Open Source Mobile Robot with Raspberry Pi

E. Küçükkülahlı, and R.Güler

***Abstract*—The term of single board computer is used for computers which has built on a single circuit board with microprocessor, memory, input/output and other required units. In this study, control of a mobile robot in real time using Raspberry Pi which is a type of single board computer is performed via internet. In the developed system, all of the units in the mobile robot are controlled directly by Raspberry Pi, an additional microcontroller system is not used. Thus, without additional control equipment, a low cost and open source mobile robot design has been carried out.**

***Index Terms*—raspberry pi, mobile robot, remote control, websocket**

INTRODUCTION

Robots are considered as an essential part of modern life. Besides having been used in industry for a long time, they have become a part of end users' lives as a consequence of technological developments and cost reductions. In conjunction with increased use of mobile devices, remotely controlled implications have found a broader scope for themselves, which have enabled easy and fast control of such electronic devices remotely through mobile phone, tablet, or computer over internet.

There is a broad literature in remote device control, which might be distinguished according to control board and communication protocol used. Şahin and Yalvaç developed a wireless mobile robot prototype in their research [1]. Control board of the robot was designed using PIC16F877A, for the wireless control ARX-34S receiver and ATX-34S transmitter were used. Moreover, movements of the mobile robot was recorded by a wireless camera and transferred to a computer. Çayıroğlu and Şimşir controlled a camera installed-mobile robot car remotely using radio frequency signals [2]. Fong and Yusoff conducted research on a monitoring robot using Wi-Fi to control it real-time [3]. Robot's direction control was performed through internet connection with a wireless router on the mobile robot, taking the commands through Arduino and ethernet shield linked to it. Din and Lim suggested home automation system with a computer-controlled mobile robot [4]. Mobile robot is controlled through computer with XBee modules, and image transfer is done with a wireless camera. There is also wireless keys by 433MHz radio frequency modules, controlled by mobile robot. Kuo et al gave a cleaning robot wireless features with Arduino [5]. Wang et al introduced a feasible and affordable mobile vehicle [6]. Abid Al-Sahib and Azeez in their research created a mobile robot with a robotic arm and camera on it [7]. The robot can be controlled through internet. Raspberry Pi was used to get internet connection, while Arduino Mega board was used for direction control or the robot, and the movements of robotic arm. Espeset al. developed an affordable, web-controlled mobile robot for home monitoring [8]. Internet connection was attained through an Android smart phone on the mobile robot, while the information gathered was transferred to Arduino board over bluetooth, which control the robot. Prabha et al. recorded in a cloud system environmental factors such as heat and moisture with the Raspberry Pi-based mobile robot they developed [9].

In this research, an internet controlled-open source mobile robot design and application has been created. Mobile robot can be controlled through a web page, and immediate image transfer can be done by a camera on the robot. All these operations are controlled by Raspberry Pi, which is installed on the mobile robot. Second chapter will focus on hardware and software features of the mobile robot, while third chapter is a general review and suggestions about the research.

METHOD

A. Raspberry Pi

Various affordable computers have been developed in recent years, to expand the market. These computers mainly contain a processor, RAM, and input-output units, which are generally called single board computers. Received a great attention with their cheap prices from the users, these computers are mostly used at schools for computer learning.

The first of this trend, which became very popular was Mini2440'dır. Produced by Friendly ARM Company, this computer contains ARM9-based 400MHz Samsung processor, 64MB RAM, 1GB FLASH memory, and 34-pin GPIO [10]. Another board in this spectrum is BeagleBone, produced BeagleBoard.org establishment. BeagleBone computer contains 720MHz ARM Cortex-A8 processor, 256MB RAM, 2x46-pin GPIO [11].

E. Küçükkülahlı, Duzce Vocational School, Duzce University, 81010, Duzce, Turkey, (e-mail: enverkucukkulahli@duzce.edu.tr).

