YoMoPie: A User-Oriented Energy Monitor to Enhance Energy Efficiency in Households

Christoph Klemenjak, Stefan Jost, and Wilfried Elmenreich
Institute of Networked and Embedded Systems
Alpen-Adria-Universität Klagenfurt
Klagenfurt, Austria
{*name.surname*}@aau.at

***Abstract*—Computational methods for the enhancement of energy efficiency rely on a measurement process with sufficient accuracy and number of measurements. Networked energy meters, energy monitors, serve as vital link between energy consumption of households and key insights that reveal strategies to achieve significant energy savings. During the design of such an energy monitor, several aspects such as data update rate or variety of measured physical quantities have to be considered. This paper introduces YoMoPie, a user-oriented energy monitor based on the Raspberry Pi platform that aims to enable intelligent energy services in households. YoMoPie measures active as well as apparent power, stores data locally, and integrates an easy to use Python library. Furthermore, the presented energy monitor comes with a Python API enabling the execution of user-designed services to enhance energy efficiency in buildings and households. Along with the presented design, possible applications that could run on top of this system such as residential demand response, immediate user feedback, smart meter data analytics, or energy disaggregation are discussed. Finally, a case study is presented, which compares the measurement accuracy of YoMoPie to a certified energy analyser for a selection of common household appliances.**

***Index Terms*—Smart Metering, Open-Hardware, Demand Response, Data Analytics, NILM**

I. Introduction

One important step to a more sustainable future is to reduce our energy consumption, in particular to avoid wasting energy. However, when looking at electrical energy we face the problem that even in a small household we have a complex system of electricity consuming appliances. In order to reduce the energy consumption *efficiently* it is helpful to know which devices are responsible for the highest consumption. Currently, most costumers are informed about their energy consumption via a bill that comes months later after the actual energy usage took place. Thus, costumers are hardly able to remember what devices were used during that time and cannot make a connection between certain devices and a respective energy consumption. Providing feedback that is more timely, for example via a wall-mounted display or a website could change this situation by increasing energy awareness which in consequence leads to a lower energy consumption by a change of behaviour. If the information given provides also a separation by devices, the expected savings are higher. Studies [1], [2] indicate that a feedback with short delay and disaggregated to device level (e.g., displaying information in real-time while you use a device) can save more than 12% on energy consumption since users will adjust their consumption behaviour.

As a technical basis for such systems, we need a solution to measure and display the power consumption of multiple devices. One approach would be to add a metering unit to all appliances or have such a metering unit built-in [3] however this would require a high hardware effort and in consequence result in a system with a higher base load for the metering infrastructure, which might cancel out the benefits. A more efficient and therefore sustainable way is to rely on a central meter while applying load disaggregation algorithms [4]. For load disaggregation, it is necessary to regularly measure the overall energy consumption of a household with a smart meter that reports its measurements to a component performing the analysis. Currently, smart meters are installed in many households in the EU to replace the old electricity meters. While these smart meters provide a very exact measure of the consumed energy during the measurement cycle, the frequency of these reports is not very fine-grained. Presumably, a reporting frequency of 15 minutes or even slower is considered, which is not detailed enough to extract information about devices and their consumption. A faster reporting cycle might come with increased communication costs and possible infringe privacy when these data is communicated to the utility. A regular record of energy consumption allows to draw conclusions about the living habits of the residents of the measured household, thus a smart meter used for energy billing should follow the principle of data minimization, only data necessary for this purpose should be generated and communicated. Therefore, we propose a solution with a dedicated meter that stores data only locally within the household and supports frequent measurements for analysing the consumption at device-level without transmitting the data outside the household. Ideally, all required analysis and visualisation functions should run on the same device.

In this paper, we address the question how such a meter could be designed and describe a case study with a smart metering shield combined with a Raspberry Pi minicomputer. Section II investigates on the state of the art in the design of existing smart metering devices. Section III recounts the basic concepts and requirements for smart metering. Section IV describes the proposed system in hardware and software[1]. Section V discusses possibilities and applications that could run on top of the described system. Section VI puts the system to the test by evaluating its measurement performance for active and apparent power.

