



Создание расширенных систем сбора
и обработки данных на основе ПЛИС

Рафаэл Касабян, RAFA Solutions

Содержание

- ▶ Введение
- ▶ Сферы применения систем сбора и хранения данных
- ▶ Основные функции и свойства систем сбора данных (ССД)
- ▶ Основные вызовы и приемы проектирования универсальных ССД
- ▶ Приемы и советы по проектированию систем на ПЛИС
- ▶ Лучшие практики в проектировании пользовательского интерфейса ССД

Введение

- ▶ Системы сбора и хранения данных измеряют, записывают, обрабатывают и показывают физические явления.
- ▶ Эти процессы включают:
 - ▶ Преобразование физических явлений в электрический сигнал
 - ▶ Согласовка сигнала
 - ▶ Преобразование аналоговых сигналов в цифровые
 - ▶ Обработка и анализ данных
 - ▶ Хранение и представление данных в реальном времени
- ▶ Включение всех приведенных процессов в единое устройство (кроме датчиков) делает ССД комплексными и сложными.

Сферы применения систем сбора и хранения данных

Применения систем сбора и хранения данных


- ▶ Метеостанции
- ▶ Сбор данных и анализ качества и загрязнения воздуха
- ▶ Нефтегазовая промышленность
- ▶ Ветреные станции, солнечная энергетика
- ▶ Автоматизация и машиностроение
- ▶ Системы отопление, вентиляции и кондиционирования
- ▶ Структурный мониторинг состояния зданий и конструкций
- ▶ и т.д.



Основные функции и свойства систем сбора данных

Основные функции и свойства ССД

- ▶ Количество каналов
- ▶ Тип входных каналов
- ▶ Входной диапазон
- ▶ Частота дискретизации
- ▶ Разрешение
- ▶ Коммуникационные интерфейсы
- ▶ Обработка данных
- ▶ Хранение данных
- ▶ Представление в режиме реального времени
- ▶ Портативность
- ▶ Офлайн анализ данных
- ▶ Создание отчетов



Основные вызовы проектирования универсальных систем сбора данных

Основные вызовы проектирования ССД

- ▶ Автономная работа
- ▶ Портативность
- ▶ Поддержка множества датчиков для различных областей применения
- ▶ Принятие решений и контроль в режиме реального времени
- ▶ Стабильная работа
- ▶ Высокая скорость сбора данных и точность (для некоторых приложений)
- ▶ Универсальное программное обеспечение

Все эти вызовы делают аппаратные устройства на базе ПЛИС одними из лучших платформ для ССД.

Основные вызовы проектирования ССД (прод.)

- ▶ Программное обеспечение ССД должно быть масштабируемым, легко изменяемым и стабильным.
- ▶ ПО не должно быть ориентировано на конкретные датчики и приложения.
- ▶ ПО должно быть легко изменено для поддержки различных аппаратных платформ.
- ▶ Функции и свойства должны добавляться согласно требованиям аппаратного устройства и приложения, не требуя архитектурных изменений.

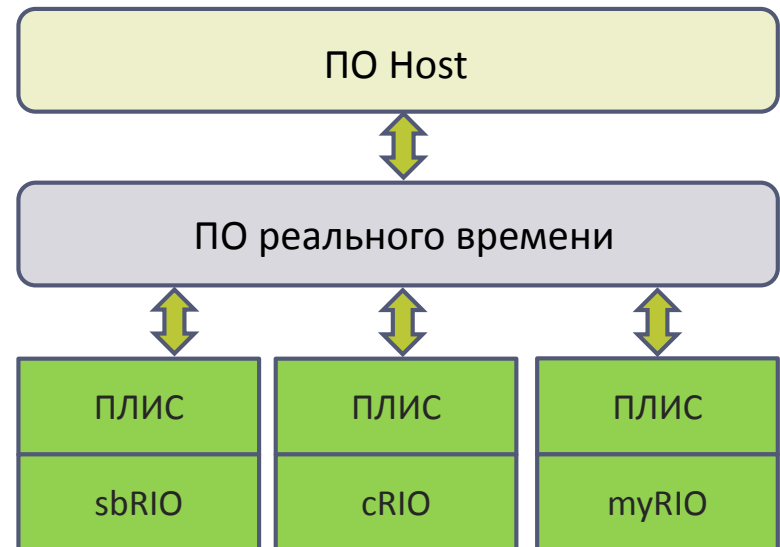
Основные приемы проектирования универсальных систем сбора данных

Основные приемы проектирования ССД

- ▶ Абстракция аппаратных средств
- ▶ Не жестко запрограммированная конфигурация – использование данных из SQL, конфигурационных файлов и т.д.
- ▶ Использование ПЛИС
- ▶ Конфигурируемое ПО с понятным интерфейсом для обеспечения легкой и простой работы
- ▶ Суб-панели
- ▶ Объектно-ориентированное программирование

