

Удалённое управление роботом посредством хтмрр-протокола

*А.Л. Печников, В.А. Жмудь, В.Г. Трубин
ФГБОУ ВПО НГТУ, Новосибирск, Россия*

Аннотация: На примере учебно-демонстрационной роботизированной платформы [1], рассматривается система удалённого управления движением посредством хтмрр-протокола. Вопросы удалённого визуального контроля движения робота не анализируются.

Ключевые слова: удалённое управление, робот, хтмрр-протокол.

Задача удалённого управления и получения информации с удалённых электронных устройств является достаточно распространённой. Например, в целях администрирования часто требуется осуществлять удалённый контроль и управление питанием и кондиционированием компьютерного и иного электронного оборудования [2]. Имеется множество аппаратно-программных средств, например [3-5], позволяющих реализовать дистанционный контроль состояния различных датчиков и управление разнообразными объектами. Общим для большинства имеющихся вариантов удалённого доступа к электронным устройствам является использование существующей инфраструктуры передачи информации, такой как Internet или GSM-сети, и стандартных протоколов, например, http, icmp, snmp в компьютерных сетях, и sms в GSM-сетях. Такое решение позволяет за сравнительно малое время построить систему удалённого управления и контроля не затрачивая больших средств на развёртывание каналов передачи данных. В силу широкой территориальной доступности сетей Internet и GSM, система может быть реализована с значительным расстоянием между оператором и объектом управления и контроля, и может иметь глобальный характер. Данный подход не применим только для очень ограниченного круга задач, когда имеются специфические требования, как-то: исключительно высокая защищённость и/или надёжность (военные системы); доступ к объектам находящимся вне пределов стандартных сетей связи.

Целью работы являлось расширение функциональных возможностей учебно-демонстративной роботизированной платформы (рис.1; [1]) в части удалённого управления движением.

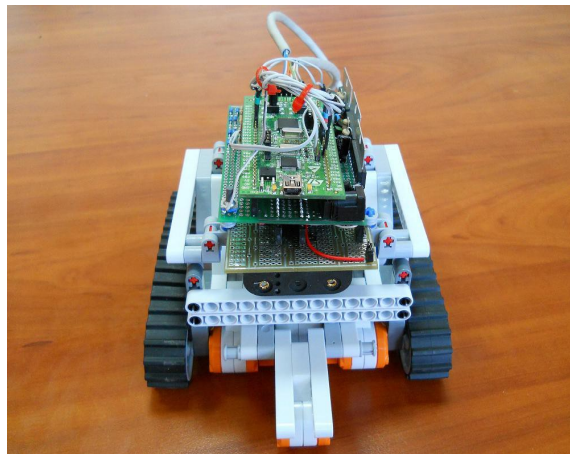


Рис.1. Учебно-демонстративная роботизированная платформа — объект удалённого управления

Исходно, на систему удалённого управления роботом были наложены следующие ограничения:

- 1) управление должно осуществляться с компьютера подключенного к сети Internet;
- 2) задача обеспечения визуального контроля движения робота с места оператора не ставилась.

Согласно [6], возможности беспроводного дистанционного управления роботизированной платформой ограничены Bluetooth каналом и простым протоколом управления движением, основанном на передачи текстовых команд. С учётом этого, на участке от удалённого, клиентского компьютера оператора до серверного компьютера, взаимодействующего с роботом по Bluetooth каналу, целесообразно использовать один из стандартных протоколов обмена мгновенными сообщениями: IRC, OSCAR (ICQ), MMP (Mail Agent), Skype, Google Talk, XMPP (Jabber). Архитектура и протокол IRC в основном ориентированы для многопользовательского тематического общения в конференциях (чат), что в нашем случае однопользовательского управления движением робота не требуется. OSCAR, используемый в сервисе ICQ, до недавнего времени был закрытым, а существующие библиотеки не были застрахованы от внесения в протокол изменений и как следствие потери работоспособности. Ситуация не изменилась и после опубликования спецификации на протокол. Для использования протокола MMP (Mail Agent) необходима предварительная регистрация аккаунта на русскоязычном сервисе mail.ru, что в основном ограничивает его применимость территорией

