BULLETIN OF NATIONAL ACADEMY OF SCIENCES
OF THE REPUBLIC OF KAZAKHSTAN
ISSN 1991-3494
Volume 1, Number 353 (2015), 26 – 34

DESIGNING AND ANALYSIS OF MOBILE ROBOTS FOR MULTI-AGENT SYSTEMS

A. A.Kuandykov[1], D. K. Kozhamzharova[2], N. Karimzhan[1], O. A.Baimuratov

[1]International Information Technologies university, Almaty, Kazakhstan,
[2]Kazakh National Technical university named after K.I. Satpayev, Almaty, Kazakhstan
[3]Suleyman Demirel University, Kazakhstan.
E-mail: abu.kuandykov@gmail.com; dinara887@gmail.com; nkarimzhan@gmail.com;
olimzhon.baimuratov@sdu.edu.kz

Key words: mobile robot, multi-agent systems, remote control, Arduino, Raspberry Pi.

Abstract. Theory and technology of multiagent systems (MAS) develops at the junction of a number of sciences, scientific and technical fields, the main among them are: artificial intelligence, object-oriented design and programming, distributed computing, systems theory and systems analysis, sociology, linguistics, cognitive science, analytical philosophy. At the heart of the MAS is the notion of an artificial agent - special software or hardware-software system - which implements a unique model of behavior in the real world or in the information environment.

The article describes the main stages of the construction and management of a mobile robot using a mobile device, the connection between the robot and the device that controls the robot, as well as the codes of the mobile robot control program.

The paper describes the control of the mobile robot using a mobile device, the connection between the robot and the device, as well as provides program codes for remote robot control.

УДК 004.896

КОНСТРУИРОВАНИЕ И АНАЛИЗ МОБИЛЬНОГО РОБОТА ДЛЯ МУЛЬТИАГЕНТНЫХ СИСТЕМ

А. А. Куандыков[1], Д. Х. Кожамжарова[2], Н. Каримжан[1], О. А. Баймуратов[3]

[1]Международный университет информационных технологий, Алматы, Казахстан,
[2]Казахский Национальный технический университет им. К. И. Сатпаева, Алматы, Казахстан,
[3]Университет им. Сулеймана Демиреля, Каскелен, Казахстан.

Ключевые слова: мобильный робот, мультиагентные системы, дистанционное управление, Arduino, Raspberry Pi.

Аннотация. Теория и технология мультиагентной системы (МАС) развивается на стыке ряда наук и научно-технических направлений, главные среди которых: искусственный интеллект, объектно-ориентированное проектирование и программирование, распределенные вычисления, теория систем и системный анализ, социология, лингвистика, когнитология, аналитическая философия. В основе МАС лежит понятие искусственного агента – специальной программной или аппаратно-программной системы, – которая реализует модель некоторого уникального поведения в реальном мире или в информационной среде.

В статье рассмотрены основные этапы конструирования и управлениея мобильным роботом с помощью мобильного устройства, связь между роботом и устройством, управление роботом, а также приводятся коды программы управления мобильным роботом.

Введение. Анализ современных мультиагентных систем [1-13] показывает, что с каждым годом МАС достигают нового совершенного уровня благодаря следующим аспектам:

- Простоте управления;
- Оптимизации алгоритмов управления;

- Усовершенствованию комплектующих (улучшение технических характеристик);
- Программам для различных типов роботов;
- Программам для управления роботами;
- Увеличению функциональных задач (мультифункциональности);
- Мобильности и т.д.

Мультиагентная система встречается во многих научно-технических направлениях, в различных отраслях промышленности, в компьютерных науках [1-5], в медицине [6], в робототехнике, в охране природы, в сельском хозяйстве и др. [7-15, 20-23]. МАС играет важную роль с точки зрения практического применения и создания программ для мобильных роботов.

Ряд преимуществ МАС достигается с помощью программного обеспечения разрабатываемых для самих агентов (роботов, мобильных роботов, информационных роботов и т.д.) и для управления ими.

В данной работе мы акцентировали внимание на простоте управления мобильным роботом через мобильное устройство, на связь между мобильным роботом и мобильным устройством и на разработку программ.

