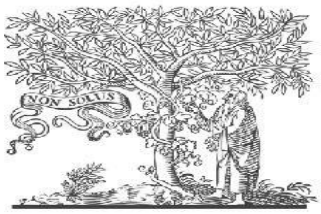
International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

COPY RIGHT



ELSEVIER
SSRN

IJIEMR Transactions, online available on 15th Nov 2021. Link
:http://www.ijiemr.org/downloads.php?vol=Volume-10&issue=ISSUE-11

DOI: 10.48047/IJIEMR/V10/I11/19

Title **Implementation of Smart Street Light System with the Assistance of Cloud Based Online Access**

Volume 10, Issue 11, Pages: 131-138

Paper Authors
Mr. Pritam Sarkar, Mr.Snehashis Saha, Mr. Tamal Das, Mr. Krishna Mangrati, Dr. V Siva Brahmaiah Rama,Dr. S S P M Sharma B,Lone Faisal

USE THIS BARCODE TO ACCESS YOUR ONLINE PAPER

To Secure Your Paper As Per UGC Guidelines We Are Providing A Electronic Bar Code

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

Implementation of Smart Street Light System with the Assistance of Cloud Based Online Access

[1]Mr. Pritam Sarkar, [1]Mr.Snehashis Saha, [1]Mr. Tamal Das,
[1]Mr. Krishna Mangrati, [2]Dr. V Siva Brahmaiah Rama,[3]Dr. S S P M Sharma B,[4]Lone Faisal
[1]*student, Department of Electrical Engineering, Budge Budge institute of Technology Kolkata West Bengal*
[2]*Associate Professor, Department of Electrical Engineering,Budge Budge institute of Technology Kolkata, West Bengal*
[3]*Assistant Professor, Department of Mechatronics Engineering, Parul University, Vadodara Gujrat*
[4] *Assistant Professor, Department of Electrical Engineering, Mewar University Chittorgarh, Rajasthan*
.

Abstract:

This paper proposes a street lighting system utilizing minimal expense microcontroller-based Arduino. Since there is a need for the industry is to connect heterogeneous equipment parts to the cloud-based internet-based admittance framework for checking and figuring out their conduct from time to time. The principle objective is to plan energy effective keen streetlamp for energy protection in existing streetlamps and safeguarding from theft issue of batteries and sunlight-based chargers in the country and metropolitan region and only for brilliant urban areas, and this paper is to record and send the readings of the proposed module to IoT which building up the reasonable web application on the cloud for Data logging. Ordinarily, we see that streetlamps are remain turned ON during daytime, absence of upkeep of batteries prompts uses misfortune. So for decreasing the referred to issues a portion of the electrical gadgets can guide specific boundaries to the microcontroller which are taken care of with the assistance of Ethernet of Wi-Fi module and associated with the cloud-based framework. A test arrangement has been made to know the situation with gadgets through Laptop/Mobile/PC at any space while associated with the cloud.

Key Words: IoT, Arduino, ESP32, smart street light, micro controller, solar panel, batteries

I. Introduction

.The ESP32 development board I loaded with lots of new features. The most relevant is it combines Wi-Fi and Bluetooth wireless capabilities and its dual-core. When it comes to the ESP32 chip specifications, it is dual-core, this means it has 2 processors, it has Wi-Fi and Bluetooth built-in, it runs 32-bit programs, the clock frequency can go up to 240MHZ and it has 512KB RAM, the particular board has 30or 36 pins, 15 in each row. It also has a wide variety of peripherals available, like capacitive touch, ADCs, DACs, UART, SPI, I2C and much more. It comes with a built-in Hall Effect sensor and a built-in temperature sensor.