России и стран СНГ. Протокол Skype является закрытым, поэтому его использование проблематично. Сервис Google Talk, базирующийся на XMPP протоколе имеет тенденцию к закрытию [7] с заменой на сервис Hangouts с закрытым протоколом. В качестве преимуществ XMPP протокола, по сравнению с другими указанными, можно выделить его открытость, глобальное использование и большой выбор готовых библиотек под разные языки программирования и платформы. Для ускорения

реализации системы удалённого управления роботом, и упрощения переносимости кода на различные платформы, в качестве языка программирования был выбран скриптовый язык Python (питон).

Структурная схема системы удалённого управления движением робота, при использовании двух различных xmpp-серверов для клиентской и серверной частей, приведена на рис. 2.

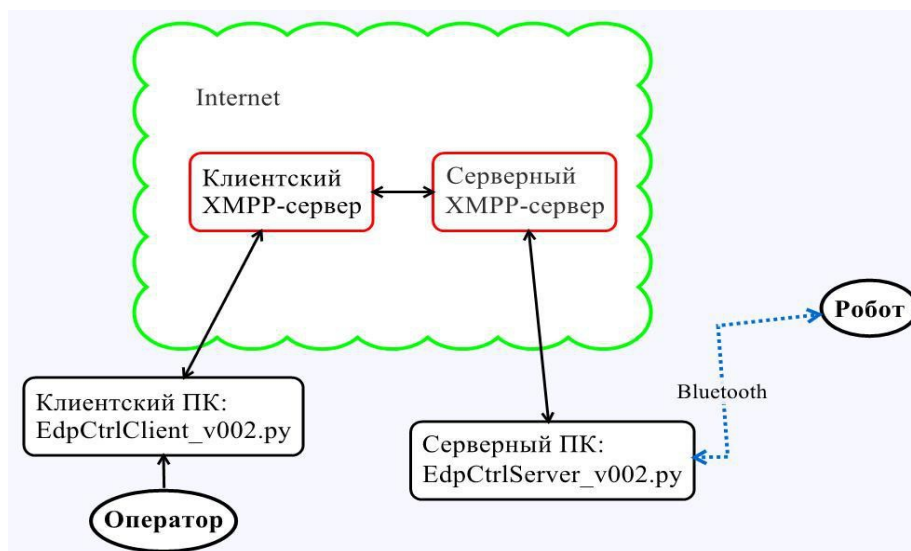


Рис.2. Структурная схема системы удалённого управления движением робота

В соответствии с структурой системы представленной на рис. 2 была выполнена разработка двух программных компонентов: клиентской «EdpCtrlClient_v002.py» и серверной «EdpCtrlServer_v002.py».

Клиентская часть обладает следующей функциональностью:

- 1) подключение к xmpp-серверу с помощью зарегистрированного аккаунта;
- 2) настройка имени jabber-аккаунта серверной части;
- 3) сканирование клавиатуры для распознавания команд управления роботом и трансляцию этих команд в xmpp-сообщения;
- 4) передачу командных сообщений управления роботом.

Серверная часть обладает следующей функциональностью:

- 1) подключение к xmpp-серверу с помощью зарегистрированного аккаунта;
- 2) приём управляющих сообщений;
- 3) передачу управляющих команд через Bluetooth канал роботу.

Наличие библиотеки xmppru [8] значительно упрощает реализацию подключения к xmpp-серверу, отправку и получение сообщений. Например, подключение к xmpp-серверу клиентской программы и отправка сообщения серверной части (реализация п.1 и п.4 функциональности) могут быть выполнены небольшим количеством кода:

```
import xmpp # подключение xmpp-библиотеки
#-----
# Set and create Jabber client
# JID
jid=xmpp.protocol.JID(jclient)
# где jclient — строка с JID,
# например, user_name@jabber.ru
# Create client instance
client=xmpp.Client(jid.getDomain(),debug=[None])
# Do client connection
client.connect()
client.auth(jid.getNode(), password)
# где password — строка-пароль
client.sendInitPresence()
# Make a status
presence=xmpp.Presence(status =
u'OOO "КБ Автоматика"', show = \
'chat', priority = '1')
client.send(presence)
# Register Message receive handler for jabber client
client.RegisterHandler('message',
MsgReceiveHandler)
#-----
# Make a message
msg=xmpp.protocol.Message(jserver, u'There is a
connection to the edp-robot server.', 'chat')
# Send the message form sender to recipient
client.send(msg)
msg.setBody('')
client.send(msg)
```

