WIRELESSLY SYNCHRONIZED ROBOTIC ARM

By

Abdul Moiz Jadoon

Saad Qaisar

M Adeel Chaudhry

Hammad Hassan

Project Supervisor: Lec. Obaidullah Khalid

Submitted to the Department of Electrical Engineering, Military College of Signals,

National University of Sciences and Technology, Rawalpindi.

In partial fulfillment for the requirements of a B.E Degree in

Telecom Engineering

June 2012

ABSTRACT

WIRELESSLY SYNCHRONIZED ROBOTIC ARM

Robots, through the middle Ages and before were used primarily for entertainment. However, from its beginning, the 20th century featured a drastic boom in the development of industrial and commercial robots. This design of a robotic arm replicates casual human arm motion via bend sensors. The system consists of a bend sensor system attached to the human user and a teleoperated robotic arm that mimics the user's casual motions. The robotic arm is designed to be lifelike but simplified. User motions are sensed by bend sensors, processed by a central microcontroller, and replicated by servomotors on the robotic arm. All the industrial manufacturing has got some sort of robotic aid to increase its production & improve quality, space satellites are in need of human like robots that could help exploration in space stations. Medical field is also becoming dependent on equipments that could aid in performing remote surgery.

The wireless units are tested for transmission and reception of digital signals. The wireless system will then be connected to microcontroller and compatibility and functionality between them will be checked. The accuracy and resolution of the motion data collected by the sensors will be tested, as well as the compatibility of the sensors with the microcontroller. Throughout all of the testing and analysis, C programming will be written, tested, debugged, and retested.

DECLARATION

We declare that no portion of the work presented in this dissertation has been submitted in support of another award or qualification either at this institution or elsewhere.

بسم الله الرحمن الرحيم

DEDICATION

To our Parents and Teachers

ACKNOWLEDGEMENTS

First of all, all our praise goes to Allah Almighty, who made us able to complete the project with efficiency and within the specified time.

Special thanks to our supervisor, Lec Obaidullah Khalid, Faculty member of Electrical Engineering, Military College of Signals, National University of Sciences and Technology, for all of his efforts towards completion of our goals throughout the year.

Great deal of appreciation goes to Mr. Shahid head of Science World, Rawalpindi, for helping us in many problems that occur during the project. His help had been tremendous in solving many of our problems.

We would like to thank Mr. Ramzan, from Nowshera for supporting us during the manufacturing of the mechanical structure of the project. We would also like to thanks Mr. Sada Muhammad and Mr. Shoukat for this purpose. Also Mr. Saleem from Nowshera has been an invaluable asset in assisting us to solve many problems related to electronics.

Lab Engineer Intisar Rizwan MCS, NUST, has been a source of invaluable support and has always helped us and enlightened us with his creative and educated ideas.

We also wish to extend our gratitude to Ms. Mahnum Maqsood, Product Manager DigiMac Hi-Technologies, for her enormous help without which the documentation process of our project could not have been completed.

Table of Contents

LIST OF FIGURES

FIGURE NAME **PAGE NO**

Chapter 1

Introduction

1.1 Why are robots important?

Robots, through the middle Ages and before were used primarily for entertainment[8][9]. However, from its beginning, the 20th century featured a drastic boom in the development of industrial and commercial robots. Throughout the rest of the 20th century, robots altered the structure of society and permitted for safer conditions for not only labor but also common man. In addition to this, the discharge of advanced robotics in the military and, as an example, NASA has changed the scenario of national defense and outer space exploration. Robots have also been influential in the media, profitable for toy manufacturers and also provide a safe tool for humans.

Industries have benefited drastically from the stretch of a robotic work force. Automated machines have taken over the duties of risky and everyday jobs from humans, allowing superior efficiency. Because robots don’t have any feelings, they are never tired and, as a consequence, extra shifts have been supplemented to factories. Farmers have also taken advantage of new knowledge with automated harvesters; the municipal industry has also implemented robots in its dirtier jobs, and the medical industry benefits, the most, from the advancements in assisted surgical robotics. The idea of a factory with no humans has come to execution. A multinational company, IBM, runs a "LIGHTS OFF" factory in Texas. It is completely staffed by fully autonomous robots making computers. The military has also initiated and implemented various programs in robotic technology, most deadly and successfully the Reaper unmanned aerial reconnaissance and attack vehicle that allow an operator (pilot) to control the drone from vast and safe distances. The vehicle can perform high-altitude surveillance and ground attack for long periods without having a support of a live pilot, and when needed, the planes can launch precision strikes on targets in hostile zones.

The accuracy and precision of robots can also be increased if the robot has a physical shape of that of a human body part, say a hand or a leg. If such a robot is made operational, it can perform multiple tasks as compared to conventional robots.

Medicine has employed robots and has been successful more than any other discipline. It has resulted in saving the lives of millions of people and continues to do so. Since robots are more precise and accurate, their preference over human hands is ever increasing.

Today, robots are the sole reason for every type of luxury that we enjoy. From luxury cars to ATM machines, all provide us the easiness in life. This method is useful in a perspective as it will provide a very easy interface as shown in figure 1.1. Similarly figure 1.2 shows the flow chart of the system.

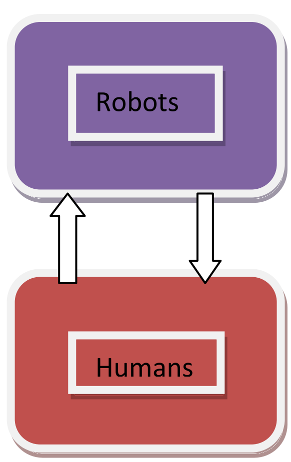

Figure1.1 Interphase

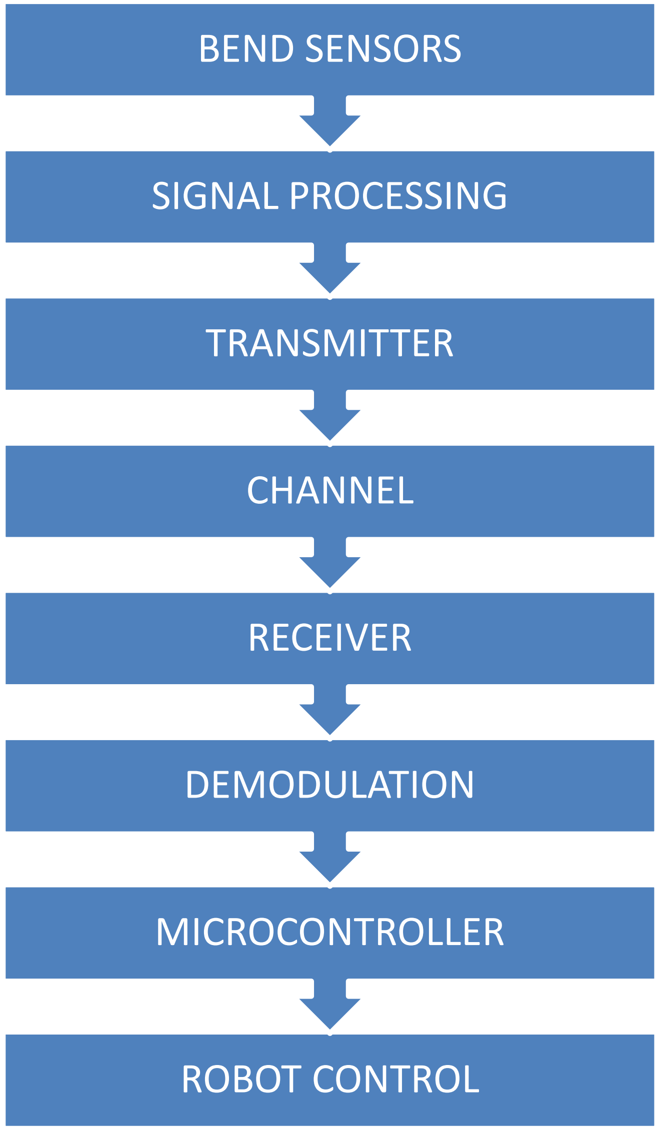

Figure 1.2 Flow Chart Of Wsra

1.2. Organization of the Document

Chapter1 tells us about the meaning of robots and their importance.

Chapter 2 deals with the conventional and the current modern robots and robotic arms. This chapter will explain the advantages and disadvantages of each of these robots and establish the background and need for our proposed project.

Chapter 3 defines and explains our project and also explains the benefits and advantages of our system. It also enlightens about the perfect environment needed for the project execution.

Chapter 4 explains about the hardware and the software needed to run the project. This chapter also explains the controlling procedure of the vehicle using dc motors and servo motor in VCM.

Chapter 5 deals with setting up the hardware and software necessities of the project. It elaborates how to set up our hardware part of the robotic arm.

Chapter 6 deals with explaining the setting up of the equipment in the classroom and the coverage of the classroom by the cameras using the two cameras.

Chapter 7 deals with the literature review of the three algorithms for face detection and recognition. Viola Jones, Efficient and Rank Deficient, Skin Color Segmentation are the face detection algorithms. The best one is chosen on the basis of the results shown. Principal Component Analysis-PCA, Independent 3 Component Analysis (ICA), Linear Discriminant Analysis (LDA) are the recognition algorithms.

Chapter 8 explains future enhancements are discussed and the report is concluded. This chapter also explains the integration of the WSRA with other technologies.

Chapter 9 explains the complete working of the robotic arm and it also covers how to operate the robotic arm in different modes.

Chapter 2

Current Systems

There are many ways to control robots, both wired and wirelessly .Many media are used to control the robots. These media may include Wi-Fi, copper, Lan or optical cable. All of these traditional systems have their own advantages and disadvantages. The use of one format in one situation is better than the use of the other format. Few of these format are listed with their advantages and disadvantages.

2.1. Remote Control Robots

This technology includes the control of robot by using remote or switches. The user presses the button to make the robot to perform certain actions. The medium used is Wi-Fi or RF. The robot performs the specific task depending upon the button being pressed. The range of the operation is directly proportion to the type of medium that have been used for communication. If RF is used the range will be high. If Wi-Fi is used the range will be little less as shown in figure 1.3 below.

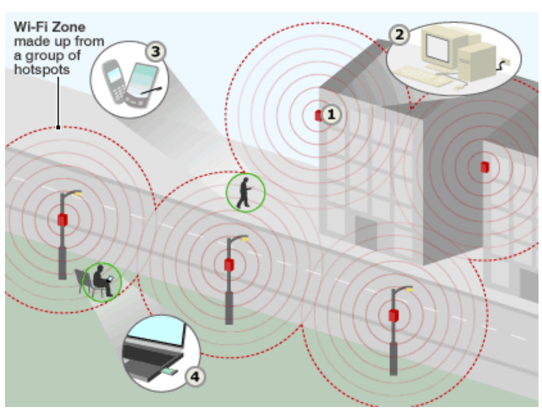

Figure 1.3 Showing Less Range Of Wi-Fi

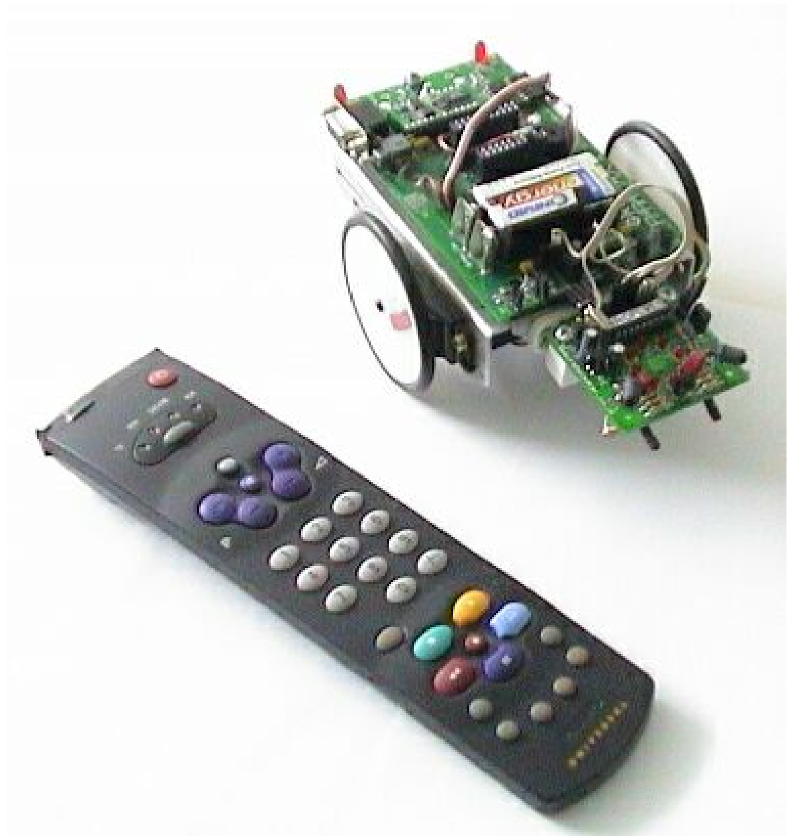

Figure 2.1 Remote Control Robots

2.1.1. Advantages

- Since the medium used is wireless therefore there are no portability issues. If RF is used as medium then the robot can be operated from other rooms as well. The use of the remote control is easy and is a very mature technique. It is also very economical.

