# IZETModbus

interface IZETModbus

### Описание

Данный интерфейс определяет работу с сетью Modbus устройств.

## Препроцессорные макросы

### MAX\_PDU\_SIZE

Данный макрос определяет максимальный размер PDU пакета, Modbus протокола, в байтах.

#define MAX\_PDU\_SIZE 253

***Замечания***

PDU (Protocol Data Unit) – элементарный пакет протокола Modbus (см. спецификацию).

### MIN\_PDU\_SIZE

Данный макрос определяет минимальный размер PDU пакета, Modbus протокола, в байтах.

#define MIN\_PDU\_SIZE 1

***Замечания***

PDU (Protocol Data Unit) – элементарный пакет протокола Modbus (см. спецификацию).

### MAX\_ADU\_SIZE

Данный макрос определяет максимальный размер ADU пакета, Modbus протокола, в байтах.

#define MAX\_ADU\_SIZE (MAX\_PDU\_SIZE + 3)

***Замечания***

ADU (Application Data Unit) –пакет передачи данных по физическим линиям связи протокола Modbus (см. спецификацию).

### MODBUS\_ERR\_UNKNOWN\_ADDR

Номер ошибки протокола Modbus, которую посылает устройство, в случае, если запрашиваемый адрес регистра устройства не доступен.

#define MODBUS\_ERR\_UNKNOWN\_ADDR 0x02

***Замечания***

Нет.

### MODBUS\_ERR\_DEV\_BUSY

Номер ошибки протокола Modbus, которую посылает устройство, в случае, если оно занято и не может сформировать требуемый ответ.

#define MODBUS\_ERR\_DEV\_BUSY 0x06

***Замечания***

Нет.

## Методы интерфейса

### IZETModbus::ReadCoilStatus

Данный метод осуществляет получение значений регистров флагов устройства в сети Modbus.

virtual BYTE ReadCoilStatus(

BYTE dev\_addr,

WORD addr,

WORD\* pBuff,

unsigned int flags\_num

) = 0;

***Параметры***

dev\_addr

[in] – адрес устройства, значения регистров флагов которого, необходимо получить.

addr

[in] – стартовый адрес регистров флагов, значения которых необходимо получить.

pBuff

[out] – указатель на буфер, в который необходимо записать значения указанных регистров флагов.

flags\_num

[in] – количество флагов хранения, значения которых необходимо получить.

***Возвращаемые значения***

Данный метод возвращает код ошибки, которое вернуло устройство.

***Замечания***

Нет.

### IZETModbus::ReadDiscreteInputs

Данный метод осуществляет получение значений дискретных входов устройства в сети Modbus.

virtual BYTE ReadDiscreteInputs(

BYTE dev\_addr,

WORD addr,

WORD\* pBuff,

unsigned int inputs\_num

) = 0;

***Параметры***

dev\_addr

[in] – адрес устройства, значения дискретных входов которого, необходимо получить.

addr

[in] – стартовый адрес дискретных входов, значения которых необходимо получить.

pBuff

[out] – указатель на буфер, в который необходимо записать значения указанных дискретных входов.

inputs\_num

[in] – количество дискретных входов, значения которых необходимо получить.

***Возвращаемые значения***

Данный метод возвращает код ошибки, которое вернуло устройство.

***Замечания***

Нет.

### IZETModbus::ReadHoldingRegisters

Данный метод осуществляет получение значений регистров хранения устройства в сети Modbus.

virtual BYTE ReadHoldingRegisters(

BYTE dev\_addr,

WORD addr,

WORD\* pBuff,

unsigned int& regs\_num

) = 0;

***Параметры***

dev\_addr

[in] – адрес устройства, значения регистров хранения которого, необходимо получить.

addr

[in] – стартовый адрес регистров хранения, значения которых необходимо получить.

pBuff

[out] – указатель на буфер, в который необходимо записать значения указанных регистров хранения.

regs\_num

[in|out] – количество регистров хранения, значения которых необходимо получить. В данный параметр записывается количество успешно прочитанных регистров.