R. Güler, Duzce Vocational School, Duzce University, 81010, Duzce, Turkey, (e-mail: recepguler@duzce.edu.tr).

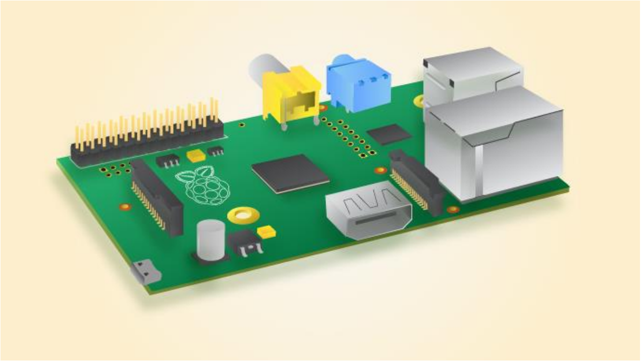

Fig. 1. Raspberry Pi Model B

Raspberry Pi board was introduced by Raspberry Pi establishment in 2011 [12]. Came out as the first model, Raspberry Pi Model B landed a sale over 2 million. Raspberry Pi Model B computer has 700MHz ARM-based Broadcom processor, 512MB RAM, 2 USB2.0 ports, 26-pin GPIO. 35$ price is the greatest factor in this computer's sale chart.

Many researchers conducted their research with the tendency to make more use of the advantages of Raspberry Pi computer. Dave Akerman and his friends connected Raspberry Pi to a meteorology balloon, and took Earth pictures up from 40 km. Prof. Simon Cox and his team composed 64 Raspberry Pi boards in order to create an experimental super computer [13].

Despite being a strong computer considering its price, Raspberry Pi also has limits. It can be used as a desktop, whereas its performance is almost the same as a mobile device (such as a tablet). Raspberry Pi establishment stated that the general performance of Model B is as much as the performance of a 300MHz Pentium 2 processor PC. Although its graphical performance is better, it is equal to the first version of Xbox game console [13].

Raspberry Pi's operating system is Linux. There is various Linux distribution is available for Raspberry Pi, which might be downloaded on www.raspberrypi.org webpage. The official and suggested distribution is Raspbian, which was developed by Mike Thompson and Peter Green. Operating system should be copied on an SD card after downloading. Raspberry Pi does not have an embedded memory, which only works through an SD card [13].

It is possible to use Raspberry Pi as a desktop when it is connected with peripherals such as screen, keyboard, and mouse, however, what distinguishes Raspberry Pi from other desktops is its GPIO, namely its general-purpose input/output pins which can be controlled by software (Figure 2). Therefore, units such as buttons, LED, engine might be controlled using Raspberry Pi board as in a micro controller, nevertheless it should be considered that Raspberry Pi's operating voltage is 3.3V [13].

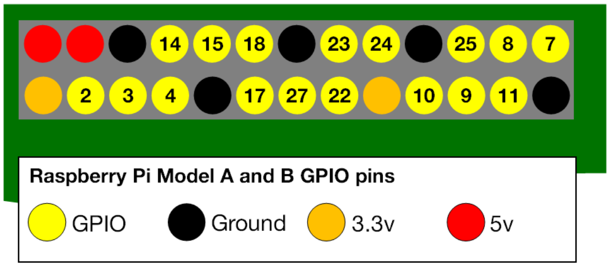

Fig. 2. Raspberry Pi GPIO pin outs

There are different methods to control Raspberry Pi's GPIO pins. In this research, input/output pins have been controlled using *node.jsrpi-gpio* library. Primarily, the library for software updating mode was downloaded. *setup()* method is used to read or write a pin [14].

```
var gpio = require('rpi-gpio');
gpio.setup(7, gpio.DIR_IN, readInput);
function readInput() {
    gpio.read(7, function(err, value) {
        console.log('The value is ' + value);
    });
}
```

A pin was set as input in the sample codes given above, and the output value was recorded [14]. "*gpio.setup(7, gpio.DIR_IN, readInput);*" code line with GPIO pin number 7 was set as input. Afterwards a function was created with the name "*readInput()*" and the pin was recorded by using "*gpio.read()*" command.

