Table of Contents

Executive Summary

In today's society robots are everywhere, whether is it just simply a toy design to entertain children, a computer that perform a task at a scheduled time, a machine that is suppose to help workers perform his or her job faster and better, a car that can take someone from point A to point B without the use of a driver, one good example would be a Taxi Cab that can drive you anywhere you want within a certain area. Or as we have seen in many movies, a robot can be design to be a spy used for a number operations. All these different types of robots have one important thing in common; which is a certain level of intelligence; a program that tells it what to do or look for, and when to do it. In a few decades from now, robots will probably be as commonplace as seeing someone led by a seeing-eye dog on the streets.

The robot we are building is intended to be a remotely-controlled spy-robot, with some of its functions being controlled thru an RF joystick controller. It is going to be more or less shaped like a spider, with six legs and a field of vision. The purpose of the robot is to be able seek and distinguish between objects of certain colors that are going to be programmed in its memory. In order to do that, the robot will have vision unit that is compose of a color sensor that is going to be housed underneath the robot's body to a a color sensor unit and circuit will look for certain colors while it is walking; once the targeted object is found, and LED of that colored object will illuminate, indicating the found target. A microcontroller will coordinate the robot's movements and direction. Another microcontroller will decide if the color being detected is the one selected by the user. The robot will have a DC voltage source as its main power source.

Section 1: Introduction

1.1 Motivation and Rationale

The motivation of the group for this project was to gain practical experience in design and prototyping while putting what has been learned in our undergraduate classes to use. Although the group originally wanted to produce something of utility that could fill some type of need, the group realized that the experience desired would probably best come in the form of designing and building their own project from scratch. Since the different members each had different expertise and different interests, the initial phase of deciding what to build had the restriction of providing something that was of interest to each group member. This is why the group decided to build a robot. A robot design easily provided all of the areas of interest that the members wanted. There would be no real practical purpose to the robot besides building it and having it work; it would just be a massive learning experience.

Since the group members were interested in building a novel control system, radio frequency communication, computer vision/sensory attribute, and power electronics, our project just combined all of this together into a walking hexapod robot with artificial computer vision and radio control. Naturally powering this diverse system would be a challenge in and of itself, and the student interested in power electronics was content with the challenge ahead. The group also explicitly decided that elegance in design would not be a concern. In other words, although certain problems might have been solved by other methods (wheels, only infrared vision, programmed control instead of remote, wired power supply), it was decided to expressly make things more difficult in order to provide ample testing and learning grounds. Thus pragmatic issues were not concerns, the design process and creating a project with strong learning potential was the primary concern.

Section 2: Definitions

It is sometimes said that within every problem, there's a smaller problem just screaming to get out. In order to facilitate research, design, construction, and testing, the overall project problem was decomposed into several sub-problems. We will refer to these sub-problems as subsystems. Each subsystem was then further decomposed into several other sub-problems; which we will refer to as the subsystem requirements. To better put this into perspective, each requirement will be explained in relation to its goal or objective to the project as a whole.

The purpose of this section is to identify project subsystems and their respective goals and objectives. The remaining sections will present the research, design, implementation, and evaluation of these same objectives, making these definitions an essential conceptual piece of the puzzle.

2.1 Goals and Objectives

Below is a succinct recap of project scope and overall objectives in order to motivate the decomposition into subsystems.

A hexapod robot is to be designed and built such that each leg provides three degrees of freedom and the robot itself is able to move independently in 8 distinct directions, or along two axes of motion. Mounted on its chassis will be an autonomously scouting camera, capable of distinguishing between 3 colors and communicating to the user when it has found an object of the given color it is searching for.

In these next subsections, the overall project will be decomposed into subsystems, and the goals and objectives of the individual subsystems will be identified.

2.1.1 Mechanics

Perhaps the most crucial subsystem is the mechanical one. It will be composed of purely structural components, and components that provide both structure and enable motion. If this system is not properly built, all other systems will fail to adequately perform. It can be viewed as the physical framework or skeleton of the robot, before any electron is manipulated.

Structural components are the pieces that will house and support the electronics necessary for the robot. The structural components must also be able to support the robots interaction with the ground, that is, be able to balance it while it is still and prevent collapse during movement. These components not only include the main body chassis, but also the supporting legs and its connections to the body. The components must be capable of carrying loads, resisting body forces, and, if necessary, providing adequate ventilation for electronic equipment.

Since a stationary robot would not be much fun and would defeat the purpose of having legs anyway, a method of movement must be implemented. In order to provide movement as a hexapod, motion will be generated by mechanical components that can rotate or pivot about a certain axis. Some type of actuator can then intelligently provide this rotation to cause motion. It is crucial that whatever mechanical components allow for movement are not easily twisted or broken with a bit of excess force. They must be able to withstand forces greater than we expect since there will always be certain outliers in force application, and once a component starts to malfunction it will only get exponentially worse. Inadequate provision at this stage could possibly produce a robot that works well at first, but cannot sustain repeated testing due to the wear and tear of its mechanical parts.

The objectives of this subsystem are to provide a stable housing structure and a platform capable of providing reliable movement for the hexapod robot. The Drive subsystem definition in the next section will explain the concept of movement as it relates to the hexapod robot, and how it will be generated.

2.1.2 Drive

This is the subsystem responsible for not letting the robot just sit there and accumulate dust. Its overall objective is to provide a system of movement for the hexapod robot. In order to do so, it must somehow articulate the 6 legs along their three degrees of freedom in a meaningful fashion to produce movement. Since we have already been referring to the pointed chunks of material that will be protruding from the robot's chassis as legs, it seems logical to now refer to its movement as walking. We will refer to the hexapod's movement, in any direction, as walking and the method or algorithm by which it walks as its gait.

Walking can be viewed as controlled falling. Thus not only must the 6 legs relatively move in a meaningful fashion, but they must also provide enough balance and cohesion so the robot doesn't fall flat on its chassis. This will be much easier said than done. The reason for this is that there will essentially be 18 degrees of freedom that the Drive system must control, and it must be done simultaneously. If a single leg is not appropriately placed or does not touch down on time, and the remaining legs do not compensate for its error, then it will likely topple right there on the spot. In fact, the initial testing phases will consist of a group member walking along the hexapod, which is trying to walk too but tends to fall, and making sure that no important electronics are damaged nor the Mechanics system is damaged whenever the robot takes a plunge.

Although a series of sensors could be installed on the robot to assure proper gait, this would not only make for an incredibly bloated design, but would also seriously increase the overall cost of the project. As stated earlier, a goal that runs parallel to the completion of a working project is to design it well and efficiently. Therefore just because 18 sensors, or even 36 sensors, would provide for near flawless gait, it is perhaps not the optimal choice when economic factors are taken into consideration. The Research and Investigation section will further address this issue, and what type of feedback mechanisms will be employed to most efficiently solve this problem.

In order to provide the necessary torque to rotate the mechanical components responsible for leg movement, or the joints, motors will be used. The 18 degrees of freedom can then be defined as 18 distinct joints, and 3 joints will exist on every leg. Each triple pair of motors will then rotate the joints in a particular fashion, such that the leg itself moves along a given arc of motion. With the combined effort of the six legs, that is the combined six arcs of motion, the robot will then be able to move in any path linearly; that is the rotational motion will be converted to linear motion. An algorithm must be created to properly drive the legs, and a method to actually make the motors follow the given algorithm must also be created. Thus proper control of the legs will rely on not only writing proper software, but on being able to build the necessary hardware that can command the motors to execute the software in sync. This will be a significant difficulty, since all 18 motors must be controlled at once.

The main objective of this subsystem is to provide a method by which to linearly translate the position of the robot; to move it. It must provide a fluid and somewhat aesthetic movement that does not falter or crash when the robot starts, stops, or changes directions. Which directions the robot will move in, what joints will provide movement, gait speed, and other details will be covered in the specifications section (Section 3.2).

2.1.3 Vision

The vision/imaging system includes a color sensor, lens assembly, microcontroller and LEDs. The goals of the image system are to extract information from the surrounding environment of the robot, check if what is gathered contains what the user has specified (a specific color), and then notify the user if it has succeeded.

The information from the environment will be in the form of reflected light off the object being detected by the color sensor unit. The light will enter or not enter an array of photodiodes depending on what color the sensor unit is set. The array of photodiodes have color filters which is what the user is actually controlling when he/she selects a color. Those reflected light rays are tested against a set/programmed threshold. Depending on the intensity of those collected light rays the threshold may or may not be broken. Our system will detect 3 separate colors: red, green, and blue. These colors are at separate parts of the visible-light spectrum). Depending on which mode is selected by the user, an output signal in the form of turning on an LED will notify the user. The range of vision for our camera is also important. It needs to be able to capture moderately wide angle so as to limit the number of snapshots needed.

There are three separate modes of several operation for our free-roaming robot: 1) Red-color object detection, 2) Blue-colored object detection, and 3) Green-colored object detection.

These modes are simply for the user to control the robot with the remote control. The color sensing unit will be powered on with the main power supply of the robot and will thus, always be on.

2.1.4 Wireless Communication

The robot is to be controlled wirelessly by a conventional Playstation 2 controller, that is, the version equipped with the cable. Using the Playstation 2 controller, the user will be able to control the movements, the gait, and the decisions of the robot. The Playstation 2 controller will be part of the wireless unit meant to handle the wireless function of the whole system. The robot will also send out to the wireless unit vision-related data and data pertaining to decisions made by the user. The wireless unit consists of the Interpreter circuit, two transceivers (one on the mobile unit with the user, the other on the robot), the Playstation 2 controller.

The Interpreter circuitry is titled so because of one of its main task is to interpret the data coming sent out by the Playstation 2 controller through its cable. The Interpreter circuit's other task is to receive and process data transmitted by the robot and interpret them by displaying them accordingly through the LED display box. The Interpreter is built around two microcontrollers that are independent of each other and do not communicate with each other directly, i.e., there will be no direct data traffic between them, even though they will be part of the same unit and soldered on the same board. One of them is to interpret the data from the Playstation 2 controller. The relationship between this microcontroller and the Playstation 2 controller will be the same as that between the Playstation 2 console and the controller; hence, the microcontroller is to emulate the Playstation 2 console. It will receive the data from the controller, which transmits it in serial format, then process it and spit it out in parallel format to the transceiver, which will be on the same board.

The other microcontroller is to collect and organize the data sent out by the robot through the transceivers. This set of data is composed of data from the image processing unit of the robot and data concerning some decisions made by the user through the use of the Playstation 2 controller. After processing the data, the microcontroller will be programmed to display it explicitly to the user through a LED display, thereby informing the user on the status of the robot.

Evidently, two transceivers will be used in the wireless unit of the system, allowing a two-way data transfer between the mobile unit and the robot; one (let it be labeled Transceiver 1) on the mobile unit, which consists of the Playstation 2 controller and the two microcontrollers covered above; the other (labeled Transceiver 2) on the robot. In sending mode, Transceiver 1collects and transmits to Transceiver 2 the parallel data supplied by the microcontroller handling the data from the Playstation 2 controller. In receiving mode, Transceiver 1 collects the data from the robot's processing unit transmitted by Transceiver 2, and supplies them to the microcontroller handling the LED display box.

In sending mode, Transceiver 2 collects the data from the image processing unit and the robot's processing unit, consolidates them then transmits them to Transceiver 1, which, in turn, sends them to the microcontroller handling the LED display box. In receiving mode, Transceiver 2 receives data sent out by Transceiver 1 from the microcontroller handling the interpretation of the data from the Playstation 2 controller, and then transmits them to the robot's processing unit which will then convert them into movements or decisions. The transceivers are to operate in the 2.4 GHz ISM band, and each of them will be equipped with a small compact chip antenna for the sake of size and mobility.

The LED display system will be made up of four differently colored LEDs. Three of them will be red, blue and green. They will inform the user which color is detected by the camera mounted on the robot. No more than one of these LEDs will be turned on simultaneously. Only one of them can be on. The LED display box will display one color at a time, as the camera and the image processing unit will be programmed to detect one color at a time. The fourth LED will be transparent (it will emit white light). It is on when the robot is hunting for a color, and it is off when the robot is on standby.

To summarize, the wireless unit is to permit a convenient two-way wireless communication between the robot and the user, who will wirelessly control the robot in its movements and decisions.

2.1.5 Power

Depending on the projected use or application of a robot, its design requires knowledge of programming, remote sensing, artificial intelligence and power systems. A robot is device that can execute physical tasks automatically, either by supervision or by programming it to follow a general set of guidelines through artificial intelligence.
The robot can be divided into three groups that allow it to move, interact with its environment and be useful to others depending on its intended use.
The major groups include:

1. The Brain: a set of programs, or a computer.
2. Mechanical parts: servos, motors, and legs.
3. Sensors: vision, motion, etc

And all of these components require the use of power in order for them be functional. The robot relies on a battery as its main power source. All batteries in general only produce a single voltage level, while the robot's subsystems require a diversity of voltage levels in order to function properly. Therefore the power supply system must be designed to provide all the subsystems with the necessary power that they require. Given that the main power supply of the robot is battery, it consists of a combination of DC to DC converters using linear or switching regulators, which help regulate the different voltage level that each subsyste require. For efficiency need and our budget, design considerations is to be required in choosing the voltage regulators. Since the battery also power the servos and motors, which can cause noise, therefore filtering and voltage isolation is advantageous, for the robot's computer system.

Section 3: Specifications

3.1 Mechanics

Detailed specifications are contained in this section, which specify the requirements that certain mechanical components must meet. Since the two primary objectives of this subsystem are to provide stability and mobility, the chassis, legs, and joints will be analyzed separately. The main function of the chassis and legs will be stability, while the function of the joints will be mobility.

3.1.1 Chassis

3.1.1.1 Body Chassis

The chassis must be capable of housing all electronics material, both safely and securely. It must also provide adequate ventilation for them, since we do not want valuable equipment to overheat. The material itself is to be smooth in nature, so adhesives can be used, if needed for mounting. As building gets under way, it might be desirable to mount a battery pack, or similar device, with Velcro™ or tape, and if the chassis surface were rough and coarse this might not be possible. Besides its constituent material and geometry, the weight and dimensions of the chassis are very important. In order to prevent putting too much strain on the motors and joints, the chassis is to weigh no more than ¾ of a pound by itself. It should also not be wider than 3 feet at its widest point. This means that if it is trapezoidal in nature, its diagonals must not be longer than 3 feet. These constraints are imposed now in order to facilitate the research phase, they are based on the reasonable assumptions that if the chassis is too large or too heavy the motors necessary to move such a thing will either be too bulky or too expensive.

The chassis must provide some method of connection for the six legs, and must allow the joints to exhibit their full range of motion without inhibition. This means its perimeter must be sufficiently large such that all six legs can be connected through their hip joints, and rotate as much as needed. If these requirements are not met, then motion will be impeded. Another important specification is that the connection for all of the legs must be fully parallel with each other, that is, all leg connections must lay on the same two-dimensional plane. Although different sized legs could be used if this requirement were not met, by meeting it, the same geometry can be used for all six legs. Note that this is in contrast to an actual hexapod in nature, where only leg pairs are similar. But by adhering to this spec building or obtaining the legs will become a substantially easier process. The purpose of the body chassis is not only to provide stability and structure, but also to enable integration of the legs and joints into a single cohesive unit.

3.1.1.2 Vision Unit Chasis

A separate board will be added onto the body chassis for the color sensor in the Vision subsystem. The color sensor will hang from the bottom of the robot. By mounting it onto a separate chassis, the electronic components necessary to drive it can be treated distinctly from the ones in the body chassis. Not only will this save room in the final design, but it also allows the construction of both chassis to be done independently. This is a crucial factor for the project which is to be accomplished on a strict schedule. This chassis is to be no greater than 8 inches, or smaller than 4 inches, along its widest part. It must also leave at least 5 inches space between it and the body chassis in order to fit necessary components. It is to weigh no more than a ¼ of a pound.

3.1.2 Legs

The legs must be capable of withstanding the chassis loads, and must not bend or twist while walking. Since the maximum load that the chassis will exert upon the legs is 1 pound, and a reasonable assumption is that all of the electronics will not weigh more than 4 pounds, the legs must be able to support at least 5 pounds. The load will not be directly felt by each leg. It will instead be distributed between each of the six legs. The distributed load itself will be a function of the body chassis geometry and the connection point of each leg. In order to facilitate design, the specification on legs will be that each individual leg can support at least two pounds. Thus although there is some uncertainty in exactly how much weight the legs must withstand until the chassis is obtained and connection points are decided upon, the legs should be more than capable of withstanding any load distribution given by the maximum five pound chassis. Although the joints discussed in the next section will make sure that the twisting and shearing forces on the legs are not very severe, legs obtained must also be sufficiently rugged to withstand twisting or internal forces. Therefore another specification on the legs is that they must be sturdy throughout, and not be weak on any particular bend or juncture.

The entire leg, when fully stretched, should not measure more than a foot. If the legs are too long, then not only will it be more difficult to move them, but the body forces on them will be more severe. Also, since the body chassis will be more than 3 feet at its longest point and be relatively light, having legs in excess of a foot would increase the robot's center of gravity too much and would compromise design and eventual performance. Three joints will compose the leg, the knee joint, and two hip joints. Although the joints will be discussed in greater detail in the next section, they are introduced now since they provide a natural way to measure the distribution of length in each leg. The length from the base of the leg, or the foot, to the knee joint should be between 5-6 inches, the length between the knee join to the vertical hip joint should be between 2.5-3.5 inches, and the length between the vertical hip joint to the horizontal hip join should be between 1.5-2.5 inches. This means the leg can have a maximum length of 1 foot, or a minimum length of ¾ of a foot. These ranges are not steadfast, and will be used as a guide to search for a working leg. The total leg length was decided due to its proportion with the chassis, and the length ratio for the joints was chosen to be commensurate with their load ratios, e.g., the length from the foot to the knee joint is the longest since it will carry the heaviest load. Figure 1 below better illustrates these ratios, their meaning relative to the leg and joints, and consolidates the measurement ranges. It depicts, from left to right: the foot to the knee; the knee to the horizontal hip joint; the horizontal hip joint to the vertical hip joint; the connection to the body chassis (body not shown).

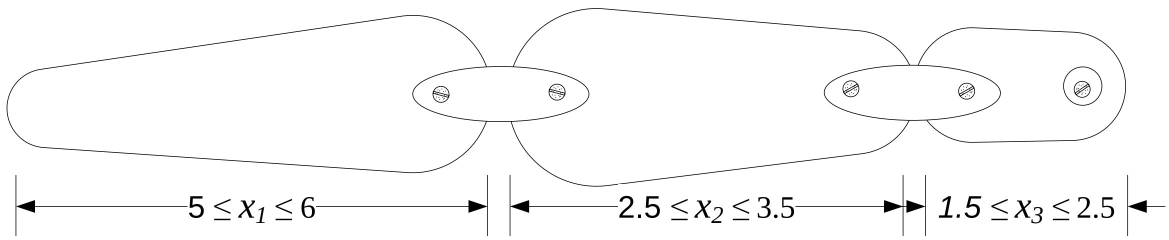


Figure 1: Concept diagram of leg with measurement ranges.

3.1.3 Joints

Each leg will have three distinct joints. Each joint will be rotated with a motor. There will be two hip joints and a single knee joint. The two hip joints will be orthogonal to each other; one will provide vertical rotation and the other horizontal rotation. The knee joint will provide rotation in the same plane as the vertical hip joint. To better illustrate this setup, a diagram is provided in figure 2 below. The angular rotation for the knee joint and the hip joint are depicted by the first two double arrows respectively, while the horizontal rotation of the hip joint is depicted by the first arrow from the right that appears to be going into the page.

The knee joint and the vertical hip joint are to provide motion in the same two-dimensional plane. To provide maximum flexibility in the design of the robot's gait, it would be desirable if each of the joints had a range of motion of 140°, or 70° to the left or right from its centered position. Several walking algorithms can be designed, and since there is no consequence in providing extra flexibility, but there certainly is in providing too little flexibility, it will be best to err on the side of caution. Even if the current gait will not take advantage of the entire range of motion present, future gaits with either a more advanced algorithm or more powerful motors could do so. Each joint must also have sufficient room to mount the motor that will control it. A direct interface between the motor and the joint is desired, without the use of any ancillary mechanical devices such as gears

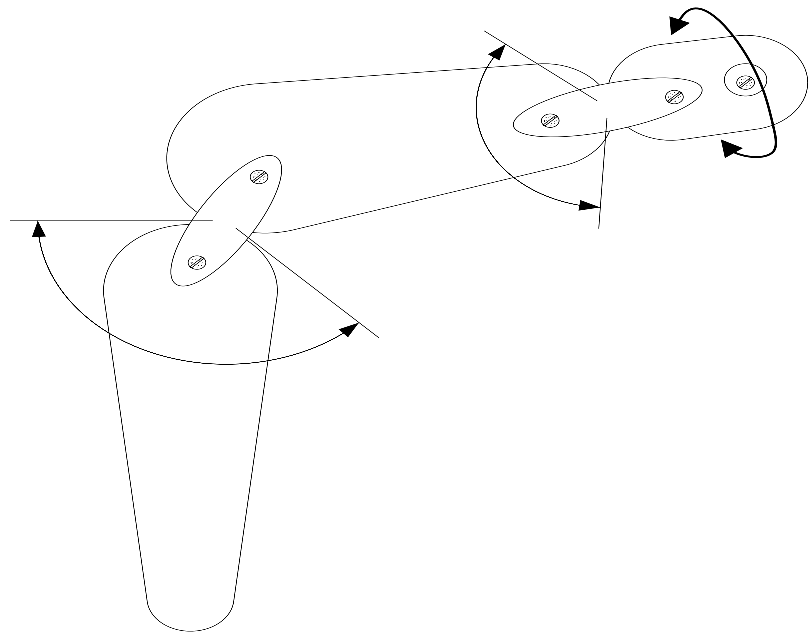


Figure 2: Concept diagram of leg with joints and respective range of motions.

3.2 Drive

Many of the specifications for the drive system will necessarily be a function of the mechanical system. However, even with limited knowledge regarding the mechanical system, several drive specifications were made since the mechanical system is at least bounded by its own set of specifications. Although the Drive subsystem stands to be fairly complicated, its high level specifications aren’t. Specifying the clearing rate, or roughly how tall of an object can the robot step over, the robot’s walking speed, and the quickness of its response to input provide a fairly comprehensive high level specification of the robot’s behavior. The robot should be able to clear, or scale, at least 15 cm, move at an average velocity of .5 m/s, and respond to commands or be able to alter its direction within 100 milliseconds. If the drive system is able to adhere to these specifications, then a fairly realistic gait should be exhibited by the hexapod robot.

3.3 Vision

Detailed specifications about the different components of the vision system are included here. The main objectives of the vision system are to detect three separate colors.

3.3.1 Color Sensor

The image sensor is the gateway to the outside environment. The color sensor must be able to detect color. A huge color range is not needed since only three colors will be selectable by the user: red, green, or blue. Different hues of these colors do exist, so an 8-bit color range seems very reasonable, that is, if a digital solution is to be implemented. If it is a digital solution, an analog video camera along with a video decoder would be used. This 8-bit range will allow for 256 total colors to be “seen” by the robot. Digitzation of the signal will allow for much easier interfacing with the video processing unit – an FPGA or microcontroller, for example. The image sensor must also be as small and lightweight as possible, to lessen the burden on the chassis and legs of the robot.

The robot will be controlled by the user via a remote control. The image sensor must scan the environment in order to save the user from re-positioning the robot. Therefore, the angle of the lens must be taken into account. The lens angle will be as close to 90 degrees as possible as to minimize the movement needed for continuously scanning the environment.

The frame rate of the image sensor is not a strong constraint because the robot will not be moving so fast that the processing cannot keep up. A frame rate of at least 2 frames per second will suffice for this project.

3.3.2 Color/Image Processing

3.3.3 Speed

Since our robot is not expected to go very fast, color and image processing does not have to be extraordinarily quick. Our robot is expected to walk about 0.25 meters/second. The objects detected by our robot will be stationary and solid colored, thus simplifying the process.

3.3.4 Image Algorithms and Processing

To detect an object of a certain color, our image processor must be able to detect boundaries or edges. Then it must determine how these edges connect. The image processor must have programmed definitions of the colors. Therefore, it must search for “blobs” of the user-specified color. The computation must take place before the next image is to be acquired or before the next obstacle is to be found. This should not be a problem, given our low frame rate specification. The source code for the image processor will be written in VHDL if using an FPGA or C if using a microcontroller and stored in an on-board ROM chip.

The image processor will look for adjacent pixels with the same value (0 to 255). If there is a high percentage of one color in one area of the screen (otherwise known as a blob), an object will be detected. What percentage like-colored pixels of the total number of pixels will denote is what must be determined. If we select a percentage too high, then only objects up close will be “seen.” If we select a percentage too low, then too many objects will be detected because the robot will “see” too much, like erroneous objects. We will set the percentage at 15%. Therefore, if a group of like-colored, adjacent pixels make up 15% of the total number of pixels or more, than this blob of pixels will count as an object. Once more testing can be done, another percentage may be applied.

3.4 Wireless Communication

3.4.1 The Controller

The Playstation 2 controller will be used to control wirelessly the many functions of the six-legged robot, including its motion and its decisions concerning targeting. To one’s comfort, the Playstation 2 controller provides two analog joysticks, a control pad (the “cross looking” buttons), and twelve buttons, including the ones that are pressed by pushing down on the joysticks, but not including the “Analog” button.

The left-hand joystick, which is a two-axis joystick, is to control the motion of the robot, that is, the movements of the legs corresponding to such motion, as it usually does for the controlled character in several Playstation 2 games. The right-hand joystick, which usually controls the sight of the video game characters, the “joystick buttons” and the control pad will not be used. For the joysticks to be operational, the controller has to be set on “Analog” mode, done so, either by pressing on the “Analog” button or by the software or video game, which activates it automatically, due mostly to the nature of the game (games that require solely the use of the analog sticks, such as flight simulators and such). When the controller is set on “Analog” mode, the red LED, located just below the “Analog” button, is lit. For simplicity, the circuit that will emulate the Playstation, or the controller interpreter, will be programmed to set the controller automatically on “Analog” mode.

The “R1” and “L1” buttons will be assigned the tasks of “Kill” and “Unlock,” respectively, while the rest will select the colors of the objects to be targeted.

Playstation Connector Pin Descriptions

Oftentimes, the Playstation 2 controller is equipped with a male connector (pinouts) matching the female port of the Playstation 2 console; rare, if not nonexistent, are the controllers that come equipped with female (no pinouts) connectors.

The connector is composed of nine pins, and a small, square metal plate at the bottom of the connector, usually referred to as the tenth pin, that is connected to the shielding of the controller’s cable. Through these pins, the controller and the console, in our case, the interface circuit, send to and receive from each other serial data, using the TTL or digital signal format (101110001...). The pins are labeled as shown in Figure 2.

PIN 1: DATA PIN

The wire to this pin is brown. Through it, the controller and the Playstation 2 console communicate in a one-way traffic where the controller sends serial data to the Playstation 2 or the interpreter circuit. Data are sent by byte least significant bit first (the lower bits of a data byte are output first) on the falling edge of the clock pin signal, while the console or the interpreter reads the data on the rising edge of the clock. It is usually high and uses positive logic (“1”=5 volts, “0”=0 volts). The maximum traffic rate is about 192000 bps.

PIN 2: COMMAND PIN

Its wire is orange. It allows one-way communication between the controller and the console where the console sends data serially to the controller. The signal is mostly high and uses positive logic. On the falling edge of the clock pin signal, the interpreter or the console sends the data least significant bit first, which the controller samples on the rising edge of the clock. The maximum data rate is about 192000 bps.

PIN 3: N/C (Not connected) 9 VOLTS

The line to this pin is purple. For the Playstation 1 version of the controller, which was basically the first version, this pin is inactive. But for the Playstation 2 Dual Shock version, which, unlike its predecessor, has a vibration mechanism, a voltage of 9 V is to be applied to this pin in order to power the vibrator. For our purpose, this function is unnecessary, so the pin is left disconnected.

PIN 4: GROUND PIN

The wire is black (or some shade of black). This pin is to be grounded, that is, connected to the negative pole of the power supply.

PIN 5: VCC PIN (POWER SUPPLY)

Its line is red. This pin is connected to the positive pole of a 5-volt power supply, supplied by the console or the interpreter circuit. This pin is also equipped with a 0.750A fuse.

PIN 6: ATT PIN (ATTENTION)

This line is yellow. Its role is to control and switch the state of the controller to one of two functions: accept or ignore data from the Playstation 2 or the interpreter. To enable the controller to accept data, which happens through the COMMAND PIN or PIN 2, the ATT PIN must be set to low ("0") before the first bit of data from the console or the interpreter is transmitted to the controller. After the transmission is done, the line is raised to high ("1"), and, from then on, if it stays high, the controller will ignore all activity on the COMMAND PIN.

PIN 7: CLOCK PIN

The color of this line is blue. The Playstation 2 console or, in our case, the interpreter circuit, transmits its clock signal to the controller through this pin, keeping them synchronized. Both the controller and the console transmit data through PIN 1 and PIN 2, respectively, at the falling edge of the clock, and receive data through PINS 2 and 1, respectively, at the rising edge of the clock. The maximum clock rate is about 192000 bps.

PIN 8: N/C

This pin is not connected to anything. There is no wire connected to it. Moreover, there are Playstation 2 controllers that do not have this pinout. If there is, leave it unconnected.

PIN 9: ACK PIN (ACKNOWLEDGE)

The line to this pin is green. The controller transmits through this line and its role is to inform the console or the interpreter that the controller is still active. After the last bit of each byte is transmitted to the controller, the line is pulled low within 60us while the ATT PIN (PIN 6) is held low, regardless of the clock. If this pin is not pulled down within that time frame after the last bit is transmitted, the console or the interpreter will start interrogating other controllers on the data port.

Single Data Byte Format

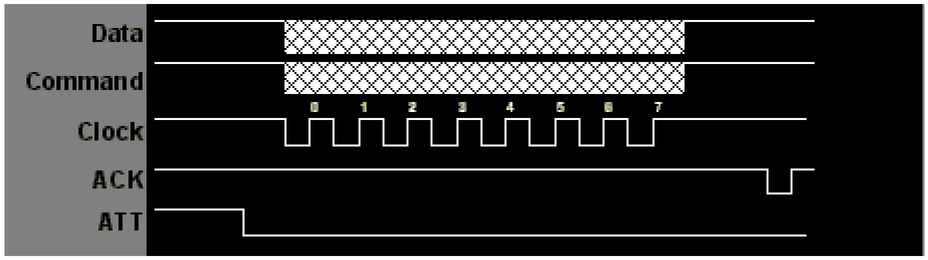


Figure 3: Diagram of a single data byte (Authorization for reprinting pending).

Figure 3 shows a typical diagram of the transmission of a data byte. It can be observed that the DATA and the COMMAND lines can transmit data at the same time, meaning that the controller cable is two-way traffic for data. The shaded areas represent whatever data is being transmitted. The ATT line is held low by the Playstation console or the interpreter to allow the data on the COMMAND line to be received by the controller. The ACK line is pulled low for a short time after the transfer of the data byte.

Data Frame Format

The Sony Playstation 2 DualShock 2 will be used for this project, since it is the cheapest and its design is the simplest available one today, facilitating the design process for an interface circuit. When the Playstation 2 console or the interpreter initiates communication with the controller, it sends out the first byte, “0x01”, through PIN 2 (COMMAND PIN) to the controller in order to startup the process. In response, the controller sends out its model number (which is “0x73” for the one that will be used in this project) through PIN 1 (DATA PIN). As the controller transmits its model number, the console simultaneously sends out through PIN 2 a second byte, “0x42”, which means “Request for data.” Then, the controller responds “0x5A”, meaning “Sending data,” while the console or the interpreter idles.

The console or the interpreter is ready to receive key data from the controller. As shown in Table 1 below, each key has a single assigned bit. The bits assigned to the keys are normally high, until the keys are pressed, which changes the corresponding bits to “0”. The analog sticks, though, cause the controller to send several bytes that inform the Playstation or the interpreter of their position or pressure.

Playstation 2 Controller in Analog Red Mode										
Byte Number	*COMMAND Line*	*DATA Line*								
1	0x01	idle								
2	0x42	0x73								
3	idle	0x5A	*0*	*1*	*2*	*3*	*4*	*5*	*6*	*7*
4	idle	data	select	L3	R3	start	up	right	down	left
5	idle	data	L2	R2	L1	R1	▲	●	X	■
6	idle	data	Right Joystick: left = 0x00 right = 0xFF							
7	idle	data	Right Joystick: up = 0x00 down = 0xFF							
8	idle	data	Left Joystick: left = 0x00 right = 0xFF							
9	idle	data	Left Joystick: up = 0x00 down = 0xFF							

Table 1: The first three data bytes at initiation of conversation, and the key assignments (Authorization for reprinting pending).

3.4.2 The Xbee OEM RF Module Transceiver

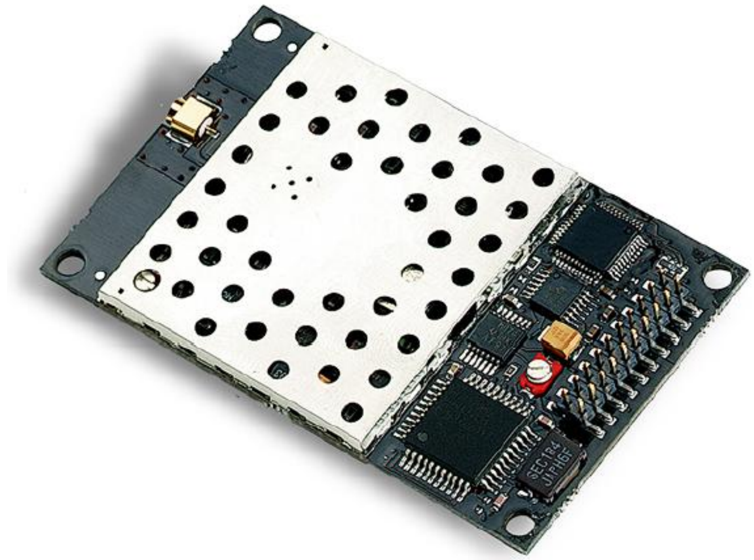


Figure 3: The Xbee OEM Module is a low-cost, low-power wireless transceiver. It requires minimal power and provides reliable delivery of data between devices. It operates within the ISM 2.4 GHz frequency band.

