International Journal of Engineering, Science and Mathematics
Vol. 6 Issue 8, December 2017 (Special Issue)
ISSN: 2320-0294 Impact Factor: 6.765
Journal Homepage: http://www.ijesm.co.in, Email: ijesmj@gmail.com Double-Blind Peer Reviewed Refereed Open Access International Journal - Included in the International Serial Directories Indexed & Listed at: Ulrich's Periodicals Directory ©, U.S.A., Open J-Gage as well as in Cabell's Directories of Publishing Opportunities, U.S.A

IDENTIFYING OBJECTS IN HAZARDOUS LOCATIONS BY CAM ROBOT USING RASPBERRY PI AND IOT TECHNOLOGY

Gedela Sireesha Devi*
Mr. K. Satya Narayana Murthy**
Allu Gowri Sankar***

Abstract

Internet of Things have received significant attention during these days mainly because of their numerous applications. They can be found in robotics, smart buildings, fabrication equipments as well as medical, aautomation, industrial, commercial, military applications. Most of the modern Internet of Things based on Raspberry pi and Auridino boards and use it for constructing mobile robots. By interfacing a Wireless and Cam module into the proposed IoT system, it can provide a control system that uses Wireless as a standard technology for connecting remote devices. The control circuit of the proposed system is designed in a way such that a user can interface any kind of desirable peripheral.Furthermore, the cam robot is provided with a broadcast ip will open a web-page which has video screen for surveillance and buttons to control robot and camera.

Keywords:

Internet of Things, Raspberry pi, Raspbian OS,putty,wireless networks,python programming,Usb cam, Robot kit,motion tool

Author correspondence:

Sireesha Devi Gedela,
She is a student of Baba Institute of Technology and Sciences, Visakhapatnam. Presently she is pursuing her M.Tech from this college and she received her MCA from Gayatri College of PG Courses, Srikakulam. Her area of intrest includes Internet of Things, App Development and Python Programmig.

1.Introduction

Recent years have witnessed a remarkable growth in the features and applications of Internet of Things. Specifically, advances in IoT with computational capacity ready to support sophisticated intelligent mechanisms and the ability to host an overabundance of sensors, cams hence providing a new degree of freedom to sophisticated applications. Internet of Things usually stand for the integration of a smart system and its software into a larger system, which often monitors and controls equipment without the need for manual intervention. In other words, they are computingsystems that have computer hardware, software and internet embedded into them to perform computationally predefined tasks repeatedly on using significant amounts of application-specific fixed function logic. They can be found wide variety of devices such as Smart watches, Smart phones, Smart Electricals, Smart microwave ovens, Smart Home Security systems, Smart Scanners, Smart Home Appliances as well as other applications (e.g. Agriculture, medical, military etc.,).

*She is a student of Baba Institute of Technology and Sciences, Visakhapatnam. Presently she is pursuing her M.Tech from this college and she received her MCA from Gayatri College of PG Courses, Srikakulam. Her area of intrest includes Internet of Things, App Development and Python Programmig.

**He has a dedicated teaching experience of Eight years. He completed B.Tech from GVPCOE, Vizag and M.Tech from Andhra University,Now he was deeply involved in doing Research, Areas of Research interests includes Data mining, Image Processing, Computer Networks & Cloud Computing

*** Completed M.tech in Embedded systems, Continue research in IOT and realtime projects on embedded systems.

Generally, the IoT Technology use a combination of hardware, software with Internet to respond to events in the environment within defined constraints. In this context, a typical IoT resonds to the environment using sensors, cam and controls the environment using constraints of those systems. One of such constraints is that because of the restrictions and single function repeatedly in contrast to a computer system which can do several functions at the same time. Another constraint is that the IoT Technology should react to the changes in its environment and respond to events in real-time without delay.

