

## **A Simple and Efficient Wireless Control Protocol for Small Scale Robotic Arm**

Shiu Kumar\*

*Department of Electronics Engineering  
Fiji National University  
kumar.shiu@hotmail.com*

Seong Ro Lee

*Department of Electronics Engineering  
Mokpo National University  
srlee@mokpo.ac.kr*

### **Abstract**

*High accuracy and repetitive tasks have become the two contradictory needs of any industrial process. By introducing autonomous robotic applications, simple repetitive tasks can be accomplished keeping the demands of the accuracy and speed in mind. However, developing these applications for industries becomes a major problem to be tackled in terms of cost. In this paper, the design, implementation and experimental results of a simple and efficient wireless control protocol for small scale robotic arm is presented. The main objective of the research was to investigate RF transmission algorithms, design and implement, and obtain a small cost effective full remote control of a robotic arm without the use of a Personal Computer. The system is equipped with joysticks, two Arduino microcontrollers communicating via RF link, and a servo controller. Joysticks are used to fully control the motion of the robotic arm.*

**Keywords:** *Robotic Arm, Pulse Width Control, Data Packets, Encapsulation, Decapsulation, Radio Frequency (RF), Servo Controller*

### **1. Introduction**

High accuracy and repetitive tasks have become the two contradictory needs of any industrial process. Within factories around the world, robots perform heroic feats of manipulation on a daily basis. They lift massive objects, move with blurring speed, and repeat complex performances with unerring precision. Yet outside of carefully controlled settings, even the most sophisticated robot would be unable to get you a glass of water. The everyday manipulation tasks we take for granted would stump the greatest robot bodies and brains in existence today.

Many surveys [1] have reported different tasks which a disabled user might use an assistive robot for. Other papers [2] have described the use of robots in real life situations. It is not appropriate to repeat these statistics. However it is useful to divide the tasks briefly into groupings; eating and drinking, personal hygiene, work, leisure and mobility.

Many of these task areas are common to all assistive robot systems. However some tasks are more appropriate for a fixed site workstation, perhaps used for a vocational

application, while others are more specific to a wheelchair mounted robot. These tasks include general reaching operations as well as more specific tasks related to mobility such as opening doors and windows and operating switches (e.g. light switches, lift call buttons).

Many different approaches to assistive robotics have been both suggested and implemented. Whilst in some situations (for example a vocational setting) a fixed site workstation is suitable [3], in other cases (for example someone living independently in their own home) a mobile device [4] is more appropriate. A project at Bath Institute of Medical Engineering Ltd was the implementation of a low cost mobile robot by mounting a manipulator on a simple non-powered trolley base, which could be moved around the home by a carer. In order to extend the flexibility of this system, the same manipulator was then mounted onto an electric wheelchair [5]. The present paper presents the outcome of the study on controlling a robotic arm remotely by the use of joysticks with microcontrollers communicating via RF link. The proposed architecture highlighted in this paper will eliminate the use of PC and presents a cost effective design.

This paper is organized as follows. The second section of the paper gives an overview of related works that has been carried out. Section III introduces the 6 degree of freedom (DOF) Robotic Arm. In sections IV-VI, the newly proposed RF control architecture is presented. Section VII discusses about the different tests that has been carried out. A general conclusion ends the paper and presents some perspectives.

## **2. Related Work**

Robotic arms have been designed and developed over the past decades to perform a variety of functions. There have been many variations of reaching devices in the national Science Foundation Engineering Senior Design Projects for persons with disabilities. One such device was the “Clutching and Gripping Device” made by Arizona State University [6]. For this device their client had a motorized scooter, for mobility and needed an assistive device to grasp and collect items out of his reach. Another reaching device project was done by the University of Massachusetts called the “Assistive Reach Mechanism”. This device reached objects up to four feet away and lifted up to five pounds. The device had the ability to pick up objects as small as a needle or as large as a can with the gripping claw.

Furthermore, another such robotic arm is an Innovative Multi-DOF Robotic Assembly for Press Shops [7]. This project deals with the design, fabrication and control of a robotic arm used to load metal sheets into a press. Two stepper motors control the motion of the arm while one controls the orientation of the wrist. Robot motion is controlled using proximity sensors placed at suitable locations on the machine press itself. The motor control is achieved using a microcontroller.