II. Proposed System

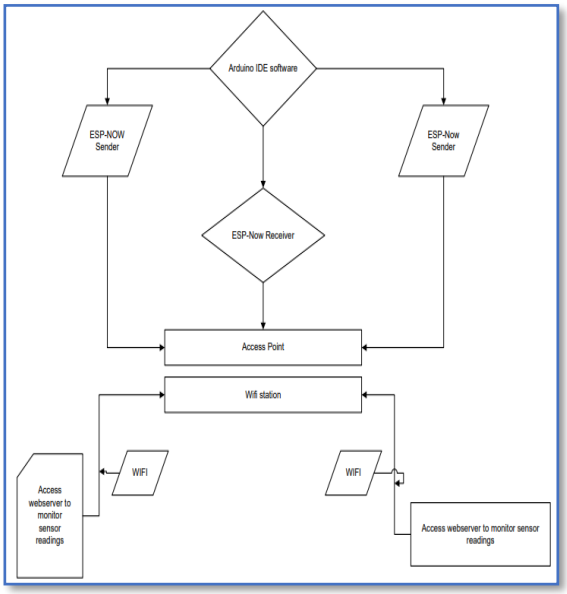


Fig 1:- Proposed system structure of ESP32

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

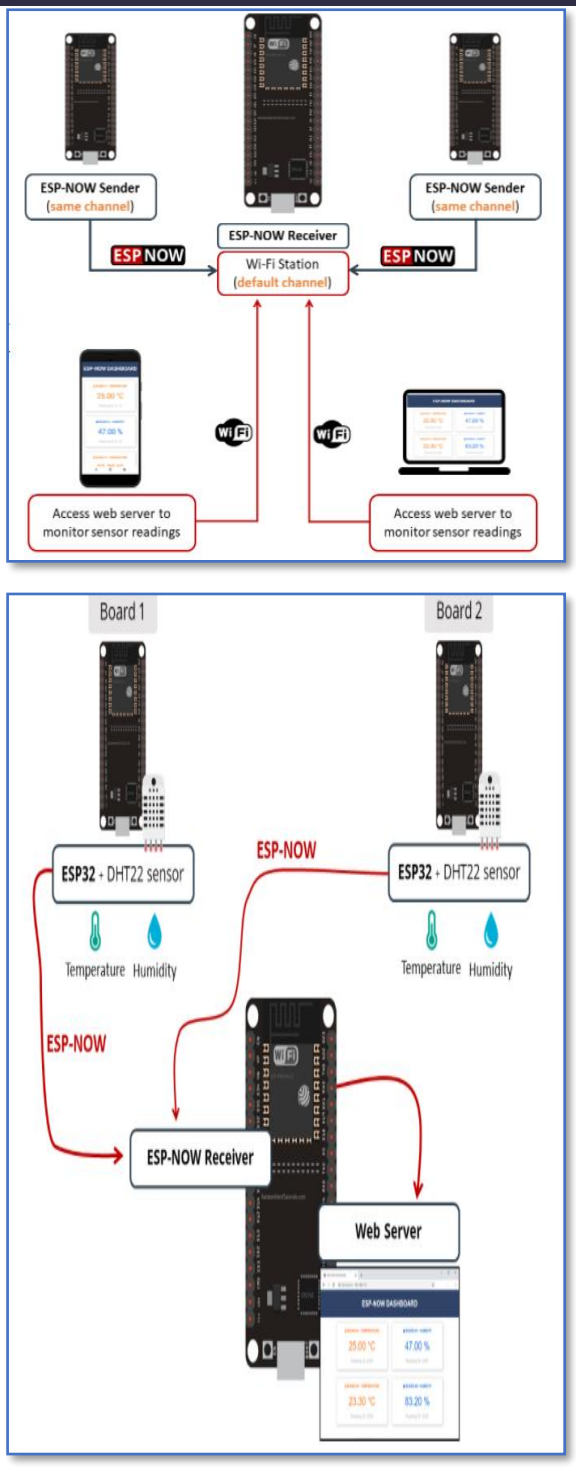


Fig 2:- Function block diagram of ESP Module

III. Procedure to Methodology

A. Installing ESP32 Board in Arduino IDE

It is compatible with Windows, Mac OSX, and Linux operating system, Arduino IDE 2.0 I an experimental software, there's an add-on for the Arduino IDE that allows programming the ESP32 using the Arduino IDE and its programming language. After installing the ESP32 add–on, if we open the Arduino IDE and it fails to compile code to ESP32 boars, we recommend re-running the Arduino IDE ESP32 add–on installation. When we try to upload a new sketch to ESP32 and it fails to connect boars, it means that ESP32 is not flashing/Uploading mode. ESP32 should have the new sketch running, with those boards.