On the Other hand, with the increasing population in the mobile robotic applications, there are calls for the design and implementation of larger and more complicated robotic architectures. The Internet of Things represents a vision in which the Internet extends into the real world embracing everyday objects. Physical items are no longer disconnected from the virtual world, but can be controlled remotely and can act as physical access points to Internet services. An Internet of Things makes computing truly ubiquitous – a concept initially put forward by Mark Weiser in the early 1990s. This development is opening up huge opportunities for both the economy and individuals. However, it also involves risks and undoubtedly represents an immense technical andsocial challenge. Now a days Internet technology has low power consumption and a secure way to connect and exchange information between devices, many researchers utilize it for connecting remote devices, propose a home appliance control system over Internet that enables remote control for home appliance with a mobile phone.

Indeed, the mobile cam robot will play an important role in our daily life in the near future, especially in military and security applications. In this paper, the Raspberry pi Board is used for designing and implementing effective IoT based cam robot. The developed system provides a control system that uses Internet – Wifi connection for connecting remote devices. For providing mechanical movements, we interface motor drivers to the Raspberry pi. In oreder to implement the whole system, software has to be written for the robot and the remote device. The code for the robot is implemented by Python Code and the graphical user interface using python and Linux environment. For cam robo navigation the user has to press the connect button, once the connection is established the user can control the cam robot in any direction. The proposed cam robot can be deployed in many applications such as military applications, intelligence transportation, seaport automation, aitport automation, agricultural applications.

2. Hardware Components

This section describes the hardware components used in designing the embedded system. The following hardware components are used in designing the mobile cam robot:

2.1 RASPBERRY PI

The Raspberry Pi is a low cost, credit-card sized computer that plugs into a computer monitor or TV, and uses a standard keyboard and mouse. Pi is a single-board computer". Pi was introduced as an educational gadget to be used for protyping by hobbyists and for those who want to learn more about programming. It certainly cannot be a substitute for our day to day Linux, Mac or Windows PC.

Pi is based on a Broadcom SoC (System of Chip) with an ARM processor [~700 MHz], a GPU and 256 to 512 MB RAM. The boot media is an SD card [which is not included], and the SD card can also be used for persist data. Now that you know that the RAM and processing power are not nearly close to the power house machines you might have at home, these Pi's can be used as a Cheap computer for some basic functions, especially for experiments and education. The Pi comes in three Configurations and we will discuss the specifications of those in the coming sections. The cost of a Pi is around $35 for a B Model and is available through many online and physical stores.

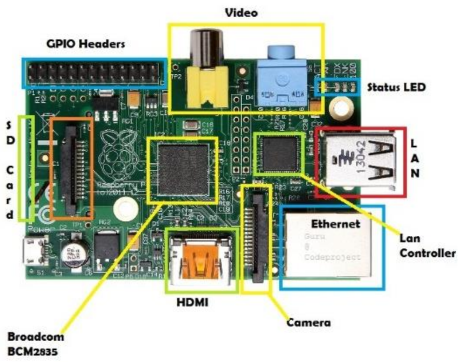

Figure 1. *Raspberry pi*

Several generations of Raspberry Pis have been released. All models feature a Broadcom system on a chip (SoC) with an integrated ARMcompatible central processing unit (CPU) and on-chip graphics processing unit (GPU). Processor speed ranges from 700 MHz to 1.2 GHz for the Pi 3 and on-board memory range from 256 MB to 1 GB RAM. Secure Digital (SD) cards are used to store the operating system and program memory in either SDHC or MicroSDHC sizes. Depending on the model; The boards have either a single USB port or up to four USB ports. For video output, HDMI and composite video are supported, with a standard 3.5 mm phono jack for audio output. Lower level output is provided by a number of GPIO pins which support common protocols like I^2C. The B-models have an 8P8C Ethernet port and the Pi 3 and Pi Zero W have on-board Wi-Fi 802.11n and Bluetooth.

