Remote Sensors

Remote Sensors are required in our everyday lives so we can remotely get Sensor Data like Weather, and that requires more than just one Sensor, or one type of Sensor, and then we need a way to interact with it, maybe we want alerts if it is raining, or too hot, and we need to do something important should this occur, but we do not want to sit at this remote location to do so, and where this Project comes in, I have articles and videos on how to build this system as a DIY, complete with all Source Code, CAD, and other files you would get if you bought it, which is optional, but I have no secrets if you buy it, this is a DIY or Do It Yourself project and is free and open source code, but the Hardware is specific to a one company in which I have no Affiliation with other than as a loyal customer, so I will set out to build to Motherboards, the first is based on the Raspberry Pi 4 and the next one will be based on the Nvidia boards.

The Remote Sensors project is about building a platform which we can build an array of Sensors using PoE cable to remote locations where sensors are required, this can be for a Smart Home, or office at work, to CNC manufacturing, the usage is limited only to what this device can handle, so I did my homework, and would like to teach you how I built my Remote Sensor Pack.

The Pack consist of a Computer, in this case a Pi and later a Nvidia board, so we can add more horsepower to it, and see which one will work better for our needs.

The Computer system will run Linux, in the case of Pi it is the Raspberry Pi OS previously called Raspbian, Nvidia has its own version of Linux that is written specifically for that platform, but in both cases we need it to record sensor readings and send out alerts if they are not within a specific range, and to serve a web page that shows all real-time data, as well as an Android Application I will write using the Android or Qt 6 framework, maybe both to compare them, so we will cover a lot of Computer Programming, but it will include all the Code at GitHub.

I am designing this for DIY or Do It Yourselfers like myself but want to also sell this due to its usefulness and complexity to build, and the way I do this is I make videos about building a unit, and then I sell that unit, and build another one with that money, and that is how I make the video series, as such, if you are in need to remotely monitor anything, then this is the project for you.

Once completed the project will have a Qt QML application to run it, it will also have a Qt C++ backend that runs as a service, and updates the web page or pages in real time, and it is from here you have various screens to set up each type of sensor, and eventually I will cover them all, so it will handle Email, SMS or voice alerts if any Sensor is out of range, for a Smart Home this can be a door or window opening, a room with water pipes that is freezing, we can have sensors for sound, light, air pressure, gas,

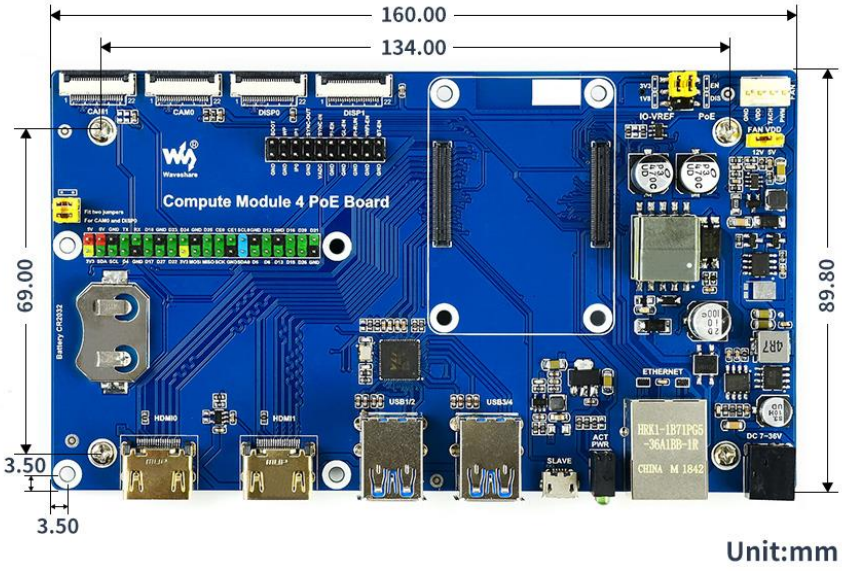
fire, smoke, humidity, temperature, water, and other things that have a specific type of sensor, like Magnetic for a Compass, and tracking the Sun and Moon, these are the things I want to teach in the coming lessons on how I built this Remote Sensor Machine.