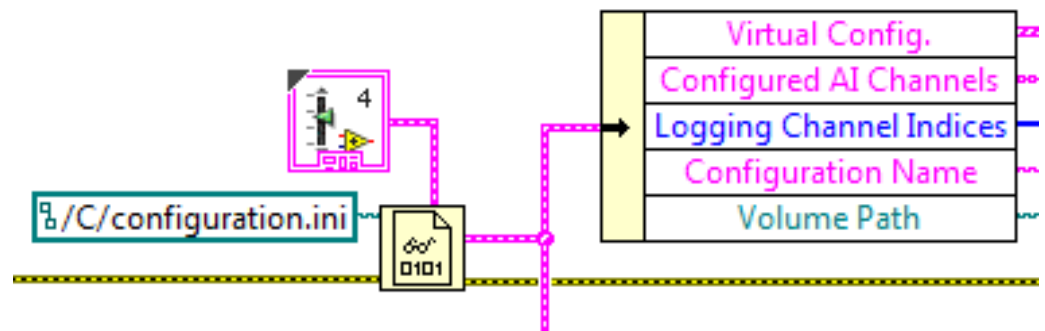
Абстракция аппаратных средств

- ▶ Позволяет легко адаптировать для различных аппаратных средств.
- ▶ ПО Host-а и реального времени не зависят от аппаратного обеспечения.
- ▶ ПО Host-а считывает с файла конфигурацию аппаратного обеспечения (количество каналов, их тип, и т.д.) и соответственно обновляет интерфейс ПО.



Абстракция аппаратных средств (прод.)

- ▶ Программа реального времени:
 - ▶ Обрабатывает конфигурационные данные, и передает их на ПЛИС,
 - ▶ Получает данные аналоговых каналов от ПЛИС через DMA,
 - ▶ Получает данные цифровых сигналов через сгруппированные логические биты (bitpacked Boolean logic)



Абстракция аппаратных средств (прод.)

- ▶ Следующим аспектом абстракции аппаратных средств является поддержка различных датчиков.
- ▶ Поддержка различных методов конвертации данных,
- ▶ Отдельный интерфейс для создания методов конвертации,
- ▶ Использование полиномиальных уравнений для создания кривых для конвертации.

The screenshot displays a software interface for configuring a device. It includes the following elements:

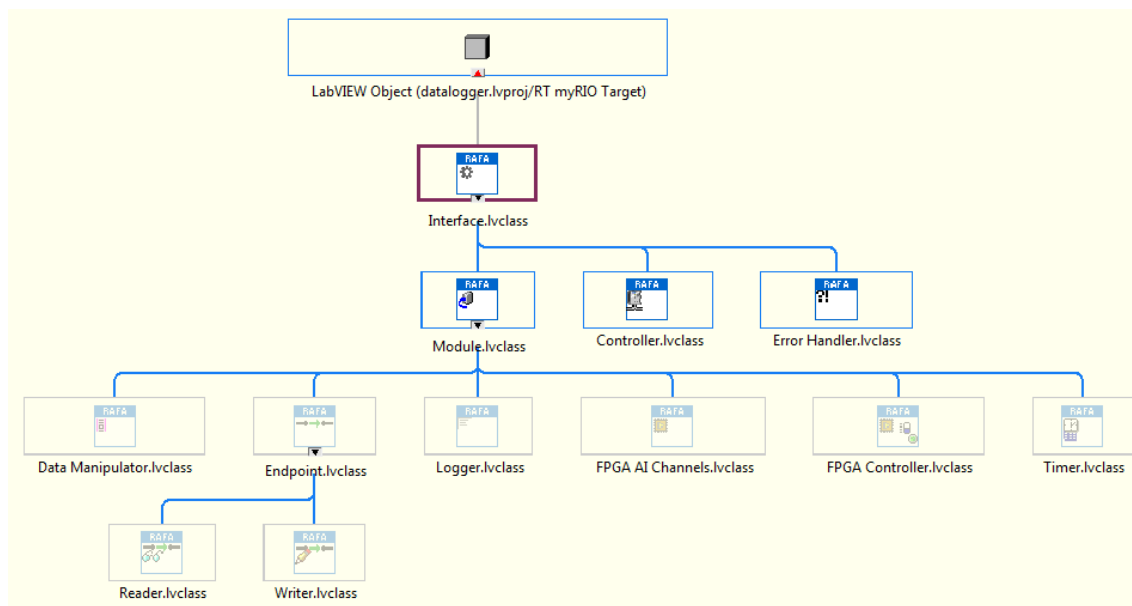
- Channel Name:** Pressure, **Unit:** psi
- Order:** 2, with a **Build** button.
- Polynomial Equation:** $y = 3.621E-16 + 1*x + 5.571E-17*x^2$
- Calibration Points Table:**

Voltage	Value
0.000000	0.000000
1.000000	1.000000
2.000000	2.000000
3.000000	3.000000
4.000000	4.000000
- Buttons:** Add, Load from File, Delete
- Calibration Graph:** A plot of Pressure-psi vs Voltage showing a linear relationship. The x-axis (Voltage) ranges from 0 to 4, and the y-axis (Pressure-psi) ranges from 0 to 4. A blue line represents the 'Voltage' data, and a red line represents the 'Fitted' curve.

