Helping Hand

Senior Design Final Documentation

Group 10 – Fall 2012

Kurt Graf

Taylor Jones

Matt Carlson

Eric Donley

12/6/2012

Table of Contents

Appendix A Copyright Permissions

Appendix B Datasheets

Appendix C Software

Chapter 1 Executive Summary

The purpose of the senior design project, from the course description of objectives is "To provide students a complete design experience, including the necessity to set design goals and objectives, integrate knowledge, exercise engineering judgment, plan to meet a budget and schedule, to work as a team member, and to communicate in writing." To accomplish these objectives our group, Helping Hand, is designing a teleoperated master-slave robot arm which motion tracks a human operator's arm motion including open-close hand tracking by the end-effector. The project is exclusively focused on the electronics and software to control an electrically operated robotic arm. Stock mechanical robot arms will be utilized as needed to accomplish the project.

1. Why study the human operated robot arm?

The future of robotics in manufacturing and assembly is increased flexibility both in mechanical performance and ubiquitous integration with human workers. The future of robotics is greater dexterity, easier and quicker program-ability, and safe operation with human co-workers. Building a tele-operated master-slave robot arm driven by sensors worn on a human arm is investigating future possibilities and general performance considerations of advanced robotics. See INTRODUCTION Unimate to Baxter.

2. Initial feasibility investigations of the project included web searches for similar projects and several were found on youtube and are illustrated in the section SIMILAR PROJECTS. Other projects reviewed more closely included previous EEL4914 projects from 2006, DUCSS a DARPA autonomous vehicle sensor system, and a 2009 project Luke's Hand, an attempt at a human prosthetic robotic hand. From searching robot websites and robot shop sales/forums sites, an initial concept of the design was determined: MEMS gyro sensors and a force sensitive resistor would provide the sensing of the human operator's arm and hand motions, an arduino-type microcontroller would process the sensor data and generate signals through a motor driver subsystem to drive the robot arm motors.

3. Because Robotics is virtually a graduate school field of study, a section on GENERAL RESEARCH ON ROBOTICS (as relates to our project) was added to connect our project to the more general knowledge base of robotics.

4. The core design of the project is the control board explained in the section Design Architecture and Related Diagrams. Fulfilling the project requirement for

at least one custom designed printed circuit board, the control board design must accomplish the control requirements of the project and

• Contain enough I/O pins to monitor all the sensors and send data to the motors.

• Have enough memory to store the control software and data gathered from the sensors.

• Be fast enough to only have a slight delay between the operator’s motions and the robot’s motions.

5. Performance specifications for the project include the robot arm's dynamic range (see 5.1.1 Robot Arm Kinematics showing mechanical views of the OWI arm) “gripper to open and close, wrist motion of 120 degrees, an extensive elbow range of 300 degrees, base rotation of 270 degrees, base motion of 180 degrees, vertical reach of 15 inches, horizontal reach of 12.6 inches, and lifting capacity of 100g

SPECIFICATIONS

Assembled Size: 9in x 6.3in x 15in

Unit Weight: 658g

Battery: 'D' size x4”

Definitive response times for the robot arm's motion have not been determined because of the project dependency on the mechanical arm actually used for final presentation. Nominally, motion-tracking of the human-operator's arm should follow within tenths of a second or better as has been observed in a youtube video *Haptic robotic arm 1* see Fig. 3.1.3 in section SIMILAR PROJECTS. Because of the slow response limitations of our initial plastic mechanical arm with small dc motors we will target a 1 second response time of the robot arm following the human operator's arm and then try to cut that to a tenth of a second or better when switching to the metal robot arm with servo motors.

6. A first milestone was achieved Nov. 4 with the successful build of the robot arm OWI 535 EDGE (see Fig. 4.2.1) which is made from all plastic parts except for the motors and self-tapping fastener screws and has a simple manual controller arrangement - double throw switches that are manually thrown to rotate clockwise and counterclockwise the five dc motors of the 535 arm. Female jumper pin wires from each dc motor are easily accessible to connect our own motor drive system. An initial budget estimate of $300 will probably have to be raised to at least $400 by the completion of the project. The OWI arm only cost $40, but it really only affords proof-of-concept of our electronic control systems. For presentation level of performance, we will probably upgrade (see section

HARDWARE UPGRADE OPTIONS) to a Lynx metal arm for $130 with five servo motors costing an additional $90 to $130 (depending on size of arm selected). Another cost overrun could be a necessity to switch to 6 DOF sensors gyroscopes/accelerometers) about $50 each instead of gyroscopic-only 3 DOF sensors at $25 each.

7. The build plan for the project has already started, as mentioned above, with the successful build of the plastic OWI robotic arm. Final electronic parts selection is derived from the control board design and component acquisition will begin in late December and early January. Details for building the Helping Hand is are in the section Chapter 7 Build Plan. A list of components already acquired is also in this section.

8. Testing of the Helping Hand will occur simultaneously while building to test individual components and then interval subsystem tests will be done to determine successful performance. A table of testing is in the section 8.0 TESTING CHECKLIST.

9. Funding for the Helping Hand is from the members of the design project and a general target is to not exceed the price of four college textbooks that do not have to be purchased for EEL4915 – about $400. Milestones beyond an initial working mechanical arm and budget details are outlined in the section ADMINISTRATIVE CONTENT.

10 Attempts were made to contact any content cut and pasted from the web, but some sources on youtube have no contact information. See Appendix A.

11. An additional section to be included in the final report on the project in eel4915 is SYSTEMS AND SIGNALS that will detail our implemented hardware and software control systems and signal analysis with as much quantitative detail as possible.

Chapter 2 Project Description

Section 2.1 Motivation and Goals

Motivation for Project

As candidates for bachelor of science diplomas in Electrical and Computer Engineering from the University of Central Florida Spring 2013, concern for real working world knowledge and skills led to the choice to design a modern application of an industrial standard robotic application - the robotic arm.

Our application is a more sophisticated technologically than the typical robotic arms like those available from consumer sources. The typical robot arm is basically an open-loop control scenario dominated by pre-programmed point-to-point motion and a fixed absolute geometry reference system. The Helping Hand is a much more spontaneous and fluid motion device looking towards a future of ubiquitous and easily programmed human-type robotics. The Helping Hand will use a closed-loop feedback system in a master-slave arrangement between a human operator's arm motion and the Helping Hands' motion-mirroring response (tracking from motion and force sensors on the human operator's arms and hand).

To learn about robotics, particularly with regard to how our Electrical and Computer Engineering undergraduate education relates to the working robotic world, is also a significant factor in our choice of senior design project. This is a very technologically challenging project; the other similar projects reviewed while researching this project seem to have fewer degrees of freedom and less overall difficulty.

The number one project goal is to create a motion-tracking system of a custom or stock robot arm that can proportionally replicate a human operator's arm motion. The initial prototype will be wired directly between sensors on the operator's arm and the controller on the Helping Hand. The Helping Hand itself is operated by control signals from the controller to motors on the robot arm controlling direction and magnitude of subsequent motion. The whole interface should be relatively easy to use and should require little assistance to set up and begin use.

Depending on how fast we can implement the basic function of the Helping Hand in the second term of senior design project, we can add features such as wireless signal transmission from the operator's arm sensors to the controller and more complex motions - add rotation to the wrist link, distanced remote operation, etc.

The overall goal is fast, accurate and stable motion tracking of the human-operator's arm by the robot arm. An exact response time is difficult to specify at this time because of the designers' unfamiliarity with the specific hardware devices and software requirements; however, research indicates that quite fast and accurate proportional motion response is attainable. Because of the slow response limitations of our initial plastic mechanical arm with small dc motors we will target a 1 second response time of the robot arm following the human operator's arm and then try to cut that to a tenth of a second or better when switching to the metal robot arm with servo motors.

The grasp and lift capability of the robot arm is a predefined feature of the mechanical robot arm employed with our electronics. A lift capability of close to 400 grams has been decided on for this project. While upgrades to the robot's motors could be applied to increase this factor a robot arm already capable of lifting a payload close to the goal would be preferred.

Section 2.2 Project Requirements and Specifications

The robot arm is going to mirror the motions of the motions made by the user's arm and therefore must meet certain requirements. These requirements are to ensure that the whole project meets as many of the goals set as possible. The end result should be an easy to use interface that produces fast accurate movement from the robot.

In order to accurately portray a human arm the robot should have at least six degrees of freedom, a rotating and bending base, a bending elbow, a wrist that bends and rotates, and an end effector capable of opening and closing. The importance of having this many degrees of freedom is for the robot to be able to replicate every movement that is possible for a human arm to do. This will result in the robot having six motors that will need to be controlled by the control board.

In order for the operator's movements to be properly read sensors will be required to be worn on their arm. A total of five sensors will be necessary in order to properly read the movements of the operators arm. These sensors must be able to detect either movement on multiple axis or rotation depending on their location. The elbow sensor and a wrist sensor will need to be able to detect rotation as they will correspond to joints that bend. The forearm sensor and a second wrist sensor will need to detect movement along several axis in order to control motors that spin.

In order to control the motors and interpret the feedback from the sensors a control board is needed. This control board will be responsible for reading the data sent from the sensors, interpreting it and using it to send signals to the motors telling them what position to move to. The control board must be able to connect to a computer through a USB connector and have room to handle all of the inputs from the sensors and outputs to the motors. The control board must also be able to perform its functions once detached from the computer.

Chapter 3 Research Related to Project Definition

Section 3.0 Introduction

A Quick History of the Robot Arm from Unimate to Baxter

The robotic arm began industrially in 1961 with the Unimate: *Without any fanfare, the world's first working robot joined the assembly line at the General Motors plant in Ewing Township in the spring of 1961. It was nothing like the metallic humanoid robots seen in movies and on television in those days in America and Japan.* [1]

An historical aside, *robotics pioneer George C. Devol, who invented the prototype of robotic arms, has died in the US aged 99.* [18 August, 2011] *The American inventor came up with the concept of Universal Automation, which led to Unimate, the first industrial robot. This was a mechanical arm which was programmable, allowing it to repeat tasks like grasping and lifting. The first Unimate prototypes were controlled by vacuum tubes. Transistors were used later as they became available. A rotating drum memory system with data parity controls was invented for the Unimate. The Unimate arm was adopted by General Motors in 1961, followed by other auto-makers Chrysler and Ford. It was first used to lift and stack die-cast metal parts from moulds, and its role then expanded to other duties like welding and painting. The rest, as they say, is industrial robotics history.* [2]

UNIMATE, the first industrial robot, began work at General Motors. Obeying step-by-step commands stored on a magnetic drum, the 4,000-pound arm sequenced and stacked hot pieces of die-cast metal. The brainchild of Joe Engelberger and George Devol, UNIMATE originally automated the manufacture of TV picture tubes. [3]

A picture of the first Unimate [4]:

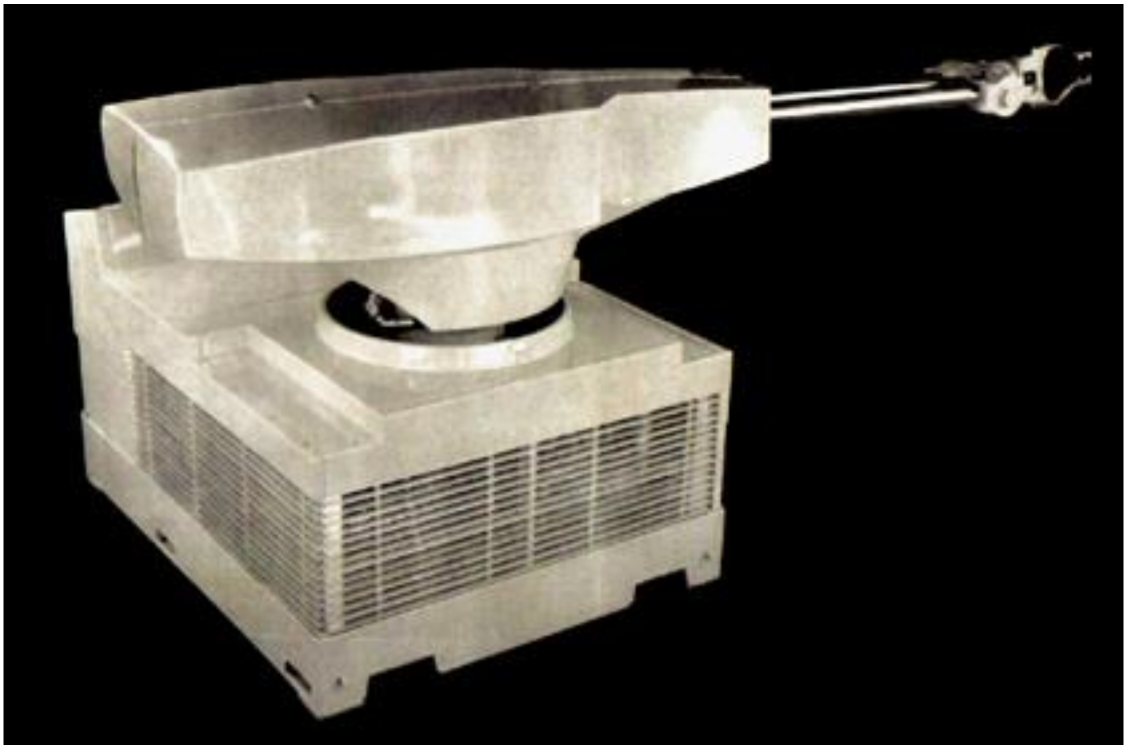
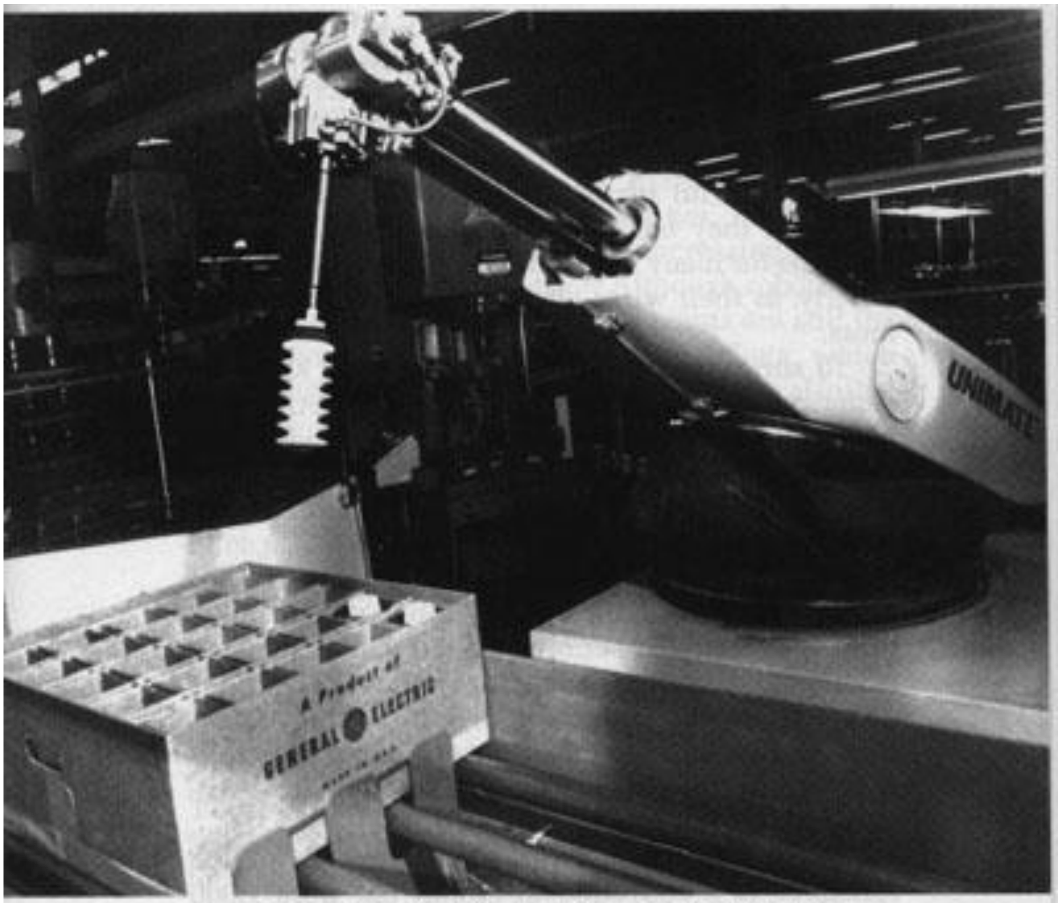


Fig. 3.0.1

A picture of the first Unimate at work [5]:



Armed for duty. A Unimate robot—really, just an arm—picks up and puts down parts in a General Electric factory.

Fig. 3.0.2

It was an automated die-casting mold that dropped red-hot door handles and other such car parts into pools of cooling liquid on a line that moved them along to workers for trimming and buffing. Its most distinct feature was a grip on a steel armature that eliminated the need for a man to touch car parts just made from molten steel. [6]

To jump to the present and take a look at the current state of industrial robotics and the new leading edge take a look at Baxter:

rethink productivity

Baxter, our flagship product, is America's first adaptive manufacturing robot. Designed to use common sense and work safely alongside people, it'll take your productivity to new heights—and deliver a true competitive edge."

rethink affordability

A game-changing price point. Trainable by non-technical personnel, with no costly programming needed. And flexible and adaptable enough to change quickly across tasks, lines and teams. [7]

THE BAXTER: [8]

Fig. 3.0.3

BAXTER on simulated assembly line: [8]

Fig. 3.0.4

BAXTER claims to be quite versatile: [8]

Fig. 3.0.5

Interestingly, Baxter uses a Unix-based OS and open source licensing, presumably as a cost saver:

open source license

Baxter is designed to be versatile and adaptive for a wide range of jobs. As such, Baxter's integrated Unix-based OS integrates open source licensing. The open source licenses are available at the link below. They are available to enable registered Baxter owners to understand the legal responsibilities and general use guidelines associated with the product. [9]

References

[1] Accessed Oct. 28, 2012, "1961: A peep into the automated future", By PAUL MICKLE / The Trentonian, http://www.capitalcentury.com/1961.html

[2] Accessed Oct. 28, 2012, http://www.electronicsnews.com.au/features/robotic-arm-inventor-george-c--devol-dies-at-99

[3] Accessed Oct. 28, 2012, http://www.idemployee.id.tue.nl/g.w.m.rauterberg/presentations/hci-history/tsld090.htm

[4] Accessed Oct. 28, 2012, http://robostuff.com/http:/robostuff.com/robots-catalog/robots-by-country/usa/unimate/

[I think this is generic historical open source content and non-proprietary]

[5] Accessed Oct.28, 2012, http://www.bootsandsabers.com/index.php/weblog/permalink/unimate

[I think this is generic historical open source content and non-proprietary]

[6] Accessed Oct. 29. 2012,"Historical Overview Robots", http://www.idemployee.id.tue.nl/g.w.m.rauterberg/presentations/hci-history/tsld089.htm

[7] Accessed Dec 1, 2012 http://www.rethinkrobotics.com/

[8] Accessed Dec 1, 2012, video screen captures from http://www.rethinkrobotics.com/index.php/support/open-source/

[info@rethinkrobotics.com sent for permission to use 12_1; no response as of 12_5]

[9] Accessed Dec 1, 2012, http://www.rethinkrobotics.com/index.php/support/open-source/

Section 3.1 Similar Projects

In order to bring this project to life extensive research was done on similar projects. Relevant ideas and designs were taken into consideration from current and past projects where robotic arms were controlled by similar mechanism discussed in the specifications of this report. Although the general idea of the implementation for this project is understood, the specifics of how to implement each individual piece were unclear. The information gathered from the projects found allowed an insight into the design of the different components that make up this project.

From the sources, insights were taken from both the innovations and the mistakes made by each. The notes gathered from each project helped further the design process and lead to a more refined design in itself. Some of the projects had a major financial backing and were very complex while others were simple proof of concepts done with spare parts. The adaptation of these technologies will ultimately lead to a midpoint between the well-funded projects and the cheaper ones.

3.1.1 Luke's Hand

One relevant project worth mentioning is a past senior design project. This project was done by group 6 of the fall 2009 semester. The project was called Luke's Hand and the goal was to create a prosthetic limb that could mimic the movement of a human hand and be controlled a number of ways. A picture of this project can be seen in figure 3.1.1.

The project's goal was to have the hand controlled by three methods, through a computer program, an RC controller, a glove that the user wore, or switches on the control board. The project ultimately ended with only the glove interface and switch interface being implemented due to various reasons. The glove interface consisted of a cloth glove that contained a flex sensor in each finger. As the user would move their hand the flex sensors would send a signal to the microcontroller telling it how far each finger was bent. The microcontroller then told the motors that controlled the robot fingers how far to move.

The robotic hand also had sensors on it in order to determine if objects were in its hand and whether or not they were slipping. The sensors used included touch sensors and a potentiometer. The touch sensors were used to provide feedback to the microcontroller to let it know that the hand was currently holding something. The potentiometer was used to detect if the object being held was slipping at all. If the object was slipping the force measured by the potentiometer would change and the microcontroller would send a signal to the motors to tighten the grip on the object.

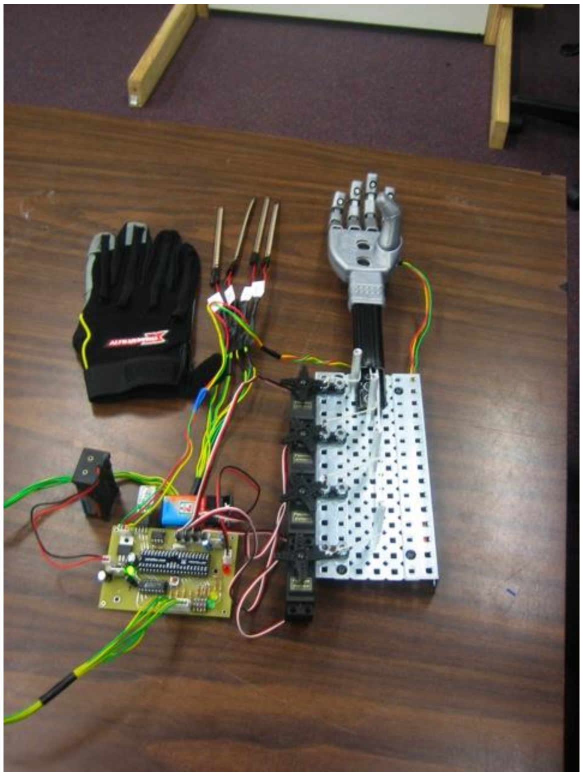


Figure 3.1.1: Luke's Hand

3.1.2 Haptic Feedback Robot Hand

Another relevant design was found in a senior design project from San Francisco State University. The goal of this project was to create a wearable interface for a robot that could provide haptic feedback to the user. Haptic feedback is a physical response that allows the user to feel what the robot is doing; similar to how an indicator light is feedback for our sense of sight. A picture of this project can be seen in figure 3.1.2.

The project resulted in a two fingered robotic hand that was controlled by a glove worn by the user. The glove had a small servo motor attached to it with a brace placed on the user's index finger. The servo motor would provide feedback on its current position and then a microcontroller would send that position to the servos on the hand causing them to move to the same position. The servo on the glove had to be modified to allow it to move with the user's finger instead of trying to stay in its set position. The servo was rigged to take the slight change in position caused by the users attempt to move their finger and use that as the servo's new set position. This allowed the servo to update its own position based on the movement of the user's finger.

In order to generate the haptic feedback the fingers of the robotic hand had force sensors on the end of them. These sensors were connected to the control loop for the servo on the glove. In the case that the robot hand was grabbing something the force sensors would generate a voltage reaching a 5V max. This would negate the update signals to the servo on the glove causing it to prevent any more motion from the user. This restraint on motion would allow the user to feel the object that was in the robot's grip.

Figure 3.1.2: Haptic Feedback Robot Hand

3.1.3 Other Projects

Many other projects were found while researching examples but most of these projects were either commercial products or projects being developed by universities. Since commercial products have monetization as a goal, the technical details were unavailable or protected by patents, as were most of the university projects. Even though these projects contained good and relevant ideas, because the technical details were not available they could not be studied in enough detail to provide insight in this group's project.

3.1.4 Similar Robot Arm Projects

A search of the internet found some similar projects on youtube and other websites. A couple of pictures of motion tracking robot arms are shown below that combine the shoulder and elbow swivel horizontal / vertical angular motion in one base unit with rotating wrist and pincers.

Comments: I notice they are both wired – so it might be a good idea to add wireless after we can get the wired working. A far heavier weight master-slave arm is shown last. I am currently trying to find out how accelerometer and gyroscope sensors can be implemented for our project.

Haptic robotic arm 1 [1]

Fig. 3.1.3

Comments: light-weight looks inexpensive and has combination shoulder/elbow plus rotating wrist and pincer/gripper

Haptic Robotic Arm 2 [2]

Comment: similar to previous except operates by sensor on operator's right hand pointer finger and not by arm movement.

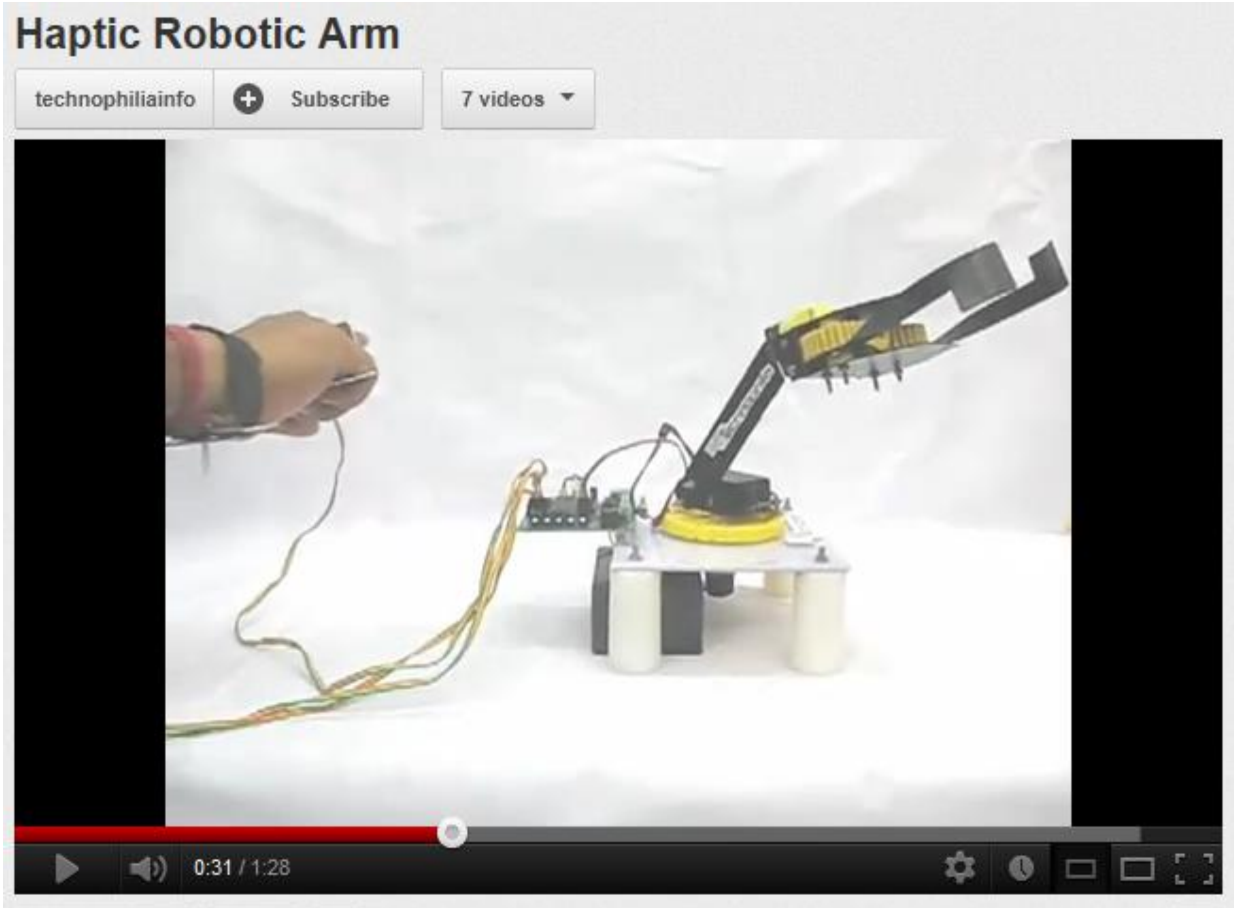


Fig. 3.1.4

Comments: Our project is currently more ambitious than these two in that we are trying to control a full 3 linkage arm featuring shoulder-to-elbow and elbow-to-wrist linkages and wrist-to-end-effector linkage (5 DOF) and not just a single linkage from base rotation shoulder to wrist and end-effector (3DOF and 4DOF with wrist rotation on *Haptic robotic arm 1*).

