

**ACCURACY COMPARISON AND AUTO-CALIBRATION
ALGORITHM OF PRESENT LOW-COST CURRENT SENSORS
FOR BUILDING ENERGY MONITORING**

BY

RUENGWIT KHWANRIT

**A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF ENGINEERING (INFORMATION AND
COMMUNICATION TECHNOLOGY FOR EMBEDDED SYSTEMS)
SIRINDHORN INTERNATIONAL INSTITUTE OF TECHNOLOGY
THAMMASAT UNIVERSITY
ACADEMIC YEAR 2017**

**ACCURACY COMPARISON AND AUTO-CALIBRATION
ALGORITHM OF PRESENT LOW-COST CURRENT SENSORS
FOR BUILDING ENERGY MONITORING**

BY

RUENGWIT KHWANRIT

**A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF ENGINEERING (INFORMATION AND COMMUNICATION
TECHNOLOGY FOR EMBEDDED SYSTEMS)
SIRINDHORN INTERNATIONAL INSTITUTE OF TECHNOLOGY
THAMMASAT UNIVERSITY
ACADEMIC YEAR 2017**

ACCURACY COMPARISON AND AUTO-CALIBRATION ALGORITHM OF
PRESENT LOW-COST CURRENT SENSORS FOR BUILDING ENERGY
MONITORING

A Thesis Presented

By

RUENGWIT KHWANRIT

Submitted to

Sirindhorn International Institute of Technology

Thammasat University

In partial fulfillment of the requirement for the degree of
MASTER OF ENGINEERING (INFORMATION AND COMMUNICATION
TECHNOLOGY FOR EMBEDDED SYSTEMS)

Approved as to style and content by

Advisor



(Asst. Prof. Dr. Somsak Kittipiyakul)

Committee Member



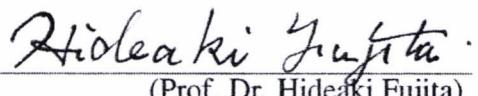
(Asst. Prof. Dr. Teerayut Horanont)

Committee Member and
Chairperson of Examination Committee



(Dr. Jasada Kudtongngam)

Committee Member



(Prof. Dr. Hideaki Fujita)

JULY 2018

Acknowledgments

This thesis would not be able to complete without many of assistance. So I would like to use this opportunity to thank to all these following persons and organizations.

First of all, I would like to thank to my advisor Asst. Prof. Dr. Somsak Kittipiyakul who accepted me as his master student and gave me the interesting research. He guided me and gave me a lot of suggestion in my work. He also reserved his own budget when we bought new equipments and components. He is such an idol for me in many way such as the way he control the emotion and situation. He always think positively. I can say that he is the best advisor that I ever had.

Beside, I would like to thank to members of thesis committees: Dr. Jasada Kudtongngam, Asst. Prof. Dr. Teerayut Horanont, and Prof. Dr. Hideaki Fujita for their valuable advice and suggestion in my work. Especially, Dr. Jasada Kudtongngam who responded to my request every time such as when I asked for some advice or a recommendation or even he was sick, he still did not ignore to answer back. Also, Prof. Dr. Hideaki Fujita who quickly responded and did not hesitate to gave me a recommendation.

I would like to thank the financial support from Thailand Advanced Institute of Science and Technology (TAIST), National Science and Technology Development Agency (NSTDA), Tokyo Institute of Technology, and Sirindhorn International Institute of Technology (SIIT), Thammasat University.

I would like to thank Mr. Manit Paisansopon (P'Manit), the SIIT senior technician, who supported me many equipments in this work. I also would like to thank Ms. Naratsita (P'Pla), Ms. Nichcha (P'Ann) and Ms. Charinnarat (P'Meen) the secretaries of ICT department at SIIT who continuously provided me pass through each step in this master degree. Especially P'Pla, she gave me a lot of nicely services. Moreover, I would like to thank all of the staffs at Tokyo-Tech office (Thailand) who provided kindly services. Especially, P'Palm and P'Cho who look after TAIST students closely.

Additionally, I would like to thank to the special person, Rujiret Seweepong, for any support and encouragement. Last but not least, I would like to thank my parents: Supap Khwanrit and Wasinee Khwanrit for unconditional financial and mental support.

Abstract

ACCURACY COMPARISON AND AUTO-CALIBRATION ALGORITHM OF PRESENT LOW-COST CURRENT SENSORS FOR BUILDING ENERGY MONITORING

by

Ruengwit Khwanrit

Bachelor of Engineering in Electrical Engineering, King Mongkut's Institute of Technology Ladkrabang, 2016

Master of Engineering (Information and Communication Technology for Embedded Systems), Sirindhorn International Institute of Technology Thammasat University, 2018

Reducing energy consumption is an important motivation for building energy management systems. In this work, we intend to build a low-cost building energy management system (BEMS) for Sirindhorn International Institute of Technology (SIIT) at Bangkadi Campus, to monitor electricity consumption of appliances, especially air conditioners which are the largest energy consuming devices. To make the system low-cost, we need to find low-cost but acceptable sensor and other components. Each wireless energy monitoring node uses the ESP8266/ESP32 microcontroller that has a 2.4GHz built-in WiFi. For the low-cost sensor, we chose to compare four different current sensors. For three of the four sensors, we need to build circuits and calibrate the sensors to accurately convert the measured voltage by the microcontroller to RMS current. We tested the accuracy of the sensors with a low-cost adjustable load system consisted of light bulbs. The load can draw current in steps of about 0.4 A and in total of 20 A maximum. We calibrate the sensors in two different methods. In the first method, we collected the calibration data and then manually fitted the samples with piecewise linear lines by ourselves. In the second method, we tried to make the system more automatic to reduce the time to calibrate each sensor. This was

done by having the curve fitting performed by the microcontroller. The fitting must satisfy the requirement on the maximum fitting error. We tested the calibration with another set of data and showed that the second method is better than the first method.

Keywords: low-cost current sensor, wifi-microcontroller, building energy management system, sensor calibration, measurement accuracy.

Table of Contents

Chapter	Title	Page
	Signature Page	i
	Acknowledgments	ii
	Abstract	iii
	Table of Contents	v
	List of Figures	viii
	List of Tables	x
1	Introduction	1
	1.1 Statement of Problem	1
	1.2 Objective of Study	2
	1.3 Significant of Study	2
	1.4 Thesis Organization	2
2	Literature Review	4
	2.1 Low-cost Measurement Component Implementation	4
	2.2 Low-cost Meter Accuracy Comparison	6
	2.3 Sensor Calibration and Linearization	6
3	Components in this work	7
	3.1 Wireless Measurement Node's Components	7
	3.1.1 Wifi-Microcontroller	7
	3.1.1.1 ESP8266	7
	3.1.1.2 ESP32	8
	3.1.2 Current Sensor	10
	3.1.2.1 ACS712	10
	3.1.2.2 WCS1800	10

3.1.2.3 SCT013	10
3.1.2.4 PZEM004T	11
3.2 Calibration and Testing Components	13
3.2.1 Reference Meter	13
3.2.2 AC Loads	13
3.2.3 Keypad	15
4 Current Calculation	16
4.1 Current Sensor's Output	16
4.1.1 ACS712 and WCS1800	16
4.1.2 SCT013	17
4.1.3 PZEM004T	17
4.2 RMS Current Calculation	18
5 Gain Calibration and Accuracy Testing Method	21
5.1 Calibration Method	21
5.1.1 Manual Calibration	21
5.1.2 Auto-Calibration	21
5.1.2.1 First step: Mode Selection	22
5.1.2.2 Second step: Inserting No. of step	23
5.1.2.3 Third step: Inserting I_{rms} and reading V_{cs_rms} value	23
5.1.2.4 Fourth step: Sorting each step's value	24
5.1.2.5 Fifth step: Applying Ordinary Least Square (OLS)	24
5.1.2.6 Sixth step: Recording constant values to non-volatile storage	28
5.2 Accuracy Testing Method	29
6 Results and Discussion	30
6.1 Gain Calibration Results	30
6.1.1 Gain Calibration Results: Manual Calibration	30
6.1.2 Gain Calibration Results: Auto-Calibration	31

6.2 Accuracy Results	38
6.2.1 Accuracy Results: Manual Calibration	38
6.2.2 Accuracy Results: Auto-Calibration	41
6.2.3 Power Measurement Accuracy	47
7 Conclusion	48
7.1 Summary of calibration and testing	48
7.2 Suggestion for future works	49
References	51
Appendices	54
Appendix A: List of publication	55
Appendix B: ESP32's analog to digital conversion	56

List of Figures

Figures	Page
3.1 Wemos D1 mini (ESP8266 chip)	7
3.2 Wemos LoLin (ESP32 chip)	8
3.3 ACS712	10
3.4 WCS1800	11
3.5 SCT013	11
3.6 PZEM004T	12
3.7 PX 120 Power meter	13
3.8 Eight 550W mercury vapor lamps	14
3.9 Six 100W incandescent lamps	14
3.10 Keypad	15
4.1 Additional circuit of ACS712 and WCS1800	17
4.2 Additional circuit of SCT013	18
5.1 Mode selection	22
5.2 Diagram of step 2 in auto-calibration operation	23
5.3 Code of bubble sorting	24
5.4 Flowchart of applying OLS step	27
5.5 Code of recording and reading values from non-nolatile memory	29
6.1 SCT013 manual calibration ($R^2 = 1$)	30
6.2 ACS712 manual calibration ($R^2 = 0.9997$)	31
6.3 WCS1800 manual calibration ($R^2 = 0.9998, 0.9996, 0.9972, 0.9958$)	31
6.4 SCT013's gains when applied 20 points auto-calibration algorithm	33
6.5 ACS712's gains when applied 20 points auto-calibration algorithm	34
6.6 WCS1800's gains when applied 20 points auto-calibration algorithm	35
6.7 SCT013's gains when applied 50 points auto-calibration algorithm	36
6.8 ACS712's gains when applied 50 points auto-calibration algorithm	37
6.9 WCS1800's gains when applied 50 points auto-calibration algorithm	38
6.10 Percentage error results of four different current sensors	39

6.11	Absolute error results of four different current sensors	39
6.12	SCT013's percentage error result from accuracy test	43
6.13	SCT013's absolute error result from accuracy test	43
6.14	ACS712's percentage error result from accuracy test	44
6.15	ACS712's absolute error result from accuracy test	44
6.16	WCS1800's percentage error result from accuracy test	45
6.17	WCS1800's absolute error result from accuracy test	45
B.1	Relationship between Digital 12-bit and Real voltage (V) of No.1 ESP32's ADC port	56
B.2	Relationship between Digital 12-bit and Real voltage (V) of No.2 ESP32's ADC port	57
B.3	Relationship between Digital 12-bit and Real voltage (V) of No.3 ESP32's ADC port	57

List of Tables

Tables	Page
3.1 ESP8266 and ESP32 details comparison [16]	9
3.2 Current sensors properties	12
6.1 Number of gain of current sensors when applied auto-calibration	31
6.2 Gain, Offset, and Boundary of SCT013 in 20 points calibration	33
6.3 Gain, Offset, and Boundary of ACS712 in 20 points calibration	34
6.4 Gain, Offset, and Boundary of WCS1800 in 20 points calibration	35
6.5 Gain, Offset, and Boundary of SCT013 in 50 points calibration	36
6.6 Gain, Offset, and Boundary of ACS712 in 50 points calibration	37
6.7 Gain, Offset, and Boundary of WCS1800 in 50 points calibration	38
6.8 Percentage error of fitting error	42
6.9 Percentage error of testing error	42
6.10 Percentage error of power measurement	47

Chapter 1

Introduction

1.1 Statement of Problem

Reducing energy consumption in building is an important motivation of building energy management system. Similarly, SIIT building has high power consumption in last many year. Nowadays, SIIT building bangkadi campus has the computer system that allow people to turn on/off air conditions in only during class time because they are the main power consumption of the building but this system can not reduce energy much as expect. Now, SIIT focuses on this problem and want to manage these energy consumption issue. Because of that, there is an intention to install the energy monitoring and management system. But if SIIT buy the system from commercial company it will cost high budget. Therefore, the building energy management system research is occurred.

For the first research of this project aims to build the low cost hardware to monitoring energy consumption of air conditions but still has enough performance that can operate correctly, continuously and can send data to the server via SIIT wifi. The main component in measurement node is current sensor. So in this work we focus on low-cost AC current measurement sensor in form of module that available in nowadays market not just a measurement IC chip because the convenience of the installation. In this research we aim to do a comparison several popular low-cost AC current sensors to find the best appropriate current sensor in creating the electrical measurement prototype for implementing in the first phase of building energy monitoring system in SIIT building.

Moreover, after we compared those low-cost current sensors, this research also aim to invent the auto-calibration algorithm for analog output current sensor to improve accuracy and provide more conveniently in calibration.

1.2 Objective of Study

The main objectives of the research are as follows:

- To find the proper current sensor by comparing measurement accuracy, price, and installation difficulty of four popular low-cost current sensor for creating measurement node of SIIT building energy monitoring system.
- To improve measurement accuracy and convenience in calibration of analog current sensor by inventing auto-calibration algorithm.

1.3 Significant of Study

This research try to compare the the measurement accuracy of several present low-cost current sensors. The other advantage beside implementing the electrical measurement node for SIIT building can be in many ways as following. Because of none of paper researched about comparing present low-cost current sensors. So this research can be very useful with other researchers who aim to use these low-cost component in their research. It does not need to be the building energy monitoring research but it can be any kind of research that involve with these components. So they can have information such as which low-cost component are available in the market nowadays, their price, accuracy, and difficulty in installation. Then, they can easily select the appropriate component for their research. Therefore, they can reduce time to learn about all of them. They just learn some of sensor which they are interested. Moreover, this research also invented a simple auto-calibration algorithm which help to calibrate analog output current senor to be more accurate and save much more time than using manual calibration method. This method just use a simple equation and also can be implemented with low-cost microcontroller. So anyone can utilize this algorithm to their own work.

1.4 Thesis Organization

The rest of this thesis is organized as follow. In chapter 2, we provide some of literature review about the past low-cost component implementation and their result,

accuracy comparison of low-cost meters, and some of calibration work. Chapter 3 describes about components that we use in this comparison and calibration. Chapter 4 shows current calculation equation of each current sensor and how microcontroller calculate RMS current. After that in chapter 5, we describe about accuracy testing method and auto-calibration algorithm and method that we used. The result and discussion are given in chapter 6. In this chapter we show the calibration and measurement accuracy result from the test and also discuss about them. Finally, Chapter 7 provides the conclusion of this thesis and gives some recommendation for future work.

Chapter 2

Literature Review

This chapter provides the literature review that related to this thesis. It includes sections on low-cost measurement component implementation, accuracy comparison, and sensor calibration.

2.1 Low-cost Measurement Component Implementation

Several researches built and implemented low-cost meter for measuring and monitor power consumption status by using low-cost components such as microcontroller and current sensor. Some papers [1, 2, 3, 4, 5, 6, 7] used Arduino board with Atmega microcontroller to process data. Many papers [2, 3, 7, 8] used ESP8266 module to be a wi-fi data transmitter or even microcontroller.

In papers [2, 9], the authors used ACS712 low-cost current sensor, And the others [3, 4, 5] used SCT013 low-cost split core current sensor.

Paper [1], The authors created prototype for monitoring energy called ELIVE device for sensing real-time energy consumption data in a smart home using wi-fi communication. The ELIVE device consists of Arduino microcontroller interface with ESP8266 wi-fi chip and nameless current transducers. The result was compared to BILLION sensor device from Taiwan company and it exposes that ELIVE performs better than BILLION based on the summary statistics.

In [2], This work implemented very low-cost wi-fi based digital energy meter by using ESP8266 wi-fi module to communicate between digital meter and web server for monitoring energy consumption and load profile. This system uses Arduino Leonardo Pro to calculate meter parameters and ACS712 current sensor device.