Объектно-ориентированное программирование (ООП) систем реального времени

Использование объектно-ориентированного программирования в программах реального времени позволяет иметь:

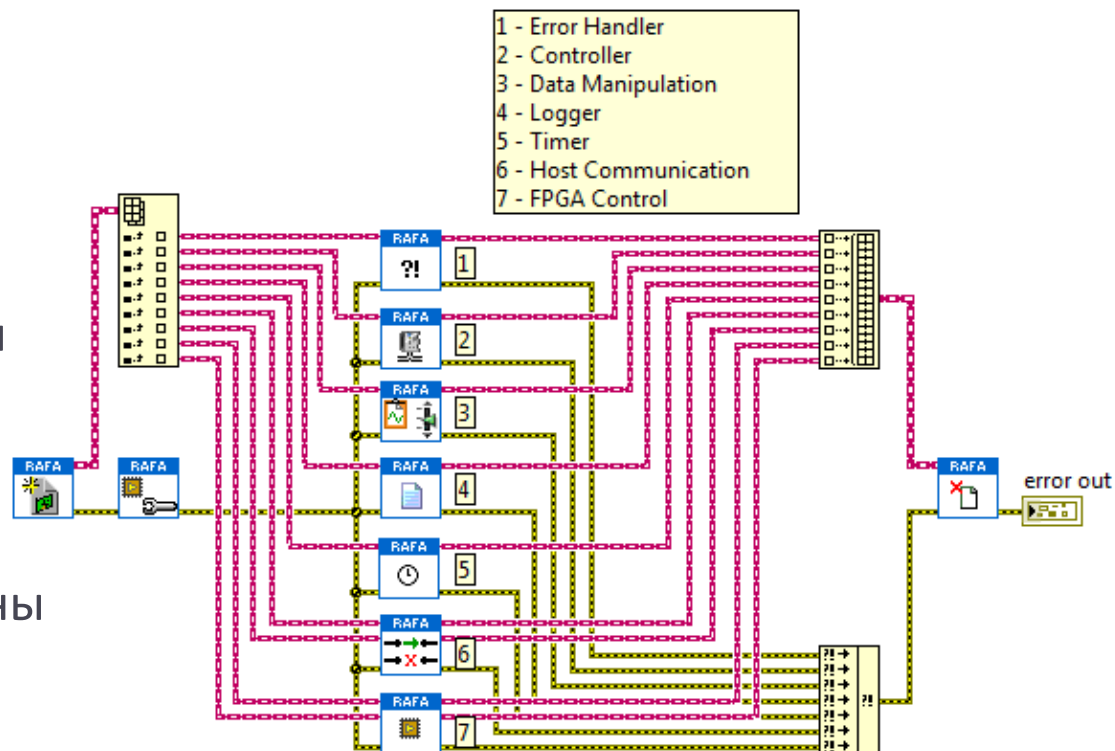
- ▶ Иметь наиболее гибкую архитектуру
- ▶ Минимизировать количество проводников между модулями
- ▶ Улучшить обслуживание кода
- ▶ Улучшить масштабируемость



ООП систем реального времени. Пример.

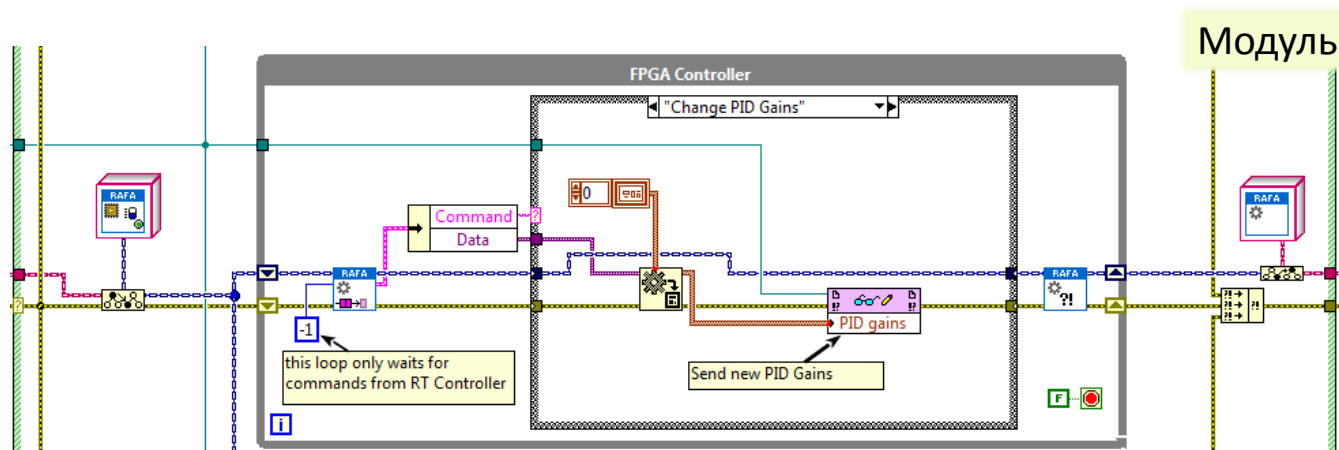
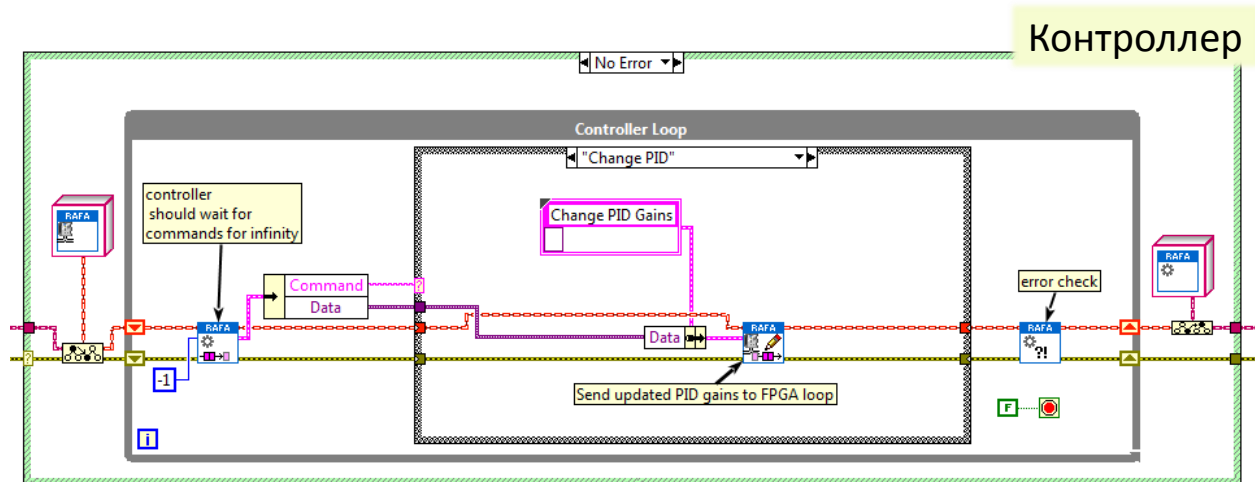
ПО реального времени для систем сбора и хранения данных.