2.1.2. Disadvantages

The functioning of the robot is very difficult through the use of buttons. In many cases combination of movements are needed to perform to make the robot to do a specific task. Wi-Fi will reduce the range of operation (see figure 1.3) and also wireless mediums have their own delays associated with it. The interference in the environment might also affect the quality of transmission since signals from other devices such as mobile phones etc working in the same or nearby frequency range may cause disturbance.

2.2. Wired Control Robots

Most of the robots used in industries are wired. They may involve computers as well and are used during the manufacturing process. The operation can be controlled by a human or can also be automatic.

Figure 2.2 Wired Control Robots

2.2.1. Advantages

These are fast and efficient to use and are used in most of the industries. They are very powerful robots are being operated by people especially in industry. An economical control of these robots makes them a perfect choice of the operators.

2.2.2. Disadvantages

Portability is the major issue with such robots. Such robots are difficult to move. The range of communication also depends upon the length of the wire of the antenna. Shorter the wire, shorter will be the range. The type of wire also determines the performance of the robot and affects its operation.

2.3. Robots Controlled Through Web Cam Based Motion

These robots will copy the action performed by the human. In this case a video camera is used. The image is taken by a camera and then processed through a code. According to various actions done by the person, the signals are transmitted to the robot that will replicate the actions.

Figure 2.3 Gesture Detection

2.3.1. Advantages

The operator is free to move.as such robots have a good (user friendly) interface. They are easily operated as no technical skills are required to operate them. Any medium can be used including Wi-Fi etc can be used to operate them.

2.3.2. Disadvantages

Portability is the major issue, again, with this type of robot. Many machines and computers are involved. Due to which certain actions are very difficult to perform Processing delays are also involved in such robots.

2.4. Autonomous Robots

Word autonomous is used to depict intelligence [10]. Autonomous robots are the one where there is no manual control by the humans. The action the robots perform, go on repeating.

2.4.1. Advantages

In such robots no human operator is required. These robots will make decisions on their own as they are programmed to do so. Safety of human life is ensured as these robots do all the risky work themselves.

2.4.2. Disadvantages

These robots are programmed just for specific purpose and will fail in unknown or new environment. They repeat the programmed task again and again and have to be reprogrammed to perform further tasks. Such robots require very expensive equipment. Processing delays are involved in such robots. Recharging of the batteries required extensively since they are frequently drained out. Their hardware is also complex.

Chapter 3

Proposed Project and Environment Selection

3.1. Project scope

Our proposed system automates the robotic movement. Our first step would be to acquire a robotic arm. Our next step is to control that robotic arm using servo motor control. We will then synchronize the movement of the robotic arm on wired medium. After that we will establish a wireless connection and synchronize the robotic arm movement with the human arm.

3.1.1. WHAT DOES EACH BLOCK DO

Bend Sensors

As shown in figure 3.1, these sensors detect the motion of human as when a pressure is applied on them. They are attached to the body of a person like a tight cloth suiting in order to detect even the slightest of the movement. Their resistance vary with the bend or pressure applied on them. Thus a voltage level corresponding to the amount of bend can be produced. Video signals are also being received at the receiver end to make sure correct operation performed.

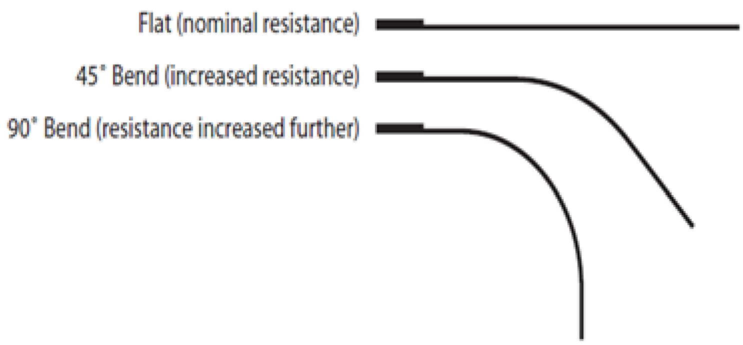

Figure 3.1 Bend Sensor

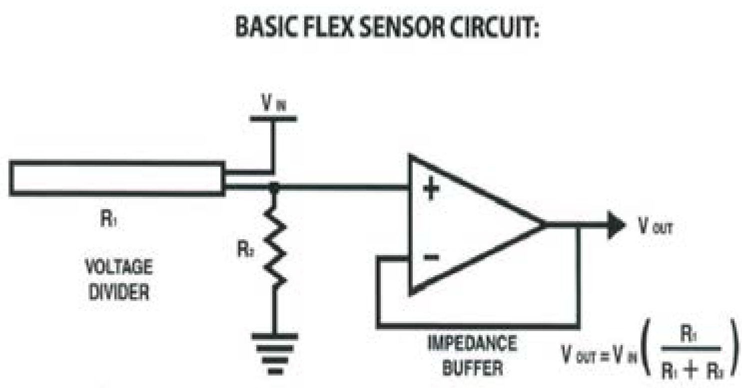

Figure 3.2 Basic Flex Sensor Circuit

Transmitter and Receiver

Transmitter transmits these signals through the channel via RF or any frequency. To enhance transmission power antennas can also be used. The transmitter for our project is shown in figure 3.3 below. It clearly shows the glove with sensors fitted in it. Receiver receives the signal transmitted by the sender. The channel used is RF.

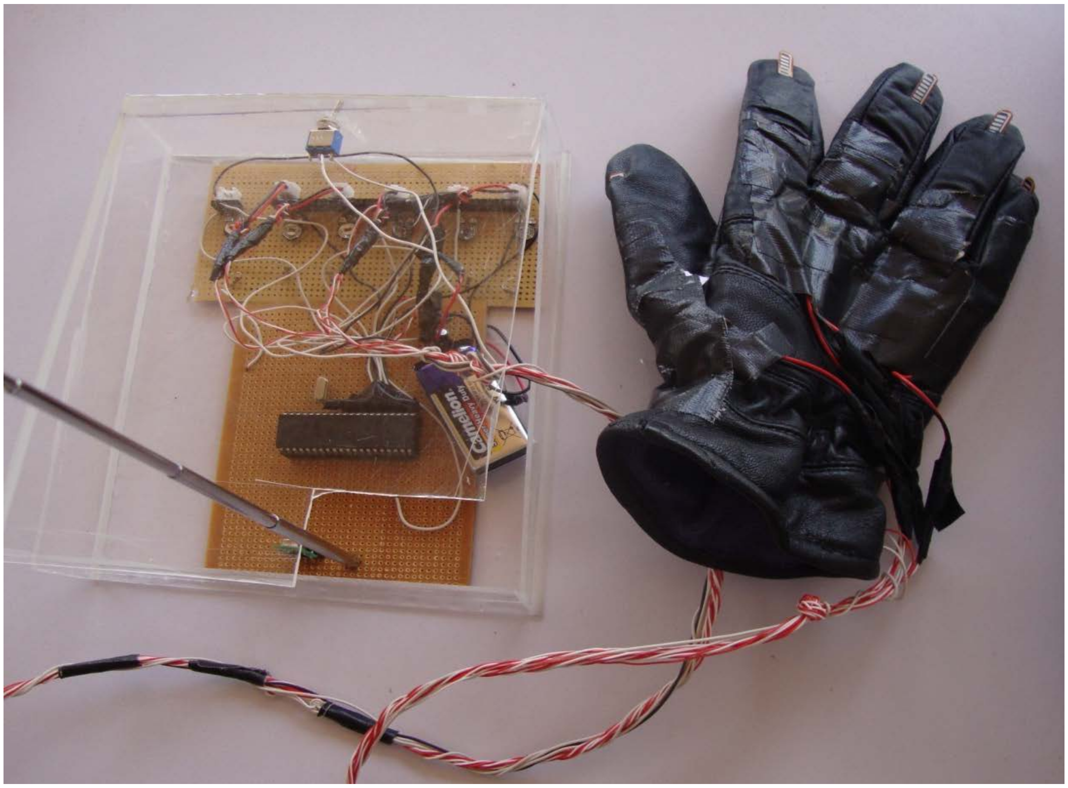

Figure 3.3 Transmitter With Glove

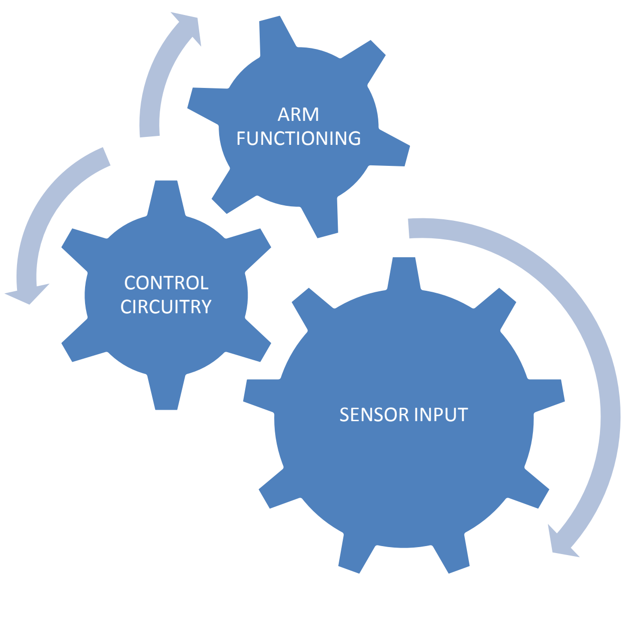

Figure 3.4 Broadened Perspectives

3.2. Advantages of This System

This project has a wide scope in industrial manufacturing as well as in space satellites. All the industrial manufacturing has got some sort of robotic aid to increase its production & improve quality, space satellites are in need of human like robots that could help exploration in space stations. Medical field is also becoming dependent on equipment that could aid in performing remote surgeries.

3.2.1. Why to Develop Such a Robot

The purpose to work on such a robot is to develop a machine that could do maintenance or repair work where conditions for human are not suitable, like space rockets launched for years, or high temperature places like industries or factories etc.

Aid for the Disabled

WSRA can be used as an aid for the disabled. As WSRA is a synchronous machine and is only maneuvered by moving the human arm on which bands are placed; it is well suited for the disabled as they can lift objects using this machine.

Military Operations

This technology has also found its way in military operations. Diffusing bombs etc could use the applications of WSRA where the operator can stand at a safe distance and diffuse the bomb. This can also be used to disarm a miscreant while the law enforcement personnel remain at a safe distance.

Remote Surgeries

Today medical field is much dependent on electronics & biotechnology, with the help of robotic arm surgeons will be able to do surgeries even being at home or in case of emergency operation. Medicine has leaped forward a thousand miles by the use of such robots.

Space Exploration

Much of the research being carried in space is done with the help of robots that could replace humans so that work is carried out more efficiently & also the commodities & living needs of human are avoided.