An operating system is just a very complicated program. It has the job of organising other programs on a computer, including sharing the computer's time, memory, hardware and other resources. Some big families of desktop operating systems that you may have heard of include GNU/Linux, Mac OS X and Microsoft Windows. Other devices also need operating systems such as phones, which may use operating systems such as Android, iOS and Windows Phone.[1]
Since the operating system has to interact with the hardware on a computer system, it also has to have specific knowledge of the hardware on a system. To allow operating systems to be used on a variety of computers, the concept of **drivers** was invented. Drivers are small bits of code that can be added and removed from the operating system in order to allow the operating system to talk to a particular piece of hardware. In this course, we do not cover how to create such removable drivers, and instead focus on making specific ones for the Raspberry Pi.

When programming normally, the programmer writes code in a programming language such as C++, Java, C#, Basic, etc, and then a program called the compiler translates what the programmer wrote into assembly code, which is the further reduced into binary code[2]. Binary code is what the computer actually understands, but it is almost impossible for humans to read. Assembly code is much better, but it can be frustrating how few commands are possible. Remember that every command you write in assembly code is something that the processor understands directly, and so the commands are simple by design, as a physical circuit must process each one.

2.2. Raspbian Os

Raspbian is a free operating system based on Debian optimized for the Raspberry Pi hardware. An operating system is the set of basic programs and utilities that make your Raspberry Pi run. However, Raspbian provides more than a pure OS: it comes with over 35,000 packages, pre-compiled software bundled in a nice format for easy installation on your Raspberry Pi.

The initial build of over 35,000 Raspbian packages, optimized for best performance on the Raspberry Pi, was completed in June of 2012. However, Raspbian is still under active development with an emphasis on improving the stability and performance of as many Debian packages as possible.

Raspbian was created by Mike Thompson and Peter Green as an independent project. The initial build was completed in June 2012. The operating system is still under active development. Raspbian is highly optimized for the Raspberry Pi line's low-performance ARM CPUs.Raspbian uses PIXEL, **P**i **I**mproved **X**windows **E**nvironment, **L**ightweight as its main desktop environment as of the latest update. It is composed of a modified LXDE desktop environment and the Openbox stacking window manager with a new theme and few other changes. The distribution is shipped with a copy of computer algebra program Mathematica and a version of Minecraftcalled Minecraft Pi as well as a lightweight version of Chromium as of the latest version.

Raspian OS is one of the official Operating systems available for free to download and use. The system is based on Debian Linux and is optimized to work efficiently with the Raspberry Pi computer. As we already know an OS is a set of basic programs and utilities that runs on a specified hardware, in this case the Pi. Debian is very lightweight and makes a great choice for the Pi. The Raspbian includes tools for browsing, python programming and a GUI desktop.

The Raspian desktop environment is known as the "Lightweight X11 Desktop Environment" or in short LXDE. This has a fairly attractive user interface that is built using the X Window System software and is a familiar point and click interface. We shall look more into how to install and use this OS in the next section.

Setting Up Raspian OS:
Let's first connect the board with all the necessary accessories to install and run an operating system.
Step 1: Take the Pi out of its anti static cover and place it on the non-metal table.
Step 2: Connect the display – Connect the HDMI cable to the HDMI port on the Pi and the other end of the HDMI cable to the HDMI port of the TV.
Step 3: Connect your Ethernet cable from the Router to the Ethernet port on the Pi
Step 4: Connect your USB mouse to one of the USB ports on the Pi
Step 5: Connect your USB Keyboard to the other USB port on the Pi
Step 6: Connect the micro USB charger to the Pi but don't connect it to the power supply yet
Step 7: Flash the SD Card with the Raspian OS.

