

Абстрактная модель контроллера СКУД.

Абстрактная модель контроллера СКУД – это модель, отражающая общие характеристики контроллера СКУД.

Как именно реализован контроллер СКУД, как он взаимодействует со считывателями, кнопками, замками, турникетами и т.п. для абстрактной модели не существенно.

Абстрактная модель СКУД позволяет сформулировать единый набор правил работы с контроллером СКУД, что, в свою очередь, позволяет создать программное обеспечение, не зависящее от конкретного типа оборудования.

Базовая абстрактная модель контроллера СКУД.

Контроллер СКУД рассматривается как черный ящик, который удовлетворяет следующим требованиям:

Каналы связи.

1. Контроллер имеет канал связи, с помощью которого можно передавать команды в контроллер и получать от него данные.

Структура контроллера.

2. Алгоритм работы контроллера с подключенным оборудованием (считыватели карт, сканеры штрих-кода, замки, разного рода кнопки и т.п.) не являются существенными и не влияют на работу модели.
3. Контроллер может иметь (а может и не иметь) набор входных (источников данных) и выходных устройств, через которые контроллер получает информацию, необходимую ему для работы.

Идентификаторы.

4. Идентификатор - непустая последовательность символов. Идентификаторы хранятся в памяти контроллера.

Система команд.

5. Контроллер должен поддерживать следующий минимальный набор команд:

№ п/п	Команда	Ответ	Прим.
1	reportstatus	Yes/No	В ответе в текстовом виде
2	Getversion	Версия контроллера	В ответ передается версия
3	Writekey key=<Код идентификатора>	OK/Error	Сообщается результат
4	deletekey key=<Код идентификатора>	OK/Error	Сообщает результат
5	Readkey key=<Код идентификатора>	OK/Error	OK если идентификатор
6	ClearKeys	OK/Error	Удаляет все идентифи
7

Журнал событий.

6. Контроллер ведет журнал событий. Формат журнала событий:

№ п/п	Поле	Обязательное поле
1	Код события	Да
2	Дата и время события	Нет
3	Прочие данные, связанные с событием.	Нет

События выводятся в ASCII, построчно, разделитель LF CR.

Валидация идентификатора.

Процесс валидации:

7. Контроллер получает от подключенного код идентификатора.
8. Контроллер выполняет проверку на наличие полученного идентификатора с теми, что хранятся в памяти контроллера.
9. Если полученный код идентификатора и хотя бы один из хранящихся в памяти контроллера совпадают, то контроллер формирует сигнал об успешной валидации на

выходные устройства. Порядок подачи этого сигнала определяется самим контроллером СКУД.

10. Процесс валидации может зависеть от других факторов, которые контроллер получает от подключенного оборудования в ходе своей работы.

Процесс валидации является основной задачей контроллера СКУД и выполняется внутри контроллера СКУД.

Процесс управления контроллером СКУД. Управление списком идентификаторов.

Управление контроллером СКУД сводится к управлению списком загруженных в него идентификаторов.

Загрузка идентификатора в контроллер.

Загрузка идентификаторов в контроллер производится командой

Writekey key=<код идентификатора>

Если необходимо загрузить несколько идентификаторов, то команда writekey выполняется для каждого идентификатора:

Writekey key=<код идентификатора 1>

Writekey key=<код идентификатора 2>

Writekey key=<код идентификатора 3>

...

Writekey key=<код идентификатора N>

Удаление идентификатора из контроллера.

Удаление идентификатора из контроллера производится командой

DeleteKey key=<код идентификатора>

Если необходимо удалить несколько идентификаторов, то команда DeleteKey выполняется для каждого идентификатора.

DeleteKey key=<код идентификатора 1>

DeleteKey key=<код идентификатора 2>

....

DeleteKey key=<код идентификатора N>

Удаление всех идентификаторов.

Удаление всех идентификаторов производится командой

ClearKeys.

После выполнения этой команды в памяти контроллера нет сохраненных идентификаторов.

Получение журнала событий.

При обращении к контроллеру он должен выдать непрочитанную часть журнала событий в виде:

<код события><Дата и время события по часам контроллера><Прочие параметры, связанные с событием>

Немного практики для демонстрации теории.

Рассмотрим физическую модель контроллера, наиболее близкую к теоретической.

Описание контроллера.

Количество подключаемых считывателей - 1 (по протоколу Wiegand или TouchMemory).

Линия связи - имеется, для определенности пусть будет RS-232.

Остальные характеристики контроллера для управления несущественны.

Пример такого контроллера - домофон на двери подъезда.

Постановка задачи.

Задача: разрешить проход через контроллер 10 ключам, которые имеются в руках у пользователя.

Коды ключей заранее неизвестны.

Порядок решения задачи.

Регистрация ключей.

1. выясняем наличие контроллера на связи командой `reportstatus`. Если контроллер на связи, то продолжаем работу.
2. даем команду на выборку журнала событий. Выбираем события до тех пор, пока они не закончатся.
3. подносим ключ к считывателю.
4. читаем журнал событий. Анализируем журнал на наличие события вида Неизвестный код ключа. Выделяем этот код ключа и запоминаем. Пусть для определенности этот код имеет значение `key="1234"`
5. Записываем код ключа в контроллер командой `writekey key="1234"`.
6. выполняем пп 3-5 для остальных ключей.

Эксплуатация системы.

1. эксплуатация системы сводится к выборке журнала событий.

Удаление ключей.

При необходимости удаления ключа (например, `key="1234"`) из памяти контроллера необходимо и достаточно:

1. сформировать команду deletekey key="1234"

После обработки этой команды ключ не сможет ходить через контроллер. Проход этому ключу будет запрещен.

Т.о., манипуляция списком ключей в памяти контроллера обеспечивает решение задачи СКУД.