3.3. Environment selection

To accomplish this task we needed an E.M. jammer free environment. This is because em wave jammer can interfere and jam the radio waves that carry the control information. The environment in which we had been working has been that of Electronics lab-1. It provided us with a perfect environment since it had all the necessary equipment that we required. However since our wirelessly synchronized robotic arm is suppose to work in rough environments therefore we moved out of the lab and selected the area between the back side of EE dept and the computer repair shop of MCS. The space available allowed us to do the work not only through the robotic hand but also allowed us to freely test our vehicle which gives it mobility.

For the code to be efficient for any condition, the environment should be as close to the actual working environment as possible. Therefore, the robotic arm was finally moved to the required place which accurately comes up to our requirement. The ground surface conditions (which should not be rough), weather etc were all meeting our conditions.

Chapter 4

Hardware and Software Specifications

4.1. Hardware

4.1.1. Metal Selection

Working on a robotic arm requires that the material chosen to make the arm should be strong as well as light. For this purpose we chose steel sheets which were not only light and rust resistant but also strong.

With such qualities our robotic arm is capable of lifting and working with heavy and heated objects without the fear of a breakage.

Brass with steel joints

Initially we made our robotic arm using the above materials. It was completed but we faced a very problematic dilemma. The combination of these two produced chemical reactions at the atomic levels causing the robotic arm to rust especially at the joints where the pivots were fixed.

Complete steel structure

This was a wise step since now the whole body and the pivots on the joints were to be of the same material i.e. steel. This avoided rusting and problems due to friction. It also brought complete uniformity and the visual appearance of the robotic arm was also improved. Therefore we decided to go with this structure and had to abandon our initial structure with brass joints.

4.2. Car Controlling

For car controlling at the receiver end the hardware include relay driver circuit. In software part the packets are sent by the transmitter in VCM.

For VHM see Glossary.

4.2.1. Hardware

For the motor control as it is known that microcontroller cannot provide the required current to rotate the dc motors so some sort of driver is required. Therefore we used relay driver circuit which switches the relay on the logic from the microcontroller as shown In the figure below.

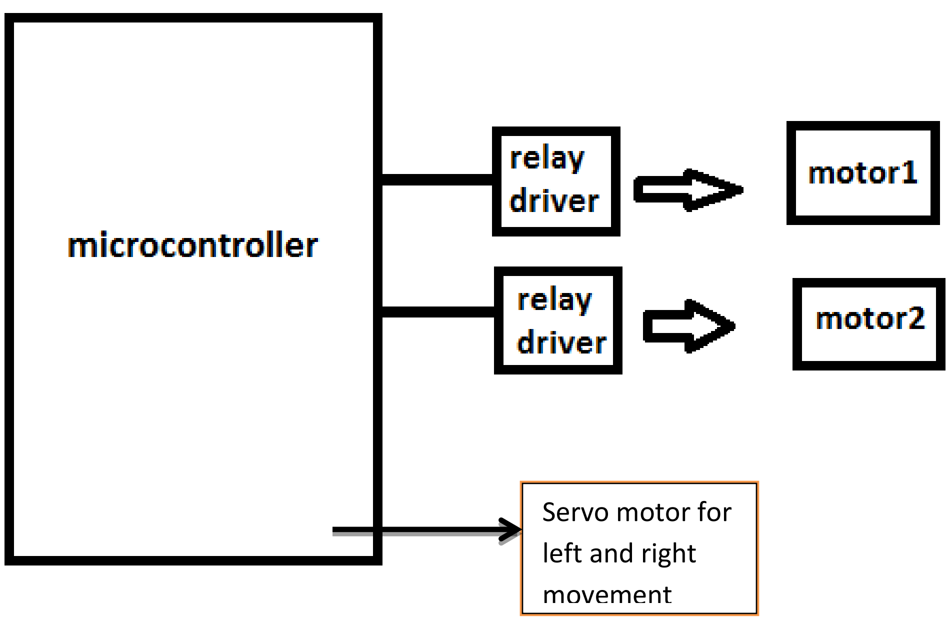

Figure 4.1 Car Controlling With Relay Driver

4.2.2. Relay driver

The relay driver circuit diagram is shown below. When the microcontroller gives a logic 1 ti the relay driver circuit the relay is switched. The power goes to the motor so it can be controlled in any direction.

Note at the top the power supplied will be according to the power required by the relay to be operated. When logic 1 comes from microcontroller the relay will switched to its second stage where it will get the power supply to operate the dc. Note we have used two power supplies, one for the relay and the other for dc motor control.

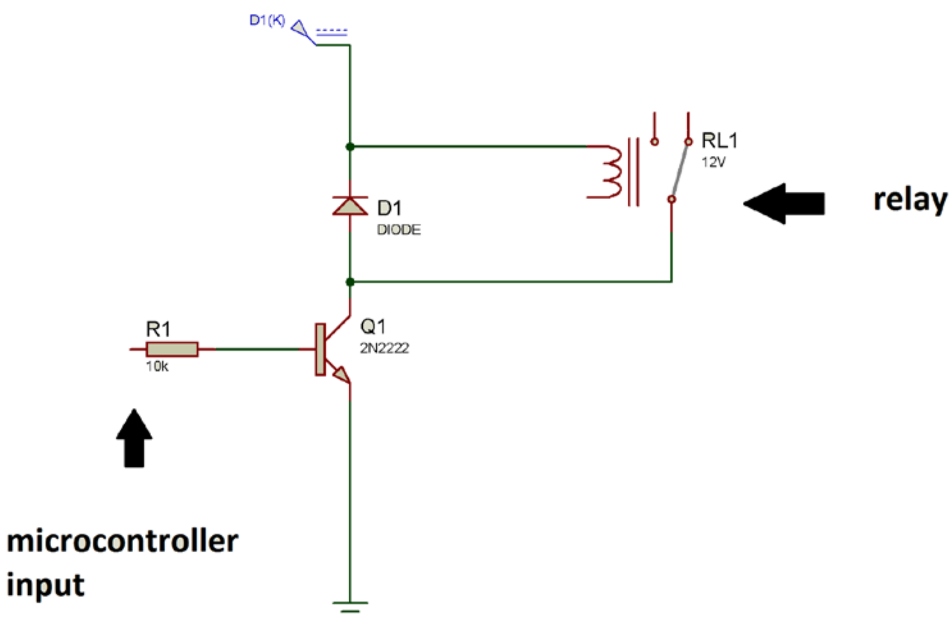

Figure 4.2 Relay Driver Circuit

4.3. Software

Mikro C is used for programming and making GUI of the system. It caters for processing speed. It has tools for programming and debugging C++ code. It provides third party software compatibility.

Proteus is used by for running various simulations of our circuit designs. It has pin pointed various errors in our design and has led us to our final circuitry which has been almost free of errors now.

Win Pic. This software has been extensively used by us to burn the microcontroller. It not only burns the microcontroller but also indicates whether there is an error or not. It can be easily installed on Microsoft Windows XP.

Operating system used is Microsoft Windows 7 and XP.

A complete flow chart is shown in figure 4.3 below.

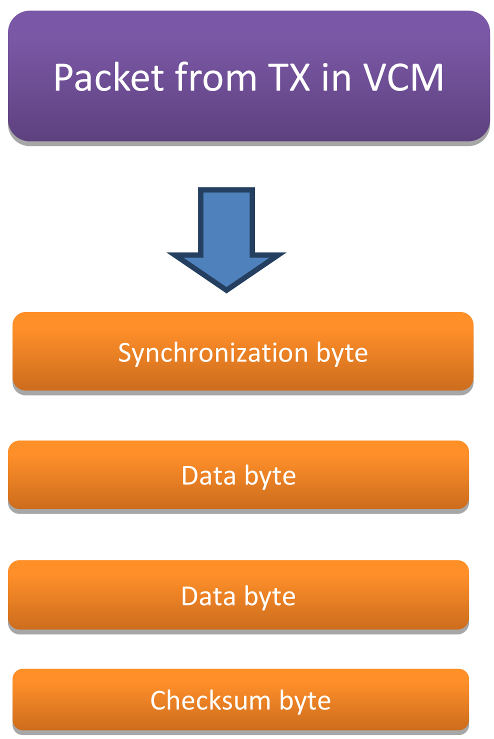

FIGURE 4.3 Flow Chart (Software Part)

The packet received at the receiver end is decoded. In accordance with the finger movement the respective signals are transmitted by the microcontroller output pin defined. The code given below is only for car controlled part at receiver end. The complete code is also given in appendix A-2. The following code will give an idea about how the microcontroller responds for different signals received at the receiver end

```
////////// DC forward //////////

if(portb.f2==0 && portb.f3==0 && portb.f4==0 && portb.f5==0)

        {

          portc.f1=0;

          portc.f2=0;

        }

if(portb.f2==1 && portb.f3==0)

        {

          portc.f1=0;

          portc.f2=1;

         }
```

For complete code see Appendix A-2.

How the microcontroller works relative to an input is shown below in table 4.1.

Table 4.1 Dc Motor Control

microcontroller output-1	microcontroller output-2	dc motor response
0	0	stop
0	1	one direction rotation
1	0	other direction rotation
1	1	stop

Chapter 5

Setting up the Equipment

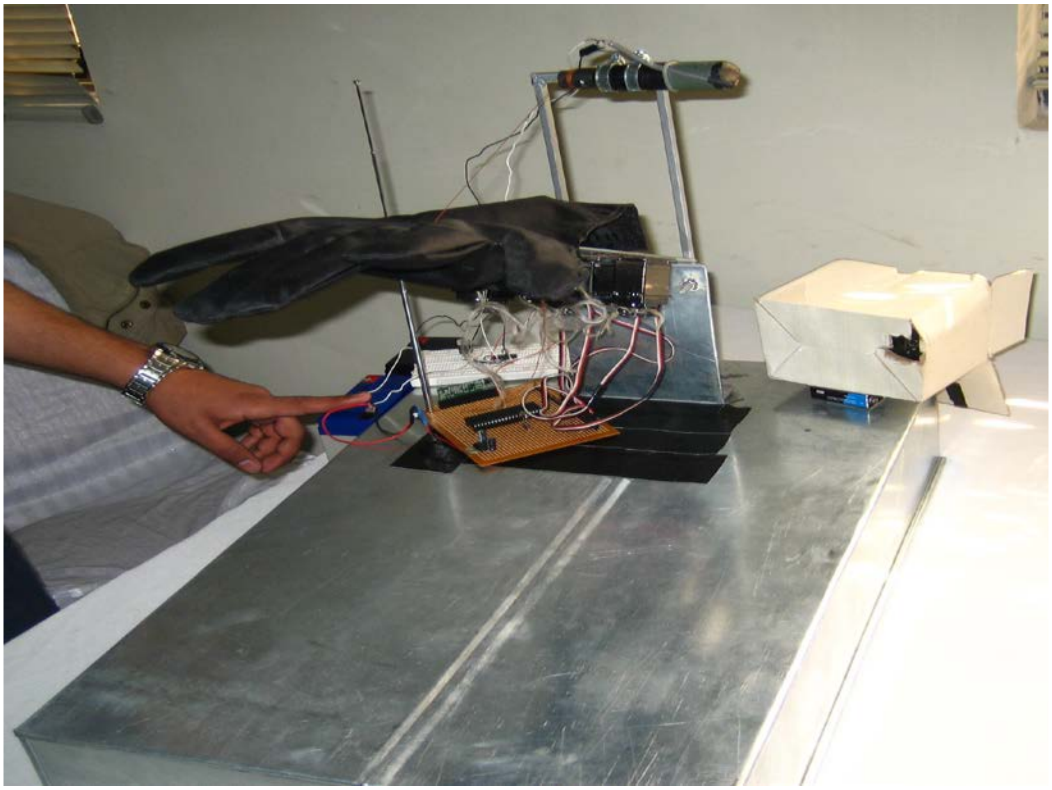

Figure 5.1 The Setup

The vehicle beneath the robotic arm is a complete separate body which can be attached or detached expediently. Though the control for the vehicle is the same as that of the robotic arm, yet it is a separate entity.

The vehicle is composed of chassis part, the driving motor with attached wheels. There are four metallic supports that are attached to the vehicle. These four structures are supposed to carry the weight of the robotic arm on itself. There are four holes in the robotic arm base. The four supports get fit in the base of the robotic arm.