***Возвращаемые значения***

Данный метод возвращает код ошибки, которое вернуло устройство.

***Замечания***

Нет.

### IZETModbus::ReadInputRegisters

Данный метод осуществляет получение значений регистров ввода устройства в сети Modbus.

virtual BYTE ReadInputRegisters(

BYTE dev\_addr,

WORD addr,

WORD\* pBuff,

unsigned int &regs\_num

) = 0;

***Параметры***

dev\_addr

[in] – адрес устройства, значения регистров ввода которого, необходимо получить.

addr

[in] – стартовый адрес регистров ввода, значения которых необходимо получить.

pBuff

[out] – указатель на буфер, в который необходимо записать значения указанных регистров ввода.

regs\_num

[in|out] – количество регистров ввода, значения которых необходимо получить. В данный параметр записывается количество успешно прочитанных регистров.

***Возвращаемые значения***

Данный метод возвращает код ошибки, которое вернуло устройство.

***Замечания***

Нет.

### IZETModbus::ForceSingleCoil

Данный метод осуществляет запись регистроа флага устройства в сети Modbus.

virtual BYTE ForceSingleCoil(

BYTE dev\_addr,

WORD addr,

bool value

) = 0;

***Параметры***

dev\_addr

[in] – адрес устройства, содержимое регистра флага которого, необходимо изменить.

addr

[in] –адрес регистра флага, содержимое которого необходимо изменить.

value

[in] – значение, которое необходимо записать в указанный регистр флага.

***Возвращаемые значения***

Данный метод возвращает код ошибки, которое вернуло устройство.

***Замечания***

Нет.

### IZETModbus::PresetSingleRegister

Данный метод осуществляет запись одного регистра ввода устройства в сети Modbus.

virtual BYTE PresetSingleRegister(

BYTE dev\_addr,

WORD addr,

WORD value

) = 0;

***Параметры***

dev\_addr

[in] – адрес устройства, содержимое регистра ввода которого, необходимо изменить.

addr

[in] –адрес регистра ввода, содержимое которого необходимо изменить.

value

[in] – значение, которое необходимо записать в указанный регистр ввода.

***Возвращаемые значения***

Данный метод возвращает код ошибки, которое вернуло устройство.

***Замечания***

Нет.

### IZETModbus::ForceMultipleCoils

Данный метод осуществляет запись регистров флагов устройства в сети Modbus.

virtual BYTE ForceMultipleCoils(

BYTE dev\_addr,

WORD addr,

WORD\* pBuff,

unsigned int flags\_num

) = 0;

***Параметры***

dev\_addr

[in] – адрес устройства, содержимое регистров флагов которого, необходимо изменить.

addr

[in] – стартовый адрес регистров флагов, содержимое которых необходимо изменить.

pBuff

[in] – указатель на буфер, содержащий данные, которые необходимо записать в указанные регистры флагов.

flags\_num

[in] – количество регистров флагов, значения которых необходимо изменить.

***Возвращаемые значения***

Данный метод возвращает код ошибки, которое вернуло устройство.

***Замечания***

Нет.

### IZETModbus::PresetMultipleRegisters

Данный метод осуществляет запись регистров ввода устройства в сети Modbus.

virtual BYTE PresetMultipleRegisters(

BYTE dev\_addr,

WORD addr,

WORD\* pBuff,

unsigned int &regs\_num

) = 0;

***Параметры***

dev\_addr

[in] – адрес устройства, содержимое регистров ввода которого, необходимо изменить.

addr

[in] – стартовый адрес регистров ввода, содержимое которых необходимо изменить.

pBuff

[in] – указатель на буфер, содержащий данные, которые необходимо записать в указанные регистры ввода.

regs\_num

[in|out] – количество регистров ввода, значения которых необходимо изменить. В данный параметр устанавливается количество успешно записанных регистров.