Long Range Data Integrity	
Indoor/Urban	up to 100 feet
Outdoor line-of-sight	up to 300 feet
Transmit Power	1 mW (0 dBm)
Receiver Sensitivity	-92 dBm
Low Power	
TX Current	45 mA (@3.3 V)
RX Current	50 mA (@3.3 V)
Power-Down Current	<10uA

Table 2

The Xbee RF module interfaces to a host device through a logic-level serial port. Through its serial port, the module can communicate through a level translator to any serial device. By default, the Xbee RF module operates in Transparent Mode, meaning it acts as a serial line replacement. All data received through the DI pin is queued up for RF transmission. When RF data is received, the data is sent out the DO pin.

The Xbee module is configured by either using Windows' Hyper Terminal or Digi's X-CTU, available to download for free from their website. To avoid using AT commands, the latter is preferable as it presents the parameters of the module in a list, which can be changed by a simple click of the mouse.

3.4.3 BasicMicro's Atom Pro 28-M

The BasicAtom Pro 28-M is the microcontroller chosen for this project. It is used and recommended by many robotics enthusiasts, mostly due to its large EEPROM and RAM storage space, necessary for the storage of constants and variables involved in calculations needed for the gait of the robot. It is also due to the convenient peripheral functions of the Atom Pro 28's I/O pins.

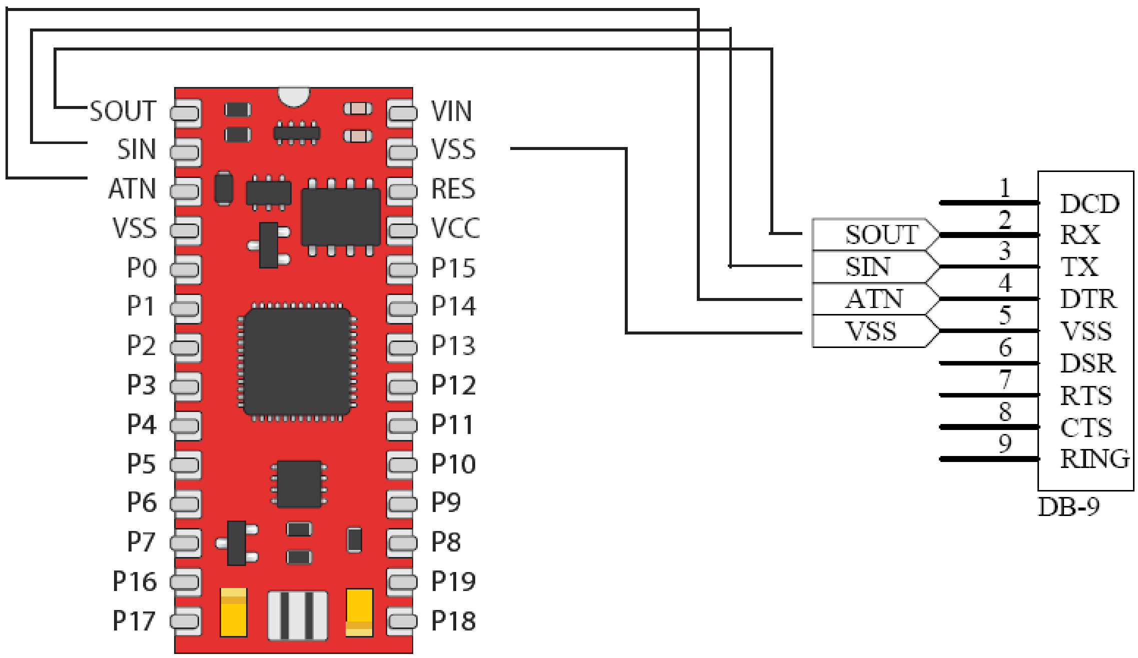
This microcontroller is easier to use than most microcontrollers because it can be programmed in BASIC language, which permits the architecture of the code to be simpler and more perceivable. BasicMicro provides the development program for free, AtomPro IDE, for download on their website www.basicmicro.com .

The Atom Pro 28-M features 32 KB of Program Space or Flash, 2 KB of User Memory or RAM, 4 KB of User Data Storage or EEPROM. It has 20 Input/Output pins, 8 of which are Analog-to-Digital capable I/O pins. It comes equipped with a built-in 5V regulator, 3 Hardware Timers, and 3 Hardware PWM I/O Pins. It is capable of interrupts, and can perform 32-bit floating point math and 32-bit integer math. Some additional characteristics are provided in the table below:

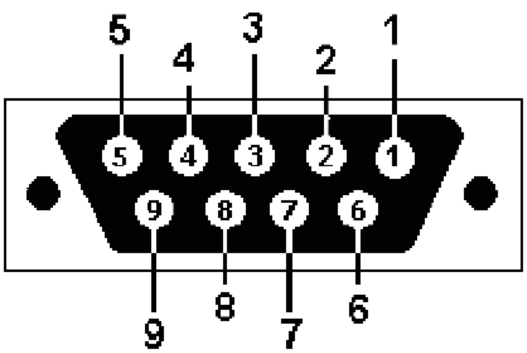
Characteristic		**Value (Units)**
VIN Range (min-max)	6 – 9VDC	
VCC Range (min-max)		4.9 – 5.2VDC
Current Draw (Sleep Mode)		10mA
Current Draw (maximum)		50mA
Current Draw (Idle)		20mA
I/O Logic		TTL
I/O Voltages (Low/High)		0.0V/5.0V
I/O Maximum Current	3 mA sink, 3 mA source	
Temperature Range	-20 to +75 C	

Table 3

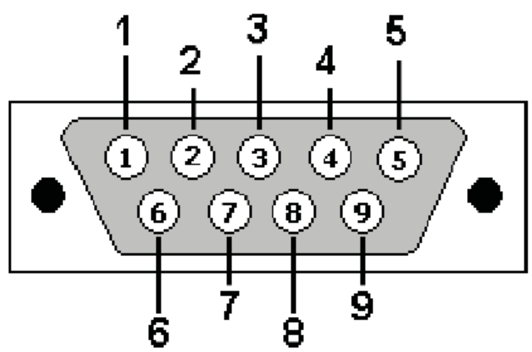
Another reason the Atom Pro is convenient is the fact that a development board or any other complex circuit is not required to program this microcontroller. It can be programmed from a RS232 serial port. Four connections (four pins of the microcontroller) to a RS232 DB9 cable are required (or a USB to serial adapter). The diagram below shows the necessary connections:



Below are pinouts for both male and female RS232 DB9 cabling:



DB9 View Looking into Female Connector



DB9 View Looking into Male Connector

DB9 PIN	SIGNAL	IN/OUT	DESCRIPTION	ATOM PRO PIN
1	DCD	In	Data Carrier Detect	NC
2	RXD	In	Receive Data	SOUT
3	TXD	Out	Transmit Data	SIN
4	DTR	Out	Data Terminal Ready	NC
5	VSS	-	Ground	VSS
6	DSR	In	Data Set Ready	NC
7	RTS	Out	Request To Send	ATN
8	CTS	In	Clear To Send	NC
9	RI	In	Ring Indicator	NC

Table 4: The output voltage characteristics of the AC4424-10 (Authorization to reprint pending).

Maximum System Throughput and System Timing

When configured as shown in the table above, an AC4424 transceiver is capable of achieving the listed throughputs, but may not meet these throughputs in the presence of interference or at longer ranges.

RF Mode	Interface Baud Rate	Duplex	Direction	Throughputs (bps)
Acknowledge	115200	Half	One way	80k
Acknowledge	115200	Full	Both ways	40k

Table 5: Maximum throughputs of the AC4424-10 Transceivers according to certain configurations (Authorization to reprint pending).

When configured as shown in the table above, an AC4424 transceiver is capable of achieving the listed throughputs, but may not meet these throughputs in the presence of interference or at longer ranges.

The AC4424 is a frequency hopping spread spectrum radio. Frequency hopping allows the system to hop around interference in order to provide a better wireless link. Listed in table 6 below is the duration of the hop and of the hop cycle (period).

Parameter	Typical Time (ms)
Hop Time	1
Hop Period	8

Table 6: Duration of the hop and the hop cycle of the AC4424-10 Transceiver (Authorization to reprint pending).

AC4424 AT Commands and Command Mode

The AT Command mode implemented in the AC4424-10 creates a virtual version of the Command/Data pin. The “Enter AT Command Mode” Command asserts this virtual pin low (to enable Command Mode) and the “Exit AT Command Mode” Command asserts this virtual pin high (to enable Data). When in AT Command Mode, the user cannot send or receive RF packets. However, a period of approximately 10ms exists where, if the “Enter AT Command Mode” command has been sent to the transceiver at the same time an RF packet is being received, the RF packet could be sent to the host circuitry before the “Enter AT Command Mode” command response is sent to the host circuitry.

Prior to sending the “Enter AT Command Mode” command to the transceiver, the host circuitry must ensure that the RF transmit buffer of the transceiver is empty (if the buffer is not empty, the ”Enter AT Command Mode” command will be interpreted as packet data and will be transmitted out over the RF). This can be accomplished by waiting up to one second between the last transmit packet and the AT Command. The host circuitry must also ensure that the RF Packet Size for the transceiver is set to a minimum of six. The Enter AT Command mode command is as follows:

Host Circuit Command:

41h	54h	2Bh	2Bh	2Bh	0Dh

Transceiver Response:

CCh	43h	4Fh	4Dh

The Exit AT Command is as follows:

Host Circuit Command:

CCh	41h	54h	4Fh	0Dh

Transceiver Response:

CCh	44h	41h	54h

Various commands for the transceiver are listed in Table 4.3 of the datasheet documenting the AC4424 transceiver, located in Appendix A.

3.4.3 The Microcontrollers

3.4.3.1 Microchip's PIC18F458 MCU

The PIC18F458 MCU is a low-power 8-bit microcontroller based on RISC (Reduced Instructions Set Computer) architecture equipped with high speed Enhanced Flash technology and program memory addressing up to 2 Mbytes.

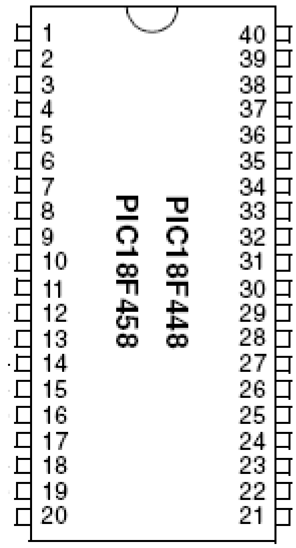


Figure 4: The PIC18F458 MCU from Microchip (Authorization to reprint obtained from Microchip: refer to A.3).

Absolute Maximum Ratings	
Operating Temperature	-40°C to +125°C
Storage Temperature	-65°C to +150°C
Voltage on any pin with respect to V_{SS} (except V_{DD}, MCLR and RA4)	-0.3V to (VDD + 0.3V)
Voltage on V_{DD} with respect to V_{SS}	-0.3V to +7.5V
Voltage on MCLR with respect to VSS	0V to +13.25V
Voltage on RA4 with respect to VSS	0V to +8.5V
Total power dissipation	1.0 W
Maximum current out of VSS pin	300 mA
Maximum current into VDD pin	250 mA
Input clamp current, IIK (VI < 0 or VI > VDD)	±20 mA
Output clamp current, IOK (VO < 0 or VO > VDD)	±20 mA
Maximum output current sunk by any I/O pin	25 mA
Maximum output current sourced by any I/O pin	25 mA
Maximum current sunk by all ports (combined)	200 mA
Maximum current sourced by all ports (combined)	200 mA

Table 7: Absolute Maximum Ratings of the PIC18F458 MCU (Authorization to reprint obtained from Microchip: refer to A.3).

DC Characteristics (TA = -40°C to 85°C)						
Symbol	**Parameter**	**Condition**	**Min**	**Typ**	**Max**	**Units**
V_{DD}	Supply Voltage	HS, XT, RC and LP Oscillator modes	2.0		5.5	V
V_{DR}	RAM Data Retention Voltage		1.5			V
V_{POR}	VDD Start Voltage for Power-on Reset signal		0.05			V/ms

V_{BOR}	Brown-out Reset					
	Voltage		1.96		2.16	V
	BORV1:BORV0 = 11		2.64		2.92	V
	BORV1:BORV0 = 11		4.07		4.59	V
	BORV1:BORV0 = 11		4.36		4.92	V
	BORV1:BORV0 = 11					
I_{DD}	Supply Current	XT Oscillator Configuration				
		FOSC = 4 MHz				
		VDD = 2.0V, +25°C,		0.7	2	mA
		VDD = 2.0V, -40°C to +85°C		0.7	2	mA
		VDD = 4.2V, -40°C to +85°C		1.7	4	mA
		RC oscillator configuration				
		VDD = 2.0V, +25°C		1	2.5	mA
		VDD = 2.0V, -40°C to +85°C		1	2.5	mA
		VDD = 4.2V, -40°C to +85°C		2.5	5	mA
		RCIO oscillator configuration				
		VDD = 2.0V, +25°C		0.7	2.5	mA
		VDD = 2.0V, -40°C to +85°C		0.7	2.5	mA
		VDD = 4.2V, -40°C to +85°C		1.8	4	mA

Table 8: The DC Characteristics of the PIC18F458 MCU (Authorization to reprint obtained from Microchip: refer to A.3)

Architecture Overview

Features		PC18F458 MCU
Operating Frequency		DC – 40 MHz
Internal Program Memory	Bytes	32 K
	# of Single-Word Instructions	16384
Data Memory (Bytes)		1536
Data EEPROM Memory (Bytes)		256
Interrupt Sources		21
I/O Ports		Ports A, B, C, D, E
Timers		4
Capture/Compare/PWM modules		1
Enhanced Capture/Compare/PWM modules		1
Serial Communications		MSSP, CAN, Addressable USART
Parallel Communications		Yes
10-bit Analog-to-Digital		8 input channels
Analog Comparators		2
Analog Comparators VREF Output		Yes
Resets (and Delays)		POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST)
Programmable Low-Voltage Detect		Yes
Programmable Brown-out Reset		Yes
CAN Module		Yes
In-Circuit Serial Programming (ICSP)		Yes
Instruction Set		75 instructions

Table 9: Some features of the PIC18F458 MCU (Authorization to reprint obtained from Microchip: refer to A.3).

3.4.3.2 Atmel's AT90S2313

The AVR AT90S2313 is a low-power CMOS 8-bit microcontroller based on Atmel's AVR RISC (Reduced Instructions Set Computer) architecture. It has 32 general purpose working registers that are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle.

Absolute Maximum Ratings	
Operating Temperature	-55°C to +125°C
Storage Temperature	-65°C to +150°C
Voltage on any pin except RESET with respect to ground	-1.0V to VCC+0.5V
Voltage on RESET with respect to ground	-1.0V to +13.0V
Maximum Operating Voltage	6.6V
DC Current per I/O pin	40.0 mA
DC Current VCC and GND pins	40.0 mA

Table 10: Absolute Maximum Ratings of the AT90S2313 microcontroller (Authorization to reprint obtained from Atmel: refer to A.4).

DC Characteristics (TA = -40°C to 85°C, VCC = 2.7V to 6.0V)						
Symbol	**Parameter**	**Condition**	**Min**	**Typ**	**Max**	**Units**
V_{IL}	Input Low Voltage	(Except XTAL1)	-0.5		$0.3V_{CC}$	V
V_{IL1}	Input Low Voltage	(XTAL1)	-0.5		0.1	V
V_{IH}	Input High Voltage	(Except XTAL1, RESET)	$0.6V_{CC}$		$V_{CC}+0.5$	V
V_{IH1}	Input High Voltage	(XTAL1)	$0.7V_{CC}$		$V_{CC}+0.5$	V
V_{IH2}	Input High Voltage	(RESET)	$0.85V_{CC}$		$V_{CC}+0.5$	V
V_{OL}	Output Low Voltage (Ports B, D)	I_{OL} = 20 mA, V_{CC} = 5V I_{OL} = 10 mA, V_{CC} = 3V			0.6 0.5	V V

V_{OH}	Output High Voltage (Ports B, D)	I_{OH} = -3 mA, V_{CC} = 5V I_{OH} = -1.5 mA, V_{CC} = 3V	4.3 2.3			V V
I_{IL}	Input Leakage Current I/O pin	Vcc = 6V, pin low (absolute value)			1.5	μA
I_{IH}	Input Leakage Current I/O pin	Vcc = 6V, pin high (absolute value)			980	nA
RRST	Reset Pull-up Resistor		100		500	kΩ
$R_{I/O}$	I/O Pin Pull-Up Resistor		35		120	kΩ
I_{CC}	Power Supply Current	Active Mode V_{CC} = 3V, 4MHz Idle Mode V_{CC} = 3V, 4MHz			3.0 1.0	mA mA
I_{CC}	Power Down Mode	WDT enabled, V_{CC} = 3V WDT disabled, V_{CC} = 3V		9 <1	15.0 2.0	μA μA
V_{ACIO}	Analog Comparator Input Offset Voltage	V_{CC} = 5V			40	mV
I_{ACLK}	Analog Comparator Input Leakage Current	V_{CC} = 5V Vin = V_{CC}/2	-50		50	nA
t_{ACPD}	Analog Comparator Propagation Delay	V_{CC} = 2.7V V_{CC} = 4.0V		750 500		ns

Table 11: The DC Characteristics of the AT90S2313 microcontroller (Authorization to reprint obtained from Atmel: refer to A.4).

Pin Descriptions

PIN1: RESET

Reset input. A low level on this pin for more than 50 ns will generate a reset, even if the clock is not running. Shorter pulses are not guaranteed to generate a reset.

PIN 2-PIN 3, PIN 6-PIN 9, PIN 11: PORT D (PD0...PD6)

Port D has seven bi-directional I/O port with internal pull-up resistors, PD6...PD0. The Port D output buffers can sink 20 mA. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not active.

PIN 4: XTAL2

Output from the inverting oscillator amplifier.

PIN 5: XTAL1

Input to the inverting oscillator amplifier and input to the internal clock operating circuit.

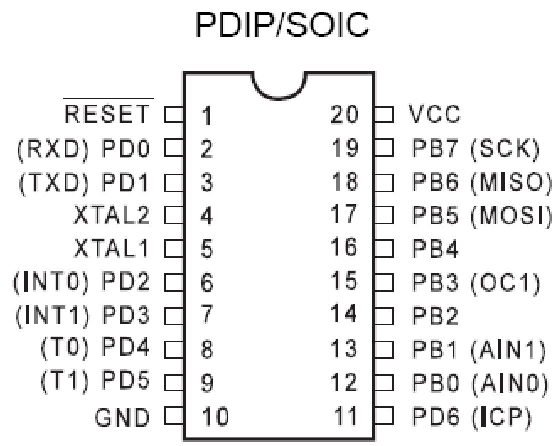


Figure 5: Pin configuration of the AT90S2313 microcontroller (Authorization to reprint obtained from Atmel: refer to A.4).

PIN 10: GND

Ground pin.

PIN 20: VCC

Supply voltage pin.

PIN 12 – PIN 19: PORT B (PB0...PB7)

Port B is an 8-bit bi-directional I/O port. Port pins can provide internal pull-up resistors (selected for each bit). PB0 and PB1also serve as the positive input (AIN0) and the negative input (AIN1), respectively, of the on-chip analog comparator. The Port B output buffers can sink 20mA and can drive LED displays directly. When pins PB0 to PB7 are used as inputs and are externally pulled low, they will source current if the internal pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not active.

Architecture Overview

The AT90S2313 provides the following features: 2K bytes of In-System Programmable Flash, 128 bytes EEPROM, 128 bytes SRAM, 15 general-purpose I/O lines, 32 general-purpose working registers, flexible timer/counters with compare modes, internal and external interrupts, a programmable serial UART, programmable Watchdog Timer with internal oscillator, an SPI serial port for Flash memory downloading and two software selectable power-saving modes. The on-chip In-System Programmable Flash allows the program memory to be reprogrammed in-system through an SPI serial interface or by a conventional nonvolatile memory programmer.

In-System Programmable Flash

The AT90S2313 contains 2K bytes on-chip In-System Programmable Flash memory for program storage. Since all instructions are 16- or 32-bit words, the Flash is organized as 1K x 16. The Flash memory has an endurance of at least 1000 write/erase cycles. The AT90S2313 Program Counter PC is 10 bits wide, thus addressing the 1024 program memory addresses.

EEPROM

The AT90S2313 contains 128 bytes of EEPROM data memory. It is organized as a separate data space, in which single bytes can be read and written. The EEPROM has an endurance of at least 100,000 write/erase cycles.

The EEPROM access registers are accessible in the I/O space. The write access time is in the range of 2.5 - 4ms, depending on the VCC voltages. A self-timing function, however, lets the user software detect when the next byte can be written. When the EEPROM is read or written, the CPU is halted for two clock cycles before the next instruction is executed.

UART

The AT90S2313 features a full duplex (separate receive and transmit registers) Universal Asynchronous Receiver and Transmitter (UART). The main features are:

- Baud rate generator that can generate a large number of baud rates (bps)
- High baud rates at low XTAL frequencies
- 8 or 9 bits data
- Noise filtering
- Overrun detection
- Framing Error detection
- False Start Bit detection
- Three separate interrupts on TX Complete, TX Data Register Empty and RX Complete

Data transmission is initiated by writing the data to be transmitted to the UART I/O Data Register, UDR. Data is transferred from UDR to the Transmit shift register when:

- A new character has been written to UDR after the stop bit from the previous character has been shifted out. The shift register is loaded immediately.
- A new character has been written to UDR before the stop bit from the previous character has been shifted out. The shift register is loaded when the stop bit of the character currently being transmitted has been shifted out.

The receiver front-end logic samples the signal on the RXD pin at a frequency 16 times the baud rate. While the line is idle, one single sample of logical zero will be interpreted as the falling edge of a start bit, and the start bit detection sequence is initiated.

Let sample 1 denote the first zero-sample. Following the 1 to 0-transition, the receiver samples the RXD pin at samples 8, 9 and 10. If two or more of these three samples are found to be logical ones, the start bit is rejected as a noise spike and the receiver starts looking for the next 1 to 0-transition. If however, a valid start bit is detected, sampling of the data bits following the start bit is performed. These bits are also sampled at samples 8, 9 and 10. The logical value found in at least two of the three samples is taken as the bit value. All bits are shifted into the transmitter shift register as they are sampled.

The baud rate generator is a frequency divider which generates baud-rates according to the following equation:

$$\mathrm{BAUD} = \frac{f_{\mathrm{CK}}}{16(\mathrm{UBRR}+1)}$$

BAUD = Baud-Rate
f_{CK} = Crystal Clock frequency
UBRR = Contents of the UART Baud Rate register, UBRR (0-255)

The UBRR register is an 8-bit read/write register which specifies the UART Baud Rate according to the formula above.

SRAM

224 Data Memory locations address the Register file, I/O Memory and the data SRAM. The first 96 locations address the Register File + I/O Memory, and the next 128 locations address the data SRAM. The five different addressing modes for the data memory cover: Direct, Indirect with Displacement, Indirect, Indirect with Pre-Decrement and Indirect with Post-Increment. In the register file, registers R26 to R31 feature the indirect addressing pointer registers. The Direct addressing reaches the entire data address space. The Indirect with Displacement mode features 63 address locations reach from the base address given by the Y and Z registers. When using register indirect addressing modes with automatic pre-decrement and post-increment, the address registers X, Y and Z are used and decremented and incremented. The 32 general purpose working registers, 64 I/O registers and the 128 bytes of data SRAM in the AT90S2313 are all directly accessible through all these addressing modes.

I/O

All AT90S2313 I/O and peripherals are placed in the I/O space. The I/O locations are accessed by the IN and OUT instructions transferring data between the 32 general purpose working registers and the I/O space. I/O registers within the address range $00 - $1F are directly bit-accessible using the SBI and CBI instructions. In these registers, the value of single bits can be checked by using the SBIS and SBIC instructions. When using the I/O specific commands IN, OUT the I/O addresses $00 - $3F must be used. When addressing I/O registers as SRAM, $20 must be added to this address. The CBI and SBI instructions work with registers $00 to $1F only.

Internal and External Interrupt Sources

The AT90S2313 provides 10 different interrupt sources. These interrupts and the separate reset vector, each have a separate program vector in the program memory space. All the interrupts are assigned individual enable bits which must be set (one) together with the I-bit in the status register in order to enable the interrupt. The lowest addresses in the program memory space are automatically defined as the Reset and Interrupt vectors. The lower the address the higher is the priority level. RESET has the highest priority, and next is INT0 - the External Interrupt Request 0, etc.

General-purpose Working Registers

Each register is assigned a data memory address, mapping them directly into the first 32 locations of the user Data Space. Although the register file is not physically implemented as SRAM locations, its memory organization provides great flexibility in access of the registers, as the X, Y and Z registers can be set to index any register in the file.

Timer/Counters with Compare Modes

The AT90S2313 provides two general purpose Timer/Counters - one 8-bit T/C and one 16-bit T/C. The Timer/Counters have individual prescaling selection from the same 10-bit prescaling timer. Both Timer/Counters can either be used as a timer with an internal clock timebase or as a counter with an external pin connection which triggers the counting.

Programmable Watchdog Timer with Internal Oscillator

The Watchdog Timer is clocked from a separate on-chip oscillator which runs at 1MHz This is the typical value at VCC = 5V. By controlling the Watchdog Timer prescaler, the Watchdog
reset interval can be adjusted. The WDR - Watchdog Reset - instruction resets the Watchdog Timer. Eight different clock cycle periods can be selected to determine the reset period. If the reset period expires without another Watchdog reset, the AT90S2313 resets and executes from the reset vector.

3.5 Power

3.5.1 Power Supply System

In today’s technology, robots have become commonplace, and are used in a number of different applications. Our robot which is intended to be a spy-bot, it has distinct characteristics such as color sensing abilities. The robot is semi-autonomous, because some of its functions are controlled through a remote control. For the robot to operate properly, an adequate power supply is needed. The components and subsystem of the robot are susceptible to power surges, and noise interferences, and often require large amounts of currents. Therefore the design of the power supply can efficiently handle interferences from the different sources while delivering efficiently the amount of power required by the different subsystems of the robot. After researching the different ways of building the robot’s power circuit and voltage conversions, the best and most efficient way to build the power system that delivers power and regulate the robot’s voltages efficiently is chosen. The robot is able to run for at least one hour continuously. Since this is a low cost budget, rechargeable batteries are used to minimize the project’s cost. The overall power supply of the robot combines low cost and efficiency to give us the best functional robot possible.

The design of the robot’s power supply is as efficient as possible to allow the main battery source to run as long as possible before needing recharge. The design of a power supply for a particular robot, takes into account these things and along with the consideration of cost, time, and space. The power supply of the robot meets the power requirement of every subsystem, and allow for future project expansion if needed, while the voltage regulators efficiently maximize the robot’s operating time. Calculations and results show the efficiency of the regulators and voltage at rated load. Discussions and research provided insight into need for appropriate component selections and placement, ensuring a proper and successful implementation of a DC power converter, and printed circuit board implementation.

Specifications

In the design of the power supply, thus far we have specified that

1. The main voltage source of the robot are two DC Battery
2. The source voltage is stepped down using regulators
3. The power supply system accounts for noise issues and interferences
4. Each subsystems have different voltage distortions
5. Isolation is used to reduce the effects of noise and interferences between the subsystems

- Voltage Regulators
- 1 On/Off Switch
- Capacitors
- 2 Batteries

- Resistors
- 1LED

Section 4: Research and Investigation

A significant amount of research was necessary in order to properly design this project. Given the scope of the project and the varied subfields of electrical and computer engineering that are involved, the research and design components were divided amongst the members according to both their interests and skills. This was also a natural split due to the strict schedule that must be adhered to in order to both design and build the project. By having different members become experts in their given subsystem, the team is best able to leverage skills while carefully balancing time. Although the research component for each subsystem was done by individuals, the same goal was had in mind throughout. The parts or components to be obtained must satisfy the objectives outlined in *Section 2* while staying to the specifications outlined in *Section 3* as much as possible. Before research even begun, it was expected that all of the specifications would be difficult to follow; there is a far cry between knowing the optimum piece needed for construction and actually finding it. Therefore the research aimed to find the best fit possible with whatever means were available.

Beyond the technical aspect of research and investigation, each team member also focused on procuring parts that kept us within our budget. That is, even if a part that fully met specifications was found, the cost of obtaining it was carefully balanced against its importance in the overall design of the project. By the very design of our approach to the problem of building the robot, it has been split up into a multitude of smaller problems. If a particular solution to one of these problems is simply too expensive to implement, then another solution must be found. The process of outlining specifications, research, and designing the project was both an iterative and recursive process. Even if a particular component is found that fully matches specification, there is no need to immediately believe that this part is the optimal combination of both price and performance on the market. Thus even if a matching component is found, an earnest attempt was made to find other competing companies that offered similar components. An earnest attempt was also made in considering if time would allow the piece to be custom built by the team member or members themselves; for example building the brackets necessary for motor mounting, or even the vision unit chassis, by taking a trip to the local hardware store and putting some ingenuity and creativity to good use. This was perhaps one of the more interesting and challenging aspects of the entire project, since navigating the current market is an extremely open-ended problem. For most of us, this was the first experience in parts hunting.

However, all members quickly noticed that the complexity of the markets is quite vast, and if good results were to be obtained quickly, certain methods were to be generated and adhered to. The system decided upon was to obtain a quick form of internet communication, where a list of vendors could be shared and any pertinent results posted quickly. Since not only do certain subsystems rely on each other for completion but the overall integration of the project does as well, the ability to quickly post results and evaluate them was integral to success. Also, by compiling a list of vendors and quickly notifying each other of current market trends or which companies seemed to be the best in a field, search time was drastically reduced. In the following sections, the detailed research and investigation process for the different subsystems is outlined. Competing products are compared and contrasted in relation to both our given specifications and objectives, and the choice made for the design portion is shown. For purposes of presentation, the final solution or product shown in the investigation portion will always be the design choice made for the project. Previous options are also demonstrated to show the research process and enable readers to make decisions for themselves.

4.1 Mechanics

4.1.1 Body Chassis

4.1.1.1 Handmade Custom

The first mechanical component investigated was the body chassis, since all other mechanical components will depend on it, and the rest of the project will depend on the mechanical components. At first thought, the group felt reasonably confident that building the body chassis ourselves would not be a big issue. If it was just to be a hunk of material that would be providing support, there shouldn't be much to making one. Although simply fashioning a piece of material to be of a certain size and weight in order to function as a chassis is not very difficult, making sure that it can easily enclose the necessary electronics and provide adequate ventilation further complicated the matter. Also, the connection joints to the legs were specified to all exist on the same two-dimensional plane for ease of construction, so the chassis had to somehow provide these connection points. Although initial plans were discussed, and it certainly was possible to construct the chassis, it was decided that designing a chassis, building the chassis, and making sure that it met specifications while being sufficiently rugged would require a disproportionate amount of time and effort. It was decided that a chassis would either be custom built by a company, or ordered from a robotics vendor.

4.1.1.2 Professional Custom

By searching online and asking other individuals who had a bit more experience, two different companies emerged as potential contractors for building our chassis. The two companies were *Pololu Robotics & Electronics* at www.customlasercutting.com, and *SuperDroid Robots* at www.superdroidrobots.com.

The first company looked at was *Pololu Robotics & Electronics.* Although they only provided custom laser cutting, they would also provide the material to be cut if it was needed. This was a great fact, since we did not want to spend the time in actually obtaining the material and then shipping it to them. They specifically stated that they neither stocked nor cut metal, but did stock and cut plastics. The thickest sheet they could cut was half an inch, which was fine. The laser cutting would cost $4.00 per ft^2, and so it might prove to be a little more than we hoped to spend on a body chassis. A significant problem with this company, respective to our needs, was that they would only cut the materials for us, we would still be responsible for assembling it and providing adequate enclosure for the electronics. In other words, while we needed a three-dimensional structure, they could only adequately provide two-dimensional structures. *SuperDroid Robots* provided much more promising services. However, they either required CAD drawings, or would design a CAD drawing for us. What was quickly realized was that even if the team opted for having a company build the chassis, the custom design was still going to come from the team. And if the design did not come from the team, the chassis would be way out of our budget. It seemed then that the best solution given the current constraints of our project would be to search for an already designed and already built chassis sold by a robotics vendor. At this stage in research, if a viable chassis was not found for our hexapod robot, either resource allocation or project specifications would have to be altered. Either a different chassis would be required, or extra time and money would have to be spent in custom building the body chassis.

4.1.1.3 Prebuilt

HexCrawler Hardware Kit

After significant searching online, three choices for the body chassis were discovered. The first choice looked extremely cool, and actually solved multiple problems for us right off the bat. It was the *HexCrawler Hardware Kit* from www.crustcrawler.com. The kit essentially solved the entire problem of building a mechanical system, since it already included the legs and the joints coupled with the body chassis. The only mechanical part left would be the camera chassis that would be mounted on top of it. A fully assembled kit is shown in figure 6 below.

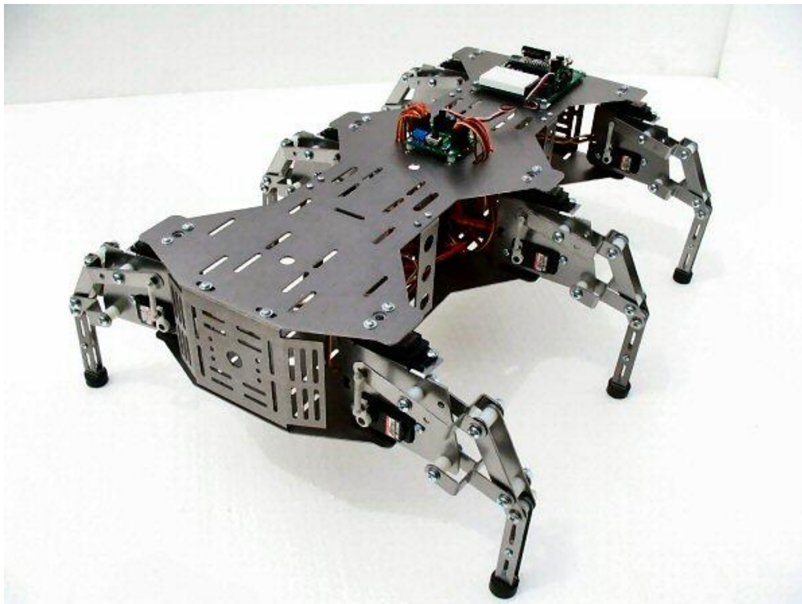


Figure 6: HexCrawler Hardware Kit, www.crustcrawler.com, (reprinted w/ permission)

Since the rest of the project still contains much electrical and computer engineering work, there was much deliberation in the group regarding purchasing this kit. A couple of problems emerged however. For one, the electronics enclosure was fairly small, and we did not believe that all necessary electronics would fit inside without significant effort. This would limit the flexibility of our design further down the road. Another deterrent was the fact that the chassis was initially built for only 2 degrees of freedom per leg. More complicated dynamics could not be achieved by this mechanical structure, unless certain adjustments were made. Thus the team was already looking at having to modify the chassis, something that was originally to be avoided. Finally, and perhaps most importantly, the $299 price tag of the kit seemed a little too high. Investigating other options, especially buying the chassis, legs, and joints separately, might yield an ultimate price tag that was significantly less than $299 and more in line with our budget.