The batteries are confined in a box. They are connected to the D.C motor of the vehicle as well as the servo motors. The servo motors are permanently attached to the robotic arm. There is an ON and OFF switch which can turn the system on or off. Power to the DC motor of the elbow shall be provided by the same power box.

There is a separate power source required for the transmitter. A 9V battery can serve that purpose.

A high strength string is attached to the elbow motor to lift up the robotic arm. A number of different equipments were used as shown in table 5.1.

Table 5.1 Number and Quantity of Equipment Used

Equipment Used	**Quantity**
Dc Motors For Arm	1
Servo Motors For Fingers	5
Servo Motor For Vehicle	1
Dc Motor For Vehicle	1
Batteries At Receiver	2
Microcontroller At TX	1
Microcontroller At RX	1
Transceiver	1
Antennas	2
Batteries At Transmitter	1

Chapter 6

Literature Review

6.1. Servo Motor

RC servos are hobbyist remote control devices servos typically employed in radio-controlled models, where they are used to provide actuation for various mechanical systems such as the steering of a car, the control surfaces on a plane, or the rudder of a boat. [1]

Due to their affordability, reliability, and simplicity of control by microprocessors, RC servos are often used in small-scale robotics applications.

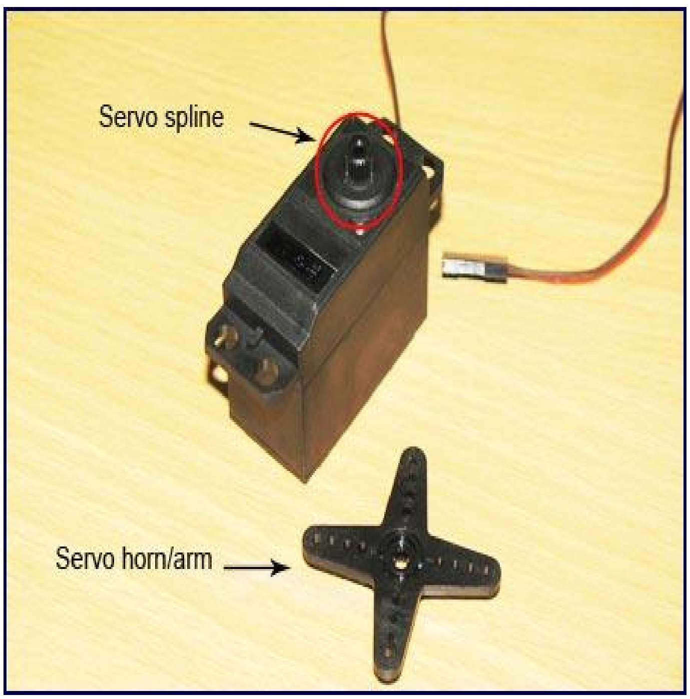

Figure 6.1 A Typical Servo Motor

RC servos are composed of an electric motor mechanically linked to a potentiometer. A standard RC receiver sends pulse-width modulation (PWM) signals to the servo. The electronics inside the servo translate the width of the pulse into a position. When the servo is signaled to rotate, the motor is powered until the potentiometer reaches the value corresponding to the signaled position.

[2] RC servos use a three-pin 0.1" spacing jack (female) which mates to standard 0.025" square pins. The most common order is signal, +voltage, ground. The standard voltage is 4.8 V DC, however 6 V and 12 V has also been seen for a few servos. The control signal is a digital PWM signal with a 50 Hz frame rate. Within each 20 ms timeframe, an active-high digital pulse controls the position. The pulse nominally ranges from 1.0 ms to 2.0 ms with 1.5 ms always being center of range. Pulse widths outside this range can be used for "over travel" -moving the servo beyond its normal range. This PWM signal is sometimes (incorrectly) called Pulse Position Modulation (PPM).

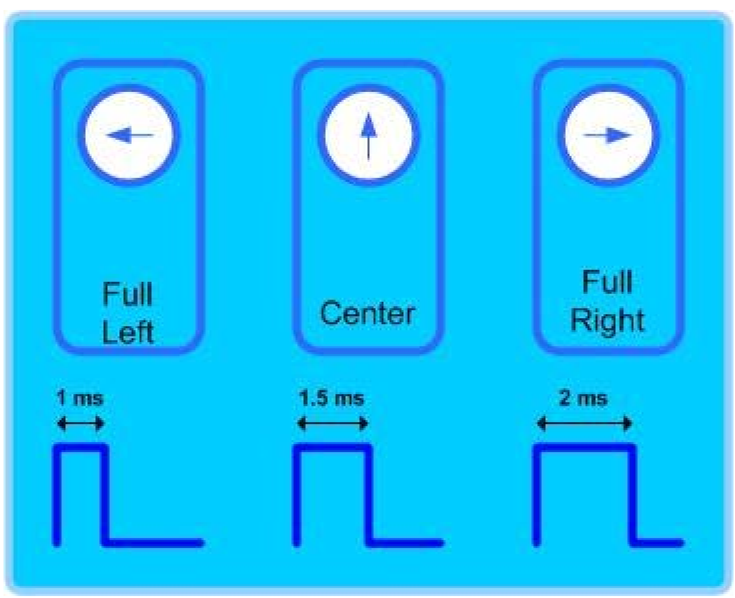

Figure 6.2 Servo Positions With Respect To Input Pulse Cycle.

The servo is controlled by three wires: ground, power, and control. The servo will move based on the pulses sent over the control wire, which set the angle of the actuator arm. [3] The servo expects a pulse every 20 ms in order to gain correct information about the alignment angle. The width of the servo pulse dictates the range of the servo's angular motion.

A servo pulse of 1.5 ms width will typically set the servo to its "neutral" position or 45°, a pulse of 1.25 ms could set it to 0° and a pulse of 1.75 ms to 90°. The physical limits and timings of the servo hardware varies between brands and models, but a general servo's angular motion will travel somewhere in the range of 90° - 120° and the neutral position is almost always at 1.5 ms. This is the "standard pulse servo mode" used by all hobby analog servos.

6.2. Flex Sensors

A flex sensor is physically a thin strip which is flexible in nature; it is able to change the resistance depending upon the amount of bend produced on the sensor [4]. The resistance is directly proportional to the deflection caused by bending the sensor. Flex sensors are analog resistors and are made of carbon material and a flexible substrate. Carbon is present on the plastic strip in the form of discontinuous patches, which when bent is brought closer to each other thereby increasing the overall resistance.

Figure 6.3 Typical Flex Sensor

More the carbon less the resistance [5]. When the sensor is bent, it produces a resistance value which is relative to the bend radius. The range of sensor resistivity is between 9kΩ and 35 kΩ.

The flex sensors give the corresponding resistance based on the bend. This resistance is converted into voltage using voltage dividers. A voltage divider uses two resistors in series to divide the input voltage by the ratio of the resistances. To measure the voltage corresponding to the resistance, we need to keep one resistance value constant and the other as variable resistance. This variable resistance is obtained by connecting the flex sensor.

6.3. RF Modules

A radio wave is an electromagnetic wave propagated by an antenna. Radio waves have different frequencies, and by tuning a radio receiver to a specific frequency you can pick up a specific signal. An RF Module is a (usually) small electronic circuit used to transmit, receive, or transceiver radio waves on one of a number of carrier frequencies.

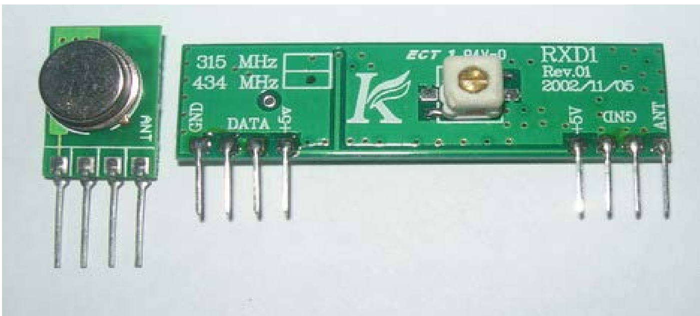

Figure 6.4 RF Modules

Several carrier frequencies are commonly used in commercially-available RF modules, including 433.92MHz, 315MHz, 868MHz and 915MHz.

When attaching an external antenna to an RF Module, superior performance can be achieved by selecting an antenna length related to the wavelength of the carrier frequency. For a 315MHz Module, use a 24 cm antenna length, while for a 433.92 MHz, use 18 cm antenna.

6.3.1. Main factors affecting RF module performance

As with any other radio-frequency device, the performance of an RF Module will depend on a number of factors [6]. For example, by increasing the transmitter power, a larger communication distance will be achieved. However, this will also result in a higher electrical power drain on the transmitter device, which will cause shorter operating life for battery powered devices. Also, using a higher transmit power will make the system more prone to interference with other RF devices, and may in fact possibly cause the device to become illegal depending on the jurisdiction.

Correspondingly, increasing the receiver sensitivity will also increase the effective communication range, but will also potentially cause malfunction due to interference with other RF devices.

The performance of the overall system may be improved by using matched antennas at each end of the communication link, such as those described earlier.

Finally, the labeled remote distance of any particular system is normally measured in an open-air line of sight configuration without any interference, but often there will be obstacles such as walls, floors to absorb the radio wave signals, so the effective operational distance will in most practical instances be less than specified.

6.3.2. Typical applications

Typical applications include vehicle monitoring, remote control, telemetry, small-range wireless network, wireless meter reading, access control systems, wireless home security systems, area paging, industrial data acquisition system, radio tags reading, RF contactless smart cards, wireless data terminals, wireless fire protection systems, biological signal acquisition, hydrological and meteorological monitoring, robot remote control, wireless data transmissions, digital video/audio transmission, digital home automation, such as remote light/switch, Industrial remote control and telemetry and remote sensing.

Alarm systems and wireless transmission for various types of low-rate digital signals is also an important application.

6.3.3. How is range determined?

In order to accurately compute range – it is essential to understand a few terms:

dB - Decibels

Decibels are logarithmic units that are often used to represent RF power. To convert from watts to dB: Power in dB = 10* (log x) where x is the power in watts.

Another unit of measure that is encountered often is dBm (dB milliwatts). The conversion formula for it is Power in dBm = 10* (log x) where x is the power in milliwatts.

Line-of-site (LOS)

Line-of-site when speaking of RF means more than just being able to see the receiving antenna from the transmitting antenna. In, order to have true line-of-site no objects (including trees, houses or the ground) can be in the Fresnel zone. The Fresnel zone is the area around the visual line-of-sight that radio waves spread out into after they leave the antenna. This area must be clear or else signal strength will weaken. There are essentially two parameters to look at when trying to determine range.

Transmit Power and Receiver sensitivity

Transmit power refers to the amount of RF power that comes out of the antenna port of the radio. Transmit power is usually measured in Watts, milliwatts or dBm.

Receiver sensitivity refers to the minimum level signal the radio can demodulate. It is convenient to use an example with sound waves; Transmit power is how loud someone is yelling and receive sensitivity would be how soft a voice someone can hear. Transmit power and receiver sensitivity together constitute what is known as “link budget”. The link budget is the total amount of signal attenuation you can have between the transmitter and receiver and still have communication occur.

Example:

TX Power: 20dBm.

RX Sensitivity: -110dBm.

Total Link budget: 130dBm.

For line-of-site situations, a mathematical formula can be used to figure out the approximate range for a given link budget. For non line-of-site applications range calculations are more complex because of the various ways the signal can be attenuated.

RF communications and data rate

Data rates are usually dictated by the system - how much data must be transferred and how often does the transfer need to take place.

Lower data rates, allow the radio module to have better receive sensitivity and thus more range. In the XStream modules the 9600 baud module has 3dB more sensitivity than the 19200 baud module.

This means about 30% more distance in line-of-sight conditions. Higher data rates allow the communication to take place in less time, potentially using less power to transmit.

6.4. Microcontrollers

Microcontrollers are a common electronic building block used for many solutions and are needed throughout the industry, commerce and everyday life.

They are found inside aircraft instruments. They are used extensively within cellular phones, modern cars, and domestic appliances such as stereos and washing machines and in automated processes throughout industry.