***Возвращаемые значения***

Данный метод возвращает код ошибки, которое вернуло устройство.

***Замечания***

Нет.

# СZETModbus

class CZETModbus : public IZETModbus

### Описание

Данный класс представляет собой описание объекта, который реализует интерфейс IZETModbus. Данный класс является абстрактным. Для использования объектов класса, наследованных от данного класса, необходимо переопределить методы и .

## Методы интерфейса IZETModbus

### CZETModbus::ReadCoilStatus

Данный метод осуществляет получение значений регистров флагов устройства в сети Modbus.

virtual BYTE ReadCoilStatus(

BYTE dev\_addr,

WORD addr,

WORD\* pBuff,

unsigned int flags\_num

) = 0;

***Параметры***

dev\_addr

[in] – адрес устройства, значения регистров флагов которого, необходимо получить.

addr

[in] – стартовый адрес регистров флагов, значения которых необходимо получить.

pBuff

[out] – указатель на буфер, в который необходимо записать значения указанных регистров флагов.

flags\_num

[in] – количество флагов хранения, значения которых необходимо получить.

***Возвращаемые значения***

Данный метод возвращает код ошибки, которое вернуло устройство.

***Замечания***

Нет.

### CZETModbus::ReadDiscreteInputs

Данный метод осуществляет получение значений дискретных входов устройства в сети Modbus.

virtual BYTE ReadDiscreteInputs(

BYTE dev\_addr,

WORD addr,

WORD\* pBuff,

unsigned int inputs\_num

) = 0;

***Параметры***

dev\_addr

[in] – адрес устройства, значения дискретных входов которого, необходимо получить.

addr

[in] – стартовый адрес дискретных входов, значения которых необходимо получить.

pBuff

[out] – указатель на буфер, в который необходимо записать значения указанных дискретных входов.

inputs\_num

[in] – количество дискретных входов, значения которых необходимо получить.

***Возвращаемые значения***

Данный метод возвращает код ошибки, которое вернуло устройство.

***Замечания***

Нет.

### CZETModbus::ReadHoldingRegisters

Данный метод осуществляет получение значений нескольких регистров хранения.

BYTE ReadHoldingRegisters(

BYTE dev\_addr,

WORD addr,

WORD\* pBuff,

unsigned int &regs\_num

);

***Параметры***

dev\_addr

[in] – адрес устройства, значения регистров хранения которого, необходимо получить.

addr

[in] – адрес первого регистра хранения, начиная с которого, необходимо получить значения.

pBuff

[out] – указатель на буфер, в который необходимо записать полученные значения регистров.

regs\_num

[in|out] – количество регистров хранения, значения которых необходимо получить.

***Возвращаемые значения***

В качестве возвращаемой величины является код ошибки, которую вернуло устройство, 0x00 – значения регистров успешно получены. Количество успешно прочитанных регистров записывается в параметр regs\_num.

***Замечания***

В устройстве ведется словная адресация, таким образом, в результате выполнения метода в буфер pBuff будет записано regs\_num\*2 байт информации, соответствующей значениям указанным регистрам хранения. В случае возникновения ошибки при выполнении функции вызывается исключительная ситуация.

### CZETModbus::ReadInputRegisters

Данный метод осуществляет получение значений нескольких регистров ввода.

BYTE ReadInputRegisters(

BYTE dev\_addr,

WORD addr,

WORD\* pBuff,

unsigned int &regs\_num

);

***Параметры***

dev\_addr

[in] – адрес устройства, значения регистров ввода которого, необходимо получить.

addr

[in] – адрес первого регистра ввода, начиная с которого, необходимо получить значения.

pBuff

[in] – указатель на буфер, в который необходимо записать полученные значения регистров.

regs\_num

[in|out] – количество регистров ввода, значения которых необходимо получить.

***Возвращаемые значения***

В качестве возвращаемой величины является код ошибки, которую вернуло устройство, 0x00 – значения регистров успешно получены. Количество успешно прочитанных регистров записывается в параметр regs\_num.