The Machine will have a Touchscreen, Keyboard and Mouse, as well as wired and wireless internet, and cellular communications, all the Computer Hardware is in one box that acts like a patch panel where you have PoE cable connectors that go to all the remote sensors, and all the Remote Sensors are in their own box, and it has a PoE cable connector on it, and that provides a wired Connection with Power, sure we can go wireless, but if we cannot power the device, or have to change batteries all the time, what is the point, but it will also have a wireless version of it, and I might make them in the same unit, as a redundant feature in case of theft we can track all the parts with a Smart Phone app, and all the boxes will have built in UPS for uninterruptable power supply.

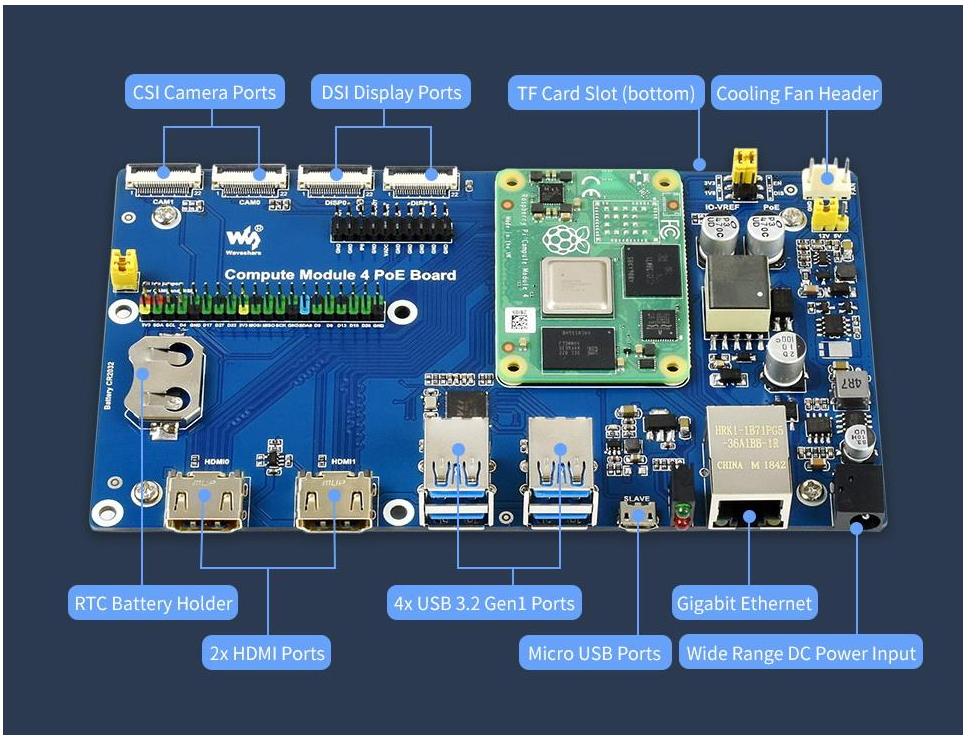
Unlike most Pi Projects, I will be using a Pi Compute Module 4 that will plug into our Motherboard.

As you can see, this is not a typical Pi Card, it has no IO only a connector on the back of it.

