NEAR EAST UNIVERSITY

Faculty Of Engineering

Department of Electrical and Electronic Engineering

ROBOTiC ARM

Graduation Project
EE-400

Student: Görkem Say (20091359)

Supervisor: Dr. Ali Serener

Lefkoşa-2014

TABLE OF CONTENTS

ACKNOWLEDGMENTS

First of all i want to thank my all teachers.under their helps , i learned a big knowledge bout my department. I want to thank Cemal Kavalcıoğlu to be my advisor. He helps me a lot f things . I want to thank another teacher Özgür Özerdem .i exploit from his knowledge.Then want to thank Ali Serener .He helps me about my graduation project and he helps me for his nowledge.

Special thans to Ersin Aytaç ,Fatih Emrem , Murat Arslan and Nurullah akkaya. They ıelp me for my project. I want to thank my other friends in NEU : Deniz Kuran Artun Faslı .

And last of all, i want to thank my family :my mom , my brother and my cousins for ıupport me.

ABSTRACT

The robotic arm is very important nowadays.Becouse its field is very wide. My robotic arm is designed to make simple operations but it can develop in different areas.

For body level, i used flexiglass material. I used it because its not heavy , ıt ıs a strong material and we can apply it easily. But on the other hand it has some disadvantages. Example it can be broken if you make a big pressure so it can't carry heavy materials. I found the robotic arm drawings from internet and i cut all flexiglass.

For electronic level, i make a circuit that control the servomotor from joystick via wifi module. The main microprocessor is a Atmega328p. I used 2 Atmega 328p. One of these processors is read analogue valu and transmit data. Another processor is receive and apply. You can control servomotor easily using this processor. The main principle is this: There is a joystick .you connect the joystick to microprocessor's analogue pins. We connect to the analogue pins, becouse there is a potentiometer the inside of the joystick and analogue pin can read analogue value and it can converts to pwm for servomotor. When it converts the analogue value to pwm, the processor send the information to another processor via xbee wifi module. The second processor takes data and apply it to digital pins.and i used LEDs for be sure if the circuit is working.

For programming level, i found some programs from internet and i adapt it for my circuit.

But in this level some of my friends help me becouse its a hard for electrical electronic engineering student.

INTRODUCTION

Technology is developing day by day.There are a lot of technology for make human fe easier. The robots are entering the human life slowly and it makes human life easier. You an see robots in every area of the life. For example; home robots , medical robots, rescue obots. As a normal robots, the robotic arms make life easier for human.

The robotic arm is a kind of robots. Ofcourse the robotic arm is very important. I hoose this project because the robotic arm can develop wide areas. You can use it in every ield of the life. And robotic arms is similar to human arm functions. You can use it in ifferent areas. For example today, the NASA is using robotic arms in the Mars for research. he other example is you can use very sensitive robotic arm for critical surgicals

In chapter 1, i introduces robots.what is robot? The history of robots, the types of obots. I explained all of these introduce.

In chapter 2, i entry the robotic hand.what is robotic arm ? where do we use? How can e use?

In chapter 3, i started to explain the hardware of the robotic arm.

I found a lot of chance to see behave of the robotic arm and i analyzed them.

CHAPTER 1

1.0-WHAT IS ROBOT

A robot is a mechanical or virtual artificial agent, usually an electro-mechanical machine that is guided by a computer program orelectronic circuitry. Robots can be autonomous or semi-autonomous and range from humanoids such as Honda's *Advanced Step in Innovative Mobility* (ASIMO) and TOSY's *TOSY Ping Pong Playing Robot* (TOPIO) to industrial robots, collectively programmed 'swarm' robots, and even microscopic nano robots. By mimicking a lifelike appearance or automating movements, a robot may convey a sense of intelligence or thought of its own.

The branch of technology that deals with the design, construction, operation, and application of robots, as well as computer systems for their control, sensory feedback, and information processing is robotics. These technologies deal with automated machines that can take the place of humans in dangerous environments or manufacturing processes, or resemble humans in appearance, behavior, and/or cognition. Many of today's robots are inspired by nature contributing to the field of bio-inspired robotics. These robots have also created a newer branch of robotics: Soft robotics.

From the time of ancient civilization there have been many accounts of user-configurable automated devices and even automataresembling animals and humans, designed primarily as entertainment. As mechanical techniques developed through the Industrial age, there appeared more practical applications such as automated machines, remote-control and wireless remote-control. Electronics evolved into the driving force of development with the advent of the first electronic autonomous robots created by William Grey Walter in Bristol,England in 1948. The first digital and programmable robot was invented by George Devol in 1954 and was named the Unimate. It was sold to General Motors in 1961 where it was used to lift pieces of hot metal from die casting machines at the Inland Fisher Guide Plant in theWest Trenton section of Ewing Township, New Jersey.

Robots have replaced humans in the assistance of performing those repetitive and dangerous tasks which humans prefer not to do, or are unable to do due to size limitations, or even those such as in outer space or at the bottom of the sea where humans could not survive the extreme environments.

1.1-SUMMARY

The word *robot* can refer to both physical robots and virtual software agents, but the latter are usually referred to as bots. There is no consensus on which machines qualify as robots but there is general agreement among experts, and the public, that robots tend to do some or all of the following: move around, operate a mechanical limb, sense and manipulate their environment, and exhibit intelligent behavior — especially behavior which mimics humans or other animals. In practical terms, "robot" usually refers to a machine which can be electronically programmed to carry out a variety of physical tasks or actions. There is no one definition of *robot* that satisfies everyone and many people have their own. For example Joseph Engelberger, a pioneer in industrial robotics, once remarked: "I can't define a robot, but I know one when I see one." The two ways that robots differ from actualbeings are, simply stated, in the domain of cognition, and in the domain of biological form. The general consensus is that a "robot" is a machine and not a being simply because it is not intelligent (it requires programming to function), regardless of how human-like it may appear. In contrast, an imaginary "machine" or "artificial life form" (as in science fiction) that could think near or above human intelligence, and had a sensory body, would no longer be a "robot" but would be some kind of "artificial being" or "cognitive robot", (see also cyborg).

According to the Encyclopaedia Britannica a robot is "any automatically operated machine that replaces human effort, though it may not resemble human beings in appearance or perform functions in a humanlike manner." Merriam-Webster describes a robot as a "machine that looks like a human being and performs various complex acts (as walking or talking) of a human being", or a "device that automatically performs complicated often repetitive tasks", or a "mechanism guided by automatic controls".

1.2 HISTORY

The idea of automata originates in the mythologies of many cultures around the world. Engineers and inventors from ancient civilizations, including Ancient China, Ancient Greece, and Ptolemaic Egypt, attempted to build self-operating machines, some resembling animals and humans. Early descriptions of automata include the artificial doves of Archytas, the artificial birds of Mozi and Lu Ban, a "speaking" automaton by Hero of Alexandria, a washstand automaton by Philo of Byzantium, and a human automaton described in the *Lie Zi*.

Renaissance Italy, Leonardo da Vinci (1452–1519) sketched plans for a humanoid robot around 1495. Da Vinci's notebooks, rediscovered in the 1950s, contained detailed drawings of a mechanical knight now known as Leonardo's robot, able to sit up, wave its arms and move its head and jaw. The design was probably based on anatomical research recorded in his *Vitruvian Man*. It is not known whether he attempted to build it.

In Japan, complex animal and human automata were built between the 17th to 19th centuries, with many described in the 18th century *Karakuri zui(Illustrated Machinery*, 1796). One such automaton was the karakuri ningyō, a mechanized puppet. Different variations of the karakuri existed: the*Butai karakuri*, which were used in theatre, the *Zashiki karakuri*, which were small and used in homes, and the *Dashi karakuri* which were used in religious festivals, where the puppets were used to perform reenactments of traditional myths and legends.

In France, between 1738 and 1739, Jacques de Vaucanson exhibited several life-sized automatons: a flute player, a pipe player and a duck. The mechanical duck could flap its wings, crane its neck, and swallow food from the exhibitor's hand, and it gave the illusion of digesting its food by excreting matter stored in a hidden compartment.

Modern autonomous robots

The first electronic autonomous robots with complex behaviour were created by William Grey Walter of the Burden Neurological Institute at Bristol, England in 1948 and 1949. He wanted to prove that rich connections between a small number of brain cells could give rise to very complexbehaviors - essentially that the secret of how the brain worked lay in how it was wired up. His first robots, named *Elmer* and *Elsie*, were constructed between 1948 and 1949 and were often described as *tortoises* due to their shape and slow rate of movement. The three-wheeled tortoise robots were capable of phototaxis, by which they could find their way to a recharging station when they ran low on battery power.

Walter stressed the importance of using purely analogue electronics to simulate brain processes at a time when his contemporaries such as Alan Turing and John von Neumann were all turning towards a view of mental processes in terms of digital computation. His work inspired subsequent generations of robotics researchers such as Rodney Brooks, Hans Moravec and Mark Tilden. Modern incarnations of Walter's *turtles* may be found in the form of BEAM robotics.

The first digitally operated and programmable robot was invented by George Devol in 1954 and was ultimately called the Unimate. This ultimately laid the foundations of the modern robotics industry. Devol sold the first Unimate to General Motors in 1960, and it was installed in 1961 in a plant inTrenton, New Jersey to lift hot pieces of metal from a die casting machine and stack them. Devol's patent for the first digitally operated programmable robotic arm represents the foundation of the modern robotics industry.[37] The first palletizing robot was introduced in 1963 by the Fuji Yusoki Kogyo Company.[38] In 1973, a robot with six electromechanically driven axes was patented [39][40] by KUKA robotics in Germany, and the programmable universal manipulation arm was invented by Victor Scheinman in 1976, and the design was sold to Unimation.

