Understanding How to Make a Sensor Work and Make the Soil NPK Sensor Work.

To Understand the Working of the Sensor, we started with temperature and humidity sensor which is the cheap Modbus sensor available.

We explored and understood the use of multiple software’s which will be used to communicate with the Modbus senor like Modbus Poll, Hercules Setup, 485 V3.0 and CAS Modbus Scanner. Then we picked the CAS Modbus Scanner software as the standard to follow.

Then after understanding the sensor is working using the software’s, we started to make a code which can also help us to communicate with the Modbus sensor. We came across 2 python libraries like pySerial and pyModbus when trying to make a code and when tried to code by using pySerial we were unable to read the complete response frame sent by the sensor, and when we used the pyModbus library we finally were able to read the whole response frame and convert the frame into integers. So, we came up with a code by using pyModbus which helps us to communicate with the sensor.

Then after clear understanding of how to communicate with the working sensor, we moved on to the NPK sensor.

The soil NPK Sensor detects the content of Nitrogen, Phosphorus, Potassium in the soil. Measuring these Macro-Nutrients helps us in determining the fertility of the soil. This sensor has high measurement accuracy, fast response speed, and it can be controlled with any microcontroller.

We have 2 possibilities to get the values out of this sensor.

1. **Using a RS485 to USB Converter and a PC**
2. **Using Microcontroller**

Using RS485 to USB Converter and a PC

To connect it with a RS485 to USB Converter and a PC. The connections will be as follows.

- Connect the Blue and Yellow wires from the Sensor to the RS485 to USB Converter
- Now, the black and brown wires to the female DC power jack adaptor.
- Connect the female DC power jack adaptor to the 12V power supply.
- Plug-in the RS485 to USB Converter to you PC

After all the Connections, Now Open your PC and install CH340 driver if it is not already installed in your device. This is needed to communicate through the USB converter.

Note down your Port Number

Now, we can get to know the sensor working in 2 ways i.e.

1. **Implementation Using Software.**
2. **Implementation Using Code.**

Implementation Using Software

Use the CAS Modbus Software to check whether the sensor is working or not. In CAS Modbus Software you need to follow the below procedure and set the connection.

1. After opening the CAS Modbus Software go to the left panel and when you right click there ‘Add Connection’ will appear and then click it.

2. Then select your Serial Port and set these conditions i.e Baud Rate – 4800, Data Bits – 8, Stop Bits – 1, Parity – None, and then click Add serial connection.
3. Now again right click on the serial port and then you should get an option ‘Add Device’, after clicking on it you will have to set the slave ID. Set it to 1 and click on Add Device.
4. Again right clicking on ‘Device : 1’ you will have to choose ‘Add Request’ where you need to choose your request function. Click on ’04 Read Holding Registers (4xxxx) and click on preferred offset and length following the documentation and then finally click on Add Request.
5. Now you are all set to read the N, P, K values from the sensor. Just click on poll and then your values will be presented in the right box in different formats.

Implementation Using Code

After following the same connections given above.

The code for the Implementation of Soil NPK Sensor is this:

```
import time
from pymodbus.client.sync import ModbusSerialClient

client = ModbusSerialClient(method='rtu', port='COM4', timeout=1, stopbits = 1, bytesize = 8,  parity='N', baudrate= 9600)
client.connect()

while True:
    Ph = client.read_holding_registers(address=6, count=1, unit=1);
    M = client.read_holding_registers(address=18, count=1, unit=1);
    T = client.read_holding_registers(address=19, count=1, unit=1);
    E = client.read_holding_registers(address=21, count=1, unit=1);
    N = client.read_holding_registers(address=30, count=1, unit=1);
    P = client.read_holding_registers(address=31, count=1, unit=1);
    K = client.read_holding_registers(address=32, count=1, unit=1);
    print(Ph.registers[0]/100.0);
    time.sleep(1)
    print(M.registers[0]/10.0);
    time.sleep(0.5)
    print(T.registers[0]/10.0);
    time.sleep(1)
    print(E.registers[0]);
    time.sleep(1)
    print(N.registers[0]);
    time.sleep(0.5)
    print(P.registers[0]);
    time.sleep(0.5)
    print(K.registers[0]);
    time.sleep(2)
```

Here we have used time library of python and pyModbus library.

Use this code using command prompt of the PC by typing **‘python filename’** then you should be getting the results if the sensor is working.