Controlling Self-Directed Vehicle via Webpage using Raspberry Pi Lighttpd Webserver

* Wajeeh Baloch, ** Bhawani Shanker, and *** Sumbul Ghulamani

Abstract — **Internet of Things (IoT) and Tele-Robotics are most advanced technologies that provide a vision where any object in this world can be connected to anything else. These technologies enable us to build communication among various entities such as Wireless Sensor Networks (WSN) devices, home appliances, medical equipment, RFIDs etc. An intelligent Self-Directed Vehicle (SDV) incorporated with features of Internet of Things (IoT) is presented in this work. In order to manually control the SDV, a web page (embedded with google maps) is developed and uploaded on Lighttpd server on Raspberry Pi. Lighttpd server can act like a normal apache server. User can navigate SDV manually using maneuvering control buttons. The onboard camera provides visual awareness. This technology can be utilized and implemented in a number of areas such as military, where SDV can undertake missions like drone surveillance, border patrol etc. Furthermore, our work highlights the use of webservers for interconnecting devices and describes the use of internet to control SDV from a remote end with Raspberry Pi Lighttpd Web Server.**

Index Terms* – *WSN, IoT, Raspberry Pi, Web Servers, SDV, Lighttpd

I. INTRODUCTION

Rapid enhancement in technology has been constantly added to the human life for last few decades. Internet of Things (IoT) and Tele Robotics are two of those latest enhancements in technology. With such technologies it’s possible to control the motion of any device from remote end even in the space like satellites. Internet is the universal medium to build such communication between two ends. Nowadays world is transforming towards the "Smart Era", homes are becoming smartphones, cars are becoming smart cars, phones are becoming smart homes. Which proves that these devices include the features of decision making and user can monitor or control these devices from any remote end. With a web server, it becomes even easier to control and monitor the devices from anywhere. While building such communication between two ends, several security issues and threats arise such as hacking, spamming etc. To avoid these threats, we present an efficient approach of controlling a self-directed vehicle (SDV) via web page that is uploaded on a web server which is set up in a single board computer, this approach can be called server independent approach. Raspberry Pi's Lighttpd web server is one such server.

Use of internet is very common these days, so it’s becoming easy to communicate with devices wirelessly. Our paper presented an internet controlled Self-Directed Vehicle which can be used and utilized in number of different areas. The onboard camera mounted on SDV helps give it visual awareness whereas the webpage to control and monitor the output of camera is uploaded on web Lighttpd web server on Raspberry Pi.

The rest of the paper is organized as follows. Section II describes the Current Technologies. In Section III, Hardware Components are described. Section IV presents the Methodology and System overview. Section V presents the implementation. In section VI, results and findings are discussed. Section VII concludes the paper.

II. RELATED WORK

A. Tele-Robotics

Tele robotics refers to the field of robotics where devices can be controlled and monitored from distance using different transmission mediums such as (Wi-Fi, Bluetooth etc.) Tele-Robotics has two major fields Teleoperation and Telepresence. Our project SDV is concerned with the field of Teleoperation which indicates the controlling of SDV at distance. One of the most successful example of teleoperation is Voyager shown in figure 1, which is a space probe sending the data from other planets remotely. In Teleoperation a human controller controls the movement of the target device in such manner that signals are sent as instructions to the device, when the device receives the signals it sends back the acknowledgement back to the user. This process of sending and receiving the signals is called telemetry. Teleoperation is usually involved in hardware projects where the devices are connected with some remote ends such as SDV.

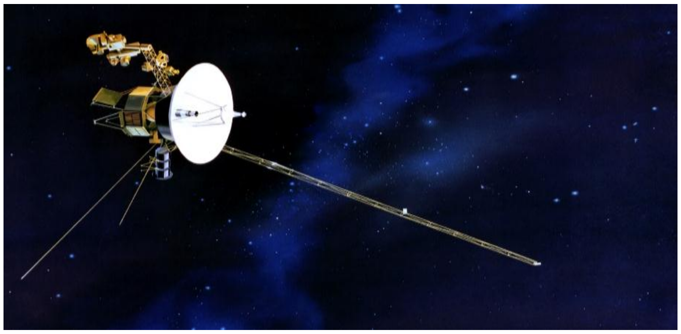

Fig. 1. Voyager Space Craft
Source: Google search query "Voyager Space Craft"