Remote Control of Robot Arm by using Master Slave System [3]

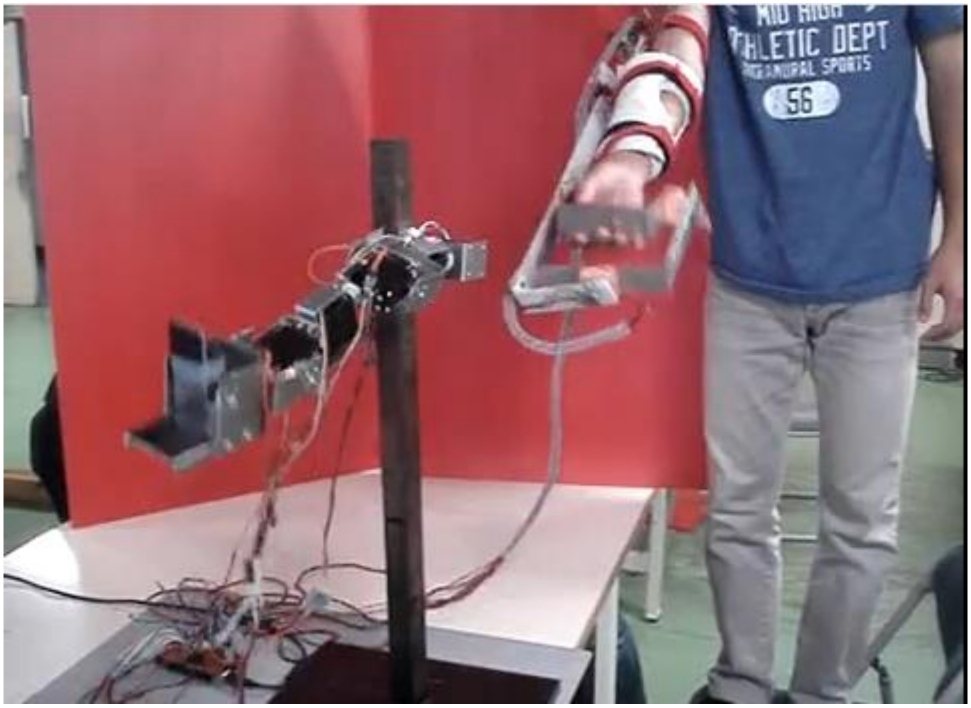


Fig. 3.1.5

"Yes, we used Arduino and KRS-788HV"

Note: Kondo KRS-788HV ICS Servo Motor Red Version $99

Comments: A way heavier duty and more expensive motor than what we are looking at. This project appears more difficult than the first two, but it's heavy-weight construction makes it cost prohibitive.

Other projects reviewed more closely included previous EEL4914 projects from 2006, DUCSS a DARPA autonomous vehicle sensor system, and a 2009 project Luke's Hand, an attempt at a human prosthetic robotic hand.

References

[1] Accessed Oct.22, 2012, http://www.youtube.com/watch?v=DlNtsQYsT-Q&feature=related [No contact info on youtube]

[2] Accessed Oct.22, 2012, http://www.youtube.com/watch?v=vqbNoALYVd4&feature=related

[3] Accessed Oct.24, 2012, http://www.youtube.com/watch?v=z3-6A4PBgmA [No contact info on youtube]

Section 3.2 GENERAL RESEARCH ON ROBOTICS

GENERAL RESEARCH ON ROBOTICS (as related to our project)

Because the Electrical Engineering program at UCF doesn't offer any undergraduate courses in Robotics - or wireless communications - it was determined appropriate content for this design project to include a section on robotics in general. This author actually took the EGN3060 course Introduction To Robotics but it was neither mechanically nor electronically oriented, it was computer science oriented programming in java of holonomic (roomba round) robots. Motion planning and maze solving are somewhat advanced topics for an introductory course. The PID stuff was easy, though. I found it curious that embedded systems EEL4742 did not do any labs with mechanical actuators either – no dc, stepper, or servo motor labs.

Robotics

The definitions of robotics are numerous and varied, ultimately they all deal with a labor-saving machine that with increasing technological capabilities gets closer and closer to human mechanical and mental capabilities. The INTRODUCTION section of this report illustrates the original industrial robot arm Unimate from 1961 - which doesn't look much like a human arm - compared with the 2012 (this year) Baxter which is basically a combination of two robot arms on a central body processing center with a humanoid face on video display for a head and a wheeled mobile base. The focus in current robotics research and industry seems to be quick and simple implementation in the workplace, with the robot being functionally programmed by non-robotic personnel – general workers, Baxter's co-workers.

A general trend in robotics is increasing autonomy, from the 2006 UCF senior design project DUCSS on DARPA autonomous vehicles referenced in OTHER PROJECTS, to the more famous Google Cars or the currently available option in some cars of automated parallel parking – autonomy from machines in previously human only functionality is on the rise.

Google's constant attempts to spread it's arms above it's search engine and advertisement placements has brought in an experimental robot car that drives all by itself (under the supervision of a human safety driver), just in case the car needs help. Google's blog said that it all works with the help of video cameras, radar sensors and a laser range finder to "see" other traffic, as well as detailed maps collected by their manually driven vehicles to navigate the road ahead.' The car drives quite well on it's own. [2] *The Darpa Grand Challenge started in 2004 and is well under way to achieve its goal "... to make one-third of all military vehicles autonomous by 2015. . .*[pending funding][3]

Beginnings

Robot as a word and concept is less than a hundred years old. Somewhere in the first half of the 20th century automated machines began to appear in manufacturing and became increasingly prevalent through the end of the century. Robots now perform many high-tolerance machine performances, particularly in Integrated Circuits and MEMS (micro-electrical-mechanical systems) manufacturing that could never be accomplished by humans alone. Robots are increasingly being used to facilitate high-tolerance surgery on human bodies both directly by the surgeon or remotely or even autonomously - "Robotic surgery, on the other hand, requires the use of a surgical robot, which may or may not involve the direct role of a surgeon during the procedure."[7] Suffice to say, robots are here to stay.

A More Technical Survey of Robotics

A robot consists of sensors, processors, actuators and power sources. Actuators include components such as motors, and the linkages the motors are driving, and end-effectors located at the end of the linkages to grasp or otherwise manipulate objects. Processors vary from small embedded systems micro-controllers to laptop or desktop pc's to even larger mainframe computer controlled robots. Power sources can be internal batteries or external ac to dc converted power sources. For digital robots there must always be an electrical power supply. Analog computing and robotics are not considered. Sensors are the marvel of the twenty-first century robotics ranging from laser range finding to micro-electrical-mechanical-systems on a chip like gyroscopes (angular motion detectors), accelerometers (linear motion detectors), and magnetometers (directional motion, like a compass, detectors).

Actuators

Our project, being on a human-interactive robotic arm, is really at the heart of industrial robotics. Industrial robots have many shapes based on their functionality; however, a human arm-like function is very common. Our current plastic robot arm uses revolute (rotary) joints driven by light-weight dc motors and a threaded drive on the motor shaft driving small plastic gear trains.

Our robot arm is what is known as a serial robot which is contrasted with parallel robots "...Most industrial robots are equipped with serial technology, where each axis is in line relative to the preceding one. The parallel robot on the other hand has three or more prismatic or rotary axes which function parallel to one another. Examples of parallel robots are Tricept, Hexapod and Delta Robots."[4] Parallel robots offer greater stability and have the advantage of averaging linkage motion errors compared to serial linkage arrangements like a robot arm that accumulates the errors in each link motion. [4] This is something I didn't learn in EGN3060 Introduction to Robotics at UCF (you can discount this sentence from the total page count, but it is still true).

By definition, robot arms are "Articulated robots with a mechanical manipulator that looks like an arm with at least three rotary joints."[5] There are numerous types of robots defined by body type and the nature of their geometrical motion range such as Cartesian, cylindrical, polar, and even spherical. Revolute robots such as the robot arm are considered the most difficult "to program offline" and the "most complex manipulator." [5] They are the most difficult to program because they can reach the same location from both above and below a given starting point or reference location with different linkage configurations. They are also the most efficient in terms of work area coverage from a given amount of floor space. [5]

A Cartesian robot has three orthogonal axes x, y, and z that define a rectangular work geometrical volume or envelope. Cartesian robots are much simpler geometry to program and have high-volume (frequency), highly repetitive task use in industry. The articulated robot arm operates in a virtual hemispherical volume or work envelope and can do both highly repetitive assembly line welding or part placement, or the extremely complex and spontaneous human-operator interactive actions such as robot surgery or remote hazardous material manipulation. The robotic arm is generally considered to operate in a polar coordinate system. However, since we are using the exteroceptive feedback system of the human-operator's vision and response, we should not need to program point-to-point locations, only proportional angle rotations. That is the idea.

Generally, there are three types of actuation moving the linkages - hydraulic, pneumatic, or electrical. Hydraulic is considered the choice for heavy-weight payload applications, pneumatics is the choice for medium-weight payloads and electrical driven systems are considered the easiest to control.[1] Our project is focused on spontaneous and fine motion-tracking actuation and has a very small payload demand. However, the robot arm could be upgraded to a heavier weight design and the control system would only need more powerful motor driving current and voltage capacity.

Three terms that are often used with articulated robots motion are yaw, pitch, and roll. It is best to remember them in this order and to associate them with the progressive motion of the robot arm from its base as horizontal plane of rotation around the base (yaw), vertical plane of rotation around each linkage joint (pitch), and perpendicular plane of rotation at the wrist (roll). Our initial design with the plastic arm does not have the option of roll at the wrist, but a rotatable wrist is an option ($35) with the Lynx metal arm.

Accuracy of the robot arm motion is a very important consideration for a motion-tracking application such as our arm, but quantitative analysis will have to wait until we get the arm actually working. Our accuracy requirement is not the traditional repeatability requirement of pick-and-place robot applications, rather an accurate proportional movement of the robot arm to the human-operator's arm that is stable and repeatable.

The end-effector of our current robot arm has foam rubber pads on the jaws of the gripper that can easily hold a smooth object such as a pen or pencil between the jaws of the gripper without breaking the object. The end-effector has a translation (open-close) motion that is driven by another dc motor (M1).

<u>Sensors</u>

The heart of our own project is currently understood to be the gyro sensors that will detect the human-operator's arm motion and send the data to the processing micro-controller. *Sensors are the perceptual system of a robot and measure physical quantities like contact, distance, light, sound, strain, rotation, magnetism, smell, temperature, inclination, pressure, or altitude. Sensors provide the raw information or signals that must be processed through the robot's computer brain to provide meaningful information. Robots are equipped with sensors so they can have an understanding of their surrounding environment and make changes in their behavior based on the information they have gathered.*[19] Another sensor employed in our project is an FSR (force sensitive resistor) that communicates to the robot arm whether the end-effector is to remain open or be closed around an object. The force sensitive resistor works by changing (decreasing) its resistance as a force is applied to the contact area and is explained in detail in the section HOW FORCE SENSORS WORK.

A quick web search of online robot shops identifies a multitude of varied and frequently quite sophisticated sensors.[22] Laser range finders, accelerometers, gyroscopes, compass modules, GPS modules, light detectors, barometric pressure and moisture sensors and many more readily available for mostly under

$100 or even $50. At this point in time, our project is only employing MEMS gyroscope sensors (3 DOF) and a force sensitive resistor sensor. Some type of simple feedback sensor like potentiometers might be employed on the robot arm joints. For details on how the gyro sensors work see section HOW MEMS GYROSCOPES WORK.

Processors

In what are called low-technology applications, robots are controlled by cams and valves called hard automation systems. Reprogramming is a problem.[1] Obviously, our robot arm will be controlled by microprocessor power on a printed circuit control board. *The controller is the robot's brain and controls the robot's movements. It's usually a computer of some type which is used to store information about the robot and the work environment and to store and execute programs which operate the robot*.[19]

As an interesting aside, regarding the future of Baxter's evolution, robotic research scientist Hans Moravec predicts a four stage progress to universal robotics – human intelligence equivalent robots[20]:

- *Year: 2010 Processing power: 3,000 MIPS Intelligence equivalent: Lizard*
- *Year: 2020 Processing power: 100,000 MIPS Intelligence equivalent: Mouse*
- *Year: 2030 Processing power: 3,000,000 MIPS Intelligence equivalent: Monkey*
- *Year: 2040 Processing power: 100,000,000 MIPS Intelligence equivalent: Human*

A research homework paper I wrote for EEE4314 indicates the Moore's Law progression hitting the wall of 3 nm by the early 2020's. The computing horsepower progression could be a problem. Also, contrast this prediction with the robot industrial expert quote at the close of this section.

Computer programming is done inside the processing power of the robot and generally involves control structures, data types, signal processing algorithms, and hardware control algorithms.[21] Our project is basically two parts; design and build the printed circuit board controller and then program the sensor data from the human operator's arm to drive the robot arm motors.

Power Sources

Our current plastic OWI robot arm uses 4 "D" cell batteries to drive the dc motors with plus or minus 3 Volts driving the motors direction in forward or reverse at about 1 amp current. While small robotic applications *Power system alternatives include batteries, fuel cells and generators, thermoelectric generators, super capacitors, flywheels and even non-storage options such as tethers*[electric cord],[18] our project really doesn't require any such exotic considerations. However, we will initially need a 5 Volt power supply to drive the control board including the motor driver output signals. If we upgrade to the metal arm with more powerful servo motors we will probably also need to upgrade our power supply capability.

Power for industrial robots can be electric, pneumatic or hydraulic. Electric motors are efficient, require little maintenance, and aren't very noisy.[19] There is quite a bit more info about pneumatic and hydraulic power sources that is not relevant to our small robotic arm project. Nor will we be using solar power or radioisotope thermal generators.

<u>System Control</u>

Initial considerations of system control are the parameters DOF (degrees of freedom), accuracy, repeatability, and resolution.[1,6] Degrees of freedom is defined as the number of independent ways a mechanism can move. In our initial robot arm design there are five degrees of freedom, one horizontal rotation (around the base) and three vertical rotations (shoulder-elbow-wrist) and one translation movement (gripper jaws open-close). We might add another degree of freedom with the rotational wrist upgrade option on the metal arm. We probably will not exceed six degrees of freedom. Industrial robot arms typically employ five or six degrees of freedom. [6]

Accuracy is a little tricky to determine for our human operator arm motion-tracking robot arm in that there is not a calculated point-to-point motion within the work envelope of the robot arms' motion. Determining the appropriate calibration of proportional movement between the human arm motion and the robot arm will be a significant part of eel4915 - after getting the sensors connected and driving the motors in correct qualitative directional response. We will probably employ something like potentiometer feedback sensors on the robot arms' joints.

Repeatability is another problem in that we are not calculating a specific point of geometric location in our quasi-open-loop application compared to pick-and-place manufacturing and assembly robot arms. However, repeatability is also defined as "... the ability of a robotic system or mechanism to repeat the same motion", not just the same specific location. [6] We will have to determine some metric for measuring repeatability of the robot arm motions. An applicable definition of repeatability for our purposes is "... a measure of the ability of the transducer (sensor) to give the same reading when returned to the same position."[8] This again depends on the programming of the gyro sensors.

Finally, resolution is alternatively defined as "The smallest increment of motion or distance that can be detected or controlled by the robotic control system" [6], or "... a measure of the ability of the transducer (sensor) to break down the angular position into fineness of detail in a spatial pattern."[8] Yet again, we will have to find out how to program the sensor data from the gyros in the microprocessor to quantify these parameters.

Additionally, a specific definition of envelope(s):

Envelope: A three-dimensional shape that defines the boundaries that the robot manipulator can reach; also known as reach envelope.

- *Maximum envelope: the envelope that encompasses the maximum designed movements of all robot parts, including the end effector, work piece and attachments.*
- *Restricted envelope is that portion of the maximum envelope which a robot is restricted by limiting devices.*
- *Operating envelope: the restricted envelope that is used by the robot while performing its programmed motions.*

Reach: The maximum horizontal distance from the center of the robot base to the end of its wrist.[6]

Yet another consideration is payload – the maximum weight or load the robot can lift – and how the robot performs both unloaded and loaded in terms of speed response. Does the robot arm response degrade in any way when moving while carrying a load versus initial unloaded tests? [6]

<u>Types and Applications of Industrial Robots</u>

Some industrial applications of robots include ”material handling, machine loading and unloading, die casting, welding, inspection, assembly, spray painting”[13] and in industries such as “medical, military, space, automobile, electronics/semi-conductors, ship-building, construction, and aircraft and aerospace”[9] The nature of our human operated motion-tracking robot arm is currently used in the medical industry for robot and robot enhanced surgery and in the attempts at ubiquitous robotic manufacturing applications like Baxter (http://www.rethinkrobotics.com).[7]

One of the major industrial applications of robotic arms in industry is the above mentioned material handling. Robots and robot arms in particular are used for palletizing (stacking parts from an assembly line operation) and de-palletizing (unstacking parts in storage and placing them onto an assembly line). Ultimately, our robot could be used for such an activity, with significant power upgrading of all parts of the system; however, the nature of research in our application tends more toward cybernetic applications like robot assisted surgery or exoskeleton robotics (http://www.rexbionics.com). Rex Bionics advertises *the world's first hands-free, self-supporting, independently controlled robotic walking device that enables a person with mobility impairment to stand up and walk.* [16] For clarification, cybernetics is defined as *the theoretical study of communication and* *control processes in biological, mechanical, and electronic systems, especially the comparison of these processes in biological and artificial systems*. [17]

<u>Programming</u>

Referring to Baxter brings up the subject of programming. Of course, Baxter's and any other robots motions are programmed in some kind of central processing computing power; however, robots like Baxter claim to be application programmable by non-technical people by direct manipulation of Baxter's arms (walk-through method) and presumably voice interactive commands. The user does not have to write any code, or perhaps, not even use a teach pendant push-button interface. [10]

There are four basic methods of industrial robot programming [14]:

- *Lead-through, in which the robot is placed in teach mode and moved through the various points in space, and each point is recorded for playback at a later time.*
- *Walk-through, in which the robot is placed in teach mode and the manipulator is walked through the various points of the program, and each point is recorded for later playback during operation*
- *Plug-in, in which a prerecorded program is uploaded into the robot's control processor*
- *Computer programming, in which programs that are written off-line on a computer are transferred to the robot's controller*

Another question is raised from this information – will our robot arm need to be adjusted or even reprogrammed when the human operator is changed from one person to another? Presumably, we will use a hinged mechanical template over the operator's arm that will be transferable from one user to another as observed in the youtube screen captures in the section OTHER PROJECTS.

An interesting observation about industrial robots - the cost of implementing an industrial robot can be three to four times the cost of the robot itself because of the cost of the tools the robot uses combined with the cost of programming the robot. [11] This explains the motivation to create independently contained robots with direct application program-ability by non-technical people and versatility for multiple work environment uses and re-uses. Baxter is arguably an evolution of a Flexible Robotic Manufacturing System that traditionally required machine and routing change-ability such that *any applications that involve high-mix, high-volume assembly require flexible automation. Manufacturers need the ability to run different products on the same line. That's much more difficult to do with hard automation.* [12]

To close this section, an observation from an industry professional:

What are some of the advancements in robotics?

The biggest advancements have been in the precision, speed and strength of robots. Learning and artificial intelligence algorithms have probably been the biggest disappointments. I don't think we will see robots even remotely approaching human intelligence by 2050. [15] A decidedly opposing viewpoint, compared to the Hans Moravec theoretical prediction for Universal Robots.

References

[1]"Robotics: an Introduction", Delmar Publishing Inc.,2nd ed.,1988, Douglas R. Malcolm, Jr http://www.robotmatrix.org; pages,

[2]Accessed Dec 3, 2012,caption for youtube video, Uploaded by *idels1* [no contact info] on Apr 30, 2011, http://www.youtube.com/watch?v=J3l5X3gYHPo

[3]http://www.nationaldefensemagazine.org/archive/2011/August/Pages/Army,CarMakersPushAheadWithDriverlessVehicleResearch.aspx

[7] Accessed Dec 3, 2012, "Robotic surgery", http://biomed.brown.edu/Courses/BI108/BI108_2005_Groups/04/

[4] Accessed Dec 3, http://www.robotmatrix.org/ParallelRobotic.htm

[5] Accessed Dec 3, http://www.robotmatrix.org/RobotConfiguration.htm

[6] Accessed Dec 3, 2012, http://www.robotmatrix.org/RoboticSpecification.htm

[8] "Robotics: an Introduction", Delmar Publishing Inc.,2nd ed.,1988,p. 189, Douglas R. Malcolm, Jr http://www.robotmatrix.org

[9] Accessed Dec 3, 2012, http://www.robotmatrix.org/RobotApplicationByIndustrial.htm

[10] Accessed Dec 3, 2012, http://www.rethinkrobotics.com

[11] Accessed Dec 3, 2012, "Industrial Robots", http://www.learnaboutrobots.com/industrial.htm

[12] Accessed Dec 3, 2012, "The Role of Robotics in Flexible Manufacturing", March 3, 2009, http://www.assemblymag.com/articles/86147-the-role-of-robotics-in-flexible-manufacturing

[13] “Robotics: an Introduction”, Delmar Publishing Inc.,2nd ed.,1988,p. 312, Douglas R. Malcolm, Jr http://www.robotmatrix.org

[14] “Robotics: an Introduction”, Delmar Publishing Inc.,2nd ed.,1988,p.387, Douglas R. Malcolm, Jr http://www.robotmatrix.org

[15] Accessed Dec 3, 2012,”Robotics Engineer”, http://www.learnaboutrobots.com/roboticsEngineer.htm

[16] Accessed Dec 3, 2012, http://www.rexbionics.com

[17] Accessed Dec 3, 2012, http://www.thefreedictionary.com/cybernetic

[18] Accessed Dec 3, 2012, “Power Sources for Small Robots”, http://www.ri.cmu.edu/pub_files/pub1/dowling_kevin_1997_1/dowling_kevin_1997_1.pdf

[19] Accessed Dec 3, 2012, “Robot Systems – Power”, http://prime.jsc.nasa.gov/ROV/systems.html#power

[20] Accessed Dec 3, 2012, “The Road to Universal Robots”, ©2005 The Tech Museum of Innovation http://www.thetech.org/exhibits/online/robotics/universal/page12.html

[21] Accessed Dec 3, 2012, http://en.wikibooks.org/wiki/Robotics/Design_Basics/What_you_should_know

[22] Accessed Dec 3, 2012, http://www.robotshop.com/all-robot-sensors.html

Note: in Appendix Permissions, Disclaimers (refers section 3.2) and Copyright notification

3.3 Robot Business

According to the Wiki site on Industrial Robotics, the robot industry has become mature since 2005. I find that claim somewhat precocious, it seems to me that the robot industry is just really beginning in the 20teens, particularly regarding humanoid work capabilities.

The first web search for 'robot business' found the *Robotics Business Review* website.[1] A surprisingly robust and actively populated website that features current news and trends in robots and high technology considerations like the increasing legal “blurring” between people's identity and people's property because of things like cyborg robotics. The title of an article perhaps best explains the conflict *The Cyborg Agenda: Blurring Body Boundaries Where will the law draw the line between your intelligent prosthetic and you?* [2]

The website advertises itself as *the #1 resource reporting and analyzing up-to-the-minute business developments, technology developments and financial transactions across the fast-changing landscape of global robotics. Includes full access to the RBR50 — complete profiles and analysis on the Top 50 compelling robotics companies worldwide* for only $995 charter rate. [1]

A very interesting article was found that mentioned Baxter and the emergent concern about Baxter in the workplace – Baxter in the workplace instead of you! **Baxter and its growing band of next-generation, co-worker robots:**

Technology, in the form of robots, is on the march, checking off likely candidate jobs for work transformation and then showing up at work sites to show business owners how well they can do the job.

The rule of thumb for switching out a human for a robot is twice the human’s annual salary. Since new-generation robots such as Baxter and its kind (others are soon to arrive from Universal Robotics, ABB, FANUC and Yaskawa) run about $20K each, the business decision is easy to enough to see.

It’s said that $500B of the US GDP’s total output of $15T (2011) could, in the near future, be thrown on the backs of robots and that these machines at critical mass could make offshoring of US manufacturing a distant memory real fast.[3]

The article continues on to specify which industry workers are most vulnerable to replacement by robotics, **The jobs most at risk…**

So, who exactly are the neighbors, friends and relatives closest in jeopardy of getting axed by these disruptors? During 2012, the following jobs have seen significant robot tryouts, nearly all with excellent results: Butchers, dairy farmers, furniture makers, kitchen helpers, lumberjacks, machinists, miners, parts packers and pharmacists.[3]

A related article explains big robotic intentions within the mining industry: *The robot trials were part of a movement toward a future of autonomous mining in which Rio Tinto is investing $3.6 billion, the ultimate goal of which the company calls the Mine of the Future. Starting in 2008 and concluding in February of 2012, a vanguard of nine robots—an excavator, track dozer, wheel dozer motor grader and five 930E-AT haul trucks—all supplied by Komatsu Ltd, underwent extensive field testing at West Angelas. The Pilbara robot trials were a real-world success. The most-watched machine was the autonomous, AT version of the Komatsu 930E, the best-selling off-road dump truck in the world—and now soon-to-be* *strategic hauling tool of choice for the Pilbara's millions of tons of China-bound iron ore. Rio Tinto is pushing forward with a full-scale, autonomous mine implementation, including $442 million for autonomous ore hauling trains over its rail network.*[4]

Similar to the DUCSS project of UCF 2006 (see SIMILAR PROJECTS) under DARPA coordination, Rio Tinto has worked with Australian academia - *Australian Centre for Field Robotics (ACFR) at the University of Sydney early on in 2007, establishing the Centre for Mine Automation to which Rio Tinto committed an initial $21m of funding. According to the University of Sydney, its mission at the Rio Tinto Centre for Mine Automation (RTCMA) is to develop and implement the vision of a fully autonomous, remotely operated mine—both open pit and underground.*[5]

Needless to say, there were related articles for each of the job fields of vulnerability to robotic replacement detailing similar robotic developments. All of which generates the section FUTURE OF ROBOTICS. However, the current topic is what exactly are the business numbers about robotics?

The market for medical robotic surgery equipment looks good. IbisWorld *Robotic Surgery Equipment Manufacturing Market Research Report* **|** *Life Sciences* **|** *Medical Devices* **|** *Oct 2011* reports:

Equipped for growth: Revenue will soar, as demand for complex surgeries increases

The article claims a market size of $2 Billion dollars and an annual growth rate (2006-2011) of 30.7% of an industry currently comprising 33 companies and only 3,173 employees. [6] The report says:

Industry Analysis & Industry Trends

Equipped for growth, Robotic surgery equipment will continue to comprise a fast-growing segment of the medical device industry. Growing acceptance of minimally invasive surgery (MIS) will bolster industry demand, as surgeons require the equipment to perform complex procedures in tight spaces. While high regulations will likely inhibit profit margins and reduce innovation activity, the industry will experience substantial growth in the next five years. [6]

What about all the industrial robots? [7]

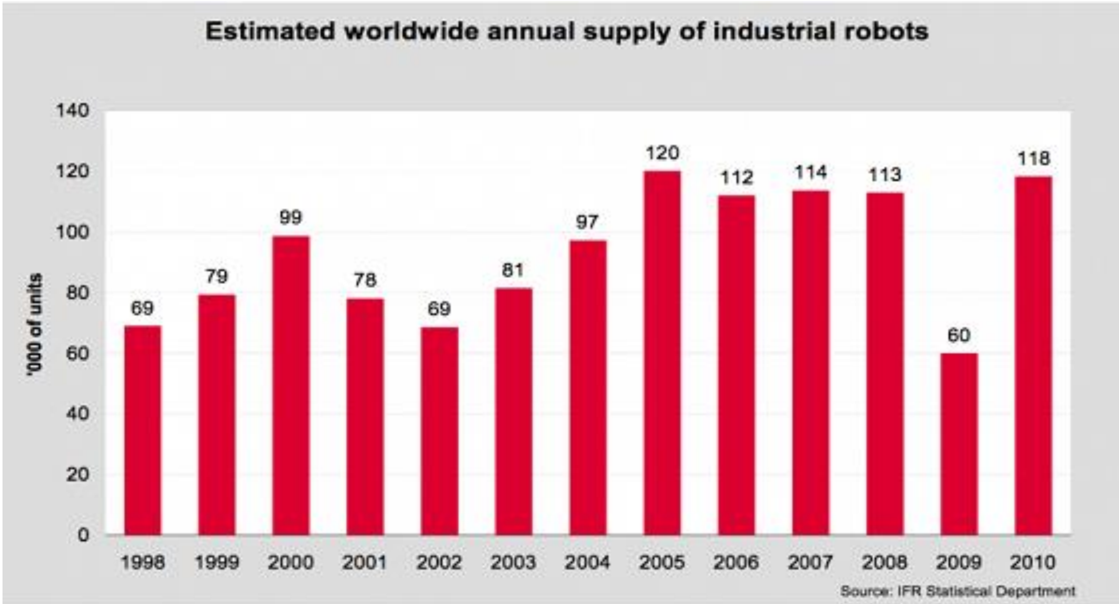


Fig. 3.3.1

And in what industries are they used? [7]

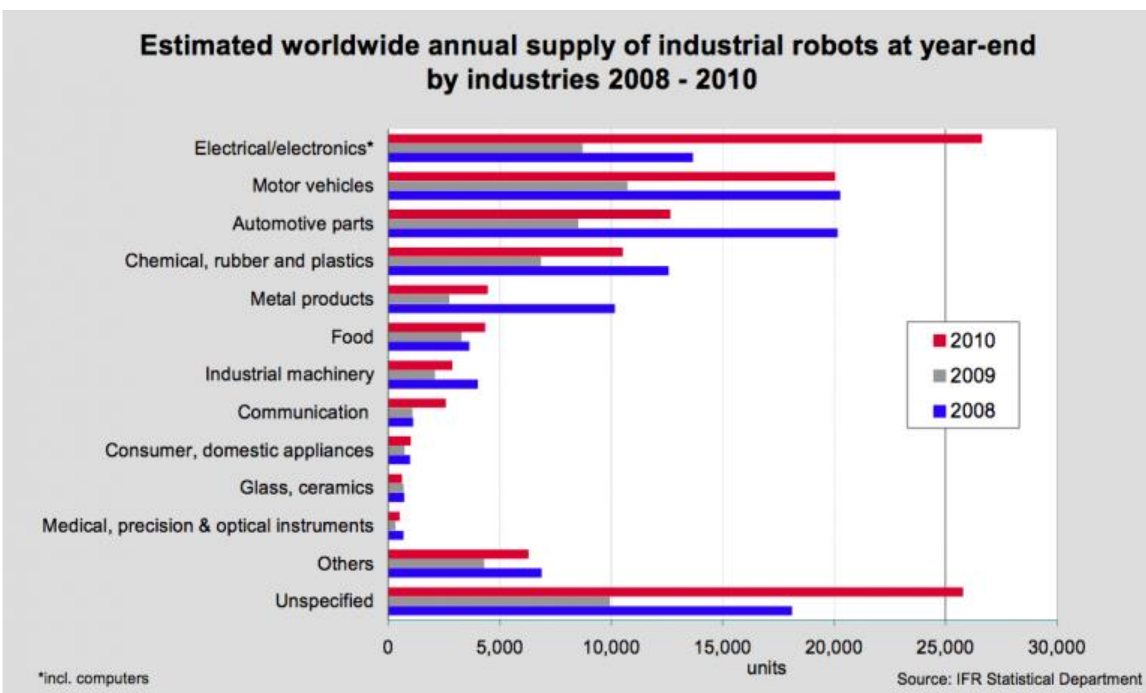


Fig. 3.3.2

To conclude, the robot industry is really just getting started at the final frontier – direct human worker replacement and augmentation as direct co-workers with humans in ubiquitous work environments.