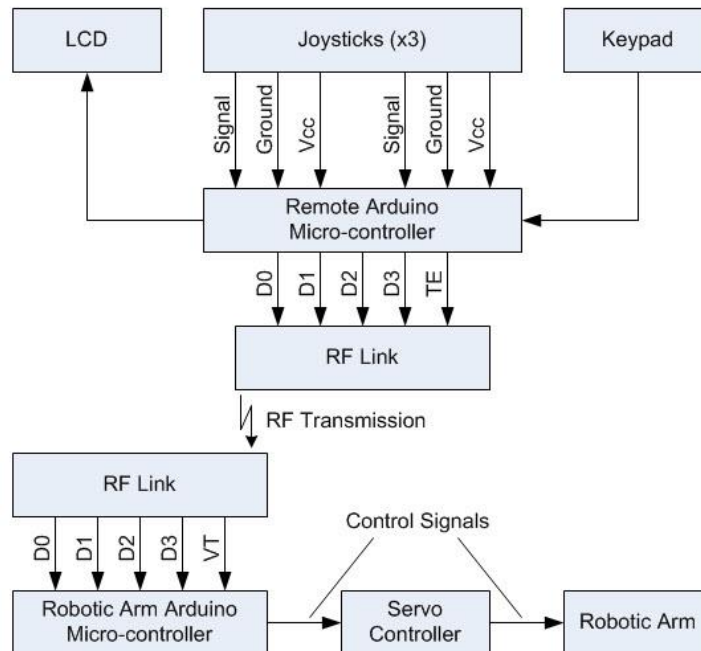
Moreover, the Gateway design team developed a robotic arm that is mounted to a wheelchair with the specific goal of helping paraplegics and quadriplegics function in their immediate environment [8]. The robotic arm for wheelchair project has been undertaken at Wright State University from 1996. However, the arm was first mounted successfully to the wheelchair by the 1998-1999 design group. The 2000-2001 group was given specific requirements since in the previous years there were a lot of drawbacks in the design. The team was successfully able to design the arm with the given requirements, however an attempt ought to be made to reduce the weight of the arm was recommended.

In order to control an assistive robot arm, most of the researchers have considered full control of the robot arm by the user at the time of operation. However, An Intelligent Assistive Robotic Manipulator which is equipped with stereovision detection in order to distinguish different objects from the background has been developed [9]. Two pre-calibrated digital cameras, facing downwards simultaneously take pictures of the surface where the object is located and the information is processed. In order to detect and find location of the object, and the needed orientation of the arm with respect to the object, scene interpretation method based on the stereo/multi vision technique [10] was applied. However, laser beams are widely used as an active system of object detection [11]. Another type of vision based robotic technique relies on the use of a marker on the object in order to make the process of object detection easier [12].

A robotic arm based on haptic technology with four degrees of freedom, which is designed for picking a specific weight and placing it at the desired location is presented in [13]. The arm is controlled via a remote, which is an arm made of Polycarbonate. Potentiometers are fitted to the joints having certain angle of rotation. Arduino duemilanove comprising of ATmega-328 is used as the microcontroller that reads the potentiometer input (electrical signal) and converts the electrical signal to digital pulses (PWM), which drives the servomotors or the arm.

In [14] a 3D robotic arm that is controlled with stereo cameras made using LabVIEW is presented. Reference [15] presents Imirok, a system using user motion to remotely control robotic arms. Low-cost, off the self mobile devices and webcam with motion tracking algorithm is used for real time motion detection. Furthermore, a voice controlled robotic arm is presented in [16].

In all the above work that has been carried out, either on the site control is required or the arm is controlled remotely. Where the arm is controlled remotely, a PC has been used. This paper intends to eliminate the use of PC and have full remote control of the robot arm using joy sticks only by the use of micro-controllers via a RF-link.