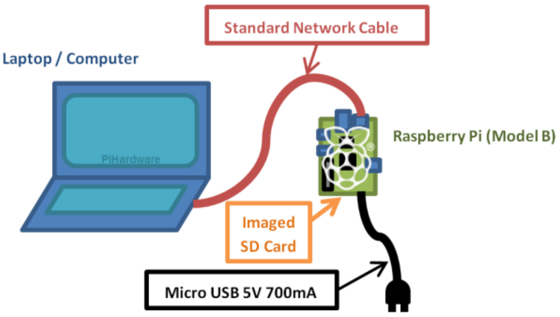

Figure 2. *Raspberry pi Remote Connection*

2.3. Wireless Networks

When the term 'wireless network' is used today, it usually refers to a wireless local area network (WLAN). A WLAN connects computers together through radio technology using standard network rules or protocols, but without the use of cabling to connect the computers together. Internet of things applications have diverse connectivity requirements in terms of range, data throughput, energy efficiency and device cost. WiFi is often an obvious choice because in-building WiFi coverage is almost ubiquitous, but it is not always the appropriate choice. This examines the role WiFi can play and two emerging IEEE standards, 802.11ah and 802.11ax.

Data transfer requirements for IoT vary from small, intermittent payloads like utility meters to large amounts of continuous data such as real-time video surveillance. Range requirements can span from very short distance for wearable to several kilometers for weather and agriculture applications.

One thing that is common is power constraint. IoT devices require constant connectivity, but they may not always have continuous access to a power source IoT devices are expected to be poer thrifty, sometimes to the extent of running button cells for years at a stretch.

WiFi, or 802.11 is a wireless protocol that was built with the intent of replacing Ethernet using wireless communication over unlicensed bands. Its goal was to provide off-the-shels, easy to implement, easy to use short-range wireless connectivity with cross-vendor interoperability. With zero spectrum cost, there was little focus on spectral efficiency and with expected use by desktop devices, power efficiency was not critical.

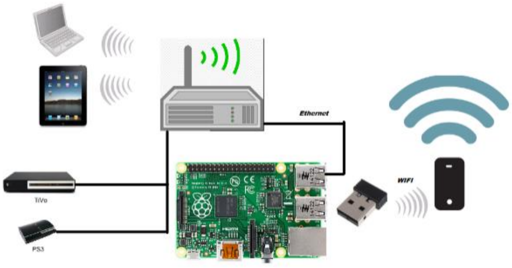

Figure 3. *Raspberry pi WiFi Connection to Laptops/Mobile devices*

3. Python Programming

Python was conceived in the late 1980s, and its implementation began in December 1989 by Guido van Rossum at Centrum Wiskunde&Informatics (CWI) in the Netherlands as a successor to the ABC language (itself inspired by SETL) capable of exception handling and interfacing with the operating system Amoeba.Well there are two main aspects to it. The language you can use to program the MCU you'd use in your IoT project; and the language used on your computer to interact with the IoT project.

That's an even larger field of possibilities. When you are working on an IoT project, with say, an Arduino, Raspberry pi, once you uploaded the code onto the MCU, the MCU and your computer can interact, usually via a serial port(USB, Bluetooth) or WiFi/Ethernet, and this is when python could come in handy. You could aggregate the data coming from one or more MCUs, pass on commands and display/log results, etc. this can be done with many languages, including python. There's no requirement for speed, since MCUs are slow and python is good enough to interact with these slow machines. And it's easy to learn and code with.

4. Raspberry pi Camera

Cams typically include a lens, an image sensor, support electronics, and may also include a microphone for sound. Video surveillance is the process of monitoring a situation, an area or a person. This generally occurs in a military scenario where surveillance of borderlines and enemy territory is essential to a country's safety. Human surveillance is achieved by deploying personnel near sensitive areas in order to constantly monitor for changes. But humans do have their limitations, and deployment in inaccessible places is not always possible.

There are also added risks of losing personnel in the event of getting caught by the enemy. With advances in technology over the years, it is possible to remotely monitor areas of importance by using robots in place of humans.

In this paper we developed a robot which can be used for video surveillance and monitoring which can be controlled through a Mobile device. The control mechanism is provided with a video transmission facility. The video transmission is practically achieved through high-speed image transmission. Initially, the robot will be equipped with a camera which will capture scenes and transfer to the SD card.