Разработка программ для мобильных роботов на данный момент представлено во многих трудах зарубежных ученных [2, 6-8, 10-12, 20-23].

В Казахстане разработка программ и конструирование мобильных роботов набирает обороты, благодаря таким мероприятиям, как международная выставка-конференция ASTEX (Astana Smart Technologies Exhibition) и многие другие международные конференции и семинары. На сегодняшний день представлены роботы, разработанные в лаборатории IT университета, такие как робот-гуманоид «Шолпан», робот-терминал «PSM» и т.д.

Рассматриваемый мобильный робот представляет собой робот, на котором размещена видеокамера, свободно перемещающаяся с помощью шасси. Получаемая информация через видеокамеру передается через Интернет на мобильное устройство или компьютер. Беспроводное управление, перемещение мобильного робота осуществляется с помощью программы, разработанной для мобильных устройств, также управление можно осуществить и с помощью персонального компьютера. Далее мобильный робот – мобильный видеорегистратор (МВР).

1. **Разработка и программная реализация.** При разработке системы мы должны учитывать такие параметры, как простота и легкость управления. То есть предполагается, чтобы любой пользователь мог управлять роботом: включать-выключать, подключать мобильное устройство к роботу и управлять роботом.

Управление следует распределить между человеком и машиной «роботом-МВР». Распределение может быть следующим:

Первый уровень: человек задает/указывает цель, робот выполняет все необходимые действия в автономном режиме.

Второй уровень: человек задает цель и пути достижения (план) поставленной цели, тогда робот выполняет прослеживание/мониторинг (непрерывный процесс наблюдения и регистрации параметров объекта, в сравнении с заданными критериями) и выполнение плана (осуществление движения по заданному пути (траектории).

Третий уровень: человек указывает действие, а робот исполняет заданное действие.

В данной работе решение проблемы взаимодействия рассмотрим снизу вверх, то есть самый нижний уровень взаимодействии – уровень указания действий – выполнения.

На рисунке 1 представлены действия, которые может совершать пользователь с помощью разработанного программного обеспечения для мобильных устройств.

В данной статье мы представляем два основных устройства, для которых необходимо разработать программу:

1) программное обеспечение для МВР;

2) программное обеспечение для управления МВР.

Как было указано выше, для разработки МВР необходимо выбрать архитектуру системы для управления роботом.

А также необходимо разработать:

- Программное обеспечение для мобильных устройств.

