

Для библиотечных задач

Более подробно о применении «Wonderfid™ Link» для библиотечных задач читайте в разделе [«UHF RFID для библиотек»](#).

Стандарт ISO 28560 RFID в библиотеках предусматривает RFID-учет всех библиотечных объектов. С помощью RFID в рамках стандарта можно учитывать:

1. библиотечный фонд – книги, журналы, диски и т.п., выдаваемые абонементам;
2. читательские билеты (метка либо вклеивается в билет, либо сам билет заменяется RFID-карточкой);
3. собственное библиотечное имущество, не выдаваемое абонементам (столы, шкафы и т.п.);
4. товары на продажу;
5. списанные объекты и объекты, ожидающие утилизации.

«Wonderfid™ Link» поддерживает всё из вышеперечисленного.

Цифровое кодирование UII

В RFID-метку UII записывается при помощи нулей и единиц. Перевод UII в нули и единицы называется бинарным кодированием UII, которое уже реализовано в продукте (самим ничего кодировать и декодировать не нужно). Самый распространенный способ записи UII – это строка, представляющая собой последовательную запись в 16-ричном формате всех байт бинарно закодированного UII, и именно в таком виде UII отображают программы, которые идут с RFID-оборудованием по умолчанию.

Строки вида «3024000003320C4063A23312», которые выдаются демопрограммой считывателя как TAG ID считанных RFID-меток, требуют декодирования в EPC или UII. Только тогда можно узнать номер библиотечного объекта, на который нанесена метка, и т.п. Все стандартные схемы кодирования/декодирования уже реализованы в «Wonderfid™ Link», ничего самим писать не нужно.

Пример декодирования UII при помощи «Wonderfid™ Link»:

```
uII = КлеверенсRFID.ДекодироватьEPCUII("608940C09996D7A00000");
```

Пример кодирования UII при помощи «Wonderfid™ Link»:

```
строка = КлеверенсRFID.UIIизБиблиотечногоКода(Неопределено, "10054123").БинарноеПредставление;
```

Общий алгоритм маркировки

Поскольку метки прошиваются конкретным библиотечным кодом, все их следует прошивать по очереди. Наиболее удобный способ – сначала оптом обклеить интересующие объекты «непрошитыми» метками, а затем по одному прошить уникальным кодом.

Пример алгоритма маркировки:

```

// получить с сервера используемый пароль на доступ к RFID-меткам
парольНаДоступ = ПолучитьПарольНаДоступRFID();
Пока Истина Цикл
// Заставить пользователя выбрать из базы конкретный объект фонда, читательский билет и т.п.
// если выбранному объекту уже сопоставлена метка – переспросить пользователя
// (например, метка могла выйти из строя и действительно требуется перемаркировка)
маркируемыйОбъект = ВыбратьЭкземпляр();
Если маркируемыйОбъект = Неопределено Тогда
    Возврат;
КонецЕсли;

режим = РежимДиалогаВопрос.ОКОтмена;
выбраннаяМетка = Неопределено;
Пока выбраннаяМетка = Неопределено Цикл
    ответ = Неопределено;
    метки = Неопределено;
    // Поискать вокруг антенны RFID-метки в течение 1й секунды (1000 миллисекунд)
    Попытка
        // Читаем не только Tag ID, но и банк TID меток, чтобы потом писать в конкретную.
        метки = считыватель.ПрочитатьМетки (5000, Истина, Ложь, Ложь);
    Исключение
        Вопрос("Ошибка поиска меток! " + КлеверенсRFID.ОписаниеОшибки(), РежимДиалогаВопрос.ОК);
        Продолжить;
    КонецПопытки;

    Если метки.Количество = 0 Тогда
        ответ = Вопрос("Положите маркируемый объект на антенну!", режим);
    Иначе
        Если метки.Количество > 0 Тогда
            ответ = Вопрос("Уберите от антенны посторонние предметы!", режим);
        Иначе
            // Выбрать единственную метку
            выбраннаяМетка = метки.Элемент(0);
        КонецЕсли;
    КонецЕсли;

    Если ответ = КодВозвратаДиалога.Отмена Тогда
        Прервать;
    КонецЕсли;
КонецЦикла;

Попытка
    // Создать UII в соответствии с тем, какой объект выбрали:
    uiI = СоздатьПравильныйUII(маркируемыйОбъект);
    // Записать UII
    считыватель.ЗаписатьEPCUIIпоTID(выбраннаяМетка.TagId, выбраннаяМетка.TID, uiI, парольНаДоступ);

    Сообщить("В метку с Tag ID [" + выбраннаяМетка.TagId + "] успешно записан новый UII [" +
        uiI.Строка() + "] (" + uiI.БинарноеПредставление + ")");
Исключение
    Предупреждение("Ошибка записи в метку! " + КлеверенсRFID.ОписаниеОшибки());
КонецПопытки;
КонецЦикла;