References

[1] Accessed Dec. 4 2012, http://www.roboticsbusinessreview.com/

[2] Accessed Dec. 4 2012, “The Cyborg Agenda: Blurring Body Boundaries Where will the law draw the line between your intelligent prosthetic and you?” By Emmet Cole, http://www.roboticsbusinessreview.com/

[3] Accessed Dec. 4 2012, “Top 10: Best Jobs for Robots 2012 Help wanted: humans need not apply”, By Tom Green, http://www.roboticsbusinessreview.com/article/top_10_best_jobs_for_robots_2012

[4] Accessed Dec. 4 2012, 'Rio Tinto’s $3.6B Bid to Marry Mining to Robotics Rio’s “Mine of the Future” project and Komatsu’s AHS technology seem a perfect match', By Tom Green, http://www.roboticsbusinessreview.com/article/rio_tintos_3_6b_bid_to_marry_mining_to_robotics

[5] Accessed Dec. 4 2012, 'Rio Tinto’s $3.6B Bid to Marry Mining to Robotics Rio’s “Mine of the Future” project and Komatsu’s AHS technology seem a perfect match', By Tom Green, http://www.roboticsbusinessreview.com/article/rio_tintos_3_6b_bid_to_marry_mining_to_robotics/P2

[6] Accessed Dec. 4, 2012, by UCF student id, “Robotic Surgery Equipment Manufacturing in the US: Market Research Report”, http://www.ibisworld.com/industry/robotic-surgery-equipment-manufacturing.html IBISWorld’s Robotic Surgery Equipment Manufacturing market research report provides the latest industry statistics and industry trends, allowing you to identify the products and customers driving revenue growth and profitability. The industry report identifies the leading companies and offers strategic industry analysis of the key factors influencing the market.

[7] Accessed Dec. 5, 2012, “Foxconn gears up to build industrial robots - world industrial robot population to double”, http://www.gizmag.com/foxconn-gears-up-to-build-industrial-robots/20389/picture/146587/

3.3.1 FUTURE OF ROBOTICS

As observed in the previous section, the future of robotics technologically is increasingly ubiquitous interactivity with people; specifically, more human-like work capabilities of Baxter and his progeny. The trend is described as “robotic devices that cooperate with people.” [1]

The business future of robotics is also looking quite optimistic. Consumer robot sales are expected to reach $15 Billion dollars by 2015. [1] There is conjecture that in the near future robots could be accounting for $500 Billion dollars of the U.S. Gross Domestic Product and significantly reduce the amount of out-sourced manufacturing. [2] Additionally, *The industrial robotics market is expected to experience huge growth in the near future with a CAG*R [Compound Annual Growth Rate] *of 5.5%. It is further forecast that with this CAGR, the global industrial robotics market shall be worth USD 32.9 billion by 2017.it is estimated that the worldwide industrial robotics market will have a volume of approximately 1,500,000 units by 2017 whereas the estimated volume in 2011 is 1,100,000 units worldwide.* [3]

For the future, it appears the quest is reducing the initial installation costs of switching labor to robots, while at the same time, increasing the speed at which the robots contribute to productivity of ever smaller sized business with a greater variety of work capabilities. Cheaper and more versatile are the successful concepts of future robotics.

References

[1] Accessed Dec. 5, 2012, “Consumer Robot Sales to Surpass $15B by 2015” http://www.roboticsbusinessreview.com/article/consumer_robot_sales_to_surpass_15b_by_2015

[2] Accessed Dec. 5, 2012, “Top 10: Best Jobs for Robots 2012”, http://www.roboticsbusinessreview.com/article/top_10_best_jobs_for_robots”,_2012/

[3] Accessed Dec. 5, 2012, “Industrial Robotics Market – Global Industry Size, Market Share, Trends, Analysis And Forecasts 2012 – 2018”, http://beforeitsnews.com/science-and-technology/2012/12/industrial-robotics-market-global-industry-size-market-share-trends-analysis-and-forecasts-2012-2018-2502558.html

Chapter 4 Research Related to Components

4.1 Relevant Technologies

Some technologies that will be looked at as relevant to our project of a master-slave motion-tracking robot arm include MEMS and IC manufactured components, artificial intelligence, simulation, and computer programming.

4.1.1 MEMS and IC

MEMS (micro-electrical mechanical systems) are based on the IC (integrated circuits) thin film manufacturing technologies, and then add the magic of under etched "free-standing" structures that create dynamic and useful electro-mechanical transducers such as the gyroscopes used in our design proposal (see next section POSSIBLE COMPONENTS). Of course, the microprocessor on our control board is a product of IC manufacturing technologies.

4.1.2 Artificial Intelligence

AI (Artificial Intelligence) is the "brains" of computers and robots. AI is how humans program intelligence into machines. Nominally, AI is math logic combined with computer programming logic to simulate human decision-making intelligence in computers and machines. AI is generally associated with very difficult computer programming applications. We have not started programming our sensor-controller-actuator system yet, so we do not know if our programming demands are at an AI level of difficulty. It can be argued that any embedded system computing application is a form of AI.

4.1.3 Simulation

Simulation in robotics is generally simulating a mechanical robotic motion or device on a computer without having to make the actual physical device. Since we do not have to make our robotic arm mechanically we do not intend to do any simulation. However, in 4915, we will attempt some system simulations in Matlab regarding the motional stability of our robot arm.

4.1.4 Computer Programming

Computer programming is the most relevant of all the technologies. We do not have to make the mechanical robot arm(s). We do not have to design or manufacture any of the MEMS or IC chips used. We will probably just use the sensors as they are from the on-line robot shops. We probably won't have to design or make any gearboxes as the robot arms either have the gearboxes included or any servo motors selected already have gearboxes built-in. What we do have to do, however, after getting the basic electronics functioning, is program a working and stable robot arm master-slave mirror-motion system. If we succeed in our attempt, our project will probably be over 90% programming.

How explain computer programming as a relevant technology to a robotics project? Everything about robotics is getting some type of labor-saving human intelligence into a machine; whether a pick-and-place robot arm on an assembly line or Baxter co-workin' with the tunes on, the mechanism to make the robot think is only computer programming. While there is still some analog machine robotics around, 99% of current robotics uses digital intelligence. In fact, a web search for analog robotics turns up not much more than *BEAM is an acronym suggesting the basic factors to consider when designing robots: biology, electronics, aesthetics, and mechanics.* [1] BEAM is also mostly theoretical research and so are the few other analog robotics mentioned.

What computer programming will our robot use? For the prototype, to try to achieve first motion, we will just use the arduino language with the arduino mega microcontroller. "The Arduino language is based on C/C++. It links against AVR Libc and allows the use of any of its functions; see its user manual for details." [2] However, the microprocessor that we will use for our own control board, the ATmega128 or ATmega256 (which is also the microprocessor on the arduino microcontroller) can also be programmed in assembly language. [3] At this point in time, we intend to do most of our programming in C/C++; however, that may change as the project evolves.

References

[1] Accessed Dec. 5, 2012, "Analog Robotics: Chaos at the Core of Intelligence", by Marcos Kuhnshttp://www2.hesston.edu/Physics/RobotMK/report.html

[2] Accessed Dec. 5, 2012, http://arduino.cc/en/Reference/HomePage

[3] Accessed Dec. 5, 2012, "AVR Studio 4 and ATmega128: A Beginner's Guide", http://eecs.oregonstate.edu/education/docs/mega128/starterguide.pdf

4.2 Possible Components

Some possible candidates for the components to our project – the new MEMS gyroscopic modules, a microcontroller board for the prototype phase of our project, motor drivers, force sensors and accessories – include:

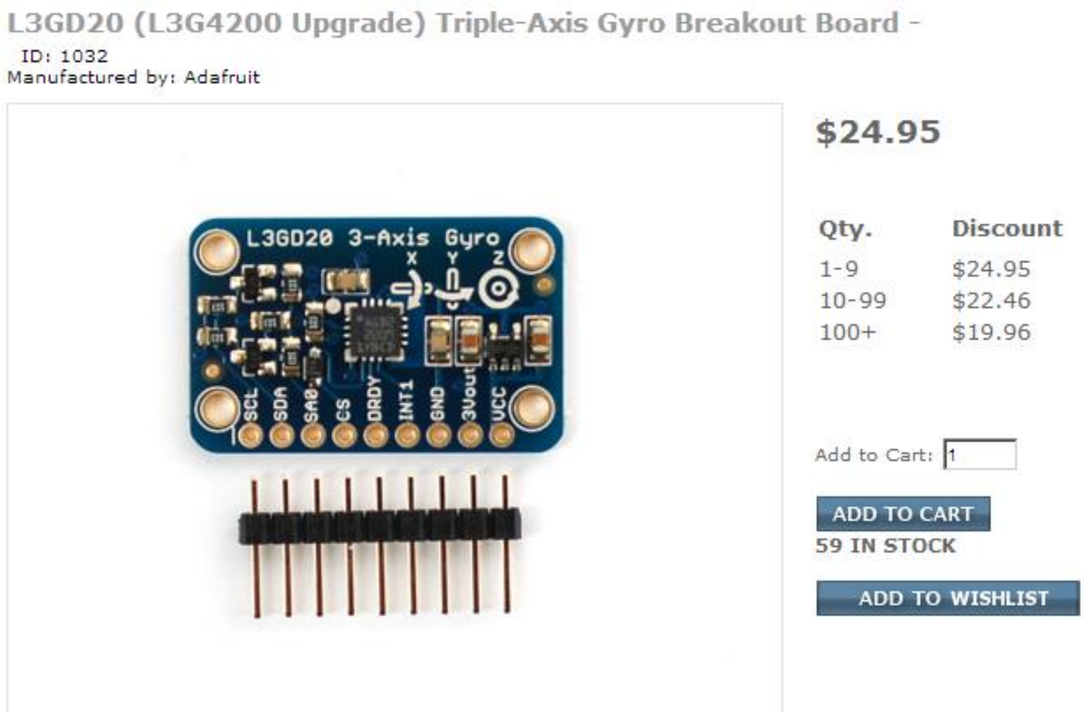


Fig. 4.2.1

This breakout board is based around the latest gyro technology, the L3GD20 from STMicro. It's the upgrade to the L3G4200 (see this app note on what to look for if upgrading an existing design to the L3GD20) with three full axes of sensing. The chip can be set to ±250, ±500, or ±2000 degree-per-second scale for a large range of sensitivity. There's also build in high and low pass sensing to make data processing easier. The chip supports both I2C and SPI so you can interface with any microcontroller easily. [1]

and: [2]

:

Fig. 4.2.2

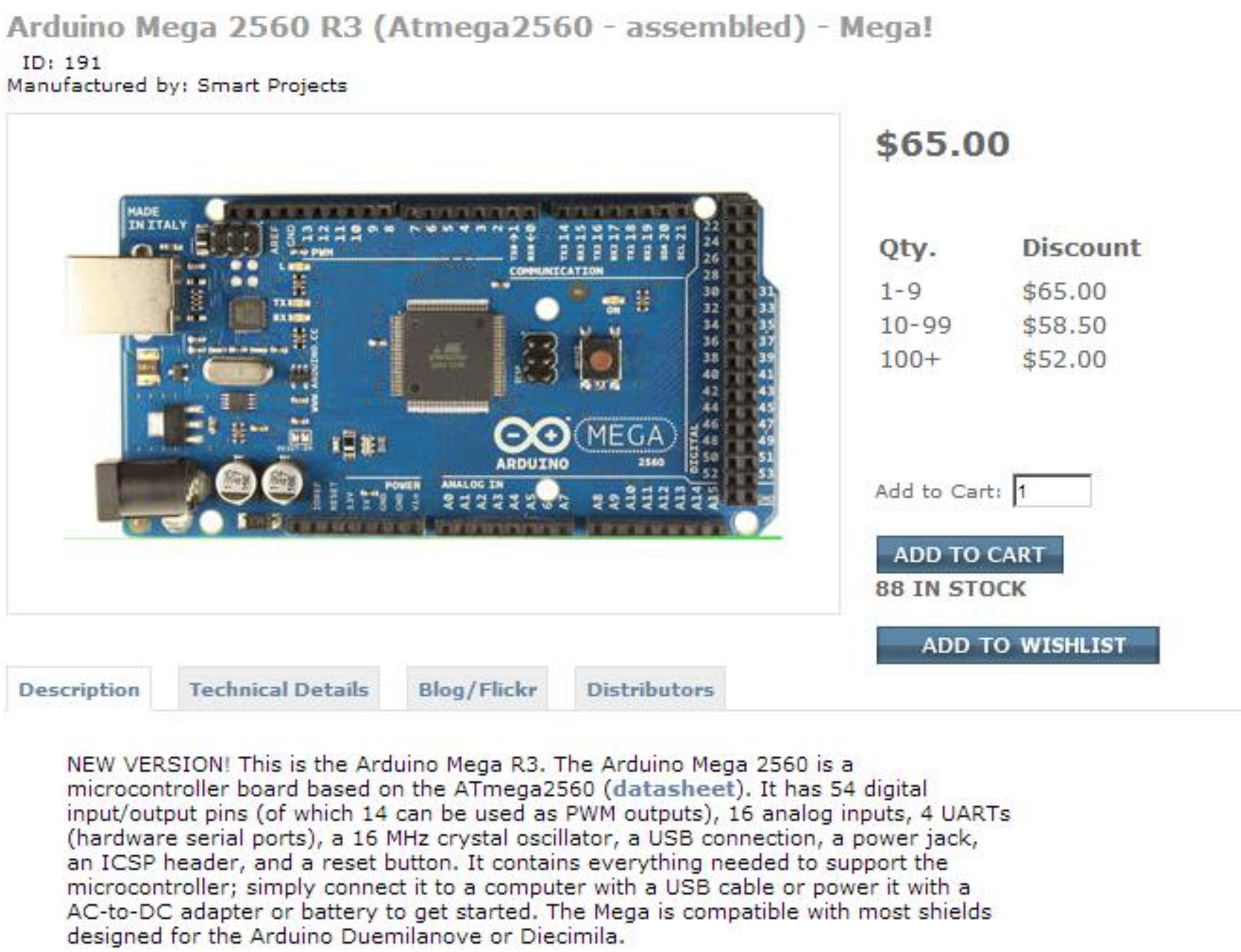


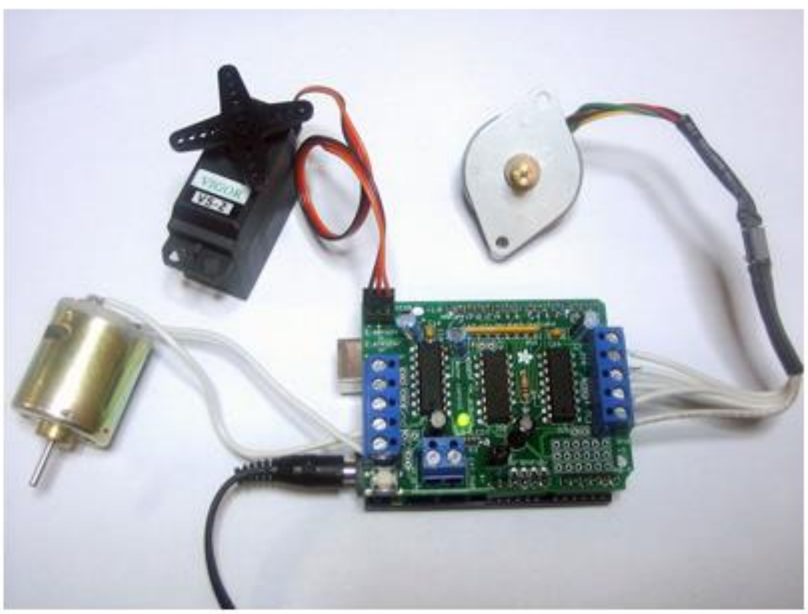
Fig. 4.2.3 [3]

Dual H-Bridge Motor Driver for DC or Steppers - 600mA - L293D -
ID: 807

$2.50

Fig. 4.2.4 [4]

Adafruit Motor/Stepper/Servo Shield for Arduino kit - v1.0
ID: 81
Manufactured by: Adafruit



$19.50

Qty.
1-9
10-99
100+

Please en
and we wi
when this
You will o
this produ

Your Nan

Your E-m

- **2 connections for 5V 'hobby' servos** connected to the Arduino's high-resolution dedicated timer - no jitter!
- 4 H-Bridges: L293D chipset provides 0.6A per bridge (1.2A peak) with thermal shutdown protection, internal kickback protection diodes. Can run motors on 4.5VDC to 25VDC.
- **Up to 4 bi-directional DC** motors with individual 8-bit speed selection (so, about 0.5% resolution)
- **Up to 2 stepper motors** (unipolar or bipolar) with single coil, double coil or interleaved stepping.
- Pull down resistors keep motors disabled during power-up
- Big terminal block connectors to easily hook up wires (18-26AWG) and power
- Arduino reset button brought up top
- 2-pin terminal block and jumper to connect external power, for separate logic/motor supplies
- Tested compatible with Arduino Mega 1280 & 2560, Diecimila, Duemilanove, and UNO
- Download the easy-to-use Arduino software library, check out the examples and you're ready to go!

Fig. 4.2.5 [5]

Servo Extension Cable - 30cm / 12" long - -

ID: 972

$1.95

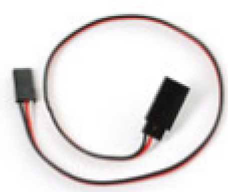


Fig. 4.2.6 [6]

ATMEGA256xx - 256k Flash Memory								
ATMEGA2560-16AU	ATmega2560 100-Pin 16MHz 256kb 8-bit Microcontroller		Yes	TQFP100	1	$10.90	1	ORDER
ATMEGA2561-16AU	ATmega2561 100-Pin 16MHz 256kb 8-bit Microcontroller		Yes	TQFP100	1	$11.90	1	ORDER

Fig. 4.2.7 [7]

Digi-Key Part Number	Manufacturer Part Number	Description	Series	Manufacturer	Core Processor	Core Size	Speed	Connectivity	Unit Price
ATXMEGA128A4U-CU-ND	ATXMEGA128A4U-CU	IC MCU 8BIT 128KB FLASH 49VFBGA	AVR® XMEGA	Atmel	AVR	8/16-Bit	32MHz	I^{2}C, IrDA, SPI, UART/USART, USB	4.75000
ATXMEGA128A4U-AU-ND	ATXMEGA128A4U-AU	IC MCU 8BIT 128KB FLASH 44TQFP	AVR® XMEGA	Atmel	AVR	8/16-Bit	32MHz	I^{2}C, IrDA, SPI, UART/USART, USB	4.40000

Fig. 4.2.8 [8]

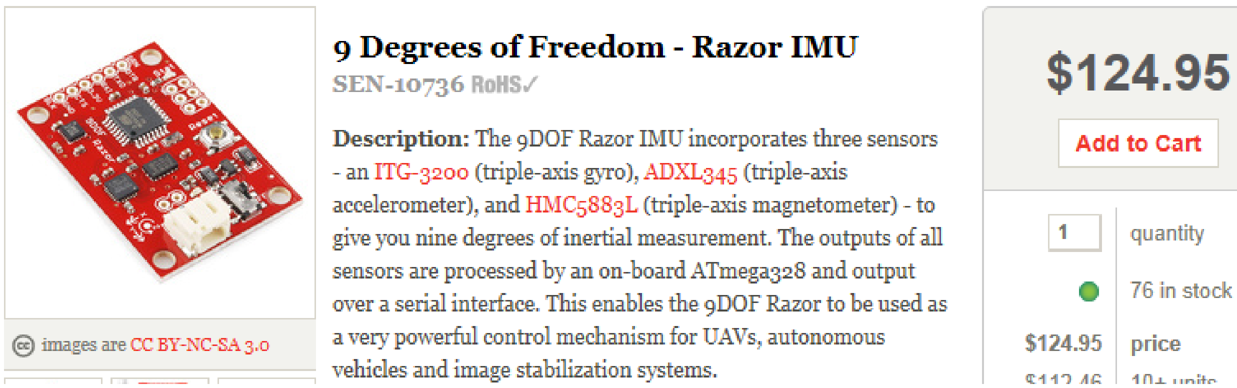


Fig. 4.2.9 [9]

At this time, we are not planning on making our robot arm mobile, but if we try, the inertial measurement sensors are available along with the gyros and accelerometers.

Initially, enough components will be acquired to start a prototype system to begin coding before we send the printed circuit board out to be made. See section 7.1 Parts Acquisition.

<u>References</u>

[1] Accessed Nov. 11, 2012, https://www.adafruit.com/products/1032

[2] Accessed Nov. 11, 2012, https://www.sparkfun.com/products/9801

[3] Accessed Nov. 11, 2012, https://www.adafruit.com/products/191

[4] Accessed Nov. 11, 2012, https://www.adafruit.com/products/807

[5] Accessed Nov. 11, 2012, https://www.adafruit.com/products/81

[6] Accessed Nov. 11, 2012, https://www.adafruit.com/products/973

[7] Accessed Nov. 11, 2012, http://www.futurlec.com/ICAtmel.shtml

[8] Accessed Nov. 11, 2012, http://www.digikey.com/scripts/DkSearch/dksus.dll?WT.z_supplier_id=313&WT.z_page_type=SP&WT.z_page_sub_type=SS&WT.z_oss_type=Keyword&v=313&lang=en&site=us&KeyWords=Atmel+ATxmega128A4U&x=13&y=8

[9] Accessed Dec. 5, 2012, https://www.sparkfun.com/products/10736

4.2.0 H/W UPGRADE OPTIONS

If the OWI-535 [1] doesn't meet necessary performance criteria, some hardware

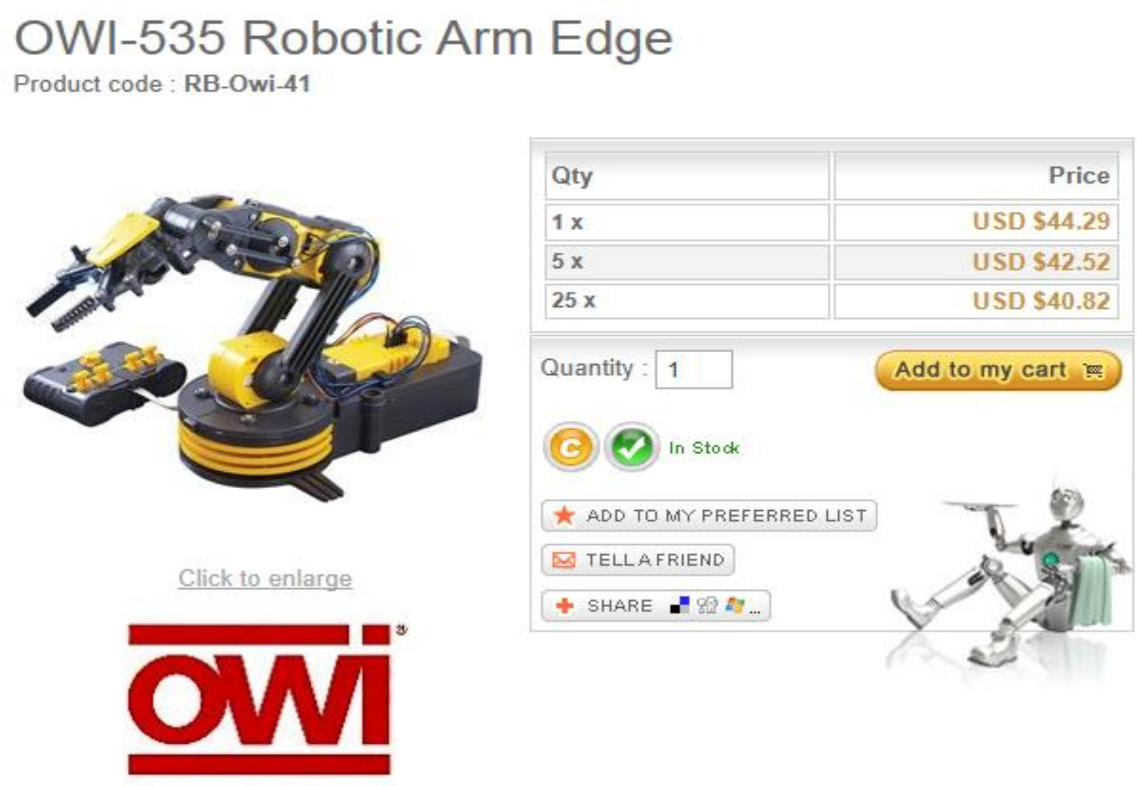


Fig. 4.2.1

upgrade options include [2]:

Home » Robots » Robot Kits » Robot Construction Kits » Lynxmotion Erector Set

Lynxmotion AL5B 4 Degrees of Freedom Robot (Hardware Only)

Product code : **RB-Lyn-268**

Qty	Price
1 x	USD $135.40
5 x	USD $129.98
25 x	USD $124.78

Options

Reach 7.5"

Version Hardware Only

View product availability

Specifications

- 4 + 1 Degree of freedom robotic arm
- 10.25" median reach
- Servo Motors sold separately
- Electronics and software sold separately
- Wrist rotate option sold separately

The **Lynxmotion AL5D 4 Degrees of Freedom Robotic Arm (Hardware Only)** line of robotic arms deliver fast, accurate, and repeatable movement. Each arm features base rotation, single plane shoulder, elbow, wrist motion and a functional gripper. The wrist rotate feature, which comes in a lightweight or heavy duty option, is sold separately. Lynxmotion designed an affordable system based on a time tested, rock solid design that will last and last. Add your own servos and electronics to complete the kit. The aluminum robotic arm is made from Lynxmotion's Servo Erector Set components for the ultimate flexibility and expandability. The kit consists of black anodized aluminum brackets, Aluminum tubing and hubs, custom injection molded components, and precision laser-cut Lexan components. For a list of suggested servos, see RB-Lyn-274.

Fig. 4.2.2

and: [3]

Fig. 4.2.3

and probably not [4]:

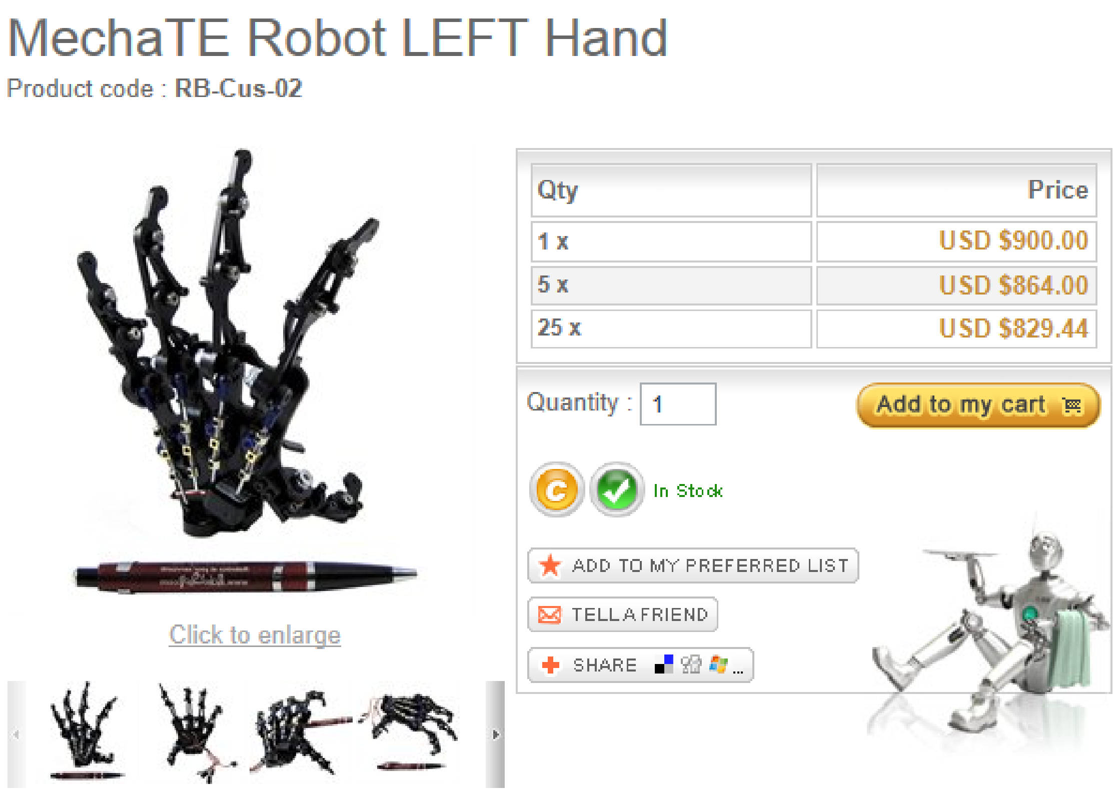


Fig. 4.2.4

References

[1] Accessed Nov 19, 2012, http://www.robotshop.com/owi-535-robotic-arm-edge-1.html

[2] Accessed Nov 19, 2012, http://www.robotshop.com/productinfo.aspx?pc=RB-Lyn-268&lang=en-US

[3] Accessed Nov 19, 2012, http://www.robotshop.com/lynxmotion-light-weight-wrist-rotate-upgrade-5.html

[4] Accessed Nov 19, 2012, http://www.robotshop.com/mechate-robot-left-hand.html

4.2.1 Microcontroller

The microcontroller will be used to monitor and store the sensor data from both the operator and the robot and use that data to control the motors in the robot so that the data from the robot matches that of the operator. Having all of the listed requirements will allow the microcontroller to be integrated into and meet the specifications for the project. In addition to the requirements for microcontroller there are other factors to consider. What language the microcontroller can be programmed in is one such factor. A microcontroller that can be programmed in C will be easy to use as it is a higher level language that the members of this project have experience with. Another factor to consider is an analog to digital convertor to read the signals sent by the sensors and convert them to usable data. For the robot to be able to respond at an acceptable speed the microcontroller must be at to process instructions quickly and the number of instructions it can process per second is another important factor to consider.