Rectangular Body Chassis

The next chassis that was investigated was a rectangular body chassis from www.LynxMotion.com. During our overall research for products, it was discovered that LynxMotion is one of the largest suppliers of robotic parts online. Many other web sites are actually selling LynxMotion's products. This was a market trend noticed, and was quickly posted online to the entire group so others could better focus their searches.

It turns out that LynxMotion specializes in hexapod robots. They sell many premade robots for several hundred dollars, and also sell kits for individuals who want to build the robots themselves. However, they sell kit pieces individually as well. This meant that the chassis sold in their hexapod kits were available for individual sale. The first hexapod chassis investigated was their rectangular body chassis. This chassis is shown in figure 7 below.

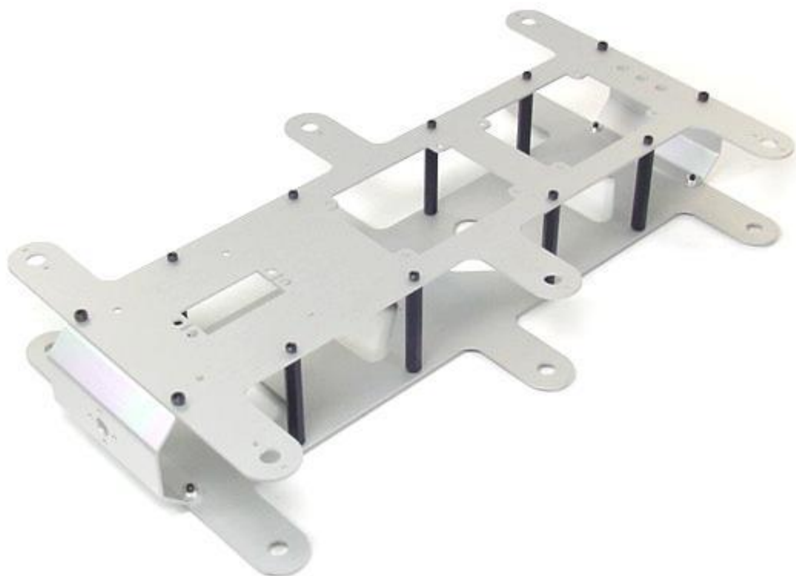


Figure 7: Hexapod Rectangular Chassis from www.LynxMotion.com (reprinted w/ permission)

The total shipping weight of the chassis is .73 lbs. therefore it met the weight spec. Its length was a little over 2 feet, so it also met the length specification. The chassis was also desirable due to it being constructed from aluminum (enabling adhesion) and the fact that it had plenty of space for enclosing electronics. All of the leg connections also lay on the same two-dimensional plane as was specified. The price of the chassis was $49.95, which was considered reasonable. It certainly seemed like it was the beginning of a much better bargain, compared to the $299 price tag on the *HexCrawler Kit*. However, it admittedly did not look nearly as cool.

Circular Body Chassis

LynxMotion also vends a circular body chassis for hexapod robots. Similar to the rectangular chassis, it contained plenty of room to store necessary electronics and met necessary weight and size specifications. In fact, this chassis was a bit smaller, and weighed a bit less with a shipping weight of .66 lbs. A circular chassis is shown in figure 8 below.

Figure 8: Hexapod Circular Chassis from www.LynxMotion.com (reprinted w/ permission)

The circular chassis cost $39.95, which was ten dollars cheaper than the rectangular chassis. Since this chassis met all necessary specifications, was lighter, smaller, and less expensive than the rectangular chassis the team was very interested in it. We also thought the cool black color looked better. Therefore, the team opted to choose the circular body chassis from LynxMotion. However, another important reason for this choice was that it would enable the hexapod to have great mobility since it could move in any direction at once. If a rectangular chassis were chosen, the robot would have to rotate before changing its heading. With a circular chassis, the robot can choose to move in any direction at any point in time sans rotation.

4.1.2 Vision Unit Chassis

4.1.2.1 Handmade Custom

Although it was decided that constructing the body chassis internally was not the best way to go about doing things, it was again considered for the vision unit chassis. The vision unit chassis would be much smaller, and would not be nearly as intricate as the body chassis. It was decided though, that professional building of the vision unit chassis would not be investigated, since the same problem that occurred for the body chassis would occur here. Although quick designs for the vision unit chassis were drawn on whiteboard, and ideas were thrown around during a meeting, it was decided that someone could check online vendors simultaneously to see what was available. A suitable vision unit chassis was quickly found online, and this option was abandoned. However, the team would like to note, that a vision unit chassis is easily constructed, and should not be contracted unless it is found to be relatively inexpensive to other considerations.

4.1.2.2 Prebuilt

LynxMotion provides an add-on deck to their body chassis. The add-on deck has a mounting hole on the top panel for a motor, which is perfect for our application. Since the add-on deck was only $9.95, and could easily be added to our existing body chassis, it was chosen immediately. Perhaps further investigation might have yielded a competing chassis, but the opportunity cost of such a search was thought to be too much when this one was right in front of us. Although the chassis did not exactly meet the initial specification of having 5 inches below for electronics housing, due to the fact that it can connect directly into the existing body chassis and that it already has a mounting hole, this was considered not to be a problem. An add-on deck is shown in figure 9 below.

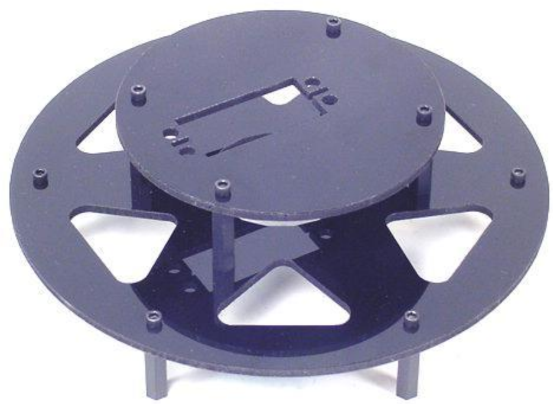


Figure 9: Camera Chassis from www.LynxMotion.com (reprinted w/ permission)

4.1.3 Legs and Joints

4.1.3.1 Single Legs

Although the team felt capable of being able to produce a chassis themselves, legs are a different matter. There was not enough time to do so. It was agreed upon that the legs would be bought from a professional manufacturer. Due to the way the mechanical system had been identified and outlined, the team believed that obtaining legs and joints would be a separate process. Thus this is the way research initially began. Various vendors were researched in this area, but it turns out that all of them were reselling merchandise from LynxMotion. Therefore, the group went straight to the source: LynxMotion's website.

It was decided that the first task would be to obtain a valid pair of legs, and then decide how they would be connected to the chassis itself. Two different types of legs were looked at initially. The first one was the *Robot Leg B Pair*, and would cost $19.95. It's size fit specifications, at about 4.9 inches, and its shipping weight was only .23 lbs. Figure 7, shown below, is a picture of this model. The servomotor shown in figure 10 is not included in the actual product.

Figure 10: Robot Leg B from www.LynxMotion.com (reprinted w/ permission)

The second leg looked at, also cost $19.95 and roughly had all of the same specifications as the previous leg. It is called the *Robot Leg A Pair* at LynxMotion. The group liked this model more, since it was closer to what they originally had in mind. That is, although there would be no apparent performance differences, and the specifications were roughly identical, this model was more aesthetically pleasing. The lower curvature of the leg model resembled the hexapod structure that the group had in mind. Since all costs would be the same the group simply opted for the more curved leg since in their opinion it looked cooler. Figure 11 shown below is a picture of the leg model, and the reader can make their own choice. Again, the leg does not come with a servomotor, it is only there for illustration purposes.

Figure 11: Robot Leg A from www.LynxMotion.com (reprinted w/ permission)

Thus it was decided that this model was the preferred leg model, and would be the one used for the project. However, this leg alone would only provide a single degree of freedom, hardly enough for the proposed drive system. Since the original plan was to first find a suitable leg, and then to find suitable joints to articulate it, the original plan was to obtain joints after the legs; to obtain them separately. It was learned during the research phase of this project that many times one goes with their gut instinct, which is usually dependent on previous solutions to previous problems. However, this instinct might not always be the best solution to a problem, and in hindsight it often isn't. Thus a bit of investigation and study should go towards every decision, even if the apparent solution that one was looking for is found.

4.1.3.2 Leg Pair Packages

Although suitable joints were found on the same site, their use would have required a bit of modification to the legs and to our original design. In essence, they would be brackets enabling the legs to rotate, and giving space to mount the motors onto the design. Due to the extra difficulty, an email was sent to LynxMotion prior to ordering explaining the scope of the problem and if they knew of any solutions. Much to our surprise, there was another solution. The solution had eluded us in the past, since the product was fairly difficult to find while navigating LynxMotion's website. Ironically, by shopping other third party vendors, such as www.active-robots.com, one could reach the product sooner. The solution was to buy a leg, with three degrees of freedom already incorporated; in essence, buying the legs and the joints together. Logically, this would cost more money.

Since the solution was appealing, but more expensive, further research was done prior to accepting it. By comparing, superficially, the cost of a 3 DOF leg pair with joints and what it would cost to buy the proper joints and the legs separately, the team realized that no significant savings would be made in money if building the legs separately was chosen. Compounding this fact with the fact that the team would still have to do further research in order to understand how to actually put the leg and the joints together, it was decided that it was not worth it. This is another example in the project of trying to balance the expenditure of time and money; both are crucial resources.

The leg and joint packages were narrowed down to two different choices. Although one model uses the *Robot Leg Pair A* and the other model uses the *Robot Leg Pair B* as shown in Figures 7 and 8 respectively, the decision made here would be influenced by more than aesthetics. The first package looked at is shown in Figure 12 below; it is the rugged 3DOF aluminum leg pair. Motors are not included.

Figure 12: Rugged 3DOF Alum. Leg Pair from www.LynxMotion.com (reprinted w/ permission)

The package cost $87.90 (two legs) and had a shipping weight of .4 lbs. The dimensions were as follows:

- Hip Hor. to Hip Vert. = 38mm
- Hip Vert. to Knee Vert. (Femur) = 57mm
- Knee to Foot (Tibia) = 124mm

These measurements correspond to the measurements depicted in **Section 3**, under specifications, and can be seen on the figure by looking at it from right to left, where the end of each joint or bracket is corresponds to one of the measurements from the top down. These measurements are within range of the specifications outlined.

The second leg package looked at, from LynxMotion as well, was an identical 3 DOF aluminum leg pair where only the foot to knee portion was changed. Figure 13 is a picture of this leg package.

Figure 13: 3DOF Alum. Leg Pair from www.LynxMotion.com (reprinted w/ permission)

This leg package was implied to not be as rugged as the previous one by its description. However, its price was $18.25 cheaper, at $69.65. Since 3 leg pairs were going to be bought, if this pair was chosen over the other pair, it would amount to a savings of $54.75, which nearly amounts to the price of another leg pair. Based on price alone this seemed like a better buy, but the curved aesthetics of the leg also made it convincing. When analyzing the specs of this package:

- Hip Hor. to Hip Vert. = 29mm
- Hip Vert. to Knee Vert. (Femur) = 57mm
- Knee to Foot (Tibia) = 141mm

It was noted that although the first and last leg portions differed by a significant amount, the total leg length in both packages differed only by 8mms, where this package was longer. This made sense, since the curved knee to foot portion would need to be a bit longer in order to support the same amount of weight that the straight version could. Since the leg pair was significantly less expensive, was more aesthetically appealing, and met necessary specifications, three of these leg pairs were ordered. The fact that these pairs were not as rugged as the previous ones were discussed with LynxMotion staff, and they mentioned that the difference was minute and that the leg pairs should have no problem meeting the team's specifications.

4.2 Drive

The research for the drive subsystem involves identifying both hardware capable of making the robot move, and then the necessary software to actuate the hardware. First the hardware portion is researched and identified, and the walking or gait algorithm is researched and identified.

4.2.1 Hardware

4.2.1.1 Servomotors

Since the position of each joint would have to be meaningfully manipulated by motors, then the position of each motor must be easily manipulated as well. Since servomotors already have a built in control system allowing the user to easily move its position, they were the motors of choice for the hardware portion. Therefore, 18 servomotors would be bought where each of them would individually control each degree of freedom to the precision necessary.

The research question was then identifying the proper servomotors to do the job. Since the team currently had a chassis and legs from LynxMotion, which was the identical combo used in some of their robots, it was natural to find out what type of servomotor, in terms of torque and response speed, was used in their robots. The answer was a Hitec HS-475HB servomotor. Figure 14 below demonstrates this motor.

Figure 14: Hitec HS-475HB servomotor from LynxMotion.com (reprinted w/ permission)

Despite clear specifications regarding torque and weight regarding each servomotor, the team was curious if there was any other reason that Hitec was used by LynxMotion. An email to their representative, and perusal of various online forums, revealed that the quality of Hitec is top-notch due to the fact that it includes Karbanite gears, one piece printed circuit board, and a ball bearing. Since the group understood that an extensive build and testing phase would exist, it was essential that high quality servomotors be purchased. Since the quality of these motors seemed fairly undisputed and was validated by the use of many hobbyists and other robot builders, the group felt confident in purchasing them. Although cheaper motors of lesser quality were available, their reduced cost did not seem to validate the threat of having one of the motors break or be bought defective. If one of the 18 motors were to break during the building or testing phase, then a brand new motor would have to be bought, which would result it not only extra expenditures for the motor but also for shipping cost. Since the project would also be on a strict time schedule, sitting around and waiting for the motor to arrive could potentially be a huge problem; especially if it was on demonstration day.

Naturally there were several vendors for these servomotors. Table 12 below encapsulates vendor information, and corresponding price.

Online Vendor	**Price**
www.servocity.com	$17.99 (each)
www.rcuniverse.com	$19.99 (each)
www.LynxMotion.com	$17.99 (each)
www.LynxMotion.com (special)	$107.94 (6 + metal horns); $17.99 (each)

Table 12: Servomotors vendors

In the end, the special offer from LynxMotion was chosen due to the fact that it included very useful metal servomotor horns for every motor. The price for every one of these horns is $3.25, and in striving to keep the mechanics of everything functioning properly through the arduous wear and tear, it was deemed that this would be a good decision. The group was already looking at purchasing metal horns separately, so this special came at a good time. Three of these packages would be bought, for a total of $323.82. Figure 15 below shows a picture of the included tapped servomotor metal horns.

Figure 15: Metal servomotor horn from LynxMotion (reprinted w/ permission)

4.2.1.2 Servomotor controller

Custom Made

The first option looked at was for the team to build a servomotor controller board themselves. However, due to the fact that there will be 18 different servomotors to control independently, as well as the fact that their control must be very closely synced together, the idea was discouraged. Instead, other methods of servomotor control were pursued; ones that could not only control multiple servomotors but control more than one of them within a fair amount of precision.

SSC-32 Controller

This servomotor controller, from LynxMotion, has the capability of controlling up to 32 servomotors at once, which was more than sufficient for our project. It has 1uS resolution along with enabling what is called ‘group’ moves, such that multiple servomotors can be controlled simultaneously. This feature will be put to very good use while designing the walking algorithm, since it would allow one to reduce some of the complexity behind controlling all 18 degrees of freedom. Figure 16 below shows the controller.

Figure 16: SSC-32 Servomotor controller from LynxMotion (reprinted w/ permission)

No other controller was found on the available market that provided both such a large amount of servo control, and the features available. Thus for $39.99, this was considered a good purchase.

4.2.2 Gait Algorithm

The gait or walking algorithm must be able to translate the 18 rotations given by the motors to the joints and legs into linear motion of the robot itself. Exactly how many sequences will be present per each step of the robot is not discussed here, but in the later sections involving design. What will be discussed here are the methods investigated for being able to derive a model and algorithm that works as specified.

4.2.2.1 Evolutionary Computation

One approach considered was that of evolutionary computation. Since deriving a working model of all 18 different motors and their rotation could turn into a difficult problem, one could simply evolve a solution using methods such as genetic algorithms or genetic programming. Although this method will be kept in mind during the course of the project, the method still involves understanding how each rotation will affect the movement of the robot as a whole. It does sidestep having to put the rotations into a meaningful sequence to effect linear motion, but it might not do so very elegantly. Thus although it can be resorted to if things do not work as planned, a more elegant solution will be pursued by analytical means.

4.2.2.2 Inverse Kinematics

The scope of inverse kinematics is determining the position of certain joints in a flexible object in order to obtain a certain movement or pose. Basically, the technique would require analytically determining what the rotation of the two hip joints and the knee joint do to the leg position of the robot. By developing an inverse kinematic model of all six legs, and understanding how to properly manipulate the joints in order to create linear motion, the drive system can be completed.

Inverse kinematics will be used to program the servomotor controller board with a proper gait algorithm. Although the technique might prove to be mathematically complex, it should not involve mathematical analysis outside of the realm of advanced trigonometry.

4.3 Vision

4.3.1 Camera/Image Sensor

There are a wide range of options of cameras for robotics applications and projects. Usually, the important requirements being image resolution and frames per second, and depending on the application, the designer chooses accordingly. Cameras for robotics can be a consumer-grade webcam with a USB or Firewire connection or a “stripped down” version of these plainly known as CMOS image sensors.

CMOS image sensors present the burgeoning CMOS technology’s greatest contribution to image science. Superbly small-scale and lightweight, low power consumption, and excellent image quality present a very attractive option. In fact these are main component in what is known as the consumer web camera, typically used for video conferencing. The drawback to using these web cameras for robotics applications are the proprietary video decoders and embedded software that is part of the integrated system. Thus, the engineer or builder must reverse-engineer these cameras to achieve what he or she wishes, if it is necessary. Since countless companies make these web cameras with little or no detailed technical specifications as to how they acquire, process, and communicate image data, it makes little sense to blindly select one for this project. The obvious reason being is connectivity to the image processor. Will it interface correctly? Can the image data be manipulated and read as desired?

These unknowns can be eliminated by choosing the CMOS image sensor. Simple, reliable, low-cost and high-quality, the CMOS image sensor is typically found in, as previously stated, digital web cameras, small low-cost digital still cameras, or cellular phones that feature digital camera capabilities. CMOS image sensors can be purchased without a lens assembly or with a lens assembly as shown below in Figures 17 and 18.

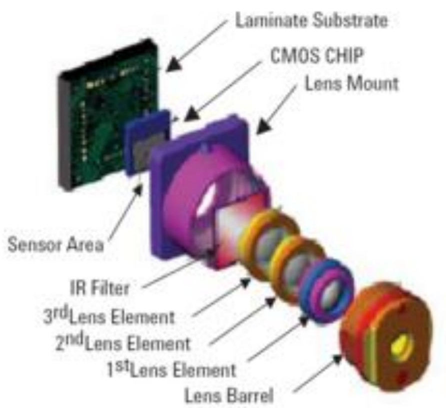


Figure 17: Basic CMOS image sensor assembly,
http://almanazir.typepad.com/almanazir/2006/10/how_a_camera_ph.html
(permission requested, pending)

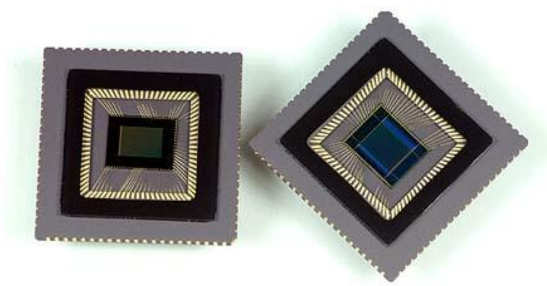


Figure 18: CMOS Image Sensor with no lens assembly.
From http://www.mobiledia.com/news/57821.html **permission requested.**

CMOS image sensors utilize CMOS (complimentary metal-oxide semiconductor) technology. They are more generally named Active Pixel Sensor Imager and contain an array of pixel sensors. Each pixel constitutes a photodetector and an active amplifier. The standard CMOS image sensor photodetector consists of a JFET photogate or pinned photodiode), a transfer gate, reset gate (M_{rst}), selection gate (M_{sel}) and source-follower readout transistor (M_{SF}), as depicted in Figure 19.

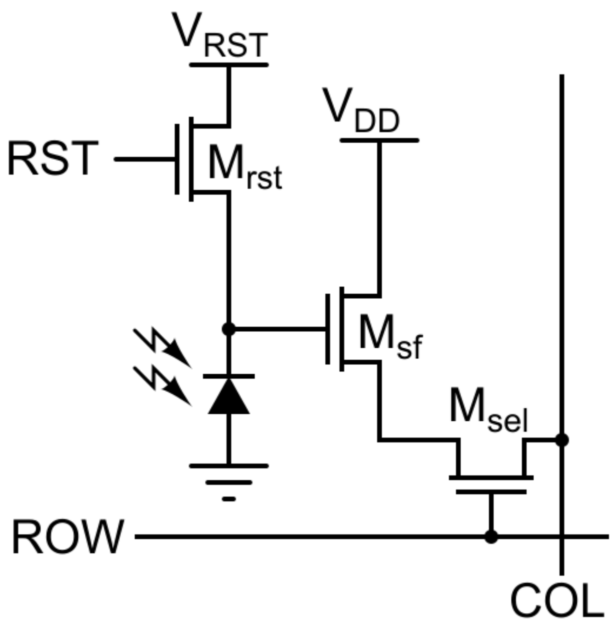


Figure 19: CMOS Image Sensor photo detector circuit (in the public domain, released by the author)

In contrast to CMOS, there exists another type of image sensor, the Charge-Coupled Device (CCD) image sensor. Instead of FET as a basic building block of pixels, the CCD image sensor uses an array of capacitors. The array is exposed to the image and a control circuit causes each capacitor to transfer its contents to its neighbor. The last capacitor in the array transfers its charge into a charge amplifier, which converts the charge into a voltage. By doing this over and over, the controlling circuit converts the entire semiconductor contents of the array to a sequence of voltages, which it samples, digitizes and stores in some form of memory as a digital image. This is older technology then CMOS and consumes much more energy than its counterpart. Also, the image acquisition is not as fast. Though for our design purposes, a CCD image sensor would have been adequate, we admired the quality-to-cost ratio of CMOS and the fact that it is the cutting edge for low-cost robotics imaging applications.

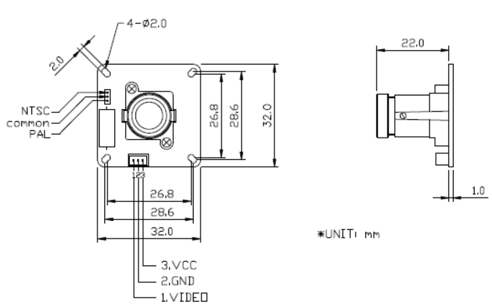


Figure : Schematic of CM-26N (from Spark Fun Electronics, permission pending)

The CMOS image sensor used in this project, the CM-26N, outputs video at 640 x 480 pixels at 30 frames per second in an analog NTSC signal. NTSC, or National Television

System Committee, is an analog video signal and most widely known as the television broadcast signal used in North America; PAL (Phase Alternating Line) being the European equivalent. The NTSC signal formation starts at red, green, and blue sensors and finally exists as a combination of two main components: a Y signal and unique color information or difference signals. Figure 20 shows a basic block diagram of the making of the NTSC signal. The Y channel or luminance (luma) channel is a composite image having brightness and containing maximum detail. It is obtained by simply adding the images seen by the red, green and blue sensors. When this is done we obtain a black and white image having brightness and detail. The Y channel is a measure of what is redundant in a signal. If we subtract this Y signal from the three (red, green, blue) image sensor signals, we can obtain what information is unique. NTSC chooses to subtract the Y signal from the red sensor to obtain an R-Y (red minus luminance) signal and a B-Y signal. If Y is the total signal, and we have the differences of R and Y and B and Y then the green signal can be determined. Thus, one signal can be thrown out and derived later if necessary.

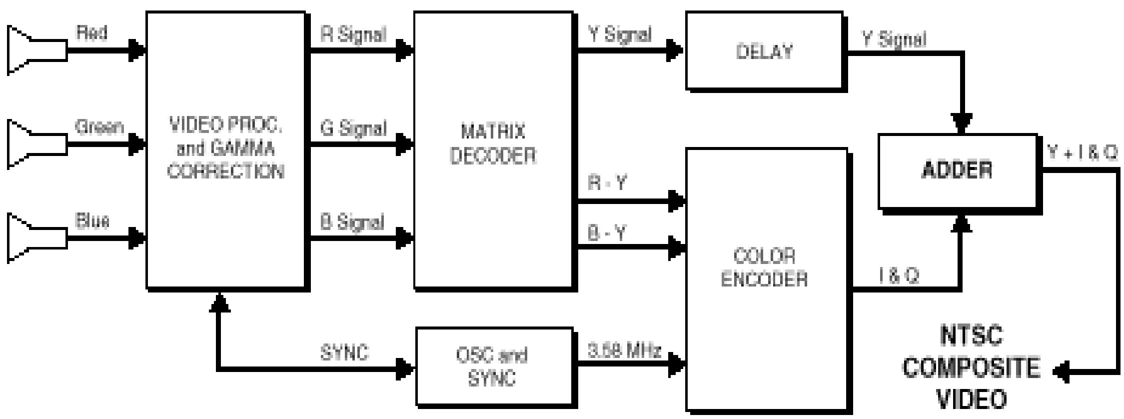


Figure 20: The Making of an NTSC signal

4.3.1.1 Color Sensor

There is another way of extracting analog image information and performing digital analysis on it. That is via a color sensor. Light shown on an object will reflect back with only the wavelengths of the colors present on the object surface (in varying intensities). Using this reflected light of certain wavelengths, we can detect what color the object is by measuring (some way) the intensity of this light. Color sensors made from simple optoelectronic devices that change their resistivity or gate voltage with incoming, incident light of specific wavelengths (colors). Color sensor circuits can be made from photoresistors or phototransistors.

Photoresistors (or light-dependent resistors, LDRs) lower their resistance with the increase of intensity of incident light. Photoresistors are characterized by a light and dark resistance. These are the resistance of the device when it is in complete or near darkness and the resistance of the device when intense light strikes it so that it is at its lowest possible resistance. These two resistance values vary from device to device, even in devices manufactured from the same process and factory.

Phototransistors increase their gate voltage as incident light intensity increases. This allows current to flow from the base to collector. It works exactly like a bipolar junction transistor except that incident photons increase the energy of electrons in the gate. Like the photoresistor, there is a light and dark characteristic. The phototransistor is characterized by a light current – the current allowed by the gate when in full lighting; and by dark current – the current allowed by the gate when in darkness. Photodiodes work in a similar fashion.

While researching parts that would be a good fit for our circuit, we had to take several things into account and make some assumptions. One is that our robot will be operating in normal room lighting conditions. The other is that the objects we wish to detect will have to be relatively close. Some research turned up two color sensors that work on the principle of turning light intensity into frequency. The more intense the incident light is, the higher the frequency of square waves is outputted. These are known as light-to-frequency generators. One really useful thing we also found in our research is that these light-to-frequency generators can be purchased with light-filters on them already allowing only red, green, or blue light to reach the optoelectronic devices. We found two major and widely-used light-to-frequency generators commercially available: 1) the TAOS TSL230RD and 2) the Avago ADJD-S371-QR999.

4.3.1.1.1 TAOS TSL230RD Color Sensor

The TAOS TSL230RD is a light-to-frequency generator integrated circuit. It is composed of an array of 64 photodiodes: 16 with red-color light filters, 16 with green-color light filters, 16 with blue-color light filters, and 16 with clear filters. The specific filters are selectable with two digital inputs. The truth table for the color selection is shown below.

Filter	S3	S2
Red	L	L
Green	H	H
Blue	H	L
Clear	L	H

Table 12: Truth Table for Color Selection on the TAOS TSL230RD

The square pulses are outputted from the IC at a frequency linear to the intensity of light.

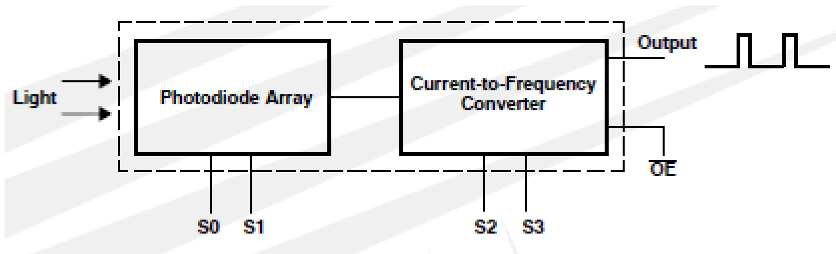


Figure 21: Functional Block Diagram of the TAOS TSL230RD

The TAOS TSL230RD is a very low-power device and will our power as well as our weight specifications as indicated in the table below:

CHARACTERISTIC	UNIT
Supply voltage	5V
High-level input voltage	5 V
Low-level input voltage	0-0.8 V
High-level output voltage	4.5 V
Low-level output voltage	0.4 V
High-level input current	5 microamperes
Low-level input current	5 micro amperes

Table 13: Some Electrical Characteristics of the TAOS TSL230RD

The TAOS TSL230RD is available on a printed circuit board with pin connectors for its input/outputs and a lens assembly from Spark Fun Electronics for $44.95. The lens assembly would be very helpful in keeping out ambient light and infrared light. Also, the board it comes on would be very easy to attach to another board and also for testing purposes. It also outputs data continuously in the form of square pulses so there is no need to program timing considerations into the image processing unit.

4.3.1.1.2 Avago ADJD-S371-QR999

The color sensor by Avago is also a light-to-frequency generator similar to the TAOS color sensor but with a bit more options and control. The Avago light-to-frequency generator allows the user to select gain levels of output signals for individual color channels as well as put the device in a sleep or inactive mode. The Avago device also allows for an external oscillator/clock. This would be useful if we were using an external memory device, as would be needed would the Avago device since it outputs a large amount of data serially.

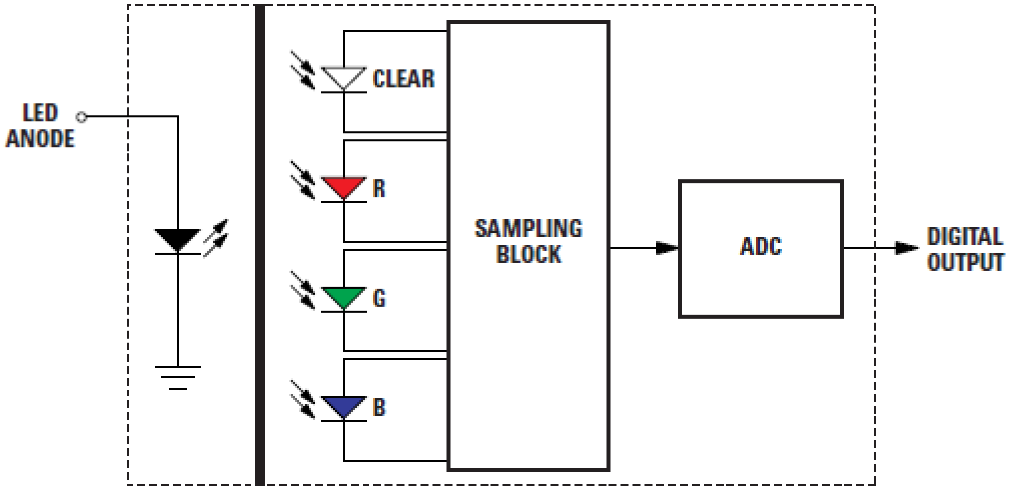


Figure 22: Functional Block Diagram of the Avago ADJD-S371-QR999

The Avago is also very, very small which is great for space considerations, but too small may make it too difficult for testing and building. The dimensions of the Avago unit are: 3.9 x 4.5 x 1.8 mm.

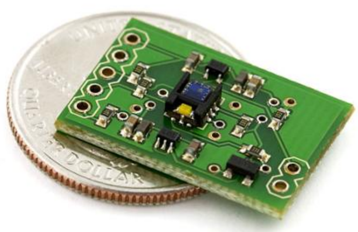


Figure 23: Avago ADJD-S371-QR999

The Avago device has a timed data transfer so the image processor (whether it be the microcontroller or FPGA) must be programmed to accept data at the start sequence.

The master initiates data transfer after a START condition. Data is transferred in bits with the master generating one clock pulse for each bit sent. For a data bit to be valid, the SDA data line must be stable during the HIGH period of the SCL clock line. Only during the LOW period of the SCL clock line can the SDA data line change state to either HIGH or LOW. A complete data transfer is 8-bits long or 1-byte. Each byte is sent most significant bit (MSB) first followed by an acknowledge bit or not acknowledge bit. Each data transfer can send an unlimited number of bytes (depending on the data format).

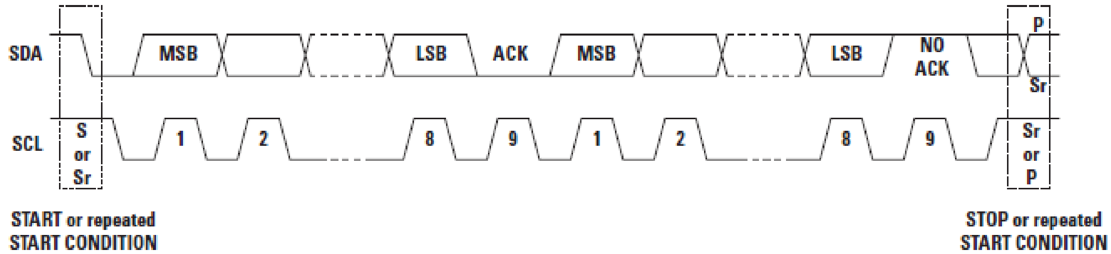


Figure 24: Avago ADJD-S371-QR999 data stream

The Avago ADJD-S371-QR999 color sensor is available from Spark Fun Electronics for $19.95 already pre-soldered onto a printed circuit board. Recapping of the prices and vitals of our image/color sensor units is shown in table.