Figure 4. *Raspberry pi Camera*

5. System Architecture

The architecture and block diagram of the proposed IoT system of the mobile cam robot is shown in Fig. 6. The main controlling device in the system is the Raspberry pi. We interface the WiFi module mentioned above to provide wireless communication. For providing mechanical movements, we interface motor drivers to the microcontroller. In order to make robot avoid crashing with any obstacles in its path, the GP2Y0A02YK0F sensor is used to measure the distance between the body of the robot and any object in its path.

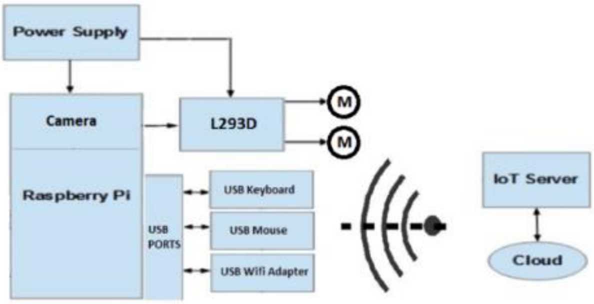

Figure 5. Block diagram of the proposed system

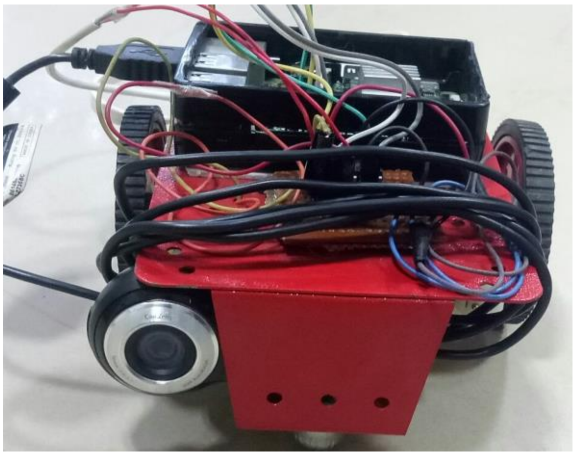

Figure 6. A front view of the proposed system

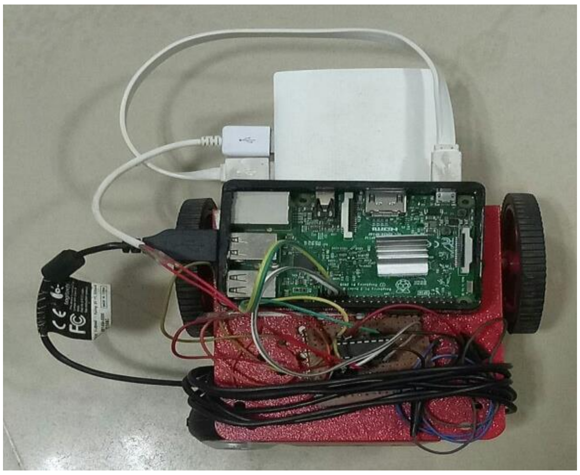

Figure 7. A Top view of the proposed system

6. Putty

Putty is a free and open-source terminal emulator, serial console and network file transfer application. It supports several network protocols, including SCP, SSH, Telnet, rlogin, and raw socket connection. It can also connect to a serial port. The name "Putty" has no definitive meaning.Putty was originally written for Microsoft Windows, but it has been ported to various other operating systems. Official ports are available for some Unix-like platforms, with work-inprogress ports to Classic Mac OS and mac OS, and unofficial ports have been contributed to platforms such as Symbian, Windows Mobile and Windows Phone,Putty supports many variations on the secure remote terminal, and provides user control over the SSH encryption key and protocol version, alternate ciphers such as 3DES, Arcfour, Blowfish, DES, and Public-key authentication. It also can emulate control sequences from xterm, VT102 or ECMA-48 terminal emulation, and allows local, remote, or dynamic port forwarding with SSH (including X11 forwarding). The network communication layer supports IPv6, and the SSH protocol supports the zlib@openssh.com delayed compression scheme. It can also be used with local serial port connections. Putty comes bundled with command-line SCP and SFTP clients, called "pscp" and "psftp" respectively, and plink, a command-line connection tool, used for non-interactive sessions.