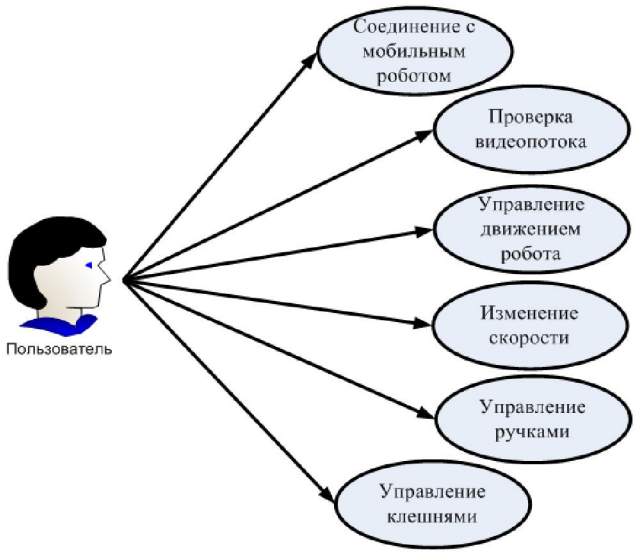


Рисунок 1 – Схема управления роботом

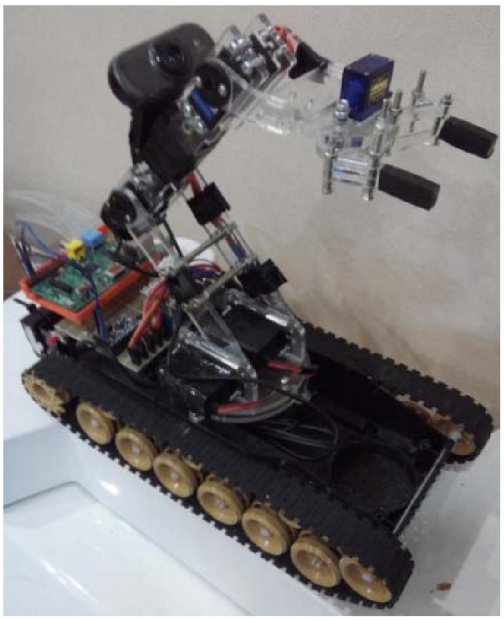


Рисунок 2 – Мобильный видеорегистратор (МВР)

- Панель управления роботом:
  а) джойстик управления двигателем для выбора направления робота;
  б) кнопка изменения скорости;
  в) кнопки управления веслом;
  г) кнопка для клешни;
  е) форма выборки для показа потокового видео.
- Панель настройки соединения.

2. Структура робота и основные коды программ. Рассмотрим подробнее структуру нашего объекта – МВР, который представлен на рисунке 2.

В таблице 1 приведены основные характеристики данного объекта.

Таблица 1 – Основные характеристики МВР

	Название	Модель/Версия
	Видеокамера	Logitech C210
	Микрокомпьютер	Raspberry Pi B 2011.12 ver
	Сервоприводы	Tower pro sg90*1 шт. [14] XQ-S3006S *4 шт. [15]
	Регуляторы напряжения	7806 *5 шт. 7805 *1 шт.
	Электропривод	L298 [16]
	Аппаратно-вычислительная платформа	Arduino Board Mini Pro [17]
	Аккумулятор	LiPo 11.1V
	WiFi:	TP-Link wn725n

На основе математических моделей и принципов, описанных в работах [1, 5, 10, 11, 18] была разработана система управления мобильным роботом. По описанным выше деталям был сконструирован «Мобильный видео- регистратор» и программное обеспечение (мобильное приложение-ПО) для управления МВР. Управление МВР осуществляется с помощью мобильных устройств различных типов. Программное обеспечение обеспечивает связь между мобильным устройством и МВР.

Одним из основных блоков в работе МВР является микроконтроллер Arduino. На рисунке 3 показана блок-схема микроконтроллера Arduino, подключенного к L298.