Commercial and industrial robots are now in widespread use performing jobs more cheaply or with greater accuracy and reliability than humans. They are also employed for jobs which are too dirty, dangerous or dull to be suitable for humans. Robots are widely used in manufacturing, assembly and packing, transport, earth and space exploration, surgery, weaponry, laboratory research, and mass production of consumer and industrial goods.

Etymology

The word *robot* was introduced to the public by the Czech interwar writer Karel Čapek in his play *R.U.R. (Rossum's Universal Robots)*, published in 1920.[42] The play begins in a factory that makes artificial people called *robots*, though they are closer to the modern ideas of androids, creatures who can be mistaken for humans. They can plainly think for themselves, though they seem happy to serve. At issue is whether the *robots* are being exploited and the consequences of their treatment.

Karel Čapek himself did not coin the word. He wrote a short letter in reference to an etymology in the *Oxford English Dictionary* in which he named his brother, the painter and writer Josef Čapek, as its actual originator.

In an article in the Czech journal *Lidové noviny* in 1933, he explained that he had originally wanted to call the creatures *laboři*("workers", from Latin *labor*). However, he did not like the word, and sought advice from his brother Josef, who suggested "roboti". The word *robota* means literally "corvée", "serf labor", and figuratively "drudgery" or "hard work" in Czech and also (more general) "work", "labor" in many Slavic languages (e.g.: Bulgarian, Russian, Serbian, Slovak, Polish, Macedonian, Ukrainian, archaic Czech). Traditionally the *robota* was the work period a serf (corvée) had to give for his lord, typically 6 months of the year. The origin of the word is the Old Church Slavonic (Old Bulgarian) *rabota* "servitude" ("work" in contemporary Bulgarian and Russian), which in turn comes from the Proto-Indo-European root **orbh-*. *Robot* is cognate with the German root *Arbeit* (work).

The word robotics, used to describe this field of study, was coined by the science fiction writer Isaac Asimov. Asimov created the "*Three Laws of Robotics*" which are a recurring theme in his books. These have since been used by many others to define laws used in fact and fiction.

1.3 TYPES OF ROBOTS

Mobile robot

Mobile robots have the capability to move around in their environment and are not fixed to one physical location. An example of a mobile robot that is in common use today is the *automated guided vehicle* or *automatic guided vehicle* (AGV). An AGV is a mobile robot that follows markers or wires in the floor, or uses vision or lasers.[citation needed] AGVs are discussed later in this article.

Mobile robots are also found in industry, military and security environments. They also appear as consumer products, for entertainment or to perform certain tasks like vacuum cleaning. Mobile robots are the focus of a great deal of current research and almost every major university has one or more labs that focus on mobile robot research.[citation needed] Mobile robots are usually used in tightly controlled environments such as on assembly lines because they have difficulty responding to unexpected interference. Because of this most humans rarely encounter robots. However domestic robots for cleaning and maintenance are increasingly common in and around homes in developed countries. Robots can also be found in military applications.

Industrial robots (manipulating)

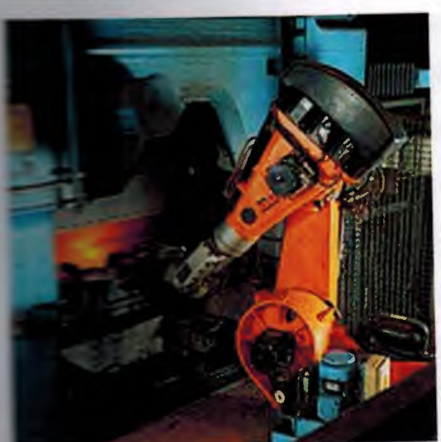


Figure 1.3.1 Industrial robots

A Pick and Place robot in a factory

Industrial robots usually consist of a jointed arm (multi-linked manipulator) and an end effector that is attached to a fixed surface. One of the most common type of end effector is a gripper assembly.

The International Organization for Standardization gives a definition of a manipulating industrial robot in ISO 8373:"an automatically controlled, reprogrammable, multipurpose, manipulator programmable in three or more axes, which may be either fixed in place or mobile for use in industrial automation applications."[46]

This definition is used by the International Federation of Robotics, the European Robotics Research Network (EURON) and many national standards committees.[47]

Service robot

Most commonly industrial robots are fixed robotic arms and manipulators used primarily for production and distribution of goods. The term "service robot" is less well-defined. The International Federation of Robotics has proposed a tentative definition, "A service robot is a robot which operates semi- or fully autonomously to perform services useful to the well-being of humans and equipment, excluding manufacturing operations."[48]

Educational Robot

Robots are used as educational assistants to teachers. From the 1980s, robots such as turtles were used in schools and programmed using the Logo language.[49][50]

There are robot kits like Lego Mindstorms, BIOLOID, OLLO from ROBOTIS, or BotBrain Educational Robots can help children to learn about mathematics, physics, programming, and electronics. Robotics have also been introduced into the lives of elementary and high school students in the form of robot competitions with the company FIRST (For Inspiration and Recognition of Science and Technology). The organization is the foundation for the FIRST Robotics Competition, FIRST LEGO League, Junior FIRST LEGO League, and FIRST Tech Challenge competitions.

There have also been devices shaped like robots such as the teaching computer, Leachim (1974), and 2-XL (1976), a robot shaped game / teaching toy based on an 8-track tapeplayer, both invented Michael J. Freeman.

Modular robot

Modular robots are a new breed of robots that are designed to increase the utilization of robots by modularizing their architecture.[51] The functionality and effectiveness of a modular robot is easier to increase compared to conventional robots. These robots are composed of a single type of identical, several different identical module types, or similarly shaped modules, which vary in size. Their architectural structure allows hyper-redundancy for modular robots, as they can be designed with more than 8 degrees of freedom (DOF). Creating the programming, inverse kinematics and dynamics for modular robots is more complex than with traditional robots. Modular robots may be composed of L-shaped modules, cubic modules, and U and H-shaped modules. ANAT technology, an early modular robotic technology patented by Robotics Design Inc., allows the creation of modular robots from U and H shaped modules that connect in a chain, and are used to form heterogeneous and homogenous modular robot systems. These "ANAT robots" can be designed with "n" DOF as each module is a complete motorized robotic system that folds relatively to the modules connected before and after it in its chain, and therefore a single module allows one degree of freedom. The more modules that are connected to one another, the more degrees of freedom it will have. L-shaped modules can also be designed in a chain, and must become increasingly smaller as the size of the chain increases, as payloads attached to the end of the chain place a greater strain on modules that are further from the base. ANAT H-shaped modules do not suffer from this problem, as their design allows a modular robot to distribute pressure and impacts evenly amongst other attached modules, and therefore payload-carrying capacity does not decrease as the length of the arm increases. Modular robots can be manually or self-reconfigured to form a different robot, that may perform different applications. Because modular robots of the same architecture type are composed of modules that compose different modular robots, a snake-arm robot can combine with another to form a dual or quadra-arm robot, or can split into several mobile robots, and mobile robots can split into multiple smaller ones, or combine with others into a larger or different one. This allows a single modular robot the ability to be fully specialized in a single task, as well as the capacity to be specialized to perform multiple different tasks.

Modular robotic technology is currently being applied in hybrid transportation, industrial automation, duct cleaning and handling. Many research centres and universities have also studied this technology, and have developed prototypes.

Collaborative robots

A *collaborative robot* or *cobot* is a robot that can safely and effectively interact with human workers while performing simple industrial tasks. However, end-effectors and other environmental conditions may create hazards, and as such risk assessments should be done before using any industrial motion-control application.

The collaborative robots most widely used in industries today are manufactured by Universal Robots in Denmark.

Baxter, introduced on September 18, 2012, is a product of Rethink Robotics (whose founder was Rodney Brooks), was an industrial robot selling for about that was designed to safely interact with neighboring human workers and be programmable for performing simple tasks. The robot stopped if its movement encountered a human in the way of its robotic arm and had a prominent off switch, which its human partner could push if necessary. The product, intended for sale to small businesses, was touted as the robotic analogue of the personal computer. Costs were projected to be the equivalent of a worker making an hour.

Military robots

Some experts and academics have questioned the use of robots for military combat, especially when such robots are given some degree of autonomous functions. There are also concerns about technology which might allow some armed robots to be controlled mainly by other robots. The US Navy has funded a report which indicates that, as military robots become more complex, there should be greater attention to implications of their ability to make autonomous decisions. One researcher states that autonomous robots might be more humane, as they could make decisions more effectively. However, other experts question this.

One robot in particular, the EATR, has generated public concerns over its fuel source, as it can continually refuel itself using organic substances. Although the engine for the EATR is designed to run on biomass and vegetation specifically selected by its sensors, which it can find on battlefields or other local environments, the project has stated that chicken fat can also be used.

Manuel De Landa has noted that "smart missiles" and autonomous bombs equipped with artificial perception can be considered robots, as they make some of their decisions autonomously. He believes this represents an important and dangerous trend in which humans are handing over important decisions to machines.

Factory robots

Car production:

Over the last three decades, automobile factories have become dominated by robots. A typical factory contains hundreds of industrial robots working on fully automated production lines, with one robot for every ten human workers. On an automated production line, a vehicle chassis on a conveyor is welded, glued, painted and finally assembled at a sequence of robot stations.

Packaging:

Industrial robots are also used extensively for palletizing and packaging of manufactured goods, for example for rapidly taking drink cartons from the end of a conveyor belt and placing them into boxes, or for loading and unloading machining centers.