B. Motor Controller

Gear set-assembled DC engine kit was used to mobilize the designed mobile robot. L293D H-bridge drive was preferred in order to control DC engines' back-and-forth direction (Figure 3).

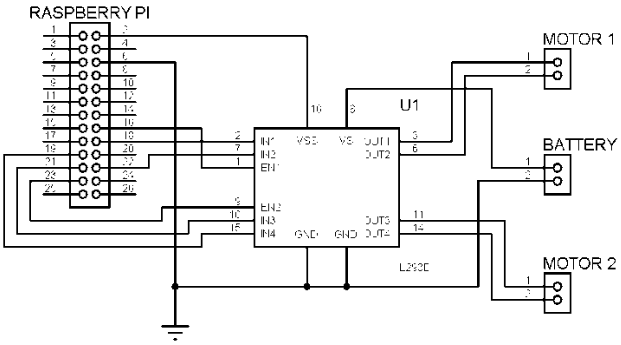

Figure 3. L293D motor driver circuit

Integrated L293D has a wide feed pitch (4.5V – 36V). Inputs of the integrated L293D are TTL-structured, and maximum output current is 600mA. Input of the circuit is controlled with GPIO pins number 19, 21, 23 and 16, 18, 22.

C. Camera

There are special camera modules that use GPIO pins for Raspberry Pi board, however, an oem webcam to work over USB port was preferred in order to reduce the cost (Figure 4). Camera uses 1.3MP CMOS sensor. A PanTilt module including two servo engines was added to make the camera move right-left and up-down. Direction control is possible directly through Raspberry Pi on the webpage.

Fig. 4. USB camera and Wi-Fi adapter

D. Wireless Adapter

Raspberry Pi might also be connected to wireless networks through its Ethernet input. The simplest and cheapest way to connect to wireless networks is to use a proper USB Wi-Fi adaptor (Figure 4). After the wireless adaptor is connected to Raspberry Pi, the system needs to be restarted in order for the system to recognize the new hardware, followed by network adjustment over operating system.

E. WebSocket and HTTP

Using an interface that works over web browser to control the mobile robot over cloud, without any other configuration enables the system to be available from everywhere. Connection between the server on the robot and a remote client over HTTP protocol might be a solution, however, some problems need to be solved in order to use HTTP protocol. HTTP was developed as a protocol to display constant HTML pages, but with technological developments it gained an expandable structure along with sessions and cookie software on the server side. Moreover, the connection was secured by the security features of HTTPS protocol [8].

HTTP protocol is based on a client starting a session on server, making a request, server replying, and ending the session. In this case, if HTTP protocol is used with a server working on robot, a secure connection is not possible, and the outcome will be a system which works independently from client. This is the reason why WebSocket technology is needed [8].

WebSocket protocol became standard in 2011 [15]. Connection in WebSocket is possible using servers and port 80. This sort of client is responsible for providing connection through URL.

The reason for using WebSocket in this research is a more secure and fast connection than HTTP protocol only by a single connection. At the same time a standard web browser is considered to be enough to monitor status of the robot, and to have an intercommunication.

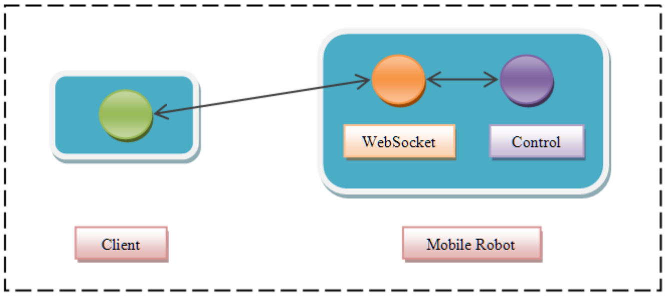

Fig. 5. Client – Mobil Robot connection

After having the connection as in the structure in Figure 5, client sends a request to WebSocket which works on the robot, and WebSocket transfers the request to robot controllers without waiting. In the meanwhile it returns the feedback from robot to client. WebSocket transfers cookies periodically from server to client and vice versa which allows the connection open during all the operations performed by robot, which prevents unintentional verbs in case of a disconnection between server and client by stopping all the operations, or by waiting for reconnection after putting all the operations on hold [8].