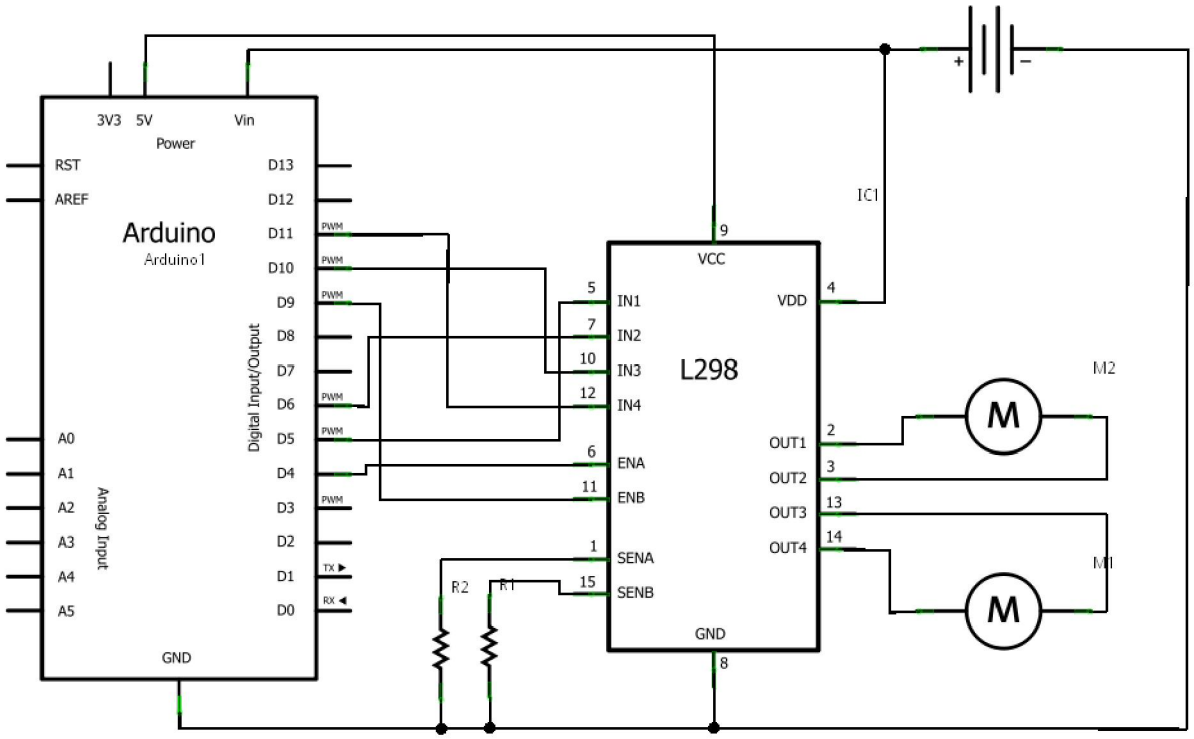


Рисунок 3 – Микроконтроллер Arduino, подключенный к L298

Разработка ПО осуществляется через программную оболочку (IDE). Язык программирования Arduino - C++. При разработке ПО можно использовать библиотеки C++. Arduino IDE не нуждается в дополнительных настройках компилятора, что уменьшает риск появления ошибок.

Для управления мобильным роботом был создан класс - `class Driver`. Данный класс описывает движение робота. В листинге 1 представлен код, определяющий направления движения каждого двигателя и функций созданные для определенного движения:

```
     class Driver{
     public:
     // motors forward
       int firstForDir, secondForDir;
     // motors backward
       int firstBackDir, secondBackDir;
     //determination PWM for motors
       int firstPwm, secondPwm;
     //class constructor
           Driver(int ffDir, int fsDir, int
fPwm,
     int sfDir, int ssDir, int sPwm){…}
      //function for the movement robot forward
       void forward(int newPwm){
      stop();
         _HIGH(firstForDir); _LOW(firstBackDir);
         _HIGH(secondForDir);_LOW(secondBackDir);
         analogWrite(firstPwm, newPwm);
         analogWrite(secondPwm, newPwm);
       }
```

```
    void backward(int newPwm){…}
    void toLeft(int newPwm){…}
    void toRight(int newPwm){…}
    void stop(){…}
  };
```

Листинг 1 – Описание направления движения каждого двигателя

Программа имеет две функции с именами `void setup()` и пустот `loop()`. Их присутствие является обязательным в любой программе в С + + для Arduino. Эти функции ни делают ничего, их в обязательном порядке нужно прописывать в коде программы, в противном случае при компиляции появиться ошибка. В данной программе, функции установлены в контакты, которые будут посылать сигналы двигателями и приводам. В листинге 2 приведен код, который открывает флажки обозначены буквами (A, C, D, E):

```
void setup() {
  Serial.begin(9600);
  A.attach(6);
  C.attach(5);
  D.attach(7);
  E.attach(8);
  F.attach(9);
  defaultPos();
  _A.is = true;
  delay(1000);
}
```

Листинг 2 – Функции для определения флажков

Листинг 3 содержит основную функцию `void loop()`, которая запускается на неопределенный промежуток времени и вызывает другие функции:

```
void loop() {
if(Serial.available() > 0){
    ch = Serial.read();
  …
    else if(ch == 'b') driver.backward(wheelPwm);//go backward
    else if(ch == 'f') driver.forward(wheelPwm); //go forward
    else if(ch == 'l') driver.toLeft(rotatePwm);//turn to the
left
    else if(ch == 'r') driver.toRight(rotatePwm);//go to the
right
    else if(ch == 's') driver.stop(); //stop wheels
    else if(ch == 'w') stopAllServos(); //stop all servos
    else if(ch == 'x') wheelPwm = 80, rotatePwm = 75;
    else if(ch == 't') wheelPwm = 180, rotatePwm = 100;
    else if(ch == 'y') wheelPwm = 255, rotatePwm = 120;
    ch = '0';
  }
…
}
```

Листинг 3 – Функция `void loop()`

Arduino удобен для управления роботизированными частями и для мониторинга датчиков. Тем не менее, имеются некоторые проблемы с потоками передачи и обработки видео. По этой причине мы должны были использовать аппаратные средства, которые могут обрабатывать большое количество данных и работать с ними. Для решения этого вопроса мы использовали Raspberry Pi.

