## ****Android Cell phone Application for the Soil NPK Sensor:****

The Android cell phone application used for monitoring the Soil NPK Sensor is designed in Android Studio. This is the same application I designed in my previous tutorial. So, I highly recommend reading this [tutorial](https://www.electroniclinic.com/how-to-create-android-app-for-arduino-sensor-monitoring-over-bluetooth/), if in case you want to design your own android cell phone application for monitoring different types of sensors, or else you can download the [apk file](https://www.electroniclinic.com/wp-content/uploads/2019/08/sensor-app-apk-file.zip).

Before, you start the programming, first of all, make sure you download all the necessary [libraries](https://www.electroniclinic.com/arduino-libraries-download-and-projects-they-are-used-in-project-codes/). The purpose of the following program is to read the Nitrogen, Phosphorus, and Potassium values from the Soil NPK sensor and then display the values on the Oled display module and also on the Android cell phone application.

## ****Modbus Command for NPK Sensor****

The information I am about to share with you guys is really important, let me say this one more time, it’s really important. Because once you understand the frame structures then programming is just a piece of cake. So far you know, the NPK Sensor supports Modbus communication and this is the reason the Modbus-RTU Communication protocol is adopted, let’s take a look at its format.

**Initial structure ≥ 4 bytes of time**

**Address code = 1 byte**

The address code is basically the transmitter address and it is unique in the entire communication network, the factory default value is 0x01.

**Function Code = 1 byte**

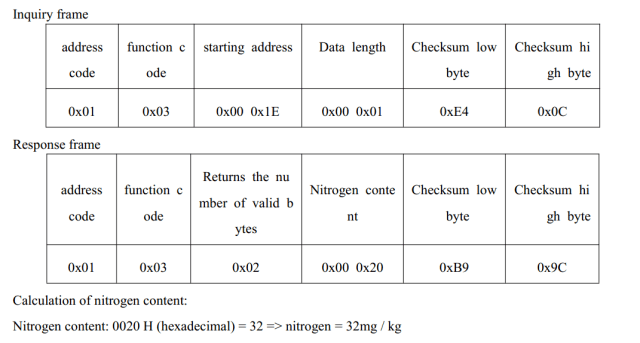
**Data area = N bytes**

This is the specific communication data.

**Error check = 16-bit RCR**

**Ending structure ≥ 4 bytes of time**

Below are the Host inquiry and Slave response frame structures. It’s simple, to read data from the NPK sensor we simply send the Host Inquire frame, and then the NPK sensor sends back the Slave response consisting of the desired data.

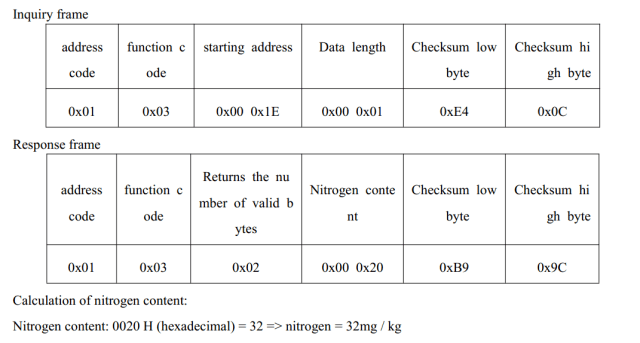
[](https://i1.wp.com/www.electroniclinic.com/wp-content/uploads/2021/03/reading-address-of-nitrogen.png?ssl=1)

As discussed earlier, on a single bus multiple devices can be connected, so this way the master can communicate with multiple slave devices. Now to avoid any confusion, this is the reason the inquire and response frames are provided with the Address code. So, we simply use the address of the device we want to communicate with, and it will have no effect on the other devices. So the transmitter will send data to that specific NPK sensor and then receive data.

As the NPK Sensor is for Nitrogen, Phosphorus, and Potassium, so it means we will be reading these three different values from the NPK Sensor. For each of these “N, P, K” we will send an inquiry frame having different starting addresses.

**Let’s start with the Nitrogen**

To read the Nitrogen value from the NPK Sensor you will need to send the following Inquiry frame, and then sensor then replies with the Response frame.

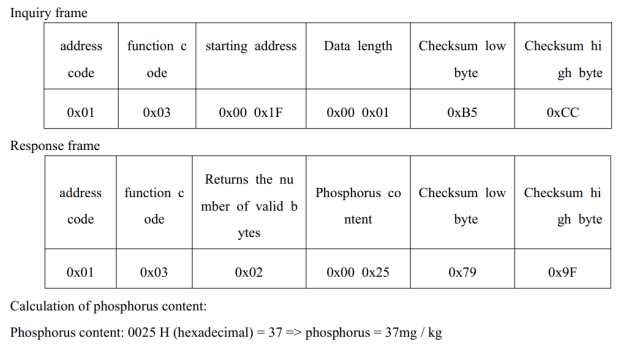
[](https://i1.wp.com/www.electroniclinic.com/wp-content/uploads/2021/03/reading-address-of-nitrogen.png?ssl=1)

So,

**Nitrogen** = 0x01, 0x03, 0x00, 0x1E, 0x00, 0x01, 0xE4, 0x0C

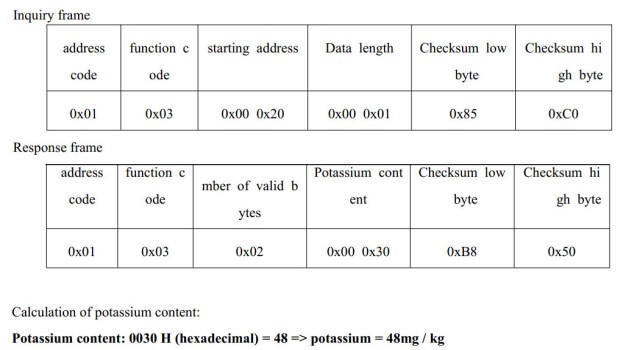
So, our inquire frame should have all the above values. In programming what we can do, is to make an array having all these values, which I will explain in the code given below.

**For the Phosphorus:**

[](https://i2.wp.com/www.electroniclinic.com/wp-content/uploads/2021/03/Reading-address-of-Phosphorus.png?ssl=1)

**Phosphorus** = 0x01, 0x03, 0x00 0x1F, 0x00, 0x01, 0xB5, 0xCC

**For the Potassium:**

[](https://i2.wp.com/www.electroniclinic.com/wp-content/uploads/2021/03/Reading-address-of-potassium.jpg?ssl=1)

**Potassium** = 0x01, 0x03, 0x00, 0x20, 0x00, 0x01, 0x85, 0xC0

So, now to read the Nitrogen, Phosphorus, and Potassium contents of the soil, we will need to send the following command one by one using Arduino. This is what we are going to do next.

**Nitrogen** = 0x01, 0x03, 0x00, 0x1E, 0x00, 0x01, 0xE4, 0x0C

**Phosphorus** = 0x01, 0x03, 0x00 0x1F, 0x00, 0x01, 0xB5, 0xCC

**Potassium** = 0x01, 0x03, 0x00, 0x20, 0x00, 0x01, 0x85, 0xC0