



Системы Модули и Компоненты

Библиотека MБee для Python

Описание



Board Revision	
Product Name	
Doc Name	sw_python_mbee
Revision Date	04.05.2018
Revision Number	2

1. ОГЛАВЛЕНИЕ

1. Оглавление	1
2. Назначение библиотеки python-mbee	2
2.1. Области применения	2
3. Установка библиотеки	3
3.1. Требования к оборудованию и программному обеспечению	3
3.2. Установка python-mbee	3
4. Работа с библиотекой	4
4.1. Подключение и настройка радиомодуля MВee.....	4
4.2. Работа с библиотекой.....	4
4.3. Поддерживаемые проекты.....	5
5. История документа	6
6. Техническая поддержка	7

2. НАЗНАЧЕНИЕ БИБЛИОТЕКИ PYTHON-MBEE

Библиотека предназначена для разработчиков программного обеспечения верхнего уровня для беспроводных сетей сбора данных и управления, основанных на радиомодулях производства фирмы «Системы, модули и компоненты». Используя *python-mbee* программист избавляется от необходимости глубокого изучения, встроенного ПО модулей. Пакет использует небольшое число внешних функций, разбитых на функциональные группы, что облегчает процедуру его освоения. Для всех членов библиотеки применяются осмысленные мнемоники, а название параметров соответствует названиям, используемым в техническом описании конкретного программного обеспечения. Подробная документация расположена в архиве, вместе с самой библиотекой ([\python-mbee-1.0\mbee\doc\html\index.html](#)) и содержит описание каждой публичной функции, исходные коды с комментариями и примеры. Применение библиотеки позволит программистам значительно сократить время входа в технологию и сосредоточиться на решении целевой задачи.

2.1. Области применения

Библиотека *python-mbee* предназначена для применения в том числе в таких областях, как:

1. Центральные контроллеры автономных систем управления и сбора данных (домашняя автоматизация, теплицы, управление освещением).
2. Программное обеспечение шлюзов, связывающих беспроводную часть сети с удаленными системами управления, обработки и накопления данных, для систем с распределенной структурой.
3. Шлюзы, предназначенные для связи беспроводной сети с Интернетом при реализации облачных решений.

3. УСТАНОВКА БИБЛИОТЕКИ

3.1. Требования к оборудованию и программному обеспечению

Библиотека *python-mbee* работоспособна как на операционных системах семейства *Windows*, так и различных *Linux* платформах. В качестве аппаратных средств могут использоваться обычные персональные компьютеры, промышленные контроллеры или разнообразные одноплатные решения типа популярного микрокомпьютера *Raspberry Pi*. Необходимым условием является наличие для выбранной платформы интерпретатора *Python* с версией ниже 3.4. Желательно также наличие пакета *pip*. Из внешних зависимостей потребуется установить пакет поддержки последовательного интерфейса *pySerial*.

3.2. Установка *python-mbee*

Пакет может быть установлен стандартными для *Python* способами: оффлайн установка с помощью файла *setup.py* и установка с помощью пакета *pip*, которая является более предпочтительной.

1. Оффлайн-установка на *Windows*-компьютерах.
 - 1.1 Скачайте архив пакета с *github* или *PyPi*, перейдя по ссылке, имеющейся на нашем сайте и распакуйте его в любом удобном месте.
 - 1.2 Запустите командный интерфейс, перейдите в каталог с пакетом и выполните следующую команду: *python setup.py install*. Для корректной установки может потребоваться запуск командной строки с правами администратора.
2. Оффлайн-установка на *Linux*-платформах.
 - 2.1 Скачайте архив пакета с *github* или *PyPi* в домашний каталог. Распакуйте его командой *tar xvzf python-mbee-1.0.tar.gz*.
 - 2.2 Перейдите в каталог с распакованным пакетом и выполните *python3 setup.py install*.
3. Установка *python-mbee* с помощью *pip*.

В командной строке отдайте команду *pip install python-mbee* для *Windows* (могут потребоваться права администратора) или *pip3 install python-mbee* для *Linux*.

4. Установка *pySerial*.

В командной строке отдайте команду *pip install pyserial* для *Windows* (могут потребоваться права администратора) или *pip3 install pyserial* для *Linux*.