Для потокового видео мы применили программу захвата видео и трансляции в сеть – MJPG-streamer. С помощью этого приложения на платформе Raspberry Pi может передавать изображения на сеть через USB-камеру. Для запуска программы необходимо запустить bash-скрипт *mjpg-streamer.sh* и добавить команду для старта. В листинге 4 представлены основные настройки для потокового видео:

```
   VIDEO_DEV="/dev/video0" - device
identifier.
   FRAME_RATE="30" - frame rate (FPS).
   RESOLUTION="640x480" - resolution
   PORT="8080" - http port for use
```

Листинг 4 – Основные настройки bash-скрипта mjpg-streamer.sh

Для подключения и передачи команд от микроконтроллера Raspberry Pi через UART использовалась библиотека языка Python – *RPi.GPIO*. Программа должна ожидать подключения клиента (мобильного устройства), затем перенести полученные данные в микроконтроллер Arduino.

В листинге 5 описывается метод создания серверного сокета для подключения клиентского сокета с мобильным устройством:

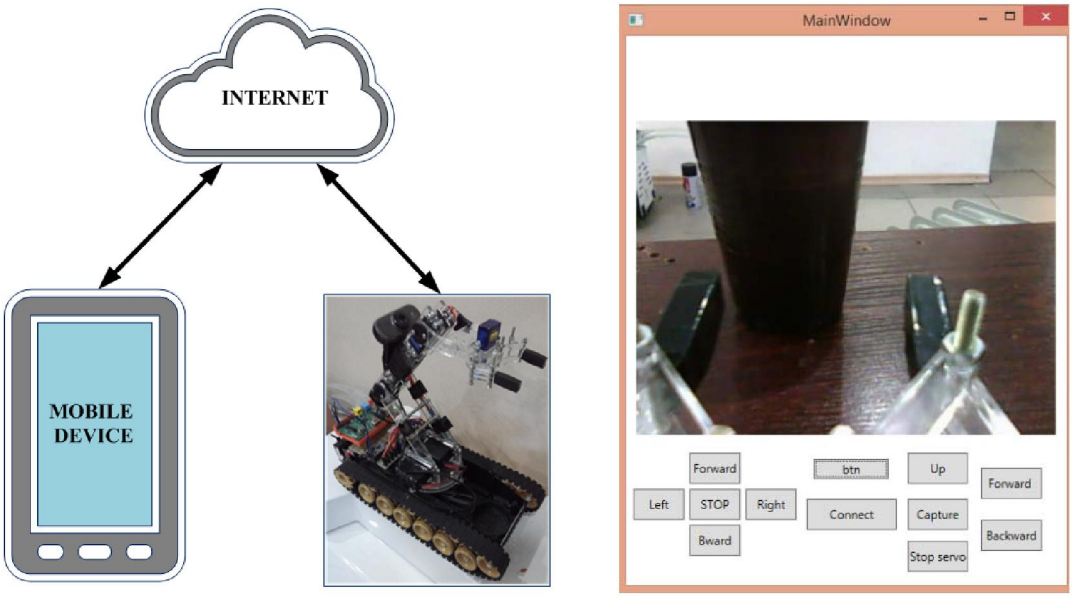
```
   # module for connection to the socket
   import socket
   # port for connecting
   …
   PORT = 9090
   # create a new socket
   sock = socket.socket()
   sock.bind(('', PORT))
   # listen to only one client
   sock.listen(1)
   conn, addr = sock.accept()
   print 'connected:', addr
   …
   conn.close()
```

Листинг 5 – Фрагмент кода для подключения к сокетам

3. Структура управления МВР. На рисунке 4 представлена структура взаимодействие мобильного устройства с МВР.