For [3] presented the design and implementation of portable energy meter for any appliances in a residence by using non-invasive current sensor, SCT013. This work used Arduino nano connect with ESP8266 wi-fi module to transmit real-time acquired data wirelessly to a server then display on a screen and a home energy monitor website. The testing was done by using a variable resistive load and a power analyzer. Average

accuracy was found out to be 92.6 %.

In research [4, 9] made a device that can measure real-time electrical quantities then collect the data in a database system and show the statistics reports on web interface by using BCM2835 (Raspberry Pi) embedded single board to calculation parameter. For [4] also used Arduino in data acquisition and used SCT013 current sensor to sensing current. Differently, in the paper [9] used ACS712 to measure current.

Work [5] proposed a smart plug which can do a real-time energy monitoring and controlled by using an android mobile. This smart plug uses Arduino duemilanove with SCT013 current sensor and ethernet module to upload data to the server. The result shown that this energy consumption monitoring made the consumers to be aware of their wastage electricity and it helps to reduce energy consumption about 15 %.

Similarly in work [8], a low-cost wi-fi smart plug is proposed for monitoring power consumption of electrical appliances. The authors also created a web application which can run on smart phone, iPad, and laptop. It used to monitor electrical status and control turning on/off appliances by using relay employed in the single phase electrical system. The smart plug uses ESP-WROOM02 (ESP8266 chip) to be the microcontroller and use STPM01 chip to measure energy consumption. The error accuracy is less than 0.5 %.

Work [6] also used SCT013 sensor to design a power meter for measuring single phase power system. The author use two channel ADCs to sampling the voltage and current separately for improving real power accuracy. Moreover, in research [7] implemented PZEM004T sensor, the new electrical measurement sensor, in their work to measure and monitor electrical consumption of a three-phase four power line in a laboratory building and integrated with IoT concept.

ESP32, a new low-cost wi-fi microcontroller also very interesting to use in low-cost work. In work [10] proposed a monitoring system of a small photovoltaic power system which designed based on low-cost sensor to measure the current and voltage of battery then this information will be stored in SD card and sent to server via wi-fi.

2.2 Low-cost Meter Accuracy Comparison

An accuracy comparison of low-cost energy meters is presented in [11]. This work compared accuracy result of 4 nameless different commercially available low-cost energy meters in 2 tests which are power and energy measurement test. In the first test, the result shown that low-cost energy meter can perform accordingly to proclaimed accuracy. But for the second test indicated that a good power measurement does not give good energy consumption measurement.

2.3 Sensor Calibration and Linearization

In work [12] and [13] presented calibration method for voltage and hall sensor. Paper [12] applied 1st to 5th polynomial regression to determine the best fitting relationship between input and output of an voltage sensor. They used 2 approaches to calculate voltage which are instantaneous calculation which is the error in measurement is $< 2.5\%$ and peak-peak voltage method that give the error $< 1\%$. And the best result came from third order polynomial regression. Work [13] proposed a novel auto-calibration method which used on-chip actuator to generate 3 reference signals to calculate second order polynomial fitting for sensor's characteristic correction to ensure reliability and accuracy.

Article [14] treated the problem of designing an optimal size for a lookup table (LUT) that applied for sensor linearization. This research proposed a theory for finding the minimum size of a lookup table which does not impact the accuracy of the sensor. Because in any small embedded systems, they just provide a small amount of memory so the lookup table (LUT) must be reduced to a minimum size.

In work [15] applied a piecewise linear interpolation technique in Field Programmable Gate Array (FPGA) to linearize the a nonlinear characteristic of thermistor for real time application. The temperature in 0-60 C was linearized using 10, 20, and 30 sets of data points. However, the more data points can give the higher accuracy of linearization.

Chapter 3

Components in this work

3.1 Wireless Measurement Node's Components

Wireless measurement node consists of 2 things which are wifi-microcontroller and current sensor. WiFi-microcontroller was used to calculate voltage level from different low-cost current sensors. These mentioned components will be described in this chapter.

3.1.1 Wifi-Microcontroller

Nowadays, there are some of low-cost wifi-microcontrollers which already have wifi chip that can process and send data wirelessly. Also, they are very easy to use. So we intend to use this kind of microcontroller in our work to calculate value from current sensor. There are 2 popular wifi-microcontrollers in today's market which are ESP8266 and ESP32 (ESP8266 descendant).

3.1.1.1 ESP8266

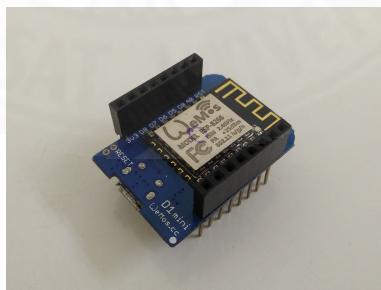


Figure 3.1 Wemos D1 mini (ESP8266 chip)

The vendor of ESP8266 is Espressif (China). Inside this chip contains Tensilica L106 single-core 32-bit microcontroller which runs at 80 MHz (up to 160 MHz) and it has a large amount of flash memory which is 4MB. This chip has 1 ADC port (10-bit). It has deep sleep mode for consuming low power and also has 2 SPI and 1 I2C port. We can use C Arduino language and Arduino IDE to develop it. So we can easily use

this chip same as Arduino. Anyway, this chip has many benefits compared to Arduino board which are the higher cpu speed and flash memory but the limitation is it has only 1 ADC port (Arduino has 6 ports). ESP8266 has several development boards such as Wemos D1 mini, NodeMCU, Witty cloud, ThaiEasyElec's ESPino, ESPDuino etc. The price of development board, for example Wemos D1 mini cost only 185 baht (6 \$). Fig. 3.1 show the Wemos D1 mini board that use ESP8266 wifi-microcontroller chip.

3.1.1.2 ESP32

ESP32 is a new low-cost wifi-microcontroller which is the ESP8266 descendant. It also was created by Espressif (China). Inside this chip contains Xtensa LX6 dual-core 32-bit microcontroller which runs at 160 MHz (up to 240 MHz). It has higher speed than ESP8266 and also has 4MB of flash memory. This chip has 18 ADC (12-bit) ports. It still has deep sleep mode and also has 4 SPI, 2 I2C. We can use C Arduino language and Arduino IDE to develop it. ESP32 has several development boards for example NodeMCU-32S, Wemos LoLin32, DOIT ESP32, Sparkfun ESP32, Adafruit ESP32, and ThaiEasyElec's ESPino32 etc. The price for development board such as Wemos LoLin cost only 330 baht (10.5 \$). Fig. 3.2 show the Wemos LoLin board which use ESP32 (WROOM-32) wifi-microcontroller.



Figure 3.2 Wemos LoLin (ESP32 chip)

From the properties of both low-cost wifi-microcontrollers that mentioned before. We chose ESP32 to use in this work because of we need multi-ADC ports but ESP8266 has only 1 ADC port which means if we want to use ESP8266. We need to use the help from a multiplexer which consumes much power from switching operation. Another benefit of ESP32 is it has more ADC resolution and more microcontroller speed. And

the price is acceptable. Hence, we selected Wemose LoLin32 development board in this work.

Table 3.1 shows the comparison details for each property of ESP8266 and ESP32 low-cost wifi-microcontroller.

Table 3.1 ESP8266 and ESP32 details comparison [16]

Chip (Module)	ESP8266 (ESP8266-12E)	ESP32 (ESP-WROOM-32)
Picture		
Information :		
CPU	Tensilica L106 32-bit at 80 MHz (up to 160 MHz)	Xtensa Dual-Core 32-bit LX6 at 160/240 MHz
Vendor	Espressif (China)	Espressif (China)
SRAM	36 KB available	520 KB
Flash memory	4 MB (up to 16 MB)	4 MB (up to 64 MB)
Operating voltage	2.5 – 3.6 V	2.3 – 3.6 V
Operating current	80 mA (average)	80 mA (average)
Deep sleep current	20 uA	10 uA
Language	C, C++, Lua, Python	C, C++, Lua, Python
Operating temperature	-40 – +125 C	-40 – +125 C
Wi-Fi	802.11 b/g/n	802.11 b/g/n
Bluetooth	-	4.2 BR/EDR + BLE
Ports :		
UART	2	3
GPIO	17	32
SPI	2	4
I2C	1	2
PWM	-	8
ADC	1 (10-bit)	18 (12-bit)
DAC	-	2 (8-bit)
Starting Price :		
Chip	120 THB (4 \$)	185 THB (6 \$)
Development board	185 THB (6 \$)	330 THB (10.5 \$)

3.1.2 Current Sensor

Current sensor is a device which can measure amount of flowing current and send the output in form of voltage level to microcontroller to calculate the measured current value back. In this work, we selected four popular low-cost current sensors to compare their accuracy. The details of those current sensors will be described in this section.

3.1.2.1 ACS712

ACS712 is a hall effect current sensor. In market, it has 3 rated current which are 5, 20, and 30. This work will use the highest rated that is 30 A rated current. ACS712 is the ultra low-cost current sensor. Its price of breakout board is about 85-100 baht (3 \$). This current sensor has some disadvantage that is we need to break the wire that we want to measure current and connect this sensor in series. Fig. 3.3 shows ACS712 current sensor.

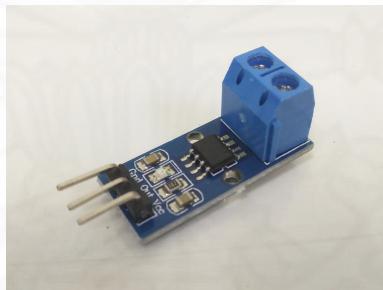


Figure 3.3 ACS712

3.1.2.2 WCS1800

WCS1800 is a hall effect current sensor which has 35 A rated current. Its breakout board price is about 300 - 400 baht (10-12 \$). We can put the wire that we want to measure through the current sensor's hole so we do not need to break the circuit and connect the sensor in series like ACS712. Fig. 3.4 shows WCS1800 current sensor.

3.1.2.3 SCT013

SCT013 is a split core clamp on current sensor. We can just simply clamp this sensor on the measured wire. This kind of current sensor use induced voltage principle.



Figure 3.4 WCS1800

The current sensor will act similar to a secondary side of a transformer. Because of that, this kind of sensor can only measure ac current. This current sensor has many rated which are 5, 30, 50, and 100 A_{rms} so we chose the 30 A_{rms} rated current in this work. Its price is about 250 - 400 baht (8-12 \$) depend on the store. Fig. 3.5 shows 30 A SCT013 current sensor.



Figure 3.5 SCT013

3.1.2.4 PZEM004T

PZEM004T is a new low-cost power sensor from China it can measure current, voltage, and power. The current can be measured by putting the wire though the sensor iron core's hole. This sensor can measure current up to 100 A_{rms}. And this price is cheap, about 280 baht (9 \$). Fig. 3.6 shows PZEM004T sensor.



Figure 3.6 PZEM004T

Table 3.2 Current sensors properties

Name / Picture	Rated current	DC/AC	Output signal	Installation	Price
ACS712 	30 A _{DC} / 21 A _{AC}	DC / AC	Analog Voltage level	Connect sensor in series with wire	85-100 baht (3 \$)
WCS1800 	35 A _{DC} / 25 A _{AC}	DC / AC	Analog Voltage level	Put wire through sensors hole	300-400 baht (10-12.5 \$)
SCT013 	30 A _{AC}	AC	Analog Voltage level	Split core clamp on wire	250-400 baht (8-12.5 \$)
PZEM004T 	100 A _{AC}	AC	UART Serial communication	Put wire through iron cores hole	280 baht (9 \$)

Table 3.2 shows the brief details of above mentioned current sensors which are

about maximum rated current that can be measured by the sensor, what kind of current that the sensor can measure, the form of output signal of the sensor, how to install the sensor, and their prices.

3.2 Calibration and Testing Components

In this section presents additional components which help us calibrate the sensors and test the current measurement accuracy of the sensors.

3.2.1 Reference Meter

First component is reference meter. The current level which will be measured in the calibration operation and in the test will be varied in 0 - 20 A_{rms} range. We use 2 of PX 120 Power meters to show reference current. The PX 120 power meter can measure and result in 2 resolution current but it can only measure current up to 10 A so we separated the electrical circuit into 2 wires and put a PX 120 meter at each wire. so we can have input current up to 20 A_{rms} and we put the current sensor to measure the input current before the circuit was separated in to 2 line. Fig. 3.7 shows PX 120 power meter.



Figure 3.7 PX 120 Power meter

3.2.2 AC Loads

Because the intended application of our current sensors is to measure current consumption by air conditioners, whose biggest size is 36,000 BTU split type which



Figure 3.8 Eight 550W mercury vapor lamps



Figure 3.9 Six 100W incandescent lamps

consumes 18 Amperes maximum, we require our sensor to support current measurement from zero to 20 A.

During calibration of the sensors, we need to generate current draw at multiple values. To do this we build an inexpensive variable dummy resistive load consisted of incandescent and street light bulbs, shown in Fig. 3.8 and Fig. 3.9 , as following:

- Six 100W incandescent bulbs ("small" bulbs)
- Eight 550W mercury vapor bulbs ("large" bulbs) used for street lighting

Each of the small and large bulbs consumes 0.435 A and a constant level between 2.1 – 2.5 A, respectively. These bulbs consume constant power after the initial warm-up period (less than 10 minutes). With these bulbs, we have 63 different combinations of the bulbs which means 63 different current steps. For example, if each large bulb

consumes 2.5A, then the current steps are 0, 0.435 A (1 small bulb), 0.87 A (2 small bulbs), 1.30 (3 small), 1.74 (4 small), 2.18 (5 small), 2.5 (1 large bulb), 2.61 (6 small), 2.94 (1 small + 1 large), and so on until 22.6 A (6 small + 8 large bulbs). But each of large bulb might consume current less than 2.5 A. So the final current is 19.6 A. Since the large bulbs cannot be immediately turned on after turned off, each calibration must start with all large bulbs on. In future, we can add in relays to control turning on/off these bulbs from the microcontroller. This will allow fully automatic data collection for calibration.

3.2.3 Keypad

We used a keypad to help us do the auto-calibration process by inserting number of point in 2 digits in and inserting measured I_{rms} in 4 digits at each value to microcontroller. For example, If loads consume current 7.42 A then we insert 0, 7, 4, and 2 respectively to the microcontroller by using this keypad.



Figure 3.10 Keypad

Chapter 4

Current Calculation

In this section, we describe about output's equation of each kind of the current sensor and how to calculate I_{rms} by using microcontroller.

4.1 Current Sensor's Output

4.1.1 ACS712 and WCS1800

Both ACS712 and WCS1800 are hall effect current sensors which give analog voltage level output that is a linear function of the instantaneous current as

$$v_{cs}(t) = \alpha(t) + \beta * i(t) \quad (4.1)$$

where

$v_{cs}(t)$ is the instantaneous sensor output voltage

$i(t)$ is the measured instantaneous current

β is called the gain

$\alpha(t)$ is the offset

The $\alpha(t)$ is used to make the sensor output voltage v_{cs} to be non-negative value since the sensor output voltage can be negative or positive. Hence, the typical value of the offset is 2.5 V. The offset is added internally by the sensor. However, the offset may vary with time due to the variation in the power supply. Since the output voltage of sensors ranges between 0.8 to 4.2 V but ESP32 only support input between 0 to 3.3 V, we need to use a voltage divider to scale down the sensor output voltage, by a factor of 2/3, to be in the proper range of 0.54 - 2.8 V for the MCU. For convenience, we change out notation a bit where we redefine $v_{cs}(t)$ in (4.1) as the instantaneous sensor voltage at the output of the voltage divider. This makes the typical value of the offset $\alpha(t)$ to be $2.5(2/3) = 1.67$ V.