F. Mobile Robot Implementation

There are DC engines, a USB camera to extract footage, a wireless adaptor enabling connection, and two servo engine PanTilt unit controlling the direction of camera on the designed mobile robot (Figure 6). DC engines used on the robot are standard engines that work with 3-4.5V voltage, and they are fed by a separate resource not to influence the working of Raspberry Pi. All the other units are fed by Raspberry Pi.

In this research, NodeJs structure was used in order to manage both WebSocket protocol and engines through Raspberry Pi [16]. On the server side web service connection, DC and servo engines control, and webcam adjustments were made. To arrange these adjustments, necessary libraries were downloaded on server side, pre-adjustments were made and also GPIO pins were adjusted. Afterwards server application started recording pre-specified port of WebSocket, and waited for a client connection.

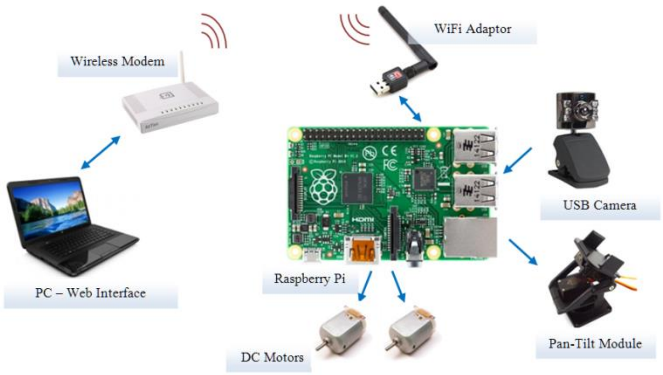

Fig. 6. Block diagram of the Mobil Robot

In the sample codes given below, the information is given that WebSocket to be working through port 8085 and the port number is specified, needed libraries are downloaded, and GPIO pins are adjusted as output port to be used by Raspberry Pi over initpins command. In conclusion, a function for run was identified, and preparations were concluded to make all the commands work on Raspberry Pi's console side.

```
vario = equire('socket.io').listen(8085)
var sys = require('sys'),
exec = require('child_process').exec
var path = '/sys/class/gpio/',
pins = [23, 24, 25, 10, 9, 11];
initPins()
functioncalistir(command) {
   exec(command, function (error, stdout, stderr)
   {
      if (error !== null)
         console.log('exec error: ' + error)
    })
}
```

After getting pre-adjustments done, a client was expected to connect to server. When a client sends a socket with the name of 'connection' to port 8085 of server, the function below runs and connection occurs.

```
io.sockets.on('connection', function (socket) {
   console.log('connection established');
})
```

After this step, connection starts working, and getting commands that would be sent from server client. In case of any data response, the command would be turned into an operation. Webpage designed to remotely control the mobile robot is seen in Figure 7. Control bars on right and left sides of the page enable controlling the right and left engines of the mobile robot separately, so robot's back and forth movements as well as its turning to right and left can be controlled. In the middle section of the page, snapshot from the camera can be displayed. Right below camera footage, there is a control bar enabling to change the direction of camera. Webpage also enables touch control on mobile devices.