Department of Computer Science, SZABIST Hyderabad Campus
* wajeeh-baloch@outlook.com
** zablakhani@gmail.com
*** sumbul.ghulamani@hyd.szabist.edu.pk

B. Internet of Things

Internet of Things (IoT) refers to the network of physical objects such as vehicles, phones, buildings etc. All these objects are considered as things, as these objects are embedded with sensors, and have network connectivity. IoT allows the devices to be sensed and controlled through remote end using some network. The vision of IoT aims to result in the reality of smart vehicles, smart homes and smart cities etc. According to the estimation of experts it has been stated that IoT devices will increase up to 50 billion by 2020 [1, 7].

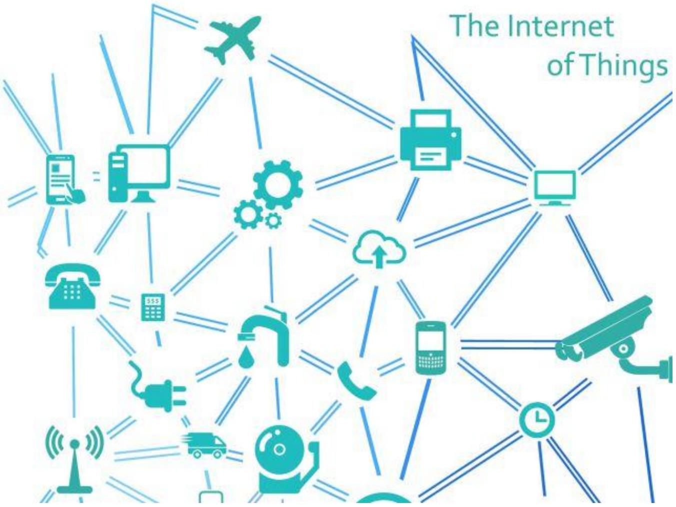

Fig. 2. Internet of Things
Source: Google Search Query "Internet of Things"

C. Web Servers

The primary role of a web server is to manage the web content that can be accessed through the internet to the end users and customers. Hypertext Transfer Protocol (HTTP) is the protocol which results in helping the communication between client and server. Pages delivered are most frequently HTML documents which may include documents, images, style sheets and scripts in addition to text content. Web servers also support server-side scripting using JavaScript's, Active Server Pages (ASP), PHP, or other scripting languages. In applications such as controlling and monitoring some physical device, Apache and Lighttpd are the most favored web server applications for Raspberry Pi [2, 4].

III. HARDWARE COMPONENTS

A. Raspberry Pi

Raspberry Pi (R Pi) is a device which is commonly known as a low cost, small credit card sized computer developed in England, United Kingdom by the Raspberry Pi Foundation. R Pi can be connected with a computer monitor or television, and with standard mouse/keyboard to give commands. It is shown in Figure 3. After installing Linux or Unix as its operating system, it can support all the scripting languages such as Python, JavaScript etc. Raspberry Pi works exactly same as the desktop computer works, users can play games, watch videos, access the internet, and write programs as well.

R Pi is used widely for the hardware projects such as our project the Self-Directed Vehicle in which the purpose is to get the commands from user to control the SDV through a webpage which is uploaded on R Pi itself.

Fig. 3. Raspberry Pi Module

B. Webcam

Logitech webcam is the key component of Self-Directed Vehicle which acts as the on-board camera to monitor the SDV and provides visual awareness of the surroundings to the user. The webcam is connected with Raspberry Pi from where the streaming and output of camera is shown on the webpage to user through Wi-Fi and Internet Adapter. The Logitech webcam is associated with the visual feedback of SDV, with 2 Mega pixel Resolution and Clear video options to experience the live streaming. The webcam is connected with the RPI USB port which is mounted on SDV.

C. Self-Directed Vehicle

Self-Directed Vehicle (SDV) is an autonomous robot car based on Wireless Sensor Networks (WSN) which detects and avoids obstacles, is aware of its location, and is able to stream the onboard video camera (eye) to internet.

SDV is programed to act in both Autonomous as well as Manual control modes. R Pi, and web camera are the basic components of the system, whereas modules such as ultrasonic, GPS, Bluetooth, Wi-Fi Adapter, magnetometer and H-Bridge are supporting components for obstacle detection, sensing, data transfer and real time tracking.

