Fall 2019

Generic Project Plan for a Mobile Robotics System

Jay Anilkumar Joshi
Western Kentucky University, jayanilkumar.joshi369@topper.wku.edu

GENERIC PROJECT PLAN FOR A MOBILE ROBOTICS SYSTEM

A Thesis
Presented to
The Faculty of the School of Engineering and Applied Sciences
Western Kentucky University
Bowling Green, Kentucky

In Partial Fulfillment
Of the Requirements for the Degree of
Master of Science

By
Jay Joshi

December 2019

GENERIC PROJECT PLAN FOR A MOBILE ROBOTICS SYSTEM

Date Recommended 11-4-2019

Dr. Mark Doggett, Director of Thesis

Dr. Mark Cambron

Dr. Mark Revels

Dean, The Graduate School | Date 11/19/19

ACKNOWLEDGEMENTS

I would like to express my gratitude to my thesis director, Dr. Mark Doggett, whose guidance, suggestions, and encouragement helped me during this research and thesis. I would like to thank my committee members, Dr. Mark Cambron, and Dr. Mark Revels, for their patience, support, and mentorship.

I would like to Paula Bowles for her assistance, that allowed me to get access to the resources from Western Kentucky University Library.

Finally, I would like to thank my family for their excellent support during my time as graduate student. I would like to thank my father who encouraged me a lot during thesis writing.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

GENERIC PROJECT PLAN FOR A MOBILE ROBOTICS SYSTEM

Jay Joshi December 2019 72 Pages

Directed by: Dr. Mark Doggett, Dr. Mark Cambron, and Dr. Mark Revels

School of Engineering and Applied Sciences Western Kentucky University

This thesis discussed the mobile land robots for the robotic competitions. The topics discussed in this thesis are robotic systems, mobile land robots, robot competitions, and example of robot designs. Question-answer sections are added to help understand the requirements to build the robot. Examples include three different teams who participated in different robotic competitions to provide a context for robotic competitions.

The thesis was divided into the five chapters. The first and second chapters explained the different kind of robotics systems, and opportunities. The focus of the information was the mobile land robots, which was explained under the third chapter, mobile land robots. The aim of the thesis was to guide those who want to design, build, and compete in the mobile robot competition. As a result, the information from various resources been gathered and has been given a form of thesis to help individuals or group of individuals to guide them through the robotic competitions.

Introduction

Robotics is a very broad subject. The purpose of a robot is to make human's work easier and safer. A robot is basically a utilitarian machine. "A discipline overlapping artificial intelligence and mechanical engineering. It is concerned with building robots: programmable devices consisting of mechanical actuators and sensory organs that are linked to a computer. The mechanical structure might involve manipulators, as in industrial robotics, or might concern the movement of the robot as a vehicle, as in mobile robotics. Robotics research is used in artificial intelligence as a framework for exploring key problems and techniques through a well-defined application" (Butterfield & Ngondi, 2016, p. 497). "The branch of robotics concerned with vehicular robot systems that are able to locomote within an environment or terrain. Mobile robots are used in research on navigation, exploration, and autonomous agents" (Butterfield, & Ngondi, 2016, p. 363).

As an interdisciplinary branch of engineering, robotics integrates aspects of mechanical engineering, electronic engineering, informatics and computer science in the design, construction and operation of robots in various industrial and scientific sectors of human activity. Robots are used in place of human labor for several reasons, including to access areas and carry out activities that are too dangerous for people as well as reduce the margin of error associated with people. Other reasons for installing robots are ensuring that unpleasant tasks are carried out and eliminating durational delays or costs associated with human necessities such as rest (Gerdes, 2014).

Even though they are an integral part of most industries today, robots were theoretical until 1961 when Joseph Engleberger and George Devol pioneered the first robot, the Unimate. Regarded as the father of robotics, Engleberger made Unimate and joined its first ever industrial line at a General Motors to work with heated die-casting machines, one of the most unpleasant work areas in the plant. In due time, the Unimate was joined to loading bays as well, and since then, the robotics industry has made massive strides and is applied in industry, science, research, medicine, entertainment as well as in the military sectors of various governments. In a modern world, the robot can be defined as, "an electromechanical device, which is capable of reacting in some way to its environment and take autonomous decisions or actions to achieve a specific task" (Rothschild, 2012, p. 17). There are many kinds of tasks a robot can perform and there are many kinds of robots one can prepare. However, this study concentrated on the mobile type land robots. As name indicates, mobile robots are robots that can move from one place to another. Moreover, a mobile robot can be controlled through a remote control or can be self-guided. A self-guided robot is called an autonomous robot. One should design a basic robot, and by making appropriate changes the same robot can be used for different type of competitions (Ruocco, 1987).