7. Software Implementation

In order to test the proposed system, software has to be written for the mobile cam robot components as well as the remote device. The code for the robot is implemented by code Putty (embedded C) to handle the motors, sensors and any data coming from the remote device. While, the code for the remote device is implemented using other programming languages, namely python. The flowchart for the GUI program shown in Fig. 8 illustrates the various steps that have to be followed to perform successfully various motions of the robot. First, user starts the software that strats immediately searching for the available ip connection. Second the user has to press the connect button; if the connection is opened properly, it will display a message for a successful connection, and otherwise an error message will be displayed. After opening a connection, a bidirectional stream between the mobile device and robot is established to send and receive data. Finally, the user can easily control the motion of the robot in any direction.Fig. 9 illustrates the GUI designed in python with several buttons for controlling the motion of the robot in a specific direction by viewing the live streaming

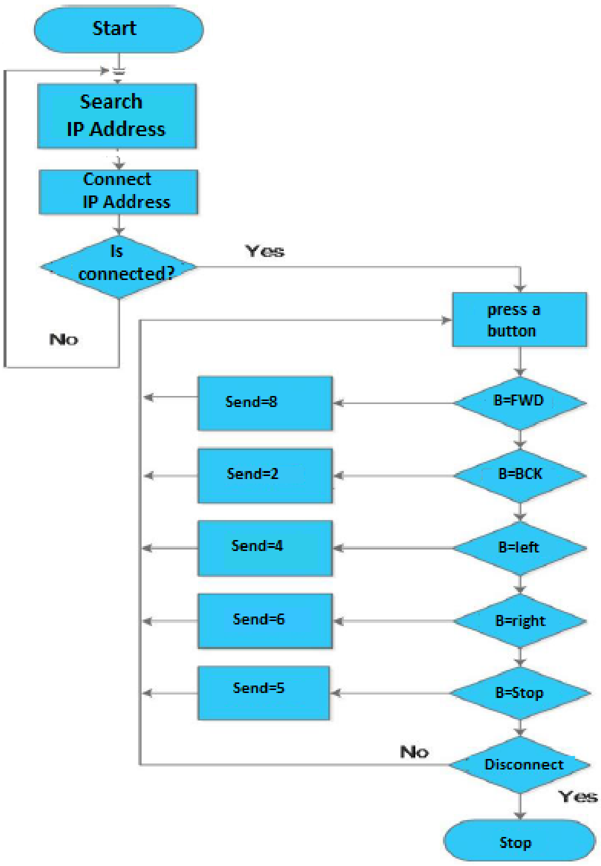

Figure 8. A Flowchart for the python GUI program

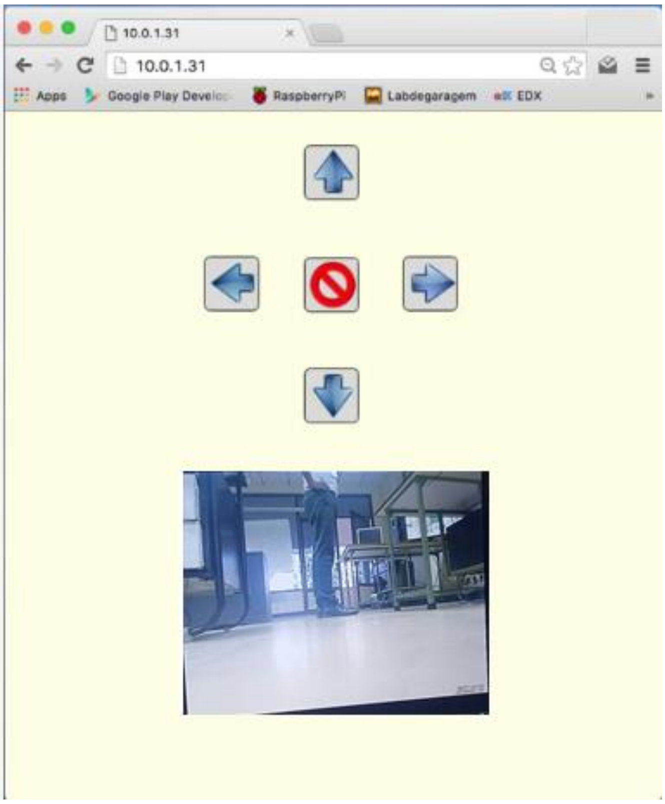

Figure 9. GUI for remotely controlling the motion of the mobile cam robot

8.System Description

The controlling of robot using system or Laptop makes the system efficient and reliable. A Wi-Fi dongle is used for connecting Raspberry Pi to Wi-Fi router. The Wi-Fi router is connected to local area network (LAN) for providing internet connectivity to the Pi.

As soon as the Raspberry Pi gets power it starts initializing the IP address allocated to the web server and establishes connection with the WiFi. Once it gets connected successfully, it remains idle until the user sends any command to it. After receiving command the Raspberry Pi decodes the command and starts capturing the video with required robotic motion.

9. Conclusion