with that setup, after uploading a new sketch, press the "ENABLE" button to restart the ESP32 and run the new upload sketch.

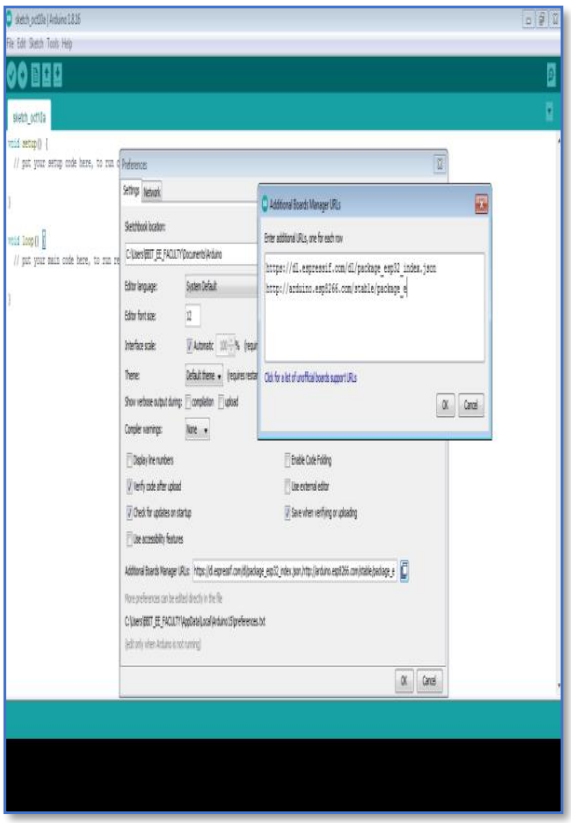


Fig 3 :- the Arduino IDE ESP32 configuration Menu

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

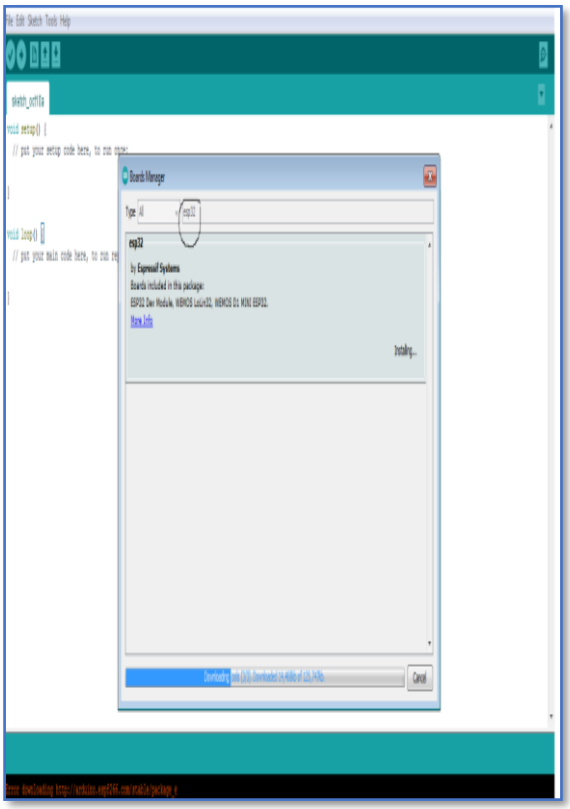


Fig 4 :-ESP 32 Board

B. ESP-NOW two-way communication between ESP 32 Boards

Two ESP32 boards will exchange sensor readings (with a range in open field up to 220 meters to 722 feet) example, we will have two ESP32 boards, Each board is connected to an OLED display and a BME280 sensor, each board gets a temperature, humidity and pressure readings from their corresponding sensors, each board send its readings, it displays them on the OLED display, after sending the readings, the board displays on the OLED if the message was successfully delivered, each board needs to know the other board MAC address to send the messages..