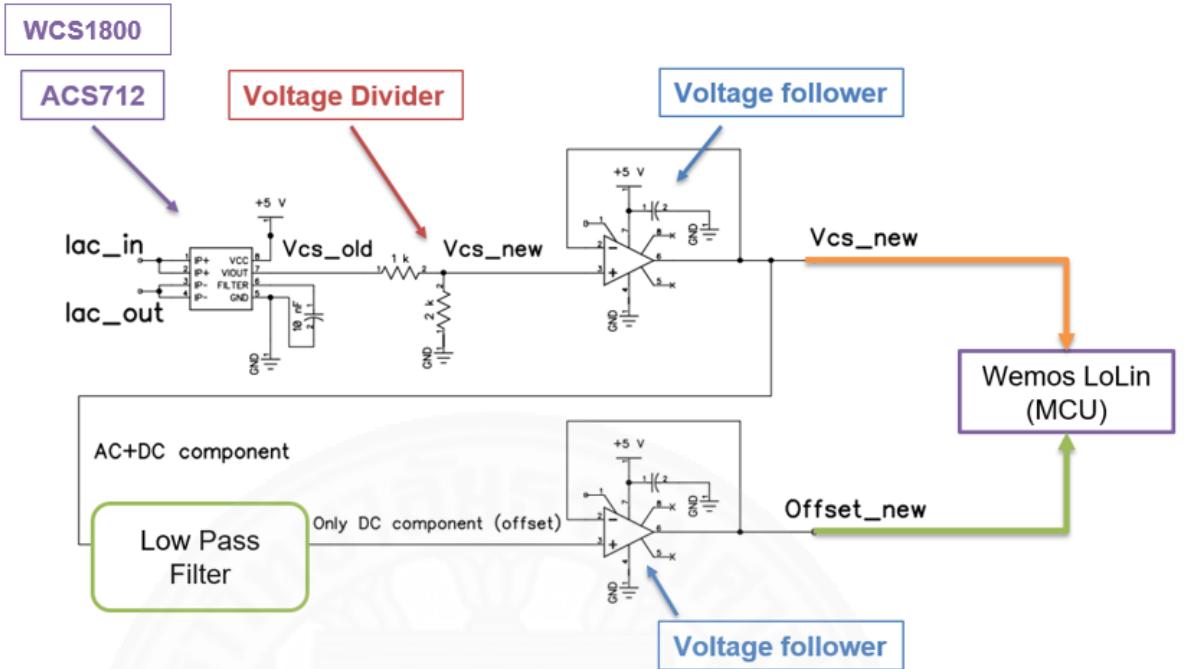


Figure 4.1 Additional circuit of ACS712 and WCS1800

4.1.2 SCT013

Similar to the previous two sensors, the sensor output voltage of SCT013 depends linearly on the instantaneous current as in (4.1). However, the offset is not added internally by the sensor. We must add the offset externally. Hence, we chose the offset to be 1.65 V. It turns out that the variation of the sensor output voltage is only between 0.72 and 2.58 V, already good for the MCU's inputs. Hence, we do not need a voltage divider as for the previous two sensors.

4.1.3 PZEM004T

The output from PZEM004T is UART serial communication in TTL level. It is not an analog voltage level as in the previous three sensors, so we do not need to create any additional circuit. We can get the measured current by using PZEM004T Library [17] to obtain the rms current I_{rms} value directly.

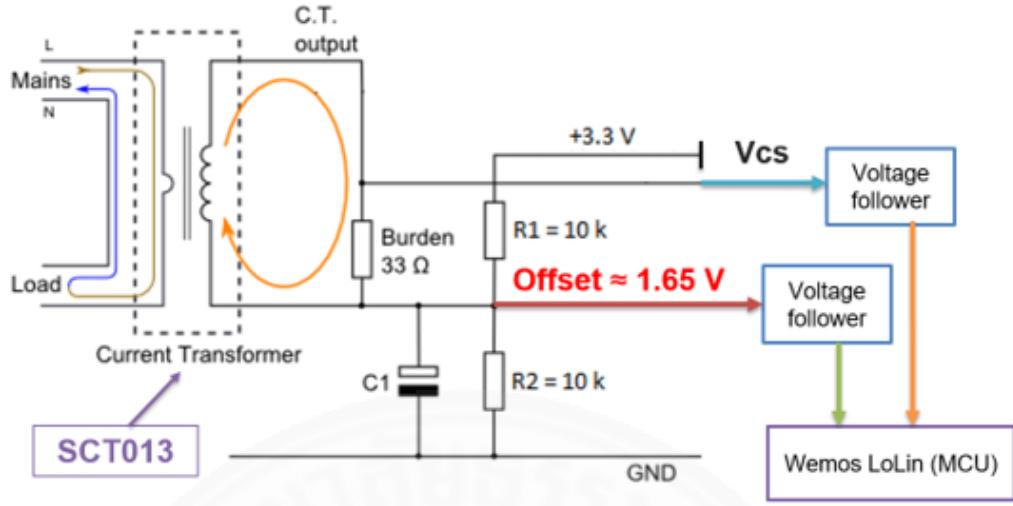


Figure 4.2 Additional circuit of SCT013

4.2 RMS Current Calculation

To determine the rms current I_{rms} of the instantaneous current $i(t)$, we need to find current sensor's non-offset output voltage ($v_{cs_no_offset}(t)$), which can be calculated in the MCU as

$$v_{cs_no_offset}(t) = v_{cs}(t) - \alpha(t) \quad (4.2)$$

Hence, to calculate the current sensor's non-offset instantaneous output voltage, the MCU must receive as inputs the instantaneous sensor voltage $v_{cs}(t)$ and the instantaneous offset $\alpha(t)$, which is determined as the DC component of v_{cs} . To get this DC component, we pass v_{cs} through a simple RC low pass filter, whose cut-off frequency is narrow enough to remove the 50 Hz AC component

For the three sensors, we need to determine the rms value from the non-offset instantaneous output voltage given in (4.2). By definition, the rms of voltage $v(t)$ is

$$V_{rms} = \sqrt{\frac{\int_0^T v^2(t) dt}{T}} \quad (4.3)$$

where T is the period of the ac voltage.

However, since we only have discrete-time samples of the output voltage

v_{cs,no_offset} , we approximate V_{rms} as

$$V_{rms} = \sqrt{\frac{\sum_{n=1}^N v_{cs,no_offset}^2(t_n)}{N}} \quad (4.4)$$

where we sample the voltage N times per second and $t_n = t_0 + n/N$ for some starting time t_0 . The summation in (4.4) could sufficiently represent the integral in (4.3) if N is large enough. In our work, we use $N = 5000$.

From (4.1), we could calculate the desired I_{rms} as

$$I_{rms} = \frac{V_{rms}}{\beta} \quad (4.5)$$

However, the linear current-voltage relation is ideal; in reality the relation is usually nonlinear (e.g., see [14]). In this work we choose to estimate the nonlinear relation between the rms current and voltage with piecewise linear interpolation from the samples collected during calibration because the estimation of the linear parameters can be easily performed by the MCU.

The piecewise linear interpolation estimates the true relation between the rms current and voltage as a piecewise linear function:

$$V_{rms} = \alpha' + \beta I_{rms} \quad (4.6)$$

for some α' and β that depend on the value of V_{rms} and are determined during the calibration process described next.

Now if the MCU sees an rms voltage V_{rms} , it estimates the rms current as

$$I_{rms} = \frac{V_{rms} - \alpha'}{\beta} \quad (4.7)$$

Note that α' is generally non-negative and β positive, due to the nonlinear relation between the rms current and voltage.

As we mentioned before, we use C Arduino language and Arduino IDE to program ESP32 microcontroller. We can obtain I_{rms} by using the following pseudocode.

Algorithm 1 RMS Current Measurement

procedure RMS CURRENT MEASUREMENT**Set** N to 5000**Set** Gain $\beta = 0.0402$, $\alpha' = 0$ ▷ for example**while** true **do** **Initialize** $sumsq$ to 0 **for** sample = 1 to N **do** **Get** $v_{cs,12bit}$, Offset α_{12bit}

$$v_{cs} = v_{cs,12bit} * 3.3 / 4096$$

$$\alpha = \alpha_{12bit} * 3.3 / 4096$$

$$v_{cs_no_offset} = v_{cs} - \alpha$$

$$sumsq = sumsq + v_{cs_no_offset}^2$$

$$V_{rms} = \sqrt{\frac{sumsq}{N}}$$

$$I_{rms} = \frac{V_{rms} - \alpha'}{\beta}$$

Chapter 5

Gain Calibration and Accuracy Testing Method

We mainly perform two sets of testing. The first set we calibrated each sensor for the gain β and non-offset intercept α' called gain calibration testing and the second set we performed accuracy testing of the current measurement for three units of the ACS712, WCS1800, SCT013, and PZEM004T sensors in manual calibration and for ACS712, WCS1800, and SCT013 in auto-calibration.

5.1 Calibration Method

5.1.1 Manual Calibration

In this *manual* calibration of the sensors. We manually collected the rms current and voltage samples and fitted a linear or multiple linear lines to find a simple explanation of the relationship between the current and voltage as shown in (4.7). Specially, we manually collected the current and voltage samples by adjusting the current loads to a sensor and using volt and current meters to measure the V_{rms} and I_{rms} samples. Then we manually attempted to fit the samples using a linear if possible, but when the fitting error is too high, we used multiple linear lines (a piecewise linear function) instead. Specifically as the result later, we used a single line for the rms current and voltage relation for the ACS712 and SCT013 sensors, but used four lines for the WCS1800 sensor.

For each of the three analog sensors: ACS712, WCS1800, SCT013, we run a calibration test on a randomly selected unit of each sensor. Recall that the PZEM004T sensor do not need calibration for the gain as it already gives the MCU the rms current values.

5.1.2 Auto-Calibration

In this part, we propose a *auto* calibration system that is an improvement of the manual calibration. This is done to improve the calibration accuracy and to allow the

data fitting be completely done by the microcontroller.

From previous chapter, we would like to know the I_{rms} value. So we need to know V_{cs_rms} , gain β , and non-zero intercept offset α' to proceed the I_{rms} calculation process.

In this auto-calibration method, we can take the advantage of equation (4.7) to create an auto-calibration algorithm. We use microcontroller to measure and calculate V_{cs_rms} value from current sensor output by itself. And we use a keypad that we have mentioned in chapter 3 to help inserting I_{rms} at each step.

Hence, we can find gain β , and non-zero intercept offset α' by applying auto-calibration algorithm. The steps of these algorithm are divided in to 6 steps that are described below.

5.1.2.1 First step: Mode Selection

First step in this auto calibration system, the microcontroller is programmed to allow a user to enter into the auto calibration mode. For easy use, we just assigned a pin to be the selecting mode pin which is if that pin has HIGH level (3.3 V), the microcontroller will enter into calibration mode. But if we do nothing just let that pin has LOW level (0 V), microcontroller will do the normal current measurement mode. Fig 5.1 show connection of pin 21 and 3.3 V.



Figure 5.1 Mode selection

5.1.2.2 Second step: Inserting No. of step

Second step, the algorithm will require the step value which is the No. of I_{rms} value which we are going to use in this calibration we can insert this value by using a keypad. For example you have 20 different current consumption steps so you insert 20. The step or point means I_{rms} and V_{cs_rms} value of the current sensor input and output relation graph. Fig 5.2 Diagram of step 2 in auto-calibration operation.

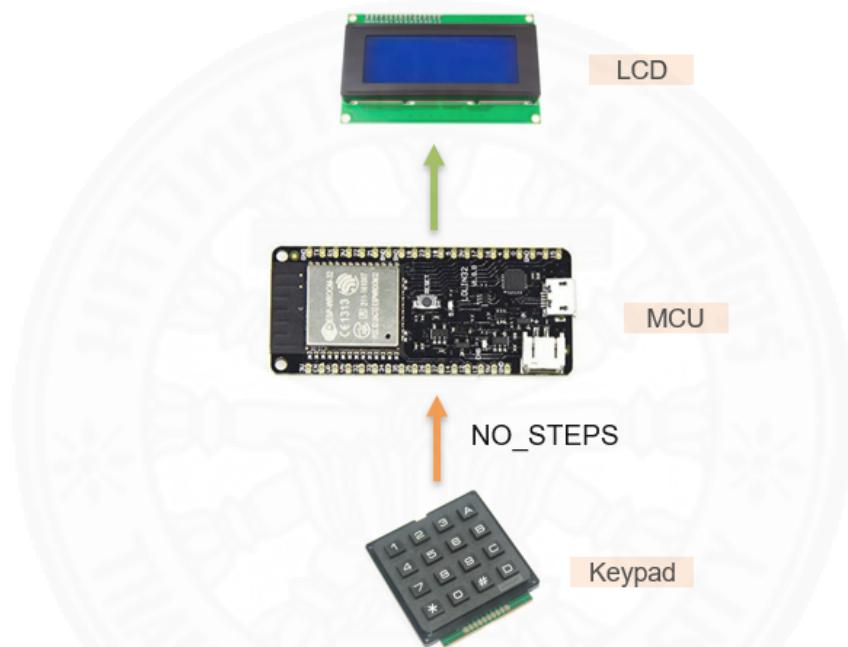


Figure 5.2 Diagram of step 2 in auto-calibration operation

5.1.2.3 Third step: Inserting I_{rms} and reading V_{cs_rms} value

Next, after we define No. of step or No. point in the relation graph. Now, the user can start manually turning on and off some bulb loads to vary the current measured by the sensor then we need to insert I_{rms} that loads consume by using a keypad when finished inserting each I_{rms} value, the microcontroller will automatically read V_{cs_rms} at that current level itself by using algorithm (1) procedure. When finished microcontroller will ask for another I_{rms} so we just turn on or off the loads to consume different current level. We need to do this until all N data points are collected. While we are inserting value, if we have any wrong inserting. We can delete that value and insert the new value

of that point again when it asks for confirmation.

5.1.2.4 Fourth step: Sorting each step's value

After we inserting and reading all step values which are I_{rms} and V_{cs_rms} . The microcontroller will apply bubble sort to sort the point based on I_{rms} value. V_{cs_rms} will be sorted along with its I_{rms} value. So this steps help the user that user do not need to turn on/off loads to consume current level respectively from low to high current level. User can turn on/off the loads any format then microcontroller will sort it automatically. I_{rms} after sorting will be in form of low to high value.

```
void bubble_sort(float Irms[], float Vcs_rms[], int n ){
    int i, j, t = 0 ;
    for( i = 0; i < n; i++){
        for( j = 1; j < n - i ; j++){
            if(Irms[j-1] > Irms[j]) {
                swap(&Irms[j-1], &Irms[j]) ;
                swap(&Vcs_rms[j-1], &Vcs_rms[j]) ;
            }
        }
    }

    void swap(float *a0, float *a1){
        float tmp = *a0 ;
        *a0 = *a1 ;
        *a1 = tmp ;
    }
}
```

Figure 5.3 Code of bubble sorting

5.1.2.5 Fifth step: Applying Ordinary Least Square (OLS)

After all collected I_{rms} were sorted, we start fitting the first line to the first group of the smallest currents via the ordinary least square (OLS) estimates of the line slope and offset. It tries to include as many samples as possible which are fitted by a single line so that the maximum fitting error among all the points is within the pre-specified error threshold. That is, the first three smallest points are fitted by a linear line. If the maximum fitting error of these points is within the error threshold, it adds in the fourth sample and re-estimate the linear line passing through these four points. If the maximum error is still acceptable, it then adds in the next sample and keep on extending the samples fitted by a single line. If after adding a new point to the group and the

parameter re-estimation results in an unacceptable maximum error, the just-added point is removed out of the group and together with the previous point are used to create a new group for the next line. the gain and offset of the previous line is then stored in the non-volatile memory for later voltage-to-current conversion use.