One microcontroller looked at for this project was the Nano 28 made by Basic Micro. This microcontroller has 24 I/O pins, 14 KB of memory, 368 bytes of RAM and 256 bytes of EEPROM. This microcontroller has enough I/O pins to handle all of the sensors that will be on the controller and robot as well as the motor controller. The amount of program space should be sufficient to hold the control algorithm for the robot and the amount of EEPROM is enough as that would be used to store default positions for the motors. The amount of RAM is also sufficient as that would be used to save the current values of the sensors to check against incoming data. This microcontroller also 11 A/D converter channels which would be enough to handle all the data from the sensors and allow simpler board design as it would not require an external converter.

Basic Micro products are provided with a free IDE called Basic Micro Studio that allows the programming of their products in BASIC. The IDE comes with an extensive library with example programs for many of their products as well as Lynxmotion robots. These libraries would be a great advantage but the use of BASIC requires learning of the language and might slow down the development of the project.

A second microcontroller considered for this project was the PICAXE-20X2 made by PICAXE. This microcontroller has 18 I/O pins, 4 KB of memory, and 256 bytes of RAM. This microcontroller has enough I/O pins to handle all of the components it will interacting with as well as enough RAM to store the temporary values of the sensors. The memory space might be an issue as it could be too small to hold the control algorithm which would then require some form of external storage which would complicate the board design. The 11 A/D converter pins that this microcontroller has will be sufficient for conversion the sensor data will require and would allow simpler board design.

PICAXE products are also provided with a free IDE called PICAXE Programming Editor. This IDE uses BASIC to program the microcontrollers but unlike Basic Micro does not have access to many libraries that could help speed the development process. PICAXE Programming Editor does have a unique feature in that it allows users to develop a program for their microcontroller using a flowchart. This feature allows users to create a flowchart that dictates a sequence of actions that should be taken by the microcontroller and then converts it to the BASIC code that would cause the desired effects. This tool might be useful in speeding the learning the BASIC language as users could create flowchart and then see what the code looks like to help the transition to just using code.

Another microprocessor considered was the ATxmega128A4U made by Atmel. This microcontroller has 34 I/O pins, 128 KB of memory, 8 KB of SRAM, and 2 KB of EEPROM. The 34 I/O pins would be more than enough to handle the components for this project. The 128 KB of memory is enough to store the control algorithm, the 8 KB of SRAM would be more than enough to store the temporary values of the various sensors, and the 2K of EEPROM is plenty to store any values that would need to be permanent such as a default position for the robot. The 12 A/D converter pins is enough to handle all the sensor data from both the user and robot.

Atmel makes many tools for its products accessible such as it's free IDE Atmel Studio 6. This IDE allows users to program their microcontroller in C/C++ or assembly language unlike PICAXE or Basic Micro. The IDE is also integrated with a library of source code for example programs. The use of C for the program language and the availability of various libraries would allow minimal effort in learning how to program for this specific microcontroller.

A final microprocessor considered for this project was the Propeller 40 pin DIP made by Parallax. This microprocessor has 32 I/O pins and 64 KB RAM/ROM. The amount if I/O pins is sufficient for this project. The amount temporary is also large enough any temporary variables that might be needed. The memory does lack any EEPROM or other semi-permanent storage for the control algorithm to be stored in as well as variables such as default states. This lack of permanent memory would require an external memory device and would take up board space. The Propeller does lack an A/D converter which affects a requirement for the microcontroller used for this project. In order for the Propeller to do A/D conversion it must rely on the manipulation of two of its registers. This would take up computing time and slow down the reaction of the robot which needs to be fast.

Parallax provides an IDE to program their Propeller microcontrollers. The Propeller microcontrollers have a unique architecture for microcontrollers in that they have eight processors that Parallax calls cogs. These processors are able to act independently of each other allowing the microcontroller to process many different things at the same time. Due to this unique architecture however Parallax must use its own programming language called Spin. Parallax does provide libraries with example programs, tutorials and schematics to help with development; however the use of an unfamiliar language might slow down development of the project.

A summary of the microcontrollers considered for use on the control board of this project is presented below in table 4.2.1 and contains information relevant to this project. The table includes the product name, number of I/O pins, the amount of memory, how many A/D converter channels, what languages they can be programmed in, and their price obtained from the producer website.

Name	I/O pins	Memory	A/D converter	Language	Price
Basic ATOM 24	24	14k code 368 RAM 256 EEPROM	11 channels	BASIC	$8.95
PICAXE-20X2	18	4k code 256 RAM	11 channels	BASIC	$3.88
ATxmega128A4U	34	128k code	12 channels	C/C++ or	$3.00

		8k SRAM 2k EEPROM		assembly	
Propeller 40 pin DIP	32	64k RAM/ROM	0 channels	Spin	$7.99

Table 4.2.1: Microcontrollers considered for the project

4.2.2 Motor Controllers

The motor controller will be used to monitor the current state of the motors and send directions to the motors. The motor controller is needed because motors operate at higher voltages that microcontrollers do not work at. The motor controller is capable of operating at these higher voltages and can receive directions from the microcontroller and pass them along to the motors. Having the listed requirements from the introduction would allow the motor controller to completely control the robot arm from a single unit. A motor controller designed for servo motors would be best as most robotic arms use servos for their control. Being able to control five motors at once would mean only one controller is needed and would less the space needed for the whole control board.

One motor controller looked at was the Phidget Adnvaced Servo 8-Motor controller. This controller is able to control up to 8 servo motors simultaneously. The controller allows the user to change, the position, velocity, and acceleration of all the motors connected. The controller also monitors the power consumption of each servo and protects the motors from overvoltage. The Phidget motor controller only connects to a USB port requiring control from a microcontroller to come through a computer. This is a conflict with the project specifications as it should work as an embedded system without any assistance from a computer.

A second motor controller researched was the 8-Servo PC Controller from ServoCity. This controller is also able to control up to 8 servo motors simultaneously. This motor controller allows you to control position, velocity, and acceleration but does not monitor power consumption of the servos. The motor controller is connectable to either a USB connection or a signal wire from a microcontroller. This flexibility allows this motor controller to be integrated directly into the control circuit for the robot and stay within the specifications for the project. The lack of monitoring of the motors means that there is the possibility of damaging the motors if care isn’t taken while building and testing the circuit.

A final motor controller looked at was the 8 channel servo controller from Cytron. This motor controller allows for the control of 8 servo motors simultaneously. It also allows the monitoring of each servo’s position and is able to deactivate or active them. Another feature of the motor controller is that it can set an initial starting position of any channel the next time it starts up. This feature would be useful for resetting the robot to a default position. The Cytron control board can also be chained together with other motor controllers to provide control over even more motors if required. The controller is able to communicate with a computer through the use of a USB to serial converter or with a microcontroller with an UART interface. This would allow the motor controller to interface directly with the rest of the control board for this project and stay in line with the specifications.

A summary of the motor controllers considered for this project is presented in table 4.2.2. The table includes how many motors the controller can control, what it can communicate through, what it monitors, and their price.

Name	Motors	Communication	Monitors	Price
Phidget Advanced Servo 8-Motor	8	USB	Position Power	$90.00
ServoCity 8-Servo PC Controller	8	USB/serial	Position	$69.99
Cytron 8 Channel Servo Controller	8	USB/UART	Position	$20.65

Table 4.2.2: Motor controllers considered for this project

4.2.3 Gyros

HOW GYROS WORK

MEMS (micro-electrical mechanical systems) gyroscopes measure the rate of change around an axis, which means to determine the amount of angular change - which is what is needed for the robot arm control – the output of the gyro has to be integrated to find the amount of angular change.[1] The gyroscopes are available in basic IC chip format.

The physics of operation of the gyro is a measured Coriolis force that occurs when a massive object has both translational and rotational motion simultaneously. If a rotational motion is applied perpendicularly to the direction of velocity of an object a Coriolis force is developed by the right-hand-rule perpendicular to both the velocity vector and the angular momentum vector. A capacitive sensing mechanism measures the force and the measurement is used to determine the rate of change of angle of motion.

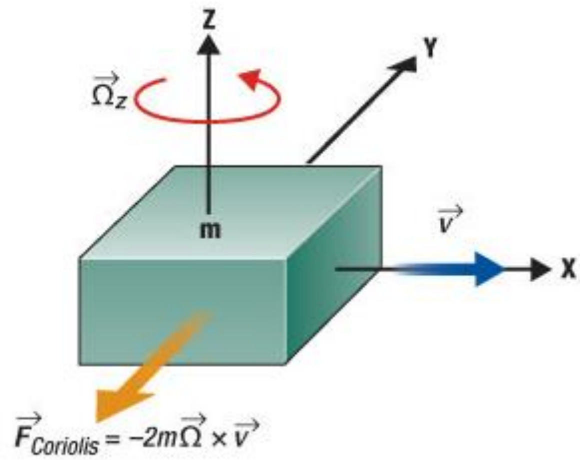


Fig. 4.2.5 [1]

This is an oversimplification of the actual MEMS gyro design – which uses two masses connected by a spring:

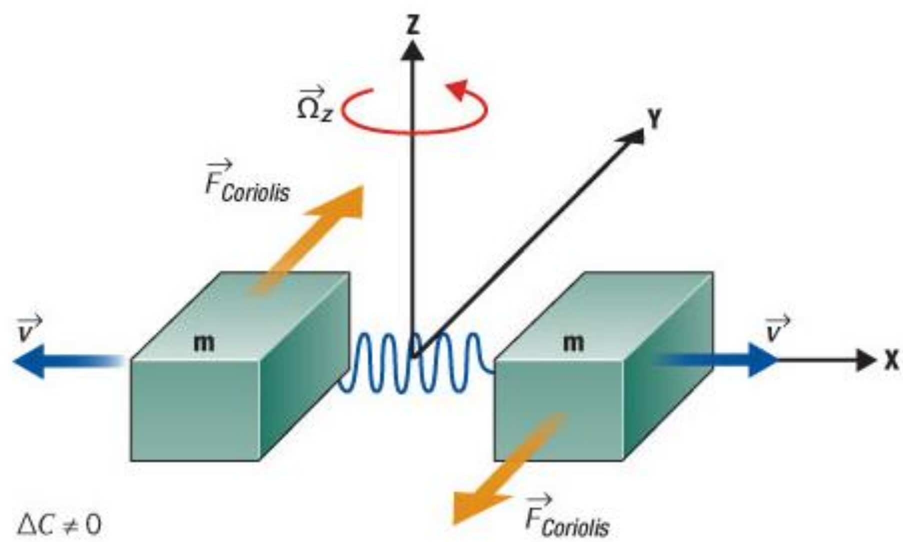


Fig. 4.2.6 [1]

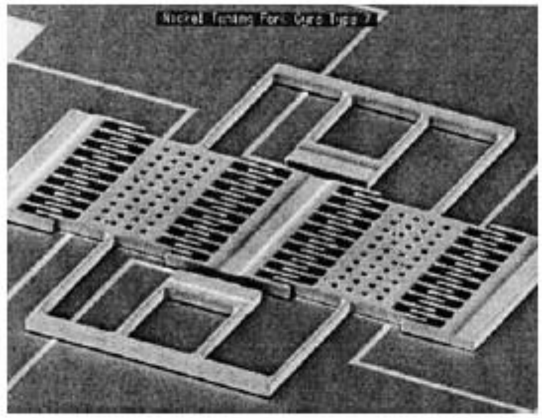


Fig. 4.2.7 [2]

MEMS implementation of above 'two masses design'

If the motion is linear only, the masses travel in the same direction and generate no change in capacitance; however, if a rotational motion occurs the force couple of the two masses moving in opposite directions generates a motion in the plates of a capacitance sensor and a measurement is made of the change in capacitance. The change in capacitance is measured as a change in voltage and processed through an analog to digital converter into the system processing power to determine rate of change of angle. Subsequent integration of the rate of rotation yields the actual amount of angular rotation. The gyroscopes are designed to measure the rate of rotation around all three axes, so one gyro can be used for any rotation measurement – yaw, pitch, or roll.

The gyroscope output can be expressed as Equation 1.

Rt = SC × (Rm - R0) (1)

Where,

Rt (dps): true angular rate

Rm (LSBs): gyroscope measurement

R0 (LSBs): zero-rate level

SC (dps/LSB): sensitivity

In order to compensate for turn-on to turn-on bias instability, after the gyroscope is powered on, one can collect 50 to 100 samples and then average these samples as the turn-on zero-rate level R0, assuming that the gyroscope is stationary. [3]

References

[1] Accessed Nov. 22, 2012, http://sensorwiki.org/doku.php/sensors/gyroscope

[] Accessed Nov. 18, 2012, http://clifton.mech.northwestern.edu/~me381/project/done/Gyroscope.pdf

[For those not registered with the CCC, send permission request in writing to Permissions Dept., Questex Media Group LLC, 275 Grove Street, Suite 2-130, Newton, Massachusetts 02466 or fax to 617-219-8310.]

[2] Accessed Nov. 21, 2012, http://www.sensorsmag.com/files/sensor/nodes/2003/970/fig6.jpg also displayed on source [1] http://www.quora.com/How-does-a-MEMS-gyroscope-work webpage

[exact same .jpg from 3 different sources – I think it is open source content]

[3] Accessed Nov. 22, 2012, “Introduction to MEMS gyroscopes”, http://www.electroiq.com/articles/stm/2010/11/introduction-to-mems-gyroscopes.html

[sent for permission 11_30 for both – an online 'contact us' form]

[no response as of 12_5]

4.3 Example Architectures and Diagrams

Many assembled control boards were viewed while researching parts for this project. Most of these examples had schematics that were viewable to study the architecture of them. These diagrams were useful for researching what components a control board would need to have, and how they should be laid on the pc board.

One control board that was studied for this project was the Arduino Leonardo seen in figure 4.3.1. Arduino builds control boards that are meant to be used to develop interactive objects. These boards can handle input from sensors, switches, and control physical objects such as motors and lights. The Arduino Leonardo has 20 I/O pins, 12 of which can be used as analog inputs, a micro USB connector, an ICSP header, a reset button and uses an ATmega32u4 microcontroller. The Arduino Leonardo is one of the first models that Arduino produced that didn’t require a second microcontroller to handle communications with a USB. This feature is relevant to the project because in making a choice for a microcontroller one that needs no other component in order to use a USB makes programming the microcontroller much easier once it is set in the PCB board. In order for Arduino Leonardo to control motors it must have a motor controller, like the Arduino Motor Shield, attached to it. The motor controller takes directions from the control board and with an external power supply capable of powering the motors, gives directions to the motors. Below is a picture of the schematic for the Arduino Leonardo.

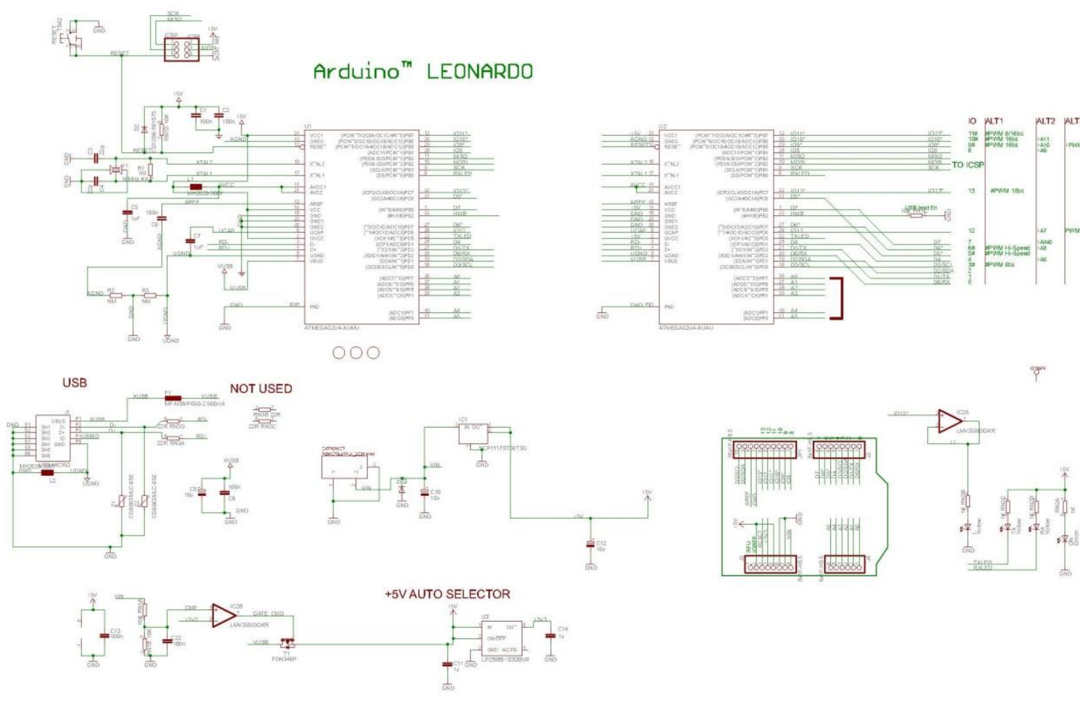


Figure 4.3.1: Arduino Leonardo schematic

A second control board looked at for this project was Polou’s Baby Orangutan robot controller shown in figure 4.3.2. The Baby Orangutan has 18 I/O lines 8 of which can be used for analog, a built in motor driver and uses an ATmega328P microcontroller. Measuring only 1.20”x0.70” the Baby Orangutan is built to be used in space constrained areas. This control board has a built in motor driver to handle the control of the motor's so unlike the Ardiuno board an extra motor control board is not needed. This reduces the amount of space the control board requires but also reduces the number of motors the control board can handle. This architecture shows that a motor driver can be used on the same pcb board as the microcontroller with no extra power source to drive the motors. The downside to this is that the motor driver on the Baby Orangutan can only power 2 motors making additional drivers necessary for more motors. Another drawback to the Baby Orangutan is that it requires an exterior component in order to program the microcontroller. The schematic for the Baby Orangutan is in the figure below

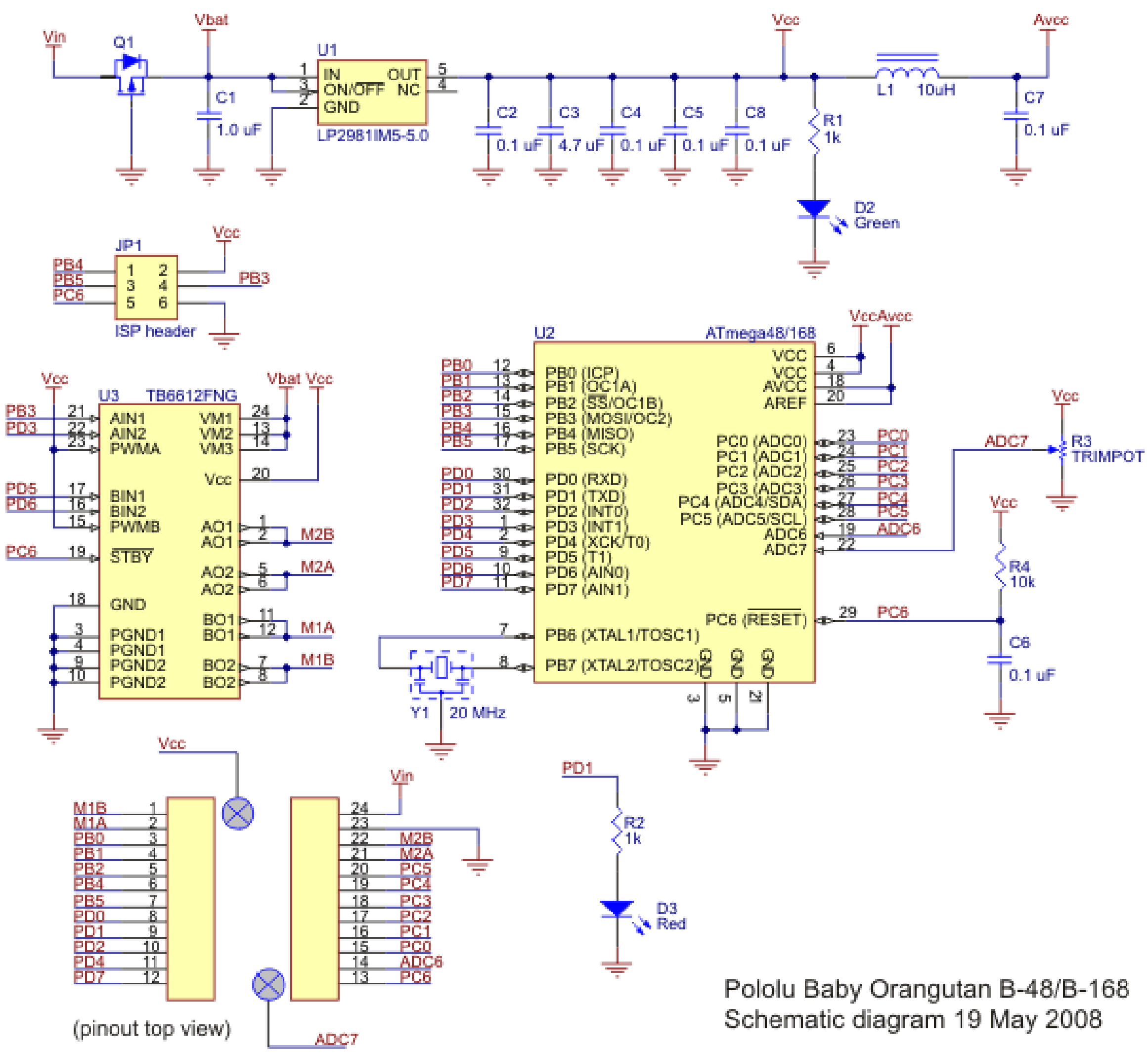


Figure 4.3.2: Pololu Baby Orangutan schematic

Another control board studied for this project was a previous senior design project from Fall 2009 shown in figure 4.3.3. The project, dubbed "Luke's Hand", also involved controlling motors based on input from sensors. This control board was particularly useful in that it did not have a separate motor controller but rather a separate power circuit for the motors with on and off signals sent from the microcontroller. This architecture uses more pins on the microcontroller since 1 pin per motor is required but saves space as a separate motor driver is not required. The downside of this control board was the microcontroller chosen. The microcontroller used was the Propeller 40 pin DIP which does not have a built in A/D converter or any internal EEPROM. The choice in microcontroller required an external A/D converter and an external EEPROM unit. Having these built into the microcontroller frees space on the board and allows for sensors to be plugged directly into the microcontroller. This also prevents the microcontroller from having to load the algorithm from the external storage on startup saving some time. Below is a picture detailing the architecture for the control board of “Luke’s Hand”.

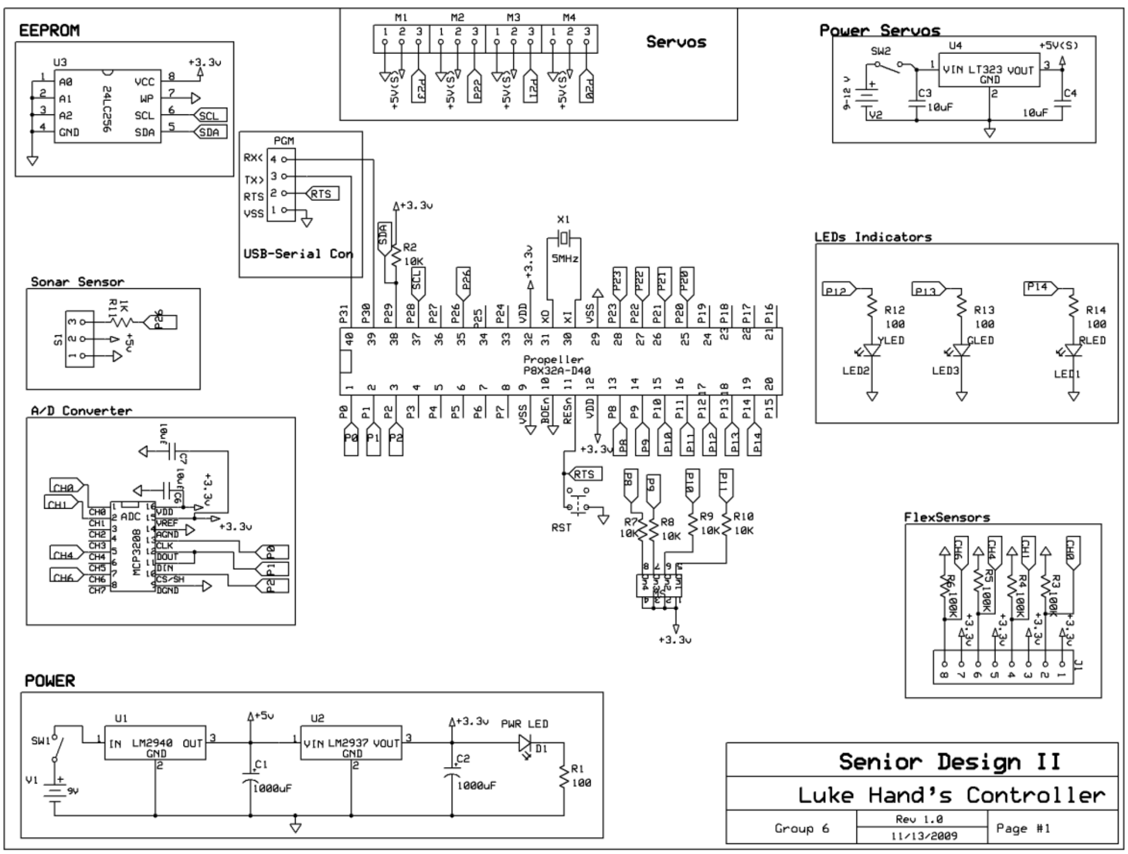


Figure 4.3.2: “Luke’s Hand” control board schematic

Chapter 5 Design Details

5.0 Initial Design Concept

Referencing the Edge 535 (Fig. 4.2.1) above, our design job is to replace the existing simple mechanical control system of the arm using double-pole contact switches to apply either +3V or -3V to each servo motor for clockwise/counterclockwise rotation [1]:

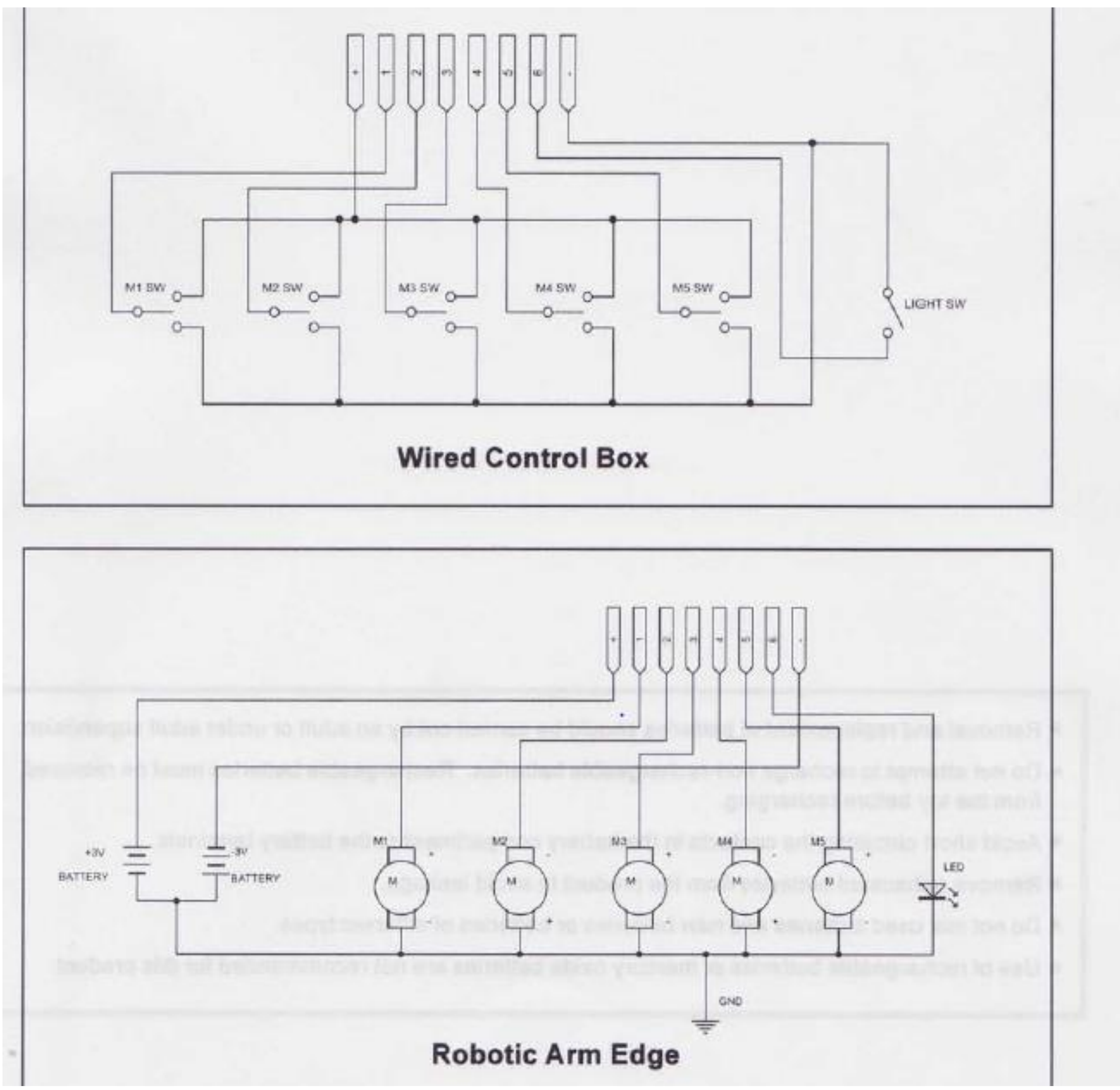


Fig. 5.0.1

Mechanical contact solder-less pc board control box [1]

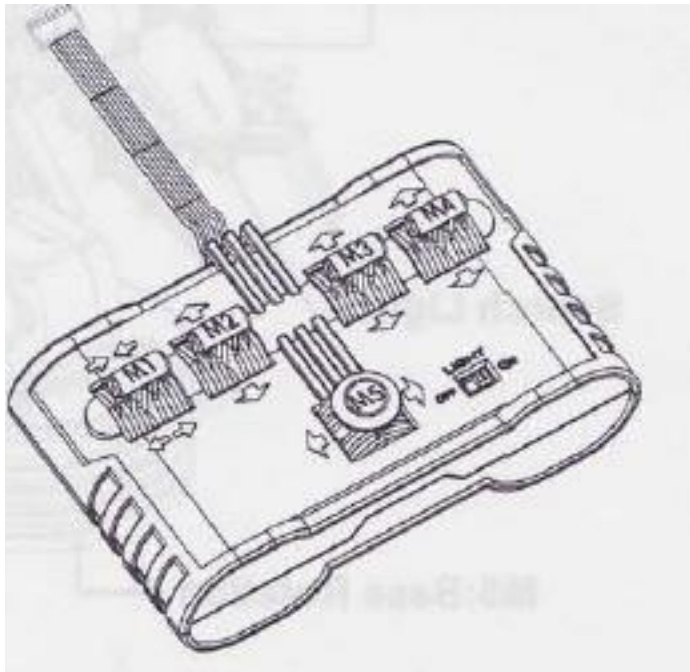


Fig. 5.0.2

attached to arm by eight-pin ribbon cable [1]:

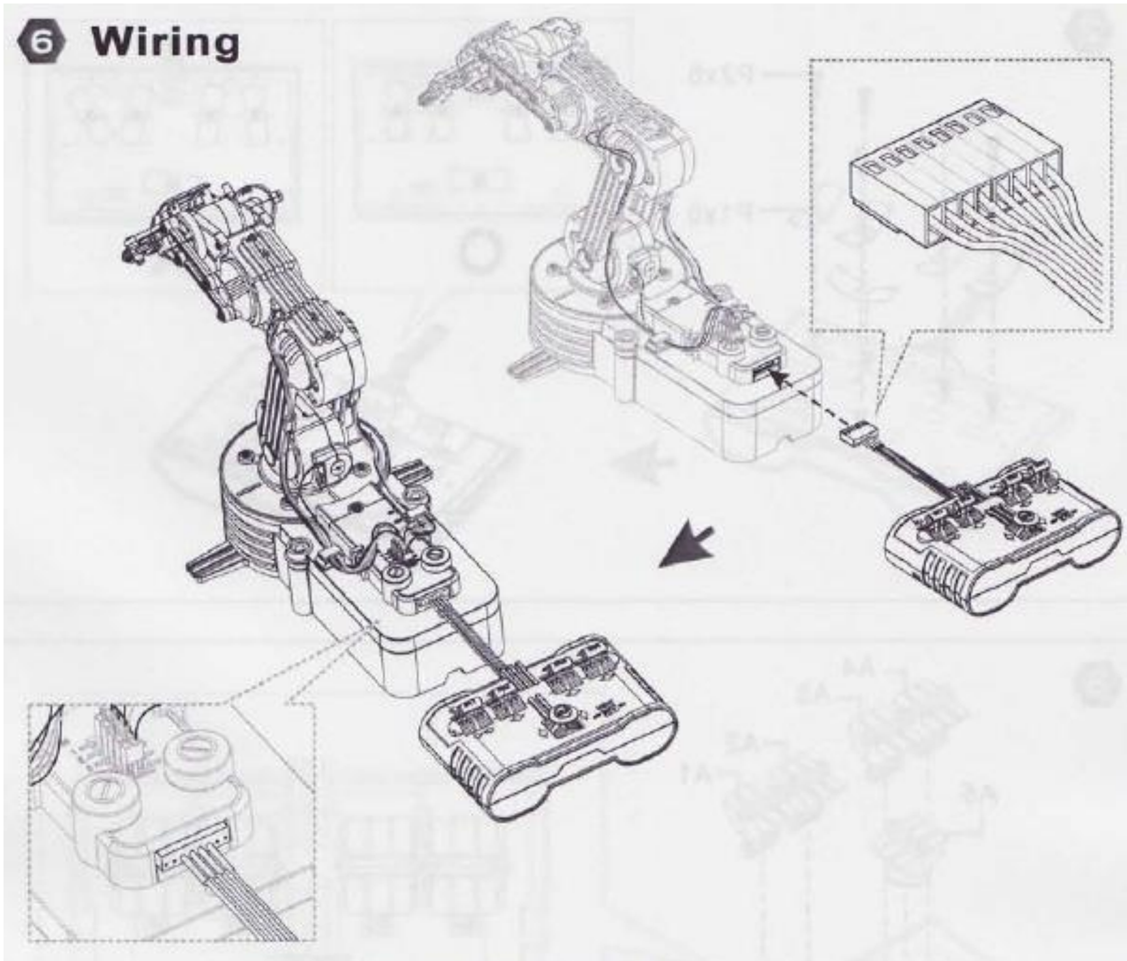


Fig. 5.0.3

the manual control box will be replaced by our own sensor-microcontroller-motor driver system with control signals generated from the sensors located on the operator's arm: [2]

Human arms anatomy diagram, showing bones and muscles while flexing. 2 D digital illustration, On white background.

Fig. 5.0.4

angle sensors – to track linkage motion and, force sensor – to detect opposing fingers to open and close gripper end-effector.

Our system will be virtually open-loop to begin, just the sensors on the operator's arm and hand and a control box to interpret the data signals and send motion control signals to the servo motors on the arm – a virtual open loop control system. The initial performance criteria of the arm are accurate imitation of the operator's arm motion and a smooth tracking motion with minimal time delay. *Time delay occurs in every electro-mechanical system. In most cases it is not noticeable, but in other cases it can render the system unstable. An idealized teleoperator causes a tool (telerobot) to move in space, matching exactly the motions of an operator. All forces imposed on the tool should be accurately reflected back to the operator.* [3]

We might have to put sensors on the robot arm to generate feedback control signals to determine that the robot arm is tracking the human arm motion properly.

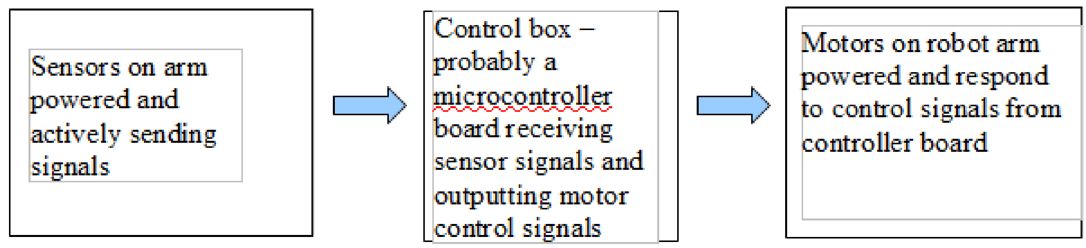


Fig. 5.0.5

Initial open-loop control scenario suitable to test basic functionality of the sensors and motors and control box, followed by:

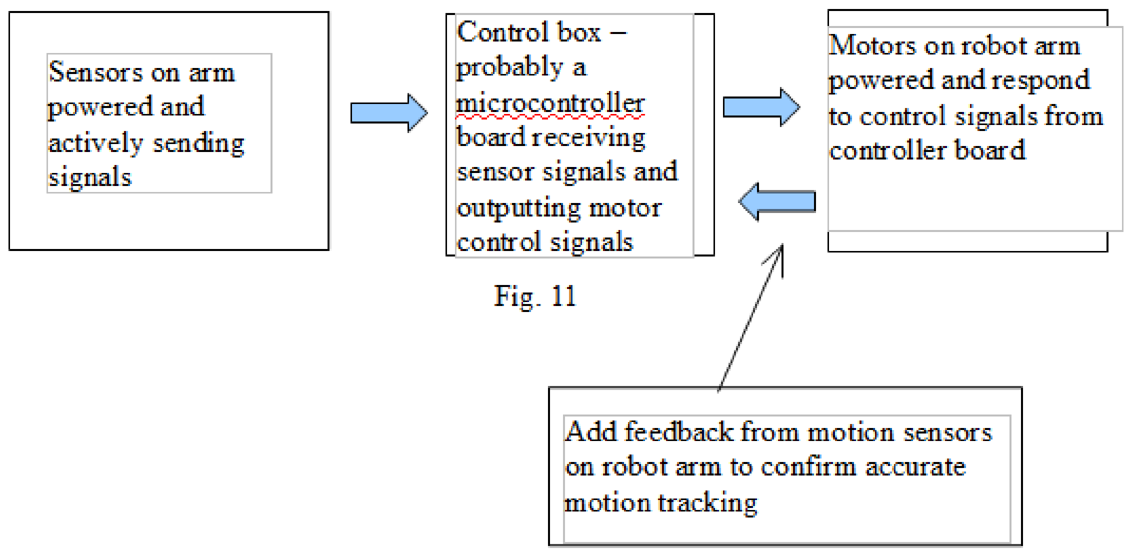


Fig. 5.0.6

A control algorithm is needed to identify the position of the robot linkages at any instant. In traditional robotics a coordinate system is defined and the position of the end-effector – the gripper – is located in an absolute position relative to the base of the motion, which in our case, is the arm base which is currently non-mobile. Because of the close proximity of the operator to the robot arm actual motion – say, across a hazmat glass or plastic barrier window – we hope to be able to use the exteroceptive feedback of the operator's vision to replace proprioceptive feedback (additional sensors on the robots linkages).[4] Initially, this will significantly reduce costs and provide usable functionality. If we add remote operation in the future, we might have to add additional sensors to the robot arm itself.

Our current control algorithm looks like this:

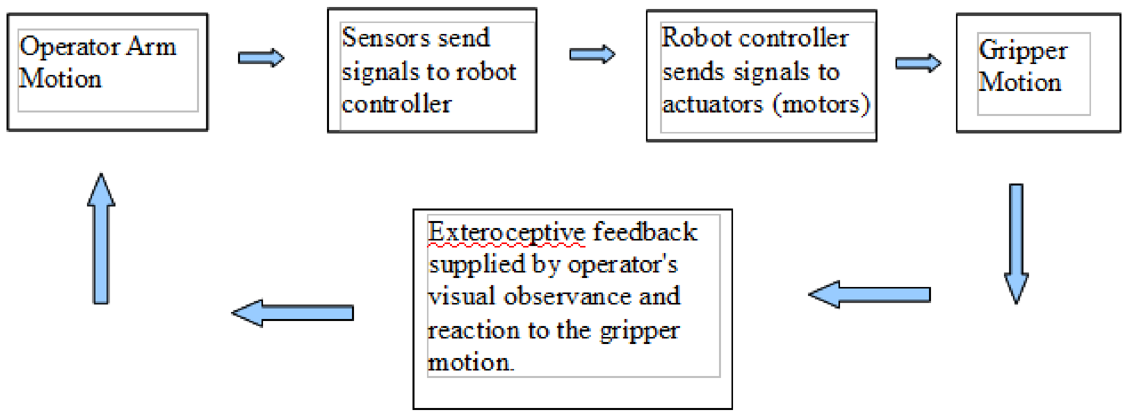


Fig. 5.0.7

References

[1] Accessed Nov. 28, 2012, “OWI Robot Arm Edge: Assembly and Instruction Manual.pdf”, http://www.robotshop.com/pdf/OWI-535_Manual.pdf

[2] Accessed Oct. 28, 2012, http://www.123rf.com/photo_11779669_human-arms-anatomy-diagram-showing-bones-and-muscles-while-flexing-2-d-digital-illustration-on-white.html

[3] Accessed Nov. 19, 2012, “A review of teleoperation control”, fcrar 2003 ppr - fiu famu afrl final ver1may2003pdf,http://dubel.org/misc/docs/fcrar%202003%20ppr%20-%20fiu%20famu%20afrl%20final%20ver%201may2003.pdf

[4] “Robot sensors and transducers”, S.R. Ruocco, Open University Press Robotic Series, 1987

5.1 Design Architecture and Related Diagrams

The robot controller schematic shows how the various components of the board are connected together. The schematic shows the connections between the motor controllers, sensors, microcontroller and also the power supply required to work. All of these components will be located on the same PCB which will be the main PCB for this project and will be used to control the robot. This is the main PCB because all the sensor data will be directed to this board and all the instructions for the motors will come from this board. Table 5.1.1 below shows all the components of this PCB.

Component	Name/Value
Resistors	R1-R5 = 1KΩ
Servo Connectors	M1-M5
Human Sensor Input	S1-S5
Robot Sensor Input	S6-S10
USB 2.0 connector	X6
Microcontroller	Atmel ATxmega128A4U

Table 5.1.1: Components of robot controller

This schematic shows that communication to the device will be through a USB 2.0 connection. This component will be used to program the microcontroller with the control algorithm for controlling the motors on the robot given the input from the sensors on the human. No other component is needed to assist with this data transfer as the ATxmega128A4U has a built in USB interface. This connects to the PR0 and PR1 pins of the microcontroller which are used to program the chip. All of the servo motors on the robot are assigned their own pins as outputs. These pins will produce a PWM (Pulse Width Modulation), a digital output that will be what tells the servo motors how far to turn and in what direction based on the length of the signal.

There are also 10 sensors connectors, each connected to their own pin on the microcontroller. These sensors are connected to the microcontroller as inputs. This allows the microcontroller to know if the operator has changed position and will know what output to provide to which motors. The inputs are connected to pins that are capable of AD conversion which allow the microcontroller to measure the input from the sensors and produce an exact position for output to the motors. The AD conversion may not be needed depending on whether the sensors themselves produce a digital or analog output. If the sensors produce a digital output then they may be moved to a different pin if needed. The Schematic of the circuitry is shown in the figure below.

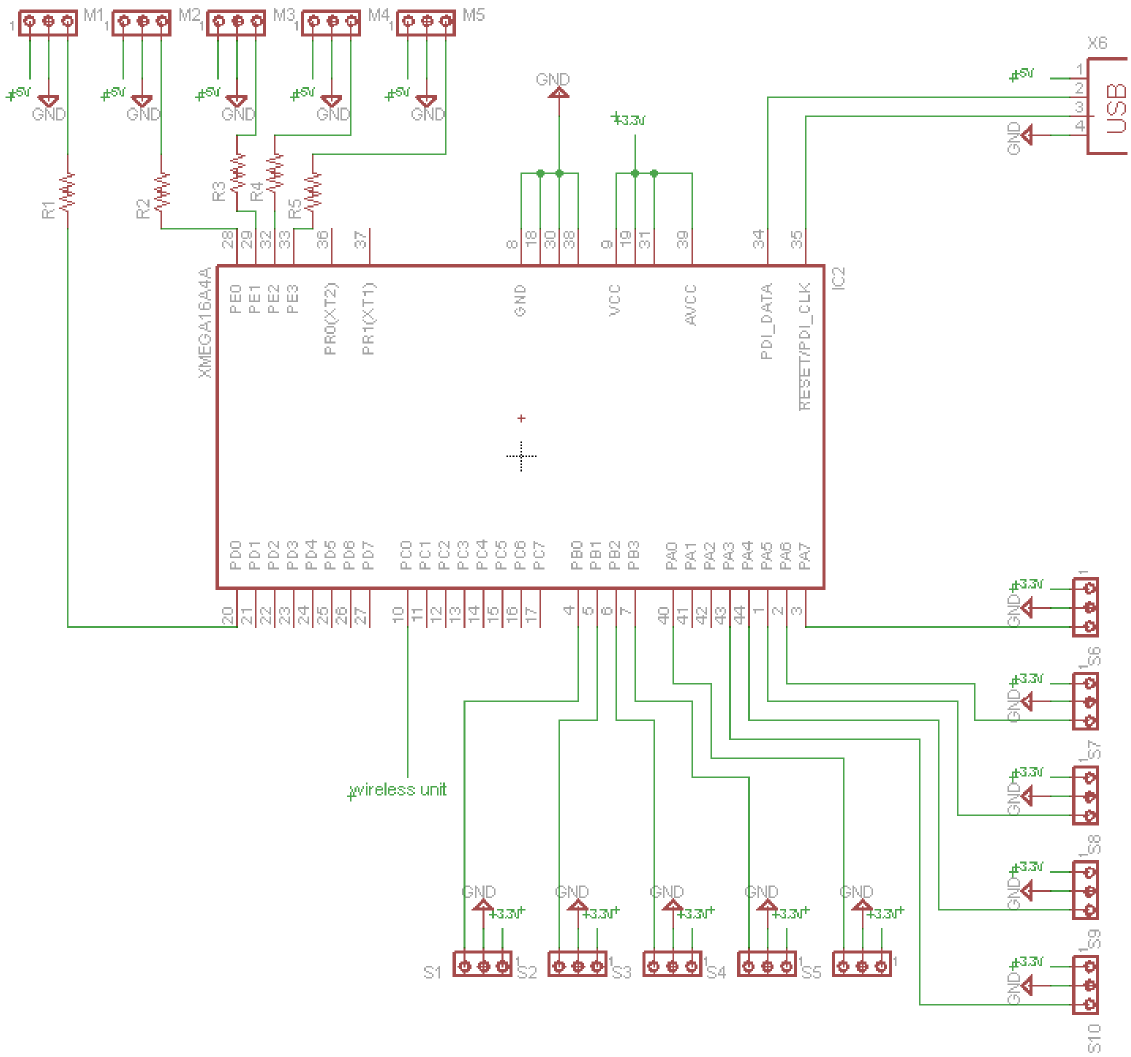


Figure 5.1.1: Schematic for robot arm controller

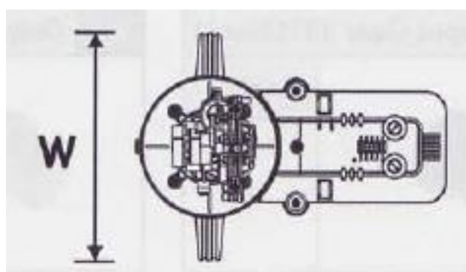
5.1.1 Robot Arm Kinematics

Kinematics is the study of the geometry of motion of linkages – like the shoulder-to-elbow and elbow-to-wrist and wrist-to-end-effector links in our robot arm. Since we are using a pre-designed and manufactured mechanical arm – the OWI 535 Robot Arm Edge – we don't have to do the dynamic analysis of the forces and stresses in the robots links and joints and motor demands, etc...

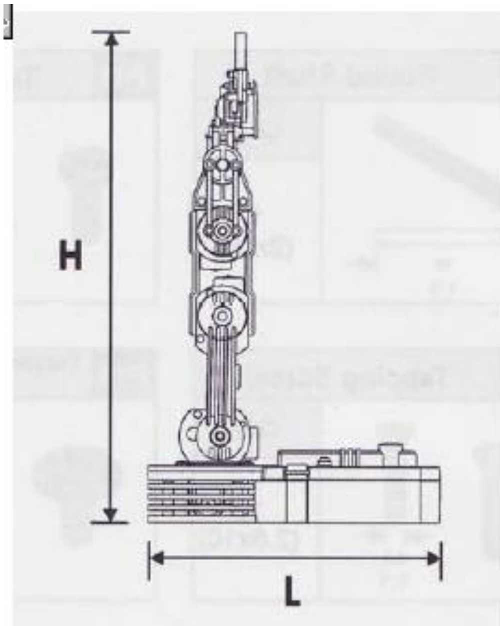
However, our robot does have size dimensions and motion range limitations, which comprise the functional kinematic analysis. Traditional kinematic analysis would require identifying every point reachable by the end-effector relative to the base by some 3- dimensional coordinate system, such as polar, cylindrical, or spherical systems. Gyro sensors are essentially evaluating three axes of angular rotation of a spherical coordinate system. By using the human operator's visual feedback control, our design doesn't have to precisely mathematically identify the exact location of the end-effector constantly the way a robot arm on a production/manufacturing/assembly line would. However, the gyro sensor data has to be accurately interpreted for the proper scaled motion of the robot arm following the human-operator's arm motion. Feedback sensors may have to be applied to the robot arms joints for better control of the slave-motion robot arm.

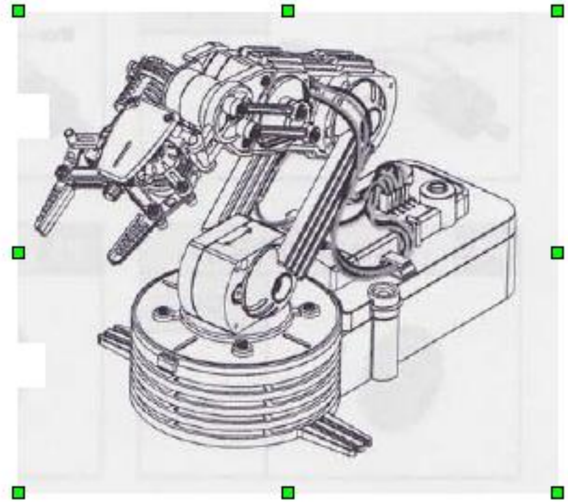
The technical specifications of the OWI 535 Robot Arm Edge kinematics are[1]:

- wrist-to-end-effector range of 120 degrees (vertical rotation only, no perpendicular rotation, which is a possibility with a hardware upgrade to a lynx arm)
- elbow-to-wrist range of nearly 300 degrees (also vertical only)
- base shoulder-to-elbow rotation of nearly 180 degrees vertically and almost 270 degrees of base horizontal rotation
- horizontal reach of 12 inches
- vertical reach of 15 inches
- lifting capacity of 100 grams
- Dimensions: 9" L x 6.3" W x 15" H
- Weight 658 grams



Top view





Side view and orthographic projection

References

[1] Accessed Nov. 28, 2012, “OWI Robot Arm Edge: Assembly and Instruction Manual.pdf”, http://www.robotshop.com/pdf/OWI-535_Manual.pdf

5.2 Motor Subsystems

5.2.1 HOW MOTOR DRIVERS WORK

A motor driver has its own power supply to take a control signal from the processing control and turn the motors on the robot arm on or off and clockwise or counterclockwise, additionally, speed can also be controlled. The basic circuit of a motor driver is the H bridge circuit that functions by alternating the applied voltage to a dc motor from plus to minus voltage by flipping one set of two switches or another set of two switches:

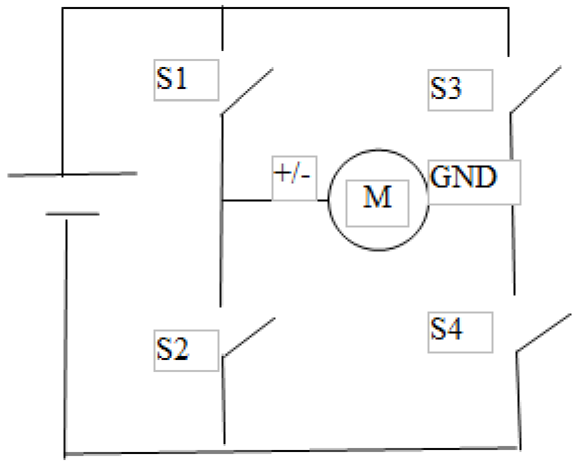


Fig. 5.2.1

If switches S1 and S4 are closed, a positive voltage is applied across the motor. If switches S2 and S3 are closed, a negative voltage is applied across the dc motor and the direction of the motor is reversed. The initial hardware circuit design is currently looking like:

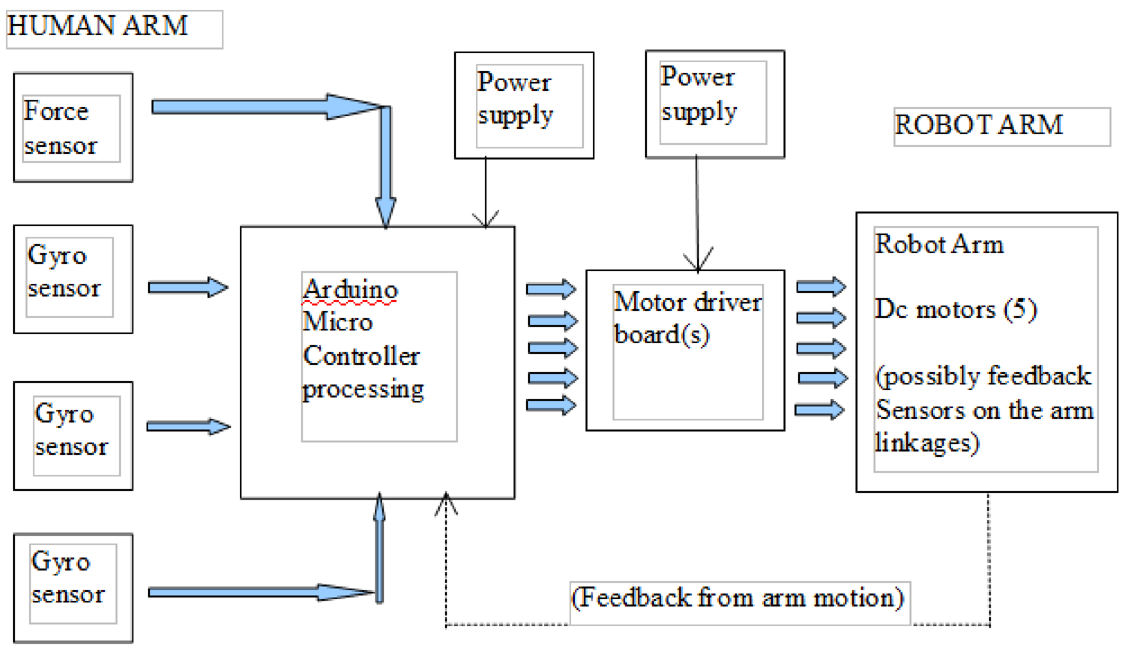


Fig. 5.2.2

The initial motor drivers considered for the design are the TB6612FNG Dual Motor Driver Carrier by Toshiba from Pololu Robots and Electronics ($8.45) and the DRV8833 Dual Motor Driver Carrier by Texas Instruments and also acquired from Pololu ($6.95):

This tiny breakout board for TI's DRV8833 dual motor driver can deliver 1.2 A per channel continuously (2 A peak) to a pair of DC motors. With an operating voltage range from 2.7 to 10.8 V and built-in protection against reverse-voltage, under-voltage, over-current, and over-temperature, this driver is a great solution for powering small, low-voltage motors.[1]

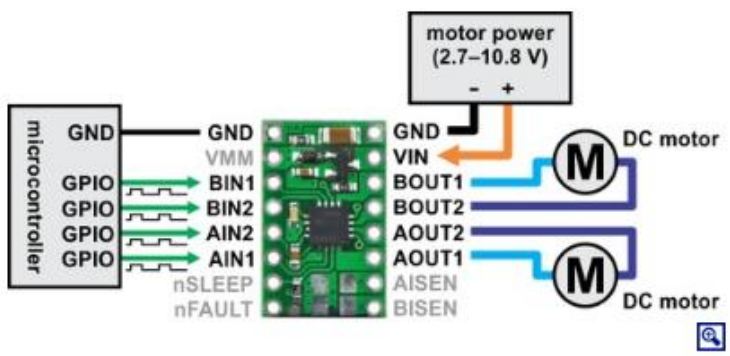
Also the Toshiba board and Pololu:

This tiny board is an easy way to use Toshiba's TB6612FNG dual motor driver, which can independently control two bidirectional DC motors or one bipolar stepper motor. A recommended motor voltage of 4.5 – 13.5 V and peak current output of 3 A per channel (1 A continuous) make this a great motor driver for low-power motors.[2]

Although the Toshiba says it is rated with a minimum of 4.5 volts, and our current robot arm dc motors are rated at 3 volts, online forum advice has communicated that there is about a 1.2 voltage loss across the drive transistors in the H bridge chips so 4.5 volts should also work – ultimately, our design has to drive 5 dc motors – if these two both work at least one more motor driver will be needed.

