ECE 445

Spring 2020

Group Project Design Document

Submetering the ECEB

Xiaolin Wu (xw5), Xiaoyi Shen (xiaoyis2), Tingyu Wang (tingyuw2)

Team Number: #2

TA: Evan Widloski

Prof: Jonathan Schuh

Table of Contents

1. Introduction

1.1. Problem and Solution Overview

Although our ECE building has the measurement of the total power generated by the solar panels on the roof and has its measure of total power consumption, it still lacks the ability to track the power consumed by each and every classroom, lab, and office. So our **problem** to solve in this project is to design a system that is able to measure the power used in individual rooms in ECEB. Therefore, aiming at the need to solve the issue for detection and display of the power consumed , our project can measure the power used in various labs and classrooms as well as offices in ECEB and have those data displayed in real time on a screen set on the first floor of the building. To accomplish this, our **solution** includes several local sensing systems that are placed in different rooms as well as a web-server that communicates with all the devices through Wi-Fi and have it displayed on the desired screen. From the aspects of customer benefits, our project could help our administrators gain a better understanding about the distribution of the power consumption in this building and thus benefit any of their further decisions for the building.

Currently, there are devices that can measure the AC power consumption using Arduino which is a little bit more expensive than our choices[1]---- where we build our communication systems based on ESP32s and we are using cloud server AWS, thus the cost of the whole project will be limited to a satisfactory state. Besides, if an Arduino is involved in the design, we would also acquire an external data transfer module (also see the figure below) which will make the project bigger and cost even more to build. In addition, due to our projects' capability to measure both 3-phase and single phase electric circuits, it is thoroughly very marketable.

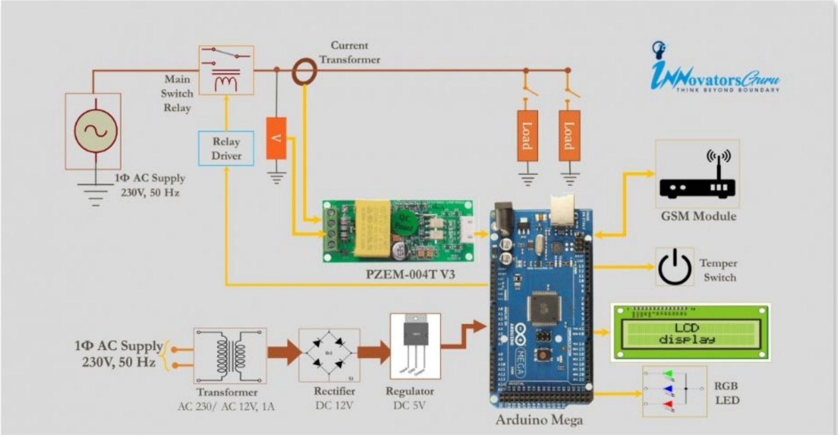

Figure 1. Previous works

1.2. Visual Aid

For a similar visual design of ours, I pictured our project as following, where the current transformer will be wired around the current wire for powering the room and so does the voltage transformer as indicated in the graph. Then the lowered current and voltage will be sent to a local resistor circuit and be mapped to the range(will be discussed below) for the analog pins of the microprocessor. Then multiples of the local devices will communicate with the AWS server for the screen to display.