Electronics:

Mass-produced printed circuit boards (PCBs) are almost exclusively manufactured by pick-and-place robots, typically with SCARA manipulators, which remove tiny electronic components from strips or trays, and place them on to PCBs with great accuracy. Such robots can place hundreds of thousands of components per hour, far out-performing a human in speed, accuracy, and reliability.

Automated guided vehicles (AGVs)

Figure 1.3.2 Automated guided vehicles

An intelligent AGV drops-off goods without needing lines or beacons in the workspace

Mobile robots, following markers or wires in the floor, or using vision or lasers, are used to transport goods around large facilities, such as warehouses, container ports, or hospitals.

Early AGV-Style Robots

Limited to tasks that could be accurately defined and had to be performed the same way every time. Very little feedback or intelligence was required, and the robots needed only the most basic exteroceptors (sensors). The limitations of these AGVs are that their paths are not easily altered and they cannot alter their paths if obstacles block them. If one AGV breaks down, it may stop the entire operation.

Interim AGV-Technologies

Developed to deploy triangulation from beacons or bar code grids for scanning on the floor or ceiling. In most factories, triangulation systems tend to require moderate to high maintenance, such as daily cleaning of all beacons or bar codes. Also, if a tall pallet or large vehicle blocks beacons or a bar code is marred, AGVs may become lost. Often such AGVs are designed to be used in human-free environments.

Intelligent AGVs (i-AGVs)

Such as SmartLoader, SpeciMinder, ADAM, Tug Eskorta, and MT 400 with Motivity are designed for people-friendly workspaces. They navigate by recognizing natural features. 3D scanners or other means of sensing the environment in two or three dimensions help to eliminate cumulative errors in dead-reckoning calculations of the AGV's current position. Some AGVs can create maps of their environment using scanning lasers with simultaneous localization and mapping (SLAM) and use those maps to navigate in real time with other path planning and obstacle avoidance algorithms. They are able to operate in complex environments and perform non-repetitive and non-sequential tasks such as transporting photomasks in a semiconductor lab, specimens in hospitals and goods in warehouses. For dynamic areas, such as warehouses full of pallets, AGVs require additional strategies using three-dimensional sensors such as time-of-flight or stereovision cameras.

Research robots

While most robots today are installed in factories or homes, performing labour or life saving jobs, many new types of robot are being developed in laboratories around the world. Much of the research in robotics focuses not on specific industrial tasks, but on investigations into new types of robot, alternative ways to think about or design robots, and new ways to manufacture them. It is expected that these new types of robot will be able to solve real world problems when they are finally realized. [1]

Three Laws of Robotics

The Three Laws of Robotics (often shortened to The Three Laws or Three Laws) are a set of rules devised by the science fiction authorIsaac Asimov. The rules were introduced in his

1942 short story "Runaround", although they had been foreshadowed in a few earlier stories. The Three Laws are:

1. A robot may not injure a human being or, through inaction, allow a human being to come to harm.
2. A robot must obey the orders given to it by human beings, except where such orders would conflict with the First Law.
3. A robot must protect its own existence as long as such protection does not conflict with the First or Second Law.[1]

CHAPTER 2

2.0 ROBOTIC ARM

A robotic arm is a type of mechanical arm, usually programmable, with similar functions to a human arm; the arm may be the sum total of the mechanism or may be part of a more complex robot. The links of such a manipulator are connected by joints allowing either rotational motion (such as in an articulated robot) or translational (linear) displacement. The links of the manipulator can be considered to form a kinematic chain. The terminus of the kinematic chain of the manipulator is called the end effector and it is analogous to the human hand.

The end effector, or robotic hand, can be designed to perform any desired task such as welding, gripping, spinning etc., depending on the application. For example robot arms in automotive assembly lines perform a variety of tasks such as welding and parts rotation and placement during assembly. In some circumstances, close emulation of the human hand is desired, as in robots designed to conduct bomb disarmament and disposal.

Figure 2.1 KUKA

6-Axis Articulated Robots from KUKA

- Cartesian robot / Gantry robot: Used for pick and place work, application of sealant, assembly operations, handling machine tools and arc welding. It's a robot whose arm has three prismatic joints, whose axes are coincident with a Cartesian coordinator.
- Cylindrical robot: Used for assembly operations, handling at machine tools, spot welding, and handling at diecasting machines. It's a robot whose axes form a cylindrical coordinate system.
- Spherical robot / Polar robot (such as the Unimate): Used for handling at machine tools, spot welding, diecasting, fettling machines, gas welding and arc welding. It's a robot whose axes form a polar coordinate system.
- SCARA robot: Used for pick and place work, application of sealant, assembly operations and handling machine tools. This robot features two parallel rotary joints to provide compliance in a plane.
- Articulated robot: Used for assembly operations, diecasting, fettling machines, gas welding, arc welding and spray painting. It's a robot whose arm has at least three rotary joints.
- Parallel robot: One use is a mobile platform handling cockpit flight simulators. It's a robot whose arms have concurrent prismatic or rotary joints.
- Anthropomorphic robot: Similar to the robotic hand Luke Skywalker receives at the end of The Empire Strikes Back. It is shaped in a way that resembles a human hand, i.e. with independent fingers and thumbs.[2]

CHAPTER 3

3.0 MECHANİCAL DESİGN

In mechanical design I used plexiglass for body of robotic arm

Six degrees of freedom

Six degrees of freedom (6DoF) refers to the freedom of movement of a rigid body in three-dimensional space. Specifically, the body is free to move forward/backward, up/down, left/right (translation in three perpendicular axes) combined with rotation about three perpendicular axes, often termed pitch, yaw, and roll.

Robotics

Serial and parallel manipulator systems are generally designed to position an end-effector with six degrees of freedom, consisting of three in translation and three in orientation. This provides a direct relationship between actuator positions and the configuration of the manipulator defined by itsforward and inverse kinematics.

Robot arms are described by their degrees of freedom. This number typically refers to the number of single-axis rotational joints in the arm, where higher number indicates an increased flexibility in positioning a tool. This is a practical metric, in contrast to the abstract definition of degrees of freedom which measures the aggregate positioning capability of a system

In 2007, Dean Kamen, inventor of the Segway, unveiled a prototype robotic arm with 14 degrees of freedom for DARPA. Humanoid robots typically have 30 or more degrees of freedom, with six degrees of freedom per arm, five or six in each leg, and several more in torso and neck.

Engineering

The term is important in mechanical systems, especially biomechanical systems for analyzing and measuring properties of these types of systems that need to account for all six degrees of freedom. Measurement of the six degrees of freedom is accomplished today through both AC and DC magnetic or electromagnetic fields in sensors that transmit positional and angular data to a processing unit. The data are made relevant through software that integrate the data based on the needs and programming of the users.

Ascension Technology Corporation has recently created a 6DoF device small enough to fit in a biopsy needle, allowing physicians to better research at minute levels. The new sensor passively senses pulsed DC magnetic fields generated by either a cubic transmitter or a flat transmitter and is available for integration and manufacturability by medicalOEMs.

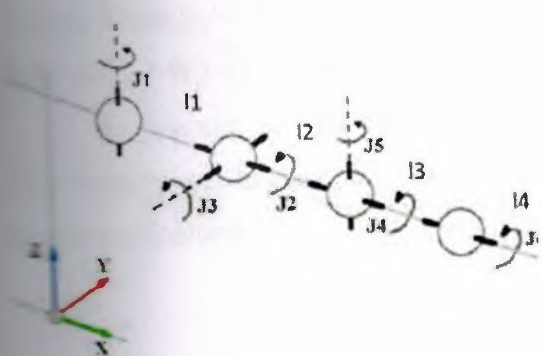
An example of six degree of freedom movement is the motion of a ship at sea. It is described as:

Translation:

1. Moving up and down (heaving);
2. Moving left and right (swaying);
3. Moving forward and backward (surging);

Rotation

1. Tilting forward and backward (pitching);
2. Turning left and right (yawing);
3. Tilting side to side (rolling).

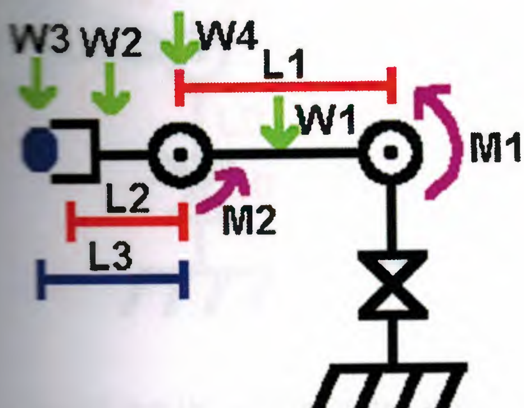


Force Calculations of Joints

This is where this tutorial starts getting heavy with math. Before even continuing, I strongly recommend you read the mechanical engineering tutorials for statics anddynamics. This will give you a fundamental understanding of moment armcalculations.

The point of doing force calculations is for motor selection. You must make sure that the motor you choose can not only support the weight of the robot arm, but also what the robot arm will carry (the blue ball in the image below).

The first step is to label your FBD, with the robot arm stretched out to its maximum length.



Choose these parameters:

weight of each linkage

weight of each joint

weight of object to lift

length of each linkage

Next you do a moment arm calculation, multiplying downward force times the linkage lengths. This calculation must be done for each lifting actuator. This particular design has just two DOF that requires lifting, and the center of mass of each linkage is assumed to be Length/2.

Torque About Joint 1:

M1 = L1/2 * W1 + L1 * W4 + (L1 + L2/2) * W2 + (L1 + L3) * W3

Torque About Joint 2:

M2 = L2/2 * W2 + L3 * W3

As you can see, for each DOF you add the math gets more complicated, and the joint weights get heavier. You will also see that shorter arm lengths allow for smaller torque requirements.