A microcontroller is very much everything that you would find inside a PC's case, but on a smaller scale. There is a processor, temporary memory for data (the RAM) and memory for programs (the ROM).

The microcontroller is capable of carrying out millions of instructions every second. And there are billions of these controllers out there in the world doing just that.

6.4.1. What exactly is a Microcontroller?

As with any electronic circuit the microcontroller circuit has three parts, the INPUT, PROCESS AND CONTROL. The input circuitry converts the real world into the electronic; the microcontroller processes the electronic signals; the output circuitry converts the electronic into the real world.

Inside the microcontroller there is however another level of conversion. The micro has input code, output code and instructions (process code), as well as variables to store data. The input code converts the electronic signals to data (numbers).

The process code manipulates the data. The output code converts the data (numbers) to electronic signals. Variables are locations in memory that data is stored in.

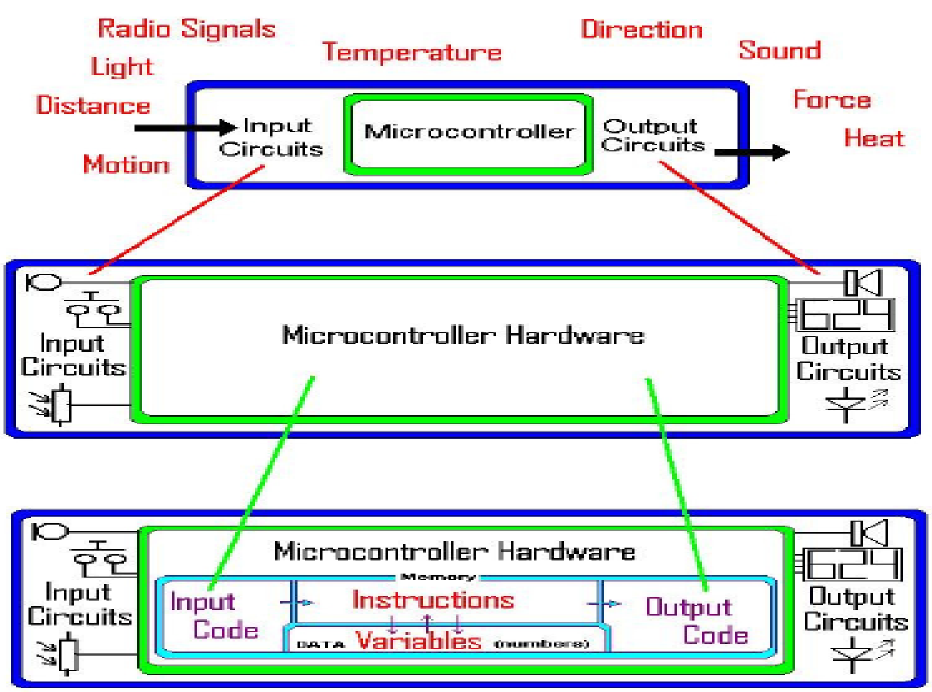

Figure 6.5 Working of Microcontroller

6.5. Code Explanation

6.5.1. Steps of coding

First of all get analog input from sensors and pass them to microcontroller. The microcontroller converts analog to digital using ADC mode of microcontroller. It then checks which value is received and transmit bit according to that received level. At receiver end it checks received bits and according to the received bits, it rotates servo motors to defined angles.

Microcontroller has eight channels of ADC. Using six channels of controller we convert six sensors input to digital, five for fingers and one for up and down motion sensors. The Controller converts analog input to ten bits digital number. By converting these digital bits to decimal values, the controller decides which bit pattern will be sent. Each analog input level has different decimal value.

So controller differentiates different analog levels using their decimal value. One decimal value of one sensor is then given two bits which mean each sensor has two bits to represent its level. Using two bits, four different levels of sensors can be transmitted. Same procedure is repeated for each sensor.

Two bits from each sensor form twelve bits. Controller adds two bits to six bits of three sensors to differentiate this byte with other. Controller adds these two bytes to create checksum byte. Then controller transmits four bytes including one synchronization byte, two data bytes and one checksum byte.

Mode change button tells controller either these sensors are being used for robotic arm or controlling car. Difference between two modes is created by differentiating synchronization byte. Transmitter tells receiver using different synchronization byte that upcoming packets are for controlling arm or car.

Controller uses '0xAA' byte for synchronization to tell receiver that upcoming packets are for robotic arm controlling. Similarly by using '0xBB' as synchronization byte transmitter tells receiver that upcoming bytes are for controlling car.

Four sensors are used for car control. One sensor moves car in forward direction and second sensor moves car in reverse direction. When the sensor is bent then the car moves and when the sensor is set in straight position the car stops. Similarly one sensor turns the car to the right side and the other turns the car towards the left.

At receiver side receiver, receiver first checks whether the received bytes are for car control or for arm control. Then controller adds two data bytes and compares it with received checksum. If both are similar, then controller moves motors according to received bytes. If received checksum is not similar to the calculated checksum then bytes are discarded.

Microcontroller according to received bits rotates the servo motors to the defined angles. Controller uses pulse width modulation to rotate motors. Microcontroller sends high pulse and then low pulse to servo motor in pulse width modulation. The high and low pulse time tells servo to rotate to a specific angle.

Dc motor of arm is controlled using motor driver ic 'L293D'. This ic is controlled using microcontroller. For dc motor of car microcontroller switches relay to move the car forward or reverse.

Chapter 7

System Modules

7.1. Flex Sensors

These sensors change their resistance when bent. They are made up of plastic material to give flexibility hence also termed as flex sensors. A very thin conducting material is also finished on the plastic surface that does the conduction of electric signals.

7.1.1. Features

These sensors have the property of angle displacement measurement as they can bend and flex physically with motion device. They are simple in construction and low profile.

Possible uses are in robotics, gaming (Virtual Motion), medical devices, computer peripherals musical instruments and physical therapy.

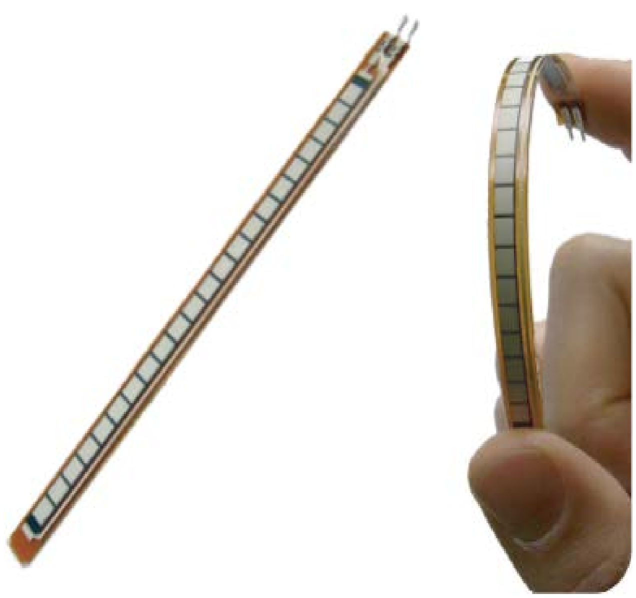

Figure 7.1 TYPICAL FLEX SENSOR

7.1.2. Mechanical Specifications

It has a life cycle of more than 1 million bends, height of 0.43mm (0.017") and temperature range of -35°C to +80°C

7.1.3. Electrical Specifications

Its resistance is around 10K Ohms when flat, resistance tolerance of ±30%. In case of full bend resistance ranges from 60K to 110K Ohms.

Power rating is 0.50 watts continuous supply with 1 watt peak power.

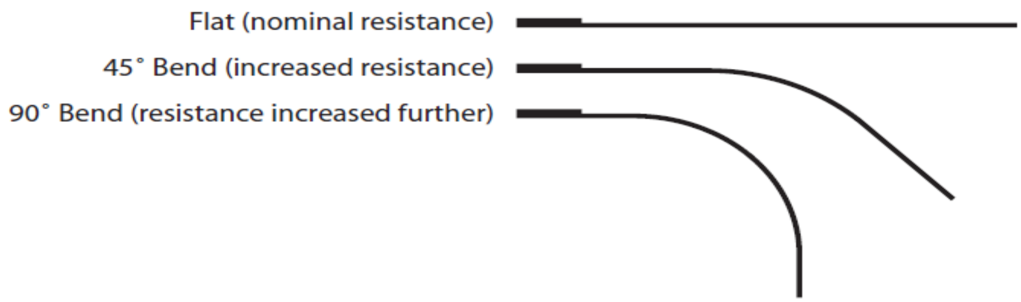

Figure 7.2 Bend Case Illustration

7.2. Servo Motors

The servo motor we have opted is FUTABA S3003 (3-pole), because it produces enough torque to bend a robotic arm finger.

7.2.1. Specifications Of Futaba S3003

Table 7.1 Futaba s3003 Specifications

VOLTS	TORQUE	SPEED
4.8v	3.2kg/cm	.23sec/60°
6.0v	4.1kg/cm	.19sec/60°

7.3 Radio Frequency Transceiver Modules

7.3.1 Transmitter

At the transmitting end the module used is **TLP 434A.**

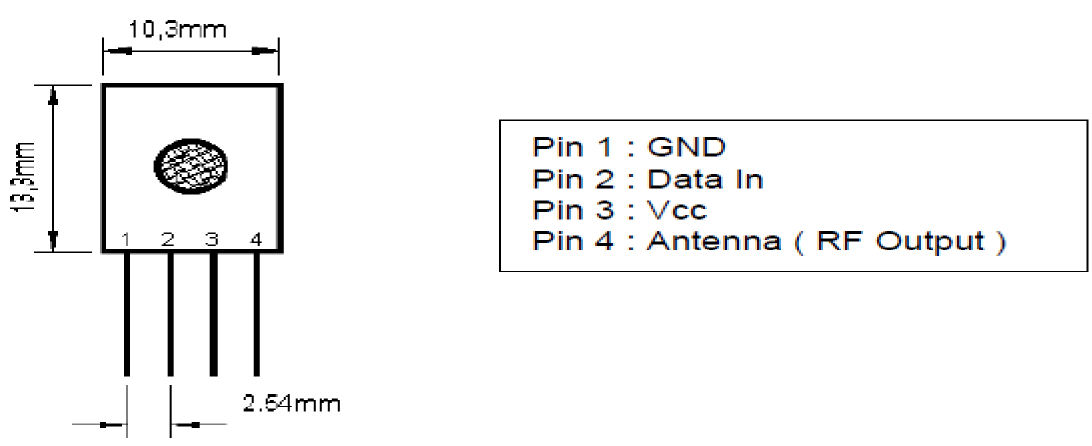

Figure 7.3 Pin Diagram of Transmitter Module

Features

Certain features of TLP 434A are:

The operating frequency is 433.92 MHZ, it uses ASK modulation for transmission of data bits.

Other features include circuit shape of SAW, maximum date rate of 8K bps and supply voltage of 2-12V.

Below, in table 7.2 is a brief description of other parameters such as Transmitter output power, etc.

Table 7.2 : TLP 434A Parameters

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
Vcc	Operating Supply Voltage		2.0	-	12.0	V
Icc	Peak Current		1,64 (2V)	-	19,4 (12V)	mA
Vh	Input High Voltage	Idata=100µA High	Vcc - 0,5	**Vcc**	Vcc + 0,5	V
Vl	Input Low Voltage	Idata=0µA Low	-	-	0,3	V
Fo	Absolute frequency		433,22	**433,92**	434,62	MHz
	Relative to 433,92MHz			**± 150**	± 200	KHz
Po	RF output Power- 50Ohm	Vcc = 9-12V	-	**14**		dBm
		Vcc = 5-6V	-	**16**		dBm
Dr	Data Rate		512	**4.800**	200.000	Bps / Baud

7.3.2. Receiver

At the receiving end the module used is RLP 434A.