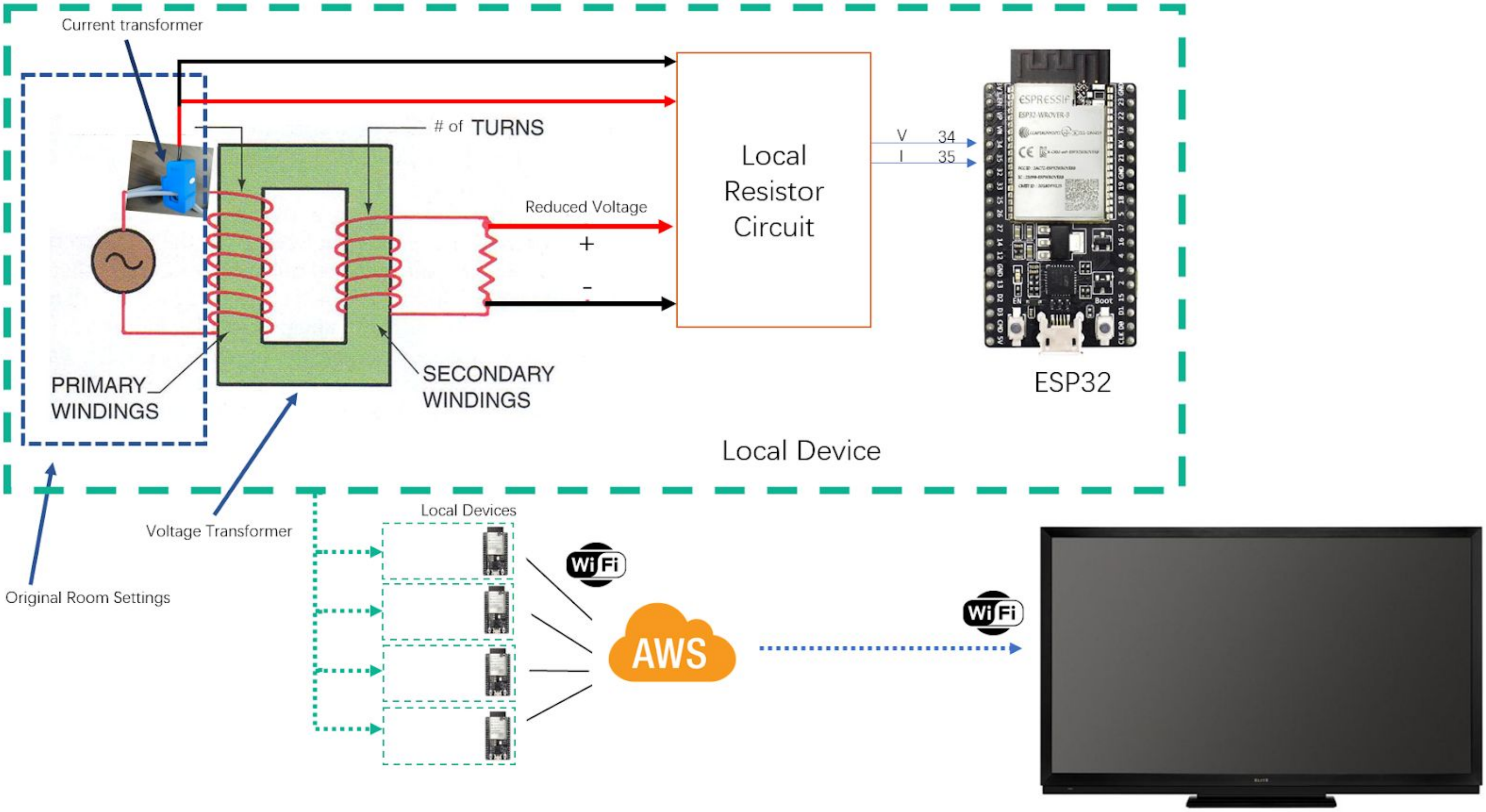

Figure 2. Visual Overview of the design

1.3. High-Level Requirement List

- Functions to measure both 3-phase and single phase power with high voltage (~208V) and high current (200A ~ 400A) with 5% accuracy.
- Ability to offload data (both real power and apparent power in Watts) for displaying and the data to web server at least 4 times per hour.
- Have enough local storage (up to 24 hours) of power measurement data.