To better understand our auto calibration process, let us introduce some notations. Suppose the N collected and already-sorted rms current and voltage samples are (i_n, v_n) for $n = 1, 2, \dots, N$ where $i_1 < i_2 < \dots < i_N$ and presumably $v_1 \leq v_2 \leq \dots \leq v_N$ since the current-voltage relation is non-decreasing.

The auto calibration algorithm is an iterative procedure, starting with first finding the gain β and offset α' in the linear equation $v = \alpha' + \beta i$ passing through the first two samples (i_n, v_n) for $i = 1, 2$. Since there are only two samples, the gain is $\beta^{(1)} = \frac{v_2 - v_1}{i_2 - i_1}$ and the offset $\alpha'^{(1)} = v_1 - \beta^{(1)}i_1$. Since a line can pass through any two points exactly, the fitting error is zero. Then, the third point (i_3, v_3) is added into the group which now has three samples $(i_n, v_n), n = 1, 2, 3$. Fitting a line now requires estimation of β and α' . We can use the OLS estimation. In general if there are M data points, the estimated gain and offset of the simple linear regression $v = \alpha' + \beta i$ are

$$\beta = \frac{\sum i_n v_n - \frac{1}{M} \sum i_n \sum v_n}{\sum i_n^2 - \frac{1}{M} (\sum i_n)^2} \quad (5.1)$$

$$\alpha' = \bar{v} - \beta \bar{i} \quad (5.2)$$

where the sample averages are $\bar{v} = \frac{\sum v_n}{M}$ and $\bar{i} = \frac{\sum i_n}{M}$.

After the line fitting, we evaluate the fitting performance by starting from the least current point in the group and evaluate the fitting error for this point. We choose our error criteria to start a new line as following:

1. If $i_n < 5$ A, we require the absolute error to be less than the absolute threshold, $AbsThreshold$, which is set to 0.3. The error is defined as

$$e_n = |i_n - \hat{i}_n| \quad (5.3)$$

where the estimated current by the line is

$$\hat{i}_n = \frac{v_n - \alpha'}{\beta} \quad (5.4)$$

2. If $i_n \geq 5$ A, we require the absolute percentage error to be less than the percentage threshold, *PercentThreshold*. The absolute percentage error is defined as

$$pe_n = \frac{|i_n - \hat{i}_n|}{i_n} \times 100\% \quad (5.5)$$

If a point n^* is found to have its fitting error greater than the corresponding threshold, the algorithm starts a new group having two points, $n^* - 1$ and n^* . Then the procedure goes on by adding the third point $n^* + 1$ into the group and doing an OLS estimate for the new line passing through this group of points. Then it checks whether the line fits the three points well enough. If not, the two points $n^* - 1$ and n^* will have its own line and the points n^* and $n^* + 1$ will start a new group. The algorithm keeps on like this until the last point N is fitted by a line, which is in the worst case the last line passes only through the last two points $N - 1$ and N . The fitting procedure is shown in Fig. 5.4.

For each line, we need to store in a lookup table the gain β and offset α' of the line and the range of measured rms voltages (the starting and ending) that we will use this line to calculate the current. Since the lines are connected and the starting point of the first line is always zero, we only need to keep the ending point, $v_{rms,boundary}$, of the interval. This lookup table is stored in the non-volatile storage of ESP32. Moreover, we need to record the *number of rows* (number of gains) in the lookup table. We can record and read the non-volatile storage by using ESP32 NVS library [18].¹

If we reset the microcontroller, it will read gains β , non-zero intercept offsets α' , $v_{rms,boundary}$, and *number of gain m* of the calibrated current sensor from MCU's non-volatile storage. Then, these value will be used in normal current measurement mode. Algorithm (2) show pseudocode of this step to find β , α' , $v_{rms,boundary}$, and number of gain (m). Fig 5.4 show a flowchart of applying this step.

¹There are other microcontrollers which also have non-volatile storage such as EEPROM.

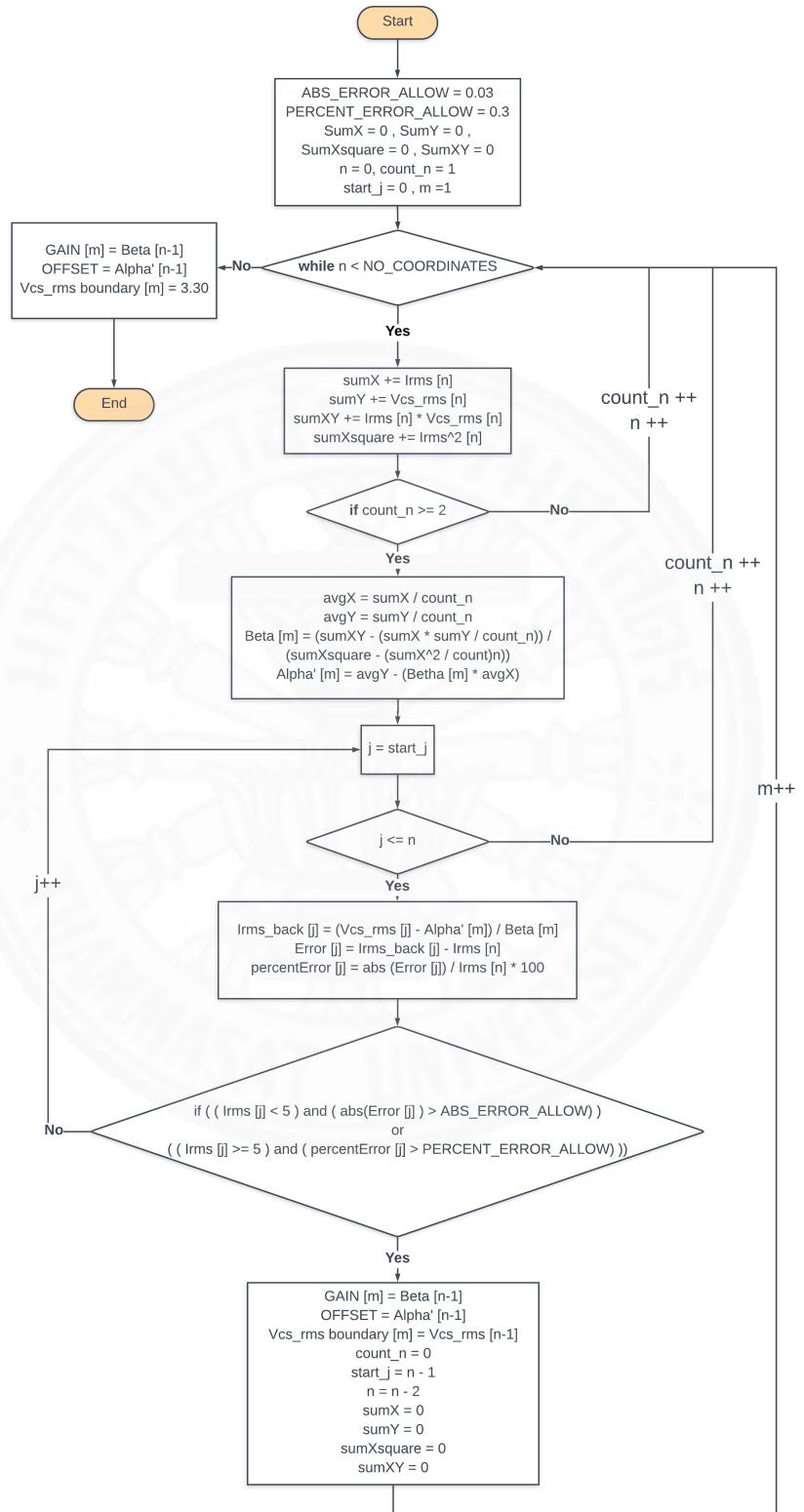


Figure 5.4 Flowchart of applying OLS step

Algorithm 2 Find Gain, Offset, V_{cs_rms} boundary, Number of gain (m)

```
procedure APPLYING OLS TO FIND  $\beta$ ,  $\alpha'$ 
    Set AbsThreshold = 0.03 A
    Set PercentThreshold = 0.3%
    Set sumX, sumY, sumXsq, sumXY = 0
    Set count_n = 1, m = 1, n = 0, start_j = 0
    while n < N do
        sumX += i_n
        sumY += v_n
        sumXsq += i_n^2
        sumXY += i_n * v_n
        if count_n >= 2 then
            avgX = sumX / count_n
            avgY = sumY / count_n
             $\beta[n] = \frac{sumXY - (sumX * sumY / count_n)}{sumXsq - (sumX)^2 / count_n}$ 
             $\alpha'[n] = avgY - (\beta[n] * avgX)$ 
            for j = start_j to n do
                 $\hat{i}[j] = (v[j] - \alpha'[n]) / \beta[n]$ 
                e[j] = |i[j] -  $\hat{i}[j]$ |
                pe[j] = (e[j] / i[j]) * 100
                if ((i[j] ≤ 5) ∧ (e[j] > AbsThreshold)) ∨ ((i[j] > 5) ∧ (pe[j] > PercentThreshold)) then
                    GAIN[m] =  $\beta[n - 1]$ 
                    OFFSET[m] =  $\alpha'[n - 1]$ 
                    v_end[m] = v[n - 1]
                    m++
                    count_n = 0
                    start_j = n - 1
                    n = n - 2
                    sumX = sumY = sumXsq = sumXY = 0
                    break
                count_n++
            
```

5.1.2.6 Sixth step: Recording constant values to non-volatile storage

Finally, after microcontroller applying OLS and got all gains β and non-zero intercept offsets α' also $v_{rms,boundary}$ boundary, all of these value will be recorded in non-volatile storage of ESP32. Moreover, it will record *number of gains* which is it will be used to read β , α' , $v_{rms,boundary}$ from ESP32's non-volatile storage in the normal measurement operation. We can simply record and read this storage by using ESP32 NVS library [18].

If we reset the microcontroller, next time it will read gains β , non-zero intercept offsets α' , $v_{rms,boundary}$, and *No.of gain* of the calibrated current sensor from its non-volatile storage. Then, these value will be used in normal current measurement mode. Fig. 5.5 shows source code of this step.

```

Serial.println("[NVS] set object to NVS") ;
NVS.setObject( "Gain", &GAIN, sizeof(GAIN) );
NVS.setObject( "Offset", &RMS_OFFSET, sizeof(RMS_OFFSET) );
NVS.setObject( "Vmargin", &MARGIN_I_Vrms, sizeof(MARGIN_I_Vrms) );
NVS.setInt("No_m", m);
Serial.println("[NVS] get object from NVS");
float* get_GAIN = (float*)NVS.getObject("Gain");
float* get_OFFSET = (float*)NVS.getObject("Offset");
float* get_VMargin = (float*)NVS.getObject("Vmmargin");
int NO_RANGE = NVS.getInt("No_m");

```

Figure 5.5 Code of recording and reading values from non-nolatile memory

5.2 Accuracy Testing Method

For both *manual* and *auto-calibration* method, we use same manner in accuracy testing. In this test, We vary constant testing loads current up to 20 A in 60 steps. We test 4 different sensors (ACS712, WCS1800, SCT013, PZEM004T) for manual calibration . Each type of sensor, we use 3 units. Then, we average them to get final value.

Not that. For manual calibration, the gain calibration is performed on only one randomly selected unit then we utilize the resulted gain for the other 2 units. This may cause some small errors since different units may have a slightly different gain. But in auto calibration, we calibrated each unit independently.

For accuracy testing in auto-calibration, we test only 3 analog output current sensors (ACS712, WCS1800, SCT013). We use both 20 and 50 points calibration to test with 60 steps current. We also find fitting error compared to testing error.

Chapter 6

Results and Discussion

6.1 Gain Calibration Results

6.1.1 Gain Calibration Results: Manual Calibration

In the *manual* calibration, the gain of the current sensor or the linear relation between V_{cs_rms} and I_{rms} and non-zero intercept offset will be a constant number. In this case, the gains β of SCT013 and ACS712 are estimated to be 0.0328 and 0.0402 respectively, with R^2 almost one and they have zero intercept as show in Fig. 6.1 and Fig. 6.2 However, the relation between V_{cs_rms} and I_{rms} of WCS1800 is not quite linear which is the slope (gain β) is decreased when I_{rms} increased. Therefore, we can arbitrarily chose to fit the relation with lines. However, we can still fit a piecewise linear function quite nicely, where we divided I_{rms} into 4 ranges as shown in Fig. 6.3

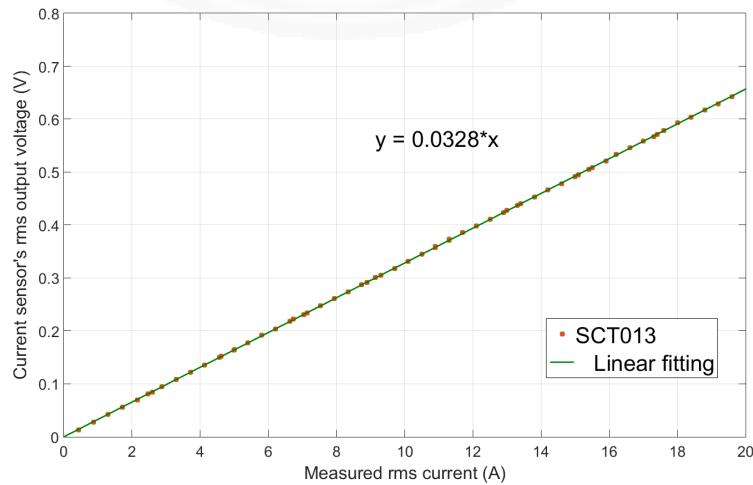


Figure 6.1 SCT013 manual calibration ($R^2 = 1$)

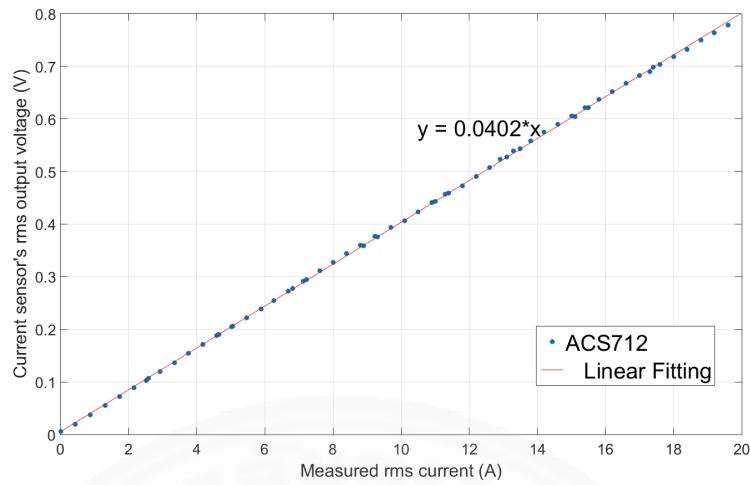


Figure 6.2 ACS712 manual calibration ($R^2 = 0.9997$)

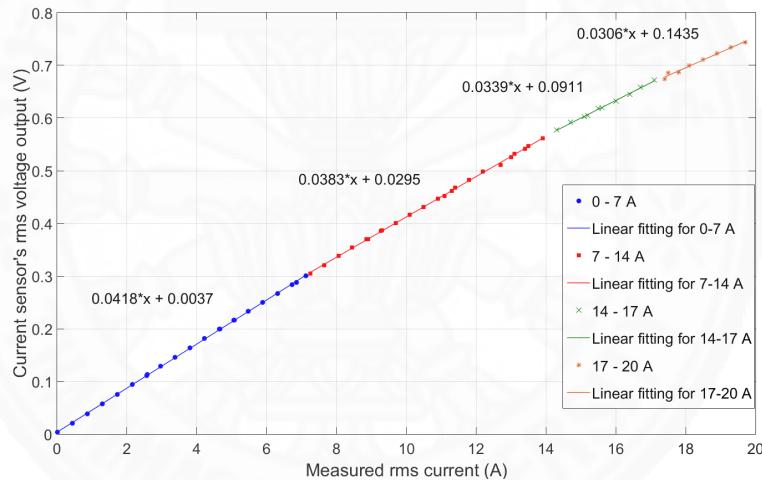


Figure 6.3 WCS1800 manual calibration ($R^2 = 0.9998, 0.9996, 0.9972, 0.9958$)

6.1.2 Gain Calibration Results: Auto-Calibration

Table 6.1 Number of gain of current sensors when applied auto-calibration

Current Sensor	50 points			20 points		
	No.1	No.2	No.3	No.1	No.2	No.3
SCT013	8	6	6	4	4	4
ACS712	11	9	9	8	8	7
WCS1800	10	11	11	7	9	9

Table 6.1 show the number of gains (number of rows) when we apply auto-calibration algorithm for 3 different analog low-cost current sensors (ACS712, WCS1800, SCT013). We tested 3 unit for each type of current sensor.

