literature on metering, billing and direct displays,” Environmental Change Institute, Univ. Oxford, Oxford, U.K., Apr. 2006.
3. C. Fischer, “Feedback on household electricity consumption: A tool for saving energy?,” Energy Efficiency, vol. 1, no. 1, pp. 79–104, Feb. 2008.
4. Electricity Metering Equipment – Particular Requirements – Part 21: Static Meters for Active Energy (Classes 1 and 2), IEC Standard 62053-21, 2020.
5. AC Static Watt-Hour Meters, Class 1 and 2 – Specification, IS 13779, Bureau of Indian Standards, New Delhi, India, 1999.
6. H. S. Kalsi, Electronic Instrumentation, 3rd ed. New Delhi, India: Tata McGraw-Hill, 2010.
7. E. O. Doebelin and D. N. Manik, Doebelin’s Measurement Systems, 6th ed. New York, NY, USA: McGraw-Hill, 2011.
8. A. S. Morris and R. Langari, Measurement and Instrumentation: Theory and Application, 2nd ed. Oxford, U.K.: Academic Press, 2016.
9. J. G. Webster, Ed., The Measurement, Instrumentation, and Sensors Handbook. Boca Raton, FL, USA: CRC Press, 1999.
10. R. S. Popovic, Hall Effect Devices, 2nd ed. Bristol, U.K.: Institute of Physics Publishing, 2004.
11. Allegro MicroSystems, “ACS712: Fully integrated, Hall effect-based linear current sensor IC with 2.1 kVRMS voltage isolation and a low-resistance current conductor,” datasheet. [Online]. Available: https://www.allegromicro.com
12. “ZMPT101B: 2 mA/2 mA voltage transformer module,” manufacturer datasheet.
13. Espressif Systems, “ESP32 series datasheet.” [Online]. Available: https://www.espressif.com
14. Espressif Systems, “Analog to digital converter (ADC) calibration driver,” ESP-IDF Programming

Guide. [Online]. Available: https://docs.espressif.com

15. Silicon Laboratories, "Improving ADC resolution by oversampling and averaging," Application Note AN118.
16. IEEE Standard Definitions for the Measurement of Electric Power Quantities Under Sinusoidal, Nonsinusoidal, Balanced, or Unbalanced Conditions, IEEE Std 1459-2010, Mar. 2010.
17. A. E. Emanuel, Power Definitions and the Physical Mechanism of Power Flow. Chichester, U.K.: Wiley-IEEE Press, 2010.
18. G. W. Hart, "Nonintrusive appliance load monitoring," Proc. IEEE, vol. 80, no. 12, pp. 1870–1891, Dec. 1992.
19. M. Zeifman and K. Roth, "Nonintrusive appliance load monitoring: Review and outlook," IEEE Trans. Consum. Electron., vol. 57, no. 1, pp. 76–84, Feb. 2011.
20. OpenEnergyMonitor, "Learn: Electricity monitoring." [Online]. Available: https://learn.openenergymonitor.org (accessed Oct. 2026).