2. Design

2.1. Block Diagram

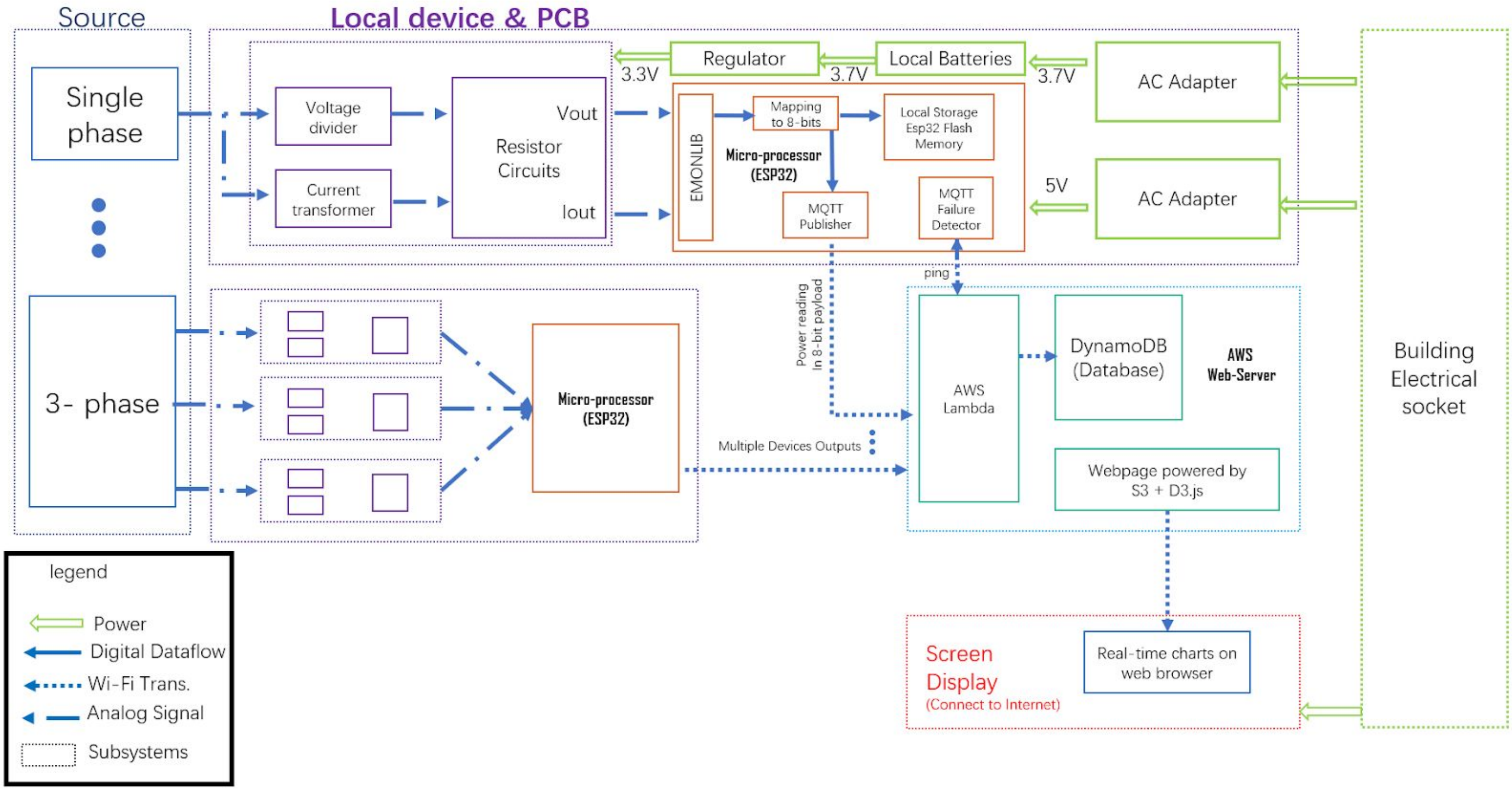

Figure 3. Block Diagram

On a larger picture of our design, we will have multiple local devices that directly take in the voltage and current powering the room and using an ESP32 as local microprocessor to help calculate the power consumption. After we generate the data, we will utilize WIFI to communicate with aws web-server and eventually the data is offloaded to a screen that can access the server and have the data displayed as we desired. The following paragraphs will explain in detail about our design by subsystems.

2.2. Subsystem Descriptions

2.2.1. Source Description

To the very beginning of our project design, the source module is representing the objects to be measured. Based on the information from the department, most of the rooms in ECEB use single phase power except ADSL, Open Lab, Optics and the power labs, ie.4020, 4024, 4026. Besides, 4020 and 4024 also have 240V DC power source that may need additional attention.

