



Дирекция ООО «Инженерная Компания», департамент
Цифровых Мультидисциплинарных Технологий»

г. Москва, Старокалужское ш.,
д.62, с1, к9, офис 406;
ст. м. Калужская, Воронцовская

+7(916) 792-4050

polator.ru

job@polator.ru



ТЕСТОВОЕ ЗАДАНИЕ

На роль «Прикладной программист C++ (QT)»,
участника разработки продукта ПОЛАТОР

1

ВВОДНОЕ СЛОВО

Уважаемый кандидат, здравствуйте!

Мы не проектируем IoT-ы и никогда их не будем изготавливать. Мы занимаемся разработкой программного обеспечения. Оно называется ПОЛАТОР – программно-определяемая лаборатория. С одной стороны, это наборы инструментов поддержки исследований. С другой - программное обеспечение по моделированию и быстрому прототипированию научных моделей и инженерных программно-аппаратных прототипов.

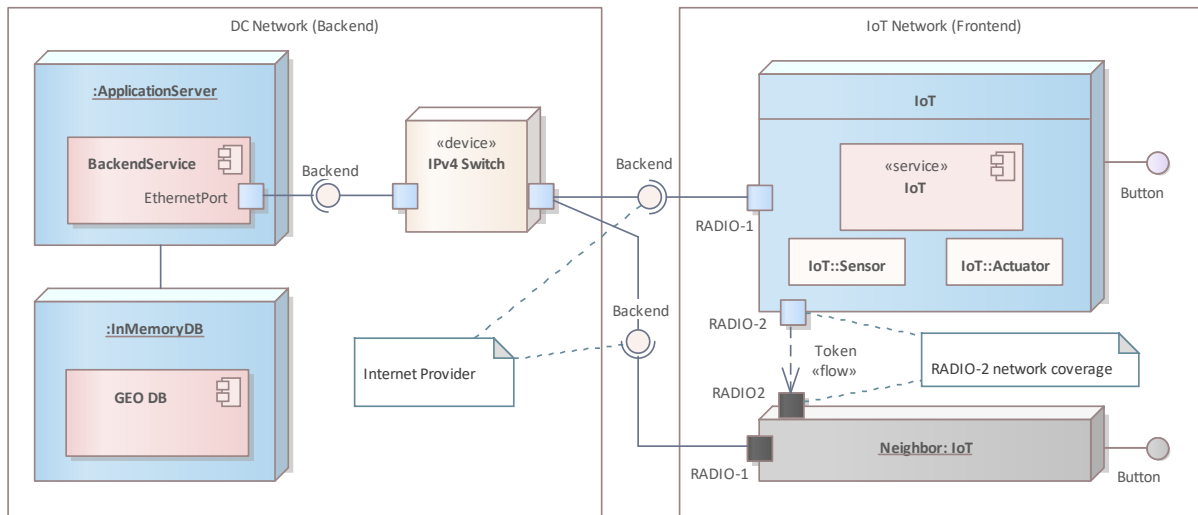
Мы постарались сделать наше задание не формальным, а интересным, живым. Мы постарались его выразить в виде игровой ситуации, похожей на реальную задачу, Однако не надо смотреть на настоящее задание как на рабочий проект – он им не является. Выполнение задания необходимо нам только для вашей оценки.

Мы ожидаем от вас демонстрацию ваших профессиональных навыков, которые для нас важны в нашем продукте.

Заранее благодарим за участие,
успехов вам!

Просим вас выполнить задание, запаковать результат в ZIP-архив, где имя файла в виде **Фамилия.zip** и отправить его по адресу **job@polator.ru**. Можно задавать вопросы неограниченное количество раз.

Просим вести переписку с сохранением истории, т.е. начиная с письма-приглашения, исключительно используя «Отправить Всем» (Reply All).