The Raspberry Pi is used to control robot using System or laptop from remote location. The time required for processing the commands from the system or laptop and responding accordingly is negligible. The experimental result has proved that the fetching of a good quality video is quick and clear which is up to 90 frames per second.

References

[1] Jun Zhang Guangming Song "An Indoor Security with a Jumping Robot as the Surveillance Terminal" <em>IEEE Transactions on Consumer Electronics</em> vol. 57 no. 4 November 2011.

[2] Christian Micheloni "An Autonomous Vehicle for Video Surveillance of Indoor Environments" <em>IEEE Transaction on Vehicular Technology</em> vol. 56 no. 2 March 2007

[3] RitikaPahujalNarender Kumar "Android Mobile Phone Controlled Bluetooth Robot Using 8051 Microcontroller" <em>International Journal of Scientific Engineering and Research (IJSER) 324–329</em> 2012.

[4] [online] Available: http://akizukidenshi.com/download/ds/rs/raspberrypi2b.pdf.

[5] L.L. Yang S.H. Yang F. Yao "Safety and Security of Remote Monitoring and Control of intelligent Home Environments" <em>IEEE Int. Conf. on Systems Man and Cybernetics</em> vol. 2 pp. 1149-1153 2006

[6] Christian Hernandez RacielPoot Lizzie Narvaez Erika Llanes Victor Chi "Design and Implementation of a System for Wireless Control of a Robot" <em>International Journal of Computer Science</em> vol. 7 no. 5 pp. 163-169

[7] KeerthiPremkumar K Gerard Joe Nigel "Smart Phone Based Robotic Arm Control Using Raspberry Pi Android and Wi-Fi" <em>IEEE Sponsored 2nd International Conference on Innovation in Information Embedded and Communication Systems</em> 2016.

[8] [online] Available: http://appinventor.mit.edu/explore/sites/all/files/Resources/ThesisFinaLAnshulBhagi.pdf

[9] Matt Richardson Shawn Wallace <em>Getting Started with Raspberry Pi. O'Reilly Media</em>.

[10] [online] Available: http://jacobsalmela.com/raspberry-pi-webcam-using-mjpg-streamer-over-internet

[11] [online] Available: https://www.cs.uky.edu/~keen/115/Haltermanpythonbook.pdf.

[12] [online] Available: https://wolfpaulus.com/joumal/embedded/raspberrypi_webcam/.

[13] International Conference on Communication and Signal Processing, April 6-8, 2016, India.