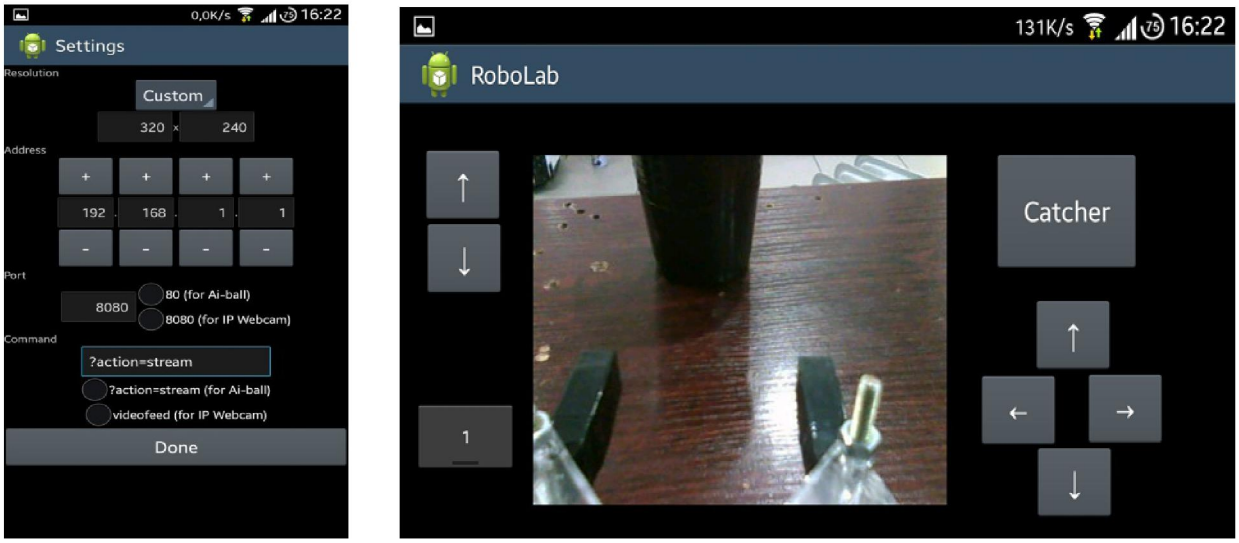
Принцип построения приложения для управления роботом на базе Android такой же, как и для ОС Windows. Приложения созданы для основных четырех классов: *MjpegActivity, SettingsActivity, MjpegInputStream и MjpegView*. Классы *MjpegInputStream* и *MjpegView* предназначены для обработки потокового видео. Класс *SettingsActivity* используется для настройки подключения к серверу (к мобильному роботу). Этот класс определяет IP-адрес для подключения, разрешение видео, потоковое видео порт и ссылку на него. На рисунке 5а представлен интерфейс настройки программы.

Последний класс *MjpegActivity* сочетает в себе все классы. В этом классе мобильное устройство, подключенное к роботу, обратывает и передает данные с помощью кнопок и джойстика, обрабатывает и отображает видеопотоки.



а) б)

Рисунок 4 – а) общая структура взаимодействия мобильного устройства с МРВ;
б) окно приложение для управления мобильным роботом на базе ОС Windows



а б

Рисунок 5 – а) интерфейс настройки программы на базе Android;
б) мобильное приложение для управления роботом

Для подключения к мобильному роботу через сокет был создан класс `ClientThread ()`, который реализуется в интерфейсе Runnable. Runnable абстрагирует блок исполняемого кода. Может разработать поток любого объекта, реализующегося через интерфейс Runnable. В листинге 6 показан класс `ClientThread ()`, где IP-адрес *server_ip Raspberry Pi* (тест IP адрес 192.168.1.1) и ServerPort – 9090:

```
…
public class ClientThread implements Runnable {
          public void run() {
          try {
             InetAddress serverAddr =
InetAddress.getByName(SERVER_IP);
       socket = new Socket(serverAddr, SERVERPORT);

             out = new PrintWriter(new
BufferedWriter(new
OutputStreamWriter(socket.getOutputStream())),true);
              } catch (UnknownHostException e1) {
                e1.printStackTrace();
             } catch (IOException e1) {
                e1.printStackTrace();
             }}}…
```

Листинг 6 – Класс подключения к серверу

Интерфейс мобильных приложений управления роботом, как в окне приложения на базе ОС Windows, имеет 4 кнопки управления движением робота, 2 кнопки для управления манипулятором, одну кнопку для клешней и 1 кнопку для контроля скорости робота. Интерфейс для мобильных устройств в действии показано на рисунке 5б.