A generic schematic of the DRV8833 [1]:

Using the motor driver



Minimal wiring diagram for connecting a microcontroller to a DRV8833 dual motor driver carrier.

Fig. 5.2.3

A pinout of the DRV8833 [1]:

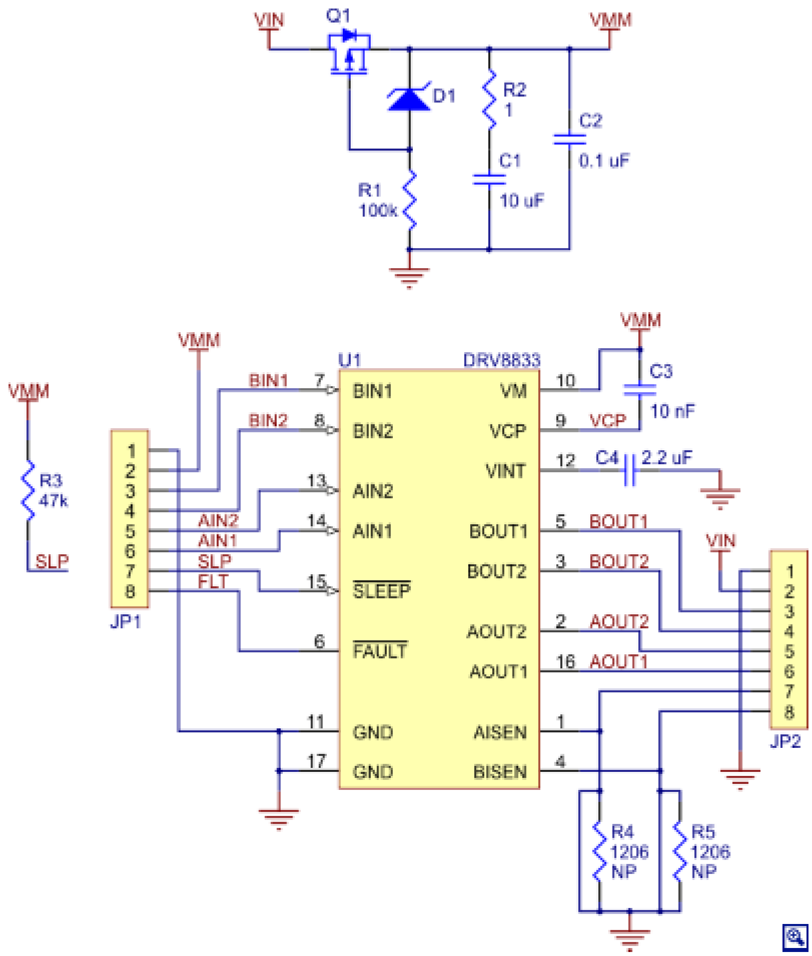
Fig. 5.2.4

Pinout descriptions:

PIN	Default State	Description
VIN		2.7–10.8 V motor power supply connection. Operation with VIN below 5 V slightly reduces the maximum current output.
VMM		This pin gives access to the motor power supply after the reverse-voltage protection MOSFET (see the board schematic below). It can be used to supply reverse-protected power to other components in the system. It is generally intended as an output, but it can also be used to supply board power.
GND		Ground connection points for the motor power supply and control ground reference. The control source and the motor driver must share a common ground.
AOUT1		The motor A half-bridge 1 output.
AOUT2		The motor A half-bridge 2 output.
BOUT1		The motor B half-bridge 1 output.
BOUT2		The motor B half-bridge 2 output.
AIN1	LOW	A logic input control for motor channel A. PWM can be applied to this pin.
AIN2	LOW	A logic input control for motor channel A. PWM can be applied to this pin.
BIN1	LOW	A logic input control for motor channel B. PWM can be applied to this pin.
BIN2	LOW	A logic input control for motor channel B. PWM can be applied to this pin.
nSLEEP	HIGH	Sleep input: when this pin is driven low, the chip enters a low-power sleep mode. (Labeled $\overline{\text{SLP}}$ on the board silkscreen.)
nFAULT	FLOAT	Fault output: driven low in the event of an over-current, over-temperature, or under-voltage condition; floating otherwise. (Labeled $\overline{\text{FLT}}$ on the board silkscreen.)
AISEN		Current sense pin for motor A. This pin is connected to ground and does not function by default, but current limiting can be enabled by making the modifications described below.
BISEN		Current sense pin for motor B. This pin is connected to ground and does not function by default, but current limiting can be enabled by making the modifications described below.

Table 5.2.1

A schematic of the DRV8833 [1]:



Schematic diagram of the DRV8833 dual motor driver carrier.

Fig. 5.2.5

<u>References</u>

[1] Accessed Nov. 17, 2012, http://www.pololu.com/catalog/product/2130

[2] Accessed Nov. 17, 2012, http://www.pololu.com/catalog/product/713

5.2.2 MOTORS

The current robot arm being used for initial testing uses small, cheap dc motors. If the design has success with basic motion requirements of the master-slave robot arm, a hardware upgrade to servo motors and a faster response mechanical arm will be undertaken.

In general, robotics uses three different types of motors to implement linkage translations and rotations – dc motors, stepper motors and servo motors. DC motors operate continuously when power is applied; nominally a forward or clockwise rotation if a positive voltage is applied across the motor and a reverse or counterclockwise motion if a negative voltage is applied across the motor. Stepper motors rotate a few degrees when power is applied. To generate continuous motion with a stepper motor requires pulsing the power to the motor. Servo motors would be the first upgrade choice and are dc motors with position feedback providing accurate continuous angle rotations.

Motor performance specifications include operating voltage, current draw, torque, and speed. The current motors are small dc motors requiring +/- 3 volts and drawing 0.6 to 1.2 amps when running, they are low torque and high-speed and have to be geared (connected to a gearbox) to be useful on the robot arm as they rotate at a few thousands of revolutions per minute or dozens of revolutions per second. Upgrading to servo motors means a robot arm can be used without external (to the motor) gearboxes, which should facilitate faster response time of the tracking motion of the slave robot arm. Servo motors are enclosed in a plastic box that includes the servo, a gearbox, a feedback device (probably a potentiometer) plus drive and control circuitry. [1]

The figure below shows our current motor on the left (dc motor) and our probable future upgrade on the far right (servo motor):

Fig. 5.2.2.1[2]

HS-422 (57 oz. in.) Standard Servo

E-mail this product to a friend

Size = Standard
Servo Erector Set = ASB-xx Standard Scale line
Range = 180°
Voltage = 4.8 - 6.0vdc
Torque = 57 oz.-in.
Weight = 1.66 oz.
Speed = 0.16s / 60 degrees
Full specs

Model Number: S422
Price: $9.99
Weight: 0.14

Quantity: 1 Add to Cart Add to Wish List

3 HS-422 needed for $29.97 total [3]

HS-645MG (133 oz. in.) Standard Servo

E-mail this product to a friend

Size = Standard
Servo Erector Set = ASB-xx Standard Scale line
Range = 180°
Voltage = 4.8 - 6.0vdc
Torque = 133 oz.-in.
Weight = 1.94 oz.
Speed = 0.20s / 60 degrees
Full specs

Model Number: S645MG
Price: $31.49
Weight: 0.12

Quantity: 1 Add to Cart Add to Wish List

1 HS-645MG needed at $31.49 [3]

HS-755HB (183 oz. in.) Quarter Scale Servo

E-mail this product to a friend

Size = Quarter
Servo Erector Set = ASB-2xx Large Scale line
Range = 180°
Voltage = 4.8 - 6.0vdc
Torque = 183 oz.-in.
Weight = 3.88 oz.
Speed = 0.28s / 60 degrees
Full specs

Model Number: S755HB
Price: $27.99
Weight: 0.31

1 HS-755HB needed at $27.99 [3]

Arm upgrade $133 + 5 servos total at $89.45 = $222.45 for mechanical equipment upgrade. Presumably, the electronics will not need any changes other than new voltage and current ratings for the new motors.

References

[1] Accessed Nov. 25, 2012, http://www.arrickrobotics.com/motors.html

[2] Accessed Nov. 25, 2012, http://openmoco.org/node/179

[3] Accessed Nov. 25, 2012, http://www.robotshop.com/

5.3 Sensor Subsystems

5.3.1 I²C SERIAL COMMUNICATION

I²C is a two-wire bi-directional communication technology for connecting integrated circuit chips to exchange data between the circuits. The two wires are serial clock (SCL) and serial data (SDA). Bi-directional means the device can output data and sense voltages on the line at the same time. I²C uses an open-collector or open-drain to generate digital data by a signal input to the base of the bipolar transistor draws the open-collector transistor to ground and then a pull-up resistor puts a high voltage on the collector if the base is open or the signal into the base is zero. The data is clocked by the SCL input. In the open-drain (MOSFET) technology, the open drain is connected to ground if a high voltage is applied to the gate terminal (generating logic 0 output) and is open if low logic is applied. A pull-up resistor generates a logic 1 output if the drain is open. The low state is considered active in open-collector/open-drain pins and the high state is considered non-active. Our design will use 1k ohm pull-up resistors to 5 Volts.

I²C stands for IIC from Inter IC (Integrated Circuits) communication technology created by Phillips Semiconductor corporation in 1982. The technology was standardized in 1992 with version 1.0 and is currently at version 4.0 in 2012. The technology was created to facilitate simpler communication design between different electronic components.

I²C operates on different components by giving each device its own address from a 7-bit address (2^7 = 128) space with 16 addresses reserved for a total of 112 different components addressable in any given electronic system.

I²C requires only two bus lines as mentioned above, each device connected to a micro-controller is on a software addressable bus and is given a unique address. In our case, each sensor will be connected through I²C to the arduino mega. The bus connectivity with I²C includes *collision detection and arbitration to prevent* *data corruption if two or more masters [sensors] simultaneously initiate data transfer.* [1]

Our application requires an 8-bit signal output from the sensor SDA pin and clocked on the SCL pin sent to the arduino micro-controller through the I²C bus. Our design will only require a maximum of 5 sensors, so we won't be running out of usable addresses. The I²C technology simplifies design of a system of components and facilitates fast implementation of a design into hardware. Most of our work will be in programming the micro-controller to properly interpret the sensor data and correctly direct the motors on the robot arm.

NOTE: future reference when building system

It is important to remember when you use two power supplies is to install a common ground between the two. You can still have individual power, you just have to have that common ground, without it, the I2C protocol will probably fail. In my testing, I noticed that my communication failed each time I unplugged either of my USB cables from the Arduino I finally figured out that if I put that common ground in there, I won't lose communication when I disconnect from the PC.

· make sure you have a minimum of a 1K pull-up resistor on each line pulling it to 5 volts. I personally used 1.5K for all my testing, and in my current robot setup.[2]

[1] UM10204.pdf "I2C-bus specification and user manual" Rev. 03 — 19 June 2007 User manual http://www.mcc-us.com/Open-collectorFAQ.htm

[2] Accessed Nov 21, 2012, "Introduction to I2C", http://www.uchobby.com/index.php/2008/09/16/introduction-to-i2c/

5.3.2 Accelerometers and MEMS Gyroscopes

The sensors should consume as little power as possible and be sensitive at frequencies of zero so when the user wishes to make a small correction they won't have to provide a large input relative to the desired output. The sensors will need to provide the orientation of the sections they are attached to and be able to give the speed and direction of which they are moving. Electronic accelerometers, gyroscopes, and compasses are capable of handling the task of sensing the movements of the human arm because of their low power consumption, small size, and their ability to accurately measure their respective forces.

There are three main types of sensing elements used in electronic accelerometers. They are piezoelectric materials, piezoresistive materials, and variable capacitance. Piezoelectric accelerometers have a frequency range of a few Hz to 30 kHz or more. They are readily available in a number of different packaging options and sensitivities, but there inability to accurately measure steady state response does not make them desirable for the application of a human motion sensing/replication where the frequency is in range of 0 Hz to less than 100 Hz. Piezoresistive accelerometers have a frequency response that includes 0 Hz, but are less sensitive than piezoelectric accelerometers. They are typically used in shock measurements. Variable capacitance accelerometers respond at 0 Hz, have good temperature stability, and high sensitivity.

Some accelerometers considered are shown in the chart below.

	ADXL346	ADXL362	BMA180	KXCJK	LIS3DH
2g Capable	Yes	Yes	Yes	Yes	Yes
Resolution	10 bit	12 bit	12 bit	12 bit	16 bit
Output Data Rate Max	1600 Hz	400 Hz	2400 Hz	1600 Hz	1.25 kHz
Sensitivity	3.9 mg/LSB	1mg/LSB	3.4mg/LSB	0.98mg/LSB	1mg/digit
Nonlinearity	±0.5%	±0.5%	±0.15%	±0.6%	Not Provided
Output	I2C	SPI	I2C/SPI	I2C	I2C/SPI
0g Output	Yes	Yes	Yes	Yes	Yes
Bandwidth	3200 Hz	3-200 Hz	1200 Hz	800 Hz	625 Hz

Table 5.3.2.1

Electronic gyroscopes use a vibrating mass and use the Coriolis effect to measure angular velocity. The differences in MEMS gyroscopes consists of the size, shape, resonant frequency of the vibrating mass and sensing elements as well as the way they are driven to their resonant frequency such as piezoelectric, electrostatic, electromagnetic. One does not seem to provide a clear advantage over the other. A gyroscope well be chosen mainly based on having a max angular frequency greater than that of the motor, cost, sensitivity, and drift.

At this point, it is important to consider the alignment between the accelerometer, the gyroscope and the axis they monitor. There will be an initial calibration needed to align the axis of the sensors with the axis of movement. After such a point, it will be integral to the control system to have the gyroscope and the accelerometer of the same axis have keep the same alignment with the axis. Any shifting of the sensors would alter the necessary signal processing to obtain the correct values and could lead to system instability.

There are inertial measurement units (IMU) that are comprised of a accelerometer and a gyroscope. They can have 2 or more degrees of freedom. They are casted on the same IC and typically have a axis alignment error of less than ±1 degree. This would also guarantee only a small compensation is needed (some manufacturers ship IMUs with the alignment error already compensated) when compared to the compensation that would be needed to align individual components soldered on a PCB board. Having the accelerometers and gyroscopes would save space on the board and possibly speed up the transmission of sensor data between the different sections to the control board. It is dependent on the how the IMU is capable of sending data. Many 6 degree of freedom IMUs communicate via a bus line having already done the analog to digital conversion and some signal processing on the data before transmitting it to the controlling processor to free up resources the controlling processor would have had to utilize before moving on to the next task. However, IMUs can be very expensive. For this project, we will build or purchase the necessary parts to create a sensor system that can measure acceleration and angular velocity along all axes of rotation.

The outputs of the accelerometers and gyroscopes may be analog. In order to transmit the information to the control board, there will be some protection from noise. Some options will be using shielding to protect the data from noise or do analog to digital conversion near the sensors and transmit the information digitally so that it is more resistant to interference. The problem with the use of shielded wire for analog outputs is that there will be at least 18 wires ran from the control board. There will be at least 9 wires ran to each sensor. This can led to the difficulty of securing each wire so that it will not be pulled out as the arm makes various movements or introduce noise from its movement. A better option will be to use a bus to transmit the data to the control board.

The robotic arm offers more flexibility in the selection of sensors. Besides the use of accelerometers and gyroscopes, incremental encoders or variable resistors can be used which have some advantages. To process the orientation from an accelerometer, the three axis’s values must be obtained and processed in order to calculate the rotation in a gravitational field. Incremental encoders and variable resistors can be discretized to represent the rotation angle of the joint it is monitoring. A disadvantage of variable resistors is that they must have a constant supply voltage or their output will change with voltage as well as resistance. This can be accounted for in a circuit such as an instrumentation amplifier comparing supply voltage and output voltage; however, this requires additional circuitry at the joint and will have an analog output that will have a greater susceptibility to noise than a digital output. The maximum number of discrete levels will be dependent on the noise level in the signal. Incremental encoders produce digital outputs so will be better suited for environments where electrical interference may be present. Encoders are either magnetic or optical sensing. Optical encoders can have more precision than magnetic encoders, but have the disadvantage of being less resilient to shock. They can output a pulse for each increment or an absolute encoder can be used that will output a a n-bit word representing the rotation angle. An output word of 10 bits can provide an arc length resolution of approximately 1.3 mm on shaft length of 20 cm. This would be acceptable for our application. If encoders are used, optical and magnetic versions will be compared based on cost, resolution, and max shock.

Selection of which sensors to be placed on the robot depend on the orientation of the robotic arm to the user and the environment that it would be used. If accelerometers and gyroscopes are used, the robot will attempt to maintain the same orientation as the human arm regardless of the orientation of the robot. If the robot is on a moving base and the base rotates, the arm will rotate the same amount in the opposite direction to maintain orientation with the user. If encoders are used, the robotic arm orientation will be independent of the base orientation. When there is not input from the user, there will be no compensating rotation of the robotic arm if the base is rotated. This would be particularly useful if a camera is mounted on the arm. If the robot goes out of view, the user will be able to control the arm as if they were standing right behind it. Through the use of gyroscopes and encoders, the arm could even provide stable transport of an item held by the arm. Accelerometers may be able to complete the task, however, this would require more resources to process the information and create a useful output.

Processing Sensor Data

The data from the accelerometers will give their orientation due to gravity, any acceleration due to movement, and vibrations. With a low pass filter, most of the high frequency vibrations can be eliminated. Care must be taken not to have the bandwidth too small or the response to quick movements of the accelerometers will be sluggish and lead to errors in calculations. This may be compensated for with measurements from the gyroscopes, but will not be determined until the testing phase. Each axis will have one output for each of the three axises measured. The three measurements can be combined to produce a vector pointing towards gravity. From this vector, it is possible to calculate rotation of the accelerometer from a predefined orientation. This will prove to be challenging to do on the operator because for every different user or for even the same user the sensors will be in a slightly different position and orientation than they were the time before.

To compensate for unique orientation each time the sensors are placed on the user, a solution is to have the user will have to go through a setup process. If the sensors are mounted on the outside of the arm between the elbow and shoulder, on the outside of the forearm, and on the back of the hand, each axis of the accelerometers can be moved into a position to experience 1g in three different positions of the arm. To account for the rotation of the sensor of the forearm, the wrist can be rotated to give the change in orientation. The fact that there will be accelerometers and gyroscopes on the same PCB would allow the measurement of the distance the joint of rotation. This would require more movements, more setup time and may have the benefit of better interpretation of the intended movement particularly the desired extension of the robotic arm.

By pairing accelerometers and gyroscopes, the signals can complement each other and help counter accumulated errors. Accelerometers alone cannot maintain the correct position of the robotic arm. The acceleration is discretized and will continually accrue. Also, the accelerometers cannot measure constant velocity so one a steady movement will seem like to separate movements to the accelerometer. Gyroscopes will provide the desired speed of the robotic arm from which distance traveled can be measured, but cannot be used for position because like the accelerometers the values are discretized and error will accumulate from one movement to the next. The values for the paired accelerometers and gyroscopes will allow the error to be reset or at least maintain a maximum error. The gyroscopes will also allow for accelerations from movements to be canceled out to keep the accelerometers output vector magnitude at a maximum of 1g which can be an indicator of the correctness of the compensation values.

5.4 Wireless Subsystems

5.4.1 Bluetooth vs. Wifi

When you mention wireless communication two options come to mind, Bluetooth and wifi. Both have similar network programming that can be split into several components as shown the diagram below:

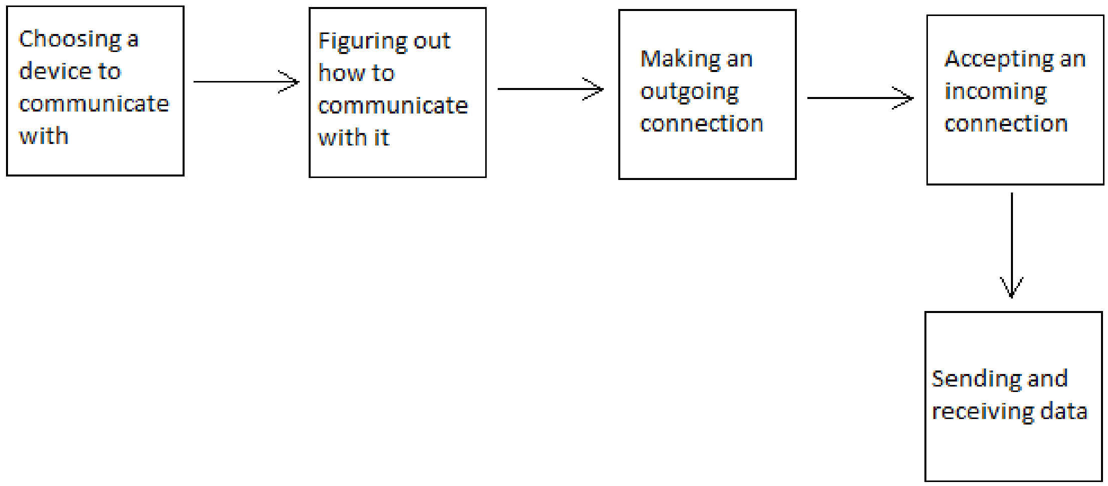


Figure: 5.4.1.1 Diagram showing how Bluetooth and Wifi connect with another network [5]

Though they serve the same general purpose, both are quite different in their standards of operation. Bluetooth is most useful when multiple devices, two or more, that are close together transfer information between one another. Examples consist of telephones, printers, modems and headsets. Wifi on the other hand allows for faster connection and works best for operating full-scale networks.

Overall wifi would seem to have the advantage over Bluetooth as it has better security, more bandwidth, greater range and higher bit rate. [1] The disadvantages though consist of a greater cost, more complex ease of use, higher power consumption and slower speed if multiple devices are connected to the same network. Bluetooth has the ability to have many devices sending and receiving data at the same time and still maintain a good speed of connection without much loss. [2] This faster speed utilized through Bluetooth helps in improving a goal set for this project involving a fast and precise response time of the robot arm to the user movement.

Based off of those factors, and a couple others, the group has decided to control the wireless communication via Bluetooth. For purposes of demonstration the robotic arm does not need to be controlled at a far distance and the arm will be transferring data between a pair of microcontrollers, which is better served through Bluetooth connection.

5.4.2 Bluetooth background

Bluetooth wireless technology, as stated before, is most beneficial for short-range communication with key features based around low power, durability and cheaper cost. Bluetooth was designed to make use of radio frequencies doing away with cables connecting portables and or fixed devices. Bluetooth technology has achieved global acceptance such that any Bluetooth enabled device, almost everywhere in the world, can connect to other Bluetooth devices that are within range. [2]

5.4.2.1 Communications range

A signal can be affected in different ways due to different environments. This leads to an issue of not being able to exactly determine the transmitting range of the Bluetooth device class. The best way to compare a devices' operating range is by measuring another devices' operating range and equate the output power. That is because a higher output power leads to a longer range of use. The three classes of Bluetooth are listed below: [3]

- Class 1 - Long Range
    - Maximum Output Power of 100mW up to 100 meter range
- Class 2 - Medium Range (the most common)
    - Maximum Output Power of 2.5mW up to 10 meter range
- Class 3 - Short Range (very rare)
    - Maximum Output Power of 1mW up to ~1 meter range

Class 2 devices are most common in modern day consisting of cell headsets, laptops, cell phones and any other consumer-level Bluetooth devices. Of the two or more devices communicating, the higher class of such devices determines the properties. Say a class 2 Bluetooth cell phone communicates with a class 1 USB device, the cell phone will limit the range of the Bluetooth radio. Due to their limited range, affecting their usefulness, it is rare to see a class 3 device.

5.4.2.2 Device Name

Bluetooth devices will have a user-friendly name majority of the time. Unlike Internet names, the user for Bluetooth may choose any unique name they desire, without requirement, though it can lead to confusion when multiple devices within the range limit also have the same name. A scenario where someone is sending a file from their phone to another, the user may have to decipher between three or four other phones all with the same title “My Phone”. This name selected by the user is shown rather than the Bluetooth address to identify a device due to people having problems recognizing or remembering 48-bit numbers like 0x024ACE5F3D9B, though the 48-bit address though is what is used in actual communication. The user overall will choose a device that matches a name given by another user they wish to connect with.

5.4.2.3 Radio Frequencies and Channel Hopping

What separates Bluetooth from other technologies is that it takes the 2.4 GHz band it occupies and divides into 79 channels which utilize channel hopping techniques. These channel hopping techniques create Bluetooth devices that are always changing the frequencies they’re transmitting and receiving on. This is done in a somewhat random order so that no channels are used more than any others. A Bluetooth device that is actively changing channels, switches every 625 microseconds (1600 times per second). Two Bluetooth devices must hop channels together while communicating so that they will always be transmitting and receiving on the same frequencies.

5.4.2.4 Bluetooth networks - piconets, masters, and slaves

A piconet is formed when two or more Bluetooth devices are transferring data with one another while channel hopping, as stated above, occurs. A piconet can have a maximum number of 8 total devices connected at one time. This brings up the question of how all the devices know to be on the same page. A model of communication known as master and slave is employed to resolve this issue. A single device on every piconet is chosen as the overall master, and this master has two functions. First it relays to all the other devices, now know as slaves, the frequencies to use and the slaves all follow. The second function makes sure that the devices take turns communicating in an ordered manor. [5]

5.4.3 Methods to connect microcontroller to Bluetooth

There are many different Bluetooth devices available on the market and the majority are simple plug and play devices. You can simply just connect one directly to your microcontroller by either an Rs232 serial interface, a UART or a USB connection. Buying a microcontroller that already has a built in serial interface and then connecting it to a Rs232 Bluetooth adaptor would be the easiest method. [4]

5.4.3.1 UART

UART is an acronym for a Universal Asynchronous Receiver and Transmitter, and is used for communication with serial input/output devices. Serial transmission reduces to cost and complexity of wiring at the expense of speed, and this is a well desired trade-off for a many applications.

UARTs provide serial data through asynchronization in two forms, one being parallel-to-serial and the other serial-to-parallel data conversion for both transmission and reception. These functions are necessary for converting the serial data stream into parallel data that is required with digital systems. Synchronization for the serial data stream is accomplished by adding start and stop bits to the transmit data to form a data character. Data integrity is insured by attaching a parity bit to the data character. The parity bit is checked by the receiver for any transmission bit errors.

To a host system, the UART appears as an 8-bit input and output port that it can read from and write to. Whenever the host has data to be sent, it just sends these data to the UART in byte format whenever the UART receives data from another serial device it will buffer these data in its FIFO, then it will indicate the availability of these data to the host through an internal register bit, or through a hardware interrupt signal. [9]

Full and Half Duplex

When a UART can simultaneously send and receive data, this is defined as Full Duplex. When a device must pause either transmitting or receiving to perform the other is known as Half Duplex, meaning the UART cannot send and receive data simultaneously.[7]

5.4.3.2 Serial Communication

Serial communication is the most common low level protocol for communicating between two or more devices. Normally, one device is a computer, while the other device can be a modem, a printer, another computer, oscilloscope or a function generator. The serial port transmits and receives bytes of information one bit at a time. These bytes are transmitted using either a binary format or in ASCII format.

Over the years, several serial port interface standards for connecting computers to peripheral devices have been developed. These standards include RS-232, RS-422, and RS-485 — all of which are supported by the serial port object. Of these, the most widely used standard is RS-232, which stands for Recommended Standard number 232.

Primary communication is accomplished using three pins: the Transmit Data pin, the Receive Data pin, and the Ground pin. [3]

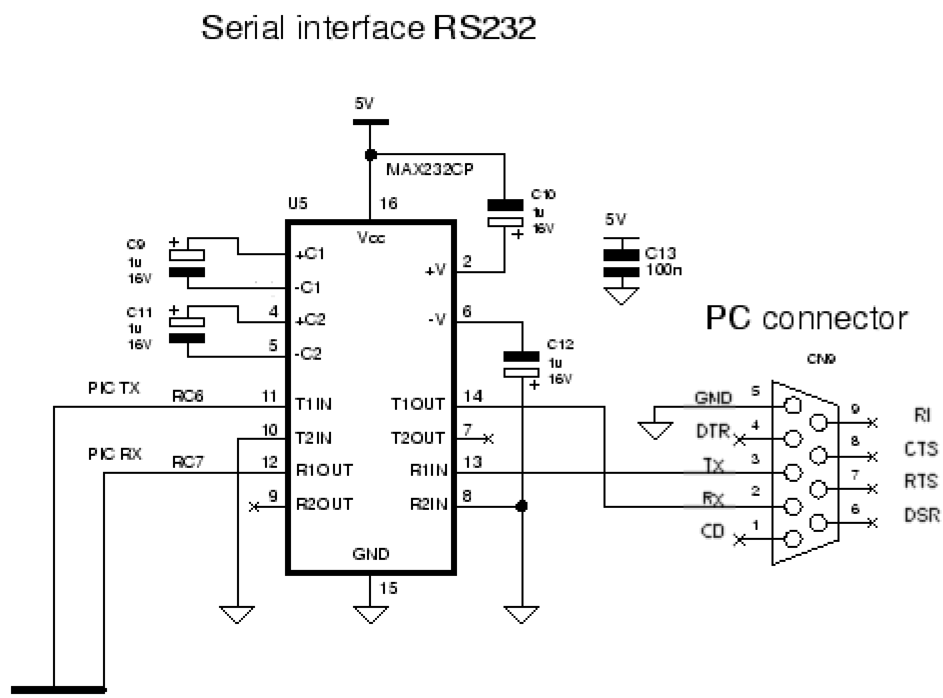


Figure: 5.4.3.2 shows a serial connection to/from the PC that lets RS232 work between it and a microcontroller [8]

Two basic types of serial communications are synchronous and asynchronous communications.