In our design, we now planned to build a measurement circuit design for single phase circumstances and for the 3-phase occasions, we would just apply the same circuits three times on each phase and add them together in the local microprocessor. The reason for doing this is that we do not want to waste resources on applying a complicated three phase design for single phase occasions, leaving the other two useless. Furthermore, we cannot ensure that the loads on each phase of the 3-phase labs are the same so measuring the power consuming in each phase seems to be our best solution.

For **requirements and verifications**, there is none since the sources are given as external resources.

2.2.2. Local device (Design on PCB)

2.2.2.1. Voltage divider

The voltage divider, chosen to be ZMPT107-1, isolates the measurement system from high voltage for safety and steps down the voltage from 208V to within 10V.

2.2.2.2. Current Transformer

The current transformer, chosen to be CTF-5SFT-0200[2], can accept the primary current up to 200A and convert it to 5A secondary current with 1% accuracy and 60HZ frequency.

2.2.2.3. Local Mapping using Resistor Circuits

The resistor circuit consists of a voltage divider circuit to convert the voltage to a measurable range(0~3.3V), which can be accepted by our local microprocessor, ESP32, and a resistor circuit to convert the current from 5A to 0~40mA. A 3.3V DC bias will be added to shift the voltage and current waveform above 0. A burden resistor will be added to convert the current to voltage for the microprocessor(esp32) to read.

2.2.2.4. Lithium Ion Polymer Battery and Regulator

The linear voltage regulator, chosen to be LD1117-3.3 TO-220 (3.3V 800mA), can provide the 3.3V dc bias from Lithium Ion Polymer Battery(3.7v 100mAh) to resistor circuit.

Requirements	Verifications
1. The voltage that the Analog pins read in ESP32 should between 0~3.3V 2. The current flowing into Analog pins in ESP32 should be smaller than 50mA	1.connect the Agilent Multimeter parallel to test that voltage which ESP32 read is smaller than 3.3V 2. connect the Agilent Multimeter in series to test the current flowing into the esp32 is smaller than 50mA

2.2.3. Local Microprocessor

We use ESP32 as the microprocessor for our local device. The output from the circuit was read through the analog input pins. We will use EmonLib[3] to calculate the power based on the current and voltage readings we collected from two separate analog input pins. The ADC inside ESP32 will utilize EmonLib to do the Analog-Digital Conversion for us. The power reading then will be mapped to 8 bit binary data. The data will be stored to local flash memory and also be packed into the payload to be sent to the web server using MQTT protocol.

2.2.3.1. EmonLib

Based on our research online, we found a public library running in the Arduino environment that is used to monitor the AC power consumption of people's houses.[4]

We looked into the code and tried to modify it to work on the ESP32 for our task. Ideally, we can use this library to read in analog inputs representing our voltage and current respectively and have the power calculated directly as output. The formula for apparent power is

$$Pa = Irms \ * \ Vs$$

where Pa represents the apparent power and Irms is the root-mean-square of the measured current in the loop and Vs is the total supply voltage in the circuit.

Hence, the real power we care about is calculated as

$Irms = Ipeak/\sqrt{2}$

$Vrms = Vpeak/\sqrt{2}$

$powerfactor = cos(\theta v - \theta i)$

$real\ power\ \ = Irms * Vrms * powerfactor$

The current and voltage are sampled at a rate of 5588 samples per second[5] which is sufficient to deal with a 60Hz original measure target as we have been told. ESP32 initializes the serial connection at 300 bits per second in order to get rid of the redundant outputs with the same meaning for us to record since our desired output is simply 2 representative samples per minute as required.

2.2.3.2. Local Storage

The ESP32 we use has 4MB flash memory. We will implement flash memory as our storage for power reading. The results will be mapped into 8bits. The codes we will use are:

```
EEPROM.write(address, value);
EEPROM.commit();
EEPROM.read(address)
```

2.2.3.3. MQTT Publisher

We will implement AWS IoT on ESP32. More specifically, we will use AWS MQTT[6], a publish/subscription protocol commonly used in IoT for our transmission. We will use the publish function to publish the readings to the server.

2.2.3.4. MQTT Failure Detector

Local devices heart-beat(continuously publish message) to the server to ensure devices are online and functioning. Local devices will also respond to pings from the server.