[1] https://klemenjak.github.io/YoMoPie

TABLE I
A COMPARISON OF ENERGY MONITORS IN ACADEMIA

	AMMeter [5]	ACme-A [6]	YoMo [7]	Demo Board [8]	IoT Sensor [9]	MEDAL [10]	YoMoPie
Communication	x	ZigBee	Wi-Fi	UART	Wi-Fi	Wi-Fi, Ethernet	Wi-Fi, Ethernet, RF
Measurements	I	P,Q,S	P,Q,S,I,V	I,V	P,I,V	P,I,V,f	P,Q,S,I,V
Sampling freq.	1 Hz	1 Hz	1 Hz	14 kHz	x	50 kHz	10 Hz
Power calculation	software	hardware	hardware	hardware	hardware	software	hardware
Open-source	yes	yes	yes	yes	no	no	yes
Basis	Arduino	x	Arduino	ATmega328p	x	ATmega324PA	Raspberry Pi 3
Costs	x	x	€65	€40	€10	x	€43

II. Related Work

In a recent survey, the authors of [11] evaluate several energy monitors and reveal shortcomings with regard to data update rate, diversity of measured quantities, and data confidentiality. Usually, the sampling rate is given as a typical parameter for energy measurement systems, for example [10] and [8] report a comparably high sampling frequency ($f > 1\,\mathrm{kHz}$). However from a systems point of view, the data update rate at which the measuring subsystem is able to provide measurements is more important. The data update rate depends on the measurement hardware and communication system. For instance, the application of UART as communication interface [8] or relaying the samples via WiFi [7], [9] between host and measurement hardware can represent a bottleneck. The variety of metered physical quantities is a decisive factor for the applicability of an energy monitor. According to [11], only 21 % of common e-monitoring solutions on the market measure the line voltage. Knowledge about the voltage level is vital for accurate energy metering since the voltage level is influenced by the present load. Furthermore, quantities such as power factor or line frequency provide deeper insights and should be considered. For instance, the power factor provides information about the ratio between active and apparent power. Existing energy monitors [5], [6], [8], [9] often only provide information of a part of these quantities which makes an assessment and comparison difficult.

III. Concepts of Smart Metering

This section reviews relevant concepts related to smart metering and advanced metering infrastructure (AMI) such as measurement quantities of interest or selection of an adequate sampling frequency for this special kind of application.

A. Energy Metering

In order to obtain a detailed overview of a household, it is obligatory to measure certain physical quantities at characteristic locations such as the feed point of the building or selected distribution boxes of the power distribution grid. These physical quantities allow the generation of accurate appliance and household models. Such models can be used in behavioural analysis to explore how energy efficiency could be improved. With respect to energy metering, physical quantities that have to be considered for monitoring are:

- Active power, P, is defined as the real electrical power of an alternating current circuit and is computed by forming the product of voltage and the part of the current that is in phase with the voltage.

$$P = U \cdot I \cdot cos(\varphi_{UI}) \tag{1}$$

- Reactive power, Q, represents the imaginary component of the power consumed in an alternating current circuit and corresponds to the product of voltage and the part of the current that is out of phase with the voltage.

$$Q = U \cdot I \cdot sin(\varphi_{UI}) \tag{2}$$

- Apparent power, S, is the product of root-mean-square (RMS) value of voltage and current and corresponds to the total power of an alternating current circuit.

$$S = U \cdot I \tag{3}$$

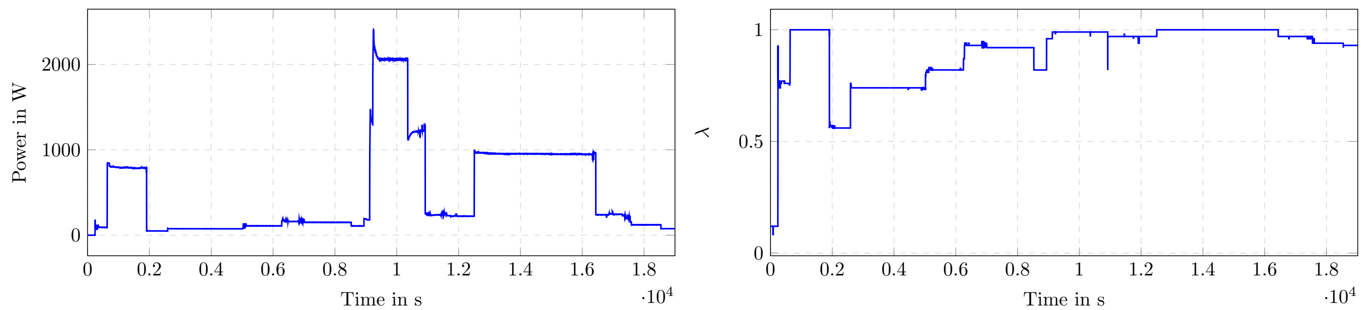
- Power factor, λ, is the ratio of the active to the apparent power in an alternating current power system and can also be expressed as the cosine of the phase shift between voltage and current.