**Figure 1. Overall Block Diagram of the Proposed System**

### 3. Design of the Proposed System

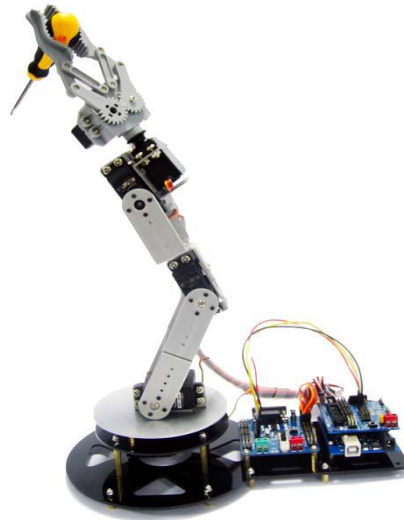
This section details the design of the proposed system carried out at Fiji National University. The 6 DOF robotic arm has been assembled using purchased materials as these decreased the development time and on the other hand, ensure less mechanical deficiencies in the system. It consists of three joysticks, two Arduino microcontrollers, RF links, keypad and liquid crystal display (LCD), and a servo controller. A block diagram of the system is shown in Figure 1.

#### 3.1. Operation of the Proposed System

The small scale robotic arm that has been assembled and control algorithm designed as such that it is capable of picking up any object (max 200g) with a maximum gripper opening of 42mm within its workspace range. The robotic arm when turned on initializes to its home position, with home positions as shown in Table 1.

**Table 1. Home Positions of the Robotic Arm**

Motor	Motor Position
Base Rotation	1600
Shoulder	1200
Elbow	0600
Gripper Movement	1200
Gripper Rotation	1500
Gripper	1600



**Figure 2. The Remote Controlled Small Scale Robotic Arm**

The initialization process takes twelve seconds for this particular application; however this can be changed depending on the requirements for other applications. A red LED is lit indicating that the robot is initializing and once the robot has initialized,

the red LED goes off and a green LED lights up. The green LED indicates to the user that the robot has successfully initialized and ready to receive control signals from the remote controller. On the other hand, the user firstly has to provide user authentication by the verification of password, which then provides the user the access to fully control the robotic arm remotely via the remote controller. Figure 2 shows the picture of the small scale robotic arm used in this research.

### 3.2. Electronics Design

The robotic arm used in this research has two RB-421 and four RB-796MG servo motors with an operating voltage of 4.8 to 6.0 volts, Pulse Width Control ranging from 500 to 2500  $\mu$ s with 1500  $\mu$ s as the neutral and an stall torque of 4.8 and 10 kg.cm at 6.0 volts respectively. It comprises of three joysticks, each used to control two servo motors of the robotic arm with analog values ranging from 0 to 1024. The joystick have auto spring back and have a neutral position of around 512, being different for each joystick, which has been taken care by the software algorithm. For the RF link, TLP434A and RLP434A four bit data transmitter and receiver in conjunction with HT-12E encoder and HT-12D decoder has been incorporated. For processing, two Arduino mega 2560 microcontroller has been used, with one being for remote controller and the other for the main controller of robotic arm.

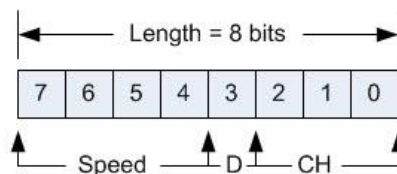
## 4. Data Communication Design

In this section, the features utilized for developing the data communication of the proposed system for controlling the robotic arm are discussed. These major features consist of packets, encapsulation and decapsulation.

### 4.1. Packets

Since the success of introduction of packets for data communication, the concept of packet has become commonly used in data abstraction, mainly in computing science and communication engineering, and is discussed elsewhere [17] hence not introduced in this section.

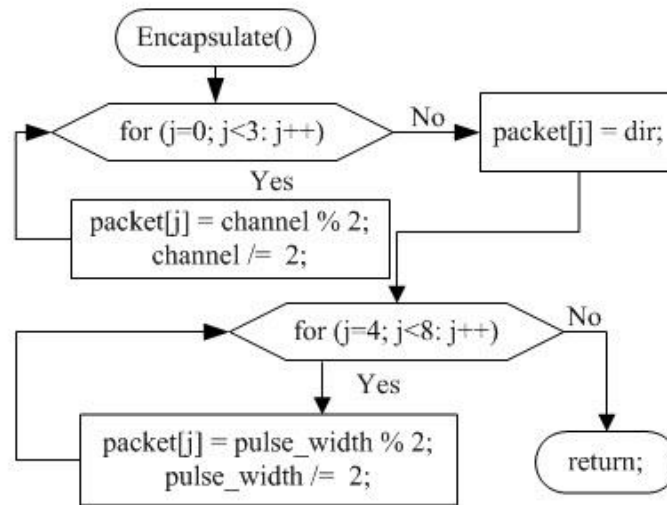
The packet used in the research reported in this article has a length of 8-bits, although limiting the size to a fixed number is not necessary because it is under the control of the systems designer to specify other length of bits, and in general,  $n$ -bits could be used. The 8-bits are divided into two sections, the first 4-bits from bit 0 to bit 3 are used to capture the servo motor direction and number while the second set of four bits from bit 4 to bit 7 are used to capture the servo motor speed/position. The maximum possible speed that can be encapsulated into a packet is 15 ( $2^4-1$ ). Figure 3 shows the packet layout used for the research reported in this article, where CH is the channel number and D is the direction.



