ТЭНДО

комплексная автоматизация бизнес-процессов

ТЭНДО
Плата I/O расширений
СВ-АС21
API
Руководство программиста

Москва 2015

Оглавление

Введение

API (Application Programming Interface) представляет собой набор функций, который программист может использовать для доступа к функциональности платы I/O расширений Tendo Connection Board AC21. API-функции позволяют осуществлять интеграцию платы в сторонние приложения. API-функции Системы написаны на языке C++. Поэтому данное руководство ориентировано на программиста, имеющего представление об этом языке программирования.

1. Состав поставки API

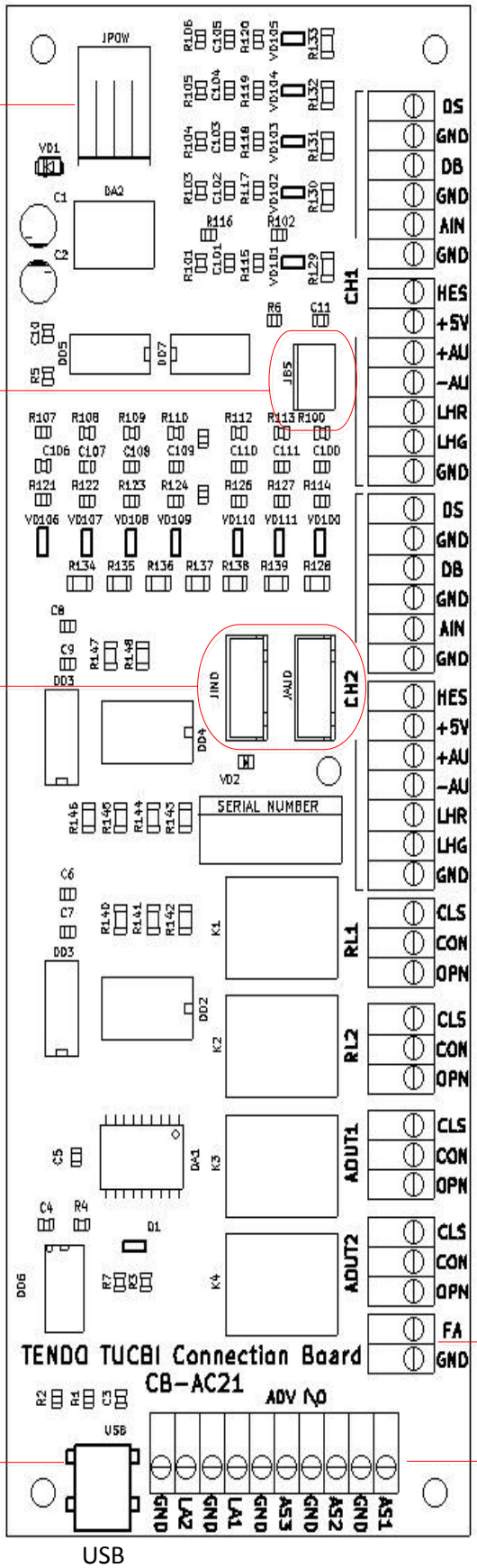
В состав поставки API входят следующие файлы:
Для Windows (x86):

- INCLUDE\tcb_ports.h – файл объявления портов платы
- INCLUDE\tcb_ext.h – файл определения макросов импорта
- INCLUDE\tcb_factory.h – файл объявления функций API
- BIN\TCBoardAPI_AC21.dll
- BIN\FTD2XX.dll
- LIB\TCBoardAPI_AC21.lib

Для Linux (Intel x86):

- INCLUDE\tcb_ports.h
- INCLUDE\tcb_ext.h
- INCLUDE\tcb_factory.h
- BIN\libTCBoardAPI_AC21.so
- BIN\libftd2xx.so