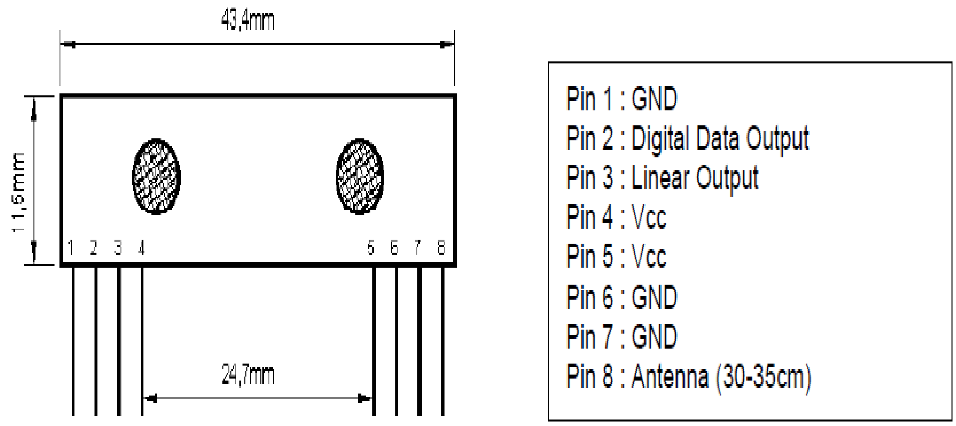

Figure 7.4 Pin Diagram Of Receiver Module

Features

Its frequency of operation is 433.92 MHZ, and demodulates ASK modulated data bits at a supply voltage of 5-6V (desired voltage level).

Below is a brief description of other parameters such as receiver sensitivity, etc.

7.4. Microcontroller

The microcontroller we have opted is micro chip's pic 18f452

TABLE 7.3 RLP 434A PARAMETERS

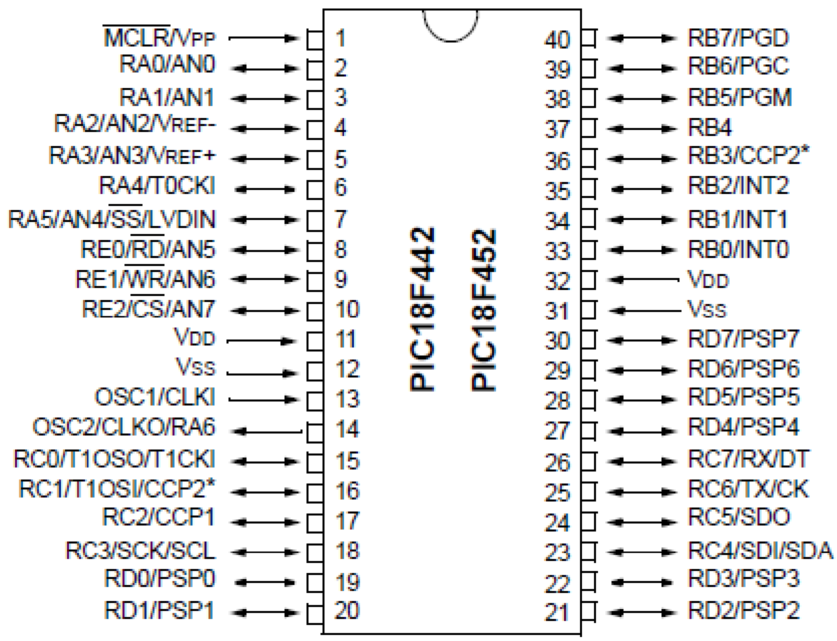

Figure 7.5 Pin Diagram Of Pic 18f452

7.4.1. Features

PIC 18F452 has an addressable USART module built in it which can supports RS-485 and RS-232. It has a compatible 10-bit Analog-to-Digital Converter module (A/D) with:

Fast sampling rate and conversion available during SLEEP

It also has a low power consumption:

< 1.6 mA typical @ 5V, 4 MHz

25 uA typical @ 3V, 32 kHz

< 0.2 uA typical standby current

7.4.2. Analog to Digital Conversion

The A/D allows conversion of an analog input signal to a corresponding 10-bit digital number.

The A/D module has four registers. These registers are:

A/D Result High Register (ADRESH)

A/D Result Low Register (ADRESL)

A/D Control Register 0 (ADCON0)

A/D Control Register 1 (ADCON1)

The ADCON0 register controls the operation of the A/D module. The ADCON1 register configures the functions of the port pins.

Adcon0 Register

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0
ADCS1	**ADCS0**	**CHS2**	**CHS1**	**CHS0**	**GO/DONE**	—	**ADON**

Bit 7 Bit 0

<u>Bit 7-6 ADCS1:ADCS0</u> are A/D Conversion Clock Select bits

TABLE 7.4 ADCON0 CLOCK CONVERSIONS

ADCON1 **<ADCS2>**	**ADCON0** **<ADCS1:ADCS0>**	**Clock Conversion**
0	00	FOSC/2
0	01	FOSC/8
0	10	FOSC/32
0	11	FRC (clock derived from the internal A/D RC oscillator)
1	00	FOSC/4
1	01	FOSC/16
1	10	FOSC/64
1	11	FRC (clock derived from the internal A/D RC oscillator)

bit 5-3 CHS2:CHS0: Analog Channel Select bits

000 = channel 0, (AN0)

001 = channel 1, (AN1)

010 = channel 2, (AN2)

011 = channel 3, (AN3)

100 = channel 4, (AN4)

101 = channel 5, (AN5)

110 = channel 6, (AN6)

111 = channel 7, (AN7)

bit 2 GO/DONE: A/D Conversion Status bit

1 = A/D conversion in progress (setting this bit starts the A/D conversion which is automatically cleared by hardware when the A/D conversion is complete)

0 = A/D conversion not in progress

bit 1 Unimplemented: Read as '0'

bit 0 ADON:

A/D On bit

1 = A/D converter module is powered up

0 = A/D converter module is shut-off and consumes no operating current

a. **Adcon1 Register**

R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	ADCS2	—	—	PCFG3	PCFG2	PCFG1	PCFG0
bit 7							bit 0

bit 7 ADFM: A/D Result Format Select bit

1 = Right justified. Six (6) Most Significant bits of ADRESH are read as ’0’.

0 = Left justified. Six (6) Least Significant bits of ADRESL are read as ’0’.

7.4.3. A/D Result Registers

The ADRESH:ADRESL register pair is the locationwhere the 10-bit A/D result is loaded at the completionof the A/D conversion. This register pair is 16-bits wide.The A/D module gives the flexibility to left or right justifythe 10-bit result in the 16-bit result register [7]. The A/D Format Select bit (ADFM) controls this justification. The diagram below gives a visual representation of this concept.

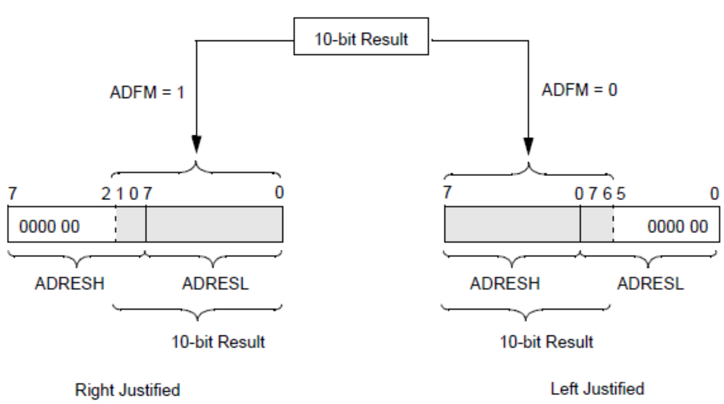

Figure 7.6 Adresh:Adresl Register Illustration

bit 6 ADCS2: A/D Conversion Clock Select bit

Table 7.5 ADCON1 clock conversion

ADCON1 <ADCS2>	**ADCON0 <ADCS1:ADCS0>**	**Clock Conversion**
0	00	FOSC/2
0	01	FOSC/8
0	10	FOSC/32
0	11	FRC (clock derived from the internal A/D RC oscillator)
1	00	FOSC/4
1	01	FOSC/16
1	10	FOSC/64
1	11	FRC (clock derived from the internal A/D RC oscillator)

<u>Bit 5-4 Unimplemented</u>: Read as '0' **<u>bit 3-0 PCFG3:PCFG0</u>:** A/D Port

Table 7.6 Analog & Digital Port Selection

PCFG <3:0>	AN7	AN6	AN5	AN4	AN3	AN2	AN1	AN0	VREF+	VREF-	C / R
0000	A	A	A	A	A	A	A	A	VDD	VSS	8 / 0
0001	A	A	A	A	VREF+	A	A	A	AN3	VSS	7 / 1
0010	D	D	D	A	A	A	A	A	VDD	VSS	5 / 0
0011	D	D	D	A	VREF+	A	A	A	AN3	VSS	4 / 1
0100	D	D	D	D	A	D	A	A	VDD	VSS	3 / 0
0101	D	D	D	D	VREF+	D	A	A	AN3	VSS	2 / 1
011x	D	D	D	D	D	D	D	D	—	—	0 / 0
1000	A	A	A	A	VREF+	VREF-	A	A	AN3	AN2	6 / 2
1001	D	D	A	A	A	A	A	A	VDD	VSS	6 / 0
1010	D	D	A	A	VREF+	A	A	A	AN3	VSS	5 / 1
1011	D	D	A	A	VREF+	VREF-	A	A	AN3	AN2	4 / 2
1100	D	D	D	A	VREF+	VREF-	A	A	AN3	AN2	3 / 2
1101	D	D	D	D	VREF+	VREF-	A	A	AN3	AN2	2 / 2
1110	D	D	D	D	D	D	D	A	VDD	VSS	1 / 0
1111	D	D	D	D	VREF+	VREF-	D	A	AN3	AN2	1 / 2

Configuration Control bits

A = Analog input **D =** Digital I/O

C/R = # of analog input channels / # of A/D voltage references

Chapter 8

Future Improvements

8.1. Future improvements

There are several changes that could be made to increase the functionality and performance of WSRA. As this would be the first version of WSRA, it is considered infeasible to address these issues in the allowable time frame.

8.1.1. Hardware improvements

It would also be beneficial to implement a bank of ultra-capacitors to help protect the batteries from the peak current draw of the motor controllers by providing a short supply of high current when needed.

They also provide adequate power and incredible sensitivity to detect even the smallest possible movement and safe control through redundant systems. They also provide integrally low cost.

8.1.2. Mechanical structure up gradation

Support and balance the whole structure of the robotic arm. The motors used on WSRA are incapable of handling large weights. Replacing these with more powerful motors, robust gears and using an inertial measurement unit would improve WSRA's performance with larger weights. More number of joints will increase the work efficiency.

8.1.3. Software improvements

Provide electric power i.e. besides being clean and extremely efficient, electric power enables fine adjustments to be made to the robotic arm (for smoothing the movement of the robotic arm), and a precise, software-based approach to maintain the required sensitivity of the robotic arm. These improvements also provide smart battery management. It should allow for more efficient energy use that leads to a longer battery life and should add in regenerative braking capability. Processing delays to be reduced at the smallest level.

8.2. Future enhancement Integration with other technologies

WSAR easy interface allows the technology not only used easily but to integrate it with other technologies as well

8.2.1. GSM

The use of GSM will increase the range to everywhere and any time. As GSM is a mature technique and the WSRA coding has been done in such a way that it can easily integrated with the technology like GSM,GPRS etc

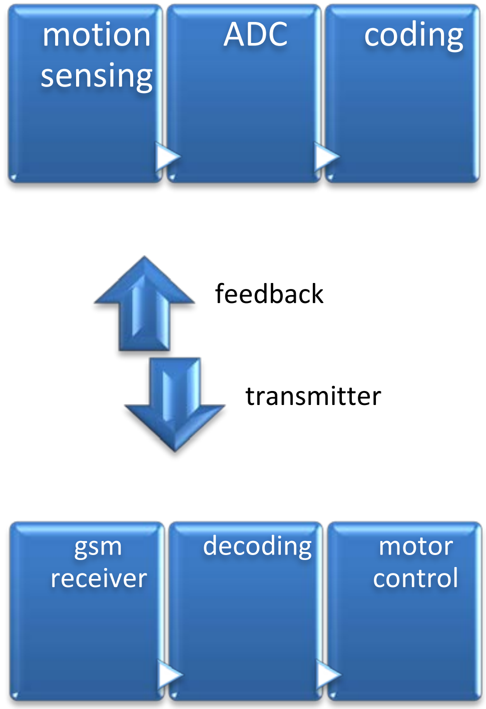

Figure 8.1 Gsm As A Wireless Medium

Its Advantages are that it provides a higher range and thus can be operated from anywhere. It is also economical since this technology is easily available.