C. Getting the Boards MAC address

To send messages between each board, we need to know their MAC address, each board has a unique MAC address. it should upload the following code to each of ESP32 boards to get their MAC address

```
//complete instructions to get and change ESP MAC address: https//solar street lightmonitoringsystem.com
#include '' WiFi.h"
Void setup () {
Serial.begin(115200);
WiFi.mode(WIFI_MODE_STA);
Serial.printIn (WiFi.macAdress());
}
```

Fig 4:- Arduino IDE sketch

D.ESP32 two-way communication ESO-Now code working procedure

Upload the following code to each of boards, before uploading the code, need to enter MAC address of the other board (the board we are sending data to) The code is well commented so that you understand what each line of code does. To use ESO-NOW we need to include the next libraries.

```
#include<esp_now.h>

#include<wifi.h>
```

In the next line, insert the MAC address of the receiver board:

Uint_t broadcastadress[]= {0x3o, oxAE, 0xA4, 0x0D};

Create variable to store temperature, humidity and pressure readings from the BME280 sensor. These readings will be sent to the other board:

```
//Define variables to store BME280 readings to be sent
Float temperature;
Float humidity;
Float pressure;
```

Create variable to store the sensor readings coming from the other board:

```
//Define variable to store incoming readings
Float incoming Temp;
Float incoming hum;
Float incoming pres;
```

The following variable will store a success message if the readings are delivered successfully to the other board.

```
//variable to store if sending data was successful
String success;
```

Create s structure that store humidity, temperature and pressure readings

```
Typdef struct struct message{
Float temp;
Float  hum;
Float pres;
} struct_message;
```

They we need to create two instances of the structure, one to receive the readings and another to store the readings to be sent.

The BME280Readings will store the readings to be sent.

```
//crate a struct_message called BME280redings to hold sensor readings
Struct_message BME280readings;
```

The incoming readings will store the data coming from the other board.

```
//create struct message to hold incoming sensor readings
Struct message incoming readings;
```

Then we need to create two callback function, one will be called when data is sent, and another will be called when data is received.

E. ESP32 webserver –Arduino IDE

Here we will create a standalone webserver with an ESP32 that control outputs (two LEDS) using the Arduino IDE programming environment. The webserver is mobile responsive and can be accessed with any device that as a browser on the local network. Here the process to create the webserver and how the code works step by step.

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

The webserver we build controls two LEDs connected to ESP32 GPIO26 and GPIO27, we can access the ESP32 webserver by typing the ESP32 IP address on browser in the local network, by clicking the buttons on webserver we can instantly change the state of each LED. Connected two LEDs to the ESP32 and one LED connected to GPIO26, and the other to GPIO27. We are suing the RSP32 DEVKIT DOIT board with 36 pins.

