

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное
образовательное учреждение высшего образования
«Южно-Уральский государственный университет
(национальный исследовательский университет)»
Высшая школа электроники и компьютерных наук
Кафедра «Электронные вычислительные машины»

РАЗРАБОТКА СЕРВЕРНОЙ ЧАСТИ СИСТЕМЫ ПОЗИЦИОНИРОВАНИЯ ВНУТРИ ПРОИЗВОДСТВЕННЫХ ПОМЕЩЕНИЙ

Автор работы:

Студент группы КЭ-406

Рябухин И.А.

Научный руководитель:

Доцент каф. ЭВМ

к.т.н. Шабуров П.О.

Актуальность

Определение положений подвижных объектов внутри помещений – одно из возможных средств оптимизации производственных и логистических процессов предприятия.

Предприятием-заказчиком была выбрана система «Marvelmind», однако её проприетарное программное обеспечение не удовлетворяет требованиям производства в полной мере, поэтому возникла необходимость в разработке собственного программного обеспечения.

Цели и задачи ВКР

Целью выпускной квалификационной работы является проектирование и реализация программного обеспечения, соответствующего всем функциональным требованиям, предоставленным предприятием ООО «ДСТ-УРАЛ».

В данной работе были поставлены и решены следующие задачи:

- при наличии аналогов проанализировать их функционал;
- выбрать язык программирования, среду и методы разработки;
- на основе технического задания предприятия спроектировать программное обеспечение, описав модули;
- реализовать функциональную версию программного обеспечения;
- провести тестирование и отладку программного продукта.

Обзор аналогов

Проприетарное программное обеспечение не удовлетворяет заказчика по следующим критериям:

- невозможность интеграции программного продукта в процессы производства;
- отсутствие разделения на серверную и клиентские части;
- нестабильная работа программного обеспечения;
- невозможность одновременного запуска нескольких экземпляров ПО;
- исходный код программного продукта недоступны, что делает невозможным редактуру приложения для нужд предприятия.