- ▶ Все модули системы управляются с контроллера.
- ▶ Коммуникация между модулями и контроллером и передача данных выполняется в помощью «очередей».
- ▶ Ссылки «очередей» внедрены в данные объектов.



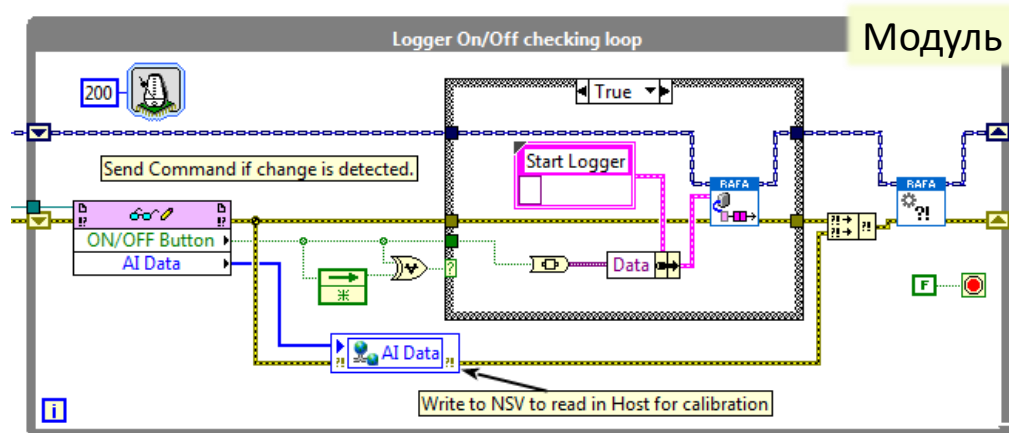
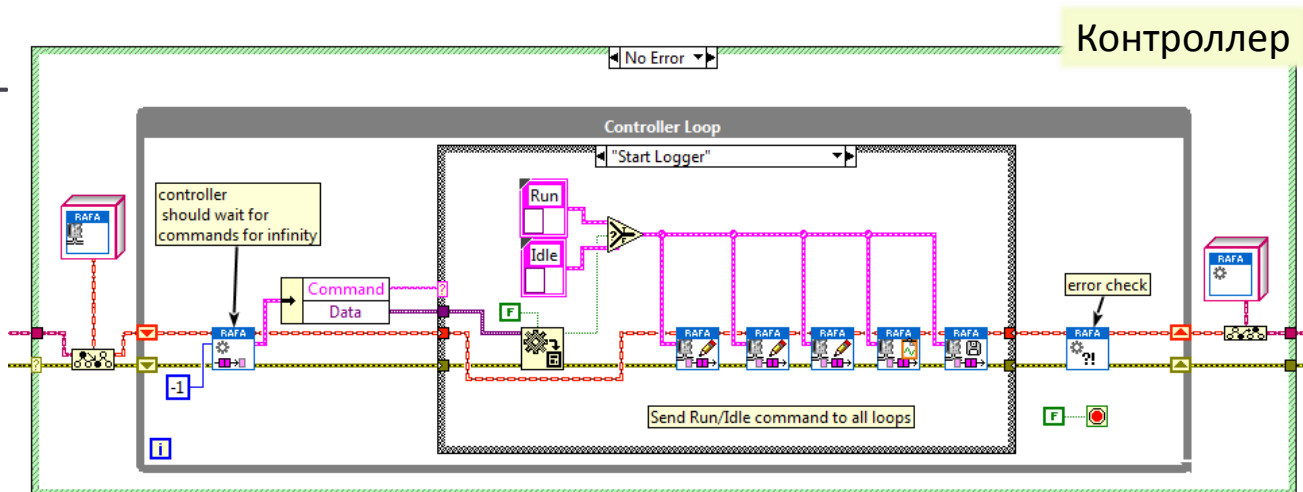
ООП систем реального времени. Пример (прод).