2. Схема расположения портов на плате



JPOW – Разъем питания

JB5 – Датчик открытия шкафа

JIND - шлейф светодиодов считывателя

JAUD - аудиовход

USB – Порт USB

CH1 - Канал 1

CH2 - Канал 2

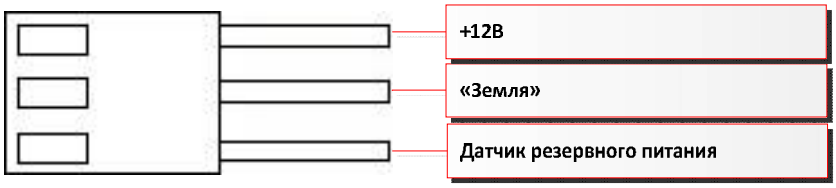
Блок реле

FA – Пожарная сигнализация

ADV I\O – Блок дополнительных входов/выходов

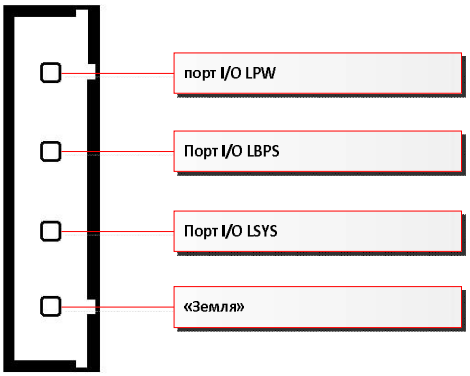
USB

2.1. JPOW – Разъем питания



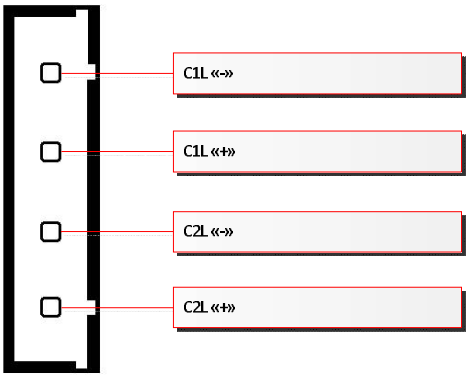
2.2. JBS – Датчик открытия шкафа

2.3. JIND – Шлейф светодиодов



Обозначение	Описание
Порт I\O LBPW	Питание от сети
Порт I\O LBPW	Резервное питание
Порт I\O LBPW	Индикатор работы системы
«Земля»	

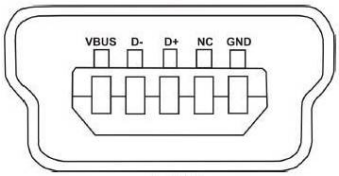
2.4. JAUD – Аудиовход



Обозначение	Описание
C1L «-»	Левый канал усилителя «-»
C1L «+»	Левый канал усилителя «+»
C1L «-»	Правый канал усилителя «-»
C1L «+»	Правый канал усилителя «+»

2.5. USB – Порт USB

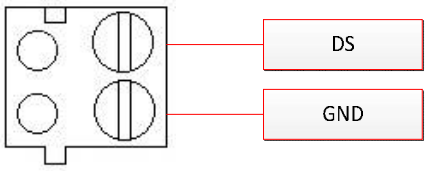
Стандартный порт mini-USB



2.6. CH1 – Канал 1/ CH2 – Канал 2

2.6.1. DS - Датчик открытия двери (Door Sensor)

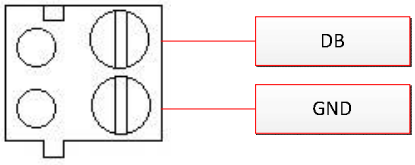
В замкнутом состоянии – дверь закрыта.



Обозначение	Описание
DS	Датчик открытия двери
GND	«Земля»

2.6.2. DB – Кнопка выхода (Door Button)

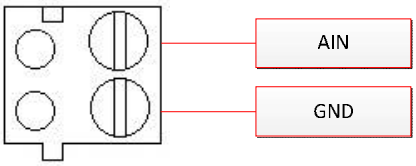
В замкнутом состоянии – кнопка нажата



Обозначение	Описание
DB	Датчик кнопки выхода
GND	«Земля»

2.6.3. AIN – Дополнительный вход (Additional In)

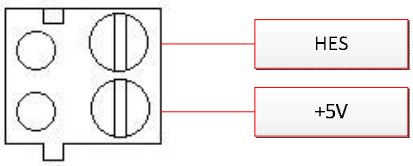
Служит для подключения датчиков других типов («Гюрза», объемные и др.), которые срабатывают по типу «сухого контакта».