Below there are some of the codes that are received when user moves the left bar.

```
socket.on('leftThrottle', function (data) {
   if (data < 0.5) {
      command = 'echo 1 > ' + path + 'gpio23/value';
      calistir(command);
      command = 'echo 1 > ' + path + 'gpio24/value';
      calistir(command);
   }
   else if (data > 0.5)
   {
      command = 'echo 1 > ' + path + 'gpio23/value';
      calistir(command);
      command = 'echo 1 > ' + path + 'gpio25/value';
      calistir(command);
   }
   else
   {
      command = 'echo 0 > ' + path + 'gpio23/value';
      calistir(command);
      command = 'echo 0 > ' + path + 'gpio24/value';
      calistir(command);
      command = 'echo 0 > ' + path + 'gpio25/value';
      calistir(command);
   }
});
```

leftThrottle information transfers information between [0-1] range. If the incoming information is less than 0.5 value, left engine will work backwards, if it is more than 0.5 value it will work forward, and if it is equal to 0.5 left engine will stop sensing that control bar was released. Same operation was written as *rightThrottle* for the right engine.

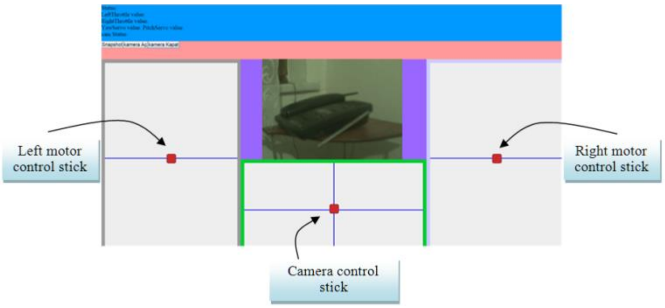

Fig. 7. Snapshot of the web site

There are two servo engines to move camera to right-left and up-down. Horizontal servo (yawServo) moves through information entered by '*servoblaster*' application [17]. The information for vertical servo is written as "*pitchServo*". "*mjpg_streamer*" application was used to get footage from camera. [18].

```
// yaw servo movement
socket.on('yawServo', function (data) {
   // data is normalized in 60-240
   var sonuc = (servoBuyuk - servoKucuk) * data +
   servoKucuk;
   command= 'echo 0=' + sonuc + ' >
   /dev/servoblaster';
   calistir(command);
});
```

On client side, a JavaScript supported HTML page was created to connect to server and get the required data from user, and at the same time transfer camera footage coming from server to user. To execute these operations, *jquery* library and open source joystick library for the user to control robot easily [19,20]. Below there are sample codes used to connect to server.

```
try {
   var socket =
   io.connect('http://192.168.2.6:8085')
}
catch (error) { }
$(document).ready(function () {
   try {
      socket.on('connect', function ()
      {
      $("#statusValue").html(connection
      established);
      })
   }
   catch (error) {
      $("#statusValue").html('can't connect to
      robot!');
   }
});
```

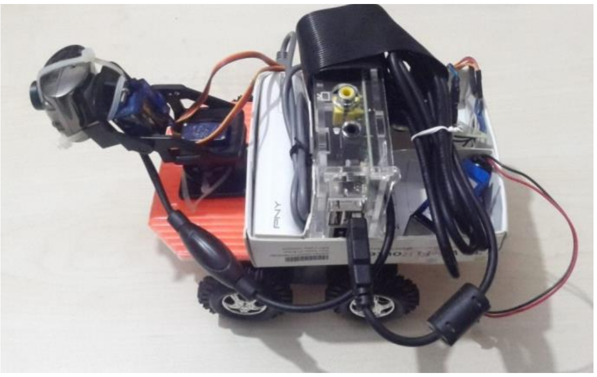

Fig. 8. Designed mobile robot

CONCULUSION AND PROPOSALS

In this research, a mobile robot design is offered using Raspberry Pi (Figure 8). Peripheral monitoring was performed by USB camera on the mobile robot. Mobile robot can be controlled real-time over a webpage, and can be monitored using a camera. During design process, all of the units on the robot were checked directly by Raspberry Pi, so an additional microcontroller system was not used, which enabled an affordable and open source mobile robot design without additional control hardware.

This study is a step to improve mobile robots especially on rough fields, using tracked system instead of wheel system. Various sensors installed on robot might be used to receive and process information. Moreover, this design might be improved more by adding a robot arm system.