$$\lambda = \frac{P}{|S|} = |cos(\varphi_{UI})| \tag{4}$$

B. Energy Disaggregation

With the roll-out of smart energy meters, the importance of effective load monitoring techniques has risen rapidly since the majority of these meters exclusively provide aggregate power readings. Therefore, techniques are required to extract the power consumption of single appliances out of aggregated power data [12]. Non-Intrusive Load Monitoring (NILM) is a set of techniques applied to identify the power consumption of electrical appliances from measurements taken at a limited number of locations in the power distribution grid of a building [13]. Recently, approaches based on machine learning experienced growing interest and contributions focused on evolutionary computing [14], Autoencoders, Convolutional Neural Networks (CNN), and Long Short-Term Memory (LSTM) [15]. NILM techniques operate in the intermediate layer between measurement layer and software services that rely on appliance-specific information. Therefore, NILM is an essential building block in situation, where sub-metering is not applicable or not possible.

A big challenge of load monitoring is performing measurements in real-time. This is an essential requirement for several applications in order to react to events inside the building e.g. arrival of residents, power faults, and defective appliances.



(a) Measured active power consumption over time

(b) Computed power factor over time

Fig. 1. Application example for YoMoPie

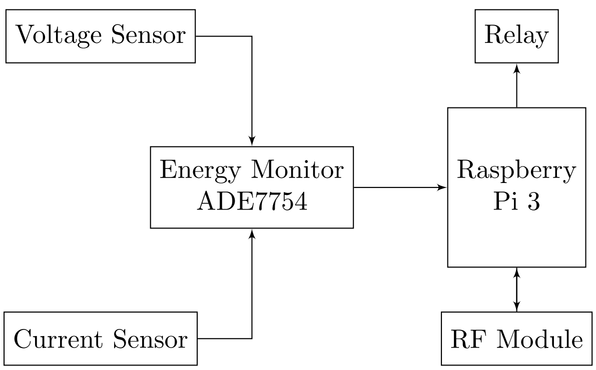


Fig. 2. YoMoPie hardware components

Researchers demonstrated that NILM algorithms can be ported to mini computers such as Raspberry Pi in order to perform real-time energy disaggregation [16].

IV. The YoMoPie System

YoMoPie is an energy monitor consisting of a Raspberry Pi and a customised extension board. Figure 4 shows the current version of the extension board. The measurement equipment is installed between the mains cable and electric appliances. The current version of YoMoPie is designed to support currents up to $16\,\mathrm{A}$ and therefore, can monitor a maximum electric load of $3.68\,\mathrm{kW\,h}$ at $230\,\mathrm{V}$. For this reason, YoMoPie can be installed in the distribution board of a small household and record household characteristics at the feed point. Table I lists further specifications. Figure 1 shows an application example for YoMoPie. In this application, active power and apparent power of a simulated household (lab setup) are recorded over time and the power factor is computed. The recorded measurement samples can be accessed instantly to process and analyse the obtained information. The application gives an impression of what YoMoPie provides to prospective users. The custom-designed extension shield integrates sensors, a relay, and communication interfaces. Figure 2 provides an overview of the hardware components of YoMoPie. A voltage divider between phase and neutral wire serves as input for an isolation amplifier, which ensures galvanic isolation between the voltage divider and the remaining electronic circuit. The output of the isolation amplifier serves as the voltage signal for the energy monitoring chip ADE7754. The current of the main wire flows through a current transformer that produces an output signal proportional to the current. This output signal serves as current signal for the energy monitor. The energy monitoring chip ADE7754 is a special kind of integrated circuits that provides readings of active energy, apparent energy, voltage RMS, and current RMS. In our application, a Raspberry Pi is used as main processing unit. The integrated mini computer communicates with the energy monitor chip to read measurements and set mode registers. Another communication interface is provided through the integration of a RF module, an NRF24L01+. This allows YoMoPie to interact with other networked (measurement) devices over the air in the low-power $2.4\,\mathrm{GHz}$ ISM band.

