

Задание программисту:

Необходимо реализовать сервер и клиент, имитирующие удаленное управление прибором. Прибор - универсальный многоканальный мультиметр (2..n каналов измерения напряжения).

Архитектура:

Операционная система - linux ;

- сервер - приложение, реализующая непосредственное управление аппаратными средствами прибора.
- клиент - приложение Qt , реализующее управление сервером и графический интерфейс пользователя на базе qml/qtquick 2.xx . Клиент и сервер взаимодействуют с помощью socket unix domain .

Протокол:

Вариант 1:

JSON-RPC 2.0

Вариант 2:

Текстовый протокол. Запрос и ответ - строка заказывающаяся CR . Формат команд: "command-name parameter1, parameter2, ... parameterN\n" Формат ответа: "ok|fail, result1, result2, ... resultN\n"

Например:

- запрос: "set_range channel0, range0".
- ответ : "ok, range0".

Команды управления:

Сервер реализует следующие функции:

Команда	Описание	Параметры	Возвращаемый результат
start_measure	запуск измерения	channel	ok/fail
set_range	настроить диапазон	channel , range	ok/fail, range
stop_measure	останов измерения	channel	ok/fail
get_status	получение статуса канала/устройства	channel	ok/fail, Код состояния - error_state/idle_state/measure_state/busy_state;

get_result	получение результата измерения	channel	ok/fail, result1,...resultN
-------------------	--------------------------------	---------	-----------------------------

Количество каналов прибора - `N`, задается на стадии компиляции, `N >= 2`. Канал задается в виде строки `channel[целое число от 0 до N-1]`, например, `"channel0"`, `"channel1"`, `"channel7"`. Диапазон канала измерения - Строка вида `range[целое число от 0 до M-1]`, где `M` равно 4. Например `"range0"`, `"range1"`, `"range2"`, `"range3"`. Сервер возвращает значения канала только в диапазоне данного канала. Использовать следующие параметры для канала: `range0`: соответствует диапазону `[0.0000001 ... 0.001)` В `range1`: соответствует диапазону `[0.001 ... 1)` В `range2`: соответствует диапазону `[1 ... 1000)` В `range3`: соответствует диапазону `[1000 ... 1000000)` В

Тип результата - текущее значение напряжения канала - `float`. Каналы работают независимо, возможно запускать и останавливать работу отдельного канала и получать статус. Коды состояния канала прибора: `error_state` - ошибка работы канала, должна быть сгенерирована в случайный момент времени при работе с сервером, клиент должен адекватно обработать состояние и индцировать его; `idle_state` - состояние готовности канала к измерениям; `measure_state` - состояние измерения канала; `busy_state` - состояние занятости канала к получению данных, генерируется случайным образом в процессе взаимодействия.

Общие требования

Язык программирования - `c++ 11/14/17`. Компилятор - `gcc`. Версия `Qt` - **5.11** и новее. Сборка под современные дистрибутивы `linux`, предпочтительно - `ubuntu`.

Требования к серверу:

- Сервер должен корректно обрабатывать ошибки.
- Сервер должен адекватно работать при подключении множества клиентов.
- Необходимо предусмотреть простую реализацию расширения списка команд, например, настроить частоту опроса или что-нибудь еще.
- Использование `qt` или иных фреймворков недопустимо.
- Логгирование работы сервера.

Требования клиенту:

- Клиент должен корректно обрабатывать ошибки.
- Подключение к серверу и реализация функции управления;
- Индикация статуса сервера.
- требования к юзабилити/удобству и так далее на усмотрение исполнителя .

Результат:

Результат должен быть представлен в виде исходных кодов, скриптов, `makefile` 'ов в открытом репозитории `Github/Bitbucket`. Должна быть предоставлено краткое описание по сборке проекта и требования к сборке в формате `*.md`.