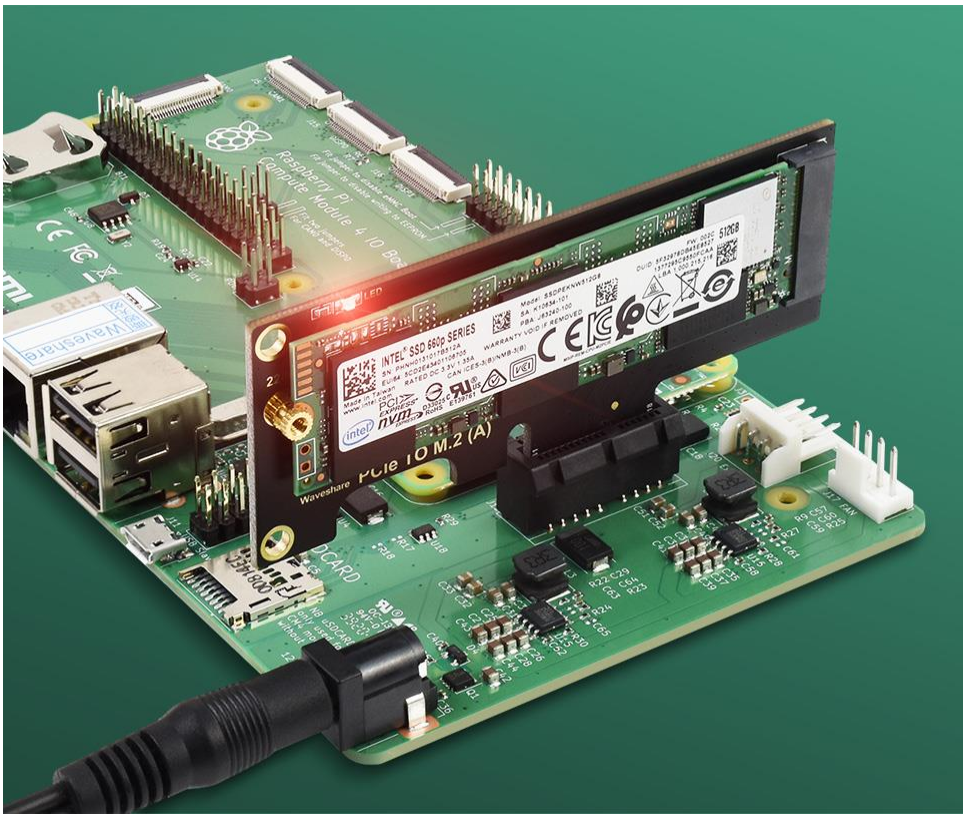
It plugs into the Motherboard where it gets all its IO from



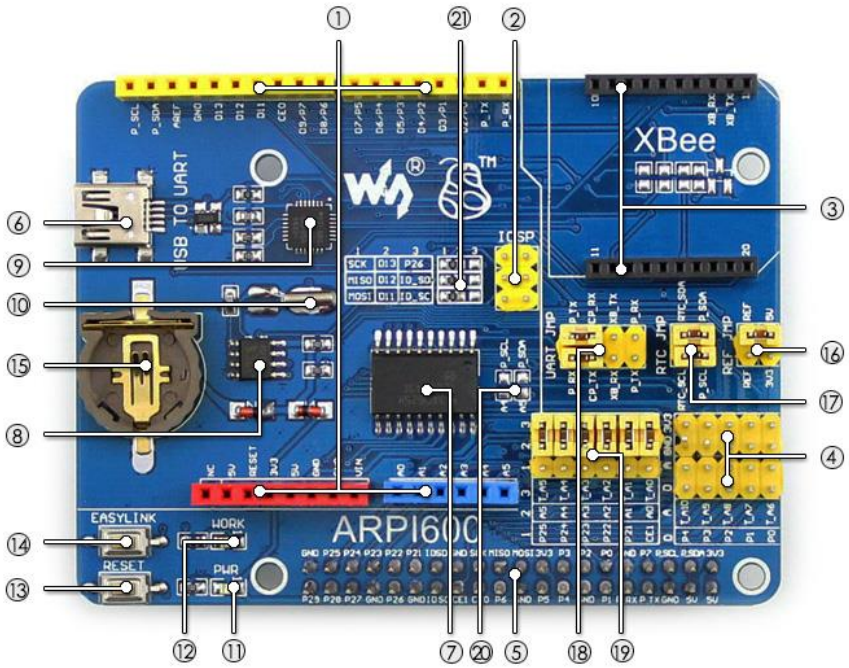
The reason for using this Motherboard called a Compute Module 4 design is simple, it is very powerful when it comes to building a Remote Sensor Package, and once we mount the Pi to it, it will be able to run Linux, and a Web Server that we can access from the Internet via a Cellular Device also from Waveshare.