Инструменты и методы разработки

Язык разработки – C#



Платформа разработки – .NET 5



Среда разработки – Microsoft Visual Studio
2019

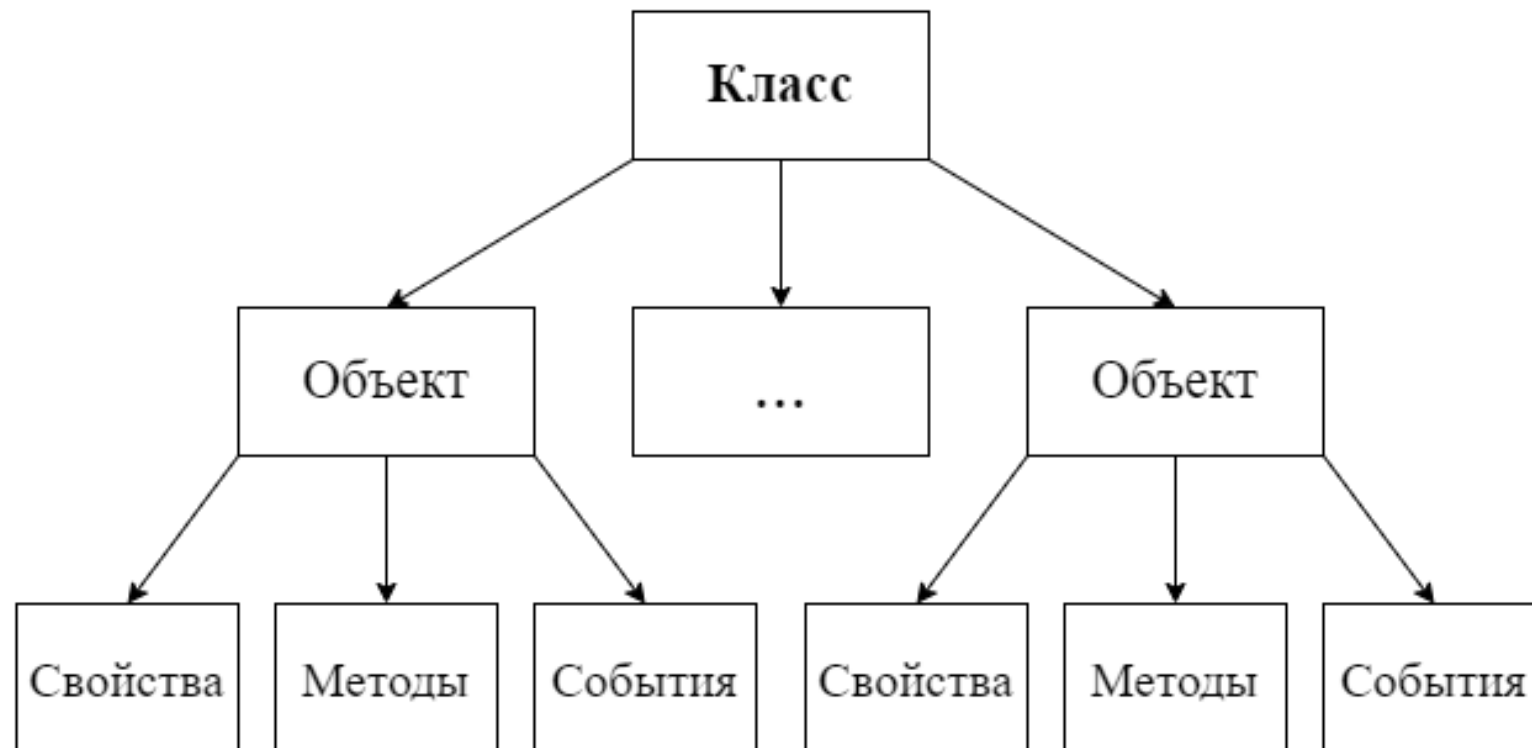


Основные требования к системе

- возможность одновременного подключения нескольких клиентских приложений;
- возможность установления соединения с аппаратной частью системы позиционирования посредством двух каналов связи: Wi-Fi и UART;
- возможность настройки аппаратной части системы позиционирования согласно официальной документации системы позиционирования;
- вывод всех полученных и отправленных пакетов данных посредством консоли для целей отладки и контроля за функционированием системы;
- вывод данных о текущих координатах всех отслеживаемых объектов и стационарных маяков посредством консоли;
- программа должна иметь модульную структуру для последующей интеграции в информационно-вычислительную систему производства;
- иметь совместимость с серверными платформами семейств Windows и Unix;
- все вычисления должны носить локальный характер.

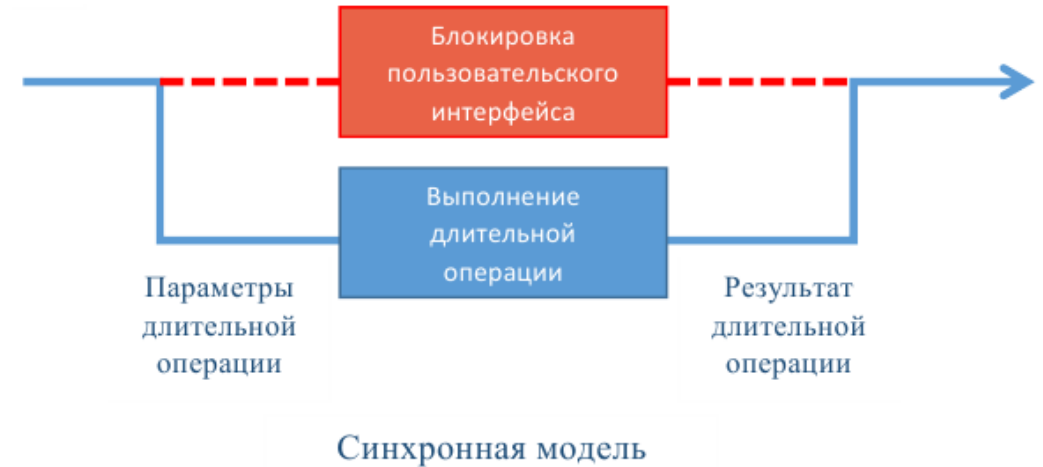
Архитектура решения

Для разработки серверного приложения были использованы объектно-ориентированная и событийная парадигмы. Их обобщенное представление изображено на схеме.