Sensor Unit	Difficulty of Interfacing to Image processor	Timing Considerations Needed?	Price	Size	Power Consumption
NTSCCamera (CMOS 26N)	High	Yes	$29.95	2 in x 2in x 1 in	10 mW
Photoresistor Circuit	Low	No	~$20.00	4 in x 4in	Unknown
Phototransistor Circuit	Low	No	~$20.00	4 in x 4 in	Unknown
TAOS TSL230RD	Low	No	$44.95	1 in x 1in	5 mW
Avago ADJD-S371-QR999	High	Yes	$19.95	2 in x 2in	5 microamp/5 microW

Table 14: Comparison of Color Sensor Solutions

4.3.2 Video Decoder

The video decoder would only be needed for the camera. This section describes that component. If a color sensor circuit is used there would be no need of course for the video decoder.

Once an image is acquired in the form of an analog NTSC signal, in order for any logical and computational operations to be done on it, it must be in a digital format. A video decoder does just this. It is a subset of analog-to-digital converters that allows an analog signal to ultimately be in a format that can be read, analyzed, and/or manipulated by a computer. There are a variety of digital formats that a video decoder can output. Figure 21 shows a basic NTSC video decoder block diagram.

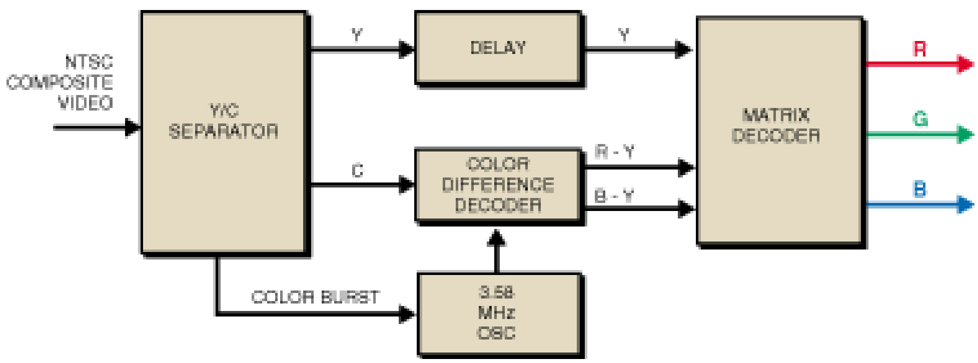


Figure 25: Basic NTSC decoding system

One of particular interest is the ITU BT.656 video signal format. It is a simple protocol for streaming data digitally from an uncompressed NTSC analog signal. The BT.656 signal will allow simple read-in to memory and analysis by our logic system. Besides raw video data, BT.656 contains status data to denote the end of active video (EAV) data and the start of active video (SAV) data. Both EAV and SAV codes are comprised of a sequence of four bytes. The first three bytes in the sequence constitute a fixed preamble: FFh – 00h – 00h. The fourth byte contains information about the field being transmitted (Field 1 or Field 2 in an interlaced video signal), the state of field blanking (Vertical) and the state of line blanking (Horizontal). These codes are embedded within the BT.656 video data stream, thereby eliminating the need for additional timing signals (HSYNC, VSYNC, BLANK) to be included as part of the interface. Figure 22 shows the standard BT.656 data stream.

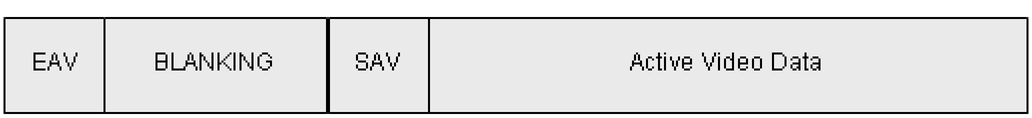


Figure 26: Standard ITU-R BT.656 data stream format for a single horizontal line of digital video data.

Active video data consists of 720 active samples of Y (luminance) and 360 samples each of C_b (blue chrominance) and C_r (red chrominance). After an SAV has been sent, active video data is then sent in the following format:

$Cb_0Y_0Cr_0Y_1Cb_1Y_2Cr_1Y_3$ .... $Cb_{359}Y_{718}Cr_{359}Y_{719}$

Each set corresponding to one pixel.

Hence, this is what will be outputted to the logic part of the imaging system, the FPGA.

4.3.3 FPGA and Image Processing Algorithms

In this section, algorithms and processes will be discussed for two kinds of implementations: the camera implementation and the light-to-frequency generator implementation. There would be no need for an image processing unit (such as a microcontroller or FPGA) for the photoresistor or phototransistor circuit.

Once the image is captured and digitized, it is sent to the FPGA where it will be processed. An FPGA represents programmable logic gates and more or less has the same role as a microprocessor or more specifically, a digital signal processor. Digital signal processors (DSP) are highly specialized components with many of them today containing many high-end embedded applications on a single integrated circuit chip. Digital signal processors store an image in memory, fetch it, do mathematical operations and then store it again in memory. Needless to say, this becomes very time-consuming, especially for larger resolutions. Even though processors can achieve frequencies in the Giga Hertz range, it is the inherent nature of image processing calculations that cause DSPs to be relatively slow, creating a bottleneck.

The mathematics of image processing makes FPGAs very attractive for algorithm realization and implementation. For instance, image convolution, a window of pixels is treated with a mask where individual locations in the window are ‘weighted’ according to a set of previously defined coefficients. For each position of the window all pixels are multiplied against their respective coefficients. The final result is then scaled to produce a single output pixel for the centre location of the window. This requires several multiplications per pixel, of which can be done in parallel. In a DSP the number of parallel multiplications would be limiting, as to not even mention the length or number of clock signals. An FPGA, since it is programmed by the user, can implement as many multipliers as physically allowed.

An FPGA has a flexible over DSPs because they are programmed by the user and do not have permanent functions defined by a manufacturer. FPGA logic is described using Verilog language or VHDL (VHSIC Hardware Description Language). Both of which have similarities to C language. However, it must be noted that writing software in C and describing hardware in VHDL or Verilog are two different types of programming altogether.

Since the FPGA will be where the robotic system "thinks" and tells the user what is "seen" or which direction to maneuver, it can be called the brain of our robot. There are many advanced and expensive FPGA development boards available with the discriminating factors being: the number of logic gates and the on-board memory (RAM or ROM). Logic gate numbers soar to the millions for some boards. Most FPGA boards run on 5 V to 12 V DC power or 5 V USB power and do not consume a substantial amount of power.

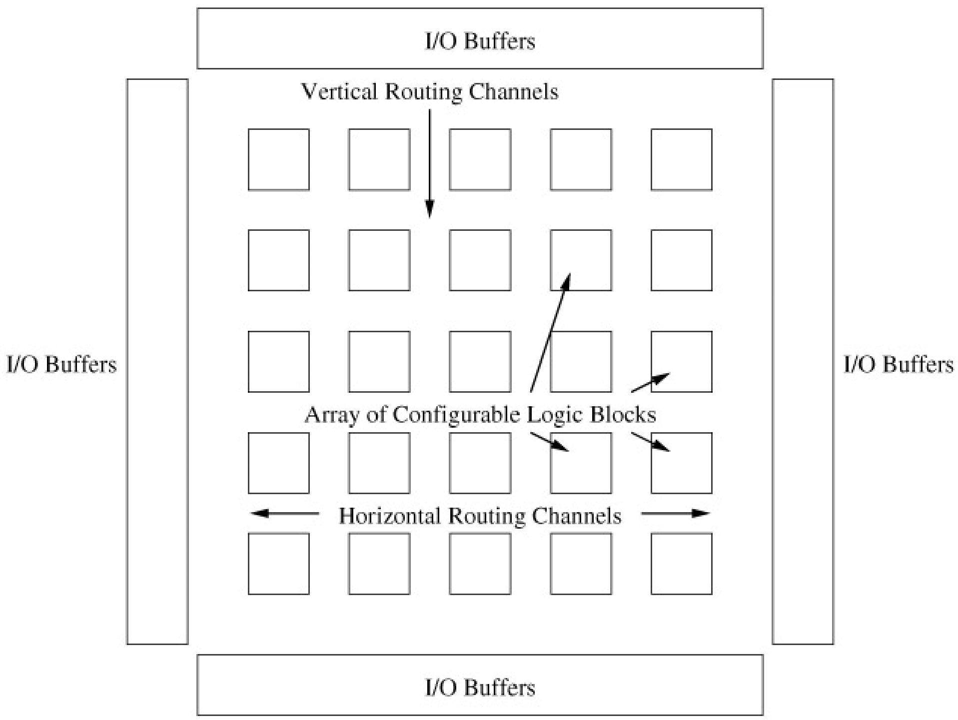


Figure 27: Simplified XILINX FPGA architecture Diagram (from Roger Meier, permission pending)

Since the specifications require that the robot be able to detect a shape and a color, the FPGA must be programmed to do so. Several algorithms exist for doing so. Using pseudo-code examples, the general algorithm can be shown.

4.3.3.1 Microcontroller and Image Processing Algorithms

Microcontrollers are small, light-weight in terms of computing power, and are relatively simple design interfaces compared to FPGAs. If we were to use a microcontroller for our image processing we would be able to program in the C language. There are a myriad of microcontrollers available.

For the camera, we would need a microcontroller that could handle the 1 frame per second of 8-bit image data and process it before the next frame was captured. One frame is equivalent to 640 rows of pixels by 480 rows of pixels, or 307200 pixels. There are 8 bits per pixel, so 2457600 bits of data streaming is what the microcontroller would have to handle or 2400 bytes/second and then do the processing. Most microcontrollers at a speed of 8 MHz could handle this.

The only real limiting factor would be memory space to hold an image in a "buffer." The buffer would be useful for comparing the current frame with the last for instance.

	ATMega644	PICAXE18X
Clock Speed	20 MHz	4 MHz
Memory Size	64 kB of program memory, 4 kB SRAM, 2 kB EEPROM	About 1000 lines of program code.
Power Consumption	240 micro-Amp @ 1 MHz (3 V)	150 micro-Amp @ 1 MHz (2-4.5 V)

Table 16: Comparison of Considered Microcontrollers

We considered the ATMega644 and the PICAXE family of microcontrollers. The PICAXE18X, in particular, was a good balance of features and usability in the PICAXE line.

These microcontrollers are readily available and have excellent documentation. They are also relatively inexpensive (usually less than $20), small in size and very low in power consumption.

4.4 Wireless Communication

The Controller

The original plan was to use a joystick, the kind used in playing flights simulators. But all of the joysticks on today's use USB connectors and ports. To extract and interpret the data for the central unit of the robot, a USB converter circuit would be necessary; this would turn out to be an unnecessarily tedious process and require some extra months. So to avoid such situation, the Playstation 2 DualShock 2 controller was chosen for the project. Using some documentation on the Playstation 1 controller found on the Internet, the data can easily be extracted directly from the 9-pin connector and interpreted with a microcontroller circuitry, with which the controller will interact as it would with a Playstation 2 console.

It would be preferable to use a Playstation 1 controller, as to follow the documentation by the letter, but only Playstation 2 DualShock 2 was on the market. After a little more research though, it has been noticed that the Playstation 2 is not even close to be barely different from the Playstation 1 controller.

The Microcontroller

Many microcontrollers on the market right now can be used to interpret the data from the controller for the RF module and the brain of the robot. So far, it has been discovered that two of them have been used to handle the task of interacting with the controller: Motorola's MC68HC11 MCU and Atmel's AT90S2313. The most recent documentation found involves the latter microcontroller was more recent, so an Interpreter built around the AVR AT90S2313 received more consideration, especially due to the fact that Atmel's microcontrollers are more commonly used in gaming electronics. But after some research, the Atom Pro 28-M was discovered, and, due to its extreme simplicity and to some of its convenient features, in comparison to the microcontrollers mentioned previously, was chosen for this project definitively.

The Transceiver

A two-way traffic of data of significant rate between the controller and the robot is needed for good control, so two RF transceivers were necessary. First, Atmel's products were considered, due to the high frequency of their products being used in the gaming field. The ATR2406 stood out, as it operates at 2.4 GHz in the ISM band and wireless gaming is one of its main applications. The only drawback though is that a circuit has to be built around it with care due to the high operating frequency. Moreover, an antenna is needed and a feeding line is necessary. The feeding line has to be designed and carefully drawn on the board with the right dimensions so that it matches the impedance of the antenna to that of the output pin of the ATR2406 chip. Development boards built around the ATR2406 are available though. They are pre-built circuit boards containing the chip equipped with a jack or a connector where to connect a whip antenna is also provided.

But its cost turns out to be a hundred times that of the chip itself. So another option was pursued.

Google was used in tracking down more compatible transceivers. It led to Texas Instruments' Website where the CC2500 was considered. But similar problems arose, as only the chip is provided. Similarly, development boards were available but were also too pricey for the budget. Then Digikey results' filter was used to find more appropriate transceivers for the project. Out of all the transceivers Digikey sells, the parameters were set: "Gaming Devices" for Applications, and "Integral Antenna". It spat out transceiver models from AeroComm like the AC4486 and the AC4424. The AC4486 operates at 868 MHz and the AC4424 operates at 2.4 GHz. Both of these transceivers come with either an integral antenna or an MMCX connector for antenna, without being included on development boards. Of course it comes out to be more expensive than the previous chips, but much cheaper than the development boards.

Linx Technologies' RF modules were next considered. Their price and their 'plug-and-play" feature caught our attention. The only downside was the fact that they were mostly surface-mount devices, PCB boards were necessary to even test them and antennas were to be installed. PCBs were designed and built but their mysterious less-than-sastifying and inconsistent performance forced us to look for other solutions

Observing another group using Xbee modules caused us to adopt them as well. Though not as straightforward as Linx Technologies RF modules, an antenna is already installed, no PCB board was required for testing and they were more definitely more reliable and consistent in performance than the Linx Technologies RF modules.

4.5 Power

Types of Batteries

In designing the robot's power supply, one of the most important things to take into consideration is the battery source amp-hour rating, because it tells the maximum power the battery can supply and for how long it can supply that power. This also decide how long the robot can run for. Therefore rechargeable batteries are 62sed to power the robot; which will save the cost of buying a lot battery.

Rechargeable Batteries

Since we want to spend the least amount of money possible on this project, using rechargeable battery make the most economic sense. The cost of buying the rechargeable battery and its charger may seem to be a lot at first. But considering the fact that a lot of tests were done on the robot, and during these tests, the robot burned power; which resulted in battery drainage and eventually the battery will not be able to provide enough voltage to satisfy the robot's power requirement. For example since the robot has a battery source voltage that is 12V, we can buy an eight pack of 1.5V AA batteries, that will cost us around five dollars at a convenient store, since these batteries are not rechargeable, we would have to keep buying these packs as we are performing tests on the robot, which will end up costing us more money on the long run. But buying a rechargeable battery and a charger, which may cost around $50, saves us money in the long run, because the battery can be recharge whenever it cannot supply the desired voltage for the robot to function properly.

Lithium-ion Batteries

Lithium-ion batteries are a type of rechargeable battery commonly used in consumer electronics. They are currently one of the most popular types of battery for portable electronics, with one of the best energy-to-weight ratios, no memory effect, and a slow loss of charge when not in use. They were first proposed in the 1960's but first commercially released by Sony in 1991. Although originally intended for consumer electronics, Lithium ion batteries are growing in popularity with the defense and aerospace industries because of their high energy density. One of the latest uses is in hybrid electric cars and eventually electric vehicles, as commodity cells.

The advantage of the Lithium-ion batteries includes its ability to be formed into a wide variety of shapes and sizes so as to efficiently fill available space in the devices they power. Li-ion batteries are lighter than other batteries; which is good for our robot, because we want it to be relatively light, and the energy is stored in these batteries through the movement of lithium ions. A key advantage of using Li-ion chemistry is the high open circuit voltage that can be obtained in comparison to aqueous batteries, such as lead acid, nickel metal hydride and nickel cadmium. The disadvantages of the Li-ion battery is that its life span is dependent upon aging from time of manufacturing regardless of whether it was charged, and not just on the number of charge/discharge cycles. Also, as batteries age, their internal resistance rises. This causes the voltage at the terminals to drop under load, reducing the maximum current that can be drawn from them. Eventually they reach a point at which the battery can no longer operate the equipment it is installed in for an adequate period. Li-ion batteries are generally more expensive but not as durable as nickel metal hydride or nickel-cadmium designs and can be extremely dangerous if mistreated. Li-ion chemistry is not safe as such, and a Li-ion cell requires several mandatory safety devices to be built in before it can be considered safe for use outside of a laboratory. Below table 14 shows the different types of chemical compositions of lithium-ion batteries.

Material	**Average Voltage**	**Gravimetric Capacity**
$LiCoO_2$	3.7V	140mAh/g
$LiMnO_2$	4.0V	100mAh/g
$LiFePO_4$	3.3V	170mAh/g
Li_2FePO_4F	3.6V	115mAh/g

Table 14: Chemical compositions of Li-ion batteries

Storing a lithium-ion battery at the correct temperature makes all the difference in maintaining its storage capacity. It is significantly advantageous to avoid storing the battery at 100% charge, because a lithium-ion battery stored at 40% charge will last longer, particularly at high temperatures. However, if the battery is stored at a very low charge, there is a risk of allowing the charge to drop below the battery's low voltage-threshold, resulting in an unrecoverable dead battery. Once the charge has dropped below this level, recharging the battery can be dangerous. To prolong the shelf life of a lithium-ion battery, it is recommended to charge or discharge it to 40% and place it in a refrigerator. Some lithium-ion batteries will provide more energy when brought to room temperature. Below table 15 shows the amount of permanent capacity loss that will occur after storage at given charge level and temperature.

Storage Temperature	40% Charge	100% Charge
0ºC (32ºF)	2% loss after one year	6% loss after one year
25ºC (77ºF)	4% loss after one year	20% loss after one year
40ºC (104ºF)	15% loss after one year	35% loss after one year
60ºC (140ºF)	25% loss after one year	40% loss after 3 months

Table 15: Permanent capacity loss versus storage conditions

Nickel metal hydride Batteries

NiMH battery technology was invented by Stanford Ovshinsky. The first consumer grade NiMH batteries began to appear in the mid-1980s. In 1994, his company, ECD Ovonics partnered with General Motors to develop high performance NiMH batteries for the GM Ev1 electric car. Currently, NiMH technology is licensed by Cobasys, which is a joint venture between Chevron Corporation and ECD Ovonics. NiMH batteries and chargers are readily available in retail stores in common sizes: AAA, AA, C and D. They are not expensive, and the voltage and performance is similar to standard alkaline batteries in those sizes; they can be substituted for most purposes. The ability to recharge hundreds of times can save a lot of money and resources.

The characteristics of Nickel metal Hydride batteries include its low internal resistance, which is advantageous for high current drain applications. This low internal resistance allows the NiMH cells to deliver a near-constant voltage until they are almost completely discharged. NiMH also has a low self discharge rate, which is ideal for very low power applications. Nickel Metal Hydride (NiMH) has a nominal output voltage of 1.25 Volts and can provide 1300 to 2500 mAh in 'AA' size. It provides a lower power to weight ratio than other technologies, but provides a medium life cycle of 300 to 500 times. It has low toxicity, but should be recycled.

Nickel Cadmium

The nickel-cadmium battery is a popular type of rechargeable battery using nickel oxide hydroxide and metallic cadmium as electrodes. The principal advantages of NiCad over other rechargeable types is lower weight for a given quantity of stored energy, good charging efficiency, small variation in terminal voltage during discharge, low internal resistance, and non-critical charging conditions. They can be used in place of regular batteries in most applications and they are more difficult to damage than other batteries, tolerating deep discharge for long periods. The disadvantages of the NiCad rechargeable batteries include its higher cost. They require extra labor to manufacture, and thus, are typically more costly than other battery types. They also exhibit a very marked negative temperature coefficient, i.e., as the cell temperature rose, the internal resistance fell. NiCad batteries also exhibit thermal runaway, where the charging current would continue to rise until the battery destroys itself. Due to all these disadvantages, the nickel-cadmium technology is falling out of favor.

Nickel Cadmium has a high life cycle recharge times of 1500 and has a moderate recharge overload tolerance. Its output is 1.25V just like NiMH and has a lower self-discharge rate of 30% per month. It is a mature technology that has been in use since the 1950's, but unfortunately it's toxic and has a low energy to weight ratio, second only to lead-acid.

Alkaline Batteries

Rechargeable alkaline battery is a type of alkaline battery that is rechargeable. This rechargeable alkaline technology was developed by Battery Technologies Inc in Canada and licensed to Rayovac, Pure Energy and Grand cell. The shapes include AAA, AA, C, D and snap-on 9-volt batteries. Unless you're looking for an inexpensive rechargeable battery that is ready to use right out of the package (without recharging), we suggest you avoid them. This type of battery is best suited for low-drain devices such as remote controls, or for devices that are used periodically such as flashlights. Chargers designed for NiCad and NiMH batteries are not suitable for charging rechargeable alkaline cells.

The advantages of rechargeable alkaline batteries include its relatively cheap cost and a high charge-capacity (Typically 2 Ah for a AA and over 6 Ah for a D cell). Rechargeable alkaline batteries also hold their charge for years, unlike the NiCad and NiMH batteries. The disadvantages of the rechargeable alkaline batteries include its maintenance of a full chargeable capacity for hundreds of cycles, only after the battery has been less than 25% discharged, at about 1.42V. Also, after a 'Deep Discharge' they can be brought to their original high-capacity charge only after many frustrating charge-discharge cycles; thus, their 'Theoretical' high-capacity is available only for 'Emergency cases'. Unlike NiMH rechargeable batteries, alkaline batteries are normally not sold with a nominal capacity. Alkaline has a high internal resistance and a high thermal coefficient of resistivity – the faster you drain an alkaline, the higher percentage of the load it dissipates as heat. Therefore, the capacity of an alkaline battery is strongly dependent on the load, even at moderate loads. A AA-sized alkaline battery might have an effective capacity of 3000 mAh at low power, but at a load of 1000 mA, which is common for digital cameras, the capacity could be as little as 700 mAh. Below table 16 shows a comparison of the different types of rechargeable batteries.

	Nickel Metal Hydride (NiMH)	Nickel Cadmium (NiCad)	Lithium Ion (Li-ion)	Alkaline
Voltage	1.25V	1.25V	1.75V	1.50V
Capacity	High	Low	High	High initially, but decreases with use
Capacity in mAH (AA type)	1300 – 2500	600 -1000	21000	2000 initially
Capacity in mAH (D type)	2200-11000	1800-4500	Not available in this size	8000 initially
Performance in high drain devices	Good	Good	Good	Poor
Recharge Ability	Excellent	Excellent	Good	Good
Recharge cycle	100's time	100's time	100's time	50 to 500 Accucell
Special Disposal needs	No	Yes	Yes	No
Self Discharge	Fast 30%/Month	Moderate 10%/Month	Very Slow	Slow
Memory Effect	No	Yes	No	No
Price for 2 AA's	$5.00	$5.49 Standard $6.99 Hi Cap	$6.99	$3.60

Table 16: Comparison of the different rechargeable battery types

Wall outlet

Since batteries are never at a constant voltage, and with the fact that our robot may consume a good amount of power, for example, we are going to use 18 servo motors to move the legs of the robot and each servo need 4.6V to 6V and can draw between 150 mAh to180 mAh so their power consumption alone would be between .69W to 1.08W each. Therefore as one of my options to power the robot as a last resort, just in case the battery we may need may be too expensive for our budget, connecting the robot to an external voltage source such as a wall outlet is included. Or instead of using the same power source for both our control circuitry and our servos and motors, the robot may use a battery for the control circuitry and use the wall out outlet to power the servomotors of the robot which would cost us less money.

Voltage Regulators

Given that the different components of the robot requires different amount of voltages and to avoid voltage fluctuation, voltage regulators are used to output a constant amount of voltage to the different subsystems of the robot. There are two major types of voltage regulators, linear and switching voltage regulators.

Linear Voltage Regulator

Some of the components of the robot supply circuit are designed to operate with a constant voltage supply. The supply voltage needed for their operation sometimes differ from the source supply voltage. A voltage regulator provides this constant DC output voltage for the circuit and contains circuitry that will continuously hold this output voltage at the designed value regardless of changes in the load current or input voltage. Therefore, a linear regulator will supply a constant output voltage to its subsystems as long as the input voltage is within the operating range of the linear regulator. Linear regulators operate by using a voltage controlled current source to force a fixed voltage to appear at the regulator output terminal. This fixed voltage is supplied to our subsystems and remains constant. The control circuitry basically monitors or senses the output voltage, and adjusts the current source as required to maintain the output voltage at the desired value. The limit of the current source determines the maximum load current the regulator can source and still maintain regulation. As can be seen the basic linear regulator functional diagram, the output voltage is controlled using a feedback loop, which requires some type of compensation to assure loop stability. The loop is important to ensure the continuous output of a steady voltage by the regulator.

Some of the linear regulators have built in compensation and therefore maintain their stability without any external components. However, other linear regulators do require a capacitance connected from the output lead to ground to assure regulator stability. Linear regulators therefore work by taking the difference between the input and the output voltages and burning it up as waste heat. Therefore, using the linear regulator also drives the need for a heat sink to conduct away all the unwanted heat from the circuit. The amount of heat produced depends on the difference between the input and the output voltages i.e. the voltage difference. In our design, the difference in voltage levels is quite small and the effects of heat will be negligible. There are three basic types of linear regulator designs: The Standard Linear Regulator; The Low Dropout or LDO Linear Regulator and the Quasi LDO Linear Regulator. The most significant difference between these Linear Regulator types is the “Dropout Voltage.” The dropout voltage is defined as the minimum voltage required across the linear regulator to maintain output voltage regulation. Therefore, the lower the regulator dropout voltage, the more efficient the regulator will operate. The regulator that operates with the smallest voltage across it dissipates the least internal power and has the highest efficiency. The Low dropout regulator requires the least voltage across it for continued operation while the standard regulator requires the most. Another significant difference between these linear regulator types is the “Ground Pin Current.” The ground pin current is the current required by the regulator when driving rated load currents. This current is however undesirable because it must be supplied by the source but does not power the load. It is therefore referred to as "wasted current" The standard regulator has the lowest ground pin current while the low dropout regulator has the highest. An analysis of the different types of linear regulators based on their characteristics and our chosen application will reveal that the Low dropout (LDO) Linear regulator will be most suited for the AREN power supply system. The LDO has very low dropout voltage which will ensure a steady power supply to our subsystems and the current draw from our systems is not nearly large enough make the ground pin current a concern. Figure 24 below shows a basic diagram of a linear regulator.

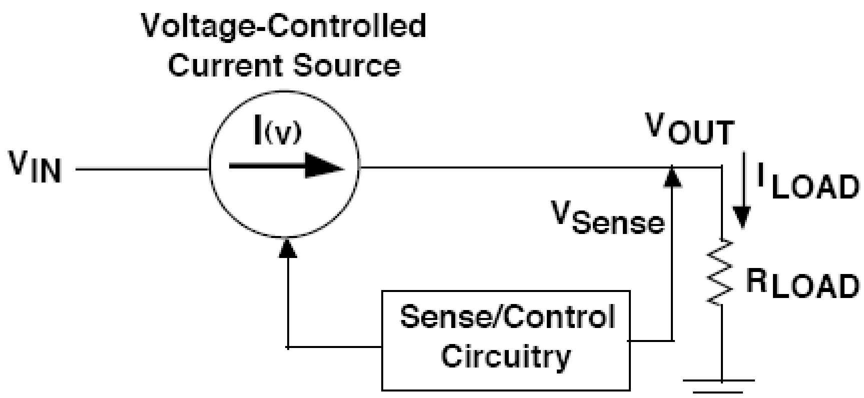


Figure 24: Linear Voltage Regulator (Reprint permission granted through site www.national.com)

Linear voltage regulators can waste a lot of power by heat dissipation, since the microcontroller and the sensors do not draw a lot of current; using a voltage regulator to regulate their required voltage is not going to waste a lot of power. But for the servos and motors the power wasted could be a lot, even though the difference in source voltage and required voltage maybe small, because they draw more current than the microcontroller and the sensor. So to minimize the waste of power, instead of using a voltage regulator, we may use a switching voltage regulator to regulate the required voltage for the servos and motors.

Switching Voltage Regulator

A switching regulator basically works by taking small chunks of energy, bit by bit, from the input voltage source, and moving them to the output. The switching regulator accomplishes this with the use of an electrical switch and a controller. This electrical switch and controller combination regulates the rate at which energy is transferred to the output. The transfer of energy from the input to the output is accompanied by losses during the transfer process. These losses involved in moving the energy around are relatively small with the switching regulators. In effect, the efficiency of these regulators is high, in the order of 85% to 90%. The regulator efficiency is normally less dependent on input voltage; therefore, they can power useful loads from higher voltage sources.

The switching regulators are rapidly gaining popularity and usage due to their high power conversion efficiency and increased design flexibility. In using the switching regulator, multiple output voltages of different polarities can be generated from a single input voltage. This enables a single input voltage fed into the switching regulator to output various output voltages. These outputs can be used to power several subsystems and sub-circuits, which increase the flexibility and efficiency of the switching regulator. Below figure 25 shows a basic diagram of a switching voltage regulator.

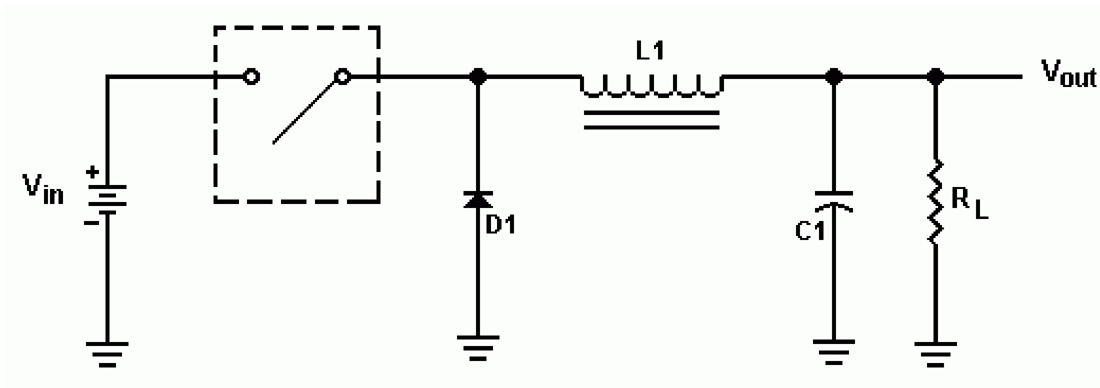


Figure 25: Switching Voltage Regulator (Reprint permission granted through site www.national.com)

There are many switching regulators; the most commonly used ones are:

1. The Flyback Converter
2. The Boost Converter
3. The Buck Converter
4. The Buck Boost Converter

Flyback Converter

The Flyback converter is a very unique converter with diverse utilities. It produces an output voltage that is less than or greater than the input voltage. This converter is capable of generating an output voltage whose voltage level is less than the input level or it can produce an output voltage whose voltage level is greater than the input level. Therefore, this converter can be used in applications that desire either a voltage step up or a voltage step down. It is also capable of producing multiple outputs. Consequently, its output voltage can be supplied to several subsystems simultaneously. The fly-back regulator is therefore the most versatile of all the regulators. The lower diagrams show the current flow paths for each of the modes when the switch is ON and when the switch is OFF. The most important feature of the fly-back regulator is the transformer phasing, as shown in figure 26 below by the dots on the primary and secondary windings.

When the switch is turned ON i.e. Mode 1, the input voltage is forced across the transformer primary which causes an increasing flow of current through it. Here, the diode will be turned OFF, preventing the flow of current in the secondary windings. Therefore, the load current is supplied by the output capacitor alone, thus, discharging the capacitor. When the switch is turned OFF i.e. Mode 2, the decreasing current flow in the primary causes the voltage at the dot end to swing positive. At the same time, the primary voltage is reflected to the secondary with the same polarity. Here, the diode is turned ON, allowing the current to flow into both the load and the output capacitor. Thus, the capacitor is charging during the OFF time. Below is a diagram of a flyback converter.

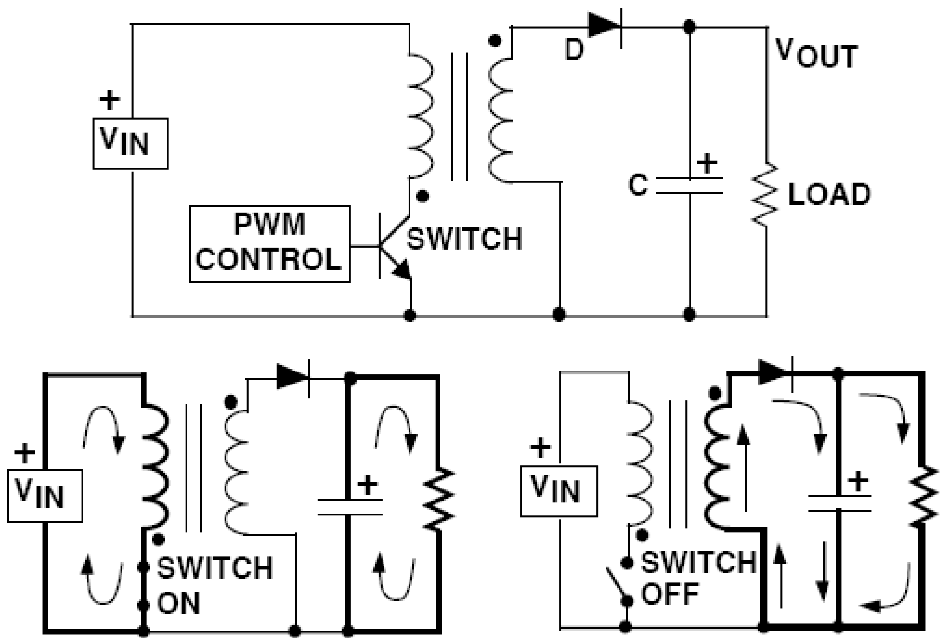


Figure 26: Diagram of a Flyback Converter (Reprint permission granted through site www.national.com)

Boost Converter

The Boost converter is basically used to provide an output voltage level that is higher than the input voltage level. It can therefore be referred to as a Step up Converter. The voltage levels here must also be of the same polarity. Boost converters are essentially used in systems that convert smaller voltage levels to larger voltage levels of the same polarity for use in desired subsystems. The boost converter operates very efficiently with very little power loss. It also utilizes a transistor as a switch that forms two different operating modes for the converter. A functional diagram of the boost converter, showing its two modes of operation is as shown in figure 27 below:

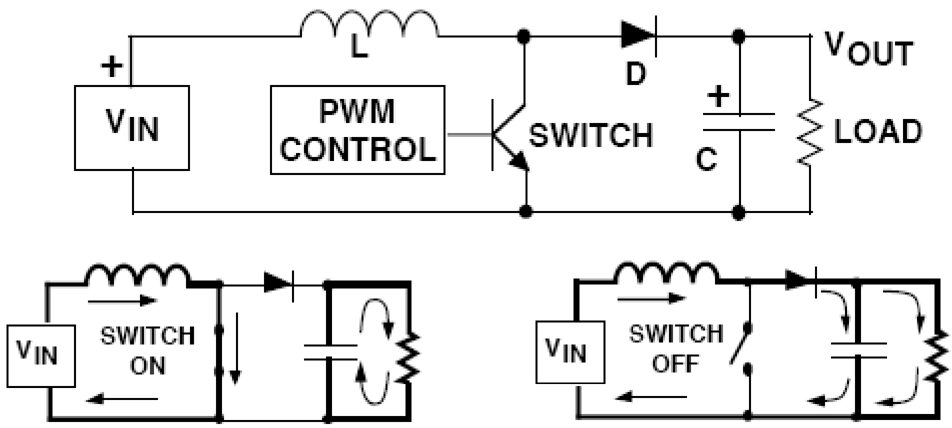


Figure 27: Diagrams of a Boost Converter (Reprint permission granted through site www.national.com)

The lower diagrams show the current flow paths for each of the modes when the switch is ON and when the switch is OFF. When the switch is turned ON i.e. Mode 1, the input voltage is forced across the inductor which causes the current through it to increase. With the Diode off, the load circuit is isolated from the source circuit in mode 1. The capacitor is therefore not charging in this mode. When the switch is turned OFF i.e. Mode 2, the decreasing inductor current forces the diode to be forward biased i.e. turned ON, allowing the capacitor to charge up to a voltage that is higher than the input voltage. Hence, the voltage supplied to the load will be higher than the source voltage.