4. РАБОТА С БИБЛИОТЕКОЙ

4.1. Подключение и настройка радиомодуля MBee

В качестве физического интерфейса связи с модулем используется последовательный интерфейс *UART* с уровнями цифровых сигналов 3,3В. При подключении к *COM*-порту необходимо использовать преобразователь уровней *RS232-UART*, например, плату *MB-serial*. Если предполагается соединение с помощью *USB*-интерфейса, то также потребуется соответствующий конвертер. Обычно в этом случае используется плата *MB-USBridge*. Кроме вышеупомянутых преобразователей производства «СМК», могут быть использованы аналогичные по функциям устройства сторонних производителей. Например, предназначенные для модулей *DIGI*, поскольку радиомодули «СМК» полностью совместимы с ними по расположению физических контактов.

Взаимодействие с модулем осуществляется с помощью обмена пакетами определенного формата. Перед первым подключением необходимо настроить модуль для работы в пакетном режиме. Для программного обеспечения *SerialStar*, например, необходимо с помощью любой терминальной программы в командном режиме последовательно отдать *AT*-команды «*AT AP 2 <CR>*» и «*AT CN<CR>*» (где *<CR>* символизирует нажатие на кнопку «*Enter*»).

4.2. Работа с библиотекой

Взаимодействие с радиомодулем может быть организовано двумя способами - периодической проверкой (*polling*) и с помощью функций обратного вызова (*callback*-функций).

Первый способ *применяется* как правило в сетях с архитектурой «звезда» с единственным ведущим узлом - Мастером. Весь процесс обмена в таких системах всегда инициируется Мастером, которому всегда известно пакет какого типа он должен получить в ответ на свой запрос. Вся процедура выглядит следующим образом:

- передача команды удаленному узлу.
- таймаут на ожидание ответа.
- опрос приемного буфера *UART* для получения от удаленного узла заданного типа ответного пакета.

Недостатком данного подхода *является* наличие таймаута, в течение которого основное приложение вынуждено бездействовать. Достоинством является простота последовательной реализации алгоритма.

Организация взаимодействия с *помощью* функций обратного вызова представляет собой наиболее универсальный подход. При таком подходе становится возможным строить более сложные сети в которых могут применяться ретрансляторы. Наличие ретрансляторов затрудняет определение таймаута, который потребовался бы Мастеру для получения ответа от удаленного узла. В общем случае этот таймаут будет различным для разных групп удаленных узлов, что еще более бы затруднило создание эффективного программного обеспечения. Только способ с *callback*-функциями делает возможным создание сети с несколькими ведущими узлами, когда невозможно точно предсказать момент приема пакета. В соответствии с этим подходом при разработке ПО рекомендуется придерживаться следующей последовательности действий:

- Определить типы пакетов, которые данный узел должен принимать и обрабатывать.
- Написать отдельную функцию обработки для каждого интересующего типа пакета.
- Зарегистрировать эти функции с помощью вызова соответствующей процедуры, имеющейся в *python-mbee*.
- В главном цикле скрипта необходимо включить вызов метода *run()*.
- Во избежание потерь пакетов не рекомендуется использовать большие задержки при реализации основного алгоритма.
- Организация основного цикла скрипта предпочтительна в виде конечного автомата.

Для более подробного знакомства с *callback*-подходом, рекомендуется обратиться к примеру, имеющемуся в документации.

4.3. Поддерживаемые проекты

Название проекта	Модуль	Дата
<i>SerialStar</i>	<i>MBee-868-x.0</i>	01.05.2018

Таблица 1

5. ИСТОРИЯ ДОКУМЕНТА

№	Дата	Редакция документа	Описание изменений
1	01.05.2018	Первая версия	-
2	04.05.2018	Текущая версия	Добавлена ссылка на подробное описание библиотеки.

Таблица 2 История документа.

6. ТЕХНИЧЕСКАЯ ПОДДЕРЖКА

Разработка и техническая поддержка	
СИСТЕМЫ, МОДУЛИ И КОМПОНЕНТЫ	
Разработчик систем автоматизации и телеметрии	
Телефон	+7 (495) 784 5766
Электронная почта	mbee@sysmc.ru
Сайт	www.sysmc.ru
	