Next, we need to add an SSD for storage space to save all our remote sensor data.



The design of the Compute Module 4 allows us to keep extending it, so we will add the Adapter Board for the Arduino & Raspberry Pi



1. Arduino connector: for connecting Arduino shields
2. ICSP interface: Arduino ICSP
3. XBee connector: for connecting XBee communication modules
4. Sensor interface: for connecting sensors
5. Raspberry Pi connector: for connecting Raspberry Pi
6. USB TO UART
7. TLC1543: AD converter
8. PCF8563: RTC
9. CP2102
10. 32.768KHz crystal: for RTC
11. Power indicator
12. XBee state LED
13. XBee and Arduino interface RESET button
14. XBee EASYLINK button
15. RTC battery holder: for CR1220 button battery
16. TLC1543 reference voltage configuration jumper
17. RTC jumper
18. UART jumper

    - when connecting P_RX and CP_TX, P_TX and CP_RX respectively, USB TO UART is connected to Raspberry Pi serial port
    - when connecting XB_RX and CP_TX, XB_TX and CP_RX respectively, USB TO UART is connected to XBee serial port
    - when connecting XB_RX and P_TX, XB_TX and P_RX respectively, Raspberry Pi serial port is connected to XBee serial port

19. Arduino AD selection jumper

    - short 2 and 3: Arduino A0-A5 as AD input
    - short 1 and 2: Arduino A0-A5 as digital control