Buck Converter

The Buck converter is basically used to reduce a DC voltage level to a lower DC voltage level. It can therefore be referred to as a Step down Converter. The voltage levels must also be of the same polarity. Buck converters are essentially used in systems that use distributed power rails such as 24 volts to 48 volts which must be locally converted to smaller voltage levels such as 15 volts, 12 volts and even 5 volts with very little power loss. The buck converter operates by using a transistor as a switch that alternately connects and disconnects the input voltage to an inductor. This alternate switching that connects and disconnects the input voltage to the inductor forms two operating modes for the converter.

The lower diagrams show the current flow paths for each of the modes when the switch is ON and when the switch is OFF. When the switch is turned ON i.e. Mode 1, the input voltage is connected to the inductor as the switch completes the circuit. The difference between the input and output voltages is then forced across the inductor, causing current through the inductor to increase, hence charging the capacitor. When the switch is turned OFF i.e. Mode 2, the input voltage applied to the inductor is removed. However, the voltage across the inductor will adjust to hold the current constant since the current in an inductor cannot change instantly. During this time, the capacitor discharges into the load, contributing to the total current being supplied to the load. Figure 28 below shows the diagrams of a buck converter.

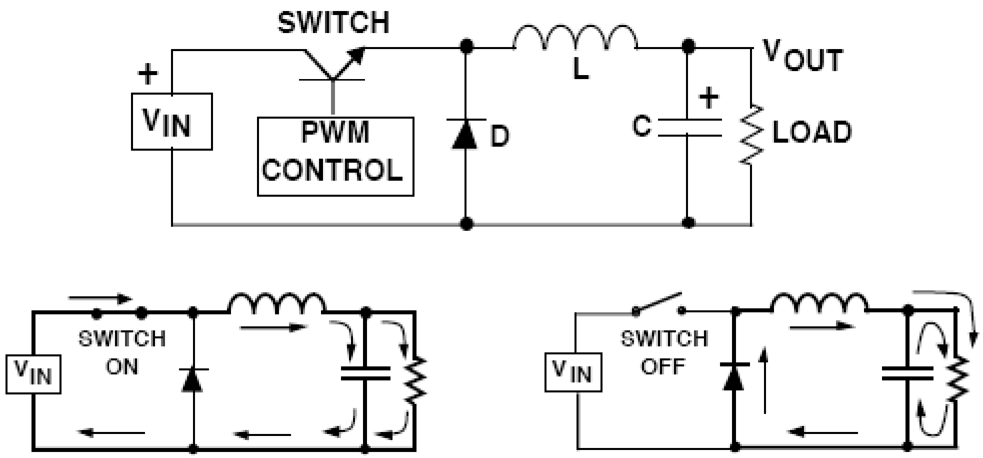


Figure 28: Diagram of a Buck Converter (Reprint permission granted through site www.national.com)

Buck Boost Converter

The Buck Boost converter is otherwise called the inverting converter. It produces an output voltage that is opposite in polarity to the input voltage. The magnitude of the output voltage can be greater than or less than the magnitude of the input voltage. Therefore, the converter can be used as a step up or step down converter. The input voltage is essentially inverted and the output voltage is of an opposite polarity to the input voltage. The buck boost converter also operates very efficiently with very little power loss. Similar to the other topologies, it utilizes a transistor as a switch that forms two different operating modes for the converter. The lower diagrams show the current flow paths for each of the modes when the switch is ON and when the switch is OFF.

When the switch is turned ON i.e. Mode 1, the input voltage is forced across the inductor, causing an increasing current flow through it. During the ON time, the discharge of the output capacitor is the only source of load current. Therefore, the capacitor is discharging during the ON time. When the switch is turned OFF i.e. Mode 2, the decreasing current flow in the inductor causes the diode to turn ON, allowing the current in the inductor to supply both the output capacitor and the load. Therefore, the capacitor is charging during the OFF time. A diagram of a buck boost converter is show in figure 29 below.

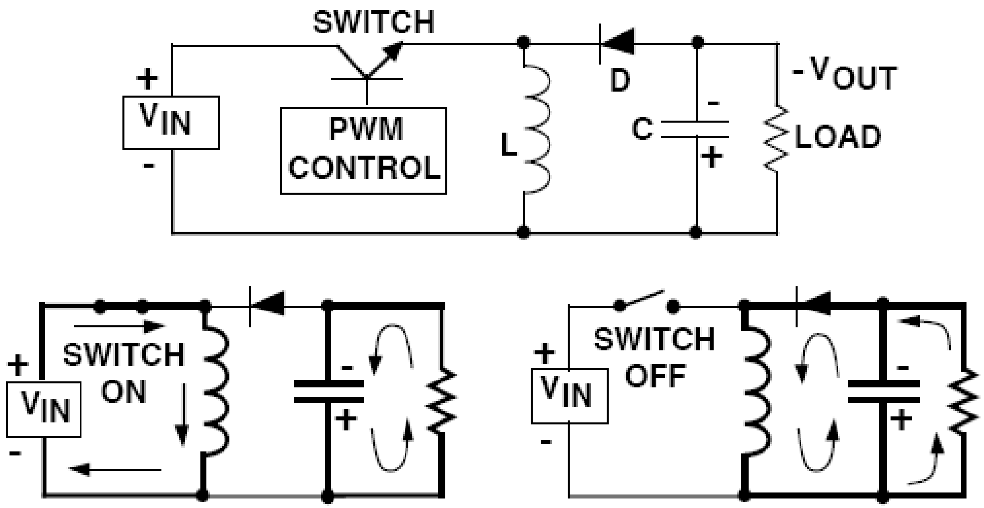


Figure 29: Diagrams of a Buck Boost Converter (Reprint permission granted through site www.national.com)

Based on the characteristics and the analysis of the different types of switching voltage regulators, using a buck converter would be better, because the robot's voltage source will always be greater than the required voltages of the subsystems, and a buck converter is design to produce a lower output voltage than the output. Since switching regulators are more expensive than linear regulators, we will probably use linear regulators as well.

Both linear and switching voltage regulators have their uses, and both have their advantages and disadvantages. To better have a clear picture on when using a linear of switching voltage regulator would be appropriate or better for a circuit, table 16 shows a comparison of both types of voltage regulators.

Parameter	Linear Voltage Regulator	Switching Voltage Regulator
Function	Step Down only, output voltage must be less than input voltage	Step up or Step Down, can produce multiple outputs
Efficiency	Low to Medium	High
size	Small to medium in portable design- may be larger if heat sinking is needed	Large for low power- may be smaller in applications where linear require heat sinking
Weight	Light	Medium
Linear and Load Regulation	Low line and load regulation percentage	Low line and load regulation percentage
Output Ripple	Negligible	Large
Noise	Low and better noise rejection	Medium to high due to rippel
Transient Response	20µs	1.0ms
Hold up time	1.0ms- 2.0ms	20ms- 30ms
Waste Heat	High, if average load and voltage difference are high	Low, components usually run cool for low power levels
Complexity	Low, usually only requires the regulator and a bypass capacitor	Medium to high, usually requires inductors, diodes, capacitors, transistors, filters, etc
Cost	Low	Medium to high

Table 16: Advantages and disadvantages of Linear and Switching Regulators

As seen from the table above, both converters have their unique advantages and disadvantages depending on its application. The choice of a converter for the robot will be to satisfy its cost requirements. Therefore, considering that it is operating at very low voltages and the heat waste is negligible, the robot is more inclined to use the low dropout linear regulators to build our power supply for the rover. The LDO regulates the input voltage, providing the desired output voltage and it works with a lower voltage across it before it stops regulating. A popular example of the low dropout regulator is the LM2940-CT-5.0. It is a low dropout regulator that fits into the same socket as a 7805 but can still regulate with less than 1 volt across it. For instance, using a regulator like this will guarantee that a 9 volt output from our 10 volt battery source is sufficient to still provide around 9 volts to our camera for the video vision operation of the robot. The use of the low dropout regulators can be considered an upgrade from the conventional 7805 style regulators that are commonly used the electronic circuits. However, if we are upgrading a power supply to an LDO, we must make sure that we have enough output capacitance to stabilize the output voltage. The commonly used 7805 style regulators have in built compensators and hence, do not need additional output capacitors on their output pins to maintain a stable output. Typically, 0.1NF bypass capacitors on the input and output terminals of the regulator are good for noise reduction and they are also critical for stability. The LM2940 series low dropout regulators, like most LDO type regulators need a minimum output capacitance. In the case of an LM2940, the precise capacitor value is 22 NF. This capacitor value can be increased as much as the design allows, but a lower capacitor value will not be acceptable. Commonly used capacitor values for low dropout regulator stability are in the range 100NF to 1000NF.

Sometimes during operation, the robot may require large amount of currents, currents that the battery may not be handle, for example if one of the motors or servos abruptly reverse while the robot is going full speed. To handle these large amounts of currents, the robot hs a quick release energy reserve. In order to provide that energy, the use of a large capacitor between 1mF to 10mF, which can store a large quantity of energy in the circuit of the robot, and release it whenever the robot needs it. High frequency noise in the circuit can cause a number of different problems; one example regarding the robot would be RF interferences. To prevent these high frequency noises, a capacitor between the ranges of 10nF to 100nF is used in the circuit. In a circuit that has noise, sometimes currents change directions, if the current in the circuit flow in the wrong direction the microcontroller in the robot may be damaged, therefore to ensure that the current in the circuit will flow in one direction, the circuit uses an LED as a power indicator and also to keep current flow in one direction. Also to conserve energy and to control when the robot is operating, a switch is used. There many different types of switches, but the robot has a single pole single throw toggle switch to turn it OFF and ON; it is simple and easier to use than the other switches.

Capacitors

Electrolytic Capacitor

For the quick release energy storage in the robot, an electrolytic capacitor is used, because electrolytic capacitors have a large capacitance per unit volume, so they are really helpful in fairly high current electrical circuits. And they are relatively cheap and are readily available.

Polyester Film Capacitor

To reduce the high frequency noise in the circuit the power supply use a metalized polyester film capacitor, they have low tolerance, which is adequate for a lot of applications. And also the this type of capacitor are temperature stable, cheap which is good for our budget since we want to spend the least money possible, and they are readily available.

Types of Fuse

Fuses, according to the time they require to respond to a high current are characterize as fast blow, which opens rapidly the rated current is reached, or slow blow time delay, which will only open in the condition when the high current is sustained. Therefore to protect the robot's servo motors and its power supply, the robot would need a slow blow fuse, but for this project, it is not necessary since the robot uses low power. To protect the wiring in the circuit it is necessary to rate the fuse to the gauge rating of the wire. Below table 17 shows the gauge to current rating.

Gauge	Current
12	41
13	35
14	32
15	28
16	22
17	19
28	16
19	14
20	11
21	9
22	7
23	4.7
24	3.5
25	2.7
26	2.2
27	1.7
28	1.4
29	1.2
30	.86
31	.7

Table 17: Gauge to current rating

Section 5: Design

This section discusses the actual design of the hexapod robot. The proposed method of implementation is shown here. While the research section demonstrated the various possible options looked at while trying to find a given solution, the design portion will discuss the solution chosen and expound upon how it will be brought into effect.

5.1 Mechanics

5.1.1 Chassis

There is not much left to design in terms of the body and camera chassis, since they were bought prebuilt. However, this was exactly the team's goal: to not have to design the chassis from scratch since once researched it was determined that the cost of doing so, in terms of time, was not warranted. Section 6 will discuss how the robot will be built, and Section 6.1 will go into detail of how both chassis will be put together.

5.1.2 Leg Connections

Similar to the chassis, the leg packages bought obviate the need of preparing a meticulous mechanical design. However, there are still various ways to connect the working legs to the chassis. The method of connection is an important one, since the walking algorithm will of course be dependent upon their particular geometry and configuration. One of the specifications for the chassis was to provide leg connections all on the same two-dimensional plane, and this specification was met by the body chassis obtained. In order to maximize this setup and allow for maximum range of motion during the robot's gait, all legs will be connected in such a manner that their vertical length is perfectly orthogonal to the chassis and the horizontal width is perfectly flush to the connection. Satisfying one requirement should satisfy the other. Pictures are present in Section 6, but to help envision such a connection, it is helpful to consider each leg as a pillar and the body chassis as a ceiling; the optimal configuration is to have all pillars perpendicular to the ceiling they are holding up. This leg configuration will enable the joints to exhibit a maximum range of motion, since the legs will be in a position where the relative distance between each leg is a maximum. Although many other configurations are possible, such as three legs at a given angle while three legs at a similar complementary angle, this setup not only provides the Drive subsystem with the most flexibility, but the team also believes it looks most natural on the circular body chassis.

5.2 Drive

5.2.1 Hardware

There will not be much more design in terms of the hardware involved for the drive subsystem. The 18 servomotors will be connected to the servomotor controller board, and will then await input from the wireless communication control. The vast majority of the design will go into the software portion of the drive system.

5.3.1 Program Design

Time and Step Definition

The technique of inverse kinematics will be used to position the motors appropriately in order to create linear motion. In order to avoid any difficulty in syncing the different steps together, each step will occur a given value δ after the previous step. That is, the time step will be uniform across all steps. So if the walking sequence consists of 6 steps, then the first step will occur, and then the second step will occur δ afterwards. The value δ is in units of time, such as milliseconds. For the purpose of analysis, δ will be equal to zero ($\delta = 0$) instantaneously after a step, and will be negative before it. Thus, if it is necessary to begin a step before another step is finished, then δ will assume a negative value. It seems to be a reasonable assumption to make right now, that the time constant between steps, whether it be negative, zero, or positive, can stand to be uniform across all steps. This fact may be altered of course as building progresses.

In order to further reduce complexity, the same linear motion algorithm will be used in order to advance the robot in any direction. Due to the fact that the robot has a circular chassis, and that the legs will be connected with even spacing, the robot will be rotationally invariant. Therefore as long as the axes of linear movement are all equiangular, the same sequence of steps can be used to cause linear movement regardless of the direction given to the robot from the radio controller. Of course, if a command is given to the robot in the middle of its sequence, the walking algorithm must be able to either immediately reset to the beginning (standard position) or be able to transition smoothly into the next position. By having the robot only change directions on timed multiples of δ, then the amount of movement that the hexapod is doing should be minimized since it will either be undergoing zero movement, or be in the transition point between two steps, where one step will be finishing and the other step will be just beginning.

Since movement in every direction will consist of the same sequence of steps, all that remains is to identify the sequence of steps that compose movement. This problem will be simplified by considering the six legs as only three legs, where the two legs that are directly across each other will move in parallel. Whatever the front leg does, the back leg will do. The only difference will be that some of the final movements will be mirrored or inversed, or else one can imagine that one leg would try to move forward while the other leg would try to move backward. Therefore the same joint articulations will exist, but they will all be mirrored. Now since there will only in essence be three legs to worry about, it seems natural to have a sequence of either 3 or 6 steps as a full sequence executing linear motion; again with a time value of δ (which may be negative) between each step for syncing purposes.

5.2.2.2 Model Derivation

In the previous subsection, the analysis of the six legs was reduced to only three legs due to the properties of symmetry in movement; a form of coupling the legs in order to reduce complexity. Therefore, only three legs need be analyzed in any direction. By making the center of the directional heading on the middle of the three legs (front legs which are coupled to the back legs) to be analyzed, there exists symmetry between the left and the right legs. Therefore, all that needs to be done is derive a model for the left (or right) leg to move, mirror it on the opposing leg, and derive a model for the front leg. Thus the entire walking algorithm can be simplified to deriving suitable models for just a single front leg and a side leg.

Deriving a model for this type of situation is somewhat intuitive if one only considers the two legs at once; it seems like a hobbled human being. The side leg should raise its vertical hip position, move its hip position horizontally, extend its knee, and then lower its vertical hip position. This will be a side step forward. The front leg should then raise its vertical hip position, maintain its horizontal hip position since it is in the direction of movement, and extend its knee. This will be a step forward, and this step will return the side knee joint to its proper position. Then the alternate side leg will mirror the sequence of the side leg. This is the model derivation concept, where the actual angular values, and consequently servomotor positions, will be dependent upon the length of the legs themselves.

5.2.2.3 Program Execution

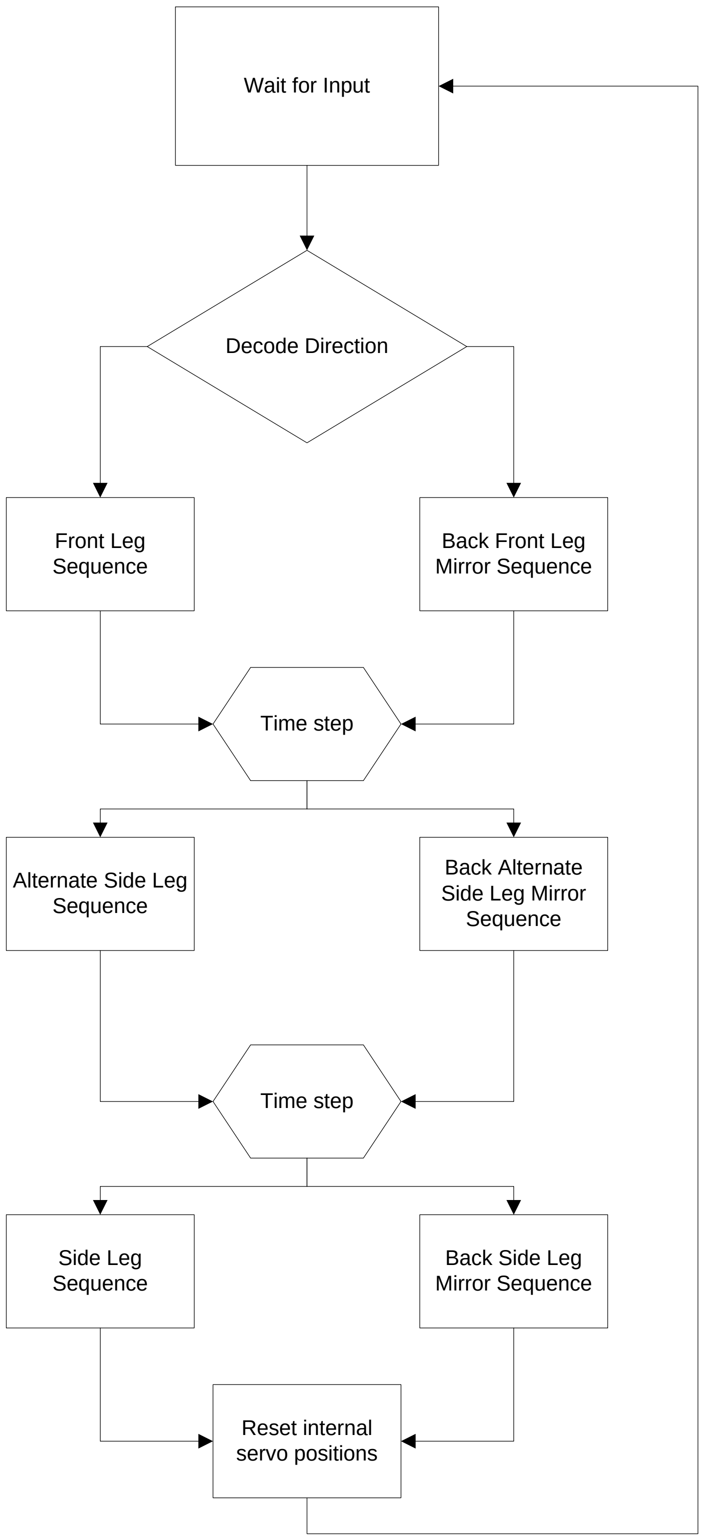


Figure 30: Block diagram of gait algorithm

In order to better illustrate the execution of the gait algorithm, a block diagram was created. Figure 30 shows this block diagram. The controller will await an input, it will then decode the appropriate direction and decide which legs will be moved, and then goes through the sequence of moving each leg with a time step in between each. Afterwards, it again waits for an input signal to determine what its next direction will be.

5.3 Vision and Color Sensing

5.3.1 Color Sensor

In lieu of the camera and FPGA/microcontroller implementation, we chose the color sensor, light-to-frequency generator. The unfamiliarity with the VHDL language and testing and building with FPGAs in general, led us to look at microcontrollers.

The microcontrollers were very easy to use but definitely had space limitations and we did not want to worry about an external RAM module for frame storage. The Interfacing the camera and microcontroller or FPGA proved to be extraordinarily difficult for the time allotted and we began testing photoresistor and phototransistor circuits.

We chose the TAOS TSL230RD light-to-frequency generator housed in the PICAXE module with the lens assembly. Its ease of use and ease of interfacing were the deciding factors.

Figure 29: PICAXE Color Sensor

The PICAXE Color Sensor will be the name used from now on in referring to the TAOS TSL230RD/lens assembly combination. The unit features 2 bright, white LEDs for illuminating the object which is being detected. This ensures that all colors of the spectrum will be available to be reflected back into the color sensor, since white light contains all wavelengths.

The lens assembly helps focus only light directly entering the color sensor, thus keeping out ambient light and ambient light.

The size of the entire PICAXE Color Sensor module is about 1 in by 1 in 0.5 in. which is well within our space considerations. Two lines, S3 and S2 are used for the color selection and will be connected in parallel with switches to inputs on the microcontroller. When in operation, the PICAXE Color sensor turns on whatever color filters are selected and then sends out square pulses at a frequency linear with the intensity of that light on the photodiodes. Thus, if there is green light getting through, and the green filters are on, then a very high frequency of square waves will be detected.

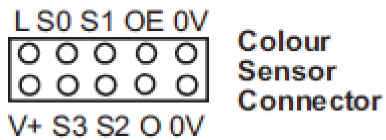


Figure 30: PICAXE Color Sensor Pin connections

- V+ will be connected to 5 V.
- S3 and S2 will be color selection.
- L will be LED power.
- OE is set to ground.
- 0 V and 0 V are set to ground.
- is for output and is connected to the microcontroller

Above lists the pins for the PICAXE color sensor.

Since it has very low power consumption. The color sensor will always remain on to avoid the added unnecessary step of a switch. The user will be allowed to change colors after turning on the sensor by a “pause/set” switch on the microcontroller.

5.3.2 Microcontroller

We chose the PICAXE18X microcontroller from Spark Fun Electronics to do our color processing. The PICAXE18X is easily interfaced with the PICAXE Color sensor. Please note that the only thing the two units have in common are the name. The PICAXE Color sensor is just the TAOS TSL230RD with a lens assembly on it to focus light.

The PICAXE18X has plenty of program and variable memory available for us as our program is relatively simple.

The PICAXE18X features a ‘COUNT’ command that counts the number of square pulses input within a certain sample time. Knowing this, we designed our program to count the number of pulses and test it against a decided threshold (found through testing) to check if the correct color is found. The user will be allowed to enter what color is desired by the two switches that will control the selection on the microcontroller and color sensor in unison. The user will also be allowed to change the color to be detected via a “set/pause” switch on the microcontroller. When the “set/pause” switch is high, the color scanning stops and the user selects a new color and then flips the switch to again begin color detection.

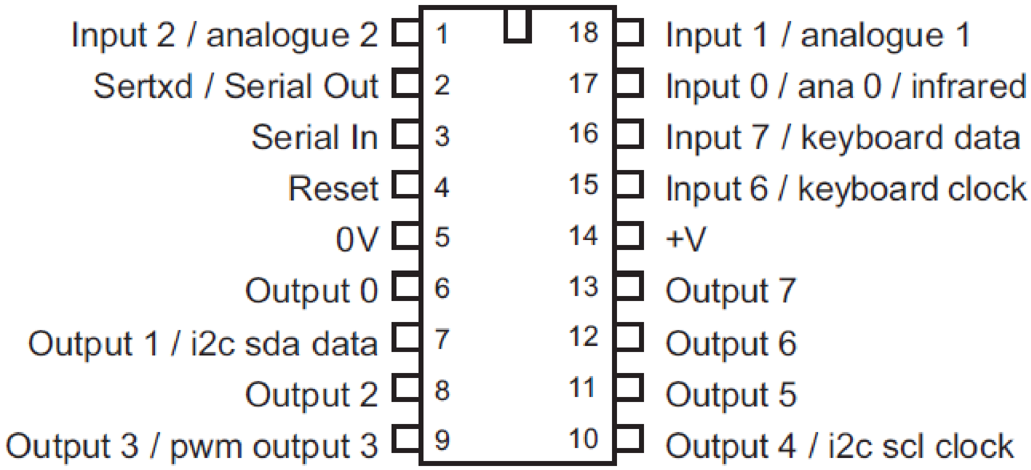


Figure 31: PICAXE18X Microcontroller Pins

The following pins of the PICAXE18X are used:

- Input 2 – color sensor input
- Input 1 – Set switch
- Input 7 – S3
- Input 6 – S2
- Output 7, 6, 5 – Red, Green, Blue LEDs

The PICAXE18X along with the switches and indicator LEDs will be housed on top of the robot on a printed circuit board.

Electrical data for the PICAXE18X is below. It meets the specifications just fine.

Absolute Maximum Ratings(1)

Ambient temperature under bias	-40°C to +125°C
Storage temperature	-65°C to +150°C
Voltage on any pin with respect to V_{SS} (except V_{DD}, $\overline{MCLR}$ and RA4)	-0.3V to (V_{DD} + 0.3V)
Voltage on V_{DD} with respect to V_{SS}	-0.3V to +5.5V
Voltage on $\overline{MCLR}$ with respect to V_{SS} **(Note 2)**	0V to +13.25V
Voltage on RA4 with respect to V_{SS}	0V to +8.5V
Total power dissipation **(Note 1)**	1.0W
Maximum current out of V_{SS} pin	300 mA
Maximum current into V_{DD} pin	250 mA
Input clamp current, I_{IK} (V_I < 0 or V_I > V_{DD})	±20 mA
Output clamp current, I_{OK} (V_O < 0 or V_O > V_{DD})	±20 mA
Maximum output current sunk by any I/O pin	25 mA
Maximum output current sourced by any I/O pin	25 mA
Maximum current sunk by all ports	200 mA
Maximum current sourced by all ports	200 mA

Table 18: Electrical Characteristics of PICAXE18X

5.3.2 Microcontroller and Color Processing Algorithms

To determine what color is being sensed, we really do not have any information directly from the object. All we have is a square pulse train at a certain frequency. Each surface reflects light back a little differently, so to develop our program, we used the same surface of different colors.

The program begins by seeing if the ‘SET/PAUSE’ button is set. If it is high then it will pause for 5 seconds and allow the user to change the color. If it is low then it will determine what color is selected by reading the inputs on input 7 (S3) and input 6 (S2). Depending on which color is selected, it will jump to the respective subroutine.

Once in the subroutine the microcontroller will sample pulses for a specific time and then test that against a determined (from testing) threshold. If higher than the threshold then the color is a match and the program will light the indicator LED for that color.

The overall program structure is illustrated below:

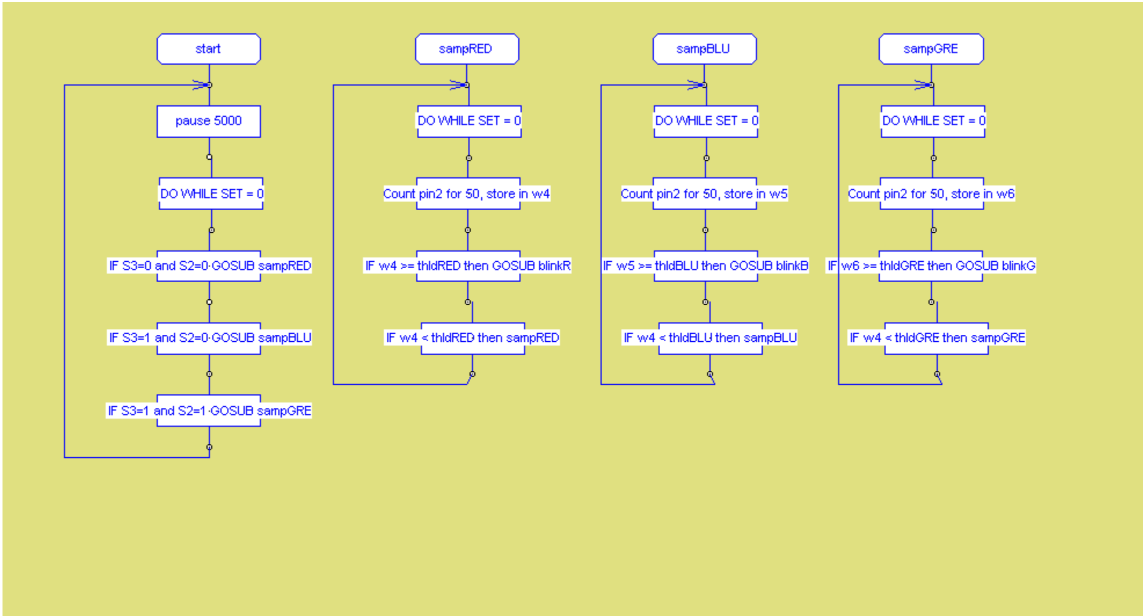


Figure 32: Overall Color detection program for PICAXE18X

5.4 Wireless Communication

The Wireless Control Unit will consist of the Playstation 2 controller, the controller interpreter, and the transceivers. It will be a two-way communication or data transfer between the controller and the interpreter, one of the transceivers and the interpreter, of course between the transceivers, then between the second transceiver and the robot.

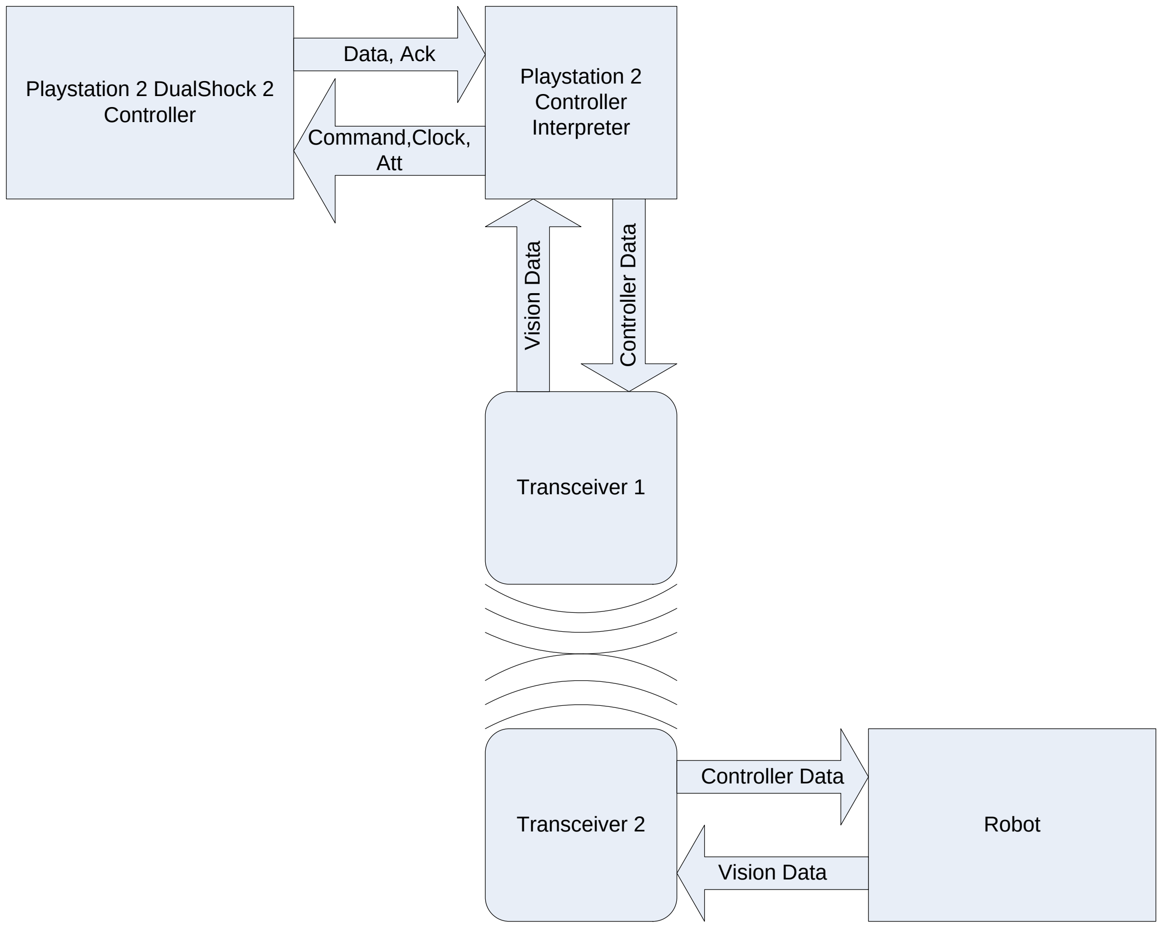


Figure 39: Block diagram of the wireless system.