As we know that relation of SCT013 is linear with R^2 very close to one so this type of current sensor has the least number of gain compared with another current sensor. The gain result of this sensor is around 0.031-0.034 mV/A.

The result also show that ACS712 has number of gain more than SCT013 because it has lower R^2 value. The gain result of this sensor is around 0.031-0.047 mV/A but most of them vary near 0.040 mV/A.

From the 50 points calibration results of WCS1800, The number of gains is the most. It might cause from the relation between V_{cs_rms} and I_{rms} of WCS1800 is not quite linear that is the gain decrease when current increase. We can see that even in manual calibration we need to divide and define the gain into 4 ranges. In this auto-calibration's gain when current increase, the gain is start from 0.043 and continuously decrease to 0.024 mV/A as in Fig. 6.6.

We can see that for all of the current sensor, the number of gains for 50 points calibration will be more than 20 points calibration because if we have more points to calculate the linear equation, it can be sure that the microcontroller will produces more accurate gains and more different value of gains also non-zero offset intercept value.

One interesting thing of this result is, the 50 and 20 points calibration will not give any different in number much for ACS712. It might be because of the gain about 8-9 gains is enough to fit the piecewise function. In many following figures and tables show the value of gains and non-zero offset intercept for SCT013, ACS712, and WCS1800 when we applied 20 points and 50 points auto-calibration.

Table 6.2 Gain, Offset, and Boundary of SCT013 in 20 points calibration

$V_{CS rms}$ (V)		Gain (mV/A)	Offset (V)
Starting point	Ending point		
0	0.192	0.03319	-0.000928
0.192	0.261	0.03219	0.004996
0.261	0.386	0.03316	-0.002901
0.386	3.30	0.03262	0.003467

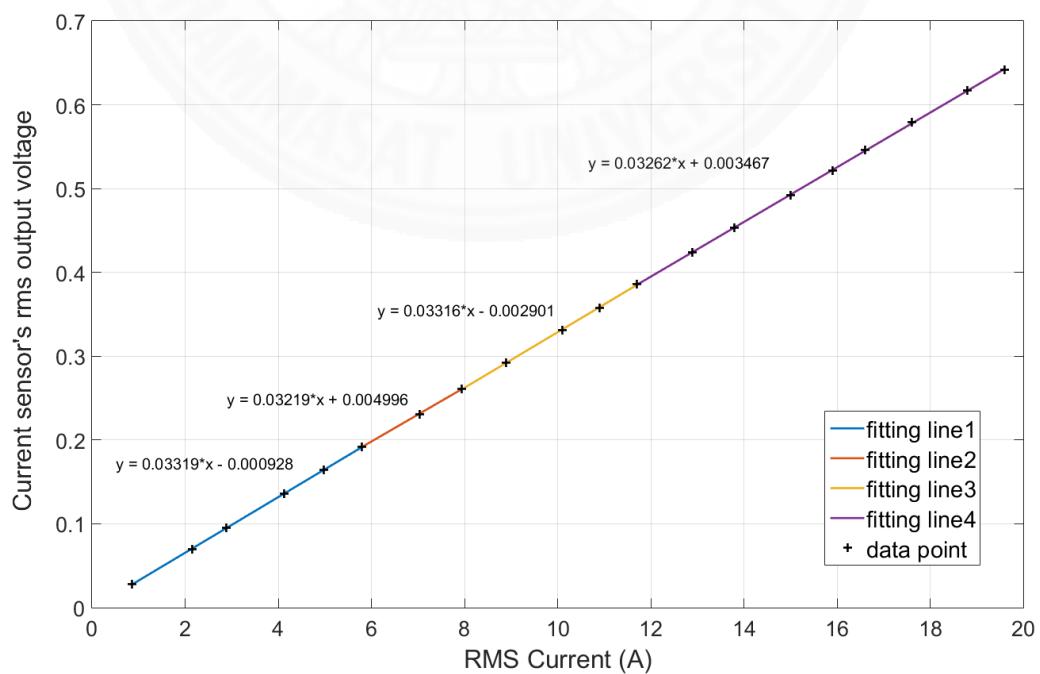


Figure 6.4 SCT013's gains when applied 20 points auto-calibration algorithm

Table 6.3 Gain, Offset, and Boundary of ACS712 in 20 points calibration

$V_{cs rms}$ (V)		Gain (mV/A)	Offset (V)
Starting point	Ending point		
0	0.188	0.04058	0.001890
0.188	0.272	0.04004	0.003892
0.272	0.377	0.04144	-0.004859
0.377	0.423	0.03622	0.042687
0.423	0.543	0.03975	0.006523
0.543	0.668	0.04020	0.001431
0.668	3.30	0.03643	0.063286

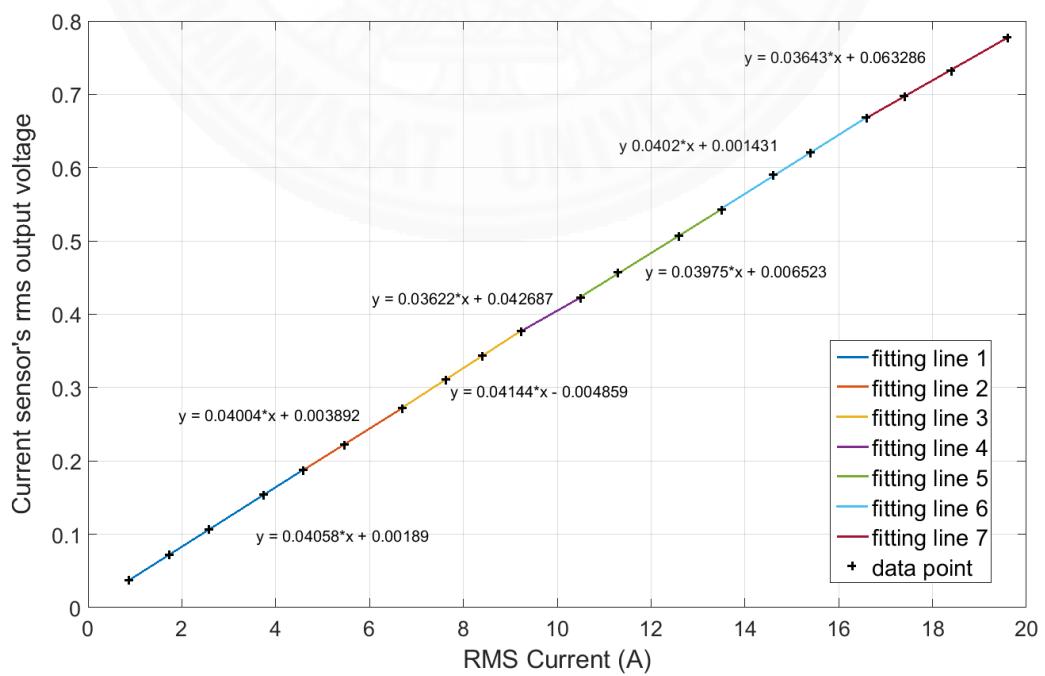


Figure 6.5 ACS712's gains when applied 20 points auto-calibration algorithm

Table 6.4 Gain, Offset, and Boundary of WCS1800 in 20 points calibration

$V_{cs rms}$ (V)		Gain (mV/A)	Offset (V)
Starting point	Ending point		
0	0.181	0.04303	0.000076
0.181	0.338	0.04084	0.009322
0.338	0.486	0.03944	0.019864
0.486	0.611	0.03810	0.036029
0.611	0.665	0.03363	0.102504
0.665	0.728	0.02864	0.186244
0.728	3.30	0.02375	0.279148

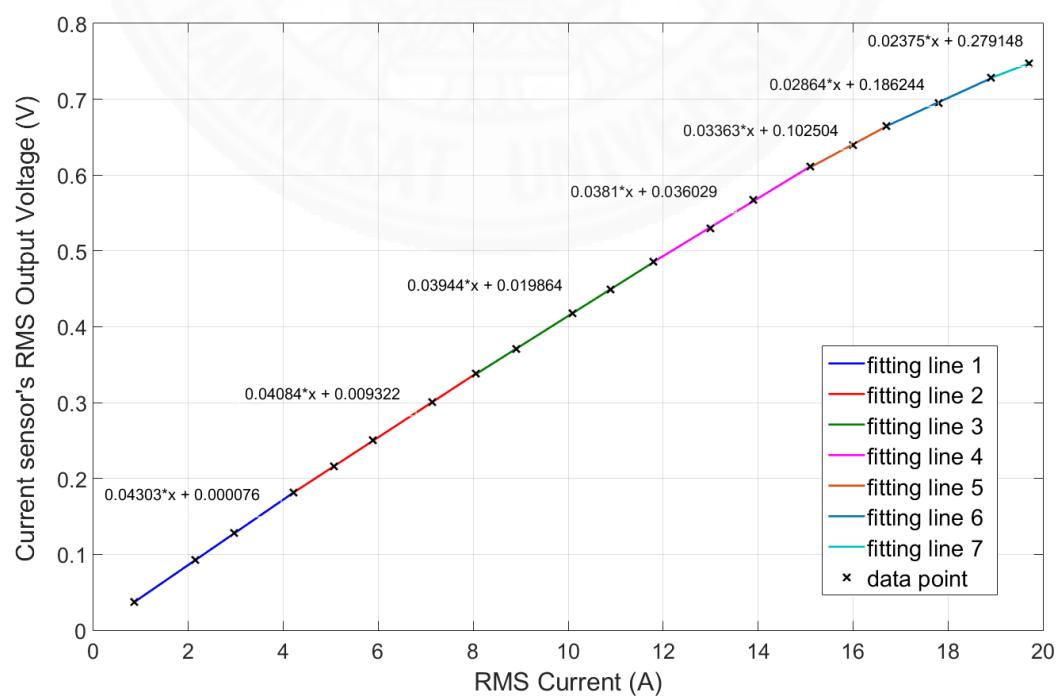


Figure 6.6 WCS1800's gains when applied 20 points auto-calibration algorithm

Table 6.5 Gain, Offset, and Boundary of SCT013 in 50 points calibration

$V_{CS rms}$ (V)		Gain (mV/A)	Offset (V)
Starting point	Ending point		
0	0.191	0.03325	-0.001637
0.191	0.301	0.03298	-0.000661
0.301	0.371	0.03255	0.003031
0.371	0.453	0.03248	0.004886
0.453	0.628	0.03283	-0.000625
0.628	3.30	0.03249	0.004185

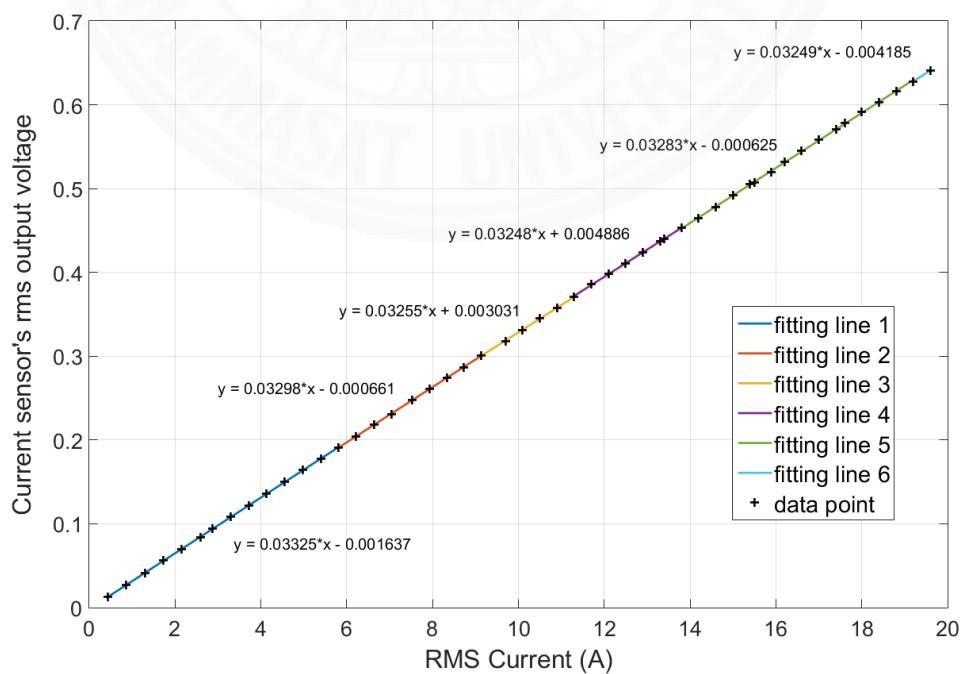


Figure 6.7 SCT013's gains when applied 50 points auto-calibration algorithm

Table 6.6 Gain, Offset, and Boundary of ACS712 in 50 points calibration

$V_{cs rms}$ (V)		Gain (mV/A)	Offset (V)
Starting point	Ending point		
0	0.205	0.04043	0.002186
0.205	0.272	0.04023	0.002676
0.272	0.377	0.04133	-0.003994
0.377	0.406	0.03342	0.068960
0.406	0.457	0.04275	-0.025673
0.457	0.473	0.03201	0.095248
0.473	0.539	0.04425	-0.049238
0.539	0.704	0.03863	0.025866
0.704	3.30	0.03757	0.042121

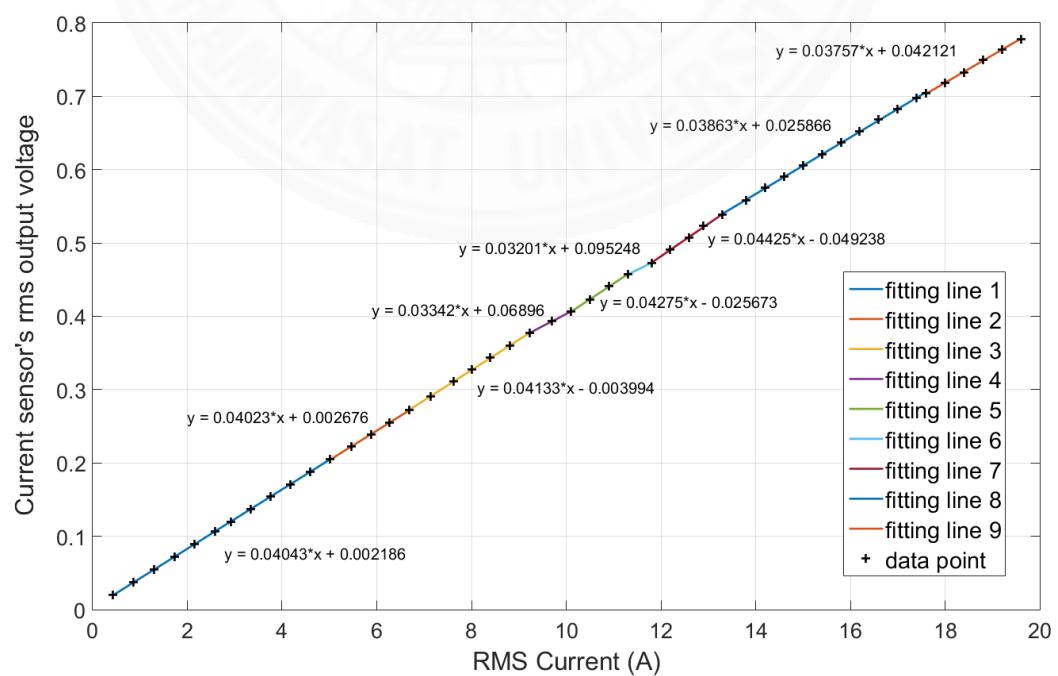


Figure 6.8 ACS712's gains when applied 50 points auto-calibration algorithm

Table 6.7 Gain, Offset, and Boundary of WCS1800 in 50 points calibration

$V_{cs rms}$ (V)		Gain (mV/A)	Offset (V)
Starting point	Ending point		
0	0.181	0.04303	0.000076
0.181	0.338	0.04084	0.009322
0.338	0.486	0.03944	0.019864
0.486	0.611	0.03810	0.036029
0.611	0.665	0.03363	0.102504
0.665	0.728	0.02864	0.186244
0.728	3.30	0.02375	0.279148