YoMoPie runs on the operating system Raspbian OS. Raspbian is a wide-spread operating system that is based on Debian Linux and shares a big number of common software packages with conventional Linux distributions such as the database management system MySQL, the high-level programming language Python, or the Apache HTTP server. A Python library provides an API (Application Programmer Interface) enabling easy access to the functionalities of YoMoPie. This API implements several SET and GET methods to access relevant registers of ADE7754 and utilise SPI as communication interface. For example, the current amount of consumed active power can be obtained with one simple command:

```
[time,power] = YoMoPie.get_active_energy()
```

Based on the library, a big variety of services is able to utilise YoMoPie and access energy readings in real-time in a simple manner.

V. Enabling Services

YoMoPie establishes hardware and software for an extendible energy monitoring system. The energy monitor integrates measurement hardware that monitors the power consumption in the household. The YoMoPie library allows convenient access to the energy readings as well as to the several hardware features of the system. Based on the provided functionalities of the library, we will discuss several Smart Grid relevant services such as *residential demand response*, *immediate user feedback*, and *data analytics*. Beside such services, our system allows to define customized services for the residents.

A. Residential Demand Response

Demand response (DR) techniques aim to shape the energy demand in a way to match the available electrical energy without adding new generation capacity [17]. In the residential sector, a key element to enable demand response is the smart meter. A smart meter or an appropriate energy monitor could communicate with the utility company, negotiate upon further steps, and schedule processes in the household. DR techniques can be divided into *incentive-based* and *price-based* approaches. Incentive-based schemes animate residents to reduce their energy demand with the help of a request offer [18]. A set of potential offers is defined in a contract between the resident and the utility company. Examples for incentive-based schemes are Direct Load Control (DLC), Emergency, or Bidding. In particular, the demand-bidding response programme represents a possible application for YoMoPie. In this programme, the utility offers a discount for a certain load reduction or load shifting. Depending on the user's defined threshold, the YoMoPie could reply to the bid and initiate a load reduction in the household. In price-based schemes, customers are confronted with time-varying costs for electricity [19]. As a result of these time-dependent prices, customers are invited to individually adapt their load scheduling. By shifting their energy consumption from peak hours to less congested times of the day, costs can be reduced. Wide-spread price-based scheme types are Time of Use (ToU), Critical Peak Pricing (CPP), and Real-Time Pricing (RTP). For instance in Time of Use, YoMoPie could play an important role by creating optimised load schedules for user-picked household appliances. Pre-defined algorithms or tailored load scheduling algorithms, which are provided by the resident, could serve to create and implement a certain load schedule.

B. Immediate User Feedback

The YoMoPie system measures energy consumption within the household. Through the integrated RF communication interface, the system also allows communication with networked sensors that provide ambient characteristics of the household such as temperature, noise level, or occupancy. As pointed out by the authors in [4], immediate feedback can help achieving energy savings up to 10%.

Energy advisor tools analyse and interpret measurement data in order to provide meaningful feedback to residents. The open-source feedback system Mjölnir was presented in [2]. This open-source framework consists of several widgets that illustrate consumption and production data, give detail about the current energy tariff, show real-time data of the present appliances as well as display current energy consumption events in form of a time line widget. Furthermore, the framework is capable of generating advices and recommendations based on provided data.

