

ИКС-Маркет

Программное обеспечение

Универсальный драйвер (OLE) для
весовых терминалов Mettler Toledo Hawk, ИКС К,
весов ICS NT, Mettler Toledo Tiger E, Jaguar
(Компоненты AddIn.Hawk, AddIn.Ikc_k,
AddIn.Ics_nt, AddIn.Jaguar)

Руководство программиста

Содержание

Поставка драйвера.....	3
Описание компоненты AddIn.Hawk.....	4
Описание компоненты AddIn.Ikc_k.....	6
Описание компоненты AddIn.Ics_nt.....	7
Описание компоненты AddIn.Jaguar.....	9

Поставка драйвера

Универсальный OLE-драйвер состоит из следующих файлов:

ICSdrv.dll – модуль драйвера;

ICSdrv.tlb – библиотека типов драйвера.

ВНИМАНИЕ!

Во время инсталляции регистрация драйвера в реестре Windows производится автоматически.

Зарегистрировать драйвер в ручном режиме можно с помощью команды, выполняемой из командной строки:

C:\<Путь к модулю драйвера> regsvr32.exe ICSdrv.dll

Удалить регистрационную информацию из реестра можно с помощью команды:

C:\<Путь к модулю драйвера> regsvr32.exe -u ICSdrv.dll

В комплекте с драйвером дополнительно поставляются тестовые примеры обработок «1С:Предприятие» версий 7.7, 8.1, 8.2 для работы с весовыми терминалами Mettler Toledo Hawk, ИКС К, весами ICS NT, Mettler Toledo Tiger E:

ICSdrv_77.ert – модуль обработки «1С:Предприятие» версии 7.7.

ICSdrv_81.epf – модуль обработки «1С:Предприятие» версии 8.1.

ICSdrv_82.epf – модуль обработки «1С:Предприятие» версии 8.2.

Пути установки по умолчанию:

C:\Program Files\ICS Market\ICSdrv\ICSdrv.dll

C:\Program Files\ICS Market\ICSdrv\Hawk.dll

C:\Program Files\ICS Market\ICSdrv\Ikc_k.dll

C:\Program Files\ICS Market\ICSdrv\Ics_nt.dll

C:\Program Files\ICS Market\ICSdrv\Jaguar.dll

C:\Program Files\ICS Market\ICSdrv\ICSdrv.tlb

C:\Program Files\ICS Market\ICSdrv\Manual.pdf

C:\Program Files\ICS Market\ICSdrv\ICSdrv_77.ert

C:\Program Files\ICS Market\ICSdrv\ICSdrv_81.epf

C:\Program Files\ICS Market\ICSdrv\ICSdrv_82.epf

Описание компоненты «AddIn.Hawk»

Компонента «AddIn.Hawk» предназначена для работы с весовым терминалом Mettler Toledo Hawk.

Методы компоненты «AddIn.Hawk»

1. Метод Init Настроить и открыть коммуникационный порт

```
int init(COM,BAUD);
```

Параметры:

<i>Параметр</i>	<i>Тип</i>	<i>Описание</i>
COM	int	Номер коммуникационного порта
BAUD	int	Скорость передачи данных (2400, 4800, 9600) бит/с

Возвращаемое значение: 0 – порт открыт, 1 – неудачная попытка открыть порт.

Примечание:

Для осуществления корректной передачи данных, весовой терминал HAWK необходимо настроить следующим образом:

Формат вывода данных:	По запросу, однострочный, только отображаемый вес;
Скорость передачи данных:	2400, 4800, 9600 бит/с;
Биты данных:	8 бит;
Стоповые биты:	1 бит;
Контроль четности:	Нет.

2. Метод Tare Тарирование терминала

```
int tare();
```

Параметры: Нет

Возвращаемое значение: 0 – тарирование произведено успешно, 1 – ошибка тарирования.

3. Метод Clear Очистка тары

```
int clear();
```

Параметры: Нет

Возвращаемое значение: 0 – очистка произведена успешно, 1 – ошибка очистки тары.

4. Метод Zero Обнуление терминала

```
int zero();
```

Параметры: Нет

Возвращаемое значение: 0 – обнуление произведено успешно, 1 – ошибка обнуления.

5. Метод Get_weight Чтение веса терминала

```
double get_weight();
```

Параметры: Нет

Возвращаемое значение: Значение веса в килограммах.

Примечание:

В случае, если значение веса не было передано терминалом, например по причине нестабильности платформы, возвращаемое значение будет равно нулю.

6. Метод Uninit Закрывать коммуникационный порт

```
int uninit();
```

Параметры: Нет

Возвращаемое значение: 0 – Коммуникационный порт закрыт успешно, 1 – ошибка закрытия коммуникационного порта.

Описание компоненты «AddIn.Ikc_k»

Компонента «AddIn.Ikc_k» предназначена для работы с весовым терминалом ИКС К.