Playstation 2 Controller and Playstation 2 Controller Interpreter

As explained in details above in Section 3.4.1, the Playstation 2 controller will be communicating with the Interpreter circuit, the latter emulating an actual Playstation 2 console. The interpreter circuit will supply the necessary voltage, the command data and the clock signal to the controller, whilst the latter will send the data essential for the control of the robot.

Playstation 2 Interpreter and Transceivers

In addition to communicating with the controller, the Interpreter circuit will also transmit and receive data from the transceiver. After processing the data from the controller, it will transmit the data to the robot's central unit though Transceivers 1 and 2. Moreover, it will receive through the Transceivers the vision-related data from the image processing unit of the robot, and process them and turn on the appropriate LED indicators.

Transceivers and Robot

The Robot will receive the movement control data from the Playstation 2 controller through the Transceivers and process them accordingly. It will also send vision-related data from its image processing unit to the Interpreter circuit which will process them and respond accordingly.

5.4.1 The Controller Interpreter

The Interpreter is designed around Atmel's AT90S2313 and Microchip's PIC18F458 microcontrollers. As shown in the diagram below, the Atmel microcontroller is the only microcontroller on the circuit communicating with the Playstation 2 controller and the brain behind the emulation of a Playstation 2 console. As for the PIC18, it is to collect and process the image data from the image processing unit of the robot through the transceivers.

5.4.1.1 The AtomPro 28-M

The AT90S2313 main role is to allow the robot to be controlled by the Playstation 2 controller. It receives the 8-bit serial data from the controller and spits out the 8-bit data in parallel format to the robot's processing unit through the transceivers. It also provides the Playstation 2 controller with a clock line to keep both of them in sync, necessary for error-free and orderly data traffic.

Software Design

Using an interrupt-based approach will allow the AT90S2313 Microcontroller to constantly poll the Playstation 2 controller, without having to use an on-demand key-reader routine. The microcontroller runs off a 4MHz clock source with a divide-by-64 prescale, giving the timer a 16 microsecond interval, which means the interrupt rate is expressed the following manner:

$$Interval(\text{seconds}) = \frac{1}{CLOCK\ RATE(hertz) \times PRESCALE}$$

Polling every 16 microseconds is too fast, so, to increase the interval, a 16-bit timer/counter register (TCCR1) is used as a time keeper, then a value of 0x0411, or "1041", is loaded into the timer-compare register (OCR1A). This increases the time interval for polling to $16\ microseconds \times (1041 + 1) = 16.672\ milliseconds$.
The key-reading interrupt used in the interpreter is based on the flowchart which is figure 40 below.

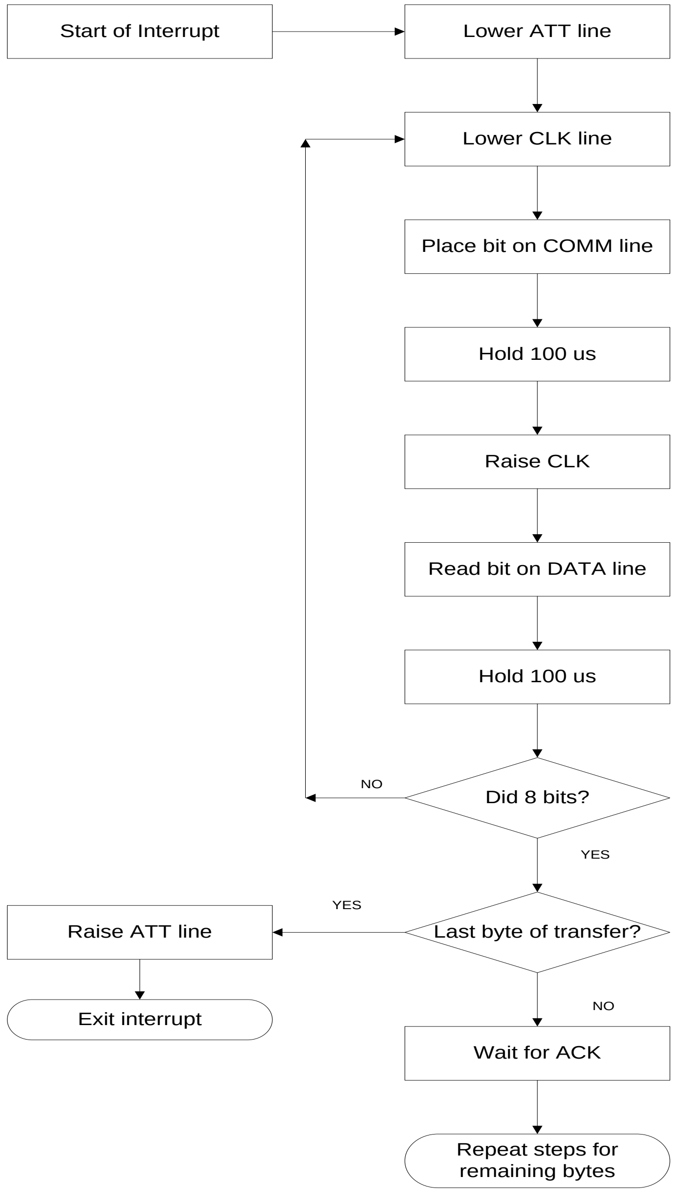


Figure 40: Flowchart of the key reading interrupt.

5.4.1.2 The PIC18F458 Microcontroller

The PIC18F4580 is to receive and process the data from the robot's processing unit pertaining to vision and decisions made by the user through the use of the Playstation 2 controller. Using the Playstation 2 controller, the user will select a color to be searched for by the autonomously controlled camera mounted on the hexapod. When found, the image processing unit will notify the user through the PIC18 that will light up the corresponding LED (red LED for red object detected, blue LED for blue object detected, green LED for green object detected). Then, the user will make the decision of whether to "kill" the detected target, or to "unlock." The decision will be sent to the robot's processing unit, which will relay it to the PIC18 that will light the LED corresponding to the decision of "kill" or "unlock."

5.4.2 The Transceivers

Obviously, two transceivers will be necessary to maintain the RF link between the controller and the robot. AeroComm's AC4424-10 transceivers are chosen for this task. They operate in the 2.4 GHz ISM band and can maintain data rates up to 1152 Mbps. However, they come equipped with MMCX Jack for compatible whip antennas.

5.5 Power

Power Supply Design

The robot's power system design is influenced by the different power requirement of its subsystems. Given that the power is a DC voltage source that only provides a constant voltage while the subsystems require different voltage levels in order to operate. Therefore the robot uses voltage regulators to convert the high source voltage to lower voltages required by the different subsystems. A voltage regulator is device that accepts a DC input voltage and produces a DC output voltage. In general the output voltage is at a different voltage level than the input voltage. The input voltage may be lower or higher than the output voltage depending on what it is used for. When the input voltage is higher than the output voltage, the regulator is referred to as a step down voltage regulator. And accordingly, when the input voltage is lower than the output voltage, the regulator is referred to as a step up voltage regulator. For our robot, the source voltage will generally be higher than the required voltage of the different subsystems; therefore the power supply implements the use of step down voltage regulators.

Regulators in general are used for the most part in electronic devices and circuits. These devices most of the time contain sub-circuits and subsystems where each of them require a different voltage level that is different from the voltage source supplied by the battery. The voltage required by these subsystems and sub-circuit can sometimes be higher or lower than the battery source voltage and possible be a negative voltage and it drains the stored voltage in the battery, therefore the source voltage of the battery will decrease. DC to DC regulators allow us to generate multiple different controlled voltages from a single battery source voltage. As a result, it saves on space and complexity of using a number of batteries to supply every individual part or sub-circuit of the robot. DC to DC voltage regulators are widely divided into linear voltage both types of regulators and examine their respective use and efficiency. The comparason between the disadvantages and advantages of both types of regulators in order to choose the regulator type that satisfies our different specifications and meets our efficiency and cost requirements. The calculation of the maximum power requirements for each individual subsystem of the robot is done; which are voltage and current required by each. The subsystem and their power requirements are listed in table 19 below.

Blocks	Voltage (V)	Current (mA)	Maximum Power (W)
Control Unit			
Micro-Controller	2.0 – 5.5	25	.138
Vision Unit			
Color sensor	5	30	.15
Micro-Controller	2.0 – 5.5	25	.138
Remote Sensing Unit			
Micro-Controller	2.0- 5.5	25	.1375
Receivers	4.9 – 5	80 – 140	.719
Transmetters	4.9 – 5	80 - 140	.719
Mechanical Unit			
Servocontroller	6 – 9	31	.279
Servo-Motors (18)	4.8- 6.0	1100	6.6

Table 19: Subsystems required voltage, current and power

The maximum power requirement of each subsystem is classified in the table above and so is the overall maximum power requirement of the robot. The table helps decide the battery source voltage and the type of regulator that is going to be needed to reduce the source voltage to the required voltage of each subsystem. To decide the voltage of the battery source, the power supply also includes the losses in the lines, losses cause by heat and interferences. Between a 5% to 10% loss factor to our battery voltage source to compensate for losses when the robot is operating is included. The battery source for the electronics is 12V and the servomotors battery source voltage is 6V. Each subsystem has its own voltage requirements; therefore regulators are used accordingly to regulate the 12V battery source to satisfy the voltage required by each subsystem.

Efficiency, cost, losses, and noises are considered in building the power supply of the robot. Losses in any power supply system are inevitable. However we can reduce the losses in the system to make sure that an adequate supply of power is delivered to every subsystem. Wiring losses are reduced by making sure that just enough of wire runs between the voltage source and the target, because the shorter the wire the smaller the loss in the wire. The right wire size is also important in power transmission; which is determined by the amount of current and voltage that is running through the wire.

Where to place the regulators are also very important. Losses are unpreventable when the robot is operating, so placing the regulators close to the voltage source causes less than desirable power supply to the robot's subsystem, which causes that particular subsystem to malfunction. Therefore the total source voltage is ran at its rated level and regulate it just before it is fed to the targeted subsystem, so we guarantee that the desired voltage is delivered to its target. Therefore there is always the need to make sure that the voltage used is high enough so that the voltage drop cause by the wiring does not drop the supply voltage lower than the minimum voltage needed for the regulator to operate. Once all the required voltages are delivered to every subsystem, we can guarantee that the robot operates properly. Since we are going to use a rechargeable battery, it is necessary to have a charger; which is going to be bought with the main battery source.

Power System Noise Issues

One of the biggest problems encountered when building a robot is noise. In general robots will have digital electronics, sensitive analog sensor processing electronics and power electronics to drive its motors and circuits. The circuitry and digital electronics are constantly subjected to noise interferences and power surges from the motors and servos. Analog signals can be corrupted with noise by either the digital or power electronics. The standard and most efficient way to prevent noise in a power circuit is isolation. Isolation ensures the separation of power electronic and motor subsystems from digital electronics subsystem. The separation of the subsystems eliminates or reduce noise interference and improve the overall performance of the robot. The power circuit could isolates the regulated power that is going to electronic subsystem circuitry from the power that is going to the servos and motors circuitry.

Another configuration is to have one unregulated power source, and then have separate regulation systems for the supply circuit of the motor, the digital supply and the analog supply circuitry. And all the systems would have a common ground at the power source. Having a common ground can be advantageous, because the control signals from the different parts of the system can be sent just like any other signal. But noise can also travel back up the control lines and corrupt the otherwise clean power. Therefore this negates the use of a common ground in our system. One more configuration is to have unregulated power sources for the major subsystem circuitries. The motors tend to have a huge current draw on the system and this can cause voltage surges and ripples which are really bad for the electronic circuitry.

Therefore this configuration proposes the use of separate power source system for the motor circuitry and a separate power system for the electronic circuitry. This configuration will make sure that the voltage distortions caused by the large draw by the motors does not affect the electronic sub-circuits. As a result the power supply would be able to eliminate the majority of the interferences and deliver a clean power to all the robot’s subsystems and sub-circuits.

For effective distribution of power to all of the robot’s subsystems and sub-circuits parts as seen in the table 1 above, the power supply separates the supplies for part of the circuit that require precise voltage such as the microcontroller and sensors, and those that do not require a precise voltage such as the servos and motors etc. This makes sure that the microcontrollers of the robot are not affected by the supply changes due to current surges in the servos and motors.

Robot Power Source

The electrical noise analysis suggests the need for a power supply that is designed to reduce or eliminate noise interference between subsystems. As a result the robot adopts the isolation configuration as a valid solution to the noise interference problems between subsystems. The primary voltage source for our robot is a DC battery voltage source. The main power is distributed to all the subsystems according to their respective voltage,

current, and power requirements. Since the power requirements for each subsystem varies, because subsystem requires a different voltage. Since interferences between each subsystem is also an issue to consider in the power supply design, the subsystems are devided into two different blocks: One block has a huge current draw and causes large voltage distortion such as the servos subsystems and the other block has a lesser current draw and does not tolerate voltage distortion such the microcontrollers subsystem.
The first block consists mainly of the servos; which drive the legs of the robot. As seen from Table.1, the robot uses 18 servo-motors to move the robot. Therefore most of the power is consumed by the servos, and since they draw a lot of current they cause large voltage distortions. These distortions affect the electronic components; therefore having a separate power supply for the motors is better and more efficient for the robot. The second block will consist mainly of the electronic devices and components that form the electronic circuitry and control of the robot. These components in general draw small amounts of currents and are very sensitive to voltage distortions; the power supply uses voltage regulators to deliver exactly the voltage and current required by these electronics so the subsystems can efficiently operate. Below figure 41 and 42 show block diagrams of the two different blocks.

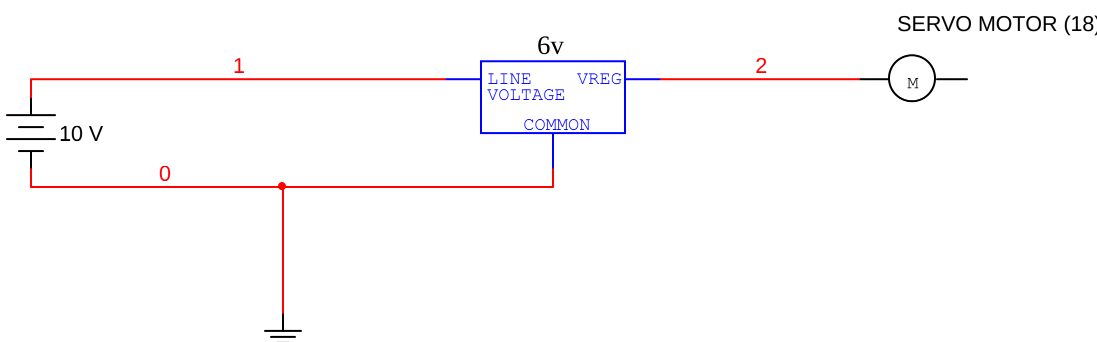


Figure 41: Power supply block diagram for servo-motors

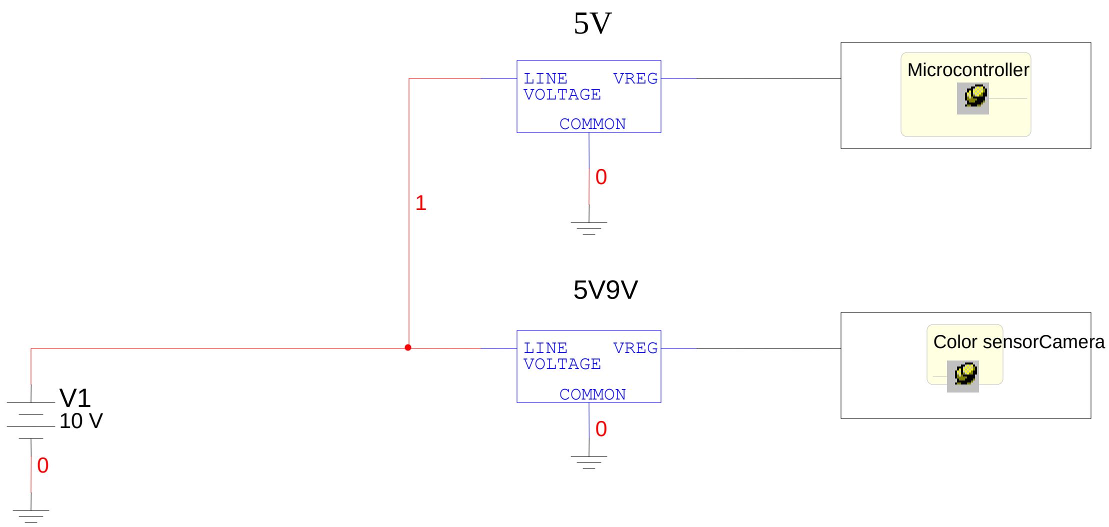


Figure 42: Power supply block diagram for the electronics subsystems

Section 6: Building Prototype

This section is concerned with providing the details necessary to physically assemble the robot. The integration of certain subsystems will be shown, and how they will come together in order to produce the final product. Since the robot has not been built yet, the pictures and figures used do not always represent the team's robot. However, the images are close enough in resemblance to help illustrate the points being made in the text. The instances where the images do not directly correspond to the project will be clearly identified.

6.1 Mechanics and Drive

6.1.1 Leg and Motor Assembly

If the legs are not properly built, and the servomotors are not properly attached, there is no hope for the Drive system to ever work, no matter how good the code is. Therefore detailed directions will be given on building each leg, building the joints, and attaching the motors. All of these steps are actually done in unison, since the motors must necessarily be integrated with the legs and joints in order to cause motion. Pictures will be provided to illustrate every step, since the details can get unwieldy. Although these steps are given for the particular leg and joint models chosen, many of these same general concepts can be applied to any model. The essence of this step is to put the motors in such a position where they can actuate the joints to exhibit the three independent degrees of motion that are desired.

6.1.1.1 Foot to Knee Assembly

Push the servo through the leg as shown in Figure 43. This servo is responsible for knee motion.

Figure 43: Knee actuator placed in leg from www.LynxMotion.com (reprinted w/ permission)

Push the spacer over the leg as shown in Figure 44. This will secure the motor and give it proper spacing.

Figure 44: Spacer inserted from www.LynxMotion.com (reprinted w/ permission)

Push the second complementary leg piece over the spacer as shown in Figure 45. This locks the servo in place and finishes the foot to knee portion of the leg.

Figure 45: Servo locked by leg structure from www.LynxMotion.com (reprinted w/ permission)

Insert the appropriate screws to fasten both the servomotor and the legs together, as shown in Figures 46 and 16.

Figure 46: Screws to fasten motor and leg from www.LynxMotion.com (reprinted w/ permission)

Attach the rubber end cap onto the end of the leg. It should cover all the way past the nut. This is shown in Figure 47.

Figure 47: Rubber cap inserted from www.LynxMotion.com (reprinted w/ permission)

Take the joint or servo hinge shown in Figure 48, and place it over the servo as shown in Figure 48. This is the interface of a motor with a leg joint.

Figure 48: Hinge shown & used for knee joint from www.LynxMotion.com (reprinted w/ permission)

This concludes construction of the first portion of the leg. The next portion to be assembled will be the hip joints with their motor connections.

6.1.1.2 Hip Joint Assemblies

Attach a ball bearing to the multi-purpose bracket as shown in Figure 49. Refer to Figure 50 for detailed information on how to do this.. Note:, this ball bearing is part of the body chassis. This bracket is necessary, since the fate of the legs, through the hip joint, is to be connected to the body chassis.

Figure 49: Bracket and connection method from www.LynxMotion.com (reprinted w/ permission)

Now, attach the two multipurpose brackets as shown in Figure 49, using the screws and nuts depicted in Figures 49 and 50.

Figure 50: Bracket attachment with hardware from www.LynxMotion.com (reprinted w/ permission)

Attach the two "C" brackets as shown in Figure 51, with the two slightly smaller screws and nuts shown in Figures 51 and 52.

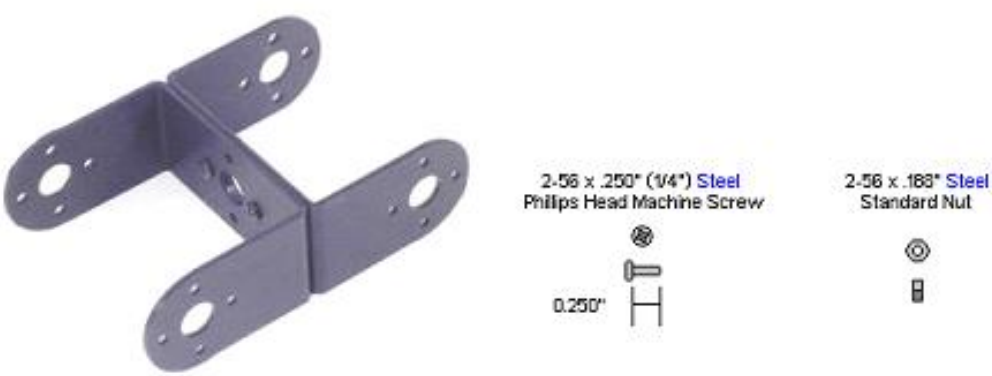


Figure 51: C Bracket connection w/ hardware from www.LynxMotion.com (reprinted w/ permission)

Connect the two joints that have just been assembled as shown in Figure 52. Figure 53 provides information of how to link the hardware.

Figure 52: Hip Joint connections w/ hardware from www.LynxMotion.com (reprinted w/ permission)

Now the joints have been assembled, and what is left is to integrate the motors. Attach the servo to the "C" bracket using the #2 tapping screws and Figure 53 below. Make sure that the servo is centered before attaching to the "C" bracket.

Figure 53: Horizontal hip motor connection from www.LynxMotion.com (reprinted w/ permission)

Do the same procedure for the second servo, and again, make sure to route the wire as shown. These motors will provide hip joint motion in both the vertical and horizontal planes; they are the final two degrees of freedom needed to build the design. Figure 54 illustrates the connection of the second servomotor.

Figure 54: Vertical hip motor connection from www.LynxMotion.com (reprinted w/ permission)

6.1.1.3 Knee to Hip Connections

Now the entire leg will be connected. Attach the servo to the "C" bracket using the #2 tapping screws and the Figure 55. Make sure that the servo is centered before attaching to the "C" bracket. Make sure that the leg is properly aligned, that is, that the knee and hip joints are all relatively orthogonal to maximize range of motion as specified in the specification and design sections.

Figure 55: Joint integration from www.LynxMotion.com (reprinted w/ permission)

Finally, attach the "C" bracket to the servo hinge as shown. Refer to Figure 55 for illustration of the connection, and Figure 56 for an illustration of the connectors. Make sure to route the servo wire as shown. Leave enough slack for the leg to be able to freely move without pulling on the servo wire.

Figure 56: C bracket connection to leg from www.LynxMotion.com (reprinted w/ permission)

This concludes the procedure for assembling an entire leg, including the leg itself, the joints, and the motors. The same process must be repeated six different times; one time for each leg. Once each leg is built, the next step is to assemble the body chassis, and connect the legs to the chassis, which is the scope of the next two subsections.

6.1.2 Chassis Assembly

Since the body and the camera chassis already come prebuilt, there is not much assembly involved. The body chassis itself comes with instructions, where all that is needed is to insert the given screws across from the top to the bottom of the body chassis and then fasten them tight. Figure 57 below shows a view (a yellow chassis, not the same chassis color but identical in shape) with the screws through it.

Figure 57: Body chassis to be fastened from LynxMotion (reprinted w/ permission)

If the necessary electronics will fit through the side of the chassis, then they may be inserted that way. If not, they will need to be inserted prior to fastening the body chassis closed. The camera chassis will simply go on top of the body chassis, and consists of fastening the camera chassis onto the center of the body chassis. This may done in any manner, as long as it is secure. Preferably, two holes will be drilled into the body chassis and secured from the back end with appropriate nuts.

6.1.3 Chassis and Leg Integration

As seen above in Figure 57, the chassis already provides tapped holes for the integration of the legs. The holes are also equally spaced, as specified in the design. Therefore, once both the body and the legs are independently assembled, they may be integrated by simply attaching the legs via the given tapped holes onto the chassis. Once must make sure that the legs are all aligned and are perfectly orthogonal with the body chassis in order to provide maximum range of motion, and the same range of motion for all legs relative to the body.

6.2 Drive

The major hardware component of the drive system is the collection of servomotors. Their assembly has already been integrated into the assembly of the joints. The servomotor controller board will then be inserted inside the body chassis, as will all other electronics. The gait algorithm itself will have already been uploaded onto the microcontroller in the servomotor controller board, and requires no extra building.

6.3 Vision

The color sensor system requires little more than connecting wires to pins, and mounting hardware on the chassis.

6.3.1 Color Sensor

The color sensor will be mounted to the bottom of the robot. It will scan the area underneath for objects of certain color. The output wire of the PICAXE Color sensor will be connected to pin 1 of the microcontroller.

6.3.2 Microcontroller

The microcontroller will be soldered onto a printed circuit board along with the indicator LEDs and switches.

Section 7: Test

Various tests will be conducted to ensure that the robot is working as it is expected to work. Since the project has been split up into various subsystems, each subsystem will be tested individually prior to integration. This method prevents the team from having to concoct very elaborate and complicated testing strategies due to having various subsystems integrated at once, and greatly reduced the potential problem space. For example, if all of the systems were integrated, and an error occurred, it would be difficult to discern which subsystem failed and caused the error. As an example, consider the event where the robot fails to locate a color. The error could obviously be a result of the color sensor system, or perhaps it could be a result of the communication system. Perhaps the robot is properly locating objects, but it is just not sending the signal back to the remote control properly. Another reason might be that the signal is being transmitted just fine, but the controller is having difficult receiving it. In order to eliminate the problem space of errors, each subsystem will undergo a battery of tests itself, and then the subsystems will be tested in each integration phase. Finally, of course, the entire robot must be tested in order to ensure proper functioning and seamless amalgamation.

7.1 Testing Environment

First of all, in order to test the robot, we will need an open area somewhere on the UCF campus. The testing environment will need to have enough space for the robot roam around while it is looking for its target. The room will also contain all the colors of various shapes that the robot will need to identify. The objects will vary in sizes and colors. To show the robot's intelligence and capability to effectively look for the different colors, the object will be scattered around the room. This will also show effectively the robot's field of vision, its ability to walk, and the accuracy with which the robot find its target.

Since it is possible that it may be raining, the preferred testing area for the robot would be an indoor open area, therefore the testing will be planned to take place indoors instead of outdoors. The size of the available testing area will determine how challenging the robot's task will be, it will also show how long the robot can run before the battery will need to be recharged.

7.2 Mechanics

The mechanics subsystem will be rather easy to test and fix or calibrate if it is not functioning properly. It is an important to make sure the mechanics subsystem is working properly before the integration of any other subsystem, because a failing in the mechanics will essentially ensure all other subsystems to fail. Hand-testing each individual joint and making sure that none are too loose or will be detached or twisted by the strength of the motors takes care of ensuring the structural integrity of the system. Then making sure that the joints themselves are loose enough to allow for rotation, but tight enough to provide security is the second test. All joints should be calibrated to pivot loosely in their given range of motion, and should be checked that they do not fail at or exceed the limits of their ranges of motion. Any loose joints should be tightened, and any joints that are too tight should be loosened. If only human bodies were this easy to fix. One should pay particular attention to the procedure followed when checking the joints, since when testing the drive system one might find that the joints are still too stiff for the given motor torque and must be loosened appropriately. Iterative testing between the mechanical and drive subsystems will be followed in order to get a strong yet smooth rotation of the joints and thus gait.

7.3 Drive

First a test program will be created to test that the full range of motion for each servomotor is working as specified. If this were not so, then clearly no other portion of the drive system would work. The second test will be that the robot is able to walk in any of the given directions as specified. That is, a test program will be loaded into the drive in order to examine the capability of the robot to walk in an arbitrary direction. At this point in time, the drive system will be tested independently of the communication system, and thus no wireless control is being given. It will be checked that the drive system is meeting the necessary specifications of clearing rate, and speed, and if it isn’t, then either the original specs will be reworked, or the parameters of motion will be reestablished. Once the robot is able to faithfully move in any direction and transition from any direction to any other direction at different points in time without any fail or falter, then the drive system will have been tested.

7.4 Color Sensing

7.4.1 Color Sensor

In the testing phase of the vision system, there are 3 main variables: the PICAXE color sensor, the microcontroller, and the object’s surface which we are detecting. The surface of the object will determine how much light is reflected back from the object. And since this is what we are detecting and turning into square pulses, this is very important to us. We also needed to ensure that the white light from the LEDs formed a solid dot on the object- no “dark space.” Dark space would mean no light is reflected to the lens. For testing we decided to use colored poster board – a reflective surface with very little color variation. We first worked on just being able to detect any reflected light into the color sensor. We found that the more the lens is screwed in, the better it focuses light onto the internal TAOS TSL230RD. This was a problem early on. Once our test program was able to detect light a given threshold we moved on to getting different colors. This required more extensive programming.

7.4.2 Microcontroller

Our microcontroller program was written in BASIC as this was the default programming language for the device. The main part of our design that was missing up until now was the threshold levels for each of the colors. Up until now we had been testing with the same threshold just hoping to get any colored light detection. In the finding of our thresholds, we were getting indications of the correct color match for two different colors.

We used a sample time of 50 ms for collecting our color data as that was recommended in the user manual of the PICAXE color sensor.

The test procedure for finding the right threshold evolved into the following process:

1. Find a red, blue or colored object.
2. Enter in a threshold for that color.
3. Hold it in front of the color sensor.
4. If nothing happened, increase the threshold by 15.
5. Hold another colored object in front of the sensor.
6. If nothing happened, hold the final color object in front of the sensor.
7. If the indicator light went off for either of the two other colors, increase the threshold by 50.

Even before we started testing with the color sensor to calibrate the thresholds, we wanted to ensure the syntax and behavior of our program was correct. The PICAXE microcontroller software includes a simulation mode that allows program code to be simulated. Each subroutine was tested to ensure that at no point there would be an unintentional infinite loop or no action being taken. On the next page is a simulation screen shot of the program test code.

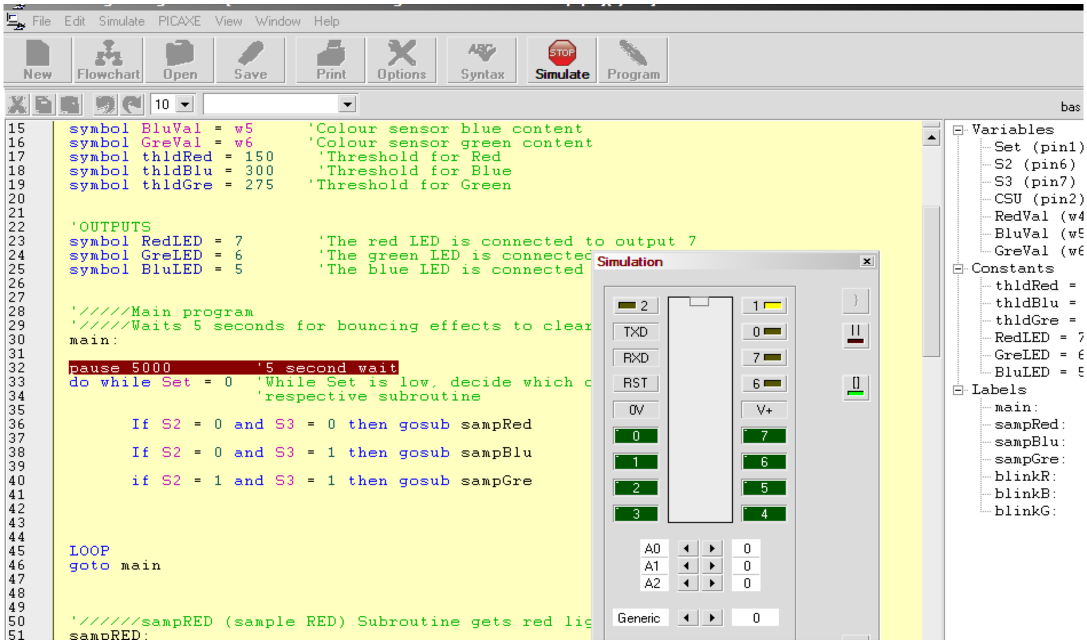


Figure 67: Screenshot of Program testing

Some pseudo-code would be:

DO WHILE SET = LOW
IF S3 = 0 and S2=0 THEN GO SAMPLE RED
.
.
.

SAMPLE RED:
COUNT PULSES ON INPUT 1 for 50 MS.
STORE NUMBER IN WORD4.
IF WORD4 >= REDTHRESHOLD, BLINK REDLED.
IF WORD4 < REDTHRESHOLD, GO TO SAMPLE RED.

This code is very simple in structure and easy to test.

7.5 Wireless Communication

Before the interpreter is put together, a test of the program in the microcontrollers is necessary. First, the interaction between the AT90S2313 MCU and the Playstation 2 controller is to be tested before everything else. The testing of the interaction between the transceivers and the AT90S2313 is to be done, in order to make sure that data traffic will be flowing smoothly between from the Interpreter circuit to the robot. Then, the communication between the PIC18F458 MCU and the transceivers has to be looked into to ensure the correct data flow from the robot to the interpreter. And, finally, the testing between the link between the LED Display box and the PIC18F458 MCU will be done.

The AT90S2313 Microcontroller and the Playstation 2 Controller

One port of the AT90S2313 microcontroller will be used to collect the data from and interact with the Playstation 2 controller. The data from the Playstation 2 controller is in 8-bit serial format. The microcontroller will organize the data and use another port to output the control data to serial in/parallel out shift register that will in turn output the data in a parallel 8-bit format just for the testing.

To check if the output matches the buttons being pressed on the Playstation 2 controller, 8 LEDs will be used. An example of this testing circuit is depicted in figure 63 below.