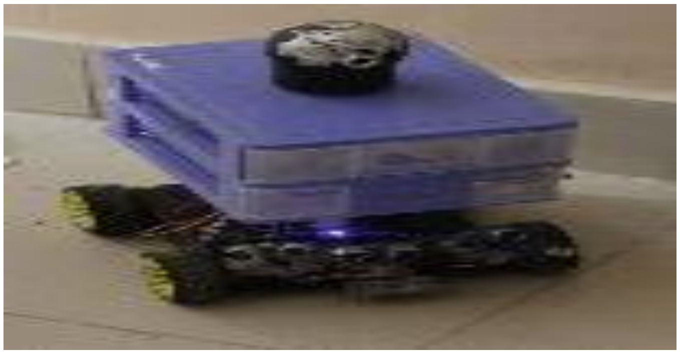

Fig. 4. Snapshot of Self-Directed Vehicle (SDV)

D. GPS

Pam 7Q GPS module is incorporated and mounted on SDV to get the location coordinates, in order to show the current location of SDV on the webpage we use GPS. It updates the location by sending the coordinates from SDV to the base station (Webpage) and on webpage the coordinates are shown inside google maps. With this feature of getting exact location of SDV, user can navigate and reach targeted location easily.

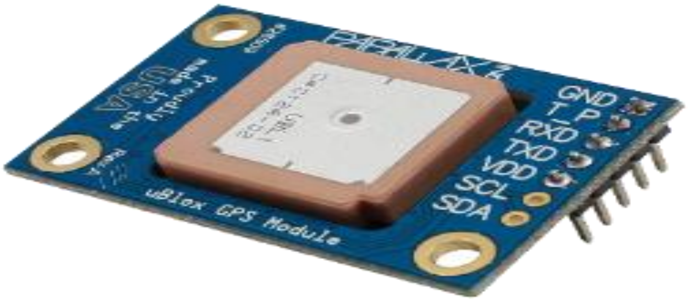

Fig. 5. Pam 7Q GPS Module

IV. System Overview

Raspberry Pi is connected to the hotspot which is created using a Wi-Fi adaptor and the user who is navigating the SDV via webpage is connected to the Wi-Fi. The webcam provides the visual scenario of surroundings SDV to monitor it remotely via webpage. Raspberry Pi requires Secure Shell (SSH) link to display the output of webpage on PC. Using this procedure, we connect and control SDV via webpage from client side PC.

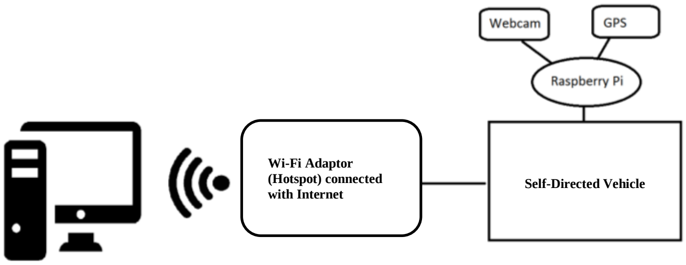

Fig. 6. Block Diagram of SDV

A. Setting Up Raspberry Pi Webserver

Raspberry Pi is considered significantly less powerful device but it can prove to be better as a web server for any useful device. With Raspberry Pi's low power consumption and small size, it can be mounted on any device. In our project the raspberry pi is mounted on SDV which sends and receives data of different sensors. There are several web server apps available in Raspberry Pi such as Lighttpd and Apache. It also supports database for holding data on backend, with multiple functionalities like storing, retrieving of data. Lighttpd server is considered relatively better than apache for Raspberry Pi.
We installed Lighttpd on Raspberry Pi using linux commands -sudo apt-get -y install lighttpd [3, 8]. PHP is also installed in the same manner on Raspberry Pi using Linux input commands. After installing all the basic tools, the Lighttpd webserver is restarted to start and take effect. The directory to save the web server files can be manually given such as /var/sdv/. The files in the directory can be called on the webpage on the request of client system. This directory is set to be owned only by the root user to access these files directly.