Архитектура решения

При разработке также было применено асинхронное программирование, позволяющее выполнять обработку данных параллельно, не блокируя остальные потоки. Это обеспечивает высокое быстродействие системы.



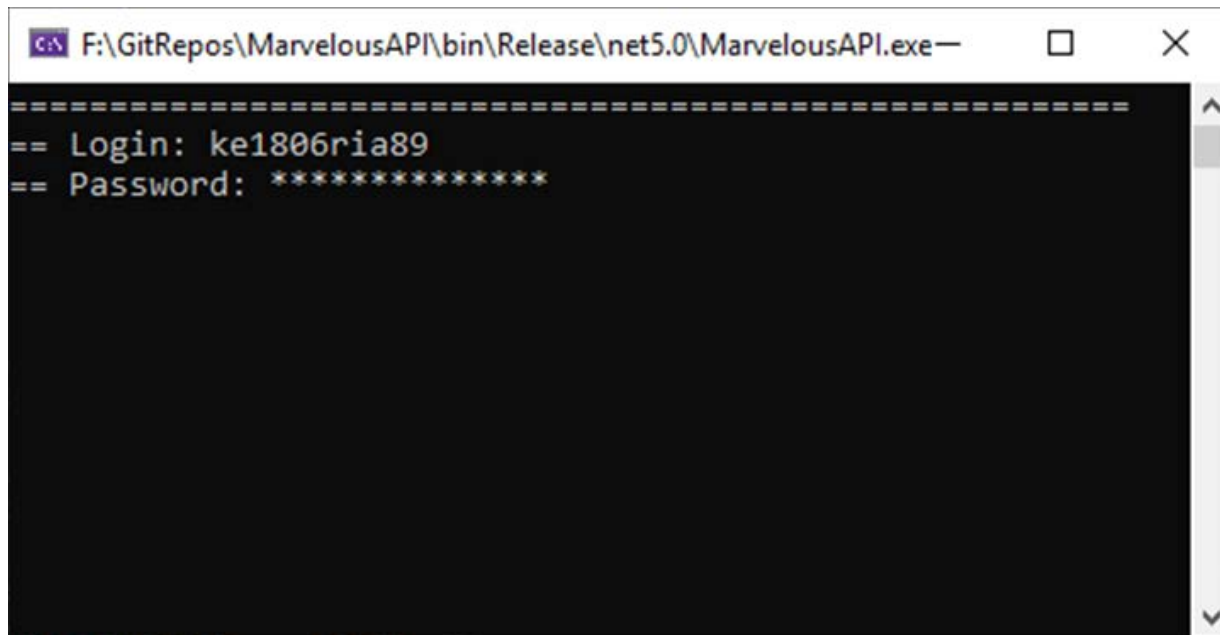
Архитектура решения

Структура разработанного программного решения содержит следующие модули:

- основной модуль программы;
- модуль описания маяков;
- модуль описания модема;
- модуль сетевого взаимодействия;
- модуль интерпретации пакетов аппаратной части;
- модуль интерпретации сетевых данных;
- модуль связи с аппаратной частью посредством UART.