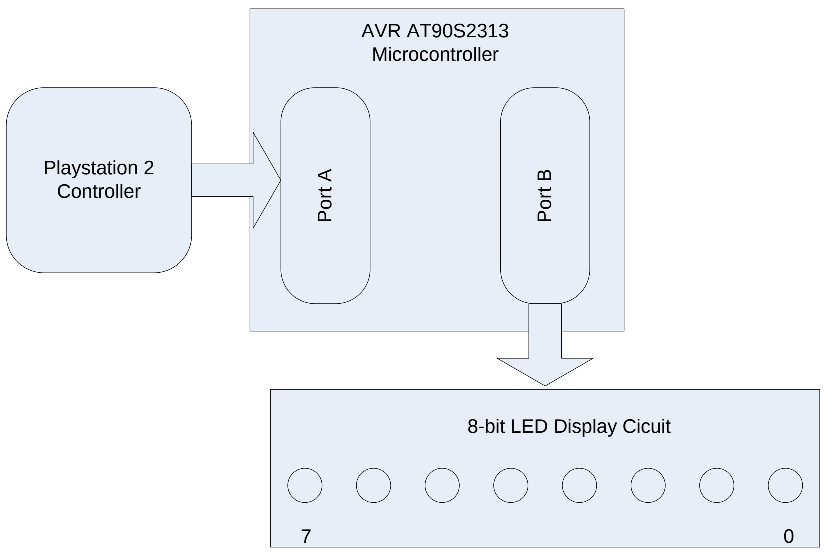


Figure 63: Block diagram of the test setup of the Playstation 2 Controller and the AT90S2313 microcontroller.

The transceivers and the AT90S2313 Microcontroller

To test the program in the transceivers, the Playstation 2 controller data from the AT90S2313 microcontroller will be used. The serial control data from the AT90S2313 will be collected by the transmitting transceiver, transmitted to the receiving transceiver (that would be on the robot). To verify the data received by the transceiver, a serial in/parallel out shift register will be used to convert the serial data output from the receiver to parallel format, which will then be displayed by LEDs. The process is depicted in figure 64 below:

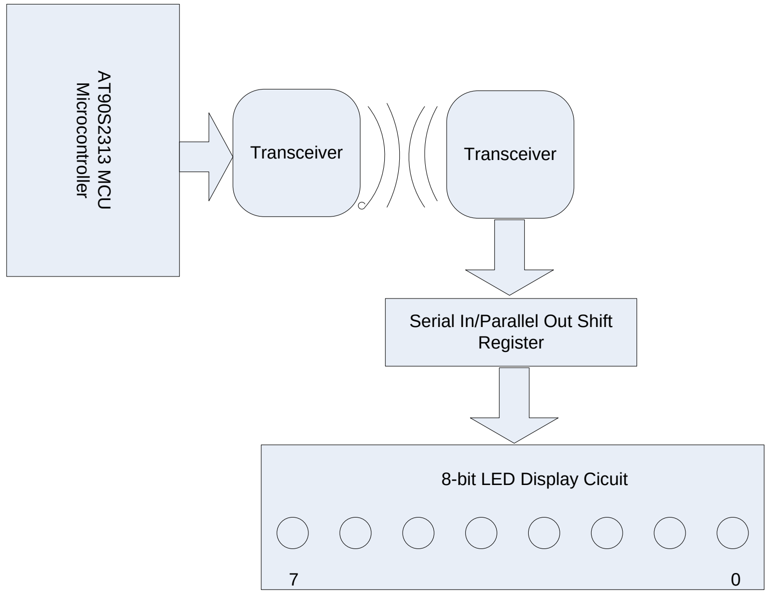


Figure 64: Block diagram of the test setup of the AT90S2313 and the transceivers.

The transceivers and the robot's processing unit

To test the transmission from the robot to the interpreter, the serial data from the robot's microcontroller will be collected by the transmitting transceiver, transmitted to the receiving transceiver (that would be on the Interpreter's circuit). To verify the data received by the transceiver, a serial in/parallel out shift register will be used to convert the serial data output from the receiver to parallel format, which will then be displayed by LEDs. The process is depicted in figure 65 below:

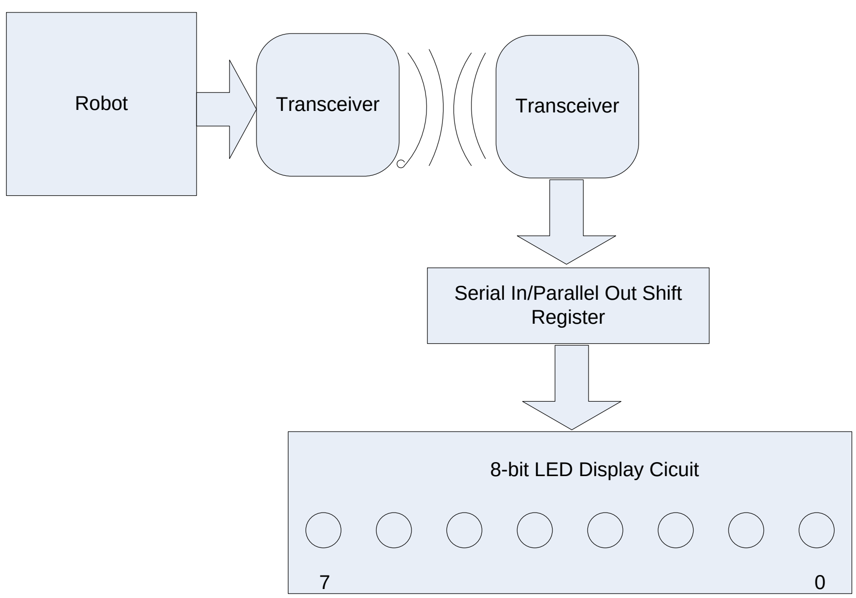
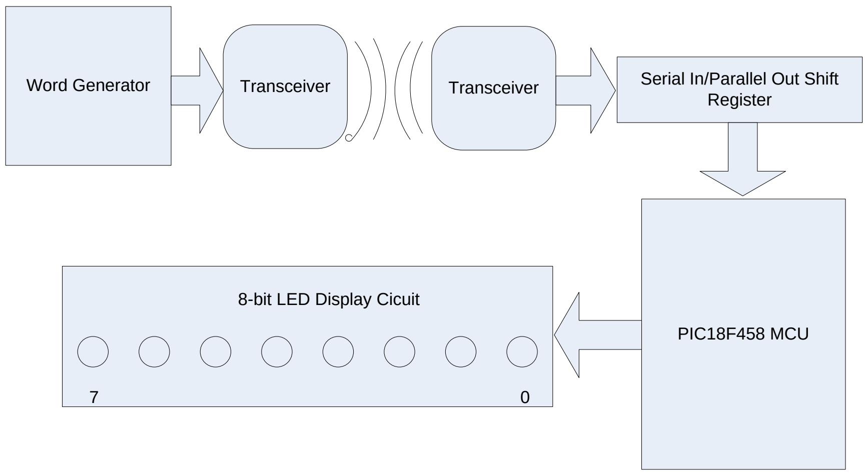


Figure 65: Block diagram of the test setup of the robot and the transceivers.

The transceivers and the PIC18F458 MCU

The information from the robot to the Interpreter would be contained in an 8-bit long word. The transceiver on the Interpreter will receive it from the transceiver on the robot, and transmit it to the PIC18F458 MCU which will process it and output the appropriate word that will display properly through the LED Display. Figure 66 shows a block diagram of this experiment is shown below:



Block 66: diagram of the test setup of the transceivers and the PIC18F458 MCU.

7.6 Wireless Communication and Drive and Vision

The first integration will be the communication and drive system. Since the communication and drive systems were shown to be working independently, this test will verify their capability to work together. The communication system will simply send commands to the drive system, and if the drive system continues to work as it was prior, then the communication and drive integration will be successful. If it is not working properly, then before looking at both the drive and communication systems individually, their integration will be analyzed. This is because the previous tests give the team a fair amount of confidence that both systems are working properly in isolation, thus the transition to having them both work in unison is what must be causing problems.

7.7 Power Simulations

Preliminary Circuit Simulations

In order to better illustrate the voltage drops for the different subsystems across the power circuit of the robot, preliminary simulations will be conducted. While these simulations are not the complete simulations of the robot's power supply circuit, they helped in the design of the best and efficient power supply circuit for the robot. The color sensor in robot will need around 5 volts in order to operate, the microcontroller will need 5 volts and the servos each will need between 4.8 and 6 volts in order to function. Figure 67 and Figure 68 below with their respective step down voltage regulators, show two preliminary circuit diagrams with the voltage drops for two different electronics

Also to ensure that the robot functions properly various tests are conducted, since the project is divided into different subsystems, each one of them have their own test procedures to make sure that the particular subsystem is working as it should be. Also it will be easier that way, because it would be a lot more complicated if the tests were to be conducted after the robot has been built, of course the robot as a whole is tested, but to make it easier every subsystem is tested first, that way after integration if the robot does not operate properly, it is easier to determine where the problem is and what should be done to correct it.

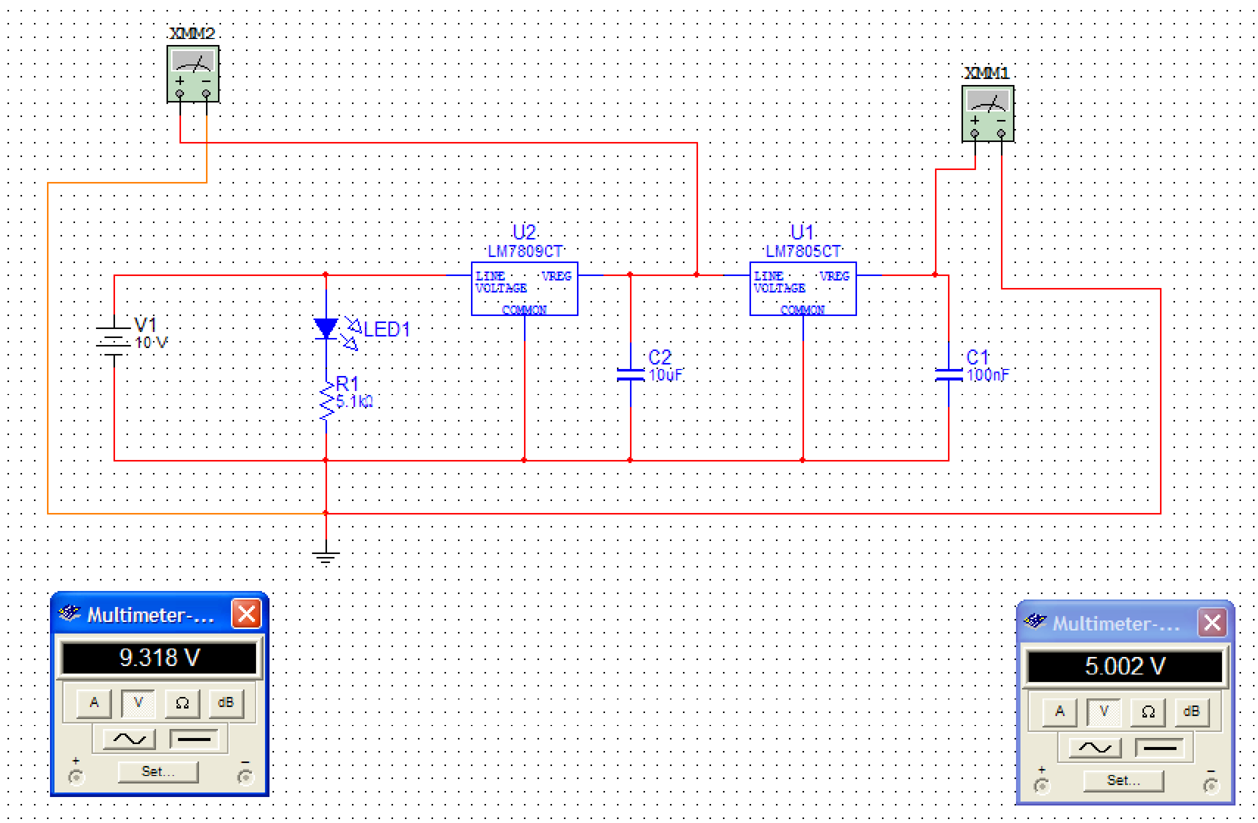


Figure 67: Circuit Diagram with the voltage drop for the camera and the microcontroller

LM 7805 Specifications

- Output voltage: 4.9V - 5.1V
- Output current : 5mA - 1A
- Drop Out Voltage: 2.0V

LM 7809 Specifications

- Output voltage: 8.82V – 9.16V
- Output current: 5mA – 1A

- Drop out voltage: 2.0V

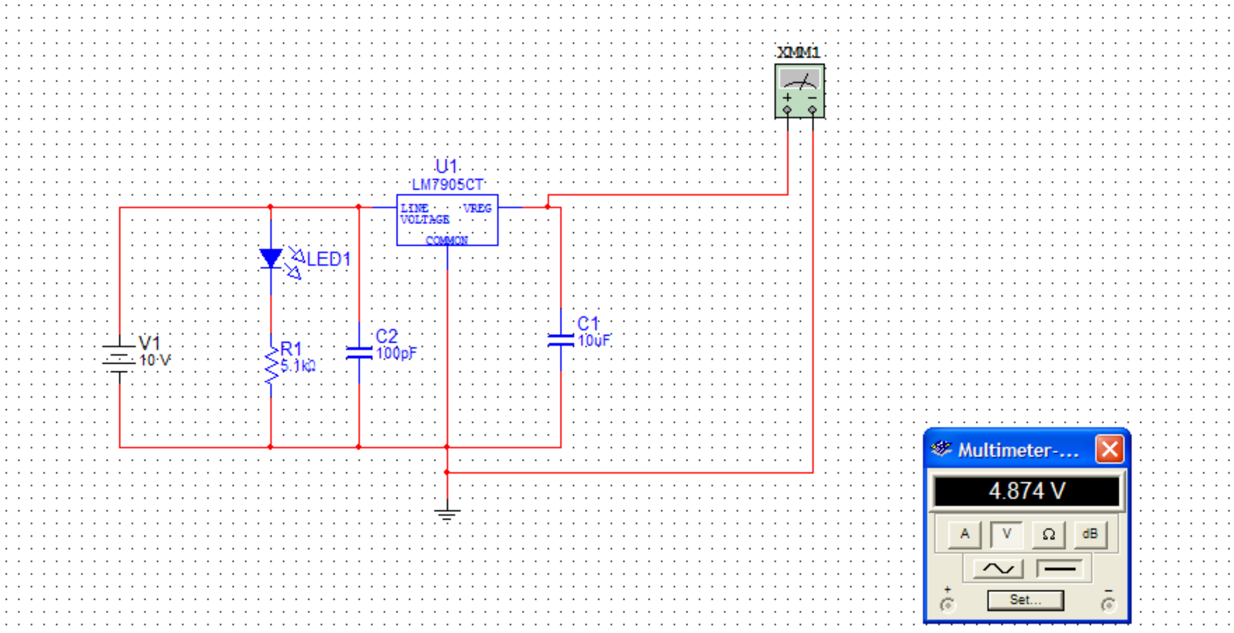


Figure 68: Circuit diagram with the voltage drop for the servo-motors

7.8 Final Tests

A thorough examination will be done confirming the proper functioning of all of the subsystems that make up the entire hexapod robot. In essence, it will be testing the vision, drive, and communication systems all together. If the robot is still receiving signals from the communication system and moving accordingly, then the drive system is working. However, the vision system must also be working concurrently and autonomously.

Therefore the robot will be made to walk around in an environment with objects of the colors specified (red, green, and blue) and depending on what user mode is an LED will turn on.

Section 8: Design Summary

Section 8.1: Mechanics and Drive

8.1.1 Legs and Body

The leg packages to be obtained are shown in Figure 69. There will be 6 of these legs. They will all be connected to the body chassis, shown in Figure 70.

Figure 69: Leg package (left) from LynxMotion (reprinted w/ permission)

Figure 70: Body chassis (right) from LynxMotion (reprintd w/ permission)

8.1.2 Drive hardware

There will be 18 servomotors for every joint, and they will be Hitec-475HB servomotors. They will be controlled using the SSC-32 servomotor controller. An Hitec-475HB servomotor is shown in Figure 71 and a SSC-32 servomotor controller is shown in Figure 72.

Figure 71: Hitec-475HB (left) from LynxMotion (reprinted w/ permission)

Figure 72: SSC-32 (right) from LynxMotion (reprinted w/ permission)

8.1.3 Drive software

The technique of inverse kinematics will be used to position the motors appropriately in order to create linear motion. In order to avoid any difficulty in syncing the different steps together, each step will occur a given value δ after the previous step. That is, the time step will be uniform across all steps. So if the walking sequence consists of 6 steps, then the first step will occur, and then the second step will occur δ afterwards. The value δ is in units of time, such as milliseconds. For the purpose of analysis, δ will be equal to zero ($\delta = 0$) instantaneously after a step, and will be negative before it. Thus, if it is necessary to begin a step before another step is finished, then δ will assume a negative value. It seems to be a reasonable assumption to make right now, that the time constant between steps, whether it be negative, zero, or positive, can stand to be uniform across all steps. This fact may be altered of course as building progresses.

In order to further reduce complexity, the same linear motion algorithm will be used in order to advance the robot in any direction. Due to the fact that the robot has a circular chassis, and that the legs will be connected with even spacing, the robot will be rotationally invariant. Therefore as long as the axes of linear movement are all equiangular, the same sequence of steps can be used to cause linear movement regardless of the direction given to the robot from the radio controller. Of course, if a command is given to the robot in the middle of its sequence, the walking algorithm must be able to either immediately reset to the beginning (standard position) or be able to transition smoothly into the next position. By having the robot only change directions on timed multiples of δ, then the amount of movement that the hexapod is doing should be minimized since it will either be undergoing zero movement, or be in the transition point between two steps, where one step will be finishing and the other step will be just beginning.

Since movement in every direction will consist of the same sequence of steps, all that remains is to identify the sequence of steps that compose movement. This problem will be simplified by considering the six legs as only three legs, where the two legs that are directly across each other will move in parallel. Whatever the front leg does, the back leg will do. The only difference will be that some of the final movements will be mirrored or inversed, or else one can imagine that one leg would try to move forward while the other leg would try to move backward. Therefore the same joint articulations will exist, but they will all be mirrored. Now since there will only in essence be three legs to worry about, it seems natural to have a sequence of either 3 or 6 steps as a full sequence executing linear motion; again with a time value of δ (which may be negative) between each step for syncing purposes.

Model Derivation

In the previous subsection, the analysis of the six legs was reduced to only three legs due to the properties of symmetry in movement; a form of coupling the legs in order to reduce complexity. Therefore, only three legs need be analyzed in any direction. By making the center of the directional heading on the middle of the three legs (front legs which are coupled to the back legs) to be analyzed, there exists symmetry between the left and the right legs. Therefore, all that needs to be done is derive a model for the left (or right) leg to move, mirror it on the opposing leg, and derive a model for the front leg. Thus the entire walking algorithm can be simplified to deriving suitable models for just a single front leg and a side leg.

Deriving a model for this type of situation is somewhat intuitive if one only considers the two legs at once; it seems like a hobbled human being. The side leg should raise its vertical hip position, move its hip position horizontally, extend its knee, and then lower its vertical hip position. This will be a side step forward. The front leg should then raise its vertical hip position, maintain its horizontal hip position since it is in the direction of movement, and extend its knee. This will be a step forward, and this step will return the side knee joint to its proper position. Then the alternate side leg will mirror the sequence of the side leg. This is the model derivation concept, where the actual angular values, and consequently servomotor positions, will be dependent upon the length of the legs themselves.

Program Execution

In order to better illustrate the execution of the gait algorithm, a block diagram was created. Figure 73 shows this block diagram. The controller will await an input, it will then decode the appropriate direction and decide which legs will be moved, and then goes through the sequence of moving each leg with a time step in between each. Afterwards, it again waits for an input signal to determine what its next direction will be.

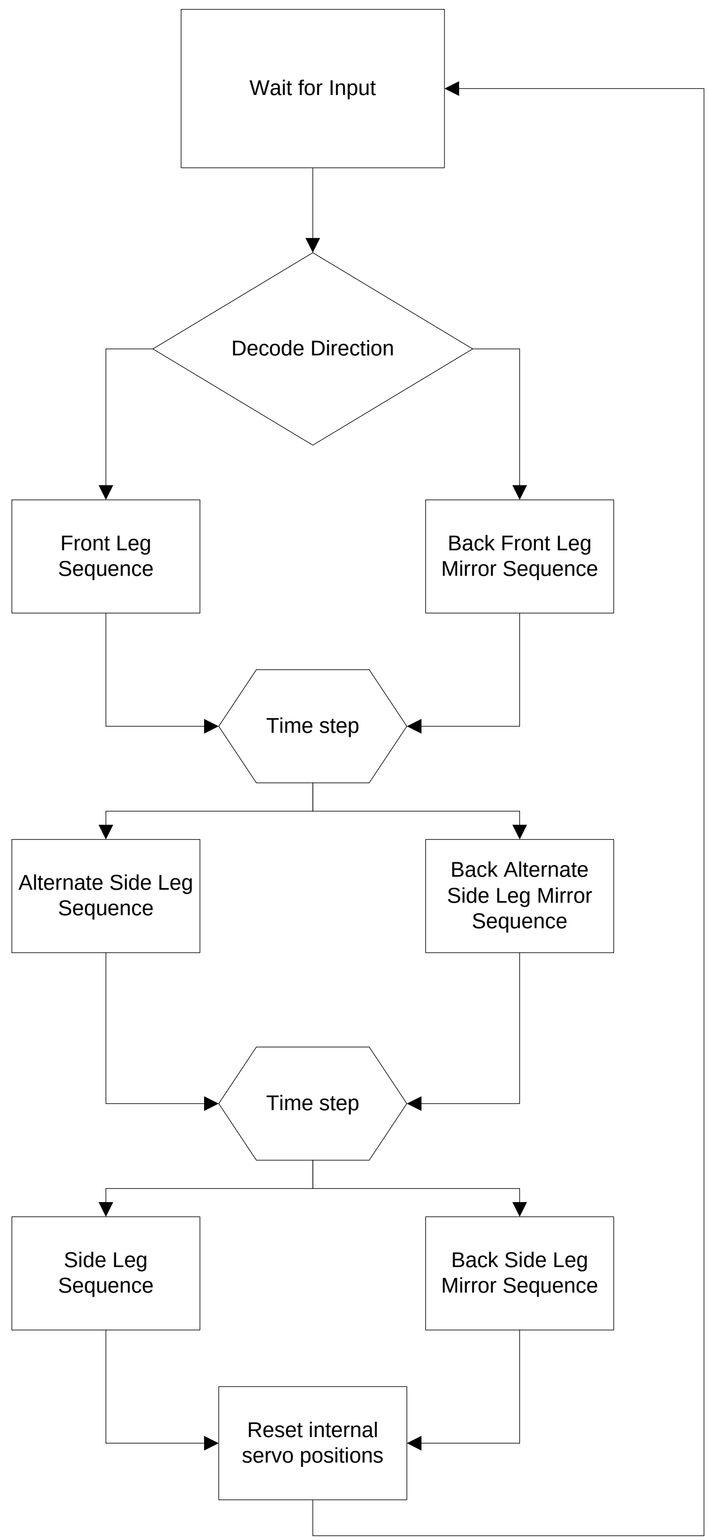
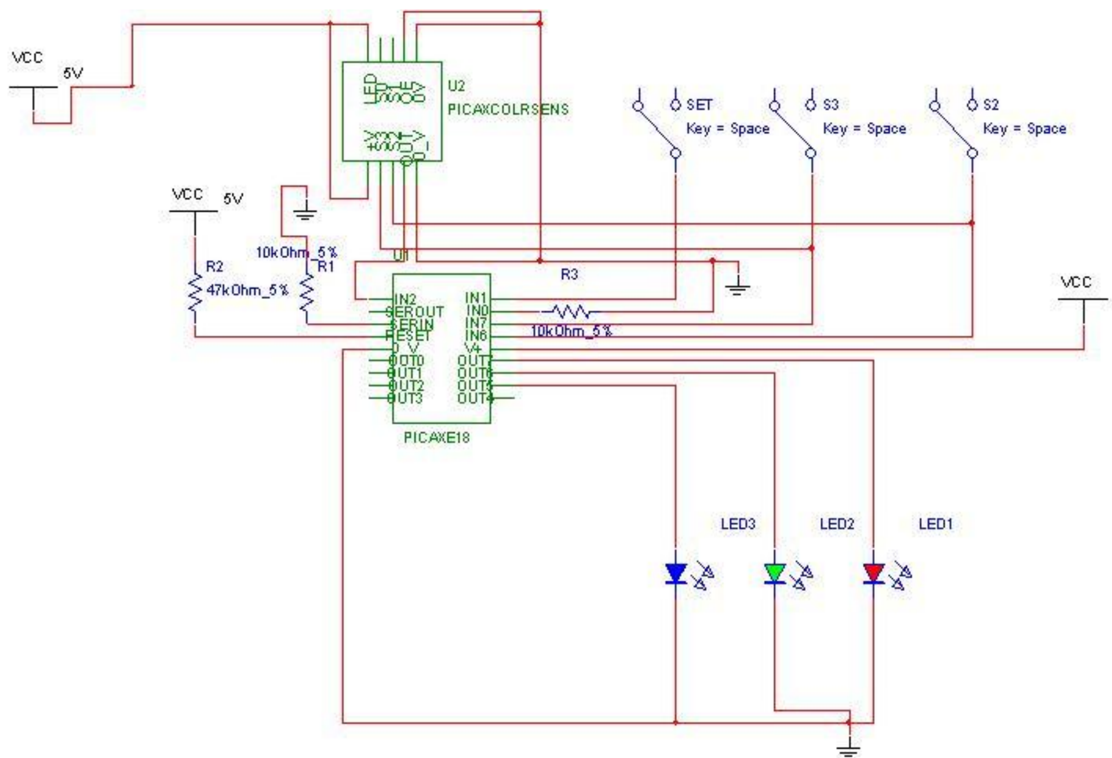


Figure 73: Block diagram of gait algorithm

Section 8.2: Color Sensing

The PICAXE Color Sensor module (a combination TAOS TSL230RD light-to-frequency converter and lens assembly) and PICAXE18X microcontroller will be our color sensing unit. The color sensor module illuminates the object with white light from 2 LEDs. The reflected light enters the lens and hits the TAOS TSL203RD. If the correct color filter is selected, square pulses will be generated. If the number of square pulses in the 50 ms sample time meets the threshold for that selected color, than an indicator LED will light.



Section 8.3: Wireless Communication

The robot is to be controlled wirelessly by a conventional Playstation 2 controller, that is, the version equipped with the cable. Using the Playstation 2 controller, the user will be able to control the movements, the gait, and the decisions of the robot. The Playstation 2 controller will be part of the wireless unit meant to handle the wireless function of the whole system. The robot will also send out to the wireless unit vision-related data and data pertaining to decisions made by the user. The wireless unit consists of the Interpreter circuit, two transceivers (one on the mobile unit with the user, the other on the robot), the Playstation 2 controller and the LED display system. It is depicted in the block diagram below.

The Interpreter circuitry is titled so because of one of its main task is to interpret the data coming sent out by the Playstation 2 controller through its cable. The Interpreter circuit's other task is to receive and process data transmitted by the robot and interpret them by displaying them accordingly through the LED display box. The Interpreter is built around two microcontrollers that are independent of each other and do not communicate with each other directly, i.e., there will be no direct data traffic between them, even though they will be part of the same unit and soldered on the same board. One of them is to interpret the data from the Playstation 2 controller. The relationship between this microcontroller and the Playstation 2 controller will be the same as that between the Playstation 2 console and the controller; hence, the microcontroller is to emulate the Playstation 2 console. It will receive the data from the controller, which transmits it in serial format, then process it and spit it to the transceiver, which will be on the same board.

The other microcontroller is to collect and organize the data sent out by the robot through the transceivers. This set of data is composed of data from the image processing unit of the robot and data concerning some decisions made by the user through the use of the Playstation 2 controller. After processing the data, the microcontroller will be programmed to display it explicitly to the user through a LED display, thereby informing the user on the status of the robot.

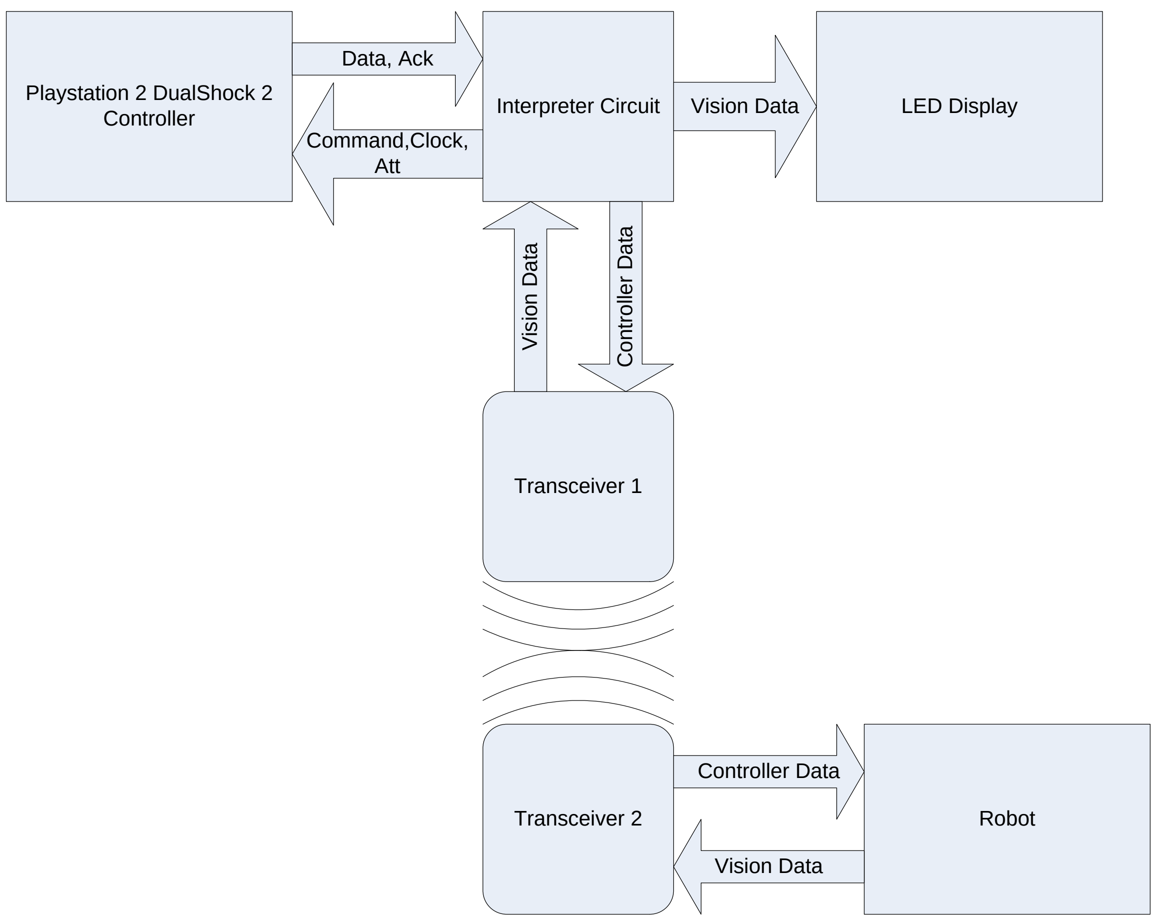


Figure 75: Generic block diagram of the wireless system.

Evidently, two transceivers will be used in the wireless unit of the system, allowing a two-way data transfer between the mobile unit and the robot; one (let it be labeled Transceiver 1) on the mobile unit, which consists of the Playstation 2 controller and the two microcontrollers covered above; the other (labeled Transceiver 2) on the robot. In sending mode, Transceiver 1collects and transmits to Transceiver 2 the parallel data supplied by the microcontroller handling the data from the Playstation 2 controller. In receiving mode, Transceiver 1 collects the data from the robot's processing unit transmitted by Transceiver 2, and supplies them to the microcontroller handling the LED display box.

In sending mode, Transceiver 2 collects the data from the image processing unit and the robot's processing unit, consolidates them then transmits them to Transceiver 1, which, in turn, sends them to the microcontroller handling the LED display box. In receiving mode, Transceiver 2 receives data sent out by Transceiver 1 from the microcontroller handling the interpretation of the data from the Playstation 2 controller, and then transmits them to the robot's processing unit which will then convert them into movements or decisions. The transceivers are to operate in the 2.4 GHz ISM band, and each of them will be equipped with a small compact chip antenna for the sake of size and mobility.

The LED display system will be made up of four differently colored LEDs. Three of them will be red, blue and green. They will inform the user which color is detected by the camera mounted on the robot. No more than one of these LEDs will be turned on simultaneously. Only one of them can be on. The LED display box will display one color at a time, as the camera and the image processing unit will be programmed to detect one color at a time. The fourth LED will be transparent (it will emit white light). It is on when the robot is hunting for a color, and it is off when the robot is on standby.

To summarize, the wireless unit is to permit a convenient two-way wireless communication between the robot and the user, who will wirelessly control the robot in its movements and decisions.

Section 8.4: Power

The design of the robot's power supply is as efficiently as possible to allow the main battery source to run as long as possible before needing recharge. The design of a power supply for a particular robot, must take into account these things and along with the consideration of cost, time, and space. The power supply of the robot meets the power requirement of every subsystem, and allow for future project expansion if needed, while the voltage regulators will efficiently maximize the robot's operating time. Calculations and results show the efficiency of the regulators efficiency and voltage at rated load. Discussions and research provided insight into need for appropriate component selections and placement, ensuring a proper and successful implementation of a DC power converter, and printed circuit board implementation.

The robot relies on a battery as its main power source. All batteries in general only produce a single voltage level, while the robot's subsystems require a diversity of voltage levels in order to function properly. Therefore the power supply system must be designed to provide all the subsystems with the necessary power that they require. Given that the main power supply of the robot is battery, it consists of a combination of DC to DC converters using linear or switching regulators, which help regulate the different voltage level that each subsystem requires. For efficiency, need and our budget, design considerations is taken into account in choosing the voltage regulators. Since the battery also powers the servos and motors, which can cause noise, therefore filtering and voltage isolation is advantageous, for the robot's computer system.

The robot's power system design is influenced by the different power requirement of its subsystems. Given that the power is a DC voltage source that only provides a constant voltage while the subsystem require different voltage levels in order to operate. Therefore voltage regulators are used to convert the high source voltage to lower voltages required by the different subsystems. A voltage regulator is device that accepts a DC input voltage and produces a DC output voltage. In general the output voltage is at a different voltage level than the input voltage. The input voltage may be lower or higher than the output voltage depending on what it is used for. When the input voltage is higher than the output voltage, the regulator is referred to as a step down voltage regulator. And accordingly, when the input voltage is lower than the output voltage, the regulator is referred to as a step up voltage regulator. For our robot, the source voltage will generally be higher than the required voltage of the different subsystems; therefore we will implement the use of step down voltage regulators. Figure 76 and Figure 77 below with their respective step down voltage regulators, show two preliminary circuit diagrams with the voltage drops for the different electronics, the microcontroller and the servo-motors, and figure 78 shows the final and complete circuit of the power suppl for the robot.