Методы компоненты «AddIn.Ikc_k»

1. Метод Init Настроить и открыть коммуникационный порт

```
int init(COM,BAUD);
```

Параметры:

<i>Параметр</i>	<i>Тип</i>	<i>Описание</i>
COM	int	Номер коммуникационного порта
BAUD	int	Скорость передачи данных (2400, 4800, 9600) бит/с

Возвращаемое значение: 0 – порт открыт, 1 – неудачная попытка открыть порт.

Примечание:

Для осуществления корректной передачи данных, весовой терминал ИКС К необходимо настроить в режим ручной либо автоматической печати.

2. Метод Get_weight Чтение веса терминала

```
double get_weight();
```

Параметры: Нет

Возвращаемое значение: Значение веса в килограммах.

Примечание:

В случае, если значение веса не было передано терминалом, например по причине нестабильности платформы, возвращаемое значение будет равно нулю.

3. Метод Uninit Закрывать коммуникационный порт

```
int uninit();
```

Параметры: Нет

Возвращаемое значение: 0 – Коммуникационный порт закрыт успешно, 1 – ошибка закрытия коммуникационного порта.

Описание компоненты «AddIn.Ics_nt»

Компонента «AddIn.Ics_nt» предназначена для работы с весами ICS NT и Mettler Toledo Tiger E.

Методы компоненты «AddIn.Ics_nt»

1. Метод Init Настроить и открыть коммуникационный порт

```
int init(COM);
```

Параметры:

<i>Параметр</i>	<i>Тип</i>	<i>Описание</i>
COM	int	Номер коммуникационного порта

Возвращаемое значение: 0 – порт открыт, 1 – неудачная попытка открыть порт.

2. Метод Tare Тарирование весов

```
int tare();
```

Параметры: Нет

Возвращаемое значение: 0 – тарирование произведено успешно, 1 – ошибка тарирования.

3. Метод Set_price Передача цены на весы

```
int set_price(price);
```

Параметры:

<i>Параметр</i>	<i>Тип</i>	<i>Описание</i>
price	int	Цена в копейках

Возвращаемое значение: 0 – цена передана успешно, 1 – неудачная попытка передачи цены.

4. Метод Get_weight Чтение веса

```
int get_weight();
```

Параметры: Нет

Возвращаемое значение: Значение веса в граммах.

Примечания:

В случае, если значение веса не было передано весами, например по причине нестабильности платформы, возвращаемое значение будет равно нулю.

После получения ненулевого веса, необходимо произвести чтение свойств price и total.

5. Метод Uninit Закрывает коммуникационный порт

```
int uninit();
```

Параметры: Нет

Возвращаемое значение: 0 – Коммуникационный порт закрыт успешно, 1 – ошибка закрытия коммуникационного порта.

Свойства компоненты «AddIn.Ics_nt»

<i>Свойство</i>	<i>Тип</i>	<i>Доступ</i>	<i>Описание</i>
price	int	R	Цена в копейках. Содержит значение после успешного выполнения метода get_weight()
total	int	R	Стоимость в копейках. Содержит значение после успешного выполнения метода get_weight()

Описание компоненты «AddIn.Jaguar»

Компонента «AddIn.Jaguar» предназначена для работы с весами Jaguar.

Методы компоненты «AddIn.Jaguar»

1. Метод Init Настроить и открыть коммуникационный порт

```
int init(COM,BAUD);
```

Параметры:

<i>Параметр</i>	<i>Тип</i>	<i>Описание</i>
COM	int	Номер коммуникационного порта
BAUD	int	Скорость передачи данных,бит/с

Возвращаемое значение: 0 – порт открыт, 1 – неудачная попытка открыть порт.

Примечание:

Для осуществления корректной передачи данных, весы Jaguar необходимо настроить следующим образом:

Скорость передачи данных: 2400, 4800, 9600 (по умолчанию), 19200, 28800, 38400, 57600 бит/с;
Биты данных: 8 бит;
Стоповые биты: 1 бит;
Контроль четности: Четный (Even).

2. Метод set_name Установка наименования товара

```
int set_name(scalenum, name);
```

Параметры:

<i>Параметр</i>	<i>Тип</i>	<i>Описание</i>
scalenum	int	Сетевой номер весов
name	BSTR	Наименование товара

Возвращаемое значение: 0 – установка произведена успешно, 1 – ошибка установки.

Примечание:

Весами поддерживаются, латинский и украинский шрифты. Поддержка русского шрифта отсутствует.

3. Метод Get_weight Чтение веса

```
double get_weight();
```

Параметры:

<i>Параметр</i>	<i>Тип</i>	<i>Описание</i>
scalenum	int	Сетевой номер весов

Возвращаемое значение: Значение веса в килограммах.

Примечание:

В случае, если значение веса не было передано весами, например по причине нестабильности платформы, возвращаемое значение будет равно нулю.

4. Метод Uninit Закрыть коммуникационный порт

```
int uninit();
```

Параметры: Нет

Возвращаемое значение: 0 – Коммуникационный порт закрыт успешно, 1 – ошибка закрытия коммуникационного порта.