In the course of a brief case study, we examined the compatibility of the energy monitor YoMoPie with the energy advisor tool Mjölnir. In this study, six common household appliances were selected for real-time user feedback. During the preparation, Mjölnir was installed on YoMoPie and the household appliances were connected to YoMoPie. Figure 3

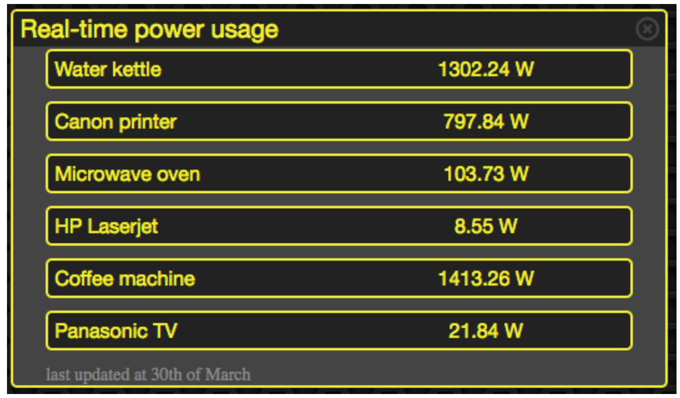


Fig. 3. Real-time power consumption of selected household appliances

shows the utilised real-time feedback widget of Mjölnir. The widget reports the instantly amount of power that the household appliances consume. From the case study shows a simple way of utilising YoMoPie to analyse and interpret energy readings of a particular household.

C. Tailored services

The architecture of YoMoPie allows not only for local processing of the measured data, but the flexibility of its structured software system allows also an extension by new services and algorithms that are user-contributed. A user might come up with a specific need and an idea how to realize this with the provided measurements, computation and interface of YoMoPie. With the right interface, users with minimum programming experience should be able to draft a new application that can be uploaded on the YoMoPie. Following this idea, YoMoPie can provide the basis for user-contributed apps, similar to those we have at smartphones. An example for such an app could be to visualize the energy consumption via an appealing image similar to the polar bear depicted on the Amphiro system used to appeal to the user [20].

Requirements for such a system are:

- *Sandbox:* User-contributed code should be well isolated from the runtime environment of YoMoPie and restricted in abilities to communicate to the Internet or to control systems
- *Simple user interface:* The configuration, packing and testing should be straightforward so that a person without experience in coding can use it
- *Unified representation of devices and interfaces:* Since the set of devices will differ between households there is a need for a unified way of addressing and describing them, so that an app works under different configurations as it is envisioned in [21].
- *Publishing and Sharing:* There should be a way to publish and share interesting apps with others.
- *Feedback and voting system:* This gives the possibility to communicate problems or suggestions to the publisher and to identify the most suitable solutions.
- *Open source concept:* Published apps should come with a license that allows use and improvement by others. This way, existing apps can be improved by different users.

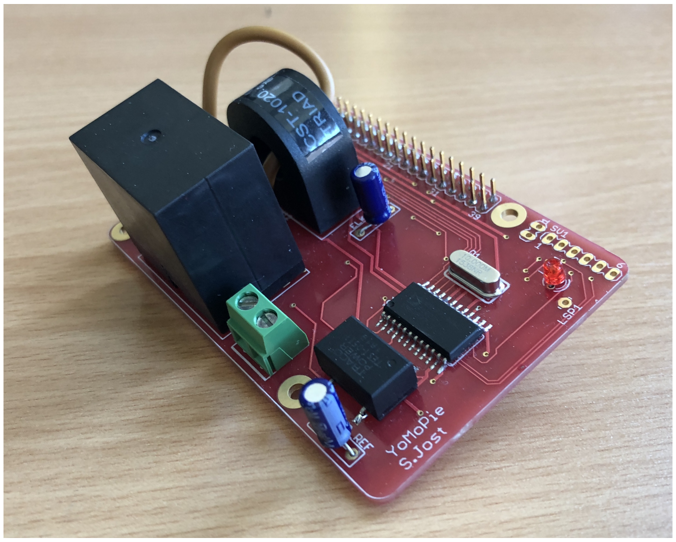


Fig. 4. Current version of YoMoPie's metering board

The current version of YoMoPie does provide the hardware and system architecture for such an approach, however implementing such a system is a subject of future work.