5.4.3.2.1 RS-232

The RS232 standard describes a communication method where information is sent bit by bit on a physical channel. The information must be broken up in data words. On PC's a length of 7 and 8 bits are most common but 5 and 6 are also available. For proper transfer additional bits are added for synchronization and error checking purposes, as described in the previous section. It is important, that the transmitter and receiver use the same number of bits, if not the data word may be misinterpreted, or not recognized at all. With RS232, the line voltage level can have two states. The on state is also known as mark, the off state as space. No other line states are possible. When the line is idle, it is kept in the mark state.

Start bit

RS232 defines an asynchronous type of communication. This means, that sending of a data word can start on each moment. If starting at each moment is possible, this can cause issues for the receiver in knowing which bit is the first receive. A solution to this issue is that each data word will be started with a start bit. The start bit is always identified by the space line level. Because the line is in mark state when idle, the start bit is easily recognized by the receiver.

Data bits

Directly following the start bit, the data bits are sent. A bit value 1 causes the line to go in mark state; the bit value 0 is represented by a space. The least significant bit is always the first bit sent.

Parity bit

A parity bit is used for error checking purposes. The idea is to add an extra bit to the data word automatically. The transmitter calculates the value of the bit depending on the information sent, while the receiver performs the same calculation and verifies if the actual parity bit value relates to the calculated value.

Stop bits

Suppose that the receiver has missed the start bit because of noise on the transmission line. It started on the first following data bit with a space value. This causes garbled date to reach the receiver. A mechanism must be present to resynchronize the communication such a method is called framing. Framing means, that all the data bits and parity bit are contained in a frame of start and stop bits. The period of time lying between the start and stop bits is a constant defined by the baud rate and number of data and parity bits.

The start bit has always space value, the stop bit always mark value. If the receiver detects a value other than mark when the stop bit should be present on the line, it knows that there is a synchronization failure. This causes a framing error condition in the receiving UART. The device then tries to resynchronize on new incoming bits. To resynch, the receiver scans the incoming data for valid start and stop bit pairs. This will work as long as there is enough variation in the bit patterns of the data words. If data value zero is sent repeatedly, resynchronization is not possible for example. [10]

5.4.4.2.1 Synchronous serial Transmission

In synchronous communications, the two devices initially synchronize themselves to each other and to stay in sync they send characters continually. Even as data is not really being sent, a constant flow of bits allows each device to know where the other is at any given time. This means that each character sent is either actual data or an idle character. Synchronous communications allows faster data transfer rates than asynchronous methods, because additional bits to mark the beginning and end of each data byte are not required. [6]

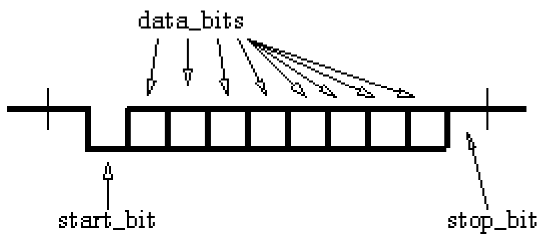
With synchronous communication, a clock or trigger signal must be present which indicates the beginning of each transfer. The absence of a clock signal makes an asynchronous communication channel cheaper to operate and fewer lines are needed in the cable. A disadvantage is that the receiver can start at the wrong moment receiving the information, which leads to resynchronization adding an increase in the amount of time to execute, as well as all data that is received during this period will be lost. Another disadvantage is that extra bits are needed in the data stream to indicate the start and end of useful information. These extra bits take up bandwidth. [10]

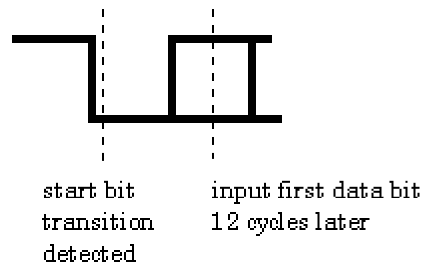
5.4.4.2.2 Asynchronous Serial Transmission

The Baud rate is defined by how many bits are sent per second, but baud only has meaning if the two communicating devices have a synchronized clock. For example, if a microcontroller crystal has a slight deviation of .1 second, meaning it thinks 1 second is actually 1.1 seconds long. This could cause your baud rates to be mismatched and fail. One solution would be to have both devices share the same clock source, but the tradeoff here is you must add extra wires.

Asynchronous transmission allows data to be transmitted without the sender having to send a clock signal to the receiver. Instead, the sender and receiver must agree on timing parameters in advance and special bits are added to each word which are used to synchronize the sending and receiving units.

When a word is given to the UART for asynchronous transmissions, a bit called the start bit is added to the beginning of each word that is to be transmitted. The start bit is used in alerting the receiver that a word of data is about to be sent, which then forces the clock in the receiver to synch with the other clock in the transmitter. These two clocks must be accurate enough to not have the frequency drift by more than 10% during the transmission of the remaining bits in the word.





When the entire data word has been sent, the transmitter may add a Parity Bit that the transmitter generates. The Parity Bit may be used by the receiver to perform simple error checking. Then at least one Stop Bit is sent by the transmitter. When all of the bits in the data word have been collected in the receiver it may check for the Parity Bits, where both sender and receiver must agree on whether a Parity Bit is to be used in the first place, and then the receiver looks for a Stop Bit. If the Stop Bit does not appear when it is supposed to the UART considers the entire word to be garbled and will report a Framing Error to the host processor when the data word is read. The usual cause of a Framing Error is that the signal was interrupted or both the sender and receiver clocks were not running at the same speed.

Regardless of whether the data was received correctly or not, the UART will automatically discard the parity, stop and start bits. If the sender and receiver are configured identically, these bits are not passed to the host. If another word is ready to be sent, the start bit for the new word can be sent as soon as the stop bit for the previous word has been sent. Long story short, the asynchronous data is 'self-synchronizing'. [7]

5.5 Controller Subsystems

As mentioned before, the main control board for this project will be responsible for reading the input from the sensors, translating to a position for the motors and then outputting that position to the motors. These actions are possible due to several subsystems on the control board.

5.5.1 Microcontroller

A major piece on the control board is the microcontroller. As explained in section 4.2.1, the microcontroller is responsible to interpreting the data from the sensors and computing motor position. Microcontrollers are essentially whole computers contained on a single integrated circuit that contain a processor core, memory, and programmable input and outputs.

The microcontroller that will be used for this project is the ATxmega128A4U. This microcontroller is made of many different parts that allow it to interpret the data passed to it and compute outputs. The block diagram of the ATxmega128A4U is shown below in figure 5.5.1.

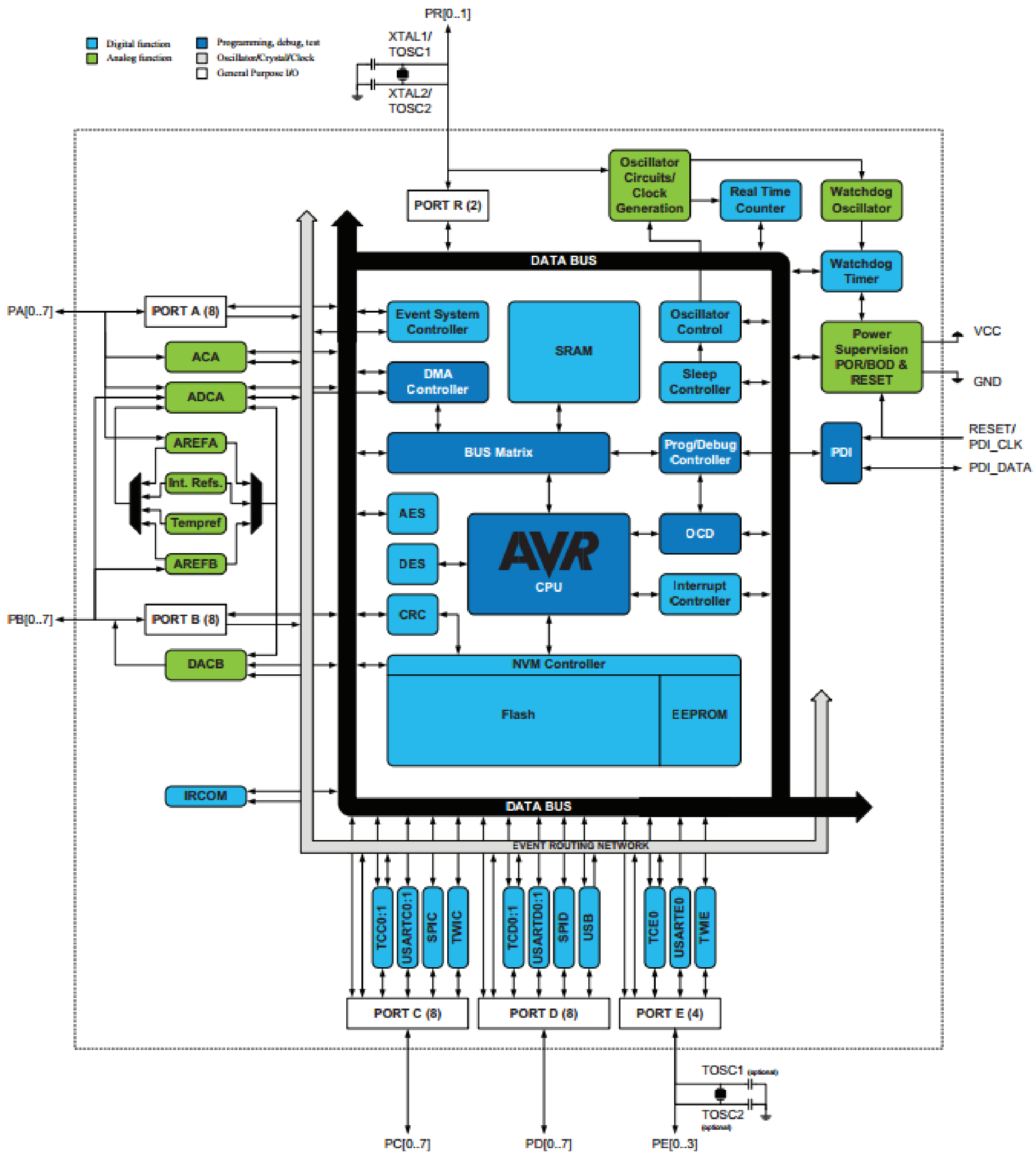


Figure 5.5.1: Block diagram of the ATxmega128A4U microcontroller

One of these parts is the analog to digital converter. This component is a vital part of allowing the microcontroller to do its job for this project. The A/D converter takes in a signal from an input port, measures the voltage of that signal and converts it to a digital value. This allows the microcontroller to communicate with devices that output voltage levels as data on one wire instead of using several wires to produces a digital value. This subsystem of the microcontroller not only saves I/O pins but also saves space on the PCB board as an external A/D converter is not necessary when one is built in.

A second subsystem of the microcontroller is the PWM (pulse width modulator). This feature of the microcontroller allows it to produce a signal that is used to control external objects such as power converters and motors. PWMs are a digital signal in the form of a square wave with a varying duty cycle. The length of the signal is used by the receiving device to determine what operation to take, in a motor's case what position to move to. This feature allows the microcontroller to control the motors in the robot and eliminates the need for a separate motor controller on the control board.

5.5.2 Power Supply

A pivotal component of the control board is how to provide power to all of the different parts. Most parts of this project are powered directly from the control board including the microcontroller and the motors in the robot. One problem with supplying power to the different components of the control board is that some components operate at voltages that are too high or too low for the other components to operate at. This requires control over the voltage from the power source and one way to do this is through voltage regulators.

Voltage regulators allow the maintenance of a constant stable voltage level. This is required for the control board as any electronic noise from the power supply could lead to the robot not properly mimicking the human operator. Voltage regulators are also able to perform DC-DC conversion allowing a power supply that has a voltage too high for certain components of the control board to still be used to power those components as the voltage regulator can lower the voltage from the supply to the required level. This would allow a single battery to power all of the components of the control board as long as voltage regulators are used to lower the voltage to what the components can handle. The use of a second battery and voltage regulator will be used to power the motors however to prevent any accidental overload of other components on the control board.

5.5.3 USB connector

A final subsystem of the control board is the USB connector. This piece allows the control board to communicate with a computer though the microcontroller. For this project this connection will be used to program the microcontroller and possibly monitor the data being passed to and from it during testing to ensure function and accuracy. The USB connector will communicate with the microcontroller through the PDI pins which are connected to the program and debug interface of the microcontroller. These pins allow the programming of the memory in the microcontroller and relay debug information to the IDE on the computer during testing.

Chapter 6 Design Summary

Our design can be quickly summarized with two graphics. One showing the conceptual schematic, and the second showing our current design of the required printed circuit board.

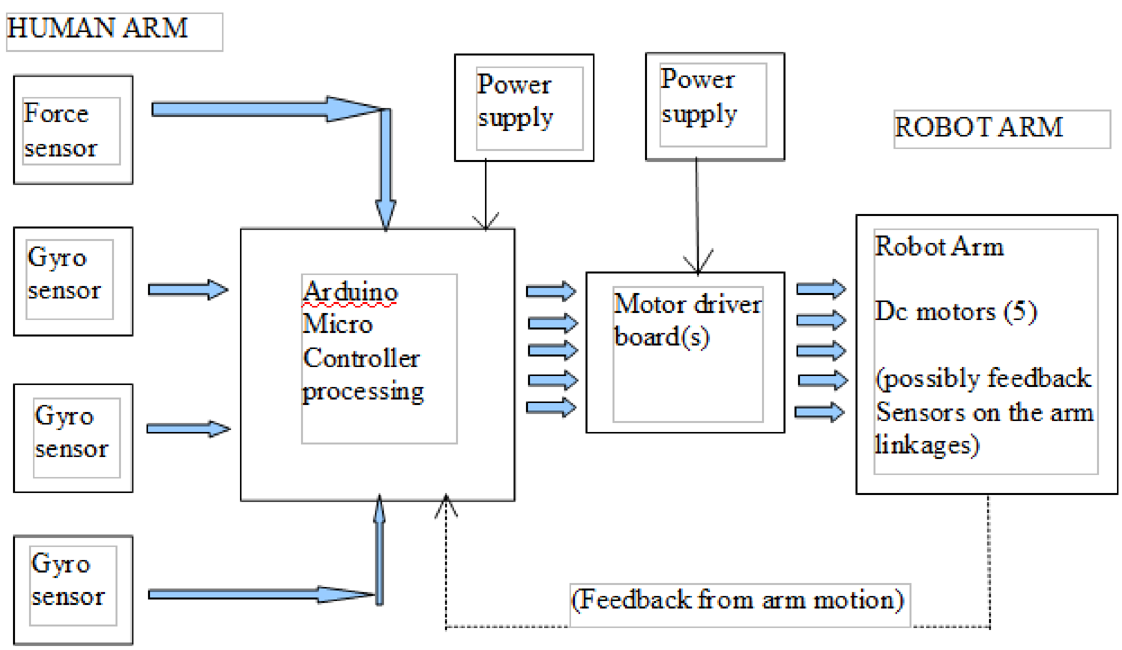


Fig. 6.1 (also Fig. 5.2.2)

and,

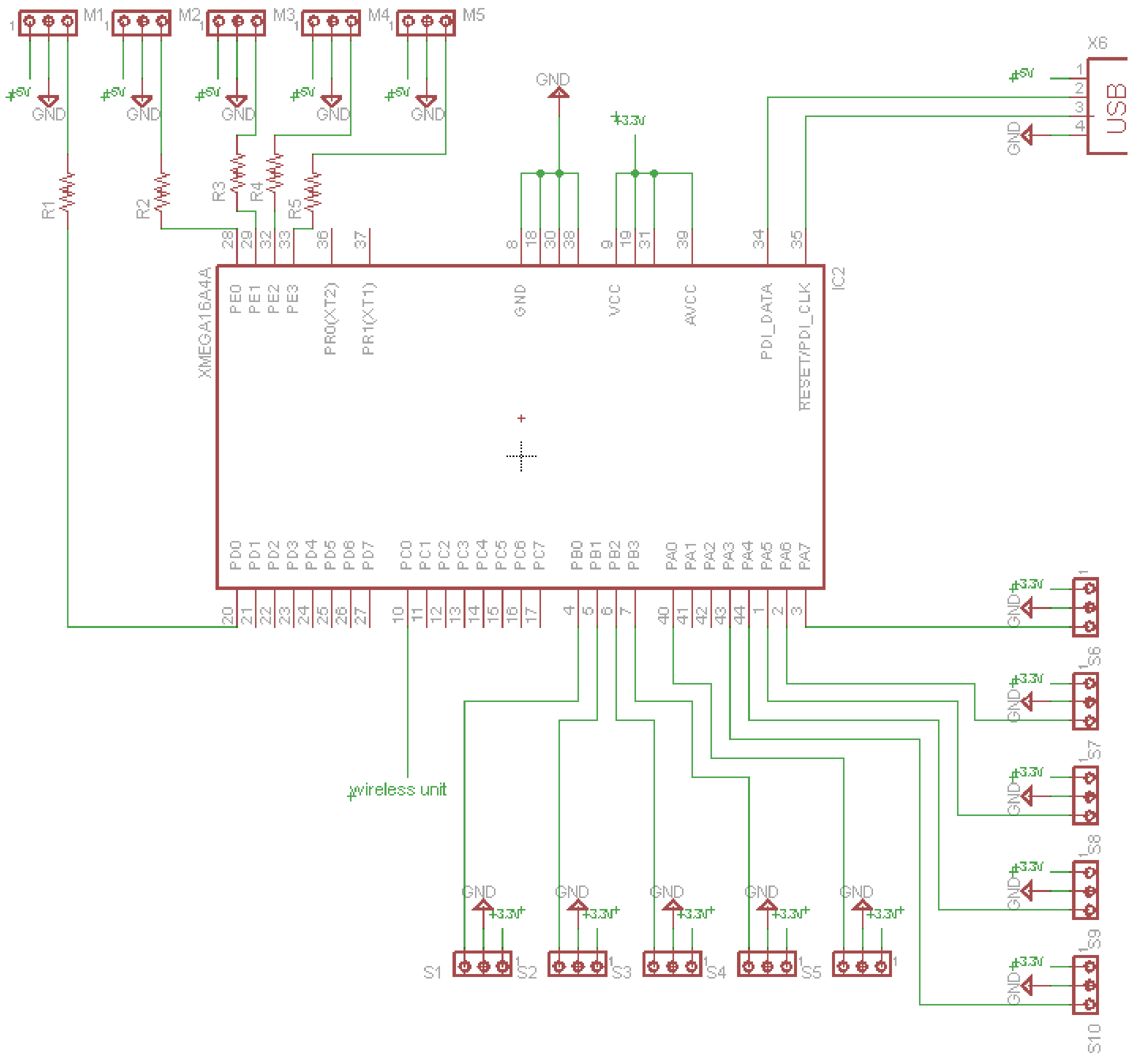


Fig. 6.2 (also Fig. 5.1.1)

Chapter 7 Build Plan

7.1 Parts Acquisition

Enough parts have already been obtained to begin working during the break between terms on a prototype. Parts already acquired include:

1 L3GD20 3-axis gyro

1 Arduino Mega (microcontroller board)

1 DRV8833 Dual Motor Driver Carrier

1 TB6612FNG Motor Driver

1 FSR 1.5 inch square

1 female jumper cables and header pins

1 small breadboard

7.2 Prototyping

The parts already acquired are sufficient to create a sensor-microcontroller-motor driver system to drive one motor of the robot arm, or perhaps two at a time. If our design prototypes successfully, we can then send the printed circuit board design out to be manufactured.

7.3 Hardware Assembly

As mentioned in Milestones we have already successfully constructed an initially usable robot arm – the OWI 535 Edge. Our next hardware assembly is the printed circuit board, which will be sent out to be manufactured after we have established sufficient design effectiveness from prototyping. A breadboard arrangement will be used for the first prototype testing.

7.4 Software Integration

Most of our project involved trying to successfully manipulate the sensor data through the microprocessor on the control board to drive the motors on the robot arm. Programming integration began immediately with the first breadboard prototype test of one gyro sensor driving one motor of the robot arm. This was achieved using an Aurduino Mega 2560. From there a pcb board was designed to allow an atmel xmega 128a4u microcontroller to be programmed to control the arm.

Chapter 8 Project Testing

8.0 Testing checklist

Testing Procedure Hardware

H/W device	Test	Pass	Fail	Comments
OWI Robot Arm	Basic mechanical operation with manual control box	X		Operates properly but response time looks slow
Gyro sensor	Sending data from simple test motion to arduino successfully?			Mechanical test only – does the device appear to be working mechanically?
Pressure sensor	Sending data from test to arduino successfully?			Mechanical test only – does the device appear to be working mechanically?
Arduino	Functioning properly?			Mechanical systems check - no H/W problems?
Motor Driver	Sending drive commands to motors successfully?			Mechanical test only – does the device appear to be working mechanically?

Testing Procedure Software

S/W function	Test	Pass	Fail	Comments
Arduino systems check	Computing and communications systems integrity good?			Any problems detected?
Programming Gyro in the Arduino	Data communication effective and processed accurately for motion analysis?			
Programming Motor Driver in the Arduino	Proper drive commands sent to the motor(s) based on gyro input?			
Programming Pressure sensor in the Arduino	Proper signals sent to the arduino for sample test pressures?			Signals appear to be proportional to test pressures?
Robot Arm motion: 1. base rotation	Is gyro data for base rotation properly processed into the robot arm motion?			
2. link 1: sholder-to-elbow vertical rotation	Is gyro data for link 1 vertical rotation properly processed into the robot arm motion?			
3. link 2: elbow-to-wrist vertical rotation	Is gyro data for link 2 vertical rotation properly processed into the robot arm			

	motion?			
4. link 3: wrist vertical rotation	Is gyro data for link 3 vertical rotation properly processed into the robot arm motion?			
5. link 4: end-effector open and close	Is pressure sensor data for link 4 lateral translation properly processed into the robot arm motion?			

8.1 Testing Environment

TESTING

ROBOT CALIBRATION

Trying to write a section on Robot Testing immediately encounters ISO (International Standards Organizations) 9283 for $162[5], and ANSI/RIA R15.05-2-1992 for $67[6]. Subsequent hunting the internet gathers some information - *ISO 9283 sets out a method whereby both accuracy and repeatability can be measured. Typically a robot is sent to a taught position a number of times and the error is measured at each return to the position after visiting 4 other positions. Repeatability is then quantified using the standard deviation of those samples in all three dimensions.* [1]

Although the industrial website (www.strobotics.com/repeat.htm) references Wiki directly, I tried the original source listed on Wiki for the definitions of accuracy and precision and found the *International vocabulary of metrology — Basic and general concepts and associated terms* [2]:

2.13 (3.5)

measurement accuracy

accuracy of measurement

accuracy

closeness of agreement between a measured quantity value and a true quantity value of a measurand

NOTE 1 The concept ‘measurement accuracy’ is not a quantity and is not given a numerical quantity value. A measurement is said to be more accurate when it offers a smaller measurement error.

NOTE 2 The term “measurement accuracy” should not be used for measurement trueness and the term measurement precision should not be used for ‘measurement accuracy’, which, however, is related to both these concepts.

NOTE 3 ‘Measurement accuracy’ is sometimes understood as closeness of agreement between measured quantity values that are being attributed to the measurand.

2.15

measurement precision

precision

closeness of agreement between indications or measured quantity values obtained by replicate [repeatability] measurements on the same or similar objects under specified conditions

NOTE 1 Measurement precision is usually expressed numerically by measures of imprecision, such as standard deviation, variance, or coefficient of variation under the specified conditions of measurement.

NOTE 2 The ‘specified conditions’ can be, for example, repeatability conditions of measurement, intermediate precision conditions of measurement, or reproducibility conditions of measurement (see ISO 5725-3:1994).

NOTE 3 Measurement precision is used to define measurement repeatability, intermediate measurement precision, and measurement reproducibility.

NOTE 4 Sometimes “measurement precision” is erroneously used to mean measurement accuracy

However, it appears a major part of our project requires proper error measurement, at least about a zero reference, of the gyroscope data so the primary source is referenced again [2]:

2.16 (3.10)

measurement error

error of measurement

error

measured quantity value minus a reference quantity value

NOTE 1 The concept of ‘measurement error’ can be used both

a) when there is a single reference quantity value to refer to, which occurs if a calibration is made by

means of a measurement standard with a measured quantity value having a negligible measurement uncertainty or if a conventional quantity value is given, in which case the measurement error is known, and

b) if a measurand is supposed to be represented by a unique true quantity value or a set of true quantity values of negligible range, in which case the measurement error is not known.

NOTE 2 Measurement error should not be confused with production error or mistake.

From what I have read so far the difficulty in our project appears to be that the MEMS gyroscopes' have a changing reference data value and require constant re-calibration of the 'measured quantity value minus a reference quantity value ' for the zero position reference.

Returning to the industrial website and merrily cutting and pasting something that I can understand finds [2]:

From Wikipedia:

Accuracy – is how closely a robot can reach a commanded position. When the absolute position of the robot is measured and compared to the commanded position the error is a measure of accuracy. Accuracy can be improved with external sensing for example a vision system [or human operator] *or Infra-Red.*

Repeatability [precision] *- is how well the robot will return to a programmed position. This is not the same as accuracy. It may be that when told to go to a certain X-Y-Z position that it gets only to within 1 mm of that position. This would be its accuracy which may be improved by calibration. But if that position is taught into controller memory and each time it is sent there it returns to within 0.1mm of the taught position then the repeatability will be within 0.1mm.*

Accuracy and repeatability are different measures. Repeatability is usually the most important criterion for a robot. ISO 9283 sets out a method whereby both accuracy and repeatability can be measured. Typically a robot is sent to a taught position a number of times and the error is measured at each return to the position after visiting 4 other positions. Repeatability is then quantified using the standard deviation of those samples in all three dimensions.

I am not certain at the present, how to interpret and analyze our proportional master-slave motion application compared to the traditional *absolute position of the robot is measured and compared to the commanded position* robot motion analysis.

Also, from Wikipedia, the best and simplest (also the only) graphic I have ever seen to explain accuracy and precision [3]:

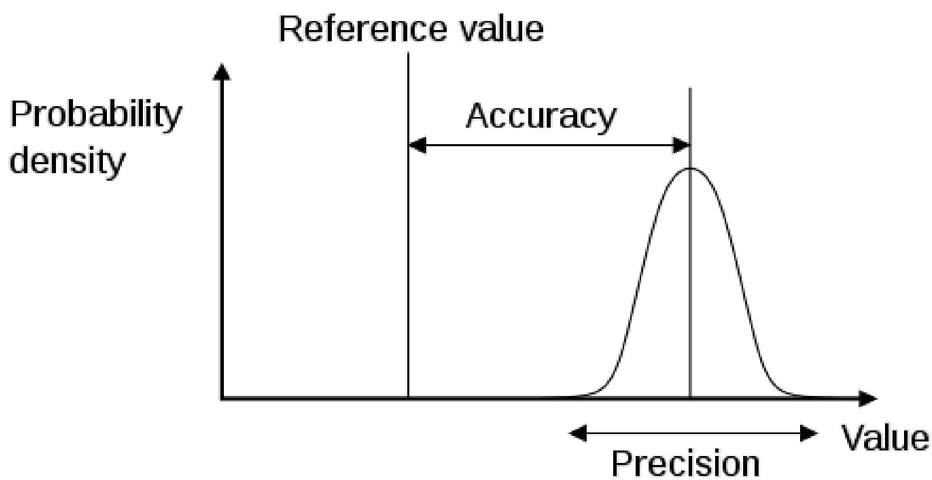
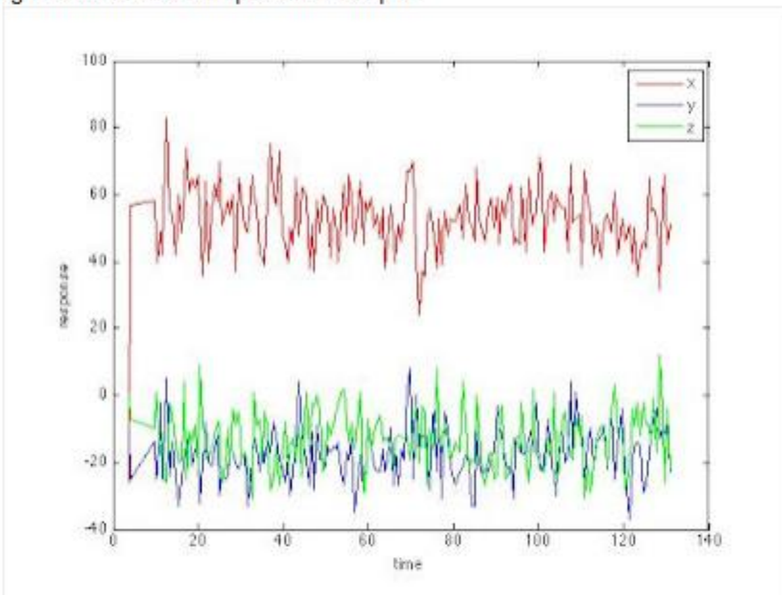


Fig. 8.1.1

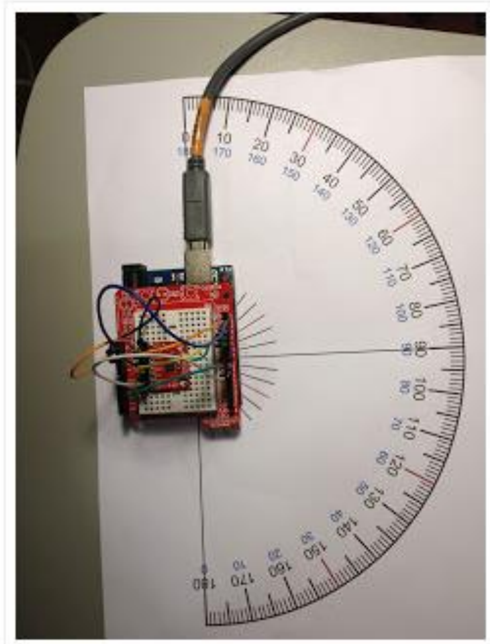
Note: precision is repeatability and accuracy is reference to some standard value.