- ▶ Коммуникация –
Контроллер ->
Модуль



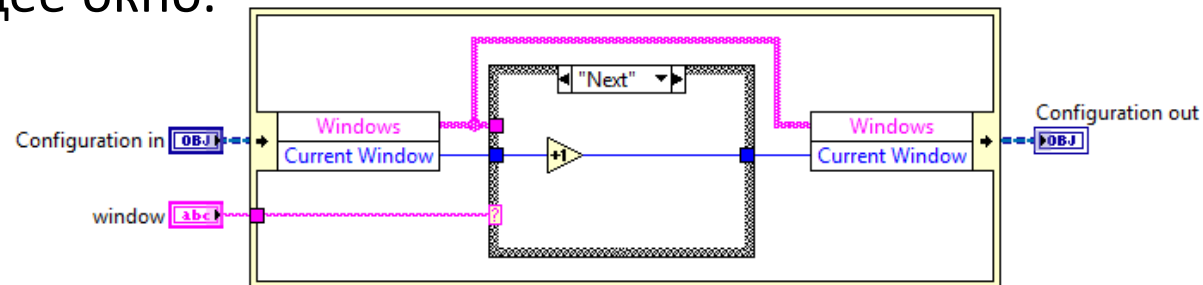
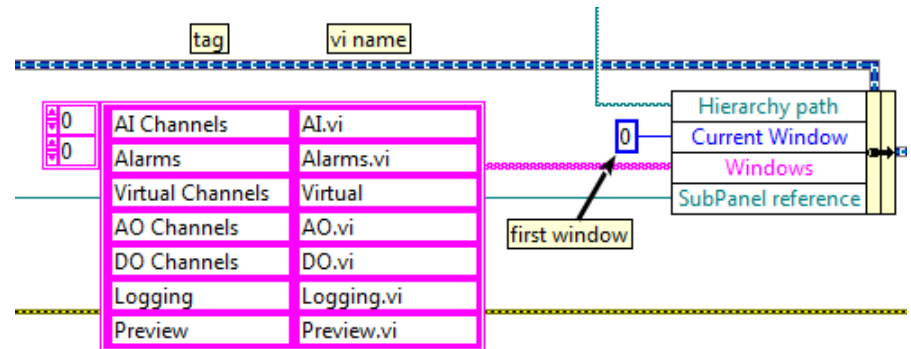
ООП систем реального времени. Пример (прод).

- ▶ Коммуникация –
Модуль ->
Контроллер



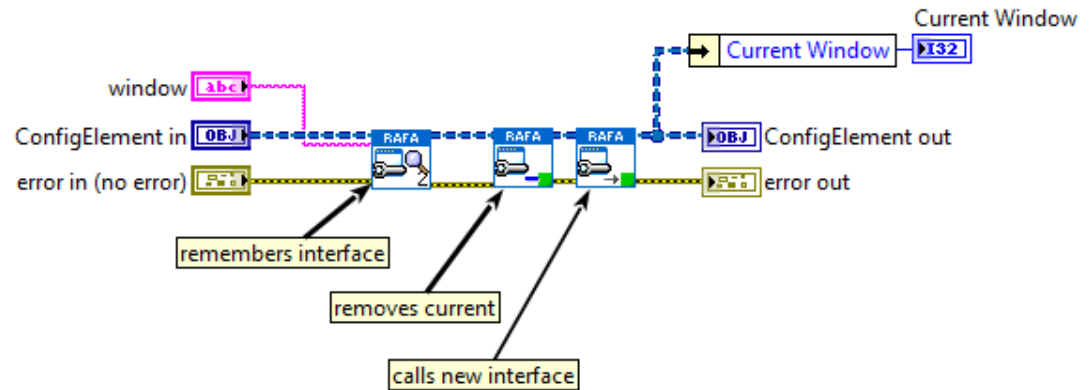
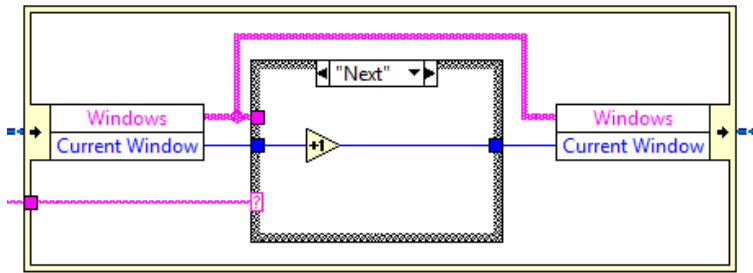
Суб-панели

- ▶ Вызов окон с любой последовательностью.
- ▶ Определение последовательности вызова окон согласно требуемым функциям.
- ▶ Текущее окно может определять следующее окно.

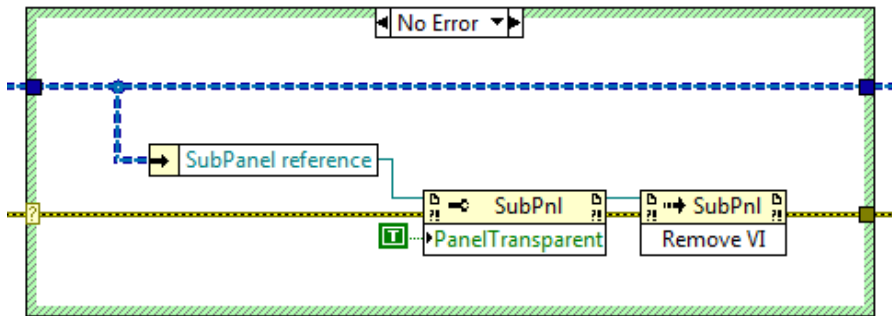


Применение суб-панели. Пример.