***Замечания***

В устройстве ведется словная адресация, таким образом, в результате выполнения метода в буфер pBuff будет записано regs\_num\*2 байт информации, соответствующей значениям указанным регистрам ввода. В случае если количество регистров задано некорректно вызывается исключительная ситуация.

### CZETModbus::ForceSingleCoil

Данный метод осуществляет запись регистроа флага устройства в сети Modbus.

virtual BYTE ForceSingleCoil(

BYTE dev\_addr,

WORD addr,

bool value

) = 0;

***Параметры***

dev\_addr

[in] – адрес устройства, содержимое регистра флага которого, необходимо изменить.

addr

[in] –адрес регистра флага, содержимое которого необходимо изменить.

value

[in] – значение, которое необходимо записать в указанный регистр флага.

***Возвращаемые значения***

Данный метод возвращает код ошибки, которое вернуло устройство.

***Замечания***

Нет.

### CZETModbus::PresetSingleRegister

Данный метод осуществляет запись одного регистра ввода устройства в сети Modbus.

virtual BYTE PresetSingleRegister(

BYTE dev\_addr,

WORD addr,

WORD value

);

***Параметры***

dev\_addr

[in] – адрес устройства, содержимое регистра ввода которого, необходимо изменить.

addr

[in] –адрес регистра ввода, содержимое которого необходимо изменить.

value

[in] – значение, которое необходимо записать в указанный регистр ввода.

***Возвращаемые значения***

Данный метод возвращает код ошибки, которое вернуло устройство.

***Замечания***

Нет.

### CZETModbus::ForceMultipleCoils

Данный метод осуществляет запись регистров флагов устройства в сети Modbus.

virtual BYTE ForceMultipleCoils(

BYTE dev\_addr,

WORD addr,

WORD\* pBuff,

unsigned int flags\_num

) = 0;

***Параметры***

dev\_addr

[in] – адрес устройства, содержимое регистров флагов которого, необходимо изменить.

addr

[in] – стартовый адрес регистров флагов, содержимое которых необходимо изменить.

pBuff

[in] – указатель на буфер, содержащий данные, которые необходимо записать в указанные регистры флагов.

flags\_num

[in] – количество регистров флагов, значения которых необходимо изменить.

***Возвращаемые значения***

Данный метод возвращает код ошибки, которое вернуло устройство.

***Замечания***

Нет.

### CZETModbus::PresetMultipleRegisters

Данный метод осуществляет запись регистров ввода устройства в сети Modbus.

virtual BYTE PresetMultipleRegisters(

BYTE dev\_addr,

WORD addr,

WORD\* pBuff,

unsigned int &regs\_num

);

***Параметры***

dev\_addr

[in] – адрес устройства, содержимое регистров ввода которого, необходимо изменить.

addr

[in] – стартовый адрес регистров ввода, содержимое которых необходимо изменить.

pBuff

[in] – указатель на буфер, содержащий данные, которые необходимо записать в указанные регистры ввода.

regs\_num

[in|out] – количество регистров ввода, значения которых необходимо изменить.

***Возвращаемые значения***

Данный метод возвращает код ошибки, которое вернуло устройство. Количество успешно записанных регистров записывается в переменную regs\_num.

***Замечания***

Нет.

## Внешние методы

### CZETModbus::CZETModbus

Конструктор класса.

CZETModbus(void);

***Параметры***

Нет.

***Замечания***

Нет.

### CZETModbus::~CZETModbus

Деструктор класса.

virtual ~CZETModbus(void);

***Замечания***

Нет.

### CZETModbus::GetHierarchicalChain

Данный метод осуществляет получение иерархической цепочки для устройства с указанным адресом.

BYTE GetHierarchicalChain(

BYTE dev\_addr,

BYTE\* pBuff

);

***Параметры***

dev\_addr

[in] – адрес устройства, иерархическую вложенность которого необходимо получить.

pBuff

[out] – указатель на буфер, в который необходимо записать адреса устройств иерархической цепочки в порядке вложенности.