V. IMPLEMENTATION

On the very first, it is necessary to install the operating system on Raspberry Pi. We had chosen Linux and installed it onto the SD card of Raspberry Pi. The SD card's memory works as the external storage to perform all the operations because the internal storage and memory of Raspberry Pi is not enough to install all the software including OS. To install the operating system, we need to connect a mouse and a keyboard along with a monitor to Raspberry Pi through USB ports and HDMI/VGA port. An external adaptor was used for power supply. Once the initial configurations and operating system is installed we can setup a desktop and start building things. So we have to connect Raspberry Pi with Local Area Network using Wi-Fi Adaptor, after connecting it with LAN we have to check the IP Address provided to the system using Linux commands. Now after setting up the connection we need to copy all the dynamic webpage files such as index files to the created directory /var/sdv. Raspberry Pi supports all the scripting languages such as C, Java, Ruby, Python etc. Among all of them Python is considered the preferable one. Python code programs are written for maneuvering control buttons such as Forward, Backward, Left, Right buttons and linked to the scripts written in PHP to make them functional. Figure 10 shows the live streaming and control buttons of SDV, the address for this manual webpage is 192.168.12.1.

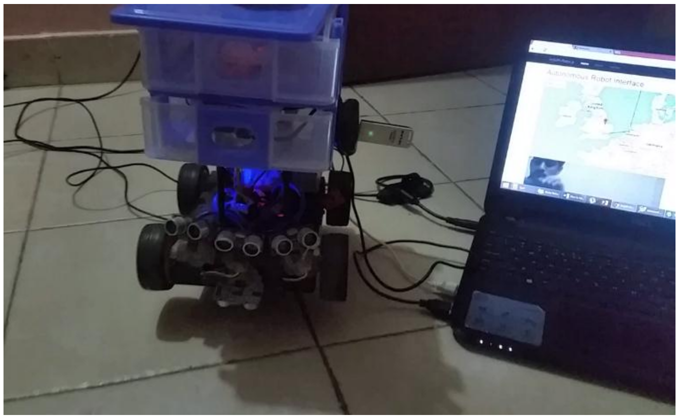

Fig. 7. The experimental SDV setup

The block diagram shown in figure 8 of SDV for controlling SDV through a webpage includes the Raspberry Pi Module and Webcam connected with Wi-Fi Adapter to send and receive signals through Local Area Network Communication. A web page - embedded with google maps - is developed and uploaded on Lighttpd server on Raspberry Pi

[5, 8]. The motors of vehicle kit in SDV works on 5v external power supply. All the components are connected to R Pi and programmed to efficiently send and receive the user commands through Wi-Fi Adapter from the webpage.

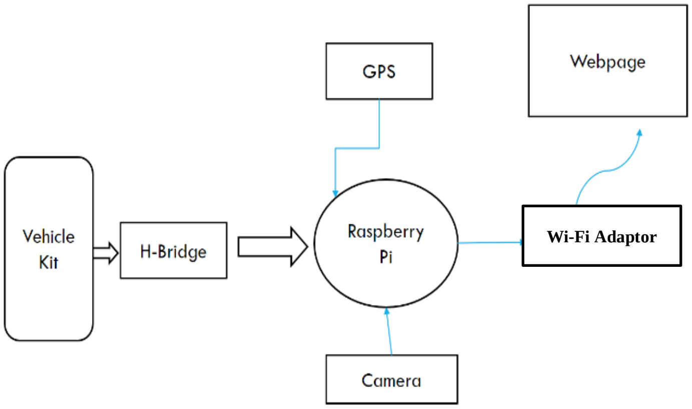

Fig. 8. Raspberry Pi Connection with Sensors and H-Bridge

VI. Results

A. *Live streaming of SDV's surrounding on the webpage*

The onboard camera on SDV provides the visual feedback, in order to achieve the live streaming which is the most integral and prior part of this project as shown in the figure below we need to create a connection between camera and webpage [6, 9]. The camera is connected with Raspberry Pi through a USB port, its output can be easily viewed on any browser by providing the IP address on the address bar such as 192.168.12.1:8081