Заключение. Из анализа современных мультиагентных систем мы определили современные направления исследования в области МАС и проблемы, которые возникают во время проектирования, создания и тестирования МАС, а так же определили некоторые затруднения, с которыми часто сталкиваются при программировании.

Простота управления роботом через мобильное устройство, связь между роботом и мобильным устройством и создание для них программ также является важной задачей, математические основы которых были рассмотрены в [5, 18]. В ранее изданных работах [1, 3, 18] был проведен сравнительный анализ по производительности каждой из моделей. Полученные результаты подтвердили эффективность применения агентного подхода.

В данной работе приведены разработанные коды программ, структура действия и основные характеристики.

Представленная работа актуальна в области создания программ для мобильных устройств, управляющих мобильными роботами; в области создания мультиагентных систем; в области организации управления группой агентов; разработчикам, инженерам и всем пользователям, которые интересуются данным техническим направлением.

Результаты, представленные в данной работе, упрощают процесс проектирования/сборки и разработки ПО для МВР, так как коды основных узлов мобильных роботов во многих случаях являются идентичными.

REFERENCES

[1] Kuandykov A.A., Kassenkhan A.M, Mukazhanov N.K., Kozhamzharova D.K and others. The Formalization of the Problem area, Implementation and Maintenance of Business process by a group of unmanned vehicles. International Journal of Computer Technology and Applications (IJCTA 2013). India. Vol. 4(5). P. 79-82.

[2] Drogoul A., Vanbergue D., Meurisse T. Multi-Agent Based Simulation: Where are the Agents? LIP6 – Universit? Paris 6, 2002. P.1-16.

[3] Uskenbayeva, R.K., Kuandykov, A.A., Cho Young Im, Kalpeyeva, Zh.B. Kozhamzharova, D.K. Organization of computing processes in the large heterogeneous distributed systems. 44th International Symposium on Robotics. 2013. P. 1-4.

[4] Jennings N.R. and Bussmann S. Agent-Based Control Systems. IEEE Control Systems Magazine. 2003. Vol. 23(3). P. 61-74.

[5] Uskenbayeva R.K., Kuandykov A.A , Cho Young, Kozhamzharova D.K., Kalpeyeva Zh.B. Models and Methods of Joint Work Management of Group of Unmanned Vehicles. 2013. 13th International Conference on Control, Automation and Systems (ICCAS 2013), Kimdaejung Convention Center, Gwangju, Korea, Oct. 20-23, 2013. P. 552-555.

[6] Zieller A., Braubach L., Pokahr A., Rothlauf F., Paulussen T.O., Lamersdorf W., Heinzl A. Evaluation of a Multi-Agent System for Hospital Patient Scheduling. International Transactions on Systems Science and Applications. 2006. Vol. 1(4). P. 375-380

[7] Hadzic M., Dillon D.S., Dillon T.S. Use and Modeling of Multi-agent Systems in Medicine. 2009. P. 303-307.

[8] Moreno A. Medical applications of Multi-Agent Systems //Computer Science & Mathematics Department, Universitat Rovira i Virgili, ETSE. Campus Sescelades. Av. dels Pa?sos Catalans. Spain, 2003. P. 1-15.

[9] Gabel T., Riedmiller M. Scaling Adaptive Agent-Based Reactive Job-Shop Scheduling to Large-Scale Problems. Proceedings of the 2007 IEEE Symposium on Computational Intelligence in Scheduling (CI-Sched 2007), 2007. P. 259-266.

[10] Gabel T., Riedmiller M. Joint Equilibrium Policy Search for Multi-Agent Scheduling Problems. 2007. P. 61-72.

[11] Agnetis A. Multi-agent scheduling problems. 2011. 100 p.