3.1 FORWARD KİNEMATİCS

Forward kinematics is the method for determining the orientation and position of the end effector, given the joint angles and link lengths of the robot arm. To calculate forward kinematics, all you need is highschool trig and algebra.

For our robot arm example, here we calculate end effector location with given joint angles and link lengths. To make visualization easier for you, I drew blue triangles and labeled the angles.

Assume that the base is located at x=0 and y=0. The first step would be to locate x and y of each joint.

Joint 0 (with x and y at base equaling 0):

x0 = 0

y0 = L0

Joint 1 (with x and y at J1 equaling 0):

cos(psi) = x1/L1 => x1 = L1*cos(psi)

sin(psi) = y1/L1 => y1 = L1*sin(psi)

Joint 2 (with x and y at J2 equaling 0):

sin(theta) = x2/L2 => x2 = *L2*sin(theta)*

cos(theta) = y2/L2 => y2 = L2*cos(theta)

End Effector Location (make sure your signs are correct):

x0 + x1 + x2, or 0 + L1*cos(psi) + L2*sin(theta)

y0 + y1 + y2, or L0 + L1*sin(psi) + L2*cos(theta)

z equals alpha, in cylindrical coordinates

The angle of the end effector, in this example, is equal to theta + psi.

Too lazy to calculate forward kinematics yourself? Check out my Robot Arm Designer v1 in excel.

3.2 INVERSE KINEMATIC

Inverse kinematics is the opposite of forward kinematics. This is when you have a desired end effector position, but need to know the joint angles required to achieve it. The robot sees a kitten and wants to grab it, what angles should each joint go to? Although way more useful than forward kinematics, this calculation is much more complicated too. As such, I will not show you how to derive the equation based on your robot arm configuration.

Instead, I will just give you the equations for our specific robot design:

psi = arccos((x^2 + y^2 - L1^2 - L2^2) / (2 * L1 * L2))

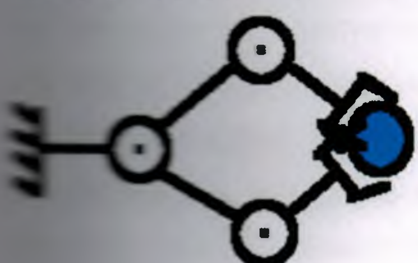
theta = arcsin((y * (L1 + L2 * c2) - x * L2 * s2) / (x^2 + y^2))

where c2 = (x^2 + y^2 - L1^2 - L2^2) / (2 * L1 * L2);

and s2 = sqrt(1 - c2^2);

So what makes inverse kinematics so hard? Well, other than the fact that it involvesnon-linear simultaneous equations, there are other reasons too.

First, there is the very likely possibility of multiple, sometimes infinite, number of solutions (as shown below). How would your arm choose which is optimal, based on torques, previous arm position, gripping angle, etc.?



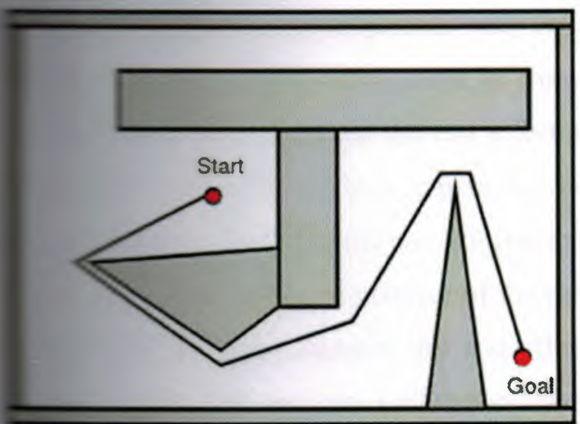
There is the possibility of zero solutions. Maybe the location is outside the workspace, or maybe the point within the workspace must be gripped at an impossible angle.

Singularities, a place of infinite acceleration, can blow up equations and/or leave motors lagging behind (motors cant achieve infinite acceleration).

And lastly, exponential equations take forever to calculate on a microcontroller. No point in having advanced equations on a processor that cant keep up.

Motion Planning

Motion planning on a robot arm is fairly complex so I will just give you the basics.



Suppose your robot arm has objects within its workspace, how does the arm move through the workspace to reach a certain point? To do this, assume your robot arm is just a simple mobile robot navigating in 3D space. The end effector will traverse the space just like a mobile robot, except now it must also make sure the other joints and links do not collide with anything too. This is extremely difficult to do . . .

What if you want your robot end effector to draw straight lines with a pencil? Getting it to go from point A to point B in a straight line is relatively simple to solve. What your robot should do, by using inverse kinematics, is go to many points between point A and point B. The final motion will come out as a smooth straight line. You can not only do this method with straight lines, but curved ones too. On expensive professional robotic arms all you need to do is program two points, and tell the robot how to go between the two points (straight line, fast as possible, etc.). For further reading, you could use the wavefront algorithm to plan this two point trajectory.

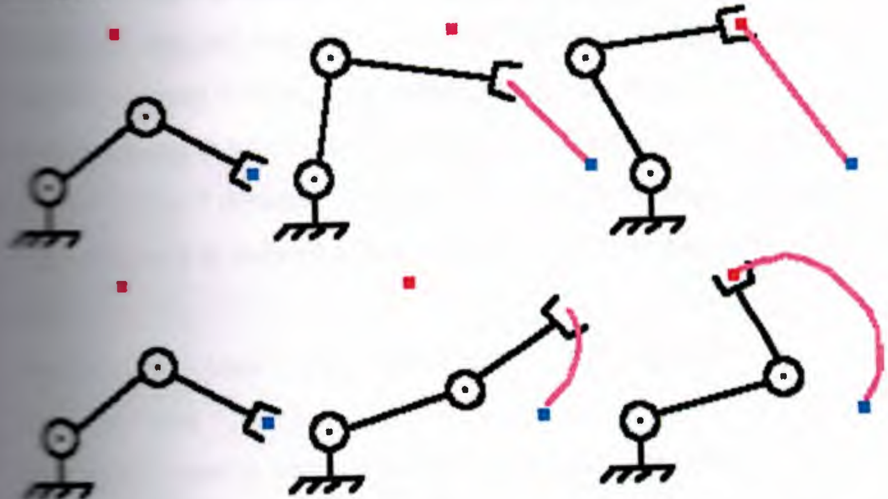
3.3 VELOCİTY (AND MORE MOTİON PLANNİNG

Calculating end effector velocity is mathematically complex, so I will go only into the basics. The simplest way to do it is assume your robot arm (held straight out) is a rotating wheel of L diameter. The joint rotates at Y rpm, so therefore the velocity is

Velocity of end effector on straight arm = 2 * pi * radius * rpm

However the end effector does not just rotate about the base, but can go in many directions. The end effector can follow a straight line, or curve, etc.

With robot arms, the quickest way between two points is often not a straight line. If two joints have two different motors, or carry different loads, then max velocity can vary between them. When you tell the end effector to go from one point to the next, you have two decisions. Have it follow a straight line between both points, or tell all the joints to go as fast as possible - leaving the end effector to possibly swing wildly between those points. In the image below the end effector of the robot arm is moving from the blue point to the red point. In the top example, the end effector travels a straight line. This is the only possible motion this arm can perform to travel a straight line. In the bottom example, the arm is told to get to the red point as fast as possible. Given many different trajectories, the arm goes the method that allows the joints to rotate the fastest.



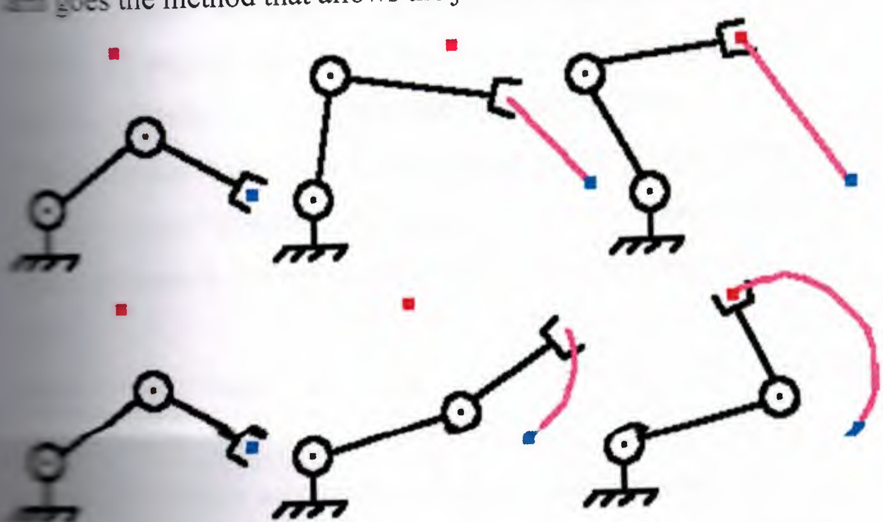
Which method is better? There are many deciding factors. Usually you want straight lines when the object the arm moves is really heavy, as it requires the momentum change for movement (momentum = mass * velocity). But for maximum speed (perhaps the arm isn't carrying anything, or just light objects) you would want maximum joint speeds.

Now suppose you want your robot arm to operate at a certain rotational velocity, how much torque would a joint need? First, lets go back to our FBD:

Velocity of end effector on straight arm = 2 * pi * radius * rpm

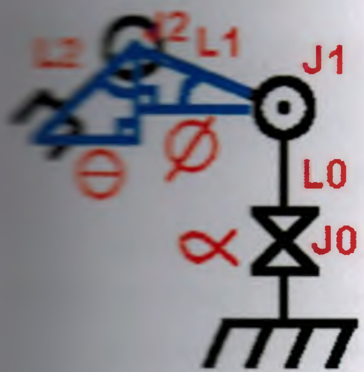
However the end effector does not just rotate about the base, but can go in many directions. The end effector can follow a straight line, or curve, etc.