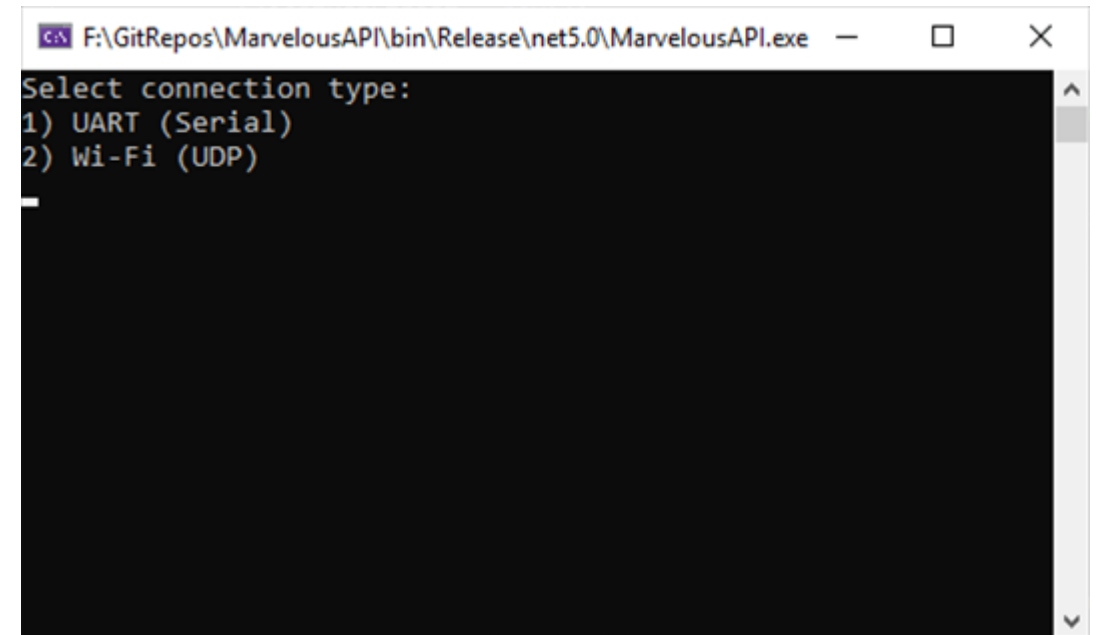
Реализация

После запуска серверного приложения пользователь обязан авторизоваться



```
C:\> F:\GitRepos\MarvelousAPI\bin\Release\net5.0\MarvelousAPI.exe

=====
== Login: ke1806ria89
== Password: *****
```



```
C:\> F:\GitRepos\MarvelousAPI\bin\Release\net5.0\MarvelousAPI.exe

Select connection type:
1) UART (Serial)
2) Wi-Fi (UDP)
```

Реализация

Меню для двух подключений: посредством UDP, либо UART

```
F:\GitRepos\MarvelousAPI\bin\Release\net5.0\MarvelousAPI.exe
Main serial menu:
1) List all beacons
2) Get modem firmware
3) Get last coordinates
4) Serial debugging
5) UDP debugging
6) Wake up beacons
7) Sleep beacons
8) Re-scan for available beacons
9) Reselect COM port
0) Change connection method
-
```

```
F:\GitRepos\MarvelousAPI\bin\Release\net5.0\MarvelousAPI.exe
Main UDP menu:
1) List all beacons
2) Client UDP debug
3) Modem UDP debug
4) Full UDP debug
5) List connected clients
0) Change connection method
-
```