Xmpp-библиотека также позволяет с помощью функции `xmpp.Client.RegisterHandler()` назначить в качестве обработчика приёма сообщений некоторую функцию и реализовать в ней требуемую реакцию.

Для считывания клавиатурных команд управления роботом на компьютере оператора (клавиши-стрелки и др.) была использована библиотека `PyWin32`.

На стороне серверной части программы `EdpCtrlServer_v002.py` использовалась библиотека `PySerial` [9], позволяющая всего несколькими строками кода выполнить отправку командных сообщений через Bluetooth, например:

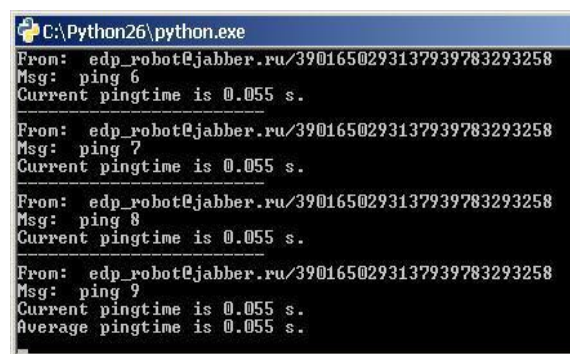
```
import serial
# Try to open COM-port for robot controlling.
try :
    Com=serial.Serial('COM3', 9600, dsrdtr = 1,
    timeout = 0)
except :
    Com='None'
if Com != 'None' : Com.write(robot_command)
# zde robot_command — строка-команда
```

Проверка работоспособности системы удалённого управления учебно-демонстрационной роботизированной платформой проводилась следующим образом. Предварительно, на корпоративном сайте был запущен xmpp-сервер и создан аккаунт через который работала серверная часть системы: программа `EdpCtrlServer_v002.py`. Также был создан другой аккаунт на публичном xmpp-сервере `jabber.ru`, через который работала клиентская часть системы: программа `EdpCtrlClient_v002.py`. Оператор и управляемый робот находились в разных городах. Для визуального контроля совершаемых оператором действий дополнительно использовалась программа `Skype`. Результатом проверки явилось подтверждение работоспособности системы удалённого управления роботом с хорошей реакцией робота на управляющие действия оператора.

Для количественной оценки временных задержек передачи команд управления роботом в систему была добавлена функция измерения этой величины. По активации процесса получения оценочного значения задержек кнопкой клавиатуры «F2», со стороны клиентской программы отправляется десять специализированных сообщений с периодичностью в секунду. Время отправки каждого сообщения фиксируется и сохраняется в массиве. Серверная часть, распознавая эти сообщения, просто возвращает их на сторону клиента. По приходу возвращённого пакета фиксируется время и также сохраняется в массиве. Разница между временем отправки и приёма возвращённого пакета составляет удвоенное время прохождения команд управления, иначе, является двусторонней задержкой. В качестве оценки задержки

используется усреднённая по десяти пакетам и делённая пополам величина. Временные задержки передачи команд управления роботом, полученные экспериментально, лежат в пределах диапазона от 90 до 140 мс. Следует учитывать, что такие задержки вносятся только системой удалённого управления и не включают задержек передачи команд от серверного компьютера к роботу по Bluetooth каналу. Отметим также, что величина задержек зависит от фактической длины трассы передачи сообщений и загруженности xmpp-серверов.

Для уменьшения временных задержек можно исключить участок канала передачи команд управления между двумя xmpp-серверами, если обе части, клиентская и серверная, будут работать через один и тот же xmpp-сервер. Практически установлено, что при такой организации системы, временные задержки передачи команд управления снижающуюся до уровня 50-80 мс (рис. 3).



```
C:\Python26\python.exe
From: edp_robot@jabber.ru/3901650293137939783293258
Msg: ping 6
Current pingtime is 0.055 s.

From: edp_robot@jabber.ru/3901650293137939783293258
Msg: ping 7
Current pingtime is 0.055 s.

From: edp_robot@jabber.ru/3901650293137939783293258
Msg: ping 8
Current pingtime is 0.055 s.

From: edp_robot@jabber.ru/3901650293137939783293258
Msg: ping 9
Current pingtime is 0.055 s.
Average pingtime is 0.055 s.
```

Рис.3. Результат определения оценки временной задержки передачи команд управления

ВЫВОД

Использование xmpp-протокола позволяет расширить функциональность электронных устройств, в нашем случае учебно-демонстрационной роботизированной платформы, в части реализации удалённого управления и контроля. Имеющиеся временные задержки передачи команд управления даже на уровне 115 мс не препятствуют хорошему качеству управления движением робота.

ЛИТЕРАТУРА

- [1] А.В. Ескин, В.А. Жмудь, В.Г. Трубин. Построение платформы моделирующей работу роботизированных средств на базе конструктора `Lego Mindstorms NXT 2.0` в части управления электродвигателями // Автоматика и программная инженерия. 2013. № 1 (3). С. 88-94. Электронная версия, URL: <http://nips.ru/images/stories/zhoumal-AIPI/3/Paper-2013-1-13.pdf> (дата обращения 05.09.2013).
- [2] В. Клименков. Управление и удаленный мониторинг систем бесперебойного питания // Технологии и средства связи. 2007. №2. С. 133. Электронная версия, URL: <http://www.tsonline.ru/articles2/in-ch-sec/upravl-udalen-monit-sist-bespereboyn-pitan> (дата обращения 05.09.2013).
- [3] Vassilis Serasidis. Удаленное управление устройствами с помощью SMS. URL: <http://www.rlocman.ru/shem/schematics.html?di=66227> (дата

обращения 05.09.2013).

[4] Denis Yagov. Удалённое управление питанием нагрузки через сети GSM. URL: <http://cxem.net/sotov/sotov116.php> (дата обращения 05.09.2013).

[5] Kibermaster. XControl. Управление через компьютер, телефон, планшет или веб-браузер. URL: <http://kibermaster.net/xcontrol/> (дата обращения 05.09.2013).

[6] А.В. Ескин, В.А. Жмудь, В.Г. Трубин. Реализация дистанционного управления по радиоканалу Bluetooth платформой, моделирующей работу роботизированных средств. 2013. № 1 (3). С. 82-87. Электронная версия, URL: <http://nips.ru/images/stories/zjournal-AIPI/3/Paper-2013-1-12.pdf> (дата обращения 05.09.2013).

[7] Santiago26. Без паники! Про то, что сделал Google с XMPP. URL: <http://habrahabr.ru/post/180159/> (дата обращения 12.09.2013).

[8] Сайт проекта библиотеки xmppru. URL: <http://xmppru.sourceforge.net/> (дата обращения 09.09.2013).

[9] Сайт документации библиотеки PySerial. URL: <http://pyserial.sourceforge.net/> (дата обращения 09.09.2013).



Андрей Львович Печников – ведущий инженер ООО «КБ Автоматика»,
e-mail: kba-emur@bk.ru



Вадим Аркадьевич Жмудь – заведующий кафедрой Автоматики НГТУ, профессор, доктор технических наук, автор более 200 научных статей, включая 10 патентов и 6 учебных пособий. Область научных интересов и компетенций – теория автоматического управления, электроника, лазерные системы, оптимизация, измерительная техника.

E-mail: oao_nips@bk.ru



Виталий Геннадьевич Трубин, зав. лаб. кафедры Автоматики НГТУ, директор ООО «КБ Автоматика». Автор 18 научных статей. Область интересов – разработка специализированной электроники.
E-mail: trubin@ngs.ru