8.2.2. Intelligence in operation

The whole robot can be given its own brain. It will be able to decide its own action in different conditions and situations. For this purpose a camera can be use. The camera captures video that will be processed and then different conditions can be detected by using image processing. Accordingly the robot can perform different functions.

Advantages of this feature are that first of all it is self operatable which means that it will be able to make its own decisions. Thus operator costs are saved.

8.2.3. Brain controlled

The future of the robots is to perform every action that you can think of. All you need is to detect different brain signals and accordingly send the particular command to robot. For this purpose electrodes are available that are used to detect very small signals.

For such type of robots no actual performance of the movements are required since the signals detected will the electric signals from the brain. It also has a large use in medical field.

8.2.4. Whole robot synchronized

The same idea can be extended to the whole robot. For the detection of the signals different sensors can be used like, bend sensors, tilt sensors, pressure sensors and web cam for detection of motion.

These sensors will detect different position of body and accordingly will transmit the data to the robot that will perform the required function

By using such robots human life becomes safe as all the risky work is done by the robots as robots perform in place of people. They are also easy to operate.

8.2.5. Haptic sensors

Haptic sensors can also be used. These sensors are used to observe the amount of the weight picked by the robot at the transmitter end. So you can also observe the weight of the object being picked. By doing this the capability of the robot will be efficiently utilized. Safety for the motors is ensured as user himself will observe the weight. Utilization of the capabilities are more efficiently done.

Chapter 9

GETTING STARTED

Figure 9.1 WIRELESSLY SYNCHRONIZED ROBOTIC ARM

9.1. Introduction to the Modes

This is a dual mode system. Each mode has a special use attached to it to make the system user friendly and easy to use. These are ‘Robotic Arm Mode’ and ‘Vehicle Control Mode.

9.1.1. Robotic Arm Mode

This mode will allow you to operate the robotic arm exclusively. This mode will be initialized by switching the ‘Mode’ button towards ‘RAM’ side. Its operation is discussed later.

9.1.2. Vehicle Control Mode

This mode will allow you to operate the vehicle, which moves the car exclusively. This mode will be initialized by switching the ‘Mode’ button towards ‘VCM’ side. Its operation is also discussed later. When in this mode, the vehicle can be switched on or off using a separate switch. This saves the batteries.

9.2. Controlling Glove

A controlling glove is provided to you. The user wears the controlling glove with the sensors attached to it. This acts as the main controller of the system.

9.2.1. Wearing The Glove

You are required to wear the controlling glove as it is the main controller. When worn, only then can the RAM and VCM can be done. Care should be taken while wearing and taking off the gloves.

9.2.2. Fingers Movement

When in Robotic Arm Mode, the movement of your fingers (while wearing the glove) is exactly replicated by the robotic arm. As you move your fingers individually or all at once, the robotic arm will also move its fingers in similarity to your motion.

9.2.3. Elbow Movement

A knob can be seen on the glove, just at the end of the palm and above the wrist position. If you want to move the whole arm either up or down, put the system in RAM. This has been already explained to you in section 2.2.1. There is a mark on the top of the knob as well as on the glove. Rotate the knob in clockwise direction from its mean position on the knob and the arm will rotate in one direction. To stop the arm, bring the knob back to its mean position on the mark. Now if you want to move the arm on the other direction, rotate the knob counterclockwise.

9.2.4. Vehicle movement

In order to move the system to the desired location of operation, the vehicle beneath the robotic arm can be used. This can be done by switching to the [13]VHM. This has been already explained in section 2.2.2. Now if you want the vehicle to move forward you have to close your index finger. To move the vehicle in backward direction, close your middle finger. To turn right or left, close your third and fourth finger respectively along with first or second finger according to the direction of the required location.

9.3. How To Operate

As can be seen, the robotic arm comes with a controlling glove.

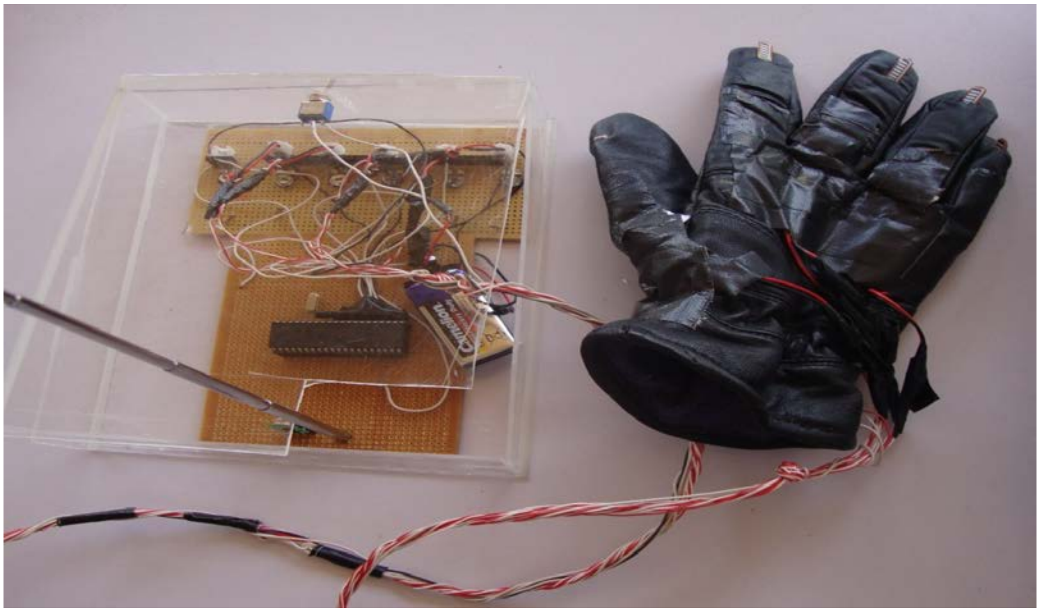

Figure 9.2 VIRTUAL SENSING GLOVE

The controlling circuitry of the glove is put in a bag (also available with the system), which you can put on for your easiness since it provides you enhanced mobility. You have to wear this glove.

The system can be taken to the desired area of operation by enabling the VCM. This can be done by the movement of the fingers as explained in 2.3.4. Once you reach there, use the movement of your hand once again to stop the vehicle. By now you are clearly aware of how to stop it as explained to you in section 2.3.4.

Now put the system in RAM. By doing this you have discontinued the signals operating the vehicle. Now move your fingers according to your desired operation/task. The robotic arm will perform the same movements and you can perform your task easily.

If you experience height problems use the elbow knob on the glove to move the robotic arm up or down. The robotic arm can work at a height of 22-inches above its base. The whole operation of RF communication summarized in figure 9.3[7].

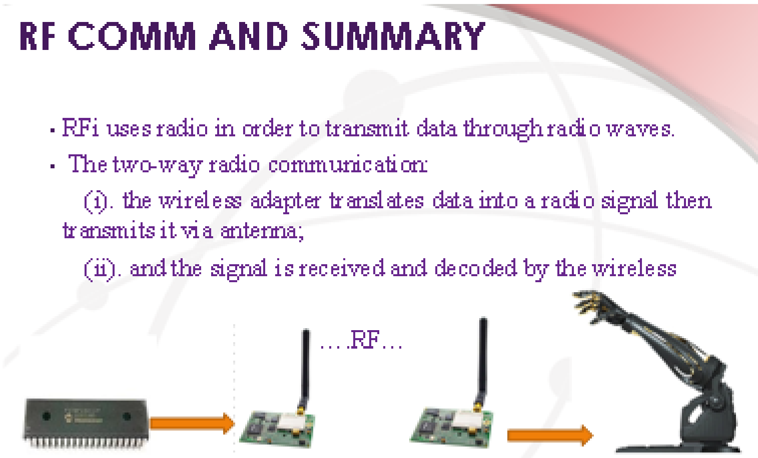

FIGURE 9.3 SUMMARY

For precautions and safety, see appendix A-1.

11.Appendix-A1

a. HOW TO CHARGE THE BATTERIES

The batteries can be charged by connecting the end of the batteries with a voltage supply greater than the voltage of either of the batteries. You should charge the batteries for at least 6-8 hours for a smooth running time of 3 hours.

b. PRECAUTIONS & SAFETY

- Before operating the circuitry, make sure no wires or parts are short circuited, that may cause a spark leading to fire or battery discharge.
- Always use the standard 6.0 volts batteries with peak current rating of 4 ampere or more.
- Avoid running car on rough places, which may lead to displacement of circuit components.
- Keep special care while wearing glove as it contains sensitive sensors.
- Due to a minute delay in the car's DC motor, wait for one signal to move the car then change direction or send the next signal.

c. TROUBLESHOOTING

In case the device do not operates, take the following measures to make the device work again:

- First of all make sure the batteries are fully charged so that the servo motors get appropriate current.
- Insert transceiver antennas in order to improve the signal strength.
- Make sure the switches are in their "ON" positions as marked.
- Mode is accurately set in which the device is required to be operated.

12.Appendix-A2

a. Code for Transmitter

```
void main()
{
unsignedinttemp_res;
intx,a,b,c,d,e,f,g,h,k,j,loop=1;
  char i,mot1,mot2,mot3,mot4,mot5,mot6,byte1,byte2,byte3,tail1=0x40,tail2=0xC0;
  ADCON1 = 0x80;  // Configure analog inputs and Vref
TRISA  = 0xFF;  // PORTA is input
TRISB  = 0x3F;  // Pins RB7, RB6 are outputs
TRISD  = 0;     // PORTD is output
  trisb.f4=1;     // Pin B4 as input
  portb.f4=0;
  UART1_Init(7200);    // Initializating UART1 module 7200 bps
Delay_ms(100);       // Wait for UART module to stabilize
  portc.f3=1;
while(1)
  {
   /////////// Analog input at channel 0 /////////////////
temp_res = Adc_Read(0); // Get results of AD conversion
      PORTD = temp_res;       // Send lower 8 bits to PORTD
      PORTB = temp_res>> 2;  // Send 2 most significant bits to RB7, RB6
      a=portb.f7;
      b=portb.f6;
      c=portd.f7;
```

```
      d=portd.f6;
      e=portd.f5;
      f=portd.f4;
      g=portd.f3;
      h=portd.f2;
      k=portd.f1;
      j=portd.f0;
      x=(a*512)+(b*256)+(c*128)+(d*64)+(e*32)+(f*16)+(g*8)+(h*4)+(k*2)+(j*1); //
Calculating received voltage level
if (x<810)          // Checking voltage level of sensor
      {
       mot1=0x00;
      }
if(x>810)
      {
       mot1=0x01;
       }
       /////////// Analog input at channel 1 /////////////////
temp_res = Adc_Read(1); // Get results of AD conversion
      PORTD = temp_res;       // Send lower 8 bits to PORTD
      PORTB = temp_res>> 2;  // Send 2 most significant bits to RB7, RB6
      a=portb.f7;
      b=portb.f6;
      c=portd.f7;
      d=portd.f6;
      e=portd.f5;
      f=portd.f4;
      g=portd.f3;
```

```
    h=portd.f2;

    k=portd.f1;

    j=portd.f0;

    x=(a*512)+(b*256)+(c*128)+(d*64)+(e*32)+(f*16)+(g*8)+(h*4)+(k*2)+(j*1); //
Calculating received voltage level

if (x<810)              // Checking voltage level of sensor

    {

     mot2=0x00;

    }

if(x>810)

    {

     mot2=0x04;

     }

delay_ms(30);

/////////// Analog input at channel 2 /////////////////

temp_res = Adc_Read(2); // Get results of AD conversion

    PORTD = temp_res;       // Send lower 8 bits to PORTD

    PORTB = temp_res>> 2;  // Send 2 most significant bits to RB7, RB6

    a=portb.f7;

    b=portb.f6;

    c=portd.f7;

    d=portd.f6;

    e=portd.f5;

    f=portd.f4;

    g=portd.f3;

    h=portd.f2;

    k=portd.f1;

    j=portd.f0;
```

```
     x=(a*512)+(b*256)+(c*128)+(d*64)+(e*32)+(f*16)+(g*8)+(h*4)+(k*2)+(j*1); //
Calculating received voltage level

if (x<810)          // Checking voltage level of sensor

     {

      mot3=0x00;

     }

if(x>810)

     {

      mot3=0x10;

      }

delay_ms(30);

      /////////// Analog input at channel 3 /////////////////

temp_res = Adc_Read(3); // Get results of AD conversion

     PORTD = temp_res;      // Send lower 8 bits to PORTD

     PORTB = temp_res>> 2;  // Send 2 most significant bits to RB7, RB6

     a=portb.f7;

     b=portb.f6;

     c=portd.f7;

     d=portd.f6;

     e=portd.f5;

     f=portd.f4;

     g=portd.f3;

     h=portd.f2;

     k=portd.f1;

     j=portd.f0;

     x=(a*512)+(b*256)+(c*128)+(d*64)+(e*32)+(f*16)+(g*8)+(h*4)+(k*2)+(j*1);  //
Calculating received voltage level

if (x<810)         // Checking voltage level of sensor
```

```
{
 mot4=0x00;
}
if(x>810)
{
 mot4=0x01;
 }
delay_ms(30);
```