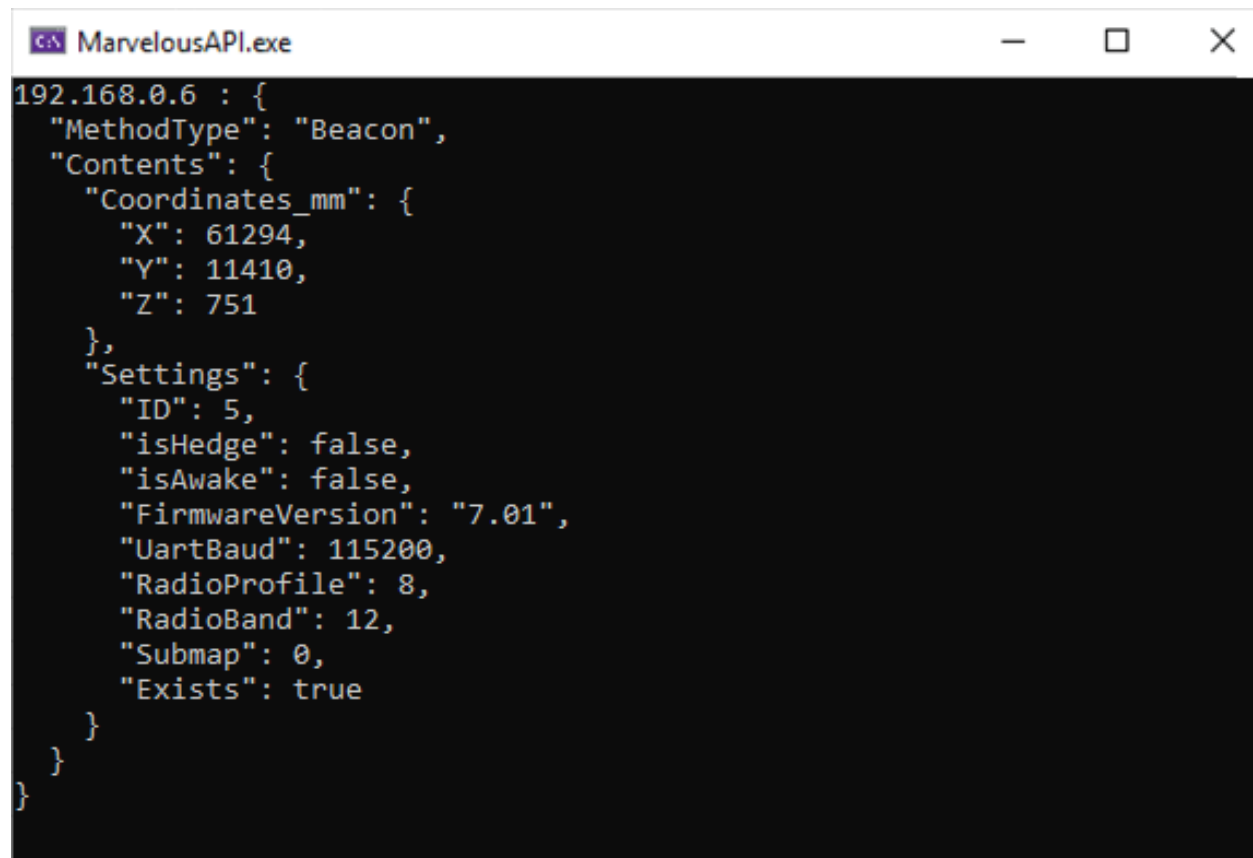
Реализация

Интерпретированный пакет данных, полученный по UART

```
F:\GitRepos\MarvelousAPI\bin\Release\net5.0\MarvelousAPI.exe
RECEIVED 0x47, 0x0011: hedge_addr=5, coord_avl=False, time_stamp=26275947, X=63373, Y=15258, Z=4294966331, CRC VALID
RECEIVED 0x47, 0x0011: hedge_addr=5, coord_avl=False, time_stamp=26276948, X=63373, Y=15258, Z=4294966331, CRC VALID
RECEIVED 0x47, 0x0011: hedge_addr=5, coord_avl=False, time_stamp=26277955, X=63373, Y=15258, Z=4294966331, CRC VALID
RECEIVED 0x47, 0x0011: hedge_addr=5, coord_avl=False, time_stamp=26278950, X=63373, Y=15258, Z=4294966331, CRC VALID
RECEIVED 0x47, 0x0011: hedge_addr=5, coord_avl=False, time_stamp=26280953, X=63373, Y=15258, Z=4294966331, CRC VALID
RECEIVED 0x47, 0x0011: hedge_addr=5, coord_avl=True, time_stamp=26281954, X=60983, Y=11741, Z=4294964644, CRC VALID
RECEIVED 0x47, 0x0011: hedge_addr=5, coord_avl=False, time_stamp=26283956, X=60983, Y=11741, Z=4294964644, CRC VALID
RECEIVED 0x47, 0x0011: hedge_addr=5, coord_avl=True, time_stamp=26284956, X=66283, Y=19387, Z=4294965522, CRC VALID
RECEIVED 0x47, 0x0011: hedge_addr=5, coord_avl=False, time_stamp=26285957, X=66283, Y=19387, Z=4294965522, CRC VALID
RECEIVED 0x47, 0x0011: hedge_addr=5, coord_avl=False, time_stamp=26286959, X=66283, Y=19387, Z=4294965522, CRC VALID
RECEIVED 0x47, 0x0011: hedge_addr=5, coord_avl=False, time_stamp=26287965, X=66283, Y=19387, Z=4294965522, CRC VALID
RECEIVED 0x47, 0x0011: hedge_addr=5, coord_avl=False, time_stamp=26288961, X=66283, Y=19387, Z=4294965522, CRC VALID
RECEIVED 0x47, 0x0011: hedge_addr=5, coord_avl=False, time_stamp=26289962, X=66283, Y=19387, Z=4294965522, CRC VALID