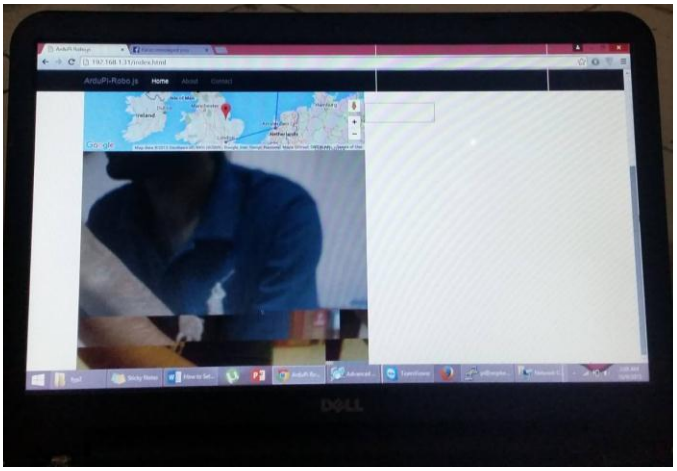

Fig. 9. Live Video Streaming via port 8081

B. *Controlling SDV with Live Video Streaming*

Once the live streaming is achieved, we need to integrate video streaming on the webpage along with the maneuvering control buttons to control and monitor the SDV via webpage. The basic Forward, Backward, Left, Right navigation buttons are created to move the SDV in any direction. These buttons are created using the PHP scripts and AJAX programming so that user can press or click these buttons to navigate SDV. Each time the button is pressed, it will execute the python program on the server side to perform the desired action. The use of AJAX benefits our project by not reloading the webpage again and again on the movement of SDV [10, 11]. Figure 10 shows the live streaming and control buttons of SDV, the address for this manual webpage is 192.168.12.1.

VII. Conclusion and Future Work

The main purpose of this project is to manually control the SDV via webpage and monitor its surroundings with live video streaming. Project SDV also supports the autonomous mode where it detects and avoids obstacles using ultrasonic sensors and GPS to reach the near safest location. This project is funded by National ICT R&D Pakistan, SDV is also tested in Computer Science Research Laboratory at SZABIST Hyderabad in December 2015 and has successfully passed the test. This system can be implemented in several different areas such as military, home security etc. In future we tend to implement path planning and image processing features to SDV by which it can recognize and identify shapes and places.

Acknowledgment

We acknowledge National ICT R&D Fund for funding our project under National Grassroots ICT Research Initiative Program

References

[1] Pannu Gurjashan Singh; Ansari Mohammad Dawud; Gupta Pritha – Design and Implementation of Autonomous Car using Raspberry Pi, International Journal of Computer Applications 2015.

[2] V. Ramya, B. Palaniappan, T. Akilan ―Embedded System for Robotic Arm Movement Control using WebServer and ZigBee Communication‖, International Conference on Research Trends in Computer Technologies, Proceedings published in International Journal of Computer Applications® (IJCA)(0975 – 8887), pg 30-34, 2013

[3] Wan MuhamadHanif Wan Kadira, Reza EzuanSaminb, Babul Salam Kader Ibrahim C. ―Internet Controlled Robotic Arm‖, Science Direct , International Symposium on Robotics and Intelligent Sensors , pg 1065-1071, 2012.

[4] Lung Ngai, Wyatt S. Newman, VincenzoLiberatore, ―An Experiment in Internet-Based, Human-Assisted Robotics‖, Case Western Reserve University, pg-1011-1015,IEEE, 2002. Huosheng Hu, Lixiang Yu, PuiWoTsui, Quan ―Internet-based Robotic Systems for Teleoperation‖ International Journal of Assembly Automation, Vol. 21, No. 2, pg 1-10, 2000

[5] Berman and Y. Edal, "Evaluation automatic guided vehicle systems," 2008.

[6] S. Lan Li, "Study of Security Architecture in the Internet of Things," in Measurement, Information and Control (MIC), 2012, Volume: 1, pp. 374-377

[7] R.A. Kadu, Prof. V.A. More, P.P. Chitte, J.G. Rana, M.R. Bendre . ―Wireless Control & Monitoring of Robotic Arm‖ International Journal of Computer Technology and Electronics Engineering (IJCTEE) Volume 2, Issue 1, pg 28-38, 2010

[8] Lighttpd Web Server http://www.lighttpd.net

[9] " The Internet of Things," ITU Report, Nov 2005

[10] Hui Suo, Jiafu Wan, Caifeng Zou, Jianqi Liu, "Security in the Internet of Things: A Review," in Computer Science and Electronics Engineering (ICCSEE), 2012, pp. 648-651