Обозначение	Описание
AIN	Датчик дополнительного входа
GND	«Земля»

2.6.4. HES – Датчик наличия руки на считывателе (Hand Exists Sensor)

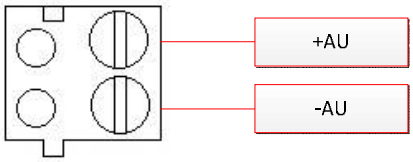
Подводка питания к датчику наличия руки. При появлении объекта над считывателем на контакт HES подается 5В и состояние датчика меняется на «1».



Обозначение	Описание
HES	Датчик наличия руки
+5V	Подача 5В

2.6.5. AU – встроенный динамик считывателя (Audio)

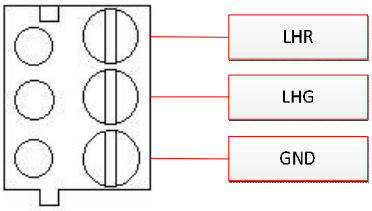
Подводка питания к встроенному динамику считывателя



Обозначение	Описание
+AU	Подвод питания
-AU	Подвод питания

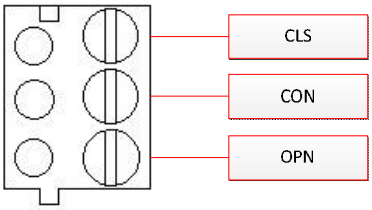
2.6.6. LH – Подключение индикаторов (светодиодов) считывателя

Поканальный индикатор состояния канала. Могут подключаться два одноцветных светодиода или один двуцветный.



Обозначение	Описание
LHR	Красный светодиод
LHG	Зеленый сетодиод
GND	«Земля»

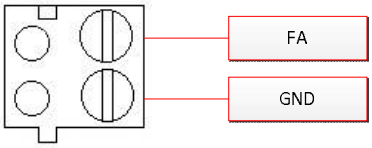
2.7. Блок реле (2 реле замков + 2 резервных выхода)



Обозначение	Описание
CLS	Выход реле нормально-замкнутый
CON	Общий выход
OPN	Выход реле нормально-разомкнутый

2.8. FA – Датчик пожарной сигнализации

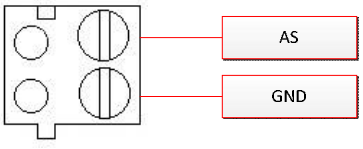
При замкнутом контакте будет включена сирена.



Обозначение	Описание
FA	Датчик пожарной сигнализации
GND	«Земля»

2.9. ADV I\O – Блок дополнительных входов/выходов

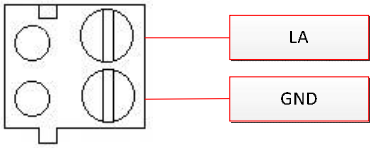
2.9.1. AS – Резервный вход датчика (3 входа)



Обозначение	Описание
AS	Вход дополнительного датчика
GND	«Земля»

2.9.2. LA – Резервный выход (2 выхода)

Для подключения дополнительных светодиодов при необходимости



Обозначение	Описание
LA	Резервный выход
GND	«Земля»

3. Объявление портов

Обращение к портам платы производится по заранее заданным именам.