```

Маркировка библиотечного фонда

Пример №1. если у библиотеки нет ISIL

```

// если у библиотеки нет ISIL, то можно передать Неопределено
uiI = КлеверенсRFID.UIIизБиблиотечногоКода(Неопределено, экземпляр.Код);

```

Пример №2. если у библиотеки есть ISIL:

```

uiI = КлеверенсRFID.UIIизБиблиотечногоКода(ISIL, экземпляр.Код);

```

Пример №3. если не жалко памяти RFID-метки:

```

uiI = КлеверенсRFID.UIIизБиблиотечногоКода(ISIL, экземпляр.Код);
// если память метки позволяет, то можно проставить в UII тип использования для объекта
uiI.ТипИспользования = КлеверенсRFID.Библиотеки.ТипыИспользования.ДляВыдачи;

```

UII следует записать в банк EPCUII.

Если в метке очень много памяти, то в банк USER можно записать дополнительную информацию об объекте фонда.

Пример №4. Прошивка в банк USER наименования книги, номера тома и места на полке:

```
// получить с сервера используемый пароль на доступ к RFID-меткам
парольНаДоступ = ПолучитьПарольНаДоступRFID();

бо = КлеверенсRFID.СоздатьБиблиотечныйОбъект();
бо.Наименование = "Л. Н. Толстой. Война и Мир, том 1й";
бо.РазмерНабора = 4; // 4 тома
бо.НомерВНаборе = 1; // 1й том
бо.МестоНаПолке = "А-14-21";

банк = бо.СформироватьUSERБанк();
// заполненные выше данные займут ровно 74 байта памяти банка USER
// метки с банком памяти USER < 74 байта не смогут быть прошитыми
считыватель.ЗаписатьUSER(метка.TagId, банк, парольНаДоступ);
```

Данные из приведенного примера займут ровно 74 байта банка памяти USER. Самые бюджетные метки в настоящий момент имеют всего 32 бита памяти USER и, соответственно, не смогут быть использованы в таком сценарии.

Маркировка читательских билетов (и RFID-карточек)

Пример №1. Формирование UИИ для читательского билета:

```
// если у библиотеки нет ISIL, то можно передать Неопределено
uИИ = КлеверенсRFID.UИИизБиблиотечногоКода(ISIL, читатель.Код, КлеверенсRFID.AFI.Библиотечный);
uИИ.ТипИспользования = КлеверенсRFID.Библиотеки.ТипыИспользования.ЧитательскийБилет;
```

Пример №2. Проверка, что тип использования у метки – любой читательский билет:

```
Если метка.Объект.Тип() = "БиблиотечныйКод" И метка.Объект.ТипИспользования <> Неопределено И
метка.Объект.ТипИспользования.КодКласса = КлеверенсRFID.Библиотеки.ТипыИспользования.ЧитательскийБилет.КодКласса
Тогда
```

Если требуется по логике и позволяет память метки, то можно прошить в банк USER некие дополнительные данные о держателе билета.

Пример №3. Прошивка в банк USER имени владельца билета:

```
// получить с сервера используемый пароль на доступ к RFID-меткам
парольНаДоступ = ПолучитьПарольНаДоступRFID();

бо = КлеверенсRFID.СоздатьБиблиотечныйОбъект();
бо.Наименование = читатель.ФИО;

банк = бо.СформироватьUSERБанк();
считыватель.ЗаписатьUSER(метка.TagId, банк, парольНаДоступ);
```

Маркировка библиотечного имущества (столы и стулья)

Пример №1. Формирование UИИ для библиотечного имущества:

```
// если у библиотеки нет ISIL, то можно передать Неопределено
uИИ = КлеверенсRFID.UИИизБиблиотечногоКода(ISIL, имущество.Код, КлеверенсRFID.AFI.НаСкладе);
uИИ.ТипИспользования = КлеверенсRFID.Библиотеки.ТипыИспользования.Имущество;
```

Пример №2. Проверка, что тип использования у метки – любое имущество:

```
Если метка.Объект.Тип() = "БиблиотечныйКод" И метка.Объект.ТипИспользования <> Неопределено И
(метка.Объект.ТипИспользования.КодКласса = КлеверенсRFID.Библиотеки.ТипыИспользования.Имущество.КодКласса или
метка.Объект.ТипИспользования.КодКласса = КлеверенсRFID.Библиотеки.ТипыИспользования.НеДляВыдачи.КодКласса) Тогда
```

Была ли статья полезна?

Нет

Да