***Возвращаемые значения***

Данный метод количество устройств, которые являются родительскими по отношению к указанному устройству.

***Замечания***

Нет.

## Защищенные методы

### CZETModbus::Recv

Чисто виртуальный метод, который осуществляет прием данных в конкретной реализации интерфейса .

virtual void Recv(

BYTE\* pBuff,

unsigned int num\_bytes

) = 0;

***Параметры***

pBuff

[in] –буфер приема данных.

num\_bytes

[in] –количество принимаемых байт данных.

***Возвращаемые значения***

Нет.

***Замечания***

Нет.

### CZETModbus::Send

Чисто виртуальный метод, который осуществляет передачу данных в конкретной реализации интерфейса .

virtual void Send(

BYTE\* pBuff,

unsigned int num\_bytes

) = 0;

***Параметры***

pBuff

[in] –буфер данных для передачи.

num\_bytes

[in] –количество байт данных для передачи.

***Возвращаемые значения***

Нет.

***Замечания***

Нет.

### CZETModbus::SendCommandPDU

Данный метод осуществляет отправку команды (PDU пакета) по протоколу Modbus.

void SendCommandPDU(

BYTE dev\_addr,

BYTE\* pPDU

);

***Параметры***

dev\_addr

[in] – адрес устройства, которому необходимо отправить указанную команду.

pPDU

[in] –указатель на буфер, содержащий команду для отправки (PDU пакет).

***Возвращаемые значения***

Нет.

***Замечания***

В случае ошибки при передаче команды возбуждается исключительная ситуация.

### CZETModbus::GetResponsePDU

Данный метод осуществляет получение ответа (PDU пакета) по текущему каналу связи на ранее посланную команду.

BYTE GetResponsePDU(

BYTE\* pPDU,

BYTE PDU\_len

);

***Параметры***

pPDU

[out] – указатель на буфер, в который необходимо записать полученный ответ.

PDU\_len

[out] – данный параметр определяет количество байт ответа.

***Возвращаемые значения***

Метод возвращает адрес устройства, от которого получен ответ. В буфер pPDU записывается ответ от устройства (PDU пакет), полученный по текущему каналу связи.

***Замечания***

Алгоритм получения ответа следующий:

* Из канала получается первые три байта, первый из которых определяет адрес устройства, пославшего ответ, а второй - номер функции, для запроса которой ожидается ответ;
* Если номер функции содержит ошибочный бит, то длина PDU пакета определяется равной 2, в противном случае длина пакета определяется исходя значений второго и третьего байтов (для каждой команды определяется индивидуально);
* В конце читаются количество байт, определяемое длиной PDU пакета.

В случае ошибки при приме ответа возбуждается исключительная ситуация.

### CZETModbus::CheckSumm

Данный метод осуществляет подсчет контрольной суммы.

WORD CheckSumm(

BYTE\* pBuff,

unsigned int len

);

***Параметры***

pBuff

[in] – указатель на буфер, для содержимого которого необходимо вычислить контрольную сумму.

len

[in] – размер буфера в байтах, для содержимого которого необходимо вычислить контрольную сумму.

***Возвращаемые значения***

В качестве возвращаемой величины является контрольная сумма содержимого указанного буфера данных.

***Замечания***

Нет.

# ModbusException

class ModbusException : public Exception

### Описание

Данный класс предназначен для обработки ошибок, которые могут происходить во время работы с интерфейсом .

## Внешние методы

### ModbusException::ModbusException

Конструктор класса.

ModbusException(

unsigned int err

);

***Параметры***

err

[in] – Номер ошибки, которая привела к исключительной ситуации.

***Замечания***

Нет.