Requirements	Verifications
1. Microprocessor is able to offload data to the server at least 4 times per hour. 2. Microprocessors have enough local storage. 3. The data is accurate (within 5% error).	1. With any given power values on the ESP32, the web-server should read those values consistently . 2. The power records for a duration of 24 hours can be tracked. 3. At the lab-scope simulation, the power displayed from ESP32 on the serial console matches the ideal value with less than 5% error.

2.2.4. Server

2.2.4.1. Server

We use AWS(Amazon Web Services)[7] as a server that accepts all data from local microprocessors through AWS MQTT. The server is the center node in the communication system and will listen to all the messages sent from all the local microprocessors for data collection and failure detection. It will store daily total power usage to an online database and will also display real time power usage on a webpage for the Screen Display Module.

2.2.4.2. AWS Lambda

We use AWS Lambda[8] to subscribe to the topics that the local microprocessors publish to. AWS Lambda could run Python Scripts that analyze the messages and stored them into the database accordingly.

AWS Lambda is also responsible for failure detection. The local microprocessor will keep sending in power usage as a heartbeat. If AWS Lambda failed to detect any messages from a specific device for more than 30 seconds this device is believed to be offline, AWS Lambda will then send a ping to this device to double check. If no response is given

AWS Lambda will send notifications to maintenance personnel through sms or email.

2.2.4.3. DynamoDB

We use AWS DynamoDB[9] to store the daily total energy for up to 4 years. DynamoDB is a NoSQL database that will work with AWS Lambda to handle the message sent from microprocessors and stores the power usage readings.

2.2.4.4. Webpage

We use D3.js to visualize real-time power usage stored in dynamoDB. The webpage will be hosted on AWS S3[10].

2.2.5. Screen Display

We will need a monitor to display the power results to the public. This could be any device that our department is willing to provide with us. The content is provided by the Display Interface.

Requirements	Verifications
A high resolution screen with Google Chrome browser.	Letters and charts should be seen clearly by observers. The aspect ratio should not be changed.

3. Cost and Schedule

3.1. Cost Analysis

3.1.1. Labor

For each partner of the project:

$\$30/hr \; * \; 8hrs/week \; * \; 11weeks \; * \; 2.5 \; = \; \$6600$

This should be our labor value building this project.

3.1.2. Parts

Our parts will include the components used in only one device for single phase measuring

Part name	Cost (in $)
ESP-WROOM-32 CP2102 (amazon)	13.99
Assorted resistors, capacitors, (Digikey; est.)	4.00
Solid Core AC transformer CTF-5SFT-0200 (automationdirect)	17.00
ZMPT107 (ebay)	2.33
Lithium Ion Polymer Battery - 3.7v 100mAh (adafruit)	5.95
3.3V 800mA Linear Voltage Regulator LD1117-3.3 TO-220LD1117-3.3 TO-220 (adafruit)	1.25
110V-240V AC to 3.7V DC Li-ion Battery Charger w/ DC Connector (TENERY POWER)	8.99
Total:	53.51

3.1.3. Grand Total

Thus, the total amount of value for a project will count as 3* the labor of each person in the group and the cost of the local device. That is,

$$6600 * 3 + 53.51 \; = \; \$19853.51$$

3.2. Schedule

Week	Xiaolin	Xiaoyi	Tingyu
2/24	Verify the resistor circuit designs	Get familiar with using Arduino IDE to burn Arduino Framework on ESP32, test Wifi Connections	Verification of EmonLib's capability for working on ESP32
3/2	Choose current transformer and voltage transformer	Draft more detailed software architecture and start programing ESP32 codes in an iterative development manner.	Confirm the target Voltage and Current range for measuring of the object room and verification
3/9	Circuits combination with the battery design and rough schematics drawing	Continue ESP32 side programming, begin AWS IoT set up.	Finish and sent PCB orders
3/16			
3/23	Design rough schematics for three phase measurement in lab-simulation environment.	Finishes ESP32 codes and finish AWS IoT setup. Verify MQTT transmission between ESP32 and AWS	Test the measurement for single phase circuit on lab-analog environment
3/30	Test the circuit for three phase measurement under lab-simulation environment	Finish Lambda codes and DynamoDB setup.	Continue on the test and debugging for solid use of the measurement for single phase occasions
4/6	Finalize the design for three phase measurement	Set up a webpage on S3, complete the structure of the scripts.	Finish and send out the PCB for three phase measurement.
4/13	Final test for single phase local measurement device	Set up the real-time chart using D3.js.	Check the data writing rate to the web-server
4/20	Continue on the test for solid use of the measurement for three phase occasions	System test. Prepare Mock Demo.	Test on the combination of the functionality of the whole project.
4/27	Verify the general functionality of measuring system	Begin final report and presentation	Begin final report and presentation
5/4	Hand in Everything	Hand in Everything	Hand in Everything