With robot arms, the quickest way between two points is often not a straight line. If two joints have two different motors, or carry different loads, then max velocity can vary between them. When you tell the end effector to go from one point to the next, you have two decisions. Have it follow a straight line between both points, or tell all the joints to go as fast as possible - leaving the end effector to possibly swing wildly between those points. In the image below the end effector of the robot arm is moving from the blue point to the red point. In the top example, the end effector travels a straight line. This is the only possible motion this arm can perform to travel a straight line. In the bottom example, the arm is told to get to the red point as fast as possible. Given many different trajectories, the arm goes the method that allows the joints to rotate the fastest.



Which method is better? There are many deciding factors. Usually you want straight lines when the object the arm moves is really heavy, as it requires the momentum change for movement (momentum = mass * velocity). But for maximum speed (perhaps the arm isn't carrying anything, or just light objects) you would want maximum joint speeds.

Now suppose you want your robot arm to operate at a certain rotational velocity, how much torque would a joint need? First, lets go back to our FBD:



Now lets suppose you want joint J0 to rotate 180 degrees in under 2 seconds, what torque does the J0 motor need? Well, J0 is not affected by gravity, so all we need to consider is momentum and inertia. Putting this in equation form we get this:

torque = moment_of_inertia * angular_acceleration

breaking that equation into sub components we get:

torque = (mass * distance^2) * (change_in_angular_velocity / change_in_time) and

change_in_angular_velocity = (angular_velocity1)-(angular_velocity0)

angular_velocity = change_in_angle / change_in_time

Now assuming at start time 0 that angular_velocity0 is zero, we get

torque = (mass * distance^2) * (angular_velocity / change_in_time)

where distance is defined as the distance from the rotation axis to the center of mass of the arm:

center of mass of the arm = distance = 1/2 * (arm_length) (use arm mass)

but you also need to account for the object your arm holds:

center of mass of the object = distance = arm_length (use object mass)

So then calculate torque for both the arm and then again for the object, then add the two torques together for the total:

torque(of_object) + torque(of_arm) = torque(for_motor)

And of course, if J0 was additionally affected by gravity, add the torque required to lift the arm to the torque required to reach the velocity you need. To avoid doing this by hand, just use the robot arm calculator.[3]

CHAPTER 4

4.0 ELECTRONIC DESIGN

I design my circuit to control robotic arm via wifi module. There is 2 circuits of my robotic arm. One of these circuits is remoted control board and other one is main board.

In remoted control board we have 2 joysticks, 1 atmega328p and 1 xbee wifi module. In main board we have servomotors ,ATmega328p and xbee wifi module.

The main logic of the system is this: there is a potentiometer in the joystick and when we move the joystick, the ATmega328p reads value from analogue pins. Then this analog value is sending by xbee wifi module .the other xbee takes datas and give the ATmega328p .ATmega328p is convert analogue value in to the digital signal and move the servos.

I used 7 servo motors but every servos different from each other.

4.1 ATMEGA 328p

Figure 4.1.1 ATmega328p

The ATmega328 is a single chip micro-controller created by Atmel and belongs to the megaAVR series.

Specifications

The high-performance Atmel 8-bit AVR RISC-based microcontroller combines 32 KB ISP flash memory with read-while-write capabilities, 1 KB EEPROM, 2 KB SRAM, 23 general purpose I/O lines, 32 general purpose working registers, three flexible timer/counters with compare modes, internal and external interrupts, serial programmable USART, a byte-oriented 2-wire serial interface, SPI serial port, 6-channel 10-bit A/D converter (8-channels in TQFP and QFN/MLF packages), programmable watchdog timer with internaloscillator, and five software selectable power saving modes. The device operates between 1.8-5.5 volts. By executing powerful instructions in a single clock cycle, the device achieves throughputs approaching 1 MIPS per MHz, balancing power consumption and processing speed.

PARAMETERS	VALUE
Flash	32 Kbytes
RAM	2 Kbytes
Pin Count	28
Max. Operating Frequency	20 MHz
CPU	8-bit AVR
# of Touch Channels	16
Hardware QTouch Acquisition	No
Max I/O Pins	26
Ext Interrupts	24
USB Interface	No
USB Speed	No

Table 4.1.1 ATmega specifications

Series alternatives

A common alternative to the ATmega328 is the ATmega328P. A comprehensive list of all other member of the megaAVR series can be found here

Applications

Today the ATmega328 is commonly used in many projects and autonomous systems where a simple, low-powered, low-cost micro-controller is needed. Perhaps the most common implementationof this chip is on the ever popular Arduino development platform, namely the Arduino Uno and Arduino Nano models.

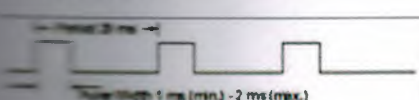
Servo Motor

Servo and receiver connections Servo control from a radio control receiver to the servos is done by sending each servo a PWM (pulse width modulation) signal, a series of repeating pulses of variable width.

Small radio control servos are connected through a standard three-wire connection: two wires for a DC power supply and one for control, carrying the pulses.

The parameters for this pulse are that it has a minimum pulse, a maximum pulse, and a repetition rate. Given the rotation constraints of the servo, neutral is defined to be the position where the servo has exactly the same amount of potential rotation in the clockwise direction as it does in the counter clockwise direction. It is important to note that different servos will have different constraints on their rotation but they all have a neutral position, and that position is always around 1.5 milliseconds (ms).

Pulse duration



The angle is determined by the duration of a pulse that is applied to the control wire. This is a form of pulse-width modulation, however servo position is not defined by the PWM duty cycle (i.e., ON vs OFF time) but only by the duration of the pulse. The servo expects to see a pulse every 20 ms, however this can vary within a wide range that differs from servo to servo. The length of the pulse will determine how far the motor turns. For example, a 1.5 ms pulse will make the motor turn to the 90 degree position (neutral position).

The "RC PWM" used in RC servo control acts very differently from the PWM used in any other system. (In particular, the confusingly similar-sounding direct PWM DC motor speed control works entirely differently). Most RC servos move to exactly the same position when they receive a 1.5 ms pulse every 6 ms (a duty cycle of 25%) as when they receive a 1.5 ms pulse every 25 ms (a duty cycle of 6%) -- in both cases, they turn to the center position (neutral position). The low time (and the total period) can vary over a wide range, and vary from one pulse to the next, without any effect on the position of the servo motor. With many RC servos, as long as the "frame rate" (how many times per second the pulse is sent, aka the pulse repetition rate) is in a range of 40 Hz to 200 Hz, the exact value of the frame rate is irrelevant. Most RC receivers send pulses to the RC servo at some constant frame rate, changing only the high time. However, it is possible to command an RC servo to move over its entire range with a function generator set to a constant 10% duty cycle by changing only the frequency (frame rate).

Force

When these servos are commanded to move they will move to the position and hold that position. If an external force pushes against the servo while the servo is holding a position, the servo will resist from moving out of that position. The maximum amount of force the servo can exert is the torque rating of the servo. Servos will not hold their position forever though; the position pulse must be repeated to instruct the servo to stay in position.

Variations

When a pulse is sent to a servo that is less than 1.5 ms the servo rotates to a position and holds its output shaft some number of degrees counterclockwise from the neutral point.

When the pulse is wider than 1.5 ms the opposite occurs. The minimal width and the maximum width of pulse that will command the servo to turn to a valid position are functions of each servo. Different brands, and even different servos of the same brand, will have different maximum and minimums. Generally the minimum pulse will be about 1 ms wide and the maximum pulse will be 2 ms wide.[4]

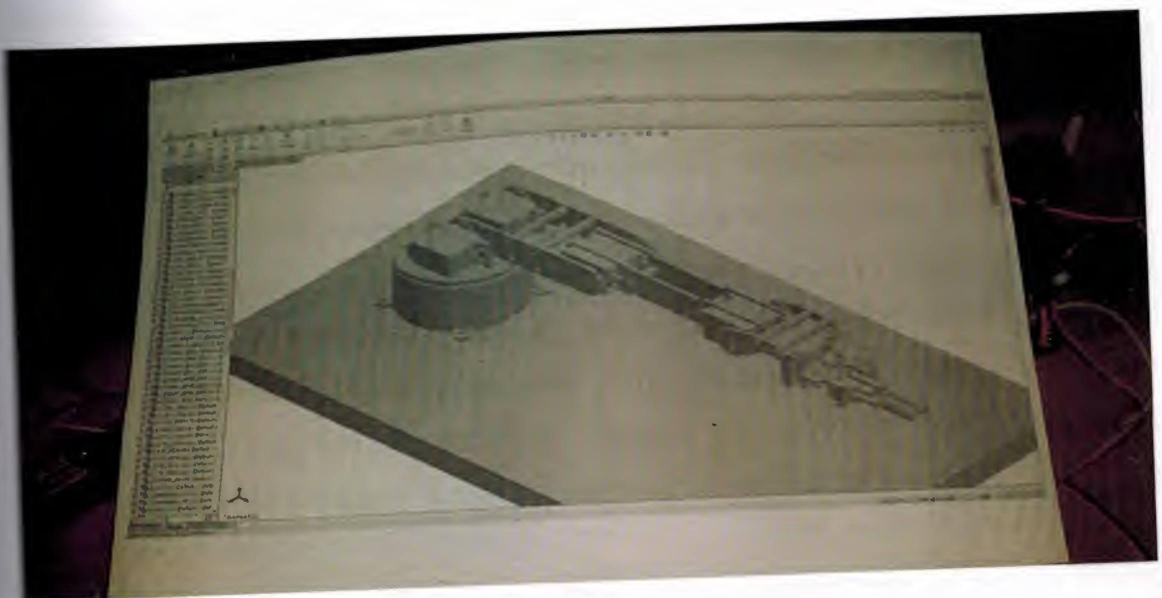


Figure 4.1.2 Robotic arm design

Figure 4.1.3 ServoMotor

4.2 XBEE

XBee is the brand name from Digi International for a family of form factor compatible radio modules. The first XBee radios were introduced under the MaxStream brand in 2005 and were based on the 802.15.4-2003 standard designed for point-to-point and star communications at over-the-air baud rates of 250 kbit/s.