	Описание	Каналы
Поканальные входы		
TCB_PORT_AC21_I_DOOR_SENSOR	Датчик открытия двери	1-й канал - 0 2-й канал - 1
TCB_PORT_AC21_I_DOOR_BUTTON	Кнопка выхода	
TCB_PORT_AC21_I_HEND_EXIST_SENSOR	Датчик наличия руки у считывателя	
TCB_PORT_AC21_I_ADDITIONAL_INPUT	Дополнительный вход	
Поканальные выходы		
TCB_PORT_AC21_O_RELAY	Выход реле замка	1-й канал - 0 2-й канал - 1
TCB_PORT_AC21_O_ADDITIONAL_OUTPUT	Резервный выход реле	
TCB_PORT_AC21_O_LIGHT_HAND_RED	Красный светодиод	
TCB_PORT_AC21_O_LIGHT_HAND_GREEN	Зеленый светодиод	
Общие входы		
TCB_PORT_AC21_I_BOX_SENSOR	Датчик вскрытия шкафа контроллера	0
TCB_PORT_AC21_I_BACKUP_POWER_SENSOR	Датчик работы от источника резервного питания	
TCB_PORT_AC21_I_ADDITIONAL_SENSOR_1	Резервный датчик 1	
TCB_PORT_AC21_I_ADDITIONAL_SENSOR_2	Резервный датчик 2	
TCB_PORT_AC21_I_ADDITIONAL_SENSOR_3	Резервный датчик 3	
TCB_PORT_AC21_I_FIRE_SENSOR	Датчик пожарной тревоги	
Общие выходы		
TCB_PORT_AC21_O_LIGHT_POWER	Индикатор питания от основной сети	0
TCB_PORT_AC21_O_LIGHT_BACKUP_POWER_SENSOR	Индикатор питания от резервного источника	
TCB_PORT_AC21_O_LIGHT_SYSTEM	Индикатор работы системы	
TCB_PORT_AC21_O_LIGHT_ADDITIONAL_1	Резервный выход индикатора 1	
TCB_PORT_AC21_O_LIGHT_ADDITIONAL_2	Резервный выход индикатора 2	

4. Использование API-функций

4.1. Вызов API-функции

4.2. Описание параметров вызова API-функций

4.2.1. Создание объекта для работы с устройством

Синтаксис:

IConnectionBoard* TCB_DECL Create(TCB_MODEL model);

Параметры:

Параметр	Тип	Диапазон значений	Описание	Примечание
model	TCB_MODEL	TCB_MODEL_AC21	Модель создаваемого объекта	Указать только TCB_MODEL_AC21

Пример вызова:

IConnectionBoard* pBoard = IConnectionBoard::Create(TCB_MODEL_AC21);

Результат работы функции:

Создан объект для работы с устройством.

4.2.2. Инициализация устройства

Синтаксис:

bool TCB_DECL Init(int serial = 0);

Параметры:

Параметр	Тип	Диапазон значений	Описание	Примечание
serial	int	0 - 65535	Серийный номер подключаемого устройства	Если указать 0, то инициализируется любое одно устройство, подключенное к компьтеру

Пример вызова:

IConnectionBoard* pBoard = IConnectionBoard::Create(TCB_MODEL_AC21);
pBoard->Init(23)

Результат работы функции:

1. Будет инициализирован объект для работы с устройством.
2. Будет возвращен статус инициализации объекта:
   0 - true – успешно
   1 - false - неудача

4.2.3. Отключение объекта

Синтаксис:

void TCB_DECL Term();

Параметры:

Без параметров

Пример вызова:

```
IConnectionBoard* pBoard = IConnectionBoard::Create(TCB_MODEL_AC21);
pBoard->Init(23);
pBoard->Term();
```

Результат работы функции:

Объект будет отключен от работы с устройством

4.2.4. Проверка инициализации

Синтаксис:

bool TCB_DECL IsInitialize();

Параметры:

Без параметров

Пример вызова:

```
if (IsInitialize())
        cout << "Device is initialized" << endl;
else
        cout << "Device is not initialized" << endl;
```

Результат работы функции:

Будет возвращен статус инициализации объекта:
0 - true – успешно
1 - false - неудача

4.2.5. Вернуть описание устройства

Синтаксис:

void TCB_DECL GetDescription(char* description, int size)

Параметры:

Параметр	Тип	Диапазон значений	Описание	Примечание
description	char*	-	Указатель на строковый буфер	-
size	int	-	Размер буфера	-

Пример вызова:

```
const int sizebuf = 255;
char strbuf[sizebuf];
pBoard->GetDescription(strbuf, sizebuf);
cout << "Description device: " << strbuf << endl;
```

Результат работы функции:

Будет возвращено описание подключенного устройства.