4. Tolerance Analysis

4.1. Resistor Circuit

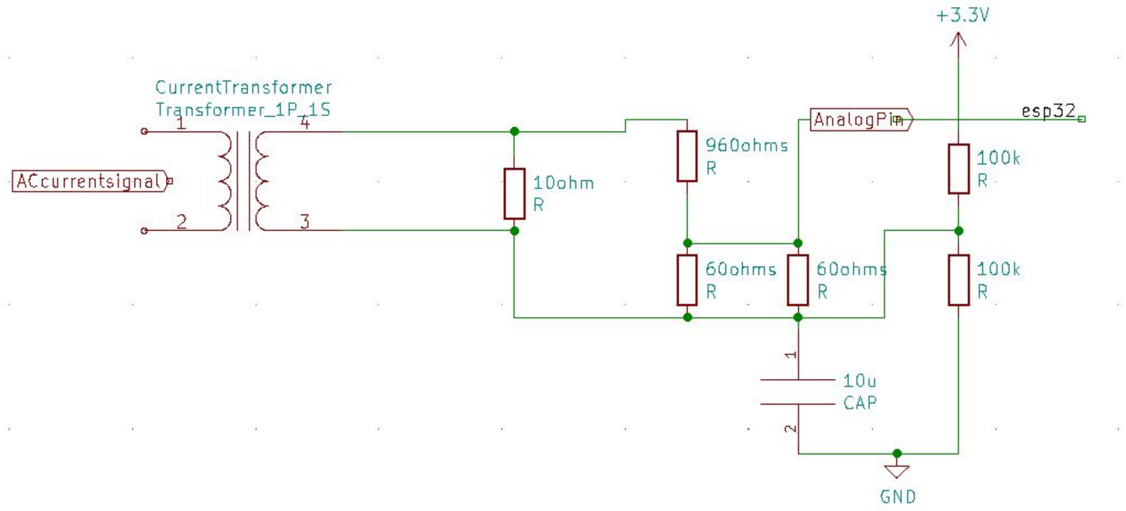

Figure 4(a). Schematic of voltage measurement

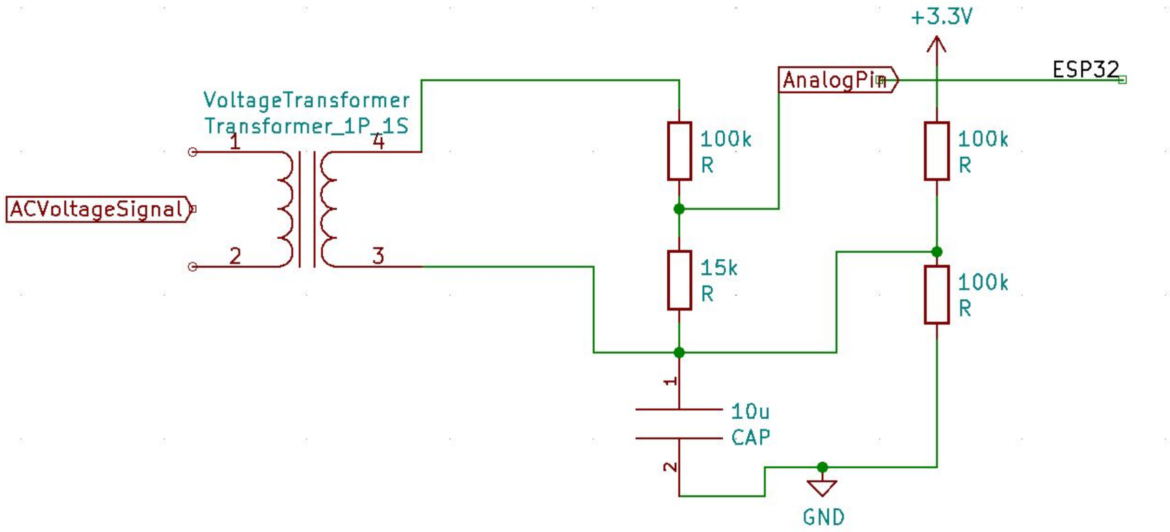

Figure4(b). Schematic of current measurement

In Fig4(a), the secondary current is up to 5A, which is not small enough for ESP32. According to the current divider rule, the current flowing through the

Resistor(960ohms) is 50mA and the current flowing through the Resistor (60ohms) is 25mA.

$$5A * (10 \div ((960 + (1/(1/60 + 1/60)) + 10)) = 50mA$$

$$50mA * 60ohm = 1.5V$$

To maximise measurement resolution, the voltage across the burden resistor(60hms) at *Ipeak* should approximately equal to one half of the analog reference voltage (3.3v) of ESP32.
In Fig4(b), the voltage divider ZMPT107 steps down the voltage from 208V to 12.65V. The following two resistors(100k and 15k) turn the voltage into 1.65V.

4.2. EmonLib

For the verification on the EmonLib measurement, a simple circuit consuming AC power is designed as follows:

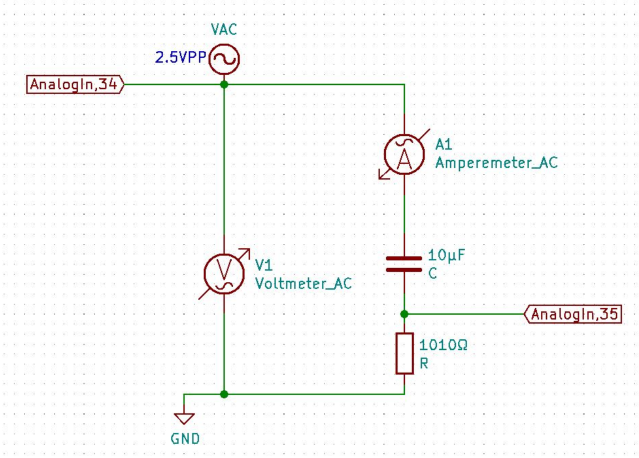

Figure5. EmonLib Verification circuits

In the circuit, a 2.5V peak-to-peak ACV to the RC circuit with their values as labeled respectively. An amp-meter and a volt-meter are involved to measure the real voltage and current seen by the components in the diagram.

We assumed the current in the primary circuit is 50A and the Voltage supply is 2.5V (means a 1:1 voltage transform) and so ideally, according to the rule to calculate power, the real power in the circuit should be around 120.62 W and with power factor of 0.95. Those data should match our measurement output on the ESP32.

4.3. ESP Flash Memory program/erase cycle.

According to this post [11], official esp32 is expected to have a minimum of 100,000 program/erase cycles.

4MB*100,000/(365*24*60*2*1B) = 12682 years approximately, if we write 2B of data per minute.

5. Ethics and Safety

5.1. Ethics

Based on the 10 ethics mentioned on the ieee code of ethics[3], we have a thorough consideration of our project and believe that our project will not violate these items.

1. **to hold paramount the safety, health, and welfare of the public, to strive to comply with ethical design and sustainable development practices, and to disclose promptly factors that might endanger the public or the environment;**

   - Our project will be enclosed in a plastic case in the wall and no smell nor harmful stuff will leak to the public.

2. **to avoid real or perceived conflicts of interest whenever possible, and to disclose them to affected parties when they do exist;**

   - Our project will not cause any conflicts to any forms of interests: it's simply a small device that helps monitoring the power consumption at our own place.

3. **to be honest and realistic in stating claims or estimates based on available data;**

   - Our project serves primarily to report the measurement from our local devices. Thus, even though tiny measurement errors may occur, there would be no dishonesty involved.