Some research has been found on the internet regarding gyroscope calibration [4]:

The first thing I did was turn on the gyroscope and let it sit stationary on the table. This estimates the zero settings. Here is an example of the output.



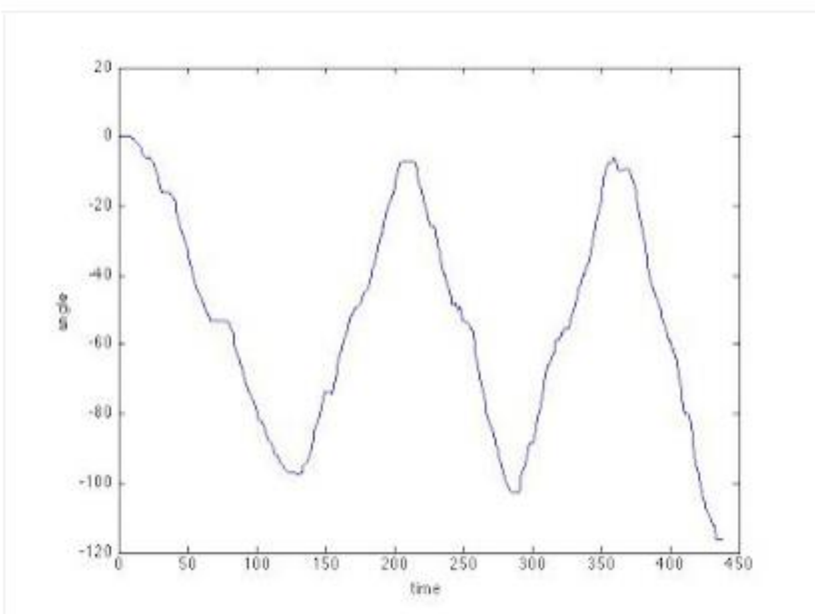
The mean of response is <52.5054, -17.3098, -12.8859>. This is used to correct the up front bias in the voltage settings: simply subtract this value as it is read to remove the bias. I figure eventually, I will add a calibration step that can be used to estimate the gain/offset parameters to get better calibration values. The next step was to make some cumulative angle measurements. I printed out a protractor image and then turned the gyroscope back and forth between 0° and 180°.



Then I integrated the z rotation angle to compute the absolute angle. The idea is that the gyroscope produces a delta angle (call it θ):

$$\theta = \int_0^t \theta^t \, dt$$

using this formulation, I generated the following plot:



Note: sample test code in appendix C refers robot calibration section 8.1

[1] Accessed Dec 4, 2012, “About Repeatability”
http://www.strobotics.com/repeat.htm

[2] Accessed Dec 4, 2012, *JCGM 200:2008 International vocabulary of metrology — Basic and general concepts and associated terms (VIM)*
http://www.bipm.org/utils/common/documents/jcgm/JCGM_200_2008.pdf:

[3] Accessed Dec 4, 2012,
http://en.wikipedia.org/wiki/File:Accuracy_and_precision.svg

[4] Accessed Nov 14, 2012, “Using the L3G4200D gyroscope with the Arduino”,http://ardadv.blogspot.com/2012/05/using-l3g4200d-with-arduino.html

[5] Accessed Dec 4, 2012, “ISO 9283:1998 Manipulating industrial robots -- Performance criteria and related test methods $162”,
http://www.iso.org/iso/catalogue_detail.htm?csnumber=22244

[6] Accessed Dec 4, 2012, “ANSI/RIA R15.05-2-1992 (R1999) $67”,
http://webstore.ansi.org/RecordDetail.aspx?sku=ANSI%2FRIA+R15.05-2-1992+(R1999)#.UL4u4YPAeAo

8.2 Motor Testing

The motors selected for the robotic arm are servo motors. They respond to the pulse width of the input signal to determine position. Because the purpose of this project is motion mirroring, the sensitivity of the servos is of great importance. In terms of servo specifications, we would like the servos to have a deadband period as small as possible. This will be tested by recording the input signal with an oscilloscope and recording the output shafts angle relative to a base line to determine the minimum change in the signals duty cycle in order to produce a change in the output shafts angle. This minimum change in output shaft angle will determine the minimum arc length created by the arm. During the testing of the motors, there will be a measurement of the power consumed by the motors under various loads to see the efficiency of the motors. This is done by measuring the current drawn by the motors and multiplying it by the voltage supplied to the motors. There will also be testing done to see the effect the input voltage has on the position of the output shaft for the same duty cycle period. This will be of great importance if the robotic arm is being ran off of batteries. This will enable the group to determine the maximum run time for the robotic arm based on the specification of the batteries and the effect the input voltage has on assumed output shaft angle.

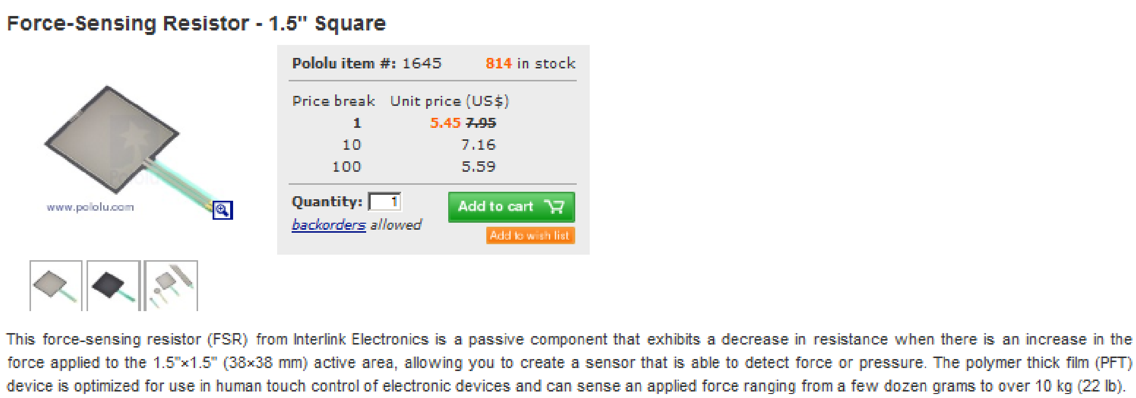
8.3 Sensor Testing

In order to test the functionality of the IMUs to be used, the board will be placed on an apparatus that can test the steady state values of the accelerometers at different angles and the drifting of the measurement. The apparatus will also be rotated some number of degrees, rotated back to its starting position, and integrated to test the accumulated error of the accelerometer. The power used by the unit will be measured to determine the necessary power to provide and the duration it can be operated off of a battery. The gyroscopes will be rotated on the apparatus at different constant velocities and different angles to measure their output and test for linearity. Any deviation from the expected values will be accounted for in the control algorithms. These tests will also allow for a testing of the communication between the sensors and the control board.

8.3.1 HOW FORCE SENSORS WORK

Tested FSR (Force Sensing Resistor) 1.5 inch square (shown below) with ohmmeter successfully.

We are considering the following sensors for use as the end-effector open-close control sensor [2]:



Or [1]:

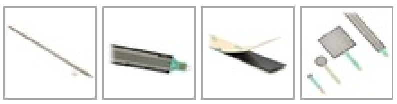
Force-Sensing Resistor - 24" x 0.25" Strip

Pololu item #: 1697 — **36** in stock

Price break	Unit price (US$)
1	**17.95**
10	16.16

Quantity: 1 — Add to cart

backorders *allowed* — Add to wish list



This force-sensing resistor (FSR) from Interlink Electronics is a passive component that exhibits a decrease in resistance when there is an increase in the force applied to the 24"×0.25" (610×6.3 mm) active area, allowing you to create a sensor that is able to detect force or pressure. The polymer thick film (PFT) device is optimized for use in human touch control of electronic devices and can sense an applied force ranging from a few dozen grams to over 10 kg (22 lb)

The devices have significantly different costs $5.45 (1.5 inch square) and $17.95 (0.25 strip) respectively, and operate on the same principle of increasing pressure on the sensor generates a decrease in resistance across the device. Specifically designed for use in human touch electronics the FSR (force sensing resistors) are 'polymer thick film' devices which exhibit an almost linear response when graphed in log-log format [3]:

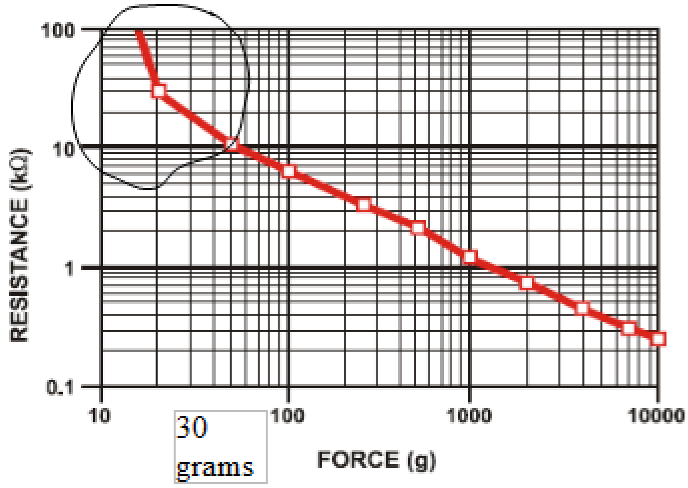


Fig. 8.3.1

The force-resistance graph above shows a virtual switch-like action when the initial pressure-force increases from zero to about 30 grams force and the resistance decreases 100k ohms to about 20k ohms. Generally, the greater the resistance the lesser the applied force. The 1.5 inch square FSR has been acquired and tested with an ohmmeter and does OL on the highest scale at zero applied force and then rapidly decreases to virtually zero resistance if heavy pressure is applied.

The force sensing resistor datasheet (fsr_datasheet.pdf) has some helpful info regarding implementing the force sensor [3]:

Suggested Electrical Interfaces
Basic FSRs

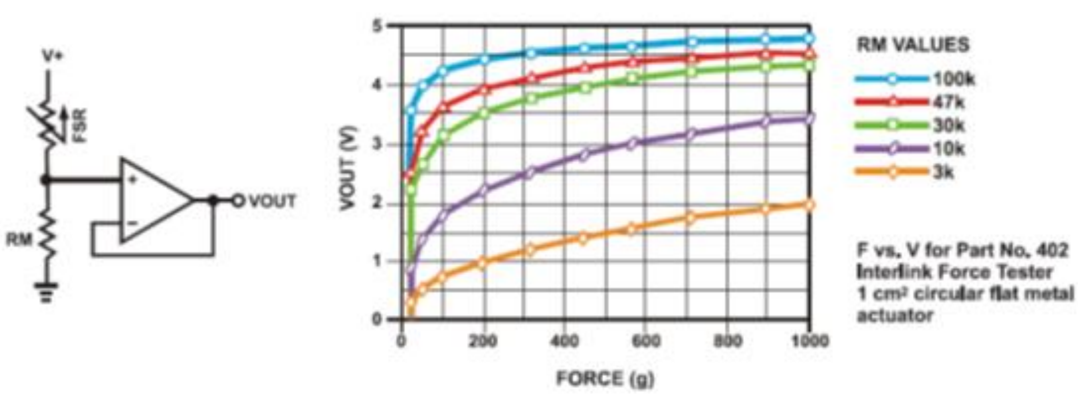


Fig. 8.3.2

FSR Voltage Divider [3]**:**

For a simple force-to-voltage conversion, the FSR device is tied to a measuring resistor in a voltage divider configuration. The output is described by the equation:

VOUT = (V+) / [1 + RFSR/RM].

In the shown configuration, the output voltage increases with increasing force. If RFSR and RM are

swapped, the output swing will decrease with increasing force. These two output forms are mirror images about the line VOUT = (V+) / 2.

The measuring resistor, RM, is chosen to maximize the desired force sensitivity range and to limit current. The current through the FSR should be limited to less than 1 mA/square cm of applied force. Suggested opamps for single sided supply designs are LM358 and LM324. FET input devices such as LF355 and TL082 are also good. The low bias currents of these op-amps reduce the error due to the source impedance of the voltage divider.

A family of FORCE vs. VOUT curves is shown on the graph above for a standard FSR in a voltage divider configuration with various RM resistors. A (V+) of +5V was used for these examples.

The fsr_datasheet.pdf includes additional adjustable buffer application circuits allowing the adjustment of gain and offset as well as multi-channel micro-controller interface for multiple FSR applications. Our design should only need one FSR and probably the larger (1.5 inch) square sensor in case the wrist gyro and force sensor need to mounted on a glove to maintain permanence of the human arm operator's equipment arrangement.

The datasheet includes a design for a variable force threshold switch for applications that require on-off switching. At this time, it is understood our design would try for continuous closing of the end-effector based on continuously increasing force; however, it might be necessary to implement a gripping-sequence from a threshold force.

References

[1] Accessed Nov 26, 2012, http://www.pololu.com/catalog/product/1697

[2] Accessed Nov 26, 2012, http://www.pololu.com/catalog/product/1645

[3] Accessed Nov 26, 2012, http://www.pololu.com/file/0J383/fsr_datasheet.pdf from www.interlinklelectronics.com

8.4 Wireless Testing

At the start, both microcontrollers chosen by the group will be activated. The microcontroller that controls the movement of the robot will connect itself with the other microcontroller via Bluetooth. For that to happen, the address of the microcontroller that uses the human arm sensors will already been written on the communication protocol on the first microcontroller. This will allow for the two microcontrollers to begin communicating with one another.

The operating mode of the robotic arm is then specified by the commands that the user wants by moving their arm. The analog signals from the user arm are converted to a digital signal by a Analog Digital Converter that has yet to be decided. These digital signals are interpreted by the second microcontroller, which processes the signals and changes them to a serial signal that contains the movement of all axes, x, y and z, of the robot arm and the robot arm grip. This serial information is then used to transmit the control signals to the other microcontroller via the Bluetooth module, which is used as a replacement of a serial RS-232 cable.

After the serial signal is directed to the other microcontroller that controls the servomotors of the robot, this second microcontroller decodes the serial signal using the other Bluetooth Module. With the interpretation of this data, the servomotors of the robotic arm will move following the commands given by the user from any point, provided within the scope range of Bluetooth modules.

8.5 Controller Testing

The confirmation that the control board is functioning at an expected level is an important step nearing the completion of the project. To ensure that the control board will function as required by the specifications, different tests will be performed to demonstrate that various components are behaving as required. The procedures to test these components are described in the following paragraphs.

The first step in making sure the control board is functioning is to ensure that the microcontroller is being properly programmed. One such way is to include on the control board an LED light and send a simple program to the microcontroller to cause the LED to blink at a certain interval. This would show that the USB connector is properly connected to the microcontroller and the program has been sent to it without any errors.

A second test would be to ensure that the microcontroller can properly control the motors on the robot. In order to do this the microcontroller should be uploaded with a program that attempts to send signals to the motors of the robot. The program would start with the base motor of the robot attempting to cause it to rotate from one extreme to another. The program would then attempt the same procedure for all other motors on the robot one at a time. This test would show that the signals for the motor are properly connected and that the microcontroller has the ability to control each motor.

The next test would be to check if the microcontroller is receiving the input from the sensors. To test this, the operator should connect the sensors to the control board and begin to move one. If the microcontroller is receiving a signal from the sensor then it should then move the motor associated with that sensor from one extreme to another. The operator should then repeat this procedure for all of the other sensors individually and ensure that the motor associated with that sensor moves when the sensor does. This test shows that the microcontroller is receiving the signals from the sensors and is relating it to the proper motors.

8.5.1 Circuit Assembly

The printed circuit board will be sent to out to be assembled. Some of the components come in a packages that do not have pins (surface mount), but rather require re-flow soldering. This is currently beyond the ability of any of the team members and in order to avoid damaging any components will be done by a qualified individual.

8.5.2 Robot Arm Assembly

The robotic arm(s) are stock and are assembled by the group. All necessary components come with the arm or are acquired stock.

8.5.3 Power Supply

Power provided will be DC and regulated by a buck regulator(s). They were chosen for their high efficiency. Linear regulators were not considered because of their low efficiency rating. The particular buck regulator will be chosen after testing has been done on all the components. This will ensure that a regulator with the correct specifications can be chosen to ensure maximum efficiency.

Chapter 9 Administrative Content

9.1 Milestones

The first milestone was achieved Nov. 4 with the successful build of the OWI 535 Robot Arm Edge which has a simple manual controller arrangement. Double throw switches that are manually thrown to cw/ccw rotate the 5 motors of the 535 arm. Mechanical working robot arm acquired - completed.

The second milestone is hoped to be completed before next term starts: Rotating the base motor M5 from a simple lateral motion of a gyroscopic 3D angle sensor connected through a microcontroller to the motor in a prototype arrangement. Initial prototype success.

The third milestone is getting the same prototype performance through our custom designed control board. Initial control board success.

The fourth milestone is getting all the sensors and motors running through our control board simultaneously. Complete control board success.

The fifth milestone is where the high-performance game begins of coordinated performance of the links to the human-operator’s arm motion. Qualitative system performance success.

The sixth milestone is getting our initial target of 1 second or less delay time from the human-operator’s arm to successful motion-tracking response of the robot arm. Initial quantitative system performance success (response time < 1 sec).

The seventh milestone is achieving a very short delay time between human arm and robot arm response. Final quantitative system performance success (response time < 0.1 sec).

9.2 Budget and Financing

The initial budget is $400 total. $40 was spent on the OWI 535 robot arm edge which gives a 5 DOF shoulder-elbow-wrist 3-linkage arm with base rotation and end-effector open-close gripper. This leaves $360 for sensors, microcontroller, and connectors, wires, misc., etc.

Upgrading to a metal robot arm with servo motors was previously detailed in section 5.2.2 Motors: Arm upgrade $133 + 5 servos total at $89.45 = $222.45 for mechanical equipment upgrade. Presumably, the electronics will not need any changes other than new voltage and current ratings for the new motors

9.3 Looking Toward EEL4915

Our initial concern for eel4914 is a working design utilizing cost feasible components. We think our current control board design can accomplish the initial performance objectives within a reasonable range of our initial budget ($400) - plus or minus 50%, emphasis on the plus. Adding wireless capability can be done for perhaps $100. Another micro-controller will probably have to be included also, to connect wirelessly from the sensors on the human arm to the controller on the robot arm.

At this point in time, the crucial first problem of implementation appears to be successful coding of the gyroscope sensor into effective motor driving response. The next problem will be coordinating the multiple sensors signals' into a coherent robot arm motion. If this is accomplished with the plastic arm with dc motors, we can then upgrade to the metal arm with servo motors and concentrate on improving system response time and overall system stability and performance. If we get stuck implementing the multiple sensors system we might have to restart with a simpler design as observed in some of the SIMILAR PROJECTS section employing fewer degrees of freedom.

Therefore, our first milestone already accomplished of a working mechanical arm is followed by a second milestone of a working gyroscope sensor - defined by successful rotation response of a robot arm link based on moving the gyro sensor. Initially, the attempt will be to rotate the base of the arm (motor 5) horizontally clockwise and counterclockwise based on horizontal motion of the gyro sensor. Next, a vertical rotation of the first shoulder-to-elbow link (motor 4) will be attempted.

A major upgrade possible with the metal arm is a rotating wrist – roll axis rotation perpendicular to the pitch axis rotation of the end-effector. Of course, this will probably require adding another gyro sensor on the human-operator's arm. We will probably upgrade to wireless function before adding any additional linkages or any other upgrade features.

On the academic side, at least one group member is taking the eel4612 spring term course on systems controls and the intent is to include Matlab based system analysis of the robot operation. An additional section SYSTEMS AND

SIGNALS will be included, to try to quantitatively analyze the robot mechanical system stability response and signal quality through the electronic system.

Appendices

Appendix A Copyrights & Permissions

Figure 4.3.2 -

Hello, Taylor.

You are welcome to use the schematic of that product in your report as long as you give us credit.

Please let me know if you have any additional questions.

Sincerely,
Jonathan Kosh
(702) 262-6648
www.pololu.com
....
Pololu Corporation
920 Pilot Rd.
Las Vegas, NV 89119
USA

Figure: 5.4.3.2

Hi Matthew,

yes you have permission to use that image.

Regards John Main

The Only Microcontroller C Programming Course
Showing You How To Make Your Own C Projects:

http://www.best-microcontroller-projects.com/c-start

Get the Microcontroller ezine for
C programming tips, projects and more.
Visit http://www.best-microcontroller-projects.com/ezine

Fig. 3.1.4 -

[info@technophilia.co.in contact info, no response as of 12/5/2012]

['I am a student in electrical engineering at Univ Central Florida in Orlando, Fl, USA.
I would like to have permission to use a screen capture of your demo on youtube http://www.youtube.com/watch?v=vqbNoALYVd4&feature=related for a senior design report on robot arms. The screen capture is of the black and yellow arm following the human operators' motions.
Kurt Graf
candidate B.S. EE
UCF Sent 11/29/2012]

Note: from section 3.2

Permissions, Disclaimers (refers section 3.2)

Disclaimer from website http://www.robotmatrix.org/Disclaimer.htm

for sources 4,5,6

While we strive to provide accurate and timely information, there may be inadvertent technical and/or factual inaccuracies and typographical errors. For which we apologize.

No Warranty, No Liability, Indemnification

THE WEBSITE AND THE MATERIALS ARE PROVIDED BY COMPANY ON AN "AS IS" BASIS AND AS AVAILABLE, WITHOUT ANY WARRANTY OR REPRESENTATION OF ANY KIND, WHETHER EXPRESS OR IMPLIED. COMPANY EXPRESSLY DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT, WITH RESPECT TO THE WEBSITE OR ANY MATERIALS.

TO THE FULL EXTENT PERMISSIBLE BY APPLICABLE LAW, ROBOTMATRIX.ORG DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. ROBOTMATRIX.ORG WILL NOT BE LIABLE FOR ANY DAMAGES OF ANY KIND ARISING FROM THE USE OF THIS SITE, INCLUDING, BUT NOT LIMITED TO DIRECT, INDIRECT, INCIDENTAL, PUNITIVE, AND CONSEQUENTIAL DAMAGES.

IN NO EVENT SHALL COMPANY BE LIABLE FOR ANY COMPENSATORY, DIRECT, INDIRECT, INCIDENTAL, CONSEQUENTIAL OR PUNITIVE DAMAGES, LOSS OF DATA, INCOME OR PROFIT, LOSS OF OR DAMAGE TO PROPERTY, OR ANY CLAIMS OF YOU OR OTHER THIRD PARTIES WHATSOEVER WITH RESPECT TO THE WEBSITE OR THE MATERIALS OR SITES LINKED FROM THE WEBSITE OR MATERIALS, REGARDLESS OF THE LEGAL THEORY ON WHICH THE CLAIM IS BASED, INCLUDING, WITHOUT LIMITATION, ANY DAMAGES THAT RESULT FROM ANY MISTAKE, OMISSION, VIRUS, DELAY OR INTERRUPTION IN OPERATION OR SERVICE OR FAILURE OF PERFORMANCE, WHETHER OR NOT RESULTING FROM AN ACT OF GOD, COMMUNICATIONS FAILURE, THEFT OR OTHERWISE. COMPANY SHALL NOT BE LIABLE FOR ANY LOSSES OR DAMAGES WHATSOEVER RESULTING FROM ANY FAILURE OF THE INTERNET.

ROBOTMATRIX.ORG MAKES NO REPRESENTATIONS REGARDING THE AVAILABILITY AND PERFORMANCE OF ITS WEBSITE. YOU HEREBY ACKNOWLEDGE THAT ANY USE OF THE WEBSITE AND RELIANCE UPON ANY MATERIALS SHALL BE AT YOUR SOLE RISK AND THAT COMPANY SHALL NOT BE LIABLE FOR ANY LOSS OF DATA, LOST PROFITS OR ANY OTHER DAMAGES OR LOSSES RESULTING FROM SUCH USE.

Copyrights

http://www.learnaboutrobots.com/copyright.htm

If you find anything on this site that you feel infringes on a copyright, please let me know and I will do my best to rectify the situation.

You may not copy or redistribute any of the digital images on this site without express written permission from the copyright holders.

Please give credit if you use more than a line or two of text from this site.

-Rich Hooper

Appendix B Datasheets

Information reference for 4915 from section 4.2 possible components:

Accelerometer & Gyro Tutorial

http://www.instructables.com/id/Accelerometer-Gyro-Tutorial/step1/The-Accelerometer/

Appendix C Software

Sample code for section 8.1

Here is the test code used to call the gyroscope class:

```
#include "gyroscope.h"

ardadv::sensors::gyroscope::Gyroscope gyroscope;

void setup()

{

Serial.begin(9600);

Serial.flush();

typedef ardadv::sensors::gyroscope::Gyroscope Gyroscope;

gyroscope.setup(Gyroscope::INTA(7), Gyroscope::INTB(6), Gyroscope::CS(10));

}

void loop()

{

gyroscope.update();

const unsigned long t = millis();

::Serial.print(t);
```

```
::Serial.print(",");

::Serial.print(gyroscope.x(), DEC);

::Serial.print(",");

::Serial.print(gyroscope.y(), DEC);

::Serial.print(",");

::Serial.println(gyroscope.z(), DEC);

::Serial.flush();

::delay(100);

}
```

L3GD20 Code References for 4915

https://github.com/adafruit/Adafruit_L3GD20

This is a library for the Adafruit Triple-Axis Gyro sensor

Designed specifically to work with the Adafruit L3GD20 Breakout ----
> https://www.adafruit.com/products/1032

These sensors use I2C or SPI to communicate, 2 pins (I2C) or 4 pins (SPI) are required to interface.

Adafruit invests time and resources providing this open source code, please support Adafruit and open-source hardware by purchasing products from Adafruit!

Check out the links above for our tutorials and wiring diagrams

Written by Kevin Townsend for Adafruit Industries.

To download. click the DOWNLOADS button in the top right corner, rename the uncompressed folder Adafruit_L3GD20. Check that the Adafruit_L3GD20 folder contains Adafruit_L3GD20.cpp and Adafruit_L3GD20.h

Place the Adafruit_L3GD20 library folder your (arduinosketchfolder)/libraries/ folder. You may need to create the libraries subfolder if it's your first library. Restart the IDE.

Adafruit_L3GD20 / examples / Adafruit_L3GD20_test

/***

This is an example for the Adafruit Triple-Axis Gyro sensor

Designed specifically to work with the Adafruit L3GD20 Breakout

----> https://www.adafruit.com/products/1032

These sensors use I2C or SPI to communicate, 2 pins (I2C)

or 4 pins (SPI) are required to interface.

Adafruit invests time and resources providing this open source code,

please support Adafruit and open-source hardware by purchasing

products from Adafruit!

Written by Kevin "KTOWN" Townsend for Adafruit Industries.

BSD license, all text above must be included in any redistribution

**/

Adafruit_L3GD20 / Adafruit_L3GD20.cpp

/***

This is a library for the L3GD20 GYROSCOPE

Designed specifically to work with the Adafruit L3GD20 Breakout

----> https://www.adafruit.com/products/1032

These sensors use I2C or SPI to communicate, 2 pins (I2C) or 4 pins (SPI) are required to interface.

Adafruit invests time and resources providing this open source code,

please support Adafruit and open-source hardware by purchasing

products from Adafruit!

Written by Kevin "KTOWN" Townsend for Adafruit Industries.

BSD license, all text above must be included in any redistribution

***/

#include <Adafruit_L3GD20.h>

/**

Adafruit_L3GD20 / README.md

#endif

This is a library for the Adafruit Triple-Axis Gyro sensor

Designed specifically to work with the Adafruit L3GD20 Breakout ----
> https://www.adafruit.com/products/1032

These sensors use I2C or SPI to communicate, 2 pins (I2C) or 4 pins (SPI) are required to interface.

Adafruit invests time and resources providing this open source code, please support Adafruit and open-source hardware by purchasing products from Adafruit!

Check out the links above for our tutorials and wiring diagrams

Written by Kevin Townsend for Adafruit Industries.
BSD license, all text above must be included in any redistribution

To download. click the DOWNLOADS button in the top right corner, rename the uncompressed folder Adafruit_L3GD20. Check that the Adafruit_L3GD20 folder contains Adafruit_L3GD20.cpp and Adafruit_L3GD20.h

Place the Adafruit_L3GD20 library folder your (arduinosketchfolder)/libraries/ folder. You may need to create the libraries subfolder if its your first library. Restart the IDE.