**Figure 3. Packet Layout of the RF Communication**

### 4.2. Encapsulation

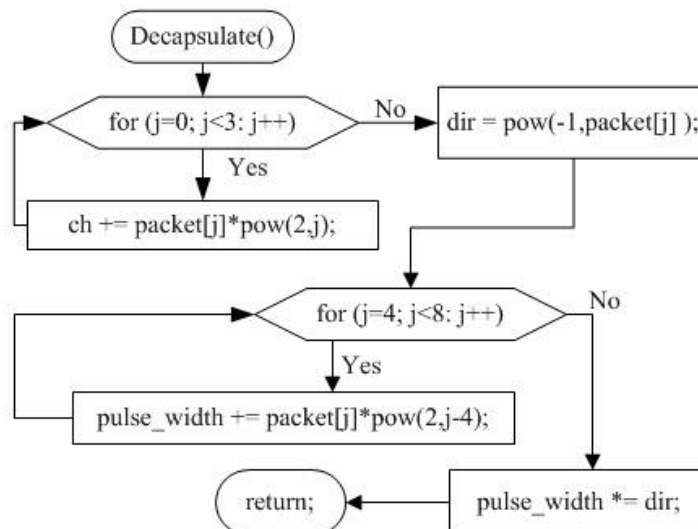
The process of encapsulation (encoding) takes place at the remote controller and involves converting the servo motor direction, number, and speed data into a packet for the purpose of transmission. The method used in realizing encapsulation for motor direction is a simple if else statement whereas for motor number and speed is that of conversion of the decimal number to its binary equivalence that are bundled into the packet as shown in Figure 4.



**Figure 4. Flowchart Showing the Encapsulation Sub-routine**

### 4.3. Decapsulation

Decapsulation (decoding) takes place at the robotic arm controller once the two sets of data comprising the packet are successfully received as shown in Figure 5.



**Figure 5. Flowchart Showing the Decapsulation Sub-routine**

#### 4.4. Method of Transmission from Remote Controller to Robotic Arm Controller

Since the data being transmitted are the entire packet, they need to be first converted into a binary format, and sent from the remote controller to the robotic arm controller via RF link. However, since the four bit RF transmitter and receiver are used, while the packet contains eight bits, two sets of data consisting of four bits each are transmitted. . Figure 6 shows the method of transmission.

There is a sequence to follow for the controller to transmit data successfully. Firstly, the green light on the robotic arm controller should be on thus indicating that it is ready to receive data and the transmit enable (TE) pin has to be HIGH as the transmitter is active LOW in order to transmit data. Secondly, all the respective data is made available at the four pins (data out) and then the TE pin is made LOW. Finally, after a certain delay the TE pin is made HIGH. This process repeats to transmit the second set of data after which the next packet is transferred.