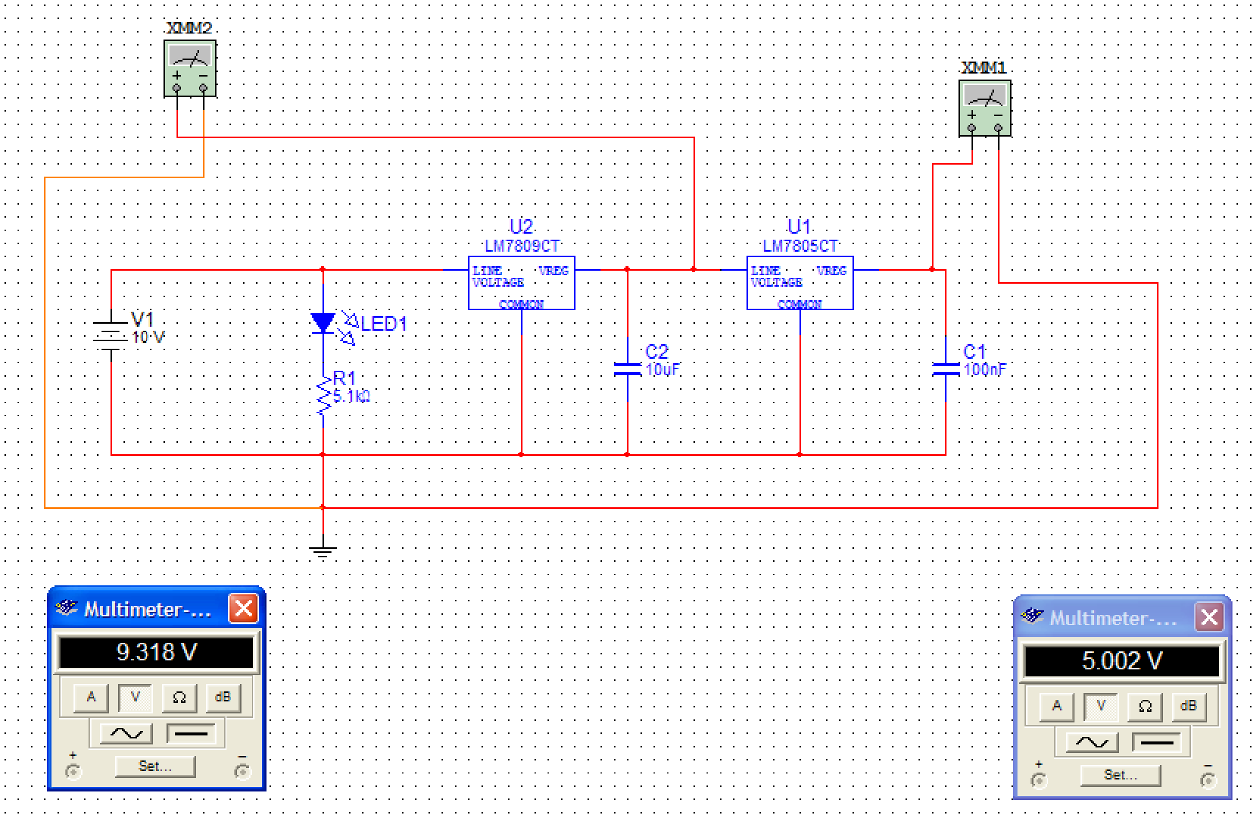


Figure 76: Circuit Diagram with the voltage drop for the camera and the microcontroller

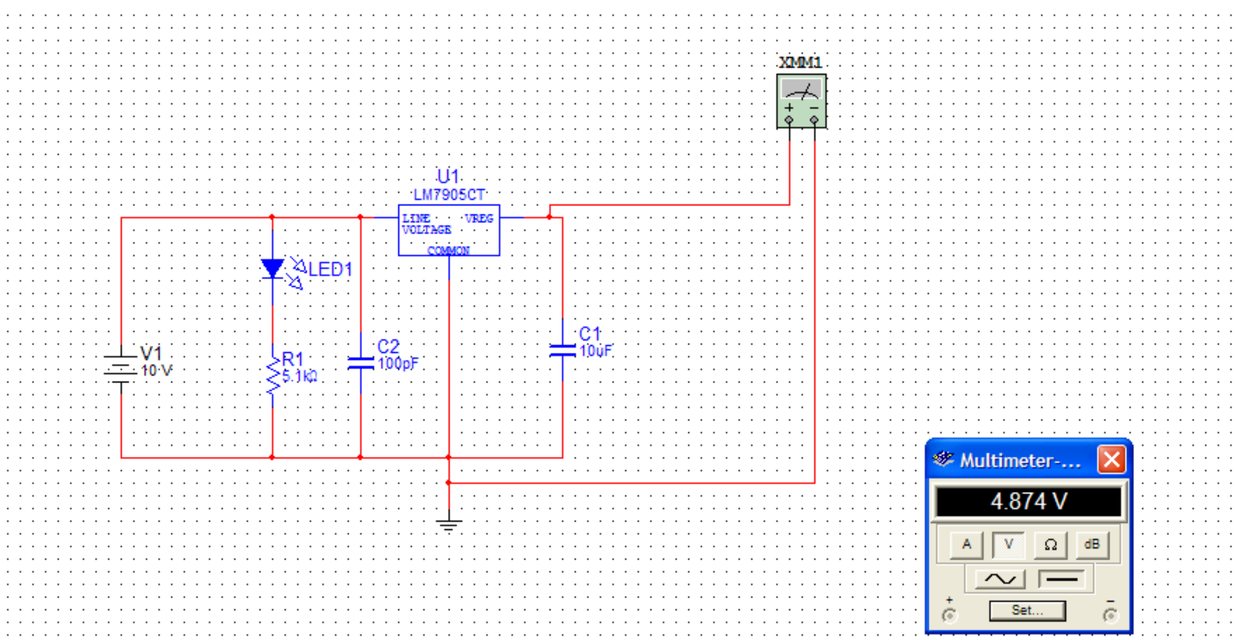


Figure 77: Circuit diagram with the voltage drop for the servo-motors

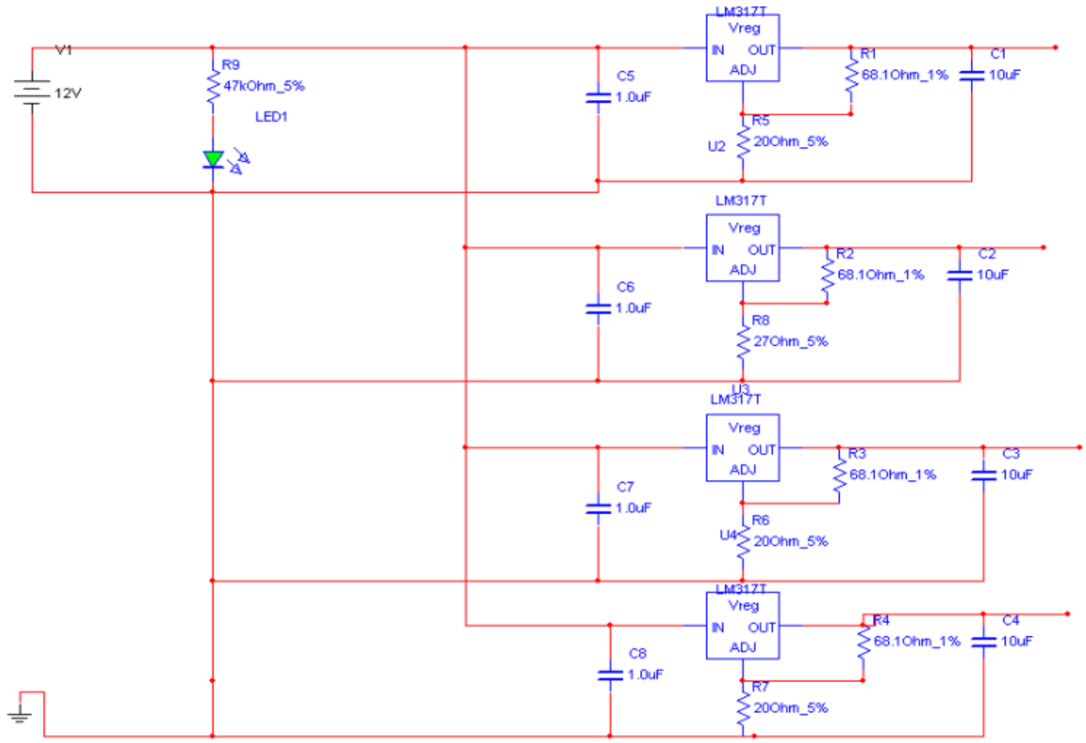


Figure 78: Complete circuit with the respective voltage drops.

Section 9: Administrative Content

9.1 Budget

The group is self sponsored with a projected amount of about $600. The cost of the project will be divided evenly among the group members. During testing procedures and while building the robot, components and parts may be damaged; therefore the budget may change accordingly to meet the needs and requirements of the project. Table 6 below show the necessary components needed and their price, in order to achieve a working prototype. Table 20 shows a preliminary cost for the project; it may be lower or higher depending on parts and components costs.

Components	Price	Quantity	Total Budget
Servos	$107.97	18	$323.82
Basic Atom Pro 28 Microcontroller	$60.00	1	$60.00
Xbee Module Adapter	$10.00	2	$20.00
Xbee Module	$20.00	2	$40.00
Color Sensors	$45.00	1	$45.00
LM317T	$.48	10	$4.8
Batteries	$22.95	2	$45.90
Chassis	$39.95	12	$39.95
Playstation 2 Controller	$15	1	$15
Charger		1	
ServocontrollerServos speed controller	$47.09	1	$47.09
Legs	$33	6	$219.05
Microcontroller	7.95	1	$20
PICAXE Color Sensor	$44.95	1	$39.99
		TOTAL (as of 12/8/08)	$534.33

Table 20: Number of components and their prices

9.2 Safety Tips

1. Over time, alkaline batteries are prone to leaking potassium hydroxide; which is a caustic agent that can cause respiratory, eye and skin irritation. This can be avoided by not mixing different battery types in the device, store the battery in a dry place, and removing the battery when the device is in storage. And also to reduce any hazards, proper disposal of the battery is recommended.

2. Lithium-ion batteries can rupture, ignite, or even explodes when expose to high temperature environments, therefore avoid exposing the battery to direct sunlight. Short-circuiting a lithium-ion battery can cause it to ignite or explode; therefore any attempt to open or modify the circuitry of the battery is dangerous and not recommended. The battery contains safety devices that protect the cells inside

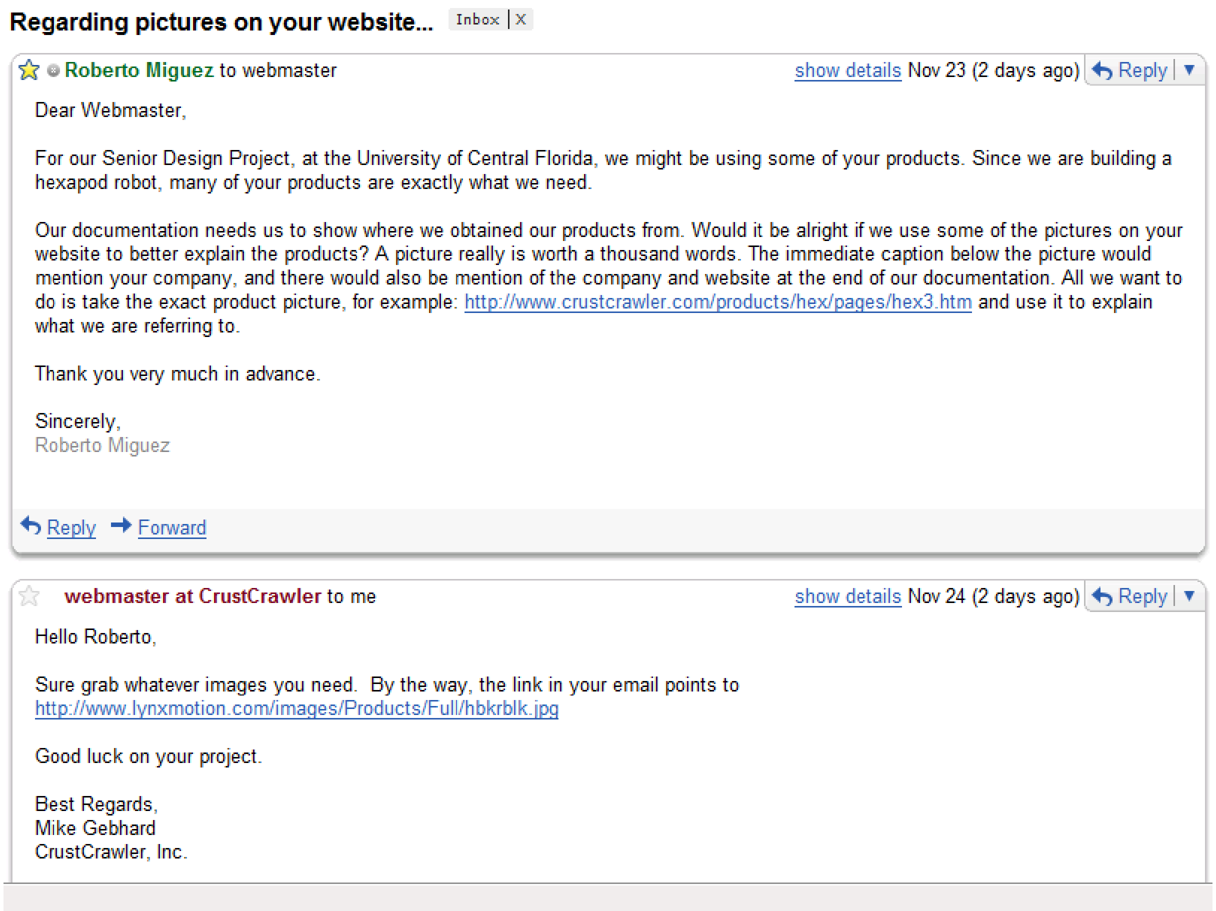
from abuse, and if damaged can cause the cells inside to ignite or explode. Proper disposal is recommended.

3. Nickel Cadmium contains cadmium, which is a toxic heavy metal and therefore requires special cares during disposal. Since cadmium is heavy metal, it can cause substantial pollution when land-filled or incinerated, because of this many countries now operate recycling programs to capture and reprocess Nickel Cadmium batteries. Never short-circuit a nickel cadmium battery, because it may cause it to explode. Never incinerate the battery, because it can explode and release a toxic gas that contains cadmium. Avoid dropping, hitting or denting the battery, because they may cause internal damage including short-circuit of the cell. Avoid rapid overcharging of the battery, because this may cause leakage of the electrolyte, out gassing, or possibly even an explosion. The safe temperature range for a nickel cadmium battery is between -20ºC and 45ºC. Proper disposal is recommended.

4. Nickel Metal Hydride batteries are safer when charged with a charger that has a resettable fuse in series with the battery, particularly of a bimetallic stripe. This fuse will open if either the current or the temperature of the battery goes too high. Modern nickel metal hydride batteries contain catalysts that immediately deal with the gases that develop as a result of over-charging without being harmed. However this only works with over-charging currents that are C/10 h (nominal capacity divided by 10 hours). As a result of this reaction, the battery will heat up considerably, which will end the charging process. Nickel Metal Hydride batteries are considered to commonly have lower environmental impact than nickel cadmium due to the absence of cadmium, which is a toxic material, however, proper disposal is advised.

Appendix A: Reprint Permissions

A.1 CrustCrawler Products Permission



Regarding pictures on your website... Inbox | X

Roberto Miguez to webmaster — show details Nov 23 (2 days ago) Reply

Dear Webmaster,

For our Senior Design Project, at the University of Central Florida, we might be using some of your products. Since we are building a hexapod robot, many of your products are exactly what we need.

Our documentation needs us to show where we obtained our products from. Would it be alright if we use some of the pictures on your website to better explain the products? A picture really is worth a thousand words. The immediate caption below the picture would mention your company, and there would also be mention of the company and website at the end of our documentation. All we want to do is take the exact product picture, for example: http://www.crustcrawler.com/products/hex/pages/hex3.htm and use it to explain what we are referring to.

Thank you very much in advance.

Sincerely,
Roberto Miguez

Reply Forward

webmaster at CrustCrawler to me — show details Nov 24 (2 days ago) Reply

Hello Roberto,

Sure grab whatever images you need. By the way, the link in your email points to
http://www.lynxmotion.com/images/Products/Full/hbkrblk.jpg

Good luck on your project.

Best Regards,
Mike Gebhard
CrustCrawler, Inc.

Figure 17 Permission from www.crustcrawler.com to use any of their images

A.2 LynxMotion Products Permission

Regarding pictures on your website... Inbox | X

Roberto Miguez to sales — show details Nov 23 (2 days ago) Reply

Dear Sales Staff,

For our Senior Design Project, at the University of Central Florida, we will be using some of your products. Since we are building a hexapod robot, many of your products are exactly what we need.

Our documentation needs us to show where we obtained our products from. Would it be alright if we use some of the pictures on your website to better explain the products? A picture really is worth a thousand words. The immediate caption below the picture would mention your company and website, and there would also be mention of the company at the end of our documentation. All we want to do is take the exact product picture, for example: http://www.lynxmotion.com/images/Products/Full/hbkrblk.jpg and use it to explain what we are referring to.

Thank you very much in advance.

Sincerely,
Roberto Miguez

Reply Forward

James Frye to me — show details 1:15 PM (2 hours ago) Reply

Hello Roberto,

Yes, feel free to use the pictures on our website in your documentation.

Thanks,
James.

Reply Reply all Forward Full view

Re: Picture usage Permission

From: **V. Scott Moore** (vsmoore2@comcast.net)
Sent: Wed 11/26/08 4:48 PM
To: A.J.N (a.j.normil@hotmail.com)

sure

scott
On Nov 26, 2008, at 12:04 AM, A.J. N wrote:

> Hello
>
> My name is Anselet Jacques, and I am a student at the University of Central Florida, while doing reaserch for my Senior Design Paper, I came across your website, I was wondering if I could use the pictures of the Linear and Switching Voltage Regulators on your website as part of my project. Your website will be sited as a refference for the pictures.
>
> Thank you for your anticipated co-operation

A.3 Microchip Permission

From: <Donna.Mason@microchip.com>
To: <ph940620@mail.ucf.edu>
Subject: RE: Authorization Request
Attachments: Mime.822 (7963 bytes) [View] [Save As]

You got it. Just follow the instruction in the information, you are good to go.

Donna Mason, Academic & Design Partner Programs
Microchip Technology Inc.
2355 W. Chandler Blvd.
Chandler, AZ 85224
(480) 792-7870

-----Original Message-----
From: Phillipe Jean-Jumeau [mailto:ph940620@mail.ucf.edu]
Sent: Friday, December 05, 2008 11:19 AM
To: Donna Mason - C08830
Subject: RE: Authorization Request

So I do have your permission, provided I write your product PIC18F458 MCU properly in the report?

A.4 Atmel Corporation Permission

From: "De Caro, Michael" <Michael.Decaro@atmel.com> Friday - December 5, 2008 6:01 PM
To: "Phillipe Jean-Jumeau" <ph940620@mail.ucf.edu>
Subject: RE: [Webmaster] Authorization Request
Attachments: Mime.822 (3341 bytes) [View] [Save As]

Phillipe,

The due diligence you have indicated in your e-mail is compliant with our policies regarding the re-use of Atmel-copyrighted material. You may do so accordingly.

Good luck with your design project.

Regards,
Michael

-----Original Message-----
From: webmaster-bounces@atmel.com [mailto:webmaster-bounces@atmel.com] On Behalf Of Phillipe Jean-Jumeau
Sent: Friday, December 05, 2008 9:44 AM
To: Webmaster@atmel.com
Subject: [Webmaster] Authorization Request

Dear Atmel Staff,

For our Senior Design Project, at the University of Central Florida, we will be using some of your products. Since we are building a hexapod robot, many of your products are exactly what we need.

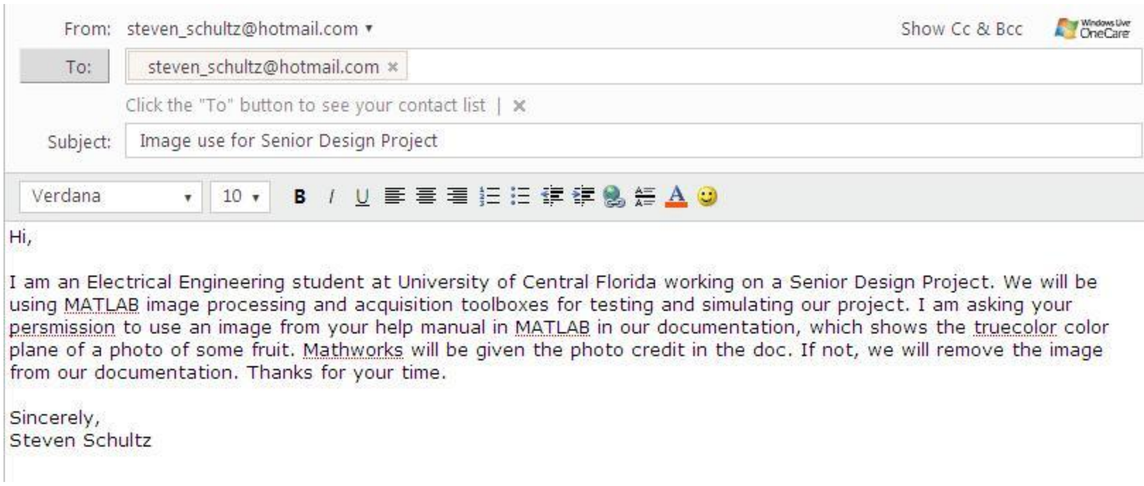
Our documentation needs us to show where we obtained our products and the information on them. In order to avoid copyright and other legal issues, we need your authorization to use pictures and some texts from your website and datasheets. The immediate caption below the picture will mention your company and website, and there would also be mention of your company at the end of our documentation. Please know that we will not make any monetary gains or other profits from this. The only purpose behind our project is to enable us to be evaluated academically by our academic coordinator.

A simple email response to this will be enough.

Thank you very much in advance.

--
Phillipe Jean-Jumeau
Electrical Engineering B.S.E.E., Undergraduate University of Central Florida

A.5 Mathworks Permission



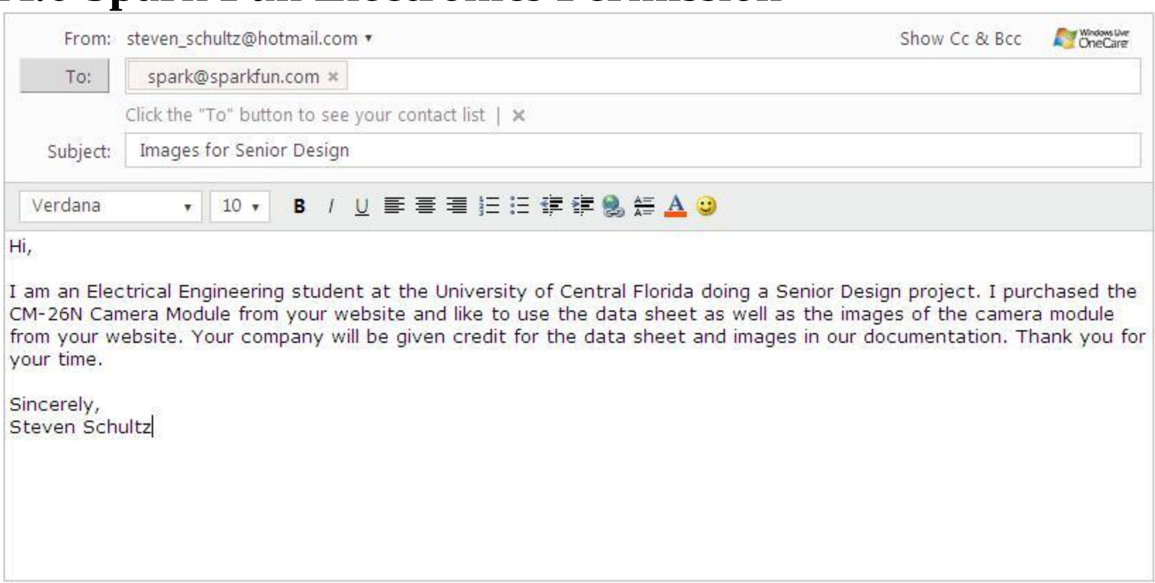
From: steven_schultz@hotmail.com
To: steven_schultz@hotmail.com
Subject: Image use for Senior Design Project

Hi,

I am an Electrical Engineering student at University of Central Florida working on a Senior Design Project. We will be using MATLAB image processing and acquisition toolboxes for testing and simulating our project. I am asking your persmission to use an image from your help manual in MATLAB in our documentation, which shows the truecolor color plane of a photo of some fruit. Mathworks will be given the photo credit in the doc. If not, we will remove the image from our documentation. Thanks for your time.

Sincerely,
Steven Schultz

A.6 Spark Fun Electronics Permission



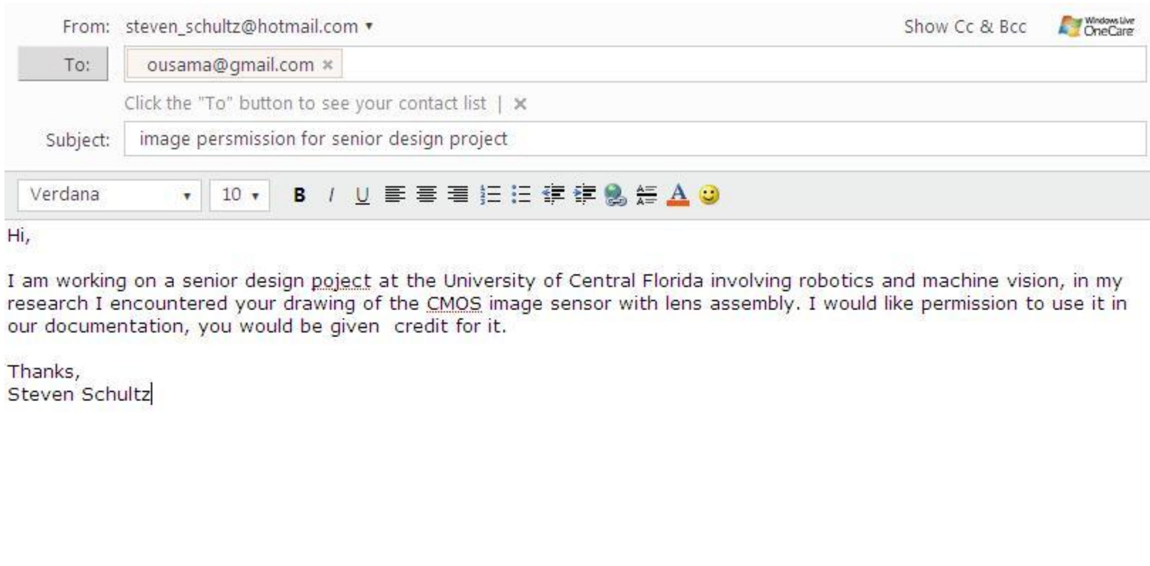
From: steven_schultz@hotmail.com
To: spark@sparkfun.com
Subject: Images for Senior Design

Hi,

I am an Electrical Engineering student at the University of Central Florida doing a Senior Design project. I purchased the CM-26N Camera Module from your website and like to use the data sheet as well as the images of the camera module from your website. Your company will be given credit for the data sheet and images in our documentation. Thank you for your time.

Sincerely,
Steven Schultz

A.7 CMOS Image Sensor (Figure 17)



From: steven_schultz@hotmail.com ▾ Show Cc & Bcc

To: ousama@gmail.com ×

Click the "To" button to see your contact list | ×

Subject: image persmission for senior design project

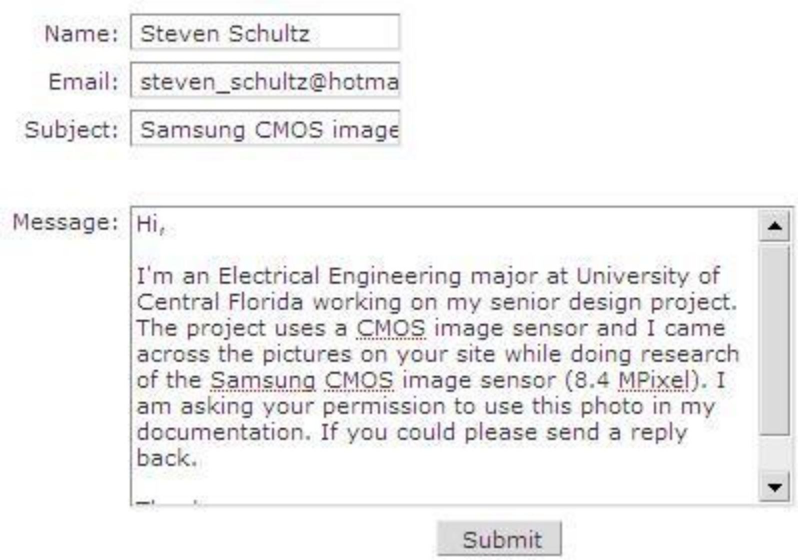
Verdana ▾ 10 ▾

Hi,

I am working on a senior design poject at the University of Central Florida involving robotics and machine vision, in my research I encountered your drawing of the CMOS image sensor with lens assembly. I would like permission to use it in our documentation, you would be given credit for it.

Thanks,
Steven Schultz

A.8 CMOS Image Sensor (Figure 18)



Name: Steven Schultz

Email: steven_schultz@hotma

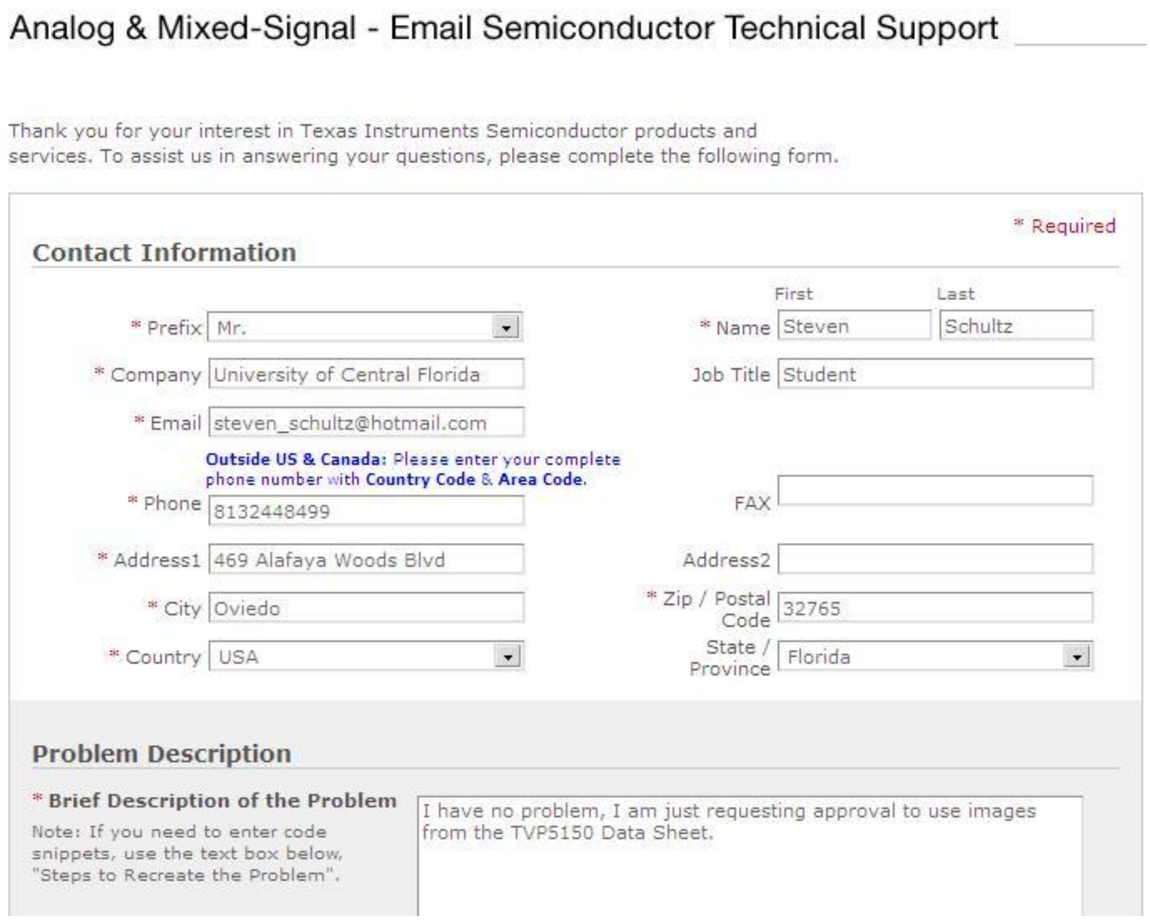
Subject: Samsung CMOS image

Message: Hi,

I'm an Electrical Engineering major at University of Central Florida working on my senior design project. The project uses a CMOS image sensor and I came across the pictures on your site while doing research of the Samsung CMOS image sensor (8.4 MPixel). I am asking your permission to use this photo in my documentation. If you could please send a reply back.

Submit

A.9 Texas Instruments TVP5150 Approval



Analog & Mixed-Signal - Email Semiconductor Technical Support

Thank you for your interest in Texas Instruments Semiconductor products and services. To assist us in answering your questions, please complete the following form.

* Required

Contact Information

* Prefix	Mr.	* Name	First: Steven / Last: Schultz
* Company	University of Central Florida	Job Title	Student
* Email	steven_schultz@hotmail.com		
* Phone	**Outside US & Canada:** Please enter your complete phone number with **Country Code** & **Area Code.** 8132448499	FAX	
* Address1	469 Alafaya Woods Blvd	Address2	
* City	Oviedo	* Zip / Postal Code	32765
* Country	USA	State / Province	Florida

Problem Description

*** Brief Description of the Problem**

Note: If you need to enter code snippets, use the text box below, "Steps to Recreate the Problem".

I have no problem, I am just requesting approval to use images from the TVP5150 Data Sheet.