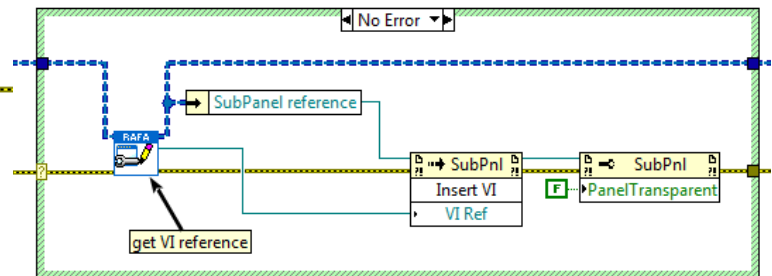
Определяет вызываемое окно



Снятие старого окна



Вызов нового окна



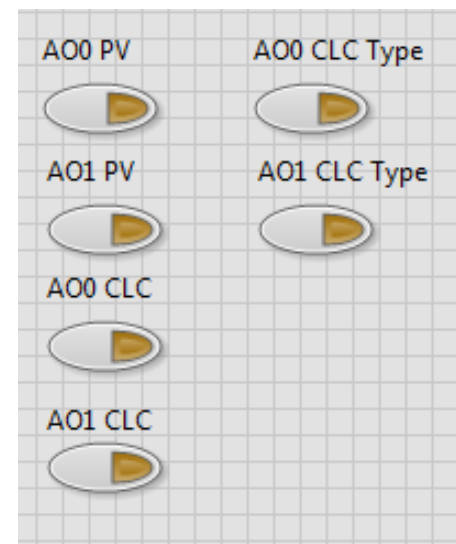
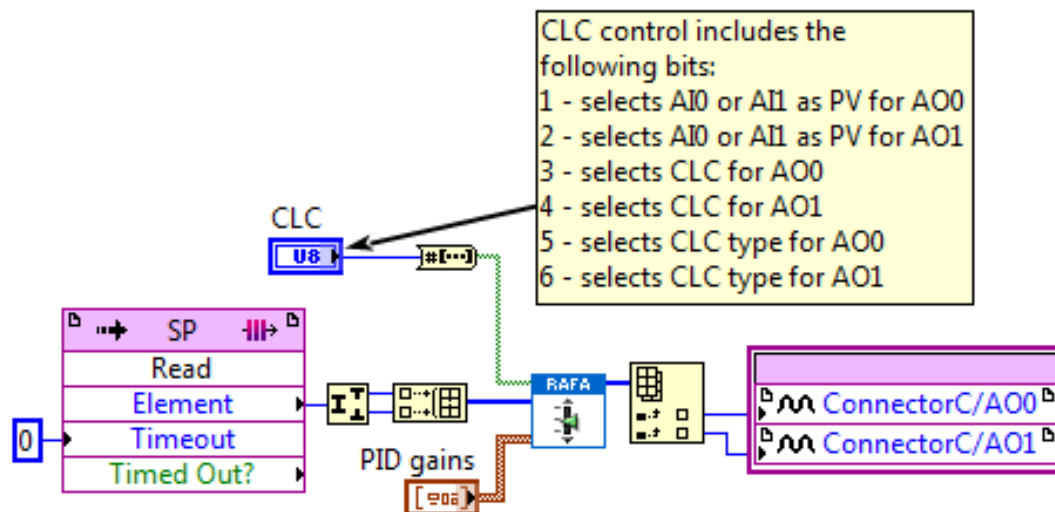
Приемы и советы по проектированию систем на ПЛИС

Приемы и советы по проектированию на ПЛИС

- ▶ Методы оптимизации ресурсов ПЛИС:
 - ▶ Группировка логических битов для уменьшения объектов на лицевой панели
 - ▶ Использование наименьшего возможного типа данных
 - ▶ Передача данных конфигурации с помощью DMA
 - ▶ Использование памяти ПЛИС для хранения данных конфигурации
- ▶ Использование группировки (interleaving) данных для легкого расширения числа каналов.
- ▶ Использование регулируемого времени цикла для установки частоты дискретизации.

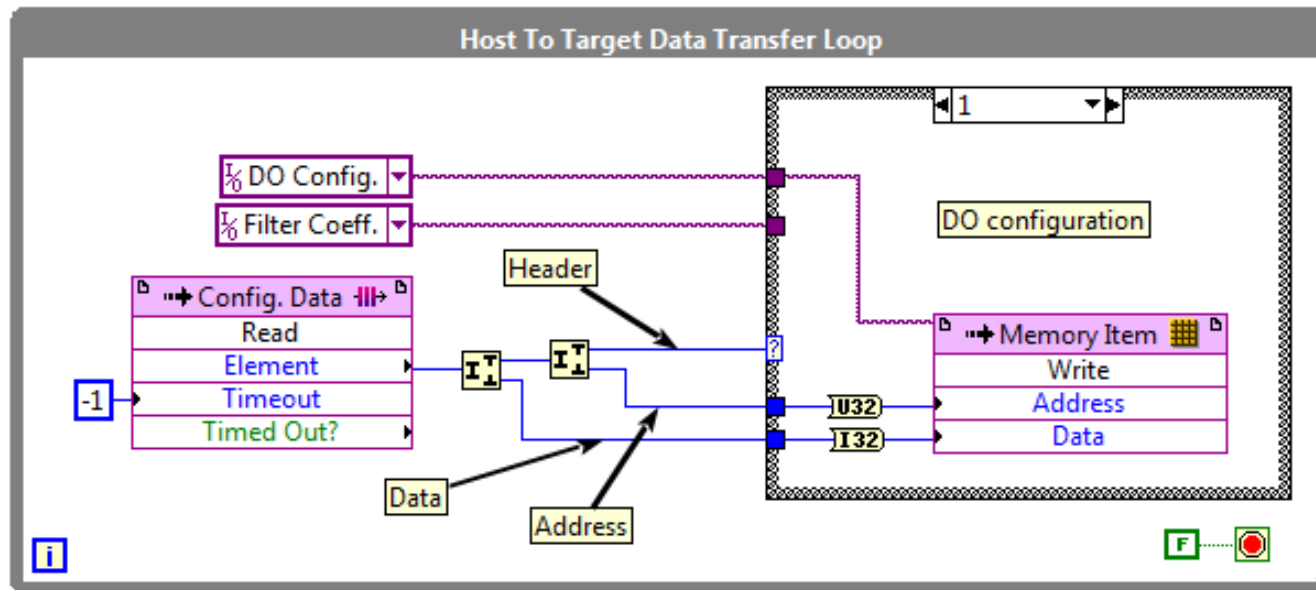
Приемы и советы по проектированию на ПЛИС

