

РЕАЛИЗАЦИЯ ВЗАИМОДЕЙСТВИЯ МИКРОКОНТРОЛЛЕРА ARDUINO И МИКРОКОМПЬЮТЕРА RASPBERRY PI3 С ИСПОЛЬЗОВАНИЕМ JSON

Мотузко Максим Александрович

УО «Брестский государственный университет имени А.С. Пушкина»

Для разработки системы управления и мониторинга автономного робота, созданного в научно-исследовательской лаборатории Брестского государственного университета, необходимо разработать некоторый интерфейс. Интерфейс должен предоставить возможность обмена данными между устройствами разного типа. Так же интерфейс должен быть оптимальным, скоростным и легко масштабируемым. Нами исследованы различные варианты создания такого интерфейса.

В настоящей статье описано одно из решений. Ниже представлены средства, программное обеспечение и технологии использованные для реализации взаимодействия двух устройств, установленных на автономную мобильную платформу. Автономная платформа включает микроконтроллер Arduino Nano и микрокомпьютер Raspberry Pi3.

В нашем решении Arduino и Raspberry физически связаны по средствам UART протокола. В ходе испытаний именно этот прокол обеспечил наиболее быструю и стабильную передачу данных. Протокол достаточно популярный (см., например, [1], [2], [3] и др.), вследствие чего можно легко найти необходимые библиотеки для работы с ним, причём на для большого числа языков программирования. В нашем случае, на стороне Arduino используется библиотека Serial написанная на языке C++. На стороне Raspberry программный код написан на языке Python с использованием библиотеки PySerial. Выбор концепции реализации и языковых средств и библиотек –это итог первого этапа исследования.

На втором этапе основной задачей сформулирована следующая: необходимо выбрать для использования эффективный формат передаваемых в сети данных. Изучив наиболее популярные варианты, решено использовать формат JSON (JavaScript Object Notation [4]). Это легко расширяемый, популярный и сериализуемый формат, не требующий больших вычислительных затрат для работы с ним. Пример передаваемого объекта в формате JSON приведен во фрагменте ниже.

Фрагмент. Программный код передаваемого фрагмента данных в формате JSON

```
{  
"type": "get",  
"data": {  
    "temp": 20,
```

```
"voltage": 12.6,  
"speed": 20,  
"distance": {  
    "0": 15,  
    "1": 20,  
    "2": 20,  
    "3": 14  
},  
"model": {  
    "0": "null",  
    "1": "fast",  
    "2": "slow",  
    "3": "manual",  
}  
"variables": {  
    "message": "low battery",  
    "status": "ok"  
}  
}  
}
```

/*

temp - температура аккумулятора,
voltage - напряжение на аккумуляторе,
speed - реальная скорость платформы,
distance - показания датчиков (количество датчиков переменного),
model - режимы доступные для установки в настоящий момент,
variables - массив дополнительных данных, размер массива переменный,
ключи и значения в котором могут быть абсолютно любыми.

*/

```
{  
"type": "set",  
"data": {  
    "speed": 32,  
    "angle": -15,  
    "model": "manual",  
    "variables": {  
        "message": "normal state"  
        "status": "ok",  
    }  
}  
}
```

```
}  
/*  
speed - установить скорость платформы,  
angle - установить угол поворота колёс,  
model - установить режим движения платформы,  
variable - массив дополнительных данных, размер массива переменный,  
ключи и значения в котором могут быть абсолютно любыми.  
*/
```

Важно также отметить, что выбранный формат не обязывает иметь строгую структуру данных, что позволяет легко расширять и модифицировать процесс обмена данными.

Для формирования пакета данных на Arduino нами была разработана собственная библиотека, решающая также задачу сериализации данных [5]. Сериализация данных необходима прежде всего для сжатия данных, а следовательно, сокращения времени их отправки по протоколу UART.

В программном коде Raspberry реализована десериализация принятых данных. Далее полученные данные в одном потоке отправляются на запись в базу данных, а в другом – программе-клиенту для мониторинга. Также написанный код предусматривает и обратный поток данных. Такой подход обусловлен необходимостью иметь возможность управления автономной платформой. Под управлением подразумевается передача экстренных сигналов на остановку, а также установка режимов поведения платформы, включение или выключение датчиков и т.д.

Схема обмена данными будет представлена нами в следующих публикациях.

ЛИТЕРАТУРА

1. UART и с чем его едят. [Электронный ресурс] – Режим доступа: <https://habr.com/ru/post/109395/>. Дата доступа: 01.02.2019.
2. Универсальный асинхронный приёмопередатчик. [Электронный ресурс] – Режим доступа: https://ru.wikipedia.org/wiki/Универсальный_асинхронный_передатчик/. Дата доступа: 03.02.2019.
3. ARM Учебный курс. USART. [Электронный ресурс] – Режим доступа: <http://easyelectronics.ru/tag/>. Дата доступа: 04.12.2018.
4. Introducing JSON. [Электронный ресурс] – Режим доступа: <https://www.json.org/>. Дата доступа: 10.12.2018.

5. Что такое сериализация данных? [Электронный ресурс] – Режим доступа: <http://qaru.site/questions/259874/what-is-data-serialization/>. Дата доступа: 14.12.2018.