VI. Evaluation

The purpose of energy monitors is to measure power consumption of households or of selected rooms. In any case, the monitor will record data that contains power consumption patterns of several appliances. To identify individual appliances within the recorded data, NILM algorithms can be utilised. Researchers have demonstrated that such algorithms can be ported to embedded hardware [22], [16]. Before applying such algorithms and utilising these new insights, an essential parameter of an energy monitor has to be evaluated, the *measurement accuracy*. In this section, the authors present the results of two conducted case studies, where measurement accuracy for active power and apparent power were evaluated.

In the case studies, five common household appliances were used to evaluate the measurement accuracy of YoMoPie: an electric heater, a baking oven, a cooling fan, a light bulb, and a refrigerator.

To record the ground truth, the authors used an HMC8015 power analyser instrument. In the active and in the apparent power case, the measured samples of YoMoPie were compared to the recorded ground truth as well as to a commercial smart meter, the Carlo Gavazzi EM24.

Table II shows the results of the case study for active power. In general, the EM24 meter shows measurement errors below 2% for active power in this case study. The only exception represents the refrigerator case with an error above 6%, which could be a consequence of the low power factor of 0.67 of the refrigerator. YoMoPie shows a similar trend but with slightly higher measurement errors. However, the measurement error didn't exceed 4% for any active power measurement in this study.

Table III shows the results of the second case study, which focusses on apparent power measurements. The study involves two kinds of common household appliances: appliances with a small amount of reactive power and appliances with a significant amount of reactive power such as refrigerators. It can be observed for both energy monitors that the measurement error is affected by the amount of consumed reactive power. For household appliances with a significant amount of reactive power, the case study reports a significantly increased measurement error compared to cases, where the reactive power is close to zero. For instance, YoMoPie shows a measurement error of 2.28% for measurements at the cooling fan and 6.63% for the refrigerator although both appliance have a similar apparent power consumption.

Both case studies, active power and apparent power, indicate that the presented energy monitor YoMoPie achieves similar measurement accuracy as a commercial smart meter. It has to be noted that this measurement accuracy reflects how accurately the energy monitors were able to meter the aggregate power, which was measured at a simulated feed point of a certain room.

VII. Discussion

The proposed energy monitor is able to measure the energy consumption of selected appliances or an entire household. With regard to the integration of renewable energy technologies into households, YoMoPie could serve as a monitoring solution to quantify the energy provided by photovoltaics or adequate wind turbines. As discussed in [23], networked energy sensors allow assessing the energy produced by a domestic renewable energy source. Such a networked sensor would provide information on energy production to a central unit (i.e. YoMoPie) by means of RF communication. Alternatively, a second YoMoPie device could serve to monitor not only the production of the energy source but also environmental characteristics by the application of ambient sensors.

VIII. Conclusions

This paper presented a user-oriented energy monitor based on the Raspberry Pi platform. In contrast to existing energy monitors, YoMoPie measures active as well as apparent power, stores the data locally, and provides a Python API enabling a variety of services to access the energy readings. We presented test results on active power and apparent power measurements indicating that the presented low-cost energy monitor provides a measurement accuracy comparable to commercial smart meters. In conjunction with energy disaggregation algorithms, YoMoPie has the potential to enable immediate feedback, demand response, data analytics, as well as self-designed services to enhance energy efficiency in buildings and households. In particular, future work will investigate how the measurement accuracy of the system can be further increased, how YoMoPie can be extended to support polyphase metering and how the proposed applications can be implemented in a simple and effective manner. Of special interest will be data analytics techniques with regard to smart meter data to enable autonomous load analysis, load forecasting and load management inside the household.

TABLE II
MEASUREMENT ACCURACY FOR ACTIVE POWER

Appliance	Power [W]	Power factor λ	EM 24 - Smart Meter meas. Power [W]	EM 24 - Smart Meter Error [%]	YoMoPie meas. Power [W]	YoMoPie Error [%]
Refrigerator	57.59	0.67	61.54	6.80	59.84	3.91
Light bulb	58.88	0.99	59.65	1.29	59.07	0.32
Cooling fan	73.36	0.91	74.25	1.20	75.05	2.29
Baking oven	676.76	0.99	678.39	0.24	685.70	1.32
Electric heater	1020.40	0.99	1026.40	0.59	1044.50	2.37
Baking oven	1212.00	0.99	1217.00	0.50	1252.00	3.30
Electric heater	1776.00	0.99	1780.20	0.14	1804.00	1.48