## Защищенные методы

## Внутренние методы

## Атрибуты

## Список ошибок

1. Неизвестная ошибка.
2. Попытка отправить команду неизвестного типа (номер функции не определен).
3. Попытка отправки некорректного PDU пакета.
4. Принят не корректный ADU пакет.
5. Принят ответ от устройства, адрес которого не совпадает с адресом устройству, которому была послана команда.
6. Код принятого ответа от устройства не совпадает с кодом отправленной команды.
7. При попытке записи одного регистра ввода (PresetSingleRegister()) устройства тело команды не совпадает с телом ответа.
8. S
9. S
10. Попытка прочитать из устройства (или записать в устройство) (ReadHoldingRegisters(), ReadInputRegisters(), PresetMultipleRegisters()) такое количество регистров, которое не может поместиться в PDU пакет.
11. Попытка отправить PDU пакет не правильного размера.
12. Произошла ошибка при попытке отправки ADU пакета по текущему каналу.
13. 2
14. 2
15. 2
16. Произошла ошибка при получении ADU пакета – ошибочная контрольная сумма в принятом пакете.

Оглавление

[IZETModbus 1](#_Toc285614742)

[Описание 1](#_Toc285614743)

[Препроцессорные макросы 1](#_Toc285614744)

[MAX\_PDU\_SIZE 1](#_Toc285614745)

[MIN\_PDU\_SIZE 1](#_Toc285614746)

[MAX\_ADU\_SIZE 1](#_Toc285614747)

[MODBUS\_ERR\_UNKNOWN\_ADDR 1](#_Toc285614748)

[MODBUS\_ERR\_DEV\_BUSY 1](#_Toc285614749)

[Методы интерфейса 1](#_Toc285614750)

[IZETModbus::ReadCoilStatus 1](#_Toc285614751)

[IZETModbus::ReadDiscreteInputs 2](#_Toc285614752)

[IZETModbus::ReadHoldingRegisters 2](#_Toc285614753)

[IZETModbus::ReadInputRegisters 3](#_Toc285614754)

[IZETModbus::ForceSingleCoil 3](#_Toc285614755)

[IZETModbus::PresetSingleRegister 4](#_Toc285614756)

[IZETModbus::ForceMultipleCoils 4](#_Toc285614757)

[IZETModbus::PresetMultipleRegisters 4](#_Toc285614758)

[СZETModbus 5](#_Toc285614759)

[Описание 5](#_Toc285614760)

[Методы интерфейса IZETModbus 5](#_Toc285614761)

[CZETModbus::ReadCoilStatus 5](#_Toc285614762)

[CZETModbus::ReadDiscreteInputs 6](#_Toc285614763)

[CZETModbus::ReadHoldingRegisters 6](#_Toc285614764)

[CZETModbus::ReadInputRegisters 7](#_Toc285614765)

[CZETModbus::ForceSingleCoil 7](#_Toc285614766)

[CZETModbus::PresetSingleRegister 8](#_Toc285614767)

[CZETModbus::ForceMultipleCoils 8](#_Toc285614768)

[CZETModbus::PresetMultipleRegisters 8](#_Toc285614769)

[Внешние методы 9](#_Toc285614770)

[CZETModbus::CZETModbus 9](#_Toc285614771)

[CZETModbus::~CZETModbus 9](#_Toc285614772)

[CZETModbus::GetHierarchicalChain 9](#_Toc285614773)

[Защищенные методы 10](#_Toc285614774)

[CZETModbus::Recv 10](#_Toc285614775)

[CZETModbus::Send 10](#_Toc285614776)

[CZETModbus::SendCommandPDU 10](#_Toc285614777)

[CZETModbus::GetResponsePDU 11](#_Toc285614778)

[CZETModbus::CheckSumm 11](#_Toc285614779)

[ModbusException 11](#_Toc285614780)

[Описание 11](#_Toc285614781)

[Внешние методы 12](#_Toc285614782)

[ModbusException::ModbusException 12](#_Toc285614783)

[Защищенные методы 12](#_Toc285614784)

[Внутренние методы 12](#_Toc285614785)

[Атрибуты 12](#_Toc285614786)

[Список ошибок 12](#_Toc285614787)