

GPS/GLONASS Modbus TCP server на ПЛК NEON.

Природа кейса

Один из наших клиентов спросил – можно ли решить задачу определения времени переходов между точками GPS координат. Логично, лучше всего дела с определением позиционирования обстоят у смартфонов и планшетов. NEON, для Андроид устройств, должен с ней справиться. Задача совсем не про АСУ ТП, но имеет интересное решение и выполняет желание клиента.

Идея

Решить эту задачу на NEON можно очень быстро. И даже получить кое-какие дивиденды:

- NEON это служба, которая стартует с Андроид и крутится в фоновом режиме. Она полностью испытана и работает стабильно. Т.о., и приложение, разработанное пользователем, получает эти свойства.

- Modbus TCP встроен в NEON, и, в случае с режимом «Server», его даже кодить не придется. Просто установить галку задать адреса.

- Вторая часть, теоретическая, для тех, кто требует больше. Выводим это все в облако с web доступом и разграничением прав и уровней доступа. Ни строчки кода, связь с облаком в NEON с «рождения». В облаке всё через диалог с мышкой в руках.

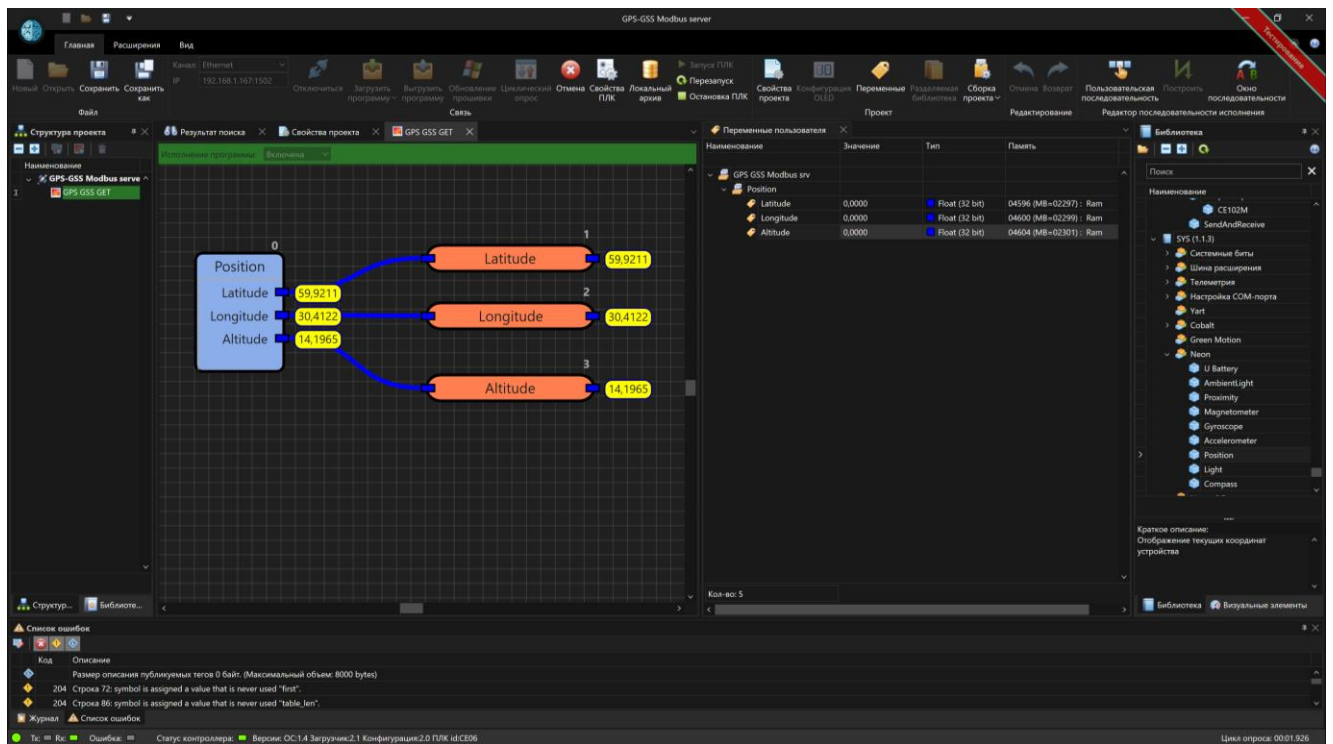
- + Реализуем не сложный алгоритм определения нахождения в радиусе точки GPS, добавляем нужные массивы точек маршрута, времени прохождения.... Даем доступ к этому из web облака.

Инструмент

Выбираю простой инструмент, который позволит, без глубокого погружения в железо и программирование, реализовать идею. Устанавливаю на свой планшет HUAWEI под Андроид (HarmonyOS 2.0.0) контроллер NEON. На планшете служба позиционирования вычисляет данные на основе показаний датчиков GPS GLONASS и сотовой сети.

Процесс

Создаю нужные переменный Latitude, Longitude, Altitude. Устанавливаю к ним доступ по Modbus (кнопка в окне переменных «Ручной расчет адреса», порт для NEON 1502. Создаю лист программы и вытаскиваю блок из Position из библиотеки SYS\Neon\ и подключаю к нему эти переменные. Загружаю в планшет и получаю картину:



В общем, сам проект Modbus GPS GLONASS сервера я сделал быстрее, чем Вы бы установили ПО для разработки Android приложений. Сделать «+» это еще не более 2 часов, с web, и всеми прелестями.

Для определения того, что устройство приблизилось к GPS можно переделать код для функционального блока проекта в YART Studio:

```
double distance(double lat1, double lon1, double lat2, double lon2, char unit)
```

```
{
    double theta, dist;
    if ((lat1 == lat2) && (lon1 == lon2)) {
        return 0;
    }
    else {
        theta = lon1 - lon2;
        dist = sin(deg2rad(lat1)) * sin(deg2rad(lat2)) + cos(deg2rad(lat1)) * cos(deg2rad(lat2)) *
cos(deg2rad(theta));
        dist = acos(dist);
        dist = rad2deg(dist);
        dist = dist * 60 * 1.1515;
        switch(unit) {
```

```

    case 'M':
        break;
    case 'K':
        dist = dist * 1.609344;
        break;
    case 'N':
        dist = dist * 0.8684;
        break;
}
return (dist);
}
}
/*:.....*/
/*:: This function converts decimal degrees to radians :*/
/*:.....*/
double deg2rad(double deg) {
    return (deg * M_PI / 180);
}
/*:.....*/
/*:: This function converts radians to decimal degrees :*/
/*:.....*/
double rad2deg(double rad) {
    return (rad * 180 / M_PI);
}

```

Возможно, совсем скоро, такой блок появится в нашей библиотеке.