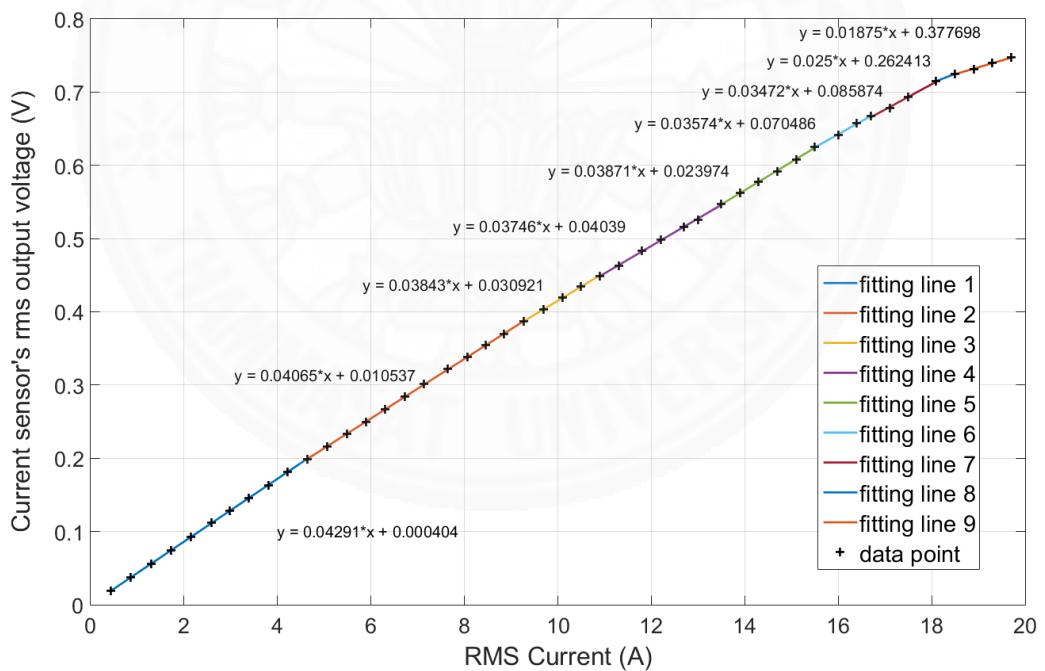


Figure 6.9 WCS1800's gains when applied 50 points auto-calibration algorithm

6.2 Accuracy Results

6.2.1 Accuracy Results: Manual Calibration

Fig. 6.10 shows the measurement error by testing 4 type of sensors. From the figure,

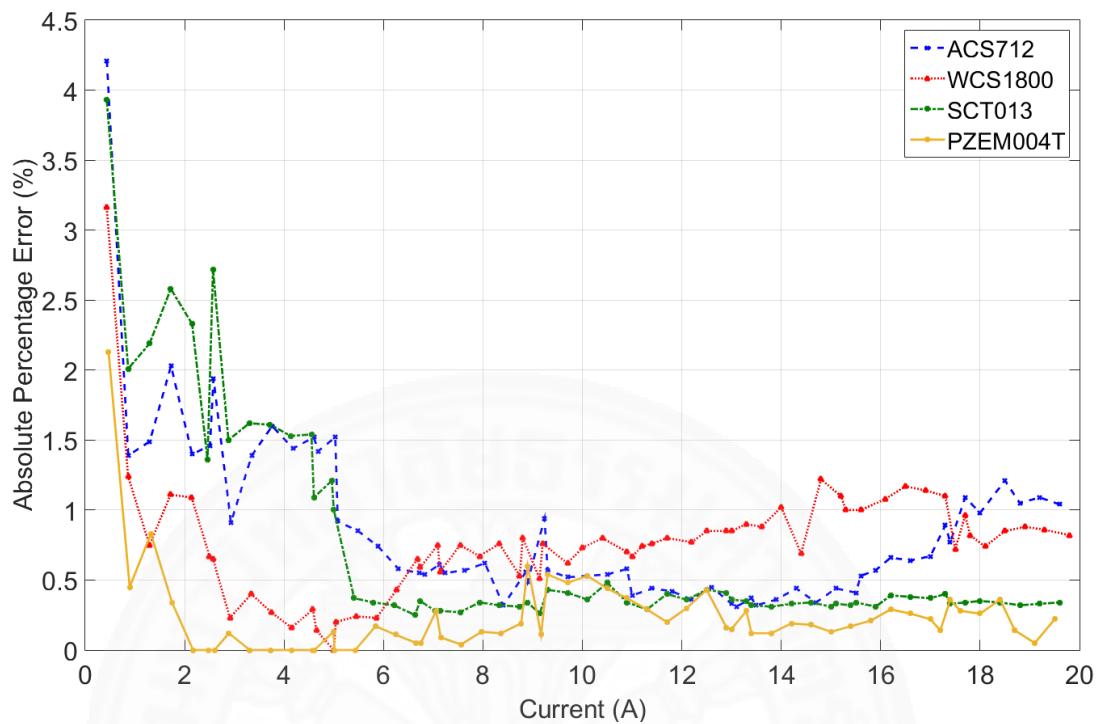


Figure 6.10 Percentage error results of four different current sensors

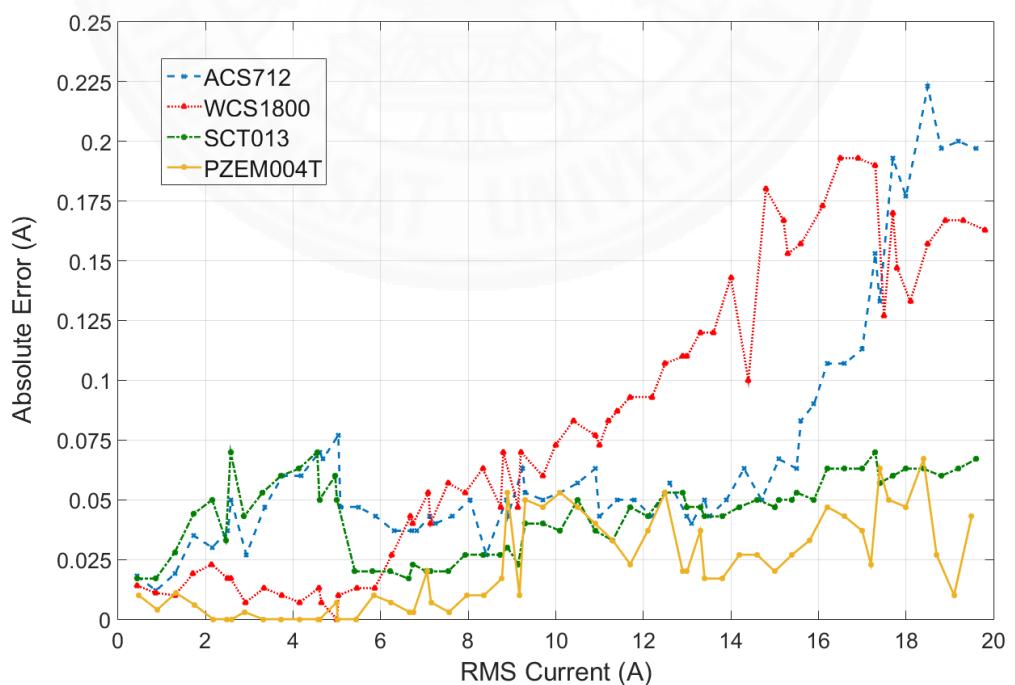


Figure 6.11 Absolute error results of four different current sensors

PZEM004T's result and SCT013's result above 5 A current is very similar and very accurate which are error $< 0.5\%$. Also, these 2 sensors have similar price and it is not expensive compared to their performance in this range. So these 2 sensors are very good option for measuring current higher than 5 A.

However, They have some difference which when we install PZEM004T sensor, we need to break a measured wire at a connection point and put the measured wire through the iron core's hole before connect the wire back. But for SCT013, we can split the iron core then just easily clamp on the measured wire. Besides, for current about 1-5 A, SCT013 produce high error result which is $> 1\%$ (if calculate absolute error it can be acceptable that is for range < 1.5 A error < 0.03 A, range 1.5-2.5 A error < 0.05 A, and 2.5-5 A error < 0.07 A) but PZEM004T still give the same percentage error which is $< 0.5\%$ (absolute error < 0.01 A throughout the range). So PZEM004T is better for measure current in the lower range.

ACS712 is a very low-cost current sensor. From the graph in Fig. 6.10, at the low current, this sensor give high error. Anyway, the error will decrease when current is higher. When measured current is about 5-11 A, the error is decrease to be equal 0.5-1 % and when current in 11-15.5 A range, the error is more decrease to be $< 0.5\%$. But for the higher current 15.5-18 A, the error is increase to be 0.5-1 %. And the very high current 18-20 A, the error slightly more than 1 % and we can see from Fig. 6.11 that absolute error is very high in this current value. It occurs because as Fig. 6.2, the slope (gain) at very high current (18-20 A) is lower than the used gain which might cause from sensor saturation. So at that range, it can be more accurate if we use a new gain but also it is not convenient and cost more time to do more manual calibration. Thus, this sensor can give good result at 5-18 A (error $< 1\%$). As this sensor give low error with the ultra-low price so we can say that this sensor is extremely worthy. But if you have more budget, PZEM004T and SCT013 will be the better choice because they give the better accuracy while still cost acceptable budget. However, ACS712 current sensor is a good option to use in ultra low-cost work with can allow some acceptable error. So it good for a small project that does not need very high accuracy measurement result.

For the last one, WCS1800. Fig. 6.10 shows that the error result is more than any other current sensor in almost all current range except current lower than 6 A that the

error is about 0.5-1 % for current 1-2.5 A (absolute error < 0.025 A) and error < 0.5 % for current 2.5-5 A (absolute error < 0.015 A) which is better than ACS712 and SCT013. Also, the error is slightly better than ACS712 in 18-20 A. So from the overall result, we can tell that this sensor did not give the better result than the other compared sensors based on this test situation. The error might cause from the gain which can not be exactly fitted because the changing of the gain. But if we fit many more gain, it might give the better result but also spend more time which is not appropriate to do in practical work. The good advantage of this sensor is the installation, we just put the measured wire through the sensor's hole like PZEM004T so it more comfortable than ACS712 installation.

6.2.2 Accuracy Results: Auto-Calibration

In this topic, we perform the test by using 3 approaches for collecting the accuracy result. First, we test the accuracy of current sensor with *manual* calibration that we already tested and we have the result from the past subsection. Second and third approach, we apply *auto*-calibration algorithm to microcontroller by using 50 and 20 steps of current then we test the accuracy of sensors. We use 20 points because we want to calibrate by using the value from each of 1 A. And we use 50 points because from our 60 steps current we have some current step that very close to each other so we cut out some point for convenience in calibration and we want to see if we use very high point in calibration how the result will come out.

In this test, we vary constant testing loads current up to 20 A. We test 3 different sensors (ACS712, WCS1800, SCT013). Each type of sensor, we use 3 units. Then, we average them to get final value. Finally, we compare manual, 20, and 50 calibration to see how the result will be.

Table 6.8 shows the fitting error which is the value of error when we use obtained β gain and α' non-offset intercept to calculate I_{rms} back by using $V_{cs,rms}$ that we got while calibration process. We can see that almost error in 50 point calibration will be not more than 0.3 % as the defined error condition (*AbsThreshold*).

As the result in Fig. 6.12 and Table 6.9, we can see that SCT013 is very accurate for all range after apply 50 points auto-calibration algorithm that is % error < 0.3 %.

Table 6.8 Percentage error of fitting error

Current Sensor	Current range (A)	Average Absolute % Error		
		50 points	20 points	Manual
SCT013	0-20	0.22	0.25	0.42
	1-5	0.29	0.32	0.64
	5-20	0.13	0.15	0.2
ACS712	0-20	0.29	0.39	0.86
	1-5	0.33	0.41	1.14
	5-20	0.2	0.26	0.59
WCS1800	0-20	0.26	0.31	0.59
	1-5	0.3	0.31	0.75
	5-20	0.23	0.28	0.32

Table 6.9 Percentage error of testing error

Current Sensor	Current range (A)	Average Absolute % Error		
		50 points	20 points	Manual
SCT013	0-20	0.23	0.35	0.73
	1-5	0.29	0.72	1.77
	5-20	0.12	0.15	0.36
ACS712	0-20	0.36	0.51	0.86
	1-5	0.41	0.63	1.51
	5-20	0.25	0.37	0.62
WCS1800	0-20	0.36	0.55	0.76
	1-5	0.29	0.46	0.48
	5-20	0.24	0.47	0.78