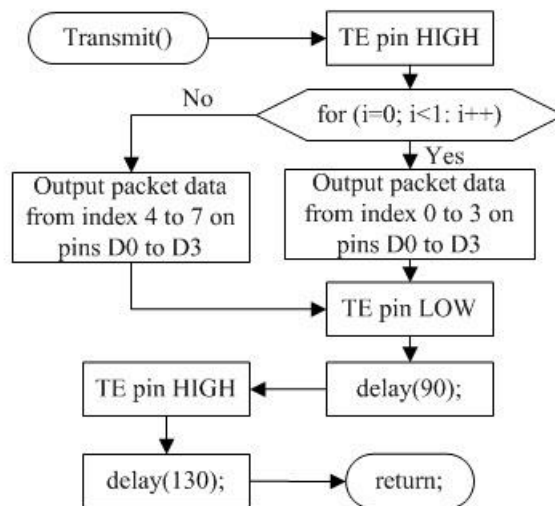


Figure 6. Flowchart Showing the Transmission Sub-routine

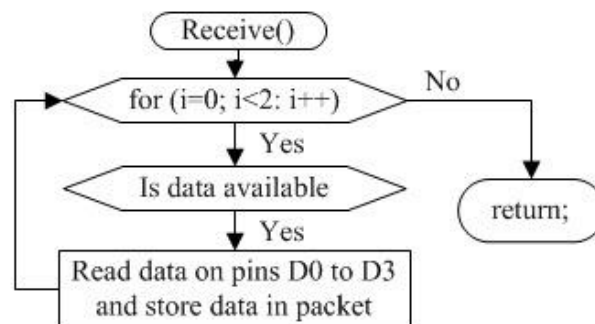


Figure 7. Flowchart Showing the Reception Sub-routine

#### 4.5. Method of Reception at the Robotic arm Controller

The robotic arm controller only receives data if it is in the data receive mode, that is after it has initialized. Once the initialization is complete, it waits to receive data.

Whenever data is transmitted, valid transmission (VT) pin goes HIGH. When VT is HIGH, the controller reads the data at the data input pins and stores this data in the packet. Once it has successfully received the two set of data from the packet, it then goes through the process of decapsulation after which the appropriate signals are sent to the servo controller. The method of reception is shown in figure 7.

#### 4.6. Servo Controller Data Communication Protocol

The servo controller used is the SSC32 servo controller which is capable of controlling 32 servo motors at one time. The TTL serial communication from the robotic arm microcontroller is used to communicate with the servo controller. Having numerous features for controlling the servo motors, only few features have been used for this application. The general command signals that are transmitted from the robotic arm microcontroller is given by equation 1, where <ch> is the channel number in decimal (0 to 31), <pw> being the pulse width in microseconds (500 to 2500) and <time> is the time in ms for the entire move (maximum value of 65535) and affects all channels. The <cr> is the carriage return character, ASCII 13, which is required to initiate the action.

$$\# \text{ <ch> P } \text{ <pw> T } \text{ <time> } \text{ <cr>} \quad (1)$$

In the Arduino microcontroller, the Serial.print() command is used to serially send the control signal to the servo controller.

### 5. Software Design

The software algorithms have been designed in such a way that it enables a standard pattern of operation and unity among the components. The software coding for this application was done in open source Arduino language.

The robotic arm controller starts of by initializing the robotic arm to its home position, indicated by a red LED. Once initialized, a green LED will light up indicating that it is ready to accept control signals from the remote controller and sits idle waiting for the control data. On the remote controller, it will start of by displaying a welcome note and requesting the user to enter a four digit authentication. Once the user enters the authentication password, it is verified by the microcontroller. If an incorrect password has been entered, the user will be prompt to re-enter the password. If a correct password has been entered, the user will be asked to enter the option whether he/she intends to control the robotic arm or change the password. In the change password option, the user will be first prompt to re-enter the old password and then to enter the new four digit password. Upon successfully changing the password, user will again be prompt to select the option.

Once the option for controlling the robotic arm is selected, the user can then control the robotic arm by using the three joysticks. The software algorithm is designed in such a way that no control signal data is transmitted if all joysticks are at their neutral position. Since only four bit transmitter and receiver are used, only one joint of the robotic arm can be controlled at any instance with the priority being given to the base rotation. The priority decreases as it moves to the gripper control. However, priority is only used if attempt to control more than a joint at any instance is made.

The process of encapsulation/decapsulation and method of transmission and reception has been covered in section IV. Equation 2 is used to convert the joystick input to a speed value between 0 to 15; where speed is the analog input value read from



the joystick. After decapsulation has taken place, the motor direction, motor number (channel number) and motor speed data is obtained. The motor speed data is then converted to a new speed value using equation 3. And finally the control signal is sent to the servo controller.

$$\text{speed} = (\text{speed} - 505)/33.5 \quad (2)$$

$$\text{final\_speed} = \text{speed} * 6 \quad (3)$$

The resolution of the servo motor is given by equation 4, where number of pulses is the difference between maximum and minimum value of pulse width.

$$\text{Res} = \frac{1}{\text{num of pulses}} \times 180^\circ = \frac{180^\circ}{2500 - 500} = 0.09^\circ \quad (4)$$