Two models were initially introduced—a lower cost 1 mW XBee and the higher power 100 mW XBee-PRO. Since the initial introduction, a number of new XBee radios have been introduced and all XBees are now marketed and sold under the Digi brand.

The XBee radios can all be used with the minimum four number of connections – power (3.3 V), ground, data in and data out (UART), with other recommended lines being Reset and Sleep. Additionally, most XBee families have some other flow control, I/O, A/D and indicator lines built in. A version of the XBees called the programmable XBee has an additional onboard processor for user's code. The programmable XBee and a new surface mount (SMT) version of the XBee radios were both introduced in 2010.

XBee Modules are available in two form-factors; Through-Hole and Surface Mount. All XBees (with the exception of the XBee 868LP) are available in the popular 20-pin Through-Hole form-factor. Certain XBee modules are also available in a 37-pad Surface Mount design, which is popular for higher volume applications due to the reduced manufacturing costs of SMT technology.

XBee Modules typically come with several antenna options, including U.FL, PCB Embedded, Wire, and RPSMA.

The XBees can operate either in a transparent data mode or in a packet-based application programming interface (API) mode. In the transparent mode, data coming into the Data IN (DIN) pin is directly transmitted over-the-air to the intended receiving radios without any modification. Incoming packets can either be directly addressed to one target (point-to-point) or broadcast to multiple targets (star). This mode is primarily used in instances where an existing protocol cannot tolerate changes to the data format. AT commands are used to control the radio's settings. In API mode the data is wrapped in a packet structure that allows for addressing, parameter setting and packet delivery feedback, including remote sensing and control of digital I/O and analog input pins.[5]

Figure 4.2.1 XBee

4.3 2-AXIS JOYSTICK

I used 2-axis joystick for remote controlled. There are 2 potantiometer and 1 button inside of the joystick. When i rotate the joystick , the resistance change. İ connected the potantiometer to atmega328's analogue pins .analogue pins are reading the valua of potantiometer and send information to other atmega.

Figure 4.3.1 Joystick

CHAPTER 5

5.0 PROGRAMMABLE DESIGN

In this section i took help from my computer engineer friends becouse its very hard to write new program. I found some programs from internet and we develope it for my project.

For program the ATmega328p , it has a device that we call a Arduino. Arduino is a single-board microcontroller, intended to make the application of interactive objects or environments more accessible. The hardware consists of an open-source hardware board designed around an 8-bit AtmelAVR microcontroller, or a 32-bit Atmel ARM. Current models feature a USBinterface, 6 analog input pins, as well as 14 digital I/O pins which allows the user to attach various extension boards.

Arduino has own program for computers and we program ATMEGA328p from this computer program. It is very easy to program.we click the ok button for compiling program. For install we click the → button.

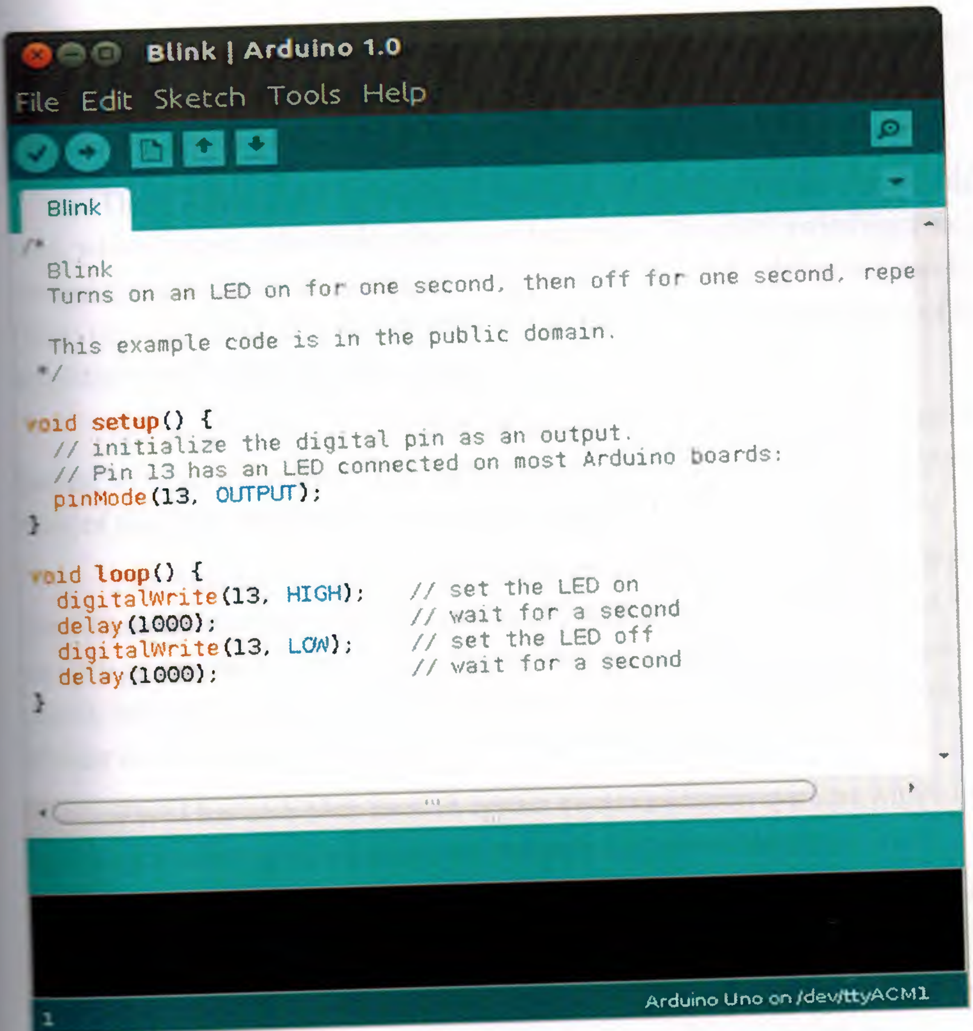


Figure 5.0.1 Arduino

5.1 PWM

Pulse-width modulation (PWM), or pulse-duration modulation (PDM), is a modulation technique that conforms the width of the pulse, formally the pulse duration, based on modulator signal information. Although this modulation technique can be used to encode information for transmission, its main use is to allow the control of the power supplied to electrical devices, especially to inertial loads such as motors. In addition, PWM is one of the two principal algorithms used in photovoltaic solar battery chargers,[1] the other being MPPT.

The average value of voltage (and current) fed to the load is controlled by turning tch between supply and load on and off at a fast pace. The longer the switch is on red to the off periods, the higher the power supplied to the load is.

The PWM switching frequency has to be much faster than what would affect the which is to say the device that uses the power. Typically switchings have to be done l times a minute in an electric stove, 120 Hz in a lamp dimmer, from few kilohertz to tens of kHz for a motor drive and well into the tens or hundreds of kHz in audio fiers and computer power supplies.

The term *duty cycle* describes the proportion of 'on' time to the regular interval or d' of time; a low duty cycle corresponds to low power, because the power is off for of the time. Duty cycle is expressed in percent, 100% being fully on.

The main advantage of PWM is that power loss in the switching devices is very When a switch is off there is practically no current, and when it is on, there is almost oltage drop across the switch. Power loss, being the product of voltage and current, is in both cases close to zero. PWM also works well with digital controls, which, because eir on/off nature, can easily set the needed duty cycle.

PWM has also been used in certain communication systems where its duty cycle has n used to convey information over a communications channel.

Pulse-width modulation uses a rectangular pulse wave whose pulse width is dulated resulting in the variation of theaverage value of the waveform. If we consider a lse waveform $f(t)$, with period T, low value y_{min}, a high value y_{max} and a duty cle D (see figure 1), the average value of the waveform is given by:

$$\bar{y} = \frac{1}{T}\int_0^T f(t)\,dt.$$

As $f(t)$ is a pulse wave, its value is y_{max} for $0 < t < D \cdot T$ and y_{min} for $D \cdot T < t < T$. The above expression then becomes:

$$\begin{aligned}\bar{y} &= \frac{1}{T}\left(\int_0^{DT} y_{max}\,dt + \int_{DT}^{T} y_{min}\,dt\right)\\ &= \frac{D \cdot T \cdot y_{max} + T\,(1-D)\,y_{min}}{T}\\ &= D \cdot y_{max} + (1-D)\,y_{min}.\end{aligned}$$

This latter expression can be fairly simplified in many cases where $y_{min} = 0$ as $\bar{y} = D \cdot y_{max}$. From this, it is obvious that the average value of the signal ($\bar{y}$) is directly dependent on the duty cycle D.