4.2.6. Вернуть серийный номер устройства

Синтаксис:

```
int TCB_DECL GetSerialNumber();
```

Параметры:

Без параметров

Пример вызова:

```
int serial = pBoard->GetSerialNumber();
cout << "Serial number device: " << serial << endl;
```

Результат работы функции:

Будет возвращен серийный номер подключенного устройства.

4.2.7. Вернуть модель устройства

Синтаксис:

```
void TCB_DECL GetModel(char* model, int size);
```

Параметры:

Параметр	Тип	Диапазон значений	Описание	Примечание
model	char*	-	Указатель на строковый буфер	-
size	int	-	Размер буфера	-

Пример вызова:

```
const int sizebuf = 255;
    char strbuf[sizebuf];
pBoard->GetModel(strbuf, sizebuf);
cout << "Model device: " << strbuf << endl;
```

Результат работы функции:

Будет возвращена строка, содержащая описание модели подключенного устройства.

4.2.8. Установить значение на выход

Синтаксис:

void TCB_DECL SetValue(PORT port, int channel, int value);

Параметры:

Параметр	Тип	Диапазон значений	Примечание
port	PORT	По перчислению PORT	
channel	Int	0;1	
value	Int	0;1	

Пример вызова:

pBoard->SetValue(PORT_O_RELAY, 0, 1);

Результат работы функции:

Значение на указанном выходе на указанном канале будет установлено в соответствии с указанным параметром.

4.2.9. Вернуть значение на порту

Синтаксис:

int TCB_DECL GetValue(PORT port, int channel)

Параметры:

Параметр	Тип	Диапазон значений	Примечание
port	PORT	По перчислению PORT	
channel	Int	0;1	

Пример вызова:

value = pBoard->GetValue(PORT_I_BOX_SENSOR, 0);

Результат работы функции:

На указанном порту на указанном канале будет возвращено текущее значение.

4.2.10. Инвертировать значение на выходе с записью установленного значения в переменную

Синтаксис:

Void TCB_DECL InvertValue(PORT port, int channel, int& newValue)

Параметры:

Параметр	Тип	Диапазон значений	Примечание
port	PORT	По перчислению PORT	
channel	Int	0;1	
newValue	Int	0;1	

Пример вызова:

```
pBoard->InvertValue(PORT_O_ADDITIONAL_OUTPUT, 0, newValue);
```

Результат работы функции:

1. Будет инвертировано (изменено на противоположное) значение на выходе указанного порта на указанном канале.
2. В указанную переменную будет возвращено установленное значение.

4.2.11. Инвертировать значение на выходе

Синтаксис:

```
void        TCB_DECL InvertValue(PORT port, int channel)
```

Параметры:

Параметр	Тип	Диапазон значений	Примечание
port	PORT	По перчислению PORT	
channel	Int	0;1	

Пример вызова:

```
pBoard->InvertValue(PORT_O_ADDITIONAL_OUTPUT, 1);
```

Результат работы функции:

Будет инвертировано (изменено на противоположное) значение на выходе указанного порта на указанном канале.

4.2.12. Проверка смены значения

Синтаксис:

```
bool        TCB_DECL IsChangeValue(PORT port, int channel, int& newValue)
```

Параметры:

Параметр	Тип	Диапазон значений	Примечание
port	PORT	По перчислению PORT	
channel	Int	0;1	
newValue	Int	0;1	

Пример вызова:

```
if (pBoard->IsChangeValue(PORT_I_FIRE_SENSOR, 0, value))
        {
                // Значение изменилось
                if (value == 1)
                {
                        // Значение перешло в единицу (0->1)
                }
                else
                {
                        // Значение перешло в ноль (1->0)
        }
```

Результат работы функции:

1. Будет произведена проверка изменения значения на указанном порту на указанном канале.
2. Значение на канале будет записано в указанную переменную.

5. Заключение

Описанное выше API поставляется на условиях «как есть». Производитель не гарантирует полное соответствие предоставленной информации ожиданиям Пользователя.