TABLE III
MEASUREMENT ACCURACY FOR APPARENT POWER

Appliance	Power [VA]	Power factor λ	EM 24 - Smart Meter meas. Power [VA]	EM 24 - Smart Meter Error [%]	YoMoPie meas. Power [VA]	YoMoPie Error [%]
Light bulb	59.41	0.99	59.54	0.21	60.55	1.92
Cooling fan	80.73	0.91	82.46	2.14	82.57	2.28
Refrigerator	81.01	0.67	86.72	7.06	86.37	6.63
Baking oven	678.28	0.99	681.44	0.47	697.37	2.82
Electric heater	1037.80	0.99	1046.50	0.84	1065.30	2.64
Baking oven	1217.60	0.99	1227.40	0.81	1257.50	3.27
Electric heater	1797.60	0.99	1810.20	0.70	1808.10	0.59

ACKNOWLEDGEMENTS

This work was partially supported by Lakeside Labs via the Smart Microgrid Lab.

REFERENCES

[1] K. Erhardt-Martinez, K. Donnelly, and J. A. S. Laitner, "Advanced metering initiatives and residential feedback programs: A meta-review for household electricity-saving opportunities," American Council for an Energy-Efficient Economy, Washington, D.C., Tech. Rep., 2010.

[2] A. Monacchi, F. Versolatto, M. Herold, D. Egarter, A. M. Tonello, and W. Elmenreich, "An open solution to provide personalized feedback for building energy management," *Journal of Ambient Intelligence and Smart Environments*, vol. 9, no. 2, pp. 147–162, 2017.

[3] W. Elmenreich and D. Egarter, "Design guidelines for smart appliances," in *Intelligent Solutions in Embedded Systems (WISES), 2012 Proceedings of the Tenth Workshop on*. IEEE, 2012, pp. 76–82.

[4] K. C. Armel, A. Gupta, G. Shrimali, and A. Albert, "Is disaggregation the holy grail of energy efficiency? the case of electricity," *Energy Policy*, vol. 52, pp. 213–234, 2013.

[5] S. Makonin, W. Sung, R. D. Cruz, B. Yarrow, B. Gill, F. Popowich, and I. V. Baji, "Inspiring energy conservation through open source metering hardware and embedded real-time load disaggregation," in *2013 IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC)*, Dec 2013, pp. 1–6.

[6] X. Jiang, S. Dawson-Haggerty, P. Dutta, and D. Culler, "Design and implementation of a high-fidelity ac metering network," in *2009 International Conference on Information Processing in Sensor Networks*, April 2009, pp. 253–264.

[7] C. Klemenjak, D. Egarter, and W. Elmenreich, "YoMo: the Arduino-based smart metering board," *Computer Science-Research and Development*, vol. 31, no. 1-2, pp. 97–103, 2016.

[8] M. Quintana, H. Lange, and M. Bergés, "Demo: Design and implementation of a low-cost Arduino-based high-frequency ac waveform meter board for the Raspberry Pi," in *Proceedings of the 4th ACM International Conference on Systems for Energy-Efficient Built Environments (BuildSys'17)*. ACM, 2017, pp. 249–250.

[9] A. R. Jadhav and P. Rajalakshmi, "Iot enabled smart and secure power monitor," in *IEEE Region 10 Symposium (TENSYMP), 2017*. IEEE, 2017, pp. 1–4.

[10] T. Kriechbaumer, A. Ul Haq, M. Kahl, and H.-A. Jacobsen, "Medal: A cost-effective high-frequency energy data acquisition system for electrical appliances," in *Proceedings of the Eighth International Conference on Future Energy Systems*. ACM, 2017, pp. 216–221.

[11] A. U. Haq and H.-A. Jacobsen, "Prospects of appliance-level load monitoring in off-the-shelf energy monitors: A technical review," *Energies*, vol. 11, no. 1, p. 189, 2018.

[12] C. Klemenjak and P. Goldsborough, "Non-intrusive load monitoring: A review and outlook," in *INFORMATIK 2016*, H. C. Mayr and M. Pinzger, Eds. Bonn: Gesellschaft für Informatik e.V., September 2016, pp. 2199–2210.