▶ Группировка логических битов



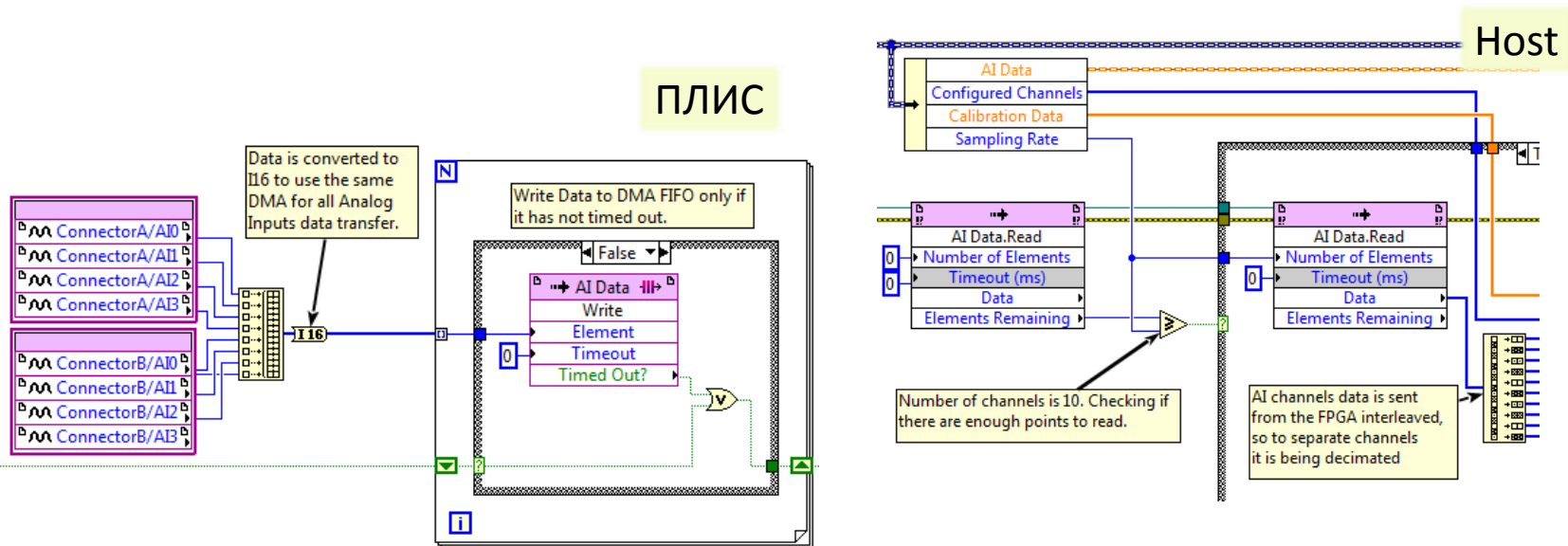
Приемы и советы по проектированию на ПЛИС

- ▶ Минимизация объектов лицевой панели используя передачу данных с помощью DMA
 - ▶ Включение заголовка в данные
 - ▶ Включение адреса при записи в память