/////////// Analog input at channel 4 /////////////////

```
temp_res = Adc_Read(4); // Get results of AD conversion
PORTD = temp_res;       // Send lower 8 bits to PORTD
PORTB = temp_res>> 2;  // Send 2 most significant bits to RB7, RB6
a=portb.f7;
b=portb.f6;
c=portd.f7;
d=portd.f6;
e=portd.f5;
f=portd.f4;
g=portd.f3;
h=portd.f2;
k=portd.f1;
j=portd.f0;
x=(a*512)+(b*256)+(c*128)+(d*64)+(e*32)+(f*16)+(g*8)+(h*4)+(k*2)+(j*1); //
Calculating received voltage level
if (x<810)             // Checking voltage level of sensor
{
 mot5=0x00;
}
```

```
if(x>810)
      {
       mot5=0x04;
       }
delay_ms(30);
       /////////// Analog input at channel 5 /////////////////
temp_res = Adc_Read(5); // Get results of AD conversion
      PORTD = temp_res;       // Send lower 8 bits to PORTD
      PORTB = temp_res>> 2;  // Send 2 most significant bits to RB7, RB6
      a=portb.f7;
      b=portb.f6;
      c=portd.f7;
      d=portd.f6;
      e=portd.f5;
      f=portd.f4;
      g=portd.f3;
      h=portd.f2;
      k=portd.f1;
      j=portd.f0;
      x=(a*512)+(b*256)+(c*128)+(d*64)+(e*32)+(f*16)+(g*8)+(h*4)+(k*2)+(j*1); //
Calculating received voltage level
if (x<520)             // Checking voltage level of sensor
      {
       mot6=0x10;
      }
if(x>520 && x<750)
      {
       mot6=0x00;
```

```
       }
if(x>750)
       {
       mot6=0x20;
       }
     byte1=mot1+mot2+mot3+tail1;     // Byte for first three motors
     byte2=mot4+mot5+mot6+tail2;     // Byte for next three motors
     byte3=byte2+byte1;      // Calculating  Checksum
for(loop=1;loop<=50;loop++)
     {
if(portb.f4 == 0)       // Checking controlling mode
       {
i=0xAA;
        UART1_Write(i);     // Transmit Synchronization Byte
i=byte1;
        UART1_Write(i);     // Transmit Byte for first three motors
i=byte2;
        UART1_Write(i);     // Transmit Byte for next three motors
i=byte3;
        UART1_Write(i);     // Transmit Checksum Byte
       }
if(portb.f4 == 1)      // if mode is for car controlling
       {
i=0xBB;
        UART1_Write(i);      // Transmit Synchronization Byte
i=byte1;
        UART1_Write(i);
```

```
i=byte2;
        UART1_Write(i);
i=byte3;
        UART1_Write(i);
       }

      }
  }
}
```

a. Code for Receiver

```
void main()
{
char i,i1,i2,a,byte1,byte2,byte3,chk;
int loop=0,b,c,d,e,f,g,j,k,l,m,n,o;
 trisc.f1=0;            // pin R1 as output
 trisc.f2=0;            // pin R2 as output
 trisc.f3=0;            // pin R3 as output
trisb=0;

trisd=0x00;             // port D as output
portd=0x00;
 UART1_Init(7200);       // Initializating UART1 module 7200 bps
Delay_ms(100);          // Wait for UART module to stabilize
  trisc.f4=0;
  portc.f4=1;
```

```
while(1)
 {
if (UART1_Data_Ready())   // Check if data is Received at receiver pin
   {
i = UART1_Read();       // Read received value and save in i
    a=i;
if(i==0xAA)
    {
do {
if (UART1_Data_Ready()) {
i = UART1_Read();}
      }   while(a==i);              //loop continues until new value is received
      byte1=i;
      a=i;
do {
if (UART1_Data_Ready()) {
i = UART1_Read();}
        } while(a==i);          //loop continues until new value is received
       byte2=i;
         a=i;
do {
if (UART1_Data_Ready()) {
i = UART1_Read();}
        } while(a==i);         //loop continues until new value is received
       byte3=i;
chk=byte1+byte2;
if(chk==byte3)          // if received checksum equals calculated checksum
```

```
{
portb=byte1;
//////////// motor 1 /////////////
if(portb.f0==0 && portb.f1==0)
{
PORTd.f0 = 1;            // Turn servo to zero degree
delay_us(500);
PORTd.f0 = 0;
delay_ms(19);
delay_us(500);
}
if(portb.f0==1 && portb.f1==0)
{
PORTd.f0 = 1;            // Turn servo to 180 degree
delay_ms(2);
delay_us(500);
PORTd.f0 = 0;
delay_ms(17);
delay_us(500);
}
/////////// motor 2  ///////////////
if(portb.f2==0 && portb.f3==0)
{
PORTd.f1 = 1;            // Turn servo to zero degree
delay_us(500);
PORTd.f1 = 0;
delay_ms(19);
```

```
delay_us(500);
        }
if(portb.f2==1 && portb.f3==0)
        {
          PORTd.f1 = 1;             // Turn servo to 180 degree
delay_ms(2);
delay_us(500);
               PORTd.f1 = 0;
delay_ms(17);
delay_us(500);
          }
       /////// motor 3 ///////////
if(portb.f4==0 && portb.f5==0)
        {
        PORTd.f2 = 1;               // Turn servo to zero degree
delay_us(500);
               PORTd.f2 = 0;
delay_ms(19);
delay_us(500);
        }
if(portb.f4==1 && portb.f5==0)
        {
          PORTd.f2 = 1;              // Turn servo to 180 degree
delay_ms(2);
delay_us(500);
               PORTd.f2 = 0;
delay_ms(17);
```

```
delay_us(500);
          }
portb=byte2;
/////// motor 4 //////////////
if(portb.f0==0 && portb.f1==0)
        {
        PORTd.f3 = 1;              // Turn servo to zero degree
delay_us(500);
              PORTd.f3 = 0;
delay_ms(19);
delay_us(500);
         }
if(portb.f0==1 && portb.f1==0)
         {
          PORTd.f3 = 1;          // Turn servo to 180 degree
delay_ms(2);
delay_us(500);
              PORTd.f3 = 0;
delay_ms(17);
delay_us(500);
          }
         //////// motor 5 //////////
if(portb.f2==0 && portb.f3==0)
        {
        PORTd.f4 = 1;              // Turn servo to zero degree
delay_us(500);
              PORTd.f4 = 0;
```

```
delay_ms(19);
delay_us(500);
        }
if(portb.f2==1 && portb.f3==0)
       {
          PORTd.f4 = 1;           // Turn servo to 180 degree
delay_ms(2);
delay_us(500);
              PORTd.f4 = 0;
delay_ms(17);
delay_us(500);
          }
        ////// DC motor controlling of Arm  //////////
if(portb.f4==0 && portb.f5==0)
       {
        portd.f6=0;                // Stop Moving DC
        portd.f7=0;
        }
if(portb.f4==1 && portb.f5==0)
        {
         portd.f6=1;              // Start Moving DC Clockwise
         portd.f7=0;
        }
if(portb.f4==0 && portb.f5==1)
        {
         portd.f6=0;              // Start Moving DC Anti-Clockwise
         portd.f7=1;
```

```
        }
      }
    }
    ////////////////////  CAR Controlling ///////////////////
if(i==0xBB)
    {
do {
if (UART1_Data_Ready()) {   // Check if data is Received at receiver pin
i = UART1_Read();}           // Read received value and save in i
      }   while(a==i);              //loop continues until new value is received
      byte1=i;
      a=i;
do {
if (UART1_Data_Ready()) {
i = UART1_Read();}
        } while(a==i);
       byte2=i;
         a=i;
do {
if (UART1_Data_Ready()) {
i = UART1_Read();}
        } while(a==i);
       byte3=i;
chk=byte1+byte2;
if(chk==byte3)
       {
portb=byte1;
```

```
////////// DC motor of car forward & stop //////////
if(portb.f2==0 && portb.f3==0 && portb.f4==0 && portb.f5==0)
        {
         portc.f1=0;              // Stop motor
         portc.f2=0;
        }
if(portb.f2==1 && portb.f3==0)
        {
         portc.f1=0;            // start motor
         portc.f2=1;
         }
///////// DC motor of car reverse & stop ////////
if(portb.f4==0 && portb.f5==0 && portb.f2==0 && portb.f3==0)
         {
          portc.f1=0;            // Stop motor
          portc.f2=0;
         }
if(portb.f4==1 && portb.f5==0)
        {
          portc.f1=1;          // start motor
          portc.f2=0;
         }
portb=byte2;
         /////// Servo of car turn Left ///////////////
if(portb.f0==0 && portb.f1==0 && portb.f2==0 && portb.f3==0)
        {
          portc.f3=1;              // turn servo to 90 degree
```

```
delay_ms(1);
delay_us(500);
                portc.f3=0;
delay_ms(18);
delay_us(500);
         }
if(portb.f0==1 && portb.f1==0)
         {
          portc.f3=1;          // turn servo to 0 degree
delay_us(500);
              portc.f3=0;
delay_ms(19);
delay_us(500);
         }
        /////// Servo of car turn Right //////////////
if(portb.f2==0 && portb.f3==0 && portb.f0==0 && portb.f1==0)
        {
        portc.f3=1;            // turn servo to 90 degree
delay_ms(1);
delay_us(500);
                portc.f3=0;
delay_ms(18);
delay_us(500);
        }
if(portb.f2==1 && portb.f3==0)
        {
         portc.f3=1;           // turn servo to 180 degree
```

```
delay_ms(2);
delay_us(500);
        portc.f3=0;
delay_ms(17);
delay_us(500);
     }
    }
   }
  }
 }
}
```

13. Appendix A-3

GLOSSARY

- **Wireless:** Communication in which electromagnetic waves (rather than some form of wire) carry the signal over the communication path
- **Database:** A structured set of data held in a computer, esp. one that is accessible in various ways
- **Graphical User Interface:** A visual way of interacting with a computer using items such as windows, icons, and menus, used by most modern operating systems
- **Modes:** A designated condition or status, as for performing a task
- **Circuitry:** Electronic equipment consisting of a system of circuits
- **Knob:** A round control switches or dial, which in this case is black and 0.5’’ long. It is used as a controlling tool
- **Robotic Arm:** A mechanical device that sometimes resembles a human and is capable of performing a variety of often complex human tasks on command or by being programmed in advance.
- **Controlling Glove:** a specially designed glove that is fitted with sensors and that drives the system machinery of the whole robot.
- **Robotic Arm Mode (RAM):** A mode in which the machine operates only the robotic arm.
- **Vehicle Control Mode (VHM):** A mode in which the machine operates only the robotic arm.
- **WSRA:** Acronym for wirelessly synchronized robotic arm.
- **Code:** Continuum of computerized language, which can be understood by machine