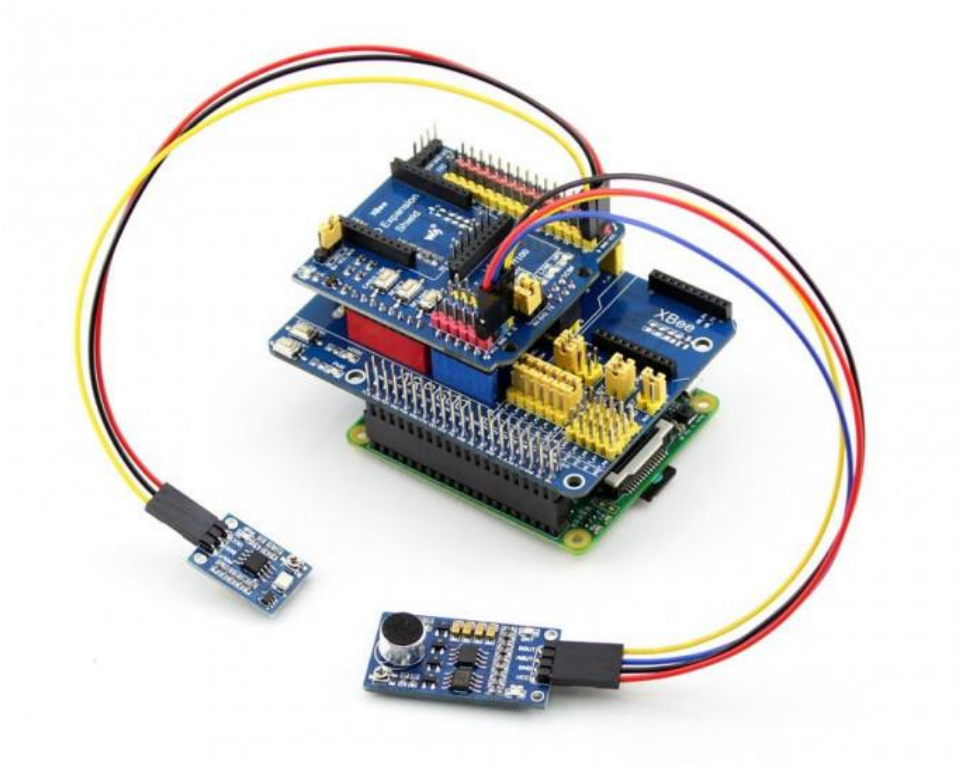
20. Arduino I2C selection jumper

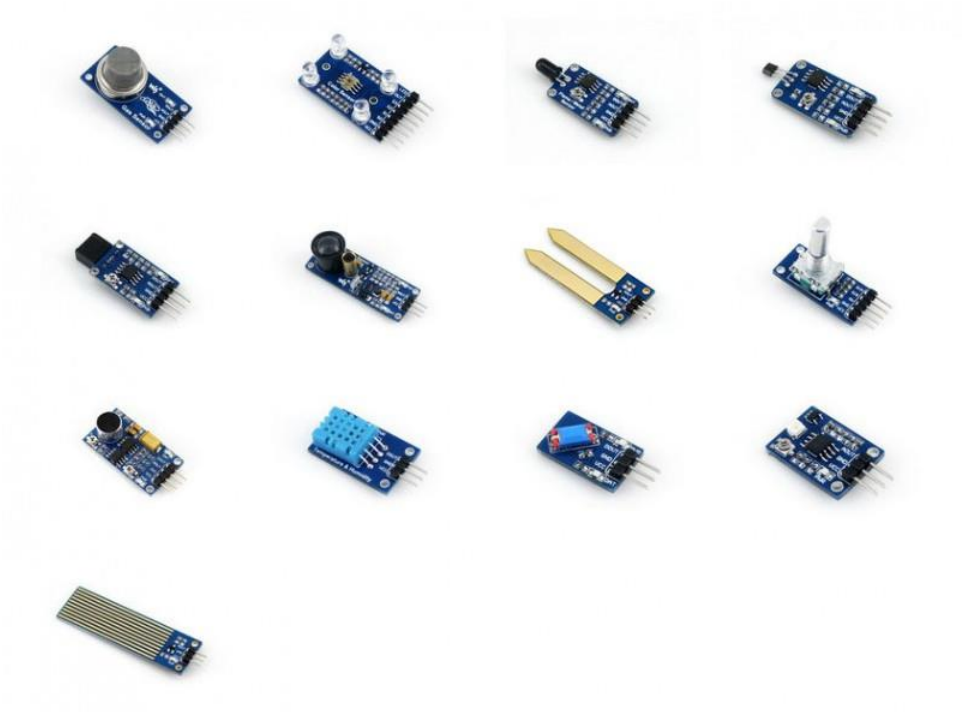
    - short the jumper: Arduino A4-A5 as I2C control (the A4-A5 of Arduino AD selection jumper should be opened)

21. Arduino SPI selection jumper

    - short 1 and 2: Arduino D11-D13 as SPI control (default)
    - short 2 and 3: Arduino D11-D13 as digital control

This will allow us to interface with all the various Remote Sensores we have





Waveshare has many Sensors, and I will dedicate one PDF and video to each sensor, so I am not rushing over many of them, and focusing only one type of Sensor, and I will cover more than the 13 basic ones.

We need a way to interface with it, and that is going to require a Cellular device, and therefore I am going with Waveshare, they have many devices to choose from https://www.waveshare.com/product/iot-communication/long-range-wireless/4g-gsm-gprs.htm like the Global Version https://www.waveshare.com/product/iot-communication/long-range-wireless/4g-gsm-gprs/sim7600g-h-4g-hat.htm

This will give us 4G / 3G / 2G / GSM / GPRS / GNSS HAT for Raspberry Pi, LTE CAT4, this is the Global Version of the SIM-7600G-H and requires a SIM card like any other Cellular Device.

This will attach to a Touchscreen for ease of use, and it will have a Wireless Keyboard and Mouse, so we can get Linux up and running, and make a web server that feeds the real-time data, so I have a lot of code to write, and will store it all under the code folder at GitHub for this Remote Sensors project.