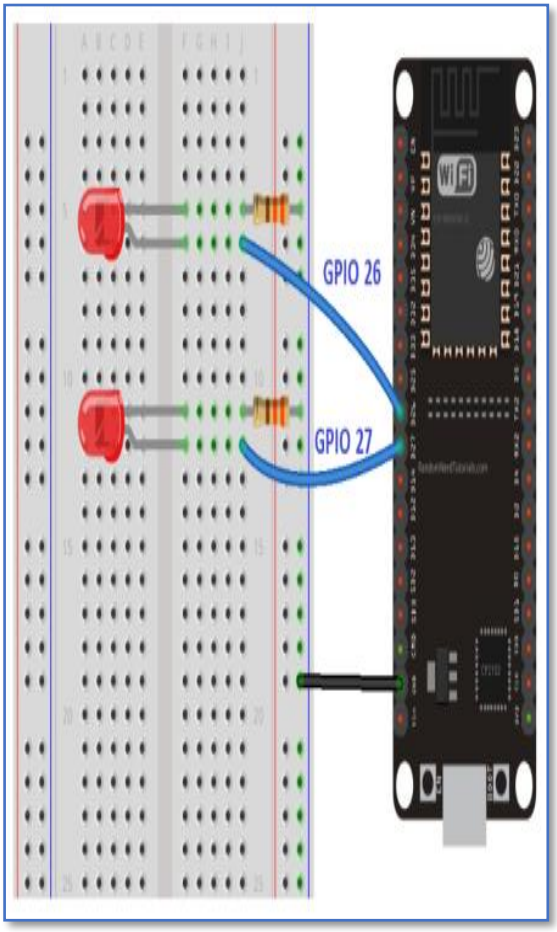


Fig 5:- LED ESP32 layout

Code and Uploading the code: - plug ESP32 board in computer, in the Arduino IDE select board in Tools > Board (in our case we are using the ESP32 DEVKIT BOIT board), next select the COM port in Tool>port. press the upload button in the Arduino IDE and wait a few seconds while the code compiles and upload to the boards. After uploading the code, open the serial monitor at a baud rate of 115200, press the ESP 32 EN button (reset). The ESP32 connection to Wi-Fi, and outputs the ESP IP address on the serial monitor. Copy that IP address, because we need it to access the ESP32 webserver.

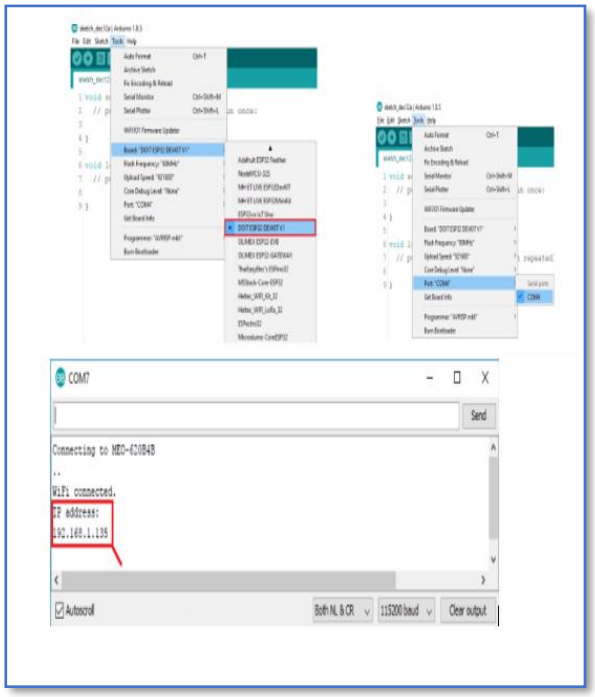


Fig 6:-ESP32 Webserver

Accessing the webserver: To access the webserver, open browser, paste the ESP32 IP address, and we see the page, in our case it is 192.168.1.135 if we take a look at the serial monitor, we can see what's happening on the back ground. The ESP receives and HTTP request form a new client.

