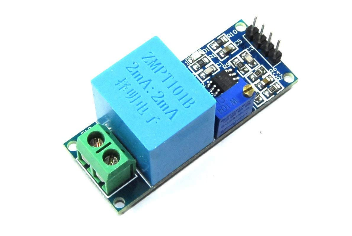
IOT part of Residential EE

Sensor data is the output of a device that detects and responds to some type of input from the physical environment. Sensors can be used to detect just about any physical element. Here we are using a few sensors to take an idea of the number and diversity of their application

ZMPT101B

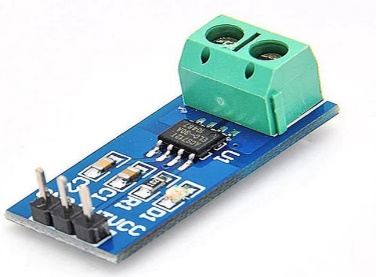


ZMPT101B AC Single Phase Voltage Transformer Sensor Relay is the best where we need to measure the accurate AC voltage with the voltage transformer. This is an ideal choice to measure the AC voltage using Arduino an open-source platform. It deals with measurements with a few basic requirements like.

* High galvanic isolation
* Wide Range
* High accuracy
* Good Consistency

ZMPT101B is a high-precision voltage Transformer. This module makes it easy to monitor AC mains voltage up to 1000 volts. A tiny little thing the size of a bouillon cube. Holds up to 4kV per breakdown voltage, and the ratio of turns is 1: 1, but this is a current transformer of 2mA: 2mA. We feed it a current and remove the current. The input current is simply set by the resistor in series R1, and we use a sampling resistor R2 in parallel to get the output voltage.

ACS712



For measuring current in a circuit, a sensor is required. ACS712 Current Sensor is the sensor that can be used to measure and calculate the amount of current applied to the A conductor without affecting the performance of the system. ACS712 Current Sensor is a fully integrated, Hall-effect-based linear sensor IC. This IC has a 2.1 kV RMS voltage isolation along with a low-resistance current conductor.

Current Sensor detects the current in a wire or conductor and generates a signal proportional to the detected current either in the form of analogue output. Current Sensing is done in ways — Direct sensing and Indirect Sensing. In Direct sensing, to detect current, Ohm's law is used to measure the voltage drop that occurred in

a wire when current flows through it. A current-carrying conductor also gives rise to a magnetic field in its surrounding. In Indirect Sensing, the current is measured by calculating this magnetic field by applying

Faraday's law.

ACS712 Current Sensor uses the Indirect Sensing method to measure the current. To sense current, a low-offset Hall effect sensor circuit is used in this module. This sensor is located at the surface of the IC on a copper conduction path. When current flows through this copper conduction path it generates a magnetic field which is sensed by the Hall effect sensor. A voltage proportional to the sensed magnetic field is generated by the Hall sensor, which is used to measure current.

Here we are using an ESP32 and the designated sensors to read the real-time value of Voltage Drop and Current passing through the system and posting these values to the cloud. To get the updated sensor value from time to time and to post these updates simultaneously we are using Arduino’s ThingSpeak Library. For cloud operations, we are using ThingSpeak MQTT API. We are using ESP32 as a WIFI station in our case.

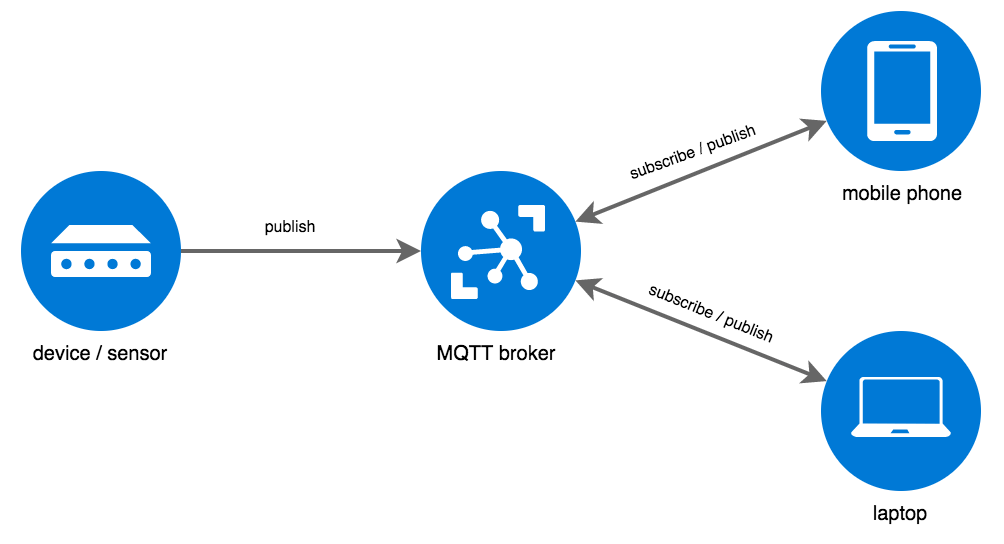
When the ESP32 is set as a Wi-Fi station, it can connect to other networks (Like our router). In this scenario, the router assigns a unique IP address to your ESP board. We can communicate with the ESP using other devices (stations) that are also connected to the same network by referring to the ESP’s unique IP address. The router is connected to the internet, so we can request information from the internet using the ESP32 board like data from APIs, and publish data to online platforms.

We are reading two analogue inputs with the ESP32 using the Arduino IDE is as simple as using the analogRead () function. It accepts as an argument, the GPIO you want to read. The ESP32 supports measurements in 18 different channels. Only 15 are available on the DEVKIT V1 DOIT board (version with 30 GPIOs). ESP32’s I2C controller operating as master is responsible for establishing communication with I2C slave devices and sending commands to trigger a slave to action, to take environmental altitude measurements, and send the atmospheric pressure readings back to the master i.e. Our ESP32.

ThingSpeak is an open-source Internet of Things (IoT) application to store and retrieve data from things, using REST and MQTT APIs. ThingSpeak enables the creation of sensor logging applications, location tracking applications, and a social network of things with status updates. MQTT is a publish/subscribe architecture that is developed primarily to connect bandwidth and power-constrained devices like our ESP32 over wireless

networks. It is a simple and lightweight protocol that runs over TCP/IP sockets or WebSockets. The publish/subscribe architecture enables messages to be pushed to the client devices without the device needing to continuously poll the server. The MQTT broker is the central point of communication, and it is in charge of dispatching all messages between the senders which are our ESP32, and the rightful receivers which are our ThingSpeak server. Our ESP32 is a client that connects to the broker and can publish or subscribe to topics to access the information. The broker delivers all messages with the matching topic to the

appropriate clients.



In our case, we are using MQTT Publish. Publish to a channel Field The Write API Key is required to publish. The broker acknowledges a correct CONNECT request with CONNACK. The MQTT protocol is supported in a built-in library in the Arduino IDE. This protocol is used to send data from our ESP32, over WIFI, to a free cloud database, such as ThingSpeak.

MQTT is a powerful standard for IoT systems. ThingSpeak enables clients to update and receive updates from channel feeds via the ThingSpeak MQTT broker.