## 6. Results and Discussion

The accuracy test of data transmitted and received was carried out. Software algorithms were incorporated to print out the data being sent and received by the microcontrollers via the RF link. Figure 8 shows that the robotic arm controller has initialized and sits idle waiting to receive control signals while figure 9 shows that the remote controller sits idle when the joysticks are at their neutral position by displaying no inputs, thus saving up unnecessary transmission and power usage. It can also be seen from the results in Figure 8 and 9 that the data sent and received are the same. The system has been tested several times and a 100 percent accuracy of correct data transmitted has been obtained.

```
Initializing to HOME Position
#0 P1600 T2000
#1 P1200 T2000
#2 P600 T2000
#3 P1200 T2000
#5 P1500 T2000
#6 P1600 T2000
Initialized.....
Byte: 1
Byte: 2
Received Data: 10001000
```

**Figure 8. Serial Monitor Screen Showing the System has Initialized and the Received Data after Transmission**

```
*****# No Input #*****
Motor: 1

Joystick Input:1

1000
1000
*****# No Input #*****
```

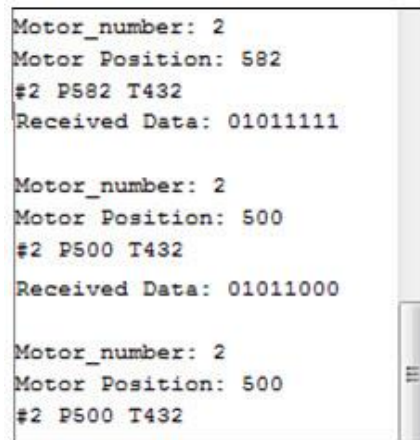
**Figure 9. Serial Monitor Screen Showing the Remote Controller Sitting Idle and the Transmitted Data**

As mentioned in the previous section, two sets of four bit data are transmitted with a time delay of 135 ms between them. Each full packet data is received after every 432 ms, thus being the response time of the system. Several values were tested for this purpose but the above value gives the desired results as decreasing the value results in incorrect data being transmitted.

Attempt was also made to fully utilize the features of servo controller and allow the user to control all the joints at any instance. However, this increased the response time to 1400 ms and the architecture used for the packet needed six four bit data to be transferred. Hence, the proposed data communication protocol has been used.

The system is also designed in such a way that the minimum and maximum pulse width is never exceeded, this is shown in figure 10. This is achieved by the software algorithm thus preventing damage to the servo motor and saving unnecessary wastage of energy. The speed of the servo motor depends on the difference in the pulse width that the servo motor has to move. For each new data received, the system is designed to move the required pulse width in a time interval of 432 ms thus allowing a smooth move of the arm without any jerks. This was possible by the use of the servo controller feature, which provides the option for the time interval for a particular move with its unique command.

Several tests were carried out by several users to pick and place different objects (within the requirements for the gripper) within its workspace and a 100 percent success rate was recorded.



```
Motor_number: 2
Motor Position: 582
#2 P582 T432
Received Data: 01011111

Motor_number: 2
Motor Position: 500
#2 P500 T432
Received Data: 01011000

Motor_number: 2
Motor Position: 500
#2 P500 T432
```

**Figure 10. Serial Monitor Screen Showing that the Pulse width Does not go below the Minimum Value off 500µs**

## 7. Conclusion

A Simple and Efficient Remote Control Protocol for Small Scale Robotic Arm was designed with the Arduino microcontroller as its main processor, thus eliminating the use of a personal computer. The method of packets has been used in the design to transmit data. The design is user friendly and requires authentication to gain access to control the system. It is recommended that the NRF24L01+ RF module be used so that response time can be further reduced by being able to send the whole packet at once rather than in two sets, which has lead to a larger response time. Finally, the proposed method can be used to control other applications that require remote control.

## Acknowledgements

This research work was carried out at the School of Electrical and Electronics Engineering and was fully funded by the research group of Fiji National University.

