Программа для настройки устройств по CAN-шине

Техническое задание

# Назначение ПО

ПО предназначено для конфигурирования параметров электронных исполнительных устройств. ПО запускается локально на ПК, по некоторому интерфейсу (первым приоритетом – шине CAN) подключается к устройству, используя предварительно сгенерированную карту настраиваемых параметров. Пользователь имеет возможность провести полную настройку устройства для его дальнейшей автономной работы.

# Требования к функционалу

Основные функции ПО:

1. Чтение и изменение каждого доступного параметра.
2. Представление параметров в виде виджетов. Расположение и вид виджетов задаются в профиле устройства.
3. Сохранение параметров в энергонезависимую память и загрузка параметров из неё.
4. Экспорт всех текущих значений в текстовый файл; импорт параметров из текстового файла. – в эксель, дальше ямл
5. Отображение графиков параметров в реальном времени (выбор параметров производится пользователем произвольно)

## Подключение к устройству

1. Настройки подключения к устройству задаются пользователем при запуске ПО. – нет, автомат
2. Настройки подключения должны сохраняться с последнего запуска ПО. . – нет, автомат
3. Подключение к устройствам может производиться:
   1. По разным интерфейсам (CAN, Ethernet, RS485)
   2. По разным протоколам в рамках одного интерфейса (пример: CanOpen, XCP, Modbus)
   3. По разным адресам в рамках одного протокола
4. Для подключения к устройствам CAN используются переходники USB-CAN сторонних производителей. На первом этапе необходимо поддержку переходников компаний Марафон (CANbus-USB) и Kvaser (Leaf Light v2).
   1. Если к ПК подключается несколько CAN-переходников, пользователь должен иметь возможность выбрать, какой из них использовать. . – нет, автомат
   2. Пользователь должен иметь возможность выбрать настройки подключения (для CAN – битрейт). . – нет, автомат

## Профили устройств

1. В папке с ПО хранятся профили для различных устройств. Каждый профиль содержит в себе список настраиваемых параметров, их адресацию и параметры представления. – в процессе
2. Профили хранятся в формате XML, YAML или JSON (будет решено позднее). – ямл, в процессе
3. После каждого изменения настроек подключения (или периодически) ПО отправляет запрос на устройство, пытаясь детектировать его тип и версию (HW и SW). Точная процедура детектирования (какие параметры считывать) определяется протоколом подключения. Если устройство отвечает, ПО выбирает наиболее подходящий профиль из имеющихся. - В процессе подбор профиля по версии ПО
4. При необходимости пользователь может выбрать профиль вручную. . – нет, автомат
5. Профиль содержит следующую информацию (список может дополняться в будущем)
   1. Тип поддерживаемого протокола
   2. Тип и название поддерживаемого устройства
   3. Минимальную версию аппаратного и программного обеспечения поддерживаемого устройства. – нет, нафига
   4. Поддержка сохранения / загрузки в энергонезависимую память
   5. Параметры
   6. Список доступных параметров
      1. Каждый параметр представляется в виде виджета.
      2. Все виджеты группируются по вкладкам.
      3. Для каждого виджета /параметра указываются:
         1. Тип переменной (размерность в битах, знаковость, для булевых переменных можно указать позицию в байте)
         2. Права доступа (read / write)
         3. Адресация переменной в рамках протокола
         4. Параметры отображения переменной (множитель, смещение, единица измерения, максимальное и минимальные значения)
         5. Виджет, который отвечает за переменную
            1. Название переменной
            2. Расположение виджета
            3. Индивидуальные параметры виджета (например, диапазон значений для окраски виджета в другой цвет)
   7. Было бы круто потом прописывать взаимосвязи между виджетами. Например, *если Переменная1 имеет значение 0, то Виджет2 неактивен*. Это можно на потом отложить.
6. При работе программы она непрерывно обновляет текущие значения параметров, доступных на текущей вкладке. Частота обновления параметров задаётся в настройках программы. – нет, всегда максимальна
7. Пользователь может менять параметры, открытые для записи. При этом их значения сразу должны отправляться в устройство.

## Построение графиков

1. В ПО предусмотрено отдельное окно для построения графиков.
2. В окне должно строиться не менее 4х каналов одновременно.
3. Все графики должны иметь строгую привязку по времени.
4. Должна быть возможность приостановить построение графиков и «отмотать» их на нужную позицию для анализа. «Глубина» сохранения истории должна задаваться настройкой.
5. Любой из параметров можно назначить для отображения на графике. При этом для него можно задать масштабирование и смещение, чтобы параметры с разными размерностями выводились в сопоставимом масштабе.
6. На графике можно задать как минимум два вертикальных курсора, чтобы было возможно считать значения всех параметров в эти моменты времени и разницу во времени между ними.
7. Частота считывания параметров графика задаётся отдельной настройкой.

## Работа с энергонезависимой памятью

1. Программа предоставляет отдельные кнопки Сохранить и Загрузить, предназначенные для отправки соответствующей команды на устройство.
2. Точный функционал кнопок определяется протоколом взаимодействия с устройством.

## Импорт и экспорт значений

1. Программа предоставляет отдельные кнопки Импорт и Экспорт.
2. По кнопке Экспорт происходит сохранение всех изменяемых пользователем переменных в текстовый файл.
3. По кнопке Импорт происходит загрузка всех переменных, содержащихся в текстовом файле, в устройство. – нет, опасно, ручками
4. Формат текстового файла определится позднее. Файл должен быть понятен для редактирования и анализа сохраненных параметров.

# Требования к реализации

1. Программа должна быть написана на языке С++ с использованием фреймворка Qt. Предпочтительна реализация GUI с использованием языка QML.
2. Программа в первую очередь предназначена для запуска на ОС Windows 10, но при разработке должна учитываться возможность переноса на ОС Linux.
3. Ведение проекта идёт на сервере Gitlab компании-заказчика.
4. Архитектура программы должна подразумевать возможные расширения для
   1. Добавления аппаратных адаптеров (USB-CAN)
   2. Добавления новых протоколов
   3. Добавления виджетов