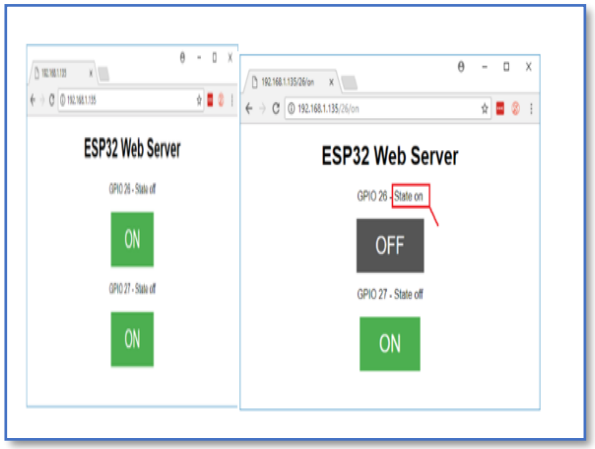


Fig 7:- ESP webserver Testing

Testing the webserver: - we can test if our webserver is working properly, click the buttons to control the LEDs, at the same time, we take a look at the serial monitor to see what s going on in the background. For example, when you click the button

to turn GPIO 26 On, ESP32 receives a request on the /26/on URL., when the ESP32 receives that request, it turns the LED attached to GPIO 26 ON and updates its state on the webpage. The button for GPIO 27 works in a similar way .

IV.Working Methodology

Here we are taking ESP 32 + DHT22 sensor 2 quantity, and ESP-NOW receiver. ESP _NPW receiver can take the readings from the both boards and place the readings in the website. Website is updating automatically due to every time server sending new readings. Each ESP Boards have standard structure. the programme is consists of temperature, humidity and reading ID . the reading ID keep track the how many no. of messages send. The purpose of this project checking the prerequisite data of the streetlight if all readings are zero means some error or maintains required.

1. Arduino ide we should install the ESP32, to get the reading from DHT sensor we should take DHT sensor from adafruit/Adafruit sensor data sheet and should install adafruit unified sensor Driver.
2. To build the webserver we should install ESPAsync Webserver and AsyncTCP.
3. For this project we need 3 ESP 32boards and following components (1. 3 ESP 32 sensor 2. 2 DHT22/DHT11 sensors 3. 2 4.7K ohm resistors, 2 Breadboards, jumper wires)
4. We send the message from sending boards, we need receiver board mac address, each board has unique mac address.
5. In Arduino soft Id software, we should write the mac address
6. In Arduino soft id for two boards we should write two unique addresses.

V. Results

Street light s monitoring by ESP32 web server with the DHT11 or DHT22 that displays temperature and humidity using Arduino IDE, the webserver build updates the readings automatically without the need to refresh the webpage. Here we should read temperature and humidity from the DHT sensor, build an asynchronous web server using the ESPAsynwebserver library, and the sensor readings automatically update without the need to refresh the webpage, to build the webserver we will use the ESPAsyncWebserver library that provides an easy way to build an asynchronous web server. Here we are connecting the data pin to GPIO27, but we can connect it to any other digital Pin. We can see in the diagram for both DHT11 and DHT22 sensors. We need to install a couple of libraries the DHT and the Ad fruit unified sensor driver libraries to read from the DHT sensor, ESPAsyncWebserver and A sync TCP libraries to build the asynchronous webserver.

Open the browser and type the ESP32 IP address, our web server should display the latest sensor readings, notice that the temperature and humidity readings are updated automatically without the need to refresh the webpage

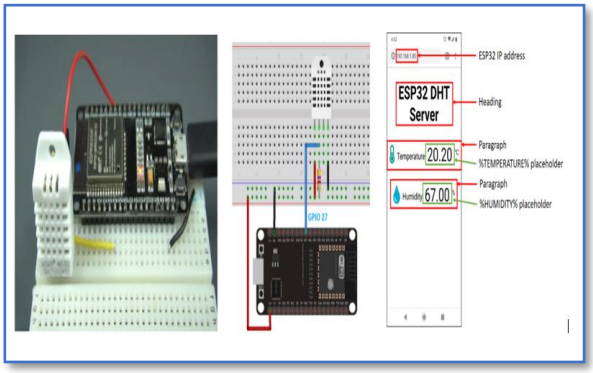


Fig 8:- ESP32 DHT Server displaying readings

VI. Conclusion

The paper presents ESP32 system using a low-cost microcontroller-based Arduino for energy-efficient smart street light to tackle the theft problems and for the judicious use of energy. It has been proved by the recorded data that the proposed system configuration is best suited for the smart street lights. Moreover the experimental setup of the system is easy to install and provides the practical results The objective of this paper has been achieved as the system proves that power electricity can be saved.