The simplest way to generate a PWM signal is the intersective method, which requires only a sawtooth or atriangle waveform (easily generated using a simple oscillator) and a comparator. When the value of the reference signal (the red sine wave in figure 2) is more than the modulation waveform (blue), the PWM signal (magenta) is in the high state, otherwise it is in the low stat

6.0 PICTURES

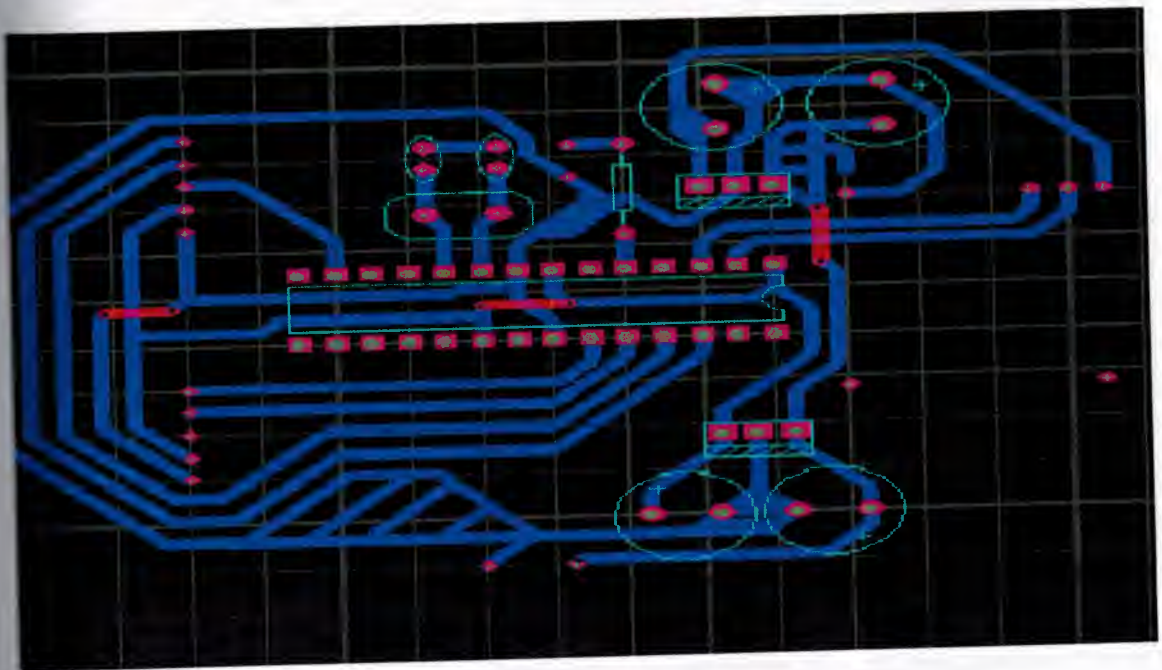


Figure 6.1 Transmitter circuit proteus design

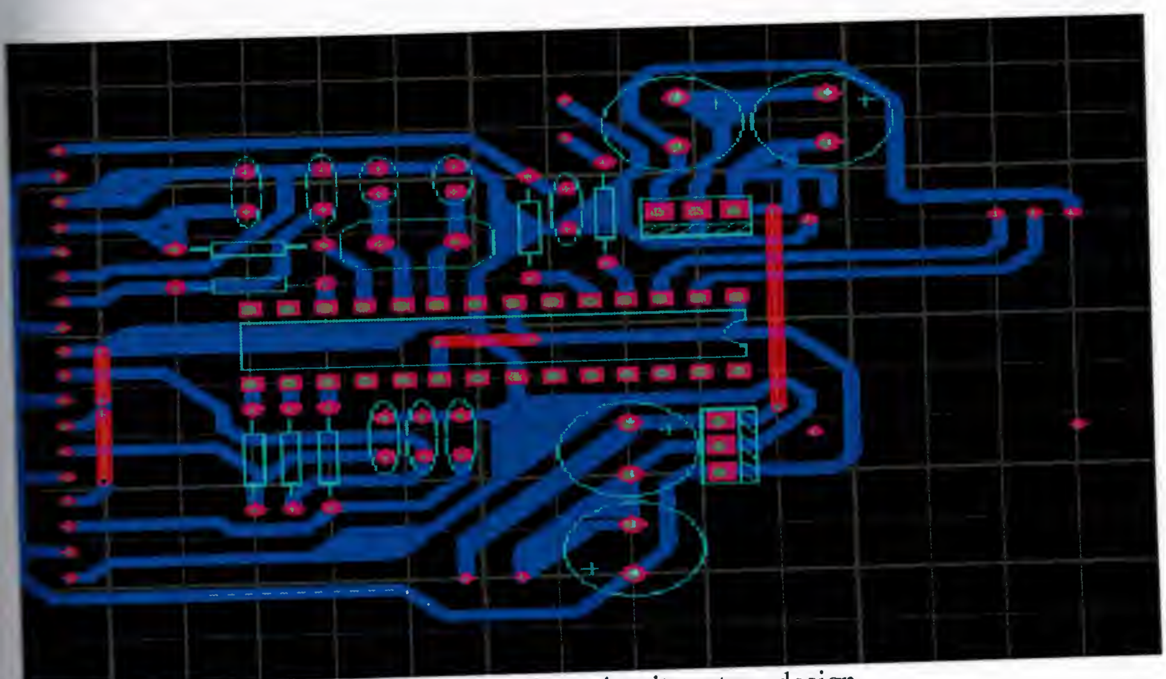


Figure 6.2 Receiver circuit proteus design

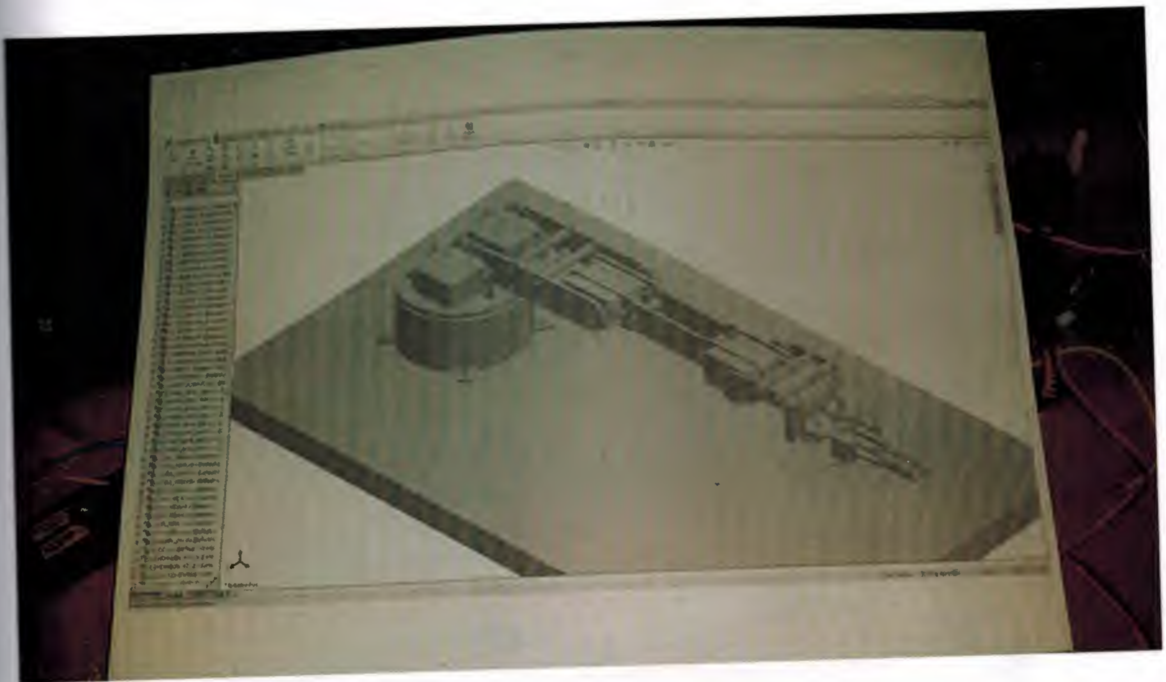


Figure 6.3 Robotic arm mechanical design

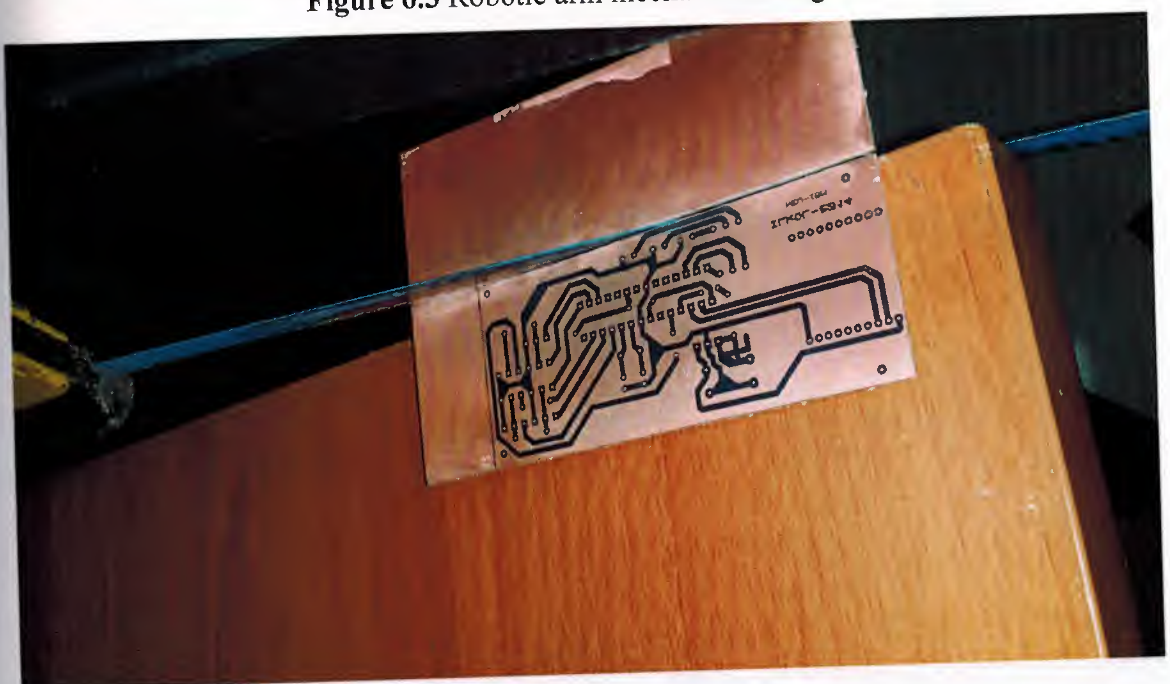


Figure 6.4 make of circuit

Figure 6.5 Robotic arm constraction

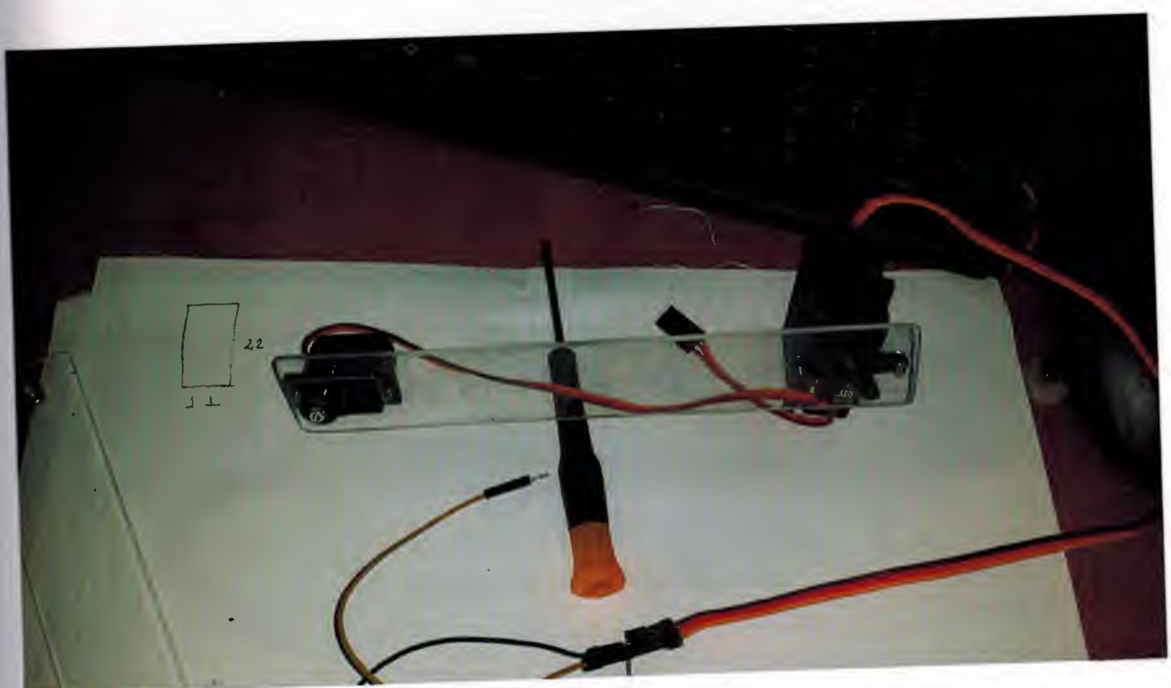


Figure 6.6 Robotic arm constraction

Figure 6.7 Hands

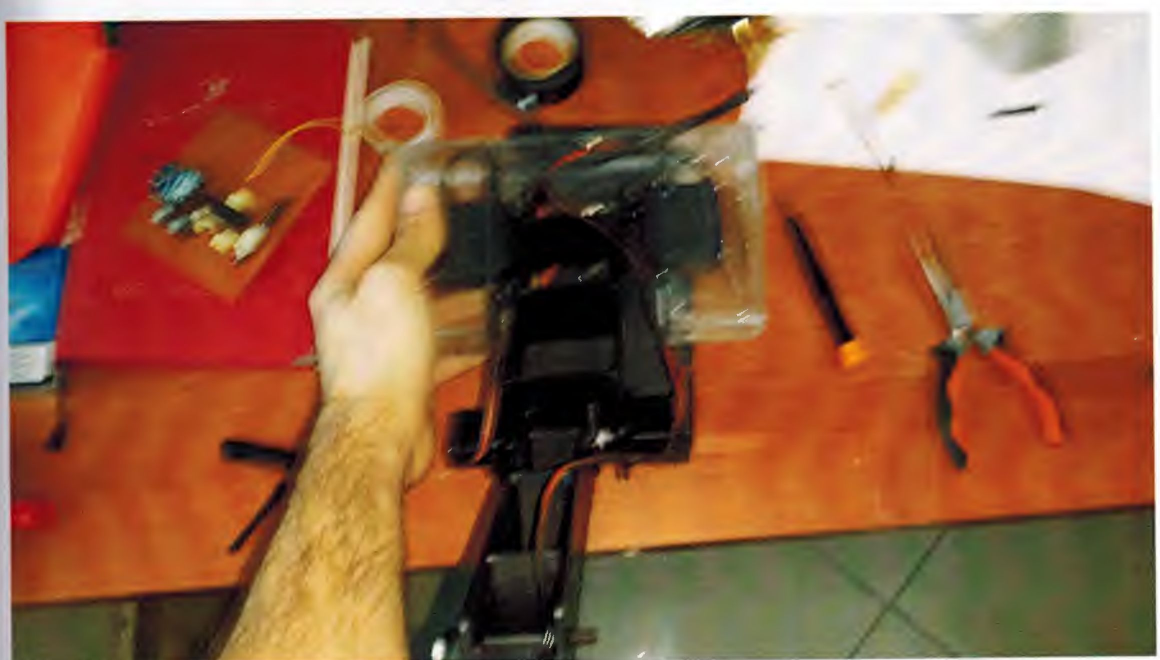


Figure 6.8 Under constraction

Figure 6.9 Robotic arm

Figure 6.10 Robotic arm

7.0 CONCLUSION

Technology is very important thing of our life. People are researching for make human life easier and robots will be very important position of our life in the future. They will make you whatever we want. And it will be very beneficial for human.

İn this project i encounter some problems.some of these problems are the wheel of the servo.they can be break down easily so we should to choice true servomotor. And other problem is the program of the controlling via joystick.but i solve all problems and my robotic arm is working nice.

There are a lot of advantage of this project to me.becouse i made my own circuit. I experienced that. I learned a lot of things from this project.

But ofcourse there are some disadvantages of this project .some of these:

-cost reason: xbee wifi modules and servo motors are very expensive.

-supply market:i couldn't find any store to but these elements from cyprus .i brought them from the Turkey.