4. **to reject bribery in all its forms;**

   - Our project is automated from front-end to back-end and so since no man-force is involved in this measurement, no bribery is possible.

5. **to improve the understanding by individuals and society of the capabilities and societal implications of conventional and emerging technologies, including intelligent systems;**

   - Our project will eventually work as a public presentation in the building for any public curiosities of any form.

6. **to maintain and improve our technical competence and to undertake technological tasks for others only if qualified by training or experience, or after full disclosure of pertinent limitations;**

   - Our project will be implemented in the lab first and then to carry out as real designs to measure the power in walls, so it should be a fully-tested project.

7. **to seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors, and to credit properly the contributions of others;**
   - We will do credit for any work we utilized and are welcome for any advice.
8. **to treat fairly all persons and to not engage in acts of discrimination based on race, religion, gender, disability, age, national origin, sexual orientation, gender identity, or gender expression;**
   - Our project has nothing to do with discrimination.
9. **to avoid injuring others, their property, reputation, or employment by false or malicious action;**
   - Our project is automated from front-end to back-end. Therefore, there is no chance for people to get injured on our project.
10. **to assist colleagues and co-workers in their professional development and to support them in following this code of ethics.**
    - Our project is built on our own thoughts and the previous works of many others and hopefully our design could be useful for any followers in this field.

5.2. Safety

As far as we know, there does not exist too many safety concerns in our design of the project. The only thing that might be dangerous is that since we are measuring the AC power flowing into each room in our building, the voltage and current might be very high and difficult to handle.

However, taking the suggestions from the professor, we believe that we can test our design using the simulation tools in the laboratory scope. If everything works as expected, we will then put them into real measurement under the supervision of the department assistance.

6. Citations

[1]:InnovatorsGuru, “IOT Based Energy Meter Reading Using Arduino: PPT: Code: Abstract: Block Diagram,” InnovatorsGuru, 29-Jan-2020. [Online]. Available: https://www.innovatorsguru.com/iot-based-energy-meter-reading-using-arduino/. [Accessed: 28-Feb-2020].

[2]“CTF-5SFT-0200,” AutomationDirect. [Online]. Available: https://www.automationdirect.com/adc/shopping/catalog/sensors_-z-_encoders/current_-a-_voltage_sensors_(ac_-a-_dc)/current_transformers/solid_core_ac_current_transformers/ctf-5sft-0200. [Accessed: 28-Feb-2020].

[3]:"openenergymonitor/EmonLib", *GitHub*, 2020. [Online]. Available: https://github.com/openenergymonitor/EmonLib. [Accessed: 13- Feb- 2020]

[4]“IEEE Code of Ethics,” *IEEE*. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: 14-Feb-2020].

[5]R. Wall, “Sampling rate of emonLib,” Community, 22-Jun-2017. [Online]. Available: https://community.openenergymonitor.org/t/sampling-rate-of-emonlib/4383. [Accessed: 28-Feb-2020].

[6]P. Waher, P. Seneviratne, B. Russell, and D. V. Duren, “IoT,” Amazon, 2016. [Online]. Available: https://docs.aws.amazon.com/iot/latest/developerguide/mqtt.html. [Accessed: 28-Feb-2020].

[7]J. Bays, “Cambridge walk about: pictorial map and guide,” *Amazon*, 1984. [Online]. Available: https://aws.amazon.com/about-aws/. [Accessed: 28-Feb-2020].

[8]R. W. Hendrix, “Lambda,” Amazon, 1983. [Online]. Available: https://aws.amazon.com/lambda/. [Accessed: 28-Feb-2020].

[9]D. Rangel, “DynamoDB: everything you need to know about Amazon Web Service's NoSQL database,” *Amazon*, 2015. [Online]. Available: https://aws.amazon.com/dynamodb/. [Accessed: 28-Feb-2020].

[10]“S3,” *Amazon*, 2002. [Online]. Available: https://aws.amazon.com/s3/. [Accessed: 28-Feb-2020].

[11] *How many times can you flash? - ESP32 Forum*. [Online]. Available: https://www.esp32.com/viewtopic.php?t=709. [Accessed: 28-Feb-2020].