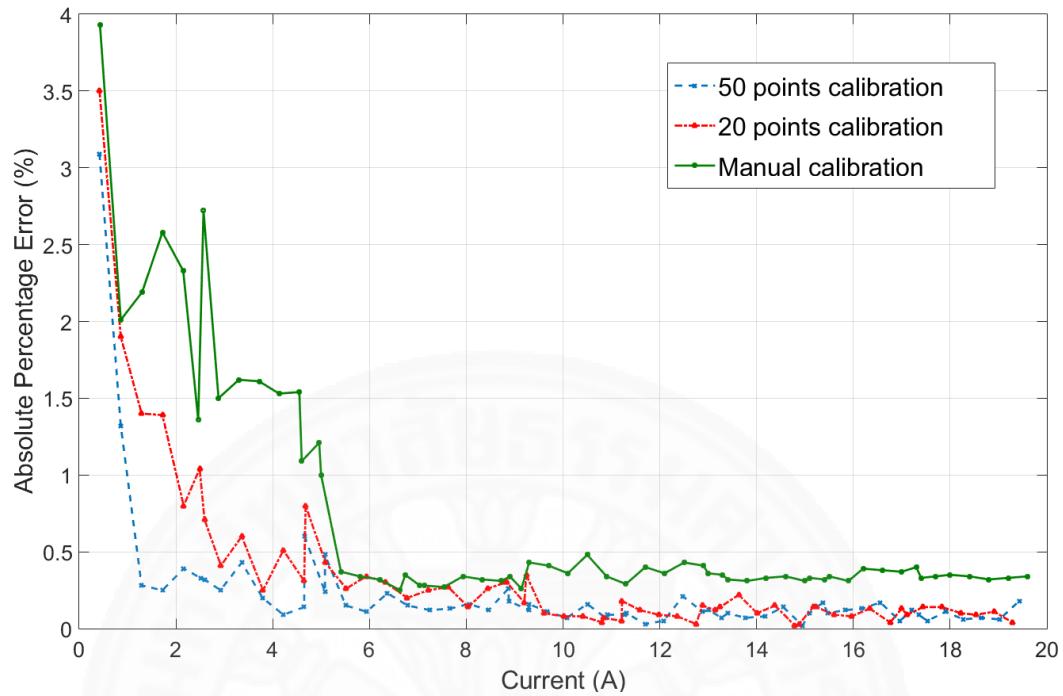


Figure 6.12 SCT013's percentage error result from accuracy test

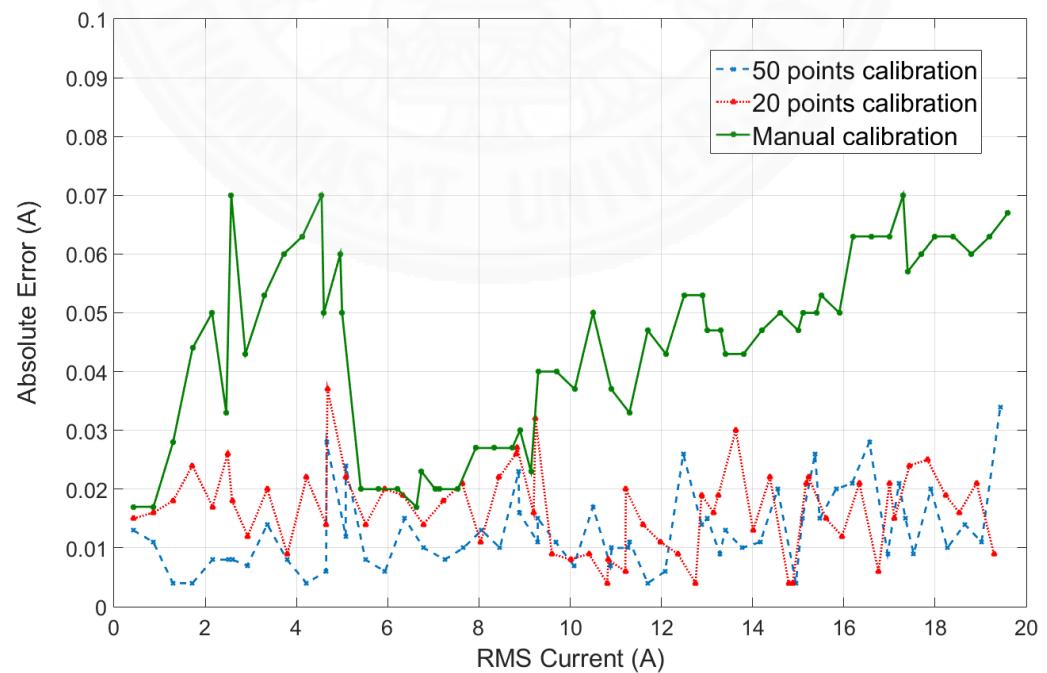


Figure 6.13 SCT013's absolute error result from accuracy test

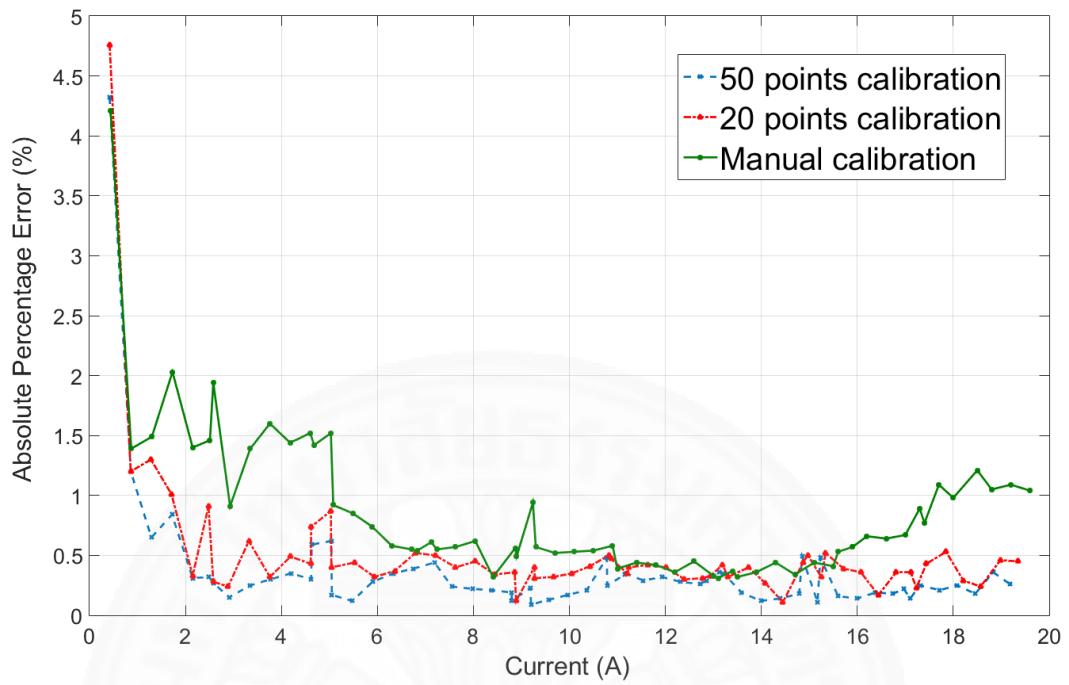


Figure 6.14 ACS712's percentage error result from accuracy test

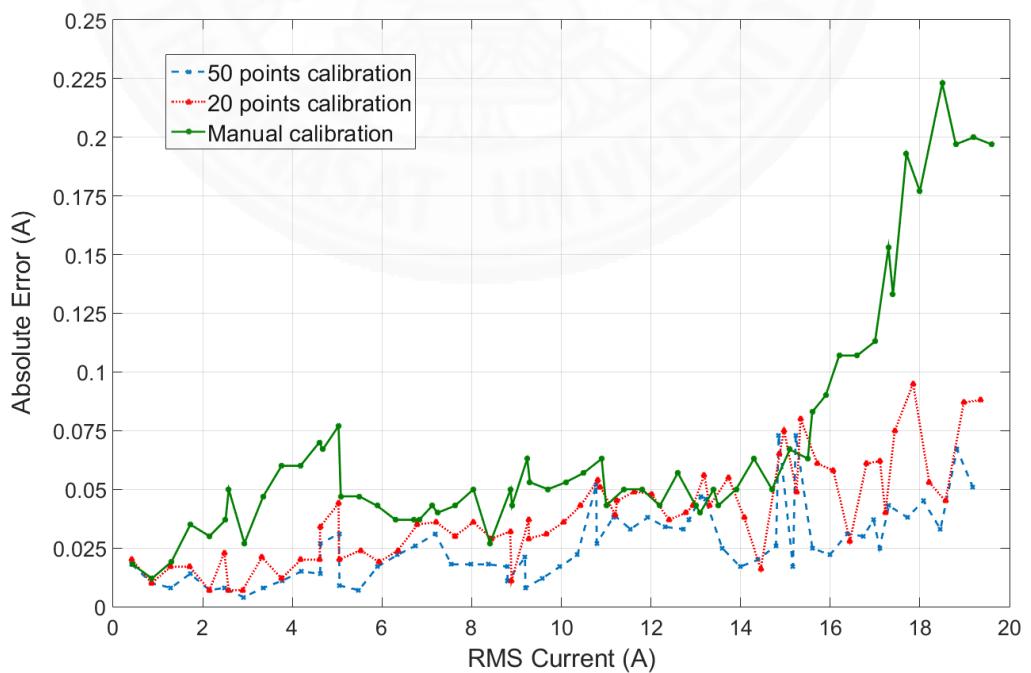


Figure 6.15 ACS712's absolute error result from accuracy test

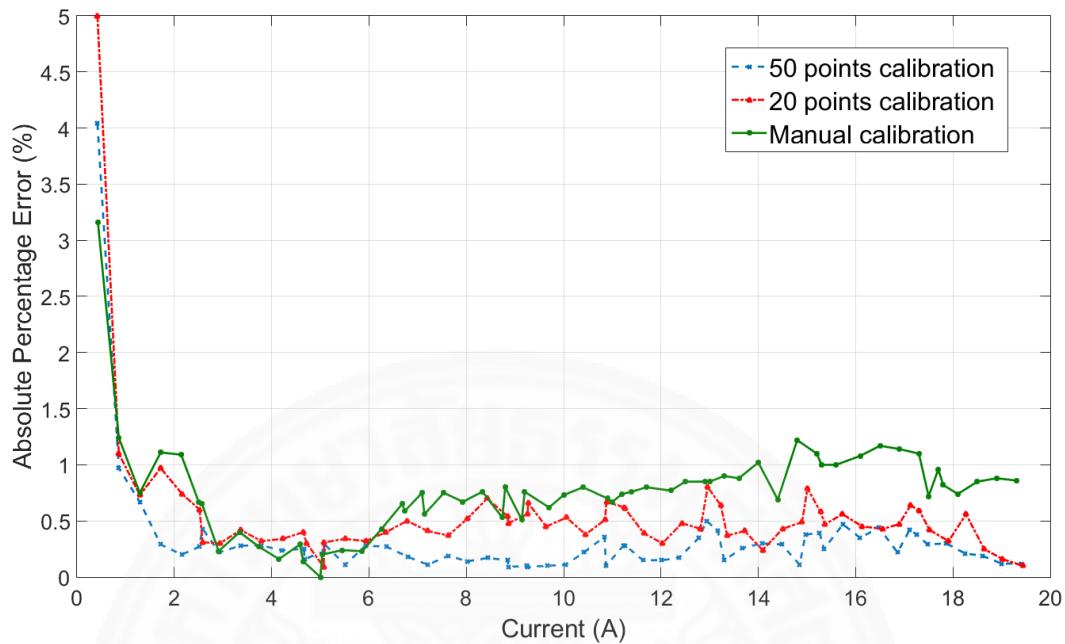


Figure 6.16 WCS1800's percentage error result from accuracy test

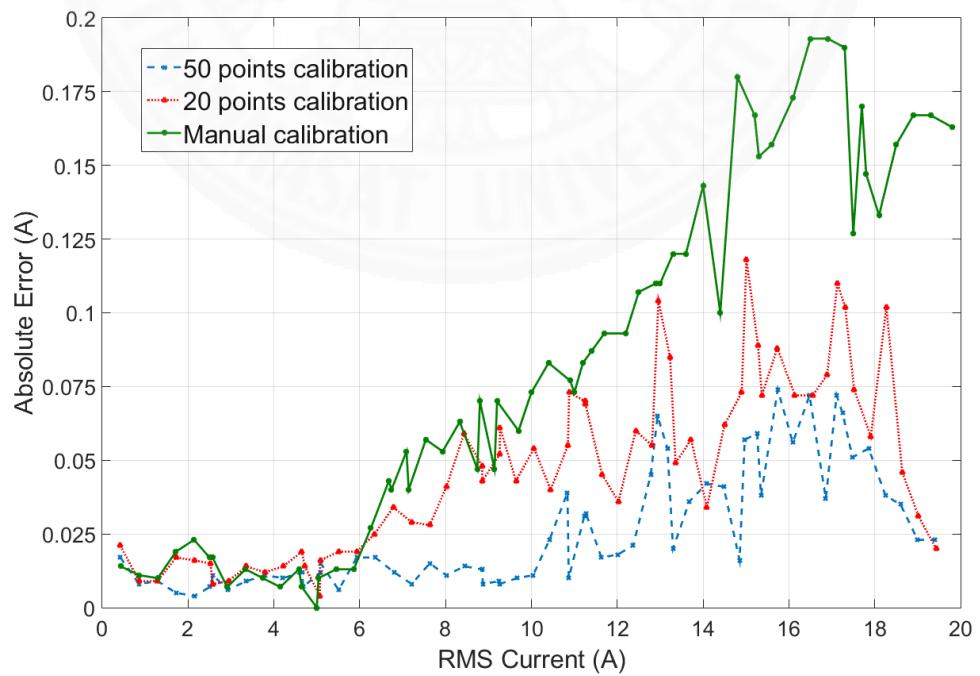


Figure 6.17 WCS1800's absolute error result from accuracy test

For 20 points calibration at low current, % error seem to be a bit high but it is still good enough. Moreover, in current > 5 A, the % error of both 20 and 50 points calibration have no different. It may cause from the relation between $V_{cs,rms}$ and I_{rms} is so linear which we can see from Fig. 6.1 that R^2 almost one. And we can see from Fig. 6.13 that the proposed auto-calibration algorithm can obviously reduce the absolute error and limit them in defined absolute threshold (0.03 A).

Fig. 6.14 and Table 6.9 shows ACS712's accuracy result that if we apply auto-calibration algorithm the result will be much better than manual calibration especially for very low current. And normally, this sensor's gain gradually decreased when it measure very high current because of the saturation of the sensor so it cause high % error. But as in the result from Fig. 6.14 and Fig. 6.15, this calibration algorithm can solve this problem by decreasing the gain to be the accurate value. From Table 6.9, we can say that 20 points calibration is enough to make this sensor to give a very acceptable accuracy that is the % error equal 0.5 %.

For WCS1800's accuracy result, the result in Fig. 6.16 and Table 6.9 show that when we apply auto-calibration algorithm, the % error is decreased in almost all range except some range which the manual calibration already give accurate measurement. We can see that 50 and 20 points calibration is obviously different. It might be because of this current sensor has the relation between $V_{cs,rms}$ and I_{rms} quite not linear as we have mentioned before so when we apply more point in the calibration the result is much better. But anyway from Table 6.9, the average % error still acceptable for 20 point calibration which is % error is about 0.5 % only for all of the current range. Also, from Fig. 6.17, we can obviously see that the auto-calibration algorithm help us to reduce absolute error effectively.

From all accuracy test result in Table 6.9, The result show that if we apply auto-calibration algorithm, it can improve accuracy of analog current sensor to be better than manual calibration as we prior expected. And we can say that if we use 50 points in auto-calibration algorithm the result will be better than using 20 points calibration but it might does not have much differences which we can see that from 20 points calibration result give very good and acceptable accuracy. So 20 points calibration will be more suitable to use in real work because it does not require so many steps of current and the

time of calibration is much less than 50 point calibration which it can improve accuracy from 20 point calibration not much.

6.2.3 Power Measurement Accuracy

In this work, we only focus on current measurement. However, in real building energy monitoring system require power measurement. So in this part we will preliminary analyze and discuss about what this calibration algorithm will affect power measurement accuracy.

Normally, average power can be calculated as follow equation.

$$P = I_{rms} * V_{rms} * PF \quad (6.1)$$

Suppose that we use the same voltage measurement which is the voltage measurement from PZEM004T that provide the % error of voltage in 200-250 V equal 1.5 %. And the power factor is some constant value so we can ignore it. we can use our current measurement's fitting error to calculate the power measurement percentage so the result of power accuracy measurement will as Table 6.10.

Table 6.10 Percentage error of power measurement

Current Sensor	Current range (A)	Average Absolute % Error		
		50 points	20 points	Manual
SCT013	0-20	0.33	0.375	0.63
	1-5	0.435	0.48	0.96
	5-20	0.195	0.225	0.3
ACS712	0-20	0.435	0.585	1.29
	1-5	0.495	0.615	1.71
	5-20	0.30	0.39	0.885
WCS1800	0-20	0.39	0.465	0.885
	1-5	0.45	0.465	0.125
	5-20	0.345	0.42	0.48

From the result, we can see that the power measurement error is less than 0.5 % for all type of sensors when we apply 50 points auto calibration.

Chapter 7

Conclusion

7.1 Summary of calibration and testing

The first part of this work, we perform a measurement accuracy comparison by testing 4 different low-cost current sensors. We also discussed about 2 more conditions to consider in selection these current sensors which are price and difficulty in installation. When we consider about price and accuracy result together, we can rank from the best to the worst performance (based on this test situation) as following, PZEM004T, SCT013, ACS712 and WCS1800. The results show that PZEM004T give the best performance with the good price also it can measure voltage and power which can be further used. SCT013 take the second place because of its very good performance with acceptable price. Follow by ACS712, it give an acceptable result with ultra-low price so it is extremely suitable for a small project. The last one is WCS1800, which is the most expensive sensor among these 4 sensors here, its overall result does not gives a good percentage measurement error and also it has the most difficult to use because of its gain change when measured current range is changed.