[13] N. Batra, O. Parson, M. Berges, A. Singh, and A. Rogers, "A comparison of non-intrusive load monitoring methods for commercial and residential buildings," *arXiv preprint arXiv:1408.6595*, 2014.

[14] D. Egarter, A. Sobe, and W. Elmenreich, "Evolving non-intrusive load monitoring," in *European Conference on the Applications of Evolutionary Computation*. Springer, 2013, pp. 182–191.

[15] J. R. Herrero, Á. L. Murciego, A. L. Barriuso, D. H. de la Iglesia, G. V. González, J. M. C. Rodríguez, and R. Carreira, "Non intrusive load monitoring (nilm): A state of the art," in *International Conference on Practical Applications of Agents and Multi-Agent Systems*. Springer, 2017, pp. 125–138.

[16] S. Makonin, F. Popowich, I. V. Bajić, B. Gill, and L. Bartram, "Exploiting hmm sparsity to perform online real-time nonintrusive load monitoring," *IEEE Transactions on Smart Grid*, vol. 7, no. 6, pp. 2575–2585, 2016.

[17] H. T. Haider, O. H. See, and W. Elmenreich, "A review of residential demand response of smart grid," *Renewable and Sustainable Energy Reviews*, vol. 59, pp. 166–178, 2016.

[18] M. Nikzad and B. Mozafari, "Reliability assessment of incentive-and priced-based demand response programs in restructured power systems," *International Journal of Electrical Power & Energy Systems*, vol. 56, pp. 83–96, 2014.

[19] J. Torriti, "Price-based demand side management: Assessing the impacts of time-of-use tariffs on residential electricity demand and peak shifting in northern italy," *Energy*, vol. 44, no. 1, pp. 576–583, 2012.

[20] V. Tiefenbeck, "Behavioral interventions to reduce residential energy and water consumption: Impact, mechanisms, and side effects," Ph.D. dissertation, ETH Zurich, 2014.

[21] D. Egarter, A. Monacchi, T. Khatib, and W. Elmenreich, "Integration of legacy appliances into home energy management systems," *Journal of Ambient Intelligence and Humanized Computing*, vol. 7, no. 2, 2016.

[22] D. Egarter, V. P. Bhuvana, and W. Elmenreich, "PALDi: Online load disaggregation via particle filtering," *IEEE Transactions on Instrumentation and Measurement*, vol. 64, no. 2, pp. 467–477, 2015.

[23] C. Klemenjak and W. Elmenreich, "YaY - An open-hardware energy measurement system for feedback and appliance detection based on the Arduino platform," in *2017 13th Workshop on Intelligent Solutions in Embedded Systems (WISES)*. IEEE, 2017, pp. 1–8.

Appendix

A. Setup

The YoMoPie Python package is available on Python Package Index (PyPI), a repository of software for the Python programming language, and can be installed by issuing one command:

```
pip3 install YoMoPie
```

Additionally, the entire source code and a manual can be obtained from the YoMoPie Github page[2].

B. User's Guide

After a successful installation process, the YoMoPie package is available system-wide and can be accessed by a simple import command:

```
import YoMoPie as yomopie
yp = yomopie.YoMoPie()
```

During initialisation, the number of line conductors has to be set (single or polyphase metering):

```
yp.set_lines(1)
```

Active power, apparent power, current, and voltage samples can be read with commands such as:

```
[t, I] = yp.get_irms()
[t, U] = yp.get_vrms()
[t, P] = yp.get_active_energy()
[t, S] = yp.get_apparent_energy()
```

In the same vein, users can activate continuous data logging or perform a fixed amount of subsequent measurements:

```
yp.do_metering()
yp.do_n_measurements(quantity, rate, file)
```

The operational mode (OPMODE) register defines the general configuration of the integrated measurement chip ADE7754. By writing to this register, A/D converters can be turned on/off, sleep mode can be activated, or a software chip reset can be triggered. For further information, we refer to the datasheet of the measurement chip.

```
yp.set_operational_mode(OPMODE)
```

It should be noted that the user guide in this appendix reflects only an excerpt of the available documentation, which is available on the Github project page and receives regular updates.

[2]https://klemenjak.github.io/YoMoPie