- It was difficult project because it need harmony of electronic engineering, computer engineering and mechanical engineering.

I will continue to develop this project. I will connect camera and i will repleace on the car so it will mobile robotic hand.

8.0 APPENDIX

Programming Atmega328p:

Transmitter program is:

```
#include <EasyTransfer.h>
EasyTransfer ET;

const int potpin1 = A0;
const int potpin2 = A1;
const int potpin3 = A2;
const int potpin4 = A3;

struct SEND_DATA_STRUCTURE{
int servo1val;
int servo2val;
int servo3val;
int servo4val;


};

SEND_DATA_STRUCTURE txdata;

void setup(){
  Serial.begin(9600);
  //  Serial.begin(115200);
  ET.begin(details(txdata), &Serial);
  // pinMode(potpin1, INPUT);
  // pinMode(potpin2, INPUT);

}
```

```
loop(){

val1 = analogRead(potpin1);
val2 = analogRead(potpin2);
t val3 = analogRead(potpin3);
t val4 = analogRead(potpin4);

al1 = map(val1, 0, 1023, 0, 179);
al2 = map(val2, 0, 1023, 0, 179);
al3 = map(val3, 0, 1023, 0, 179);
al4 = map(val4, 0, 1023, 0, 179);
xdata.servo1val = val1;
xdata.servo2val = val2;
xdata.servo3val = val3;
xdata.servo4val = val4;

ET.sendData();
```

The program of the receiver is:

```
#include <Servo.h>
#include <EasyTransfer.h>
EasyTransfer ET;

Servo myservo1;
Servo myservo2;
Servo myservo3;
```

```
servo4;
1, ST2, ST3, ST4;
CEIVE_DATA_STRUCTURE{
o1val;
o2val;
o3val;
o4val;

E_DATA_STRUCTURE txdata;

up(){
begin(9600);
l.begin(115200);

gin(details(txdata), &Serial);
o
tach( 6, 1000, 2000);
tach(5, 1000, 2000);
tach(3, 1000, 2000);
tach(9, 1000, 2000);

op(){
receiveData()){
write(txdata.servo1val);
write(txdata.servo2val);
```

```
    ST3.write(txdata.servo3val);
    ST4.write(txdata.servo4val);
    //ST1.write(map(txdata.servo1val, 0, 1023, 0, 180));
    //ST2.write(map(txdata.servo2val, 0, 1023, 0, 180));
  }
}
```

9.0 REFERENCES

[1] http://en.wikipedia.org/wiki/Robot

[2]http://en.wikipedia.org/wiki/Robotic_arm

[3]http://www.societyofrobots.com/robot_arm_calculator.shtml

[4]http://en.wikipedia.org/wiki/ATmega328

[5]http://en.wikipedia.org/wiki/XBee