## References

- [1] S. Prior, "An electric wheelchair mounted robotic arm – A survey of potential users", *Medical Engineering & Technology*, vol. 14, no. 4, (1990).
- [2] M. Hillman and J. Jepson, "Evaluation of a trolley mounted robot – A case study", *Proceedings of ICORR'97*, (1997).
- [3] J. Hammel and V. der L. HFM, "Factors in the prescription & cost effectiveness of robot systems for high level quadriplegics", *Proceedings of RESNA*, (1991).
- [4] H.H. Kwee, J.J. Duimel, J.J. Smits, A.A. T. de M.oed, J.A. van Woerden, L.W. v.d. Kolk and J.C. Rosier, "The MANUS wheelchair-borne manipulator system review and first results", *Proceedings of the 2nd Workshop on Medical & Healthcare Robotics*, (1989).
- [5] M. Hillman, K. Hagan, S. Hagan, J. Jepson and R. Orpwood, "A wheelchair mounted Assistive Robot. *Proceedings of the International Conference on Rehabilitation Robotics*", (1999); Stanford, CA.
- [6] A. Ali, M. Madariaga and D. McGeary. "Assistive robotic arm", *Final Year Project Report*, (2007).
- [7] M. Harshe, C. Menezes, B. Walzade and Prof. L. G. Navale, "An Innovative Multi-DOF robotic arm assembly for press shops", *Final Year BE Project Report*, (2007).
- [8] C. Johnson, T. Kocher, C. O'Donnell, M. Stevens, A. Weaver, and J. Webb, "Robotic Manipulator", *Final Year Project Report*, (2001).
- [9] F. Farahmand, M. Pourazad and Z. Moussavi, "An Intelligent Assistive Robotic Manipulator", *Proceedings of the 27<sup>th</sup> IEEE Annual Conference on Engineering in Medicine and Biology*, (2005); Shanghai, China.
- [10] F. Farahmand, "Development of a Novel Stereo Vision Method and its Application to a Six Degrees of Action Robot Arm as an Assistive Aid Technology", *Master's Thesis*, (2005).
- [11] R. Aufrere, C. Mertz and C.Thorpe, "Multiple sensor fusion for detecting location of curbs, walls, and barriers", *Proceedings of the IEEE Intelligent Vehicles Symposium*, (2003).
- [12] C. Martens, N. Ruchel, O. Lang, O. Ivlev and A. Graser, "A FRIEND for assisting handicapped people", *IEEE Robotics & Automation Magazine*, (2001).
- [13] R. Krishna, G. S. Bala, A. S. C. S Sastry, B. B. P. Sarma and G. S. Alla, "Design and Implementation of a Robotic Arm based on Haptic Technology", *Int. J. of Eng. Research and Applications*, vol. 2,no. 34, (2012)..
- [14] R. Szabo and A. Gontean, "Full 3D Robotic Arm Control with Stereo Cameras Made in LabVIEW", *Proceedings of Federated Conference on Computer Science and Information Systems*, (2013) September 8–11; Kraków, Poland.
- [15] T. Cheng, Z. Sun and P. Zhang, "Imirok: Real-time imitative robotic arm control for home robot applications", *Proceedings of the IEEE International Conference on Pervasive Computing and Communications Workshops*, (2011) March 21-25; Seattle, WA, USA.
- [16] B. House, J. Malkin and J. Bilmes, "The VoiceBot: A Voice Controlled Robot Arm", *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, (2009).
- [17] S. Kumar and R. Singh, "Advanced Speed Control of an AGV", *Proceedings of the IAENG International Conference on Electrical Engineering*, (2011) March 16-18; Kowloon, Hong Kong.

## Authors



**Shiu Kumar** received his Bachelor of Engineering Technology and Postgraduate Diploma in Electrical and Electronics Engineering from the University of the South Pacific in year 2009 and 2012 respectively. Currently he is pursuing degree of Master of Science in Electronics Engineering at Mokpo National University, South Korea. His research interests include Control and Automation, Wireless Sensor Networks, Artificial Intelligence, Embedded Microprocessor Applications and Digital Electronics.



**Seong Ro Lee** received the B.S. degree in electronics engineering from Korea University, Seoul, Korea, in 1987, respectively, and the M.S., and Ph.D. degrees in electrical engineering from Korea Advanced Institute of Science and Technology, Daejeon, Korea, 1990 and 1996, respectively. In September 1997, he joined the Division of Electronics Engineering, Mokpo National University, Jeonnam, Korea. His research interests include digital communication system, mobile and satellite communications system, applications of telematics, USN and embedded system. He serves as chairman of detection and estimation committee for the Korea Information and Communications Society.