RECEIVED 0x47, 0x0011: hedge_addr=5, coord_avl=False, time_stamp=26290963, X=66283, Y=19387, Z=4294965522, CRC VALID
RECEIVED 0x47, 0x0011: hedge_addr=5, coord_avl=False, time_stamp=26292970, X=61240, Y=11240, Z=4294964660, CRC VALID
RECEIVED 0x47, 0x0011: hedge_addr=5, coord_avl=False, time_stamp=26293967, X=61240, Y=11240, Z=4294964660, CRC VALID
RECEIVED 0x47, 0x0011: hedge_addr=5, coord_avl=True, time_stamp=26294967, X=61276, Y=11564, Z=4294964274, CRC VALID
RECEIVED 0x47, 0x0011: hedge_addr=5, coord_avl=True, time_stamp=26295969, X=61294, Y=11410, Z=4294964373, CRC VALID
RECEIVED 0x47, 0x0011: hedge_addr=5, coord_avl=False, time_stamp=26296969, X=61294, Y=11410, Z=4294964373, CRC VALID
RECEIVED 0x47, 0x0011: hedge_addr=5, coord_avl=False, time_stamp=26298972, X=61294, Y=11410, Z=4294964373, CRC VALID
RECEIVED 0x47, 0x0011: hedge_addr=5, coord_avl=False, time_stamp=26299973, X=61294, Y=11410, Z=4294964373, CRC VALID
RECEIVED 0x47, 0x0011: hedge_addr=5, coord_avl=False, time_stamp=26300974, X=61294, Y=11410, Z=4294964373, CRC VALID
RECEIVED 0x47, 0x0011: hedge_addr=5, coord_avl=False, time_stamp=26301976, X=61294, Y=11410, Z=4294964373, CRC VALID
RECEIVED 0x47, 0x0011: hedge_addr=5, coord_avl=False, time_stamp=26302982, X=61294, Y=11410, Z=4294964373, CRC VALID
RECEIVED 0x47, 0x0011: hedge_addr=5, coord_avl=False, time_stamp=26303977, X=61294, Y=11410, Z=4294964373, CRC VALID
RECEIVED 0x47, 0x0011: hedge_addr=5, coord_avl=False, time_stamp=26304978, X=61294, Y=11410, Z=4294964373, CRC VALID
RECEIVED 0x47, 0x0011: hedge_addr=5, coord_avl=False, time_stamp=26306981, X=61294, Y=11410, Z=4294964373, CRC VALID
```