References:

[1] L. Atzori, A. Iera, and G. Morabito, "The Internet of things: A survey,"Computer Networks, vol. 54, no. 15, pp. 2787–2805, Oct. 2010.

[2] S. Li, L.D. Xu, and S. Zhao, "The Internet of things: A survey,"Information Systems Frontiers, vol. 17, no. 2, pp. 243–259, April 2015.

[3]K.J. Singh, and D.S. Kapoor, "Create your own Internet of Things: Asurvey of IoT platforms," IEEE Consumer Electronics Magazine, vol. 6,no. 2, pp. 56-68, April 2017

[4] N. Kolban, Kolban's Book on ESP32, USA: Leanpub, 2017.

[5] A. Maier, A. Sharp, Y. Vagapov, "Comparative analysis and practical implementation of the ESP32 microcontroller module for the internet of things"2017 Internet Technologies and Applications, ITA 2017 - Proceedings of the 7th International Conference IEEE, pp 143-148, November 2017, DOI:10.1109/ITECHA.2017.8101926.

[6] I. Allafi, T. Iqbal, "Design and implementation of a low cost web server using ESP32 for real-time photovoltaic system monitoring" 2017 IEEE ElectricalPower and Energy Conference, EPEC 2017, pp 1-5, February 2018.

[7] B.S. Sarjerao, A. Prakasarao, "A Low Cost Smart Pollution Measurement System Using REST API and ESP32" 3rd International Conference for Convergence in Technology, I2CT 2018, November 2018, DOI: 10.1109/I2CT.2018.8529500.

[8] A.H. Abdullah, S. Sudin, M.I.M. Ajit, F.S.A. Saad, K. Kamaruddin, F. Ghazali, Z.A. Ahmad, M.A.A. Bakar, "Development of ESP32-based Wi-Fi Electronic Nose System for Monitoring LPG Leakage at Gas Cylinder Refurbish Plant" 2018 International Conference onComputational Approach in Smart Systems Design and Applications, August 2018, DOI: 10.1109/ICASSDA.2018.8477594

[9] D. Ghosh, A. Agrawal, N. Prakash, P. Goyal "Smart Saline Level Monitoring System Using ESP32 And MQTT-S" 20th International Conference on e-Health Networking, Applications and Services, Healthcom 2018, DOI:10.1109/HealthCom.2018.8531172.

[10] A. Iqbal, T. Iqbal, "Low-cost and Secure Communication System for Remote Micro-grids using AES Cryptography on ESP32 with LoRa Module" 2018 IEEE Electrical Power and Energy Conference (EPEC), October 2018,DOI:10.1109/EPEC.2018.8598380.

[11] S. Bipasha Biswas, M. Tariq Iqbal "Solar Water Pumping System Control Using a Low Cost ESP32 Microcontroller" Canadian Conference on Electrical and Computer Engineering, CCECE 2018, May 2018, DOI: 10.1109/CCECE.2018.8447749.

[12] P. Urban, L. Landryova, "Collaborative Operations Using Process Alarm Monitoring" IFIP WG 5.7 International Conference on Advances in Production Management Systems, APMS 2017, September 2017, DOI: 10.1007/978-3-319-66923-6_52.

[13] Alexander Maier, Andrew Sharp, Yuriy Vagapov Comparative analysis and practical implementation of the ESP32 microcontroller module for the Internet of Things" Internet of things applications, IEEE 2017.

[14] Ruengwit Khwanrit, Somsak Kittipiyakuly, Jasada
Kudtongngamz and Hideaki Fujitax, "Accuracy comparison of present low-cost current sensors for
building energy monitoring" International conference on ICESIT-ICICTES, 2018.s