For the difficulty in installation, the most comfortable in installation is SCT013 because we can just split core and clamp the sensor on the measured wire so this sensor is suitable with the work or project that does not want to turn off the power supplying or circuit breaker or disconnect any wire of electrical system. Next, the slightly more difficult to install is PZEM004T and WCS1800 because we need to break a connection point of electrical system and put the measured wire through their iron core's hole. Then, we will be able to connect the wire back. And the most difficult to install is ACS712 because we need to break the circuit of the measured wire and connect sensor in series. So this sensor suitable with a small project that does not need many current sensors to be installed. Therefore, PZEM004T is the best option if your work can allow to turn-off the circuit breaker of the measured wire for current sensor installation.

The second part, we created an auto-calibration algorithm which can apply to

analog output current sensor for improving accuracy in current measurement and convenience in the real work that needs to calibration many units of sensor such as building energy monitoring system. So we focus on current from 0-20 A that air conditioner may consume. We used ESP32 wifi-microcontroller to operate both in calibration mode and normal measurement mode. In the calibration process, we used ordinary least square method and defined error threshold to get a piecewise linear function. While we calibrated the sensor, we used many loads to consume current from 0-20 A also we used this loads in normal accuracy test. We have 2 main results to be discussed which are calibration result and accuracy result.

From the calibration result, we can conclude that the gain of SCT013 and ACS712 is less than WCS1800. Because normally, relationship between $V_{cs,rms}$ and I_{rms} of WCS1800 is quite not linear compared with SCT013 and ACS712 which seem to be linear.

And from accuracy result, we can say that this calibration operation can improve the accuracy of analog output current sensor obviously from 0.7-0.9 % error in manual calibration to 0.2-0.4 % error in auto-calibration. And the more calibration coordinate we apply, the more accuracy we get. Anyway, 20 coordinate is already accurate and very good enough to use in real work. So we might do not need to apply more coordination points in the calibration because it costs more time.

7.2 Suggestion for future works

In the first work we only focus on current measurement of the current sensor module which available in the market nowadays. However, the energy IC might also interesting to do but require much more hardware and circuitry skills to create measurement module. Then, it can be massively created according to designed circuit by any hired company.

Although our auto-calibration can improve accuracy result of current sensor, we just only use a simple mathematics equation. If any future work might apply any neuron network or artificial intelligence, it might give a better result. Moreover, our proposed method is just a semi auto-calibration that still need human involve in the calibration process to insert some value by using a keypad. Future work might create the full

auto-calibration by utilize some serial communication such as RS232 which available in high grade meters then connect the meter directly with microcontroller to provide I_{rms} for microcontroller and use many relays to turn on/off AC loads automatically by microcontroller.



References

- [1] Q. M. Ashraf, M. I. M. Yusoff, A. A. Azman, N. M. Nor, N. A. A. Fuzi, M. S. Saharedan, and N. A. Omar, “Energy monitoring prototype for internet of things: Preliminary results,” in *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*, pp. 1–5, Dec 2015.
- [2] S. Thakare, A. Shriyan, V. Thale, P. Yasarp, and K. Unni, “Implementation of an energy monitoring and control device based on iot,” in *2016 IEEE Annual India Conference (INDICON)*, pp. 1–6, Dec 2016.
- [3] W. Hlaing, S. Thepphaeng, V. Nontaboot, N. Tangsunantham, T. Sangsuwan, and C. Pira, “Implementation of wifi-based single phase smart meter for internet of things (iot),” in *2017 International Electrical Engineering Congress (iEECON)*, pp. 1–4, March 2017.
- [4] G. F. Nama, D. Despa, and Mardiana, “Real-time monitoring system of electrical quantities on ict centre building university of lampung based on embedded single board computer bcm2835,” in *2016 International Conference on Informatics and Computing (ICIC)*, pp. 394–399, Oct 2016.
- [5] A. H. Shahajan and A. Anand, “Data acquisition and control using arduino-android platform: Smart plug,” in *2013 International Conference on Energy Efficient Technologies for Sustainability*, pp. 241–244, April 2013.
- [6] N. Tamkittikhun, T. Tantidham, and P. Intakot, “Ac power meter design based on arduino: Multichannel single-phase approach,” in *2015 International Computer Science and Engineering Conference (ICSEC)*, pp. 1–5, Nov 2015.
- [7] S. Wasoontarajaroen, K. Pawasan, and V. Chamnanphrai, “Development of an iot device for monitoring electrical energy consumption,” in *2017 9th International Conference on Information Technology and Electrical Engineering (ICITEE)*, pp. 1–4, Oct 2017.
- [8] Y. Thongkhao and W. Pora, “A low-cost wi-fi smart plug with on-off and energy

- metering functions,” in *2016 13th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, pp. 1–5, June 2016.
- [9] D. Despa, A. Kurniawan, M. Komarudin, Mardiana, and G. F. Nama, “Smart monitoring of electrical quantities based on single board computer bcm2835,” in *2015 2nd International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE)*, pp. 315–320, Oct 2015.
- [10] I. Allafi and T. Iqbal, “Design and implementation of a low cost web server using esp32 for real-time photovoltaic system monitoring,” in *2017 IEEE Electrical Power and Energy Conference (EPEC)*, pp. 1–5, Oct 2017.
- [11] M. N. Meziane, P. Ravier, G. Lamarque, J. C. L. Bunetel, and Y. Raingeaud, “Accuracy comparison of low-cost energy meters for home electrical consumption assessment,” in *2016 IEEE 16th International Conference on Environment and Electrical Engineering (EEEIC)*, pp. 1–6, June 2016.
- [12] I. Abubakar, S. Abd Khalid, M. Mustafa, H. Shareef, and M. Mustapha, “Calibration of zmpt101b voltage sensor module using polynomial regression for accurate load monitoring,” vol. 12, pp. 1076–1084, 01 2017.
- [13] P. L. C. Simon, P. H. S. de Vries, and S. Middelhoek, “Autocalibration of silicon hall devices,” in *Solid-State Sensors and Actuators, 1995 and Eurosensors IX.. Transducers '95. The 8th International Conference on*, vol. 2, pp. 237–240, Jun 1995.
- [14] L. E. Bengtsson, “Lookup table optimization for sensor linearization in small embedded systems,” vol. 02, pp. 177–184, 01 2012.
- [15] D. Sonowal and M. Bhuyan, “Linearizing thermistor characteristics by piecewise linear interpolation in real time fpga,” in *2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 1976–1980, Aug 2013.

- [16] A. Maier, A. Sharp, and Y. Vagapov, “Comparative analysis and practical implementation of the esp32 microcontroller module for the internet of things,” in *2017 Internet Technologies and Applications (ITA)*, pp. 143–148, Sept 2017.
- [17] olehs, “Arduino communication library for peacefair pzem-004t energy monitor.” <https://github.com/olehs/PZEM004T>, 2016.
- [18] TridentTD, “Tridenttd esp32nvs library.” https://github.com/TridentTD/TridentTD_ESP32NVS, 2017.

Appendices

Appendix A

List of publication

1. R. Khwanrit, S. Kittipiyakul, J. Kudtongngam and H. Fujita, “Accuracy Comparison of Present Low-cost Current Sensors for Building Energy Monitoring”, *The 11th International Conference om Embedded Systems and Intelligent Technology (ICESIT 2018) in cooperation with The 9th International Conference on Information and Communication Technology for Embedded System (IC-ICTES 2018)*, Thailand, 2018.

Appendix B

ESP32's analog to digital conversion

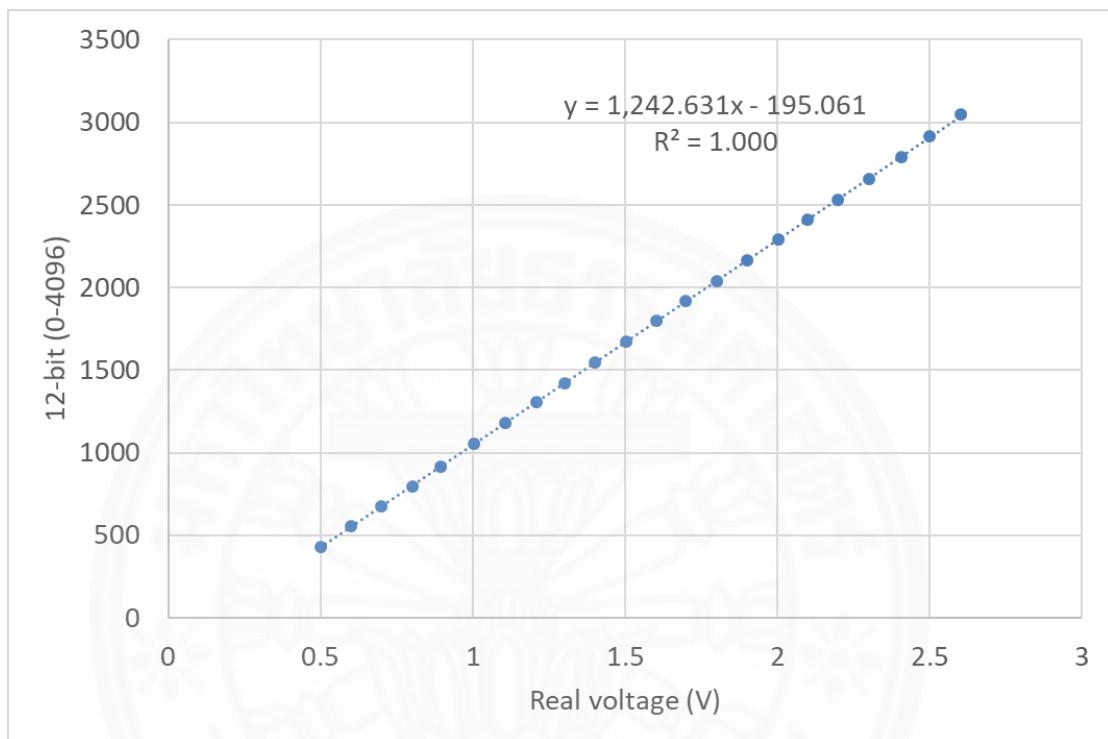


Figure B.1 Relationship between Digital 12-bit and Real voltage (V) of No.1 ESP32's ADC port

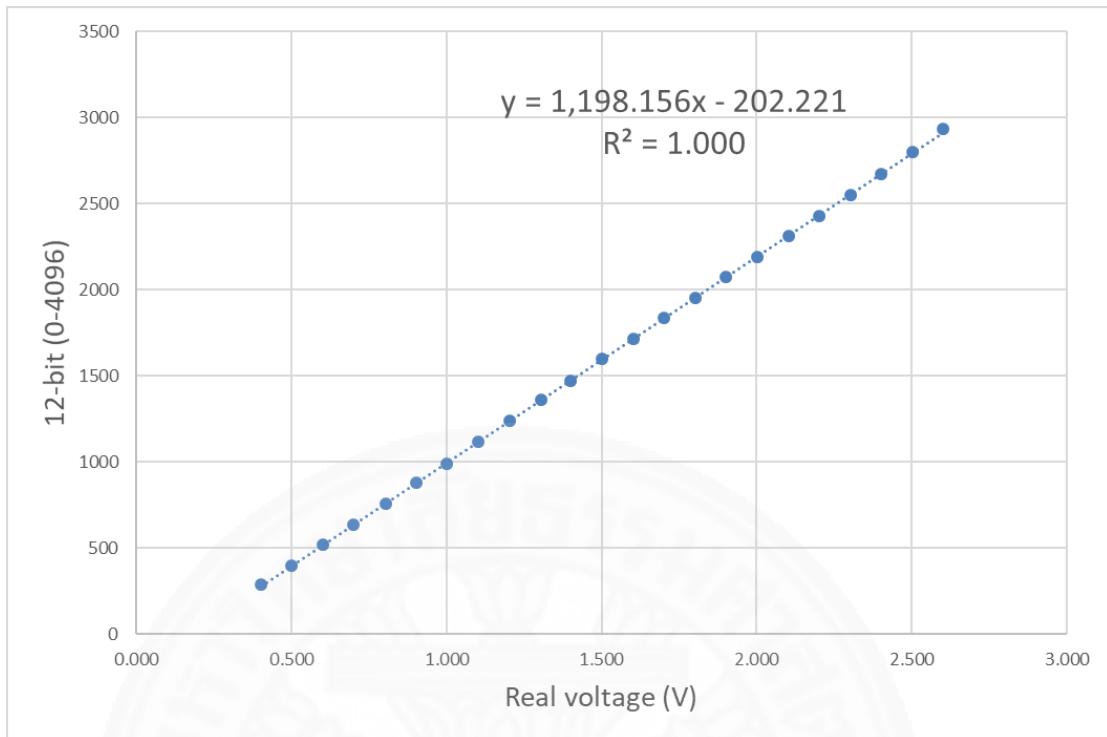


Figure B.2 Relationship between Digital 12-bit and Real voltage (V) of No.2 ESP32's ADC port

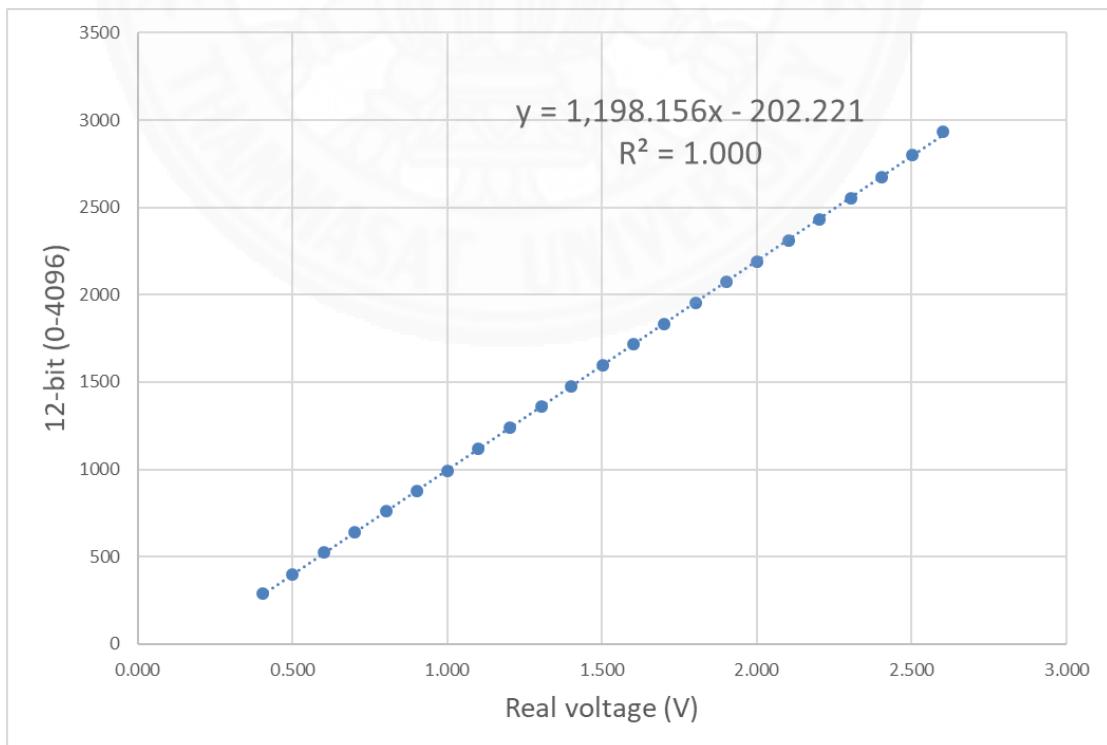


Figure B.3 Relationship between Digital 12-bit and Real voltage (V) of No.3 ESP32's ADC port