ACKNOWLEDGMENT

The study is selected from National Engineering Research Symposium 2015 (Ulusal Mühendislik Araştırmaları Sempozyumu) UMAS 2015 (Duzce University).

REFERENCES

[1] Şahin, İ., Yalvaç, M. (2012). Myrobot: Kablosuz Kontrol Edilebilen Mobil Araştırma Robotu, e-Journal of New World Science Academy NWSA-Education Sciences, 7(1), 340-347.

[2] Çayıroğlu, İ., Şimşir, M. (2008). PIC ve Step Motorla Sürülen Bir Mobil Robotun Uzaktan Kamera Sistemi ile Kontrolü, *ErciyesÜniversitesi Fen Bilimleri Enstitüsü Dergisi*, 24(1-2), 1-16.

[3] Fong, P. E., Yusoff, M. A. (2011). Real-Time Control of Wi-Fi Surveillance Robot, *Proceeding of the International Conference on Advanced Science Engineering and Information Technology*, Malasia.

[4] Ud Din, Z., Lim, W. (2013). Home Automation with Smart Robot Featuring Live Video Feed and Remotely Controlled Switches, *IEEE Conference on Sustainable Utilization and Development in Engineering and Technology*

[5] Kuo, G., Cheng, C., Wu, C. (2014). Design and Implementation of a Remote Monitoring Cleaning Robot, *International Automatic Control Conference*, Taiwan

[6] Wang, Z., Lim, E. G., Wang, W., Leach, M., Man, K. L. (2014). Design of an Arduino–based Smart Car, *International SoC Design Conference*, Korea

[7] Al-Sahib, N. K. A., Azeez, M. Z. (2015). Build and Interface Internet Mobile Robot using Raspberry Pi and Arduino, Innovative System Design and Engineering, 6(1)

[8] Espes, D., Autret, Y., Vareille, J., Le Parc, P. (2014). Designing a Low-Cost Web-Controlled Mobile Robot for Home Monitoring, *The Eighth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies*, Italy

[9] Prabha, S. S., Antony, A. J. P., Meena, M. J., Pandian, S. R. (2014). Smart Cloud Robot using Raspberry Pi, *International Conference on Recent Trends in Information Technology*, India

[10] http://www.friendlyarm.net/products/mini2440 (July '15)

[11] http://beagleboard.org (July '15)

[12] https://www.raspberrypi.org (July '15)

[13] McManus, S., Cook, M., *Raspberry Pi for Dummies*, John Wiley & Sons (2013)

[14] https://www.npmjs.com/package/rpi-gpio (July '15)

[15] The WebSocket Protocol. Internet Engineering Task Force (IETF). http://tools.ietf.org/html/rfc6455 (July '15)

[16] https://nodejs.org/ (July '15)

[17] https://github.com/richardghirst/PiBits/tree/master/ServoBlaster (July '15)

[18] https://github.com/jacksonliam/mjpg-streamer (July '15)

[19] https://jquery.com/ (July '15)

[20] https://github.com/mifi/jquery-joystick (July '15)

BIOGRAPHIES

Enver KÜÇÜKKÜLAHLI was born in Tosya, Turkiye in 1983. He received the B.Sc. degree in Computer Education and Instructional Technology from Faulty of Education, Ondokuz Mayıs University and M.Sc. degrees in Teacher Training in Electrical Works from Faulty of Technical Education, Duzce University, in 2006 and 2011. From 2007 he worked in Computer Technologies, Duzce Vocational School, Duzce University as a Lecturer. His research areas include optimization, image segmentation.

Recep GÜLER was born in Bursa, Turkiye in 1985. He received the B.Sc. and M.Sc. degrees in Electronic & Computer Education from Faulty of Technical Education, Gazi University, in 2008 and 2011. From 2008 to 2011 he worked in Electronic & Computer Education, Faulty of Technical Education, Gazi University as an Assistant Student and also work in BELTEK Educational Course as a Lecturer. His research areas include microcontroller systems, communication systems.