The Robotics Education & Competition Foundation (https://www.roboticseducation.org/) and FIRST Robotics Competition (https://www.firstinspires.org/robotics/frc) are some of the names of organizations that support robotics competitions for education. Moreover, universities and professional organizations also hold robotics competitions. To take a part in a competition, the robots must have certain specifications, which are provided by the organizer for the specific competition (Bateson, 1993). The specifications for a robot can be anything, such as maximum and minimum height, weight, length, width of a robot, and it varies for different competitions and different organizers. Participation in such a competition requires basic understanding of the fundamental systems of a robot. Resources are available to provide help in designing and making a robot, but to find information on mobile robots for a given robotics competition is not an easy task. Moreover, to gather information about how to make a robot, where to start, basic components, budget, programming, and implementation to make a robot is a daunting task (Nazaretov, and Kim, 1990).

The purpose of this thesis is to combine and condense current knowledge of mobile robotics systems to help practitioners gain understanding on the basic information about robots and provide practical examples of how to use this knowledge to make a robot for a competition. This effort was designed to help robotics students save time and effort when preparing for a mobile robotics competition. The material enclosed will help one to answer the question, is it feasible to design and develop a plan for a mobile robot at the system level?

As a beginner, one requires a basic knowledge of the platform of a mobile robotics system to take a part in robotics competition. One needs to know where to start and in which direction to move forward. This study will provide a basic knowledge and appropriate direction to get initial knowledge in mobile robotics system. The results of this research are generalized for a robotics competition. Robotic systems, mobile land robots, robot competitions, and examples of robot design are four major sections covered in this document. Robotic Systems provides a general overview of a robotic systems, and ideas on how to start to build a robot. Mobile Land Robots explains mobile type land robots, because these types of robots are required in most robotic competitions (https://www.roboticseducation.org/), (https://www.firstinspires.org/robotics/frc). Robot Competitions explains the general rules for robotics competition from the past. Examples of Robot Design discuses examples of robots designed for past robotic competitions.

Robotic Systems

The process of robot construction is divided into nine steps: defining the tasks of the robot, choosing a robotic platform, selecting actuators, selecting microcontrollers, motor controllers, the human-machine interface, power supplies, additional tools, and programming.

Categorizing robots

Robots are classified per their structure and composition as well as their areas of functionality and the form of movement and maneuver they are designed to use. For instance, robots can be classified as land type robots, air robots and underwater robots. Similarly, they can also be categorized as wheeled robots, tracked robots and leg type robots. Flying and swimming are also viable possibilities for recording and or classifying robot types. The classes of robots for this paper are limited to land, wheeled, tracked, leg, air-type and water-type robots.

Defining the tasks of the robot

It is recommended that one enhance the basic required knowledge about how to prepare a robot before starting. The first step is to think about the robot's task, the kind of robotic competition it will be, and the resources needed to get knowledge about robotics and robotic competition. This material will help one to build a customized land type robot. The first thing is to understand the task since the robot can perform various tasks depending on the design and programming (Nehmzow, 2009).

These factors can help students get an initial idea of how to design the robot. Such factors are the maximum and minimum speed of a robot, the total weight the robot will lift or carry, how much accuracy is required, how many directions the robot will need to travel, program length and a memory capacity of a microcontroller, initial cost, maintenance costs, and environmental factors, such as what kind of land the robot will travel on (Newman, 2006).

Choosing a robotic platform

The design of a robot starts with strategic vision, needs and specifications. The possibility of how many types of custom robots can be made is large. The understanding of the robotic platform helps to simplify the mission of making a robot (Colestock, 2005). The following are a few basic platforms for a robot. With good knowledge of different robotic platforms, one can change his or her robot from a land type to perform in water or or in air (D'azzo, & Houpis, 1995).

Land type robots.

Land robots are the simplest robots and consequently are the easiest to design. They operate on the surface of the earth and move using wheels, tracks or limbs. Given their simplicity, they are the most popular kind of robots since most human activities occur on the land. Manufacturing, production and observation are some of the events for which land robots are designed. There are some variations of land robots, and they are classified as wheeled, tracked and leg type robots. Specific examples of land robots include crewless ground vehicle, IRobot Warrior, planetary rovers, self-driving cars and an Energetically Autonomous Tactical Robot among others (Mihelj, et al., 2019