Приемы и советы по проектированию на ПЛИС

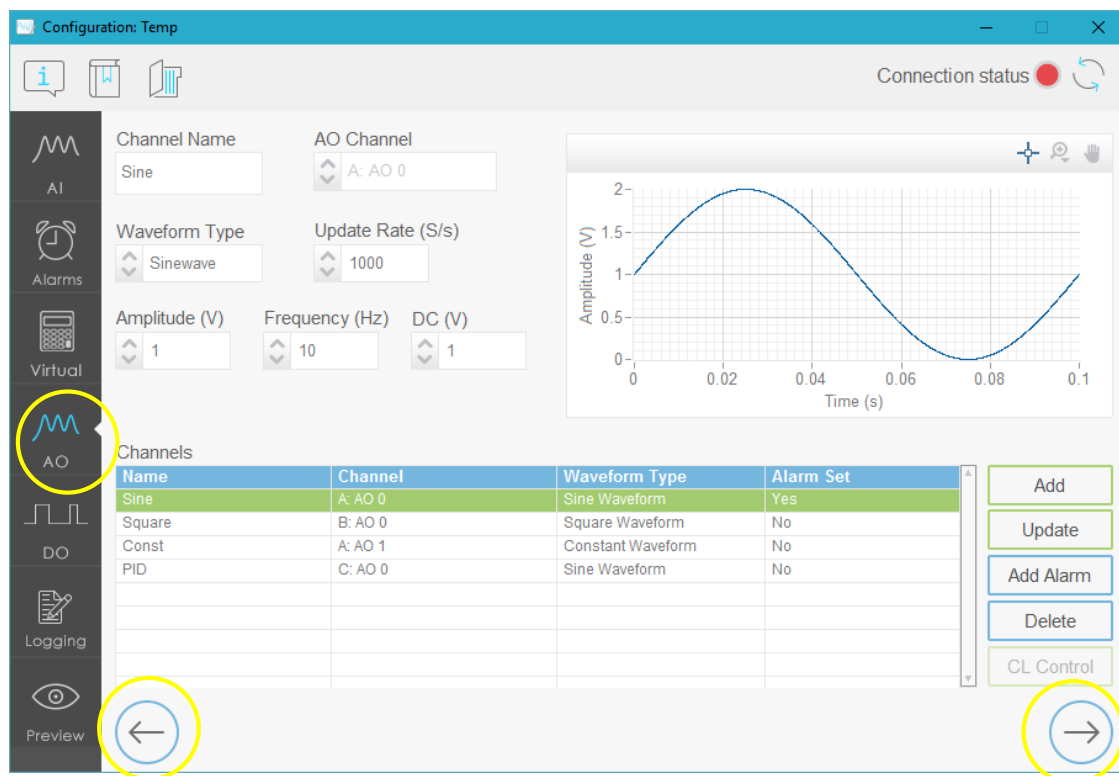
- ▶ Использование группировки (interleaving) данных для увеличения числа каналов дает возможность поддержки различных устройств.
- ▶ Количество считываемых данных в программе Host-а зависит от частоты дискретизации установленным пользователем.



Лучшие практики в проектировании пользовательского интерфейса

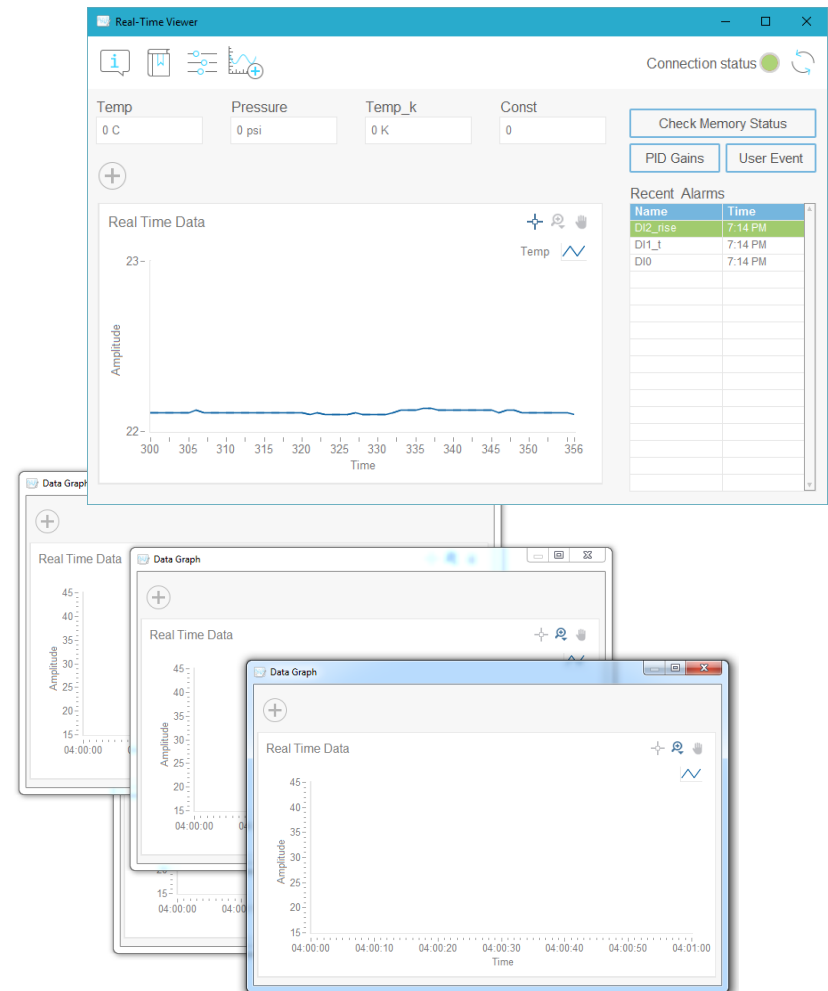
Лучшие практики в проектировании пользовательского интерфейса

- ▶ Ясный и предсказуемый интерфейс.
- ▶ Предопределенная последовательность окон согласно конфигурированию устройства.
- ▶ Указатели последующего и предшествующего шага.
- ▶ Указатель местоположения в приложении.



Лучшие практики в проектировании пользовательского интерфейса (прод.)

- ▶ Возможность открытия нескольких параллельных экранов для представления данных в режиме реального времени.
- ▶ Показ данных в виде цифровых индикаторов и графиков.





Спасибо!