Реализация

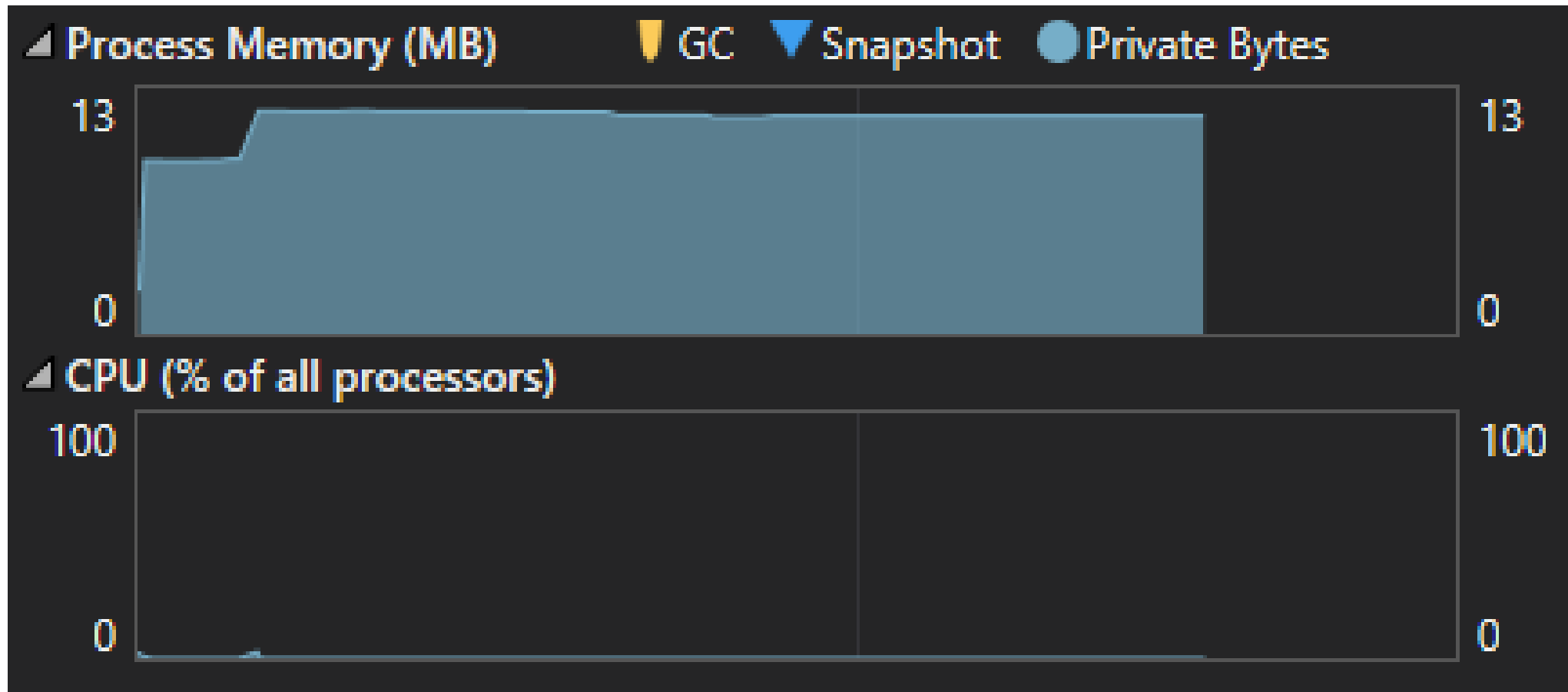
Структура пакета данных в формате JSON



```
MarvelousAPI.exe
192.168.0.6 : {
  "MethodType": "Beacon",
  "Contents": {
    "Coordinates_mm": {
      "X": 61294,
      "Y": 11410,
      "Z": 751
    },
    "Settings": {
      "ID": 5,
      "isHedge": false,
      "isAwake": false,
      "FirmwareVersion": "7.01",
      "UartBaud": 115200,
      "RadioProfile": 8,
      "RadioBand": 12,
      "Submap": 0,
      "Exists": true
    }
  }
}
```

Тестирование

График использования программой процессора и памяти



Тестирование

Функциональное тестирование подразумевает проведение проверки реализованного программного продукта на соответствие всем функциональным требованиям, установленным заказчиком.

В ходе проведения тестов в полноценном производственном помещении, ответственный представитель предприятия проверил и одобрил весь реализованный функционал приложения, а также отметил стабильную работу программного продукта, легкость его установки, настройки и пользования.

Заключение

В рамках выпускной квалификационной работы была реализована серверная часть программного обеспечения для системы позиционирования внутри производственных помещений.

Разработанный программный продукт соответствует всем требованиям предприятия-заказчика. После запуска серверное приложение устанавливает соединение с системой «Marvelmind» и доступными клиентскими приложениями. После чего возможна настройка данных соединений, а также реализуется передача всех требуемых данных клиентским приложениями. Модульная структура итогового программного решения позволит легко интегрировать его в производственные и логистические процессы предприятия.

Благодарю за внимание!