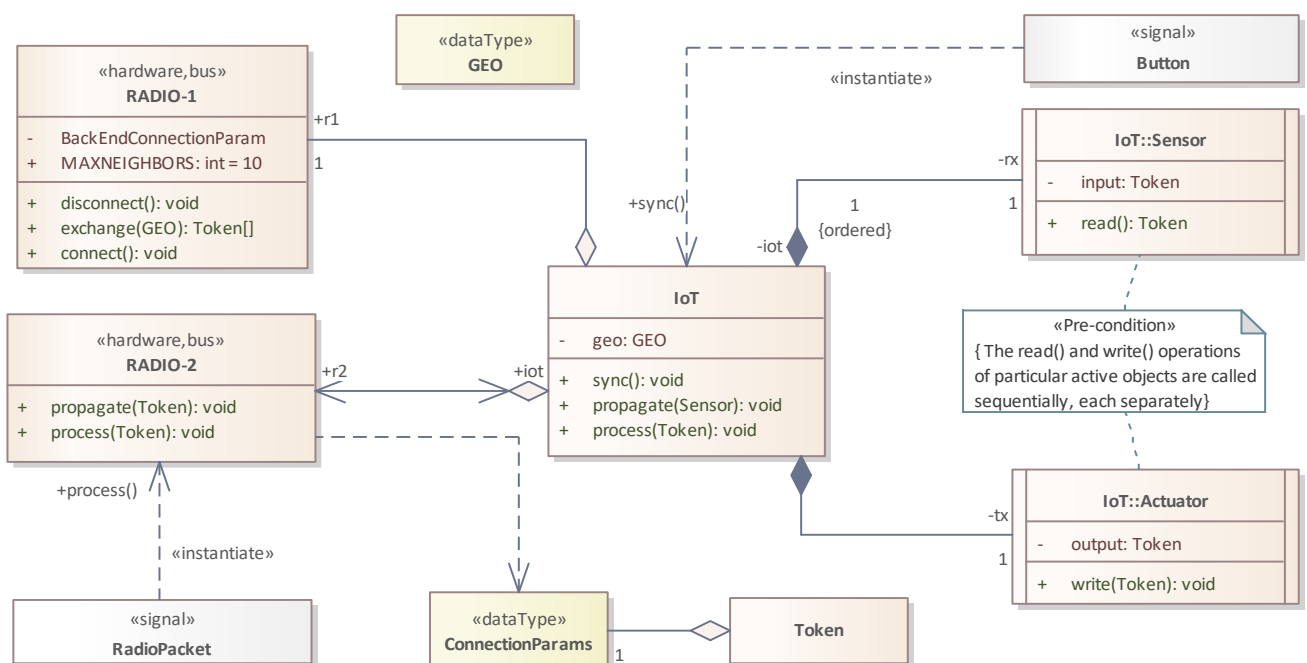


Backend не является предметом тестового задания и приводится тут лишь для полноты картины. Предметом является Frontend часть, а именно - единичный элемент IoT.

ОПИСАНИЕ ИОТ

- 1) IoT, перемещаясь в пространстве, умеет самостоятельно определять свою геопозицию (GEO) и находить локального провайдера сети интернет.
- 2) IoT, по RADIO-1 обменивается с Backend используя ориентированный на подключение протокол коммуникации. Время подключения и обмена данными несравнимо мало относительно времени ожидания между подключениями
- 3) Коммуникация с Backend по RADIO-1 выглядит следующим образом:
 - A. IoT отправляет собственную геопозицию на Backend
 - B. В качестве ответа, IoT получает список параметров подключений к «соседям» (такими же IoT) в области видимости RADIO-2, но не более 10
 - C. IoT разрывает соединение с интернет по RADIO-1
- 4) IoT, по RADIO-2 обменивается со своими «соседями» используя не ориентированный на подключение протокол коммуникации. Данные с Sensor передаются соседям на их Actuator
- 5) IoT имеет три физических канала (устройства):
 - A. Button, позволяющая инициировать коммуникацию по RADIO-1 (обновление). При ее срабатывании, происходит запуск коммуникации по RADIO-1, отработка сценария коммуникации и завершение подсоединения
 - B. Sensor, данные с которого (Token) IoT распространяет всем своим «соседям» по RADIO-2. Появление данных на сенсоре, инициирует процесс распространения данных
 - C. Actuator, на который передаются полученные IoT данные (Token), распространяемые другими IoT по RADIO-2.

Общая программная архитектура показана ниже, в нотации UML2:



ЗАДАНИЕ

Пересмотрев приведенную выше программную архитектуру и оставив неизменным текстовое описание, требуется:

- A. Внести необходимые изменения в архитектуру и реализовать [«service» IoT] в виде Finite State Machine. Использование Qt FSM обязательно. std, boost и др. - опционально
- B. Для упрощения реализации, используйте замену аппаратных частей и сетевые коммуникации их программными Mock-объектами
- C. Задание рассчитано на различную квалификацию кандидата, в том числе и на начинающий уровень. Если вы претендуете на квалификацию повыше, вам необходимо предложить решение, основанное на шаблонах (templates).
- D. Результатом являются исходные коды, скрипты, папки и тд. – все что вы считаете результатом, но без бинарных и исполнимых модулей
- E. В работе мы используем C++17, Qt 6, CMake. Но сложности по установке требуемых дистрибутивов для выполнения задания, внесли свои коррективы. Допустимо использование Qt 5.12, язык C++ 14 и привычную вам оснастку сборки. 5.12.x Offline Installers мы их скачали за вас по ссылке <https://www.qt.io/offline-installers>, и разместили на доступном нашем сервере:
 - Linux Host: [Qt 5.12.12 for Linux 64-bit \(1.3 GB\)](#)
 - macOS Host [Qt 5.12.12 for macOS \(2.7 GB\)](#)
 - Windows Host [Qt 5.12.12 for Windows \(3.7 GB\)](#)

Дополнительно, для **защиты вашего результата от других кандидатов**, просим не выкладывать работу в публичном доступе, таком как Github и др. Вы сможете это сделать после 6 месяцев после отправки нам. Нам ваша работа нужна только для вашей оценки.

[VK](#)[Telegram](#)[YouTube](#)