[12] Vidal J.M. Fundamentals of Multiagent Systems. 2010. March. 155 p.

[13] Kuandykov A.A. Fundamentals of the axiomatic theory of situational management of complex objects. Astana: Vestnik ENU. 2009. 2. P. 57-63.

[14] Kuandykov A.A. Axiomatic foundations of a formal system of managing complex objects// Proceedings National Academy of Sciences of RK series of Physics and Mathematics. 2009. № 5. P.12-15.

[15] Liu J.J., Chen Q.X., Mao N. and Lin Z.A. A multi-agent-based mould due date setting approach in stochastic production. International Journal of Production Research. Vol. 49(5). P. 1353-1371.

[16] "Electronicoscaldas" electronic devices store for constructing the mobile robots, http://www.electronicoscaldas.com/datasheet/SG90_Tower-Pro.pdf

[17] Shenzhen XQ-Power Model Electronics Co., Ltd, http://www.xq-power.com/EnProduct_Show.asp?ID=42

[18] Online retail store – SparkFun, https://www.sparkfun.com/datasheets/Robotics/L298_H_Bridge.pdf

[19] Official Arduino web-site, http://arduino.cc/en/Main/ArduinoBoardProMini

[20] Cho Y.I., Uskenbayeva R.K., Kuandykov A.A., Kozhamzharova D.K., Baimuratov O.A. The intelligent modeling of unmanned vehicles for solving the tasks of risk management in the business processes. International Symposium on Artificial Life and Robotics, AROB 19th, B-Con PLAZA, Beppu. JAPAN, January 22-24, 2014. P. 553-559.

[21] Weiss G. Multiagent Systems. The MIT Press Cambridge, Massachusetts. London, England, 1999. 609 p.

[22] DeLoach S.A. and Wood M. Developing Multiagent Systems with agent Tool. Intelligent Agents VII. Agent Theories, Architectures, and Languages. 7th. International Workshop, ATAL-2000. Boston, MA, USA, July 7-9, 2000. Proceedings, Lecture Notes in Artificial Intelligence. Springer-Verlag, Berlin, 2001. P. 1-15.

[23] Adhau S., Mittal M. L. A Multiagent Based System for Resource Allocation and Scheduling of Distributed Projects. International Journal of Modeling and Optimization. 2012. August. Vol. 2(4). P. 524-528.

МУЛЬТИАГЕНТТІК ЖҮЙЕЛЕР ҮШІН МОБИЛЬДІ РОБОТТЫ ҚҰРЫЛЫМДАУ ЖӘНЕ ТАЛДАУ

А. А. Қуандықов[1], Д. Х. Қожамжарова[2], Н. Каримжан[1], О. А. Баймұратов[3]

[1]Халықаралық ақпараттық технологиялар университеті, Алматы, Қазақстан,
[2]Қ. И. Сәтбаев атындағы Қазақ ұлттық техникалық университеті Алматы, Қазақстан,
[3]Сүлеймен Демирел атындағы университеті, Қаскелен, Қазақстан

Тірек сөздер: мобильді робот, мультиагенттік жүйелер, қашықтан басқару, Arduino, Raspberry Pi.

Аннотация. Мультиагенттік жүйелердің (МАЖ) теориясы мен технологиясы бірнеше ғылымдар мен ғылыми-техникалық бағыттардың қиылысында дамып келе жатыр, солардың арасындағы ең маңыздысы: жасанды зерде, объектілі-бағытталған жобалау және бағдарламалау, таратылған есептеулер, жүйелердің теориясы мен жүйелі анализ, әлеуметтану, лингвистика, когинитология, аналитикалық философия. МАЖ негізінде жасанды агент ұғымы жатыр – ол ақпараттық ортада немесе шынайы әлемде бірегей әрекеттің моделін жүзеге асыратын бағдарламалық немесе аппараттық-бағдарламалық жүйе болып табылады.

Мақалада мобильді роботты құраудың негізгі қадамдары және оны мобильді құрылғылар көмегімен басқару, робот пен құрылғы арасындағы байланыс, сонымен қатар роботты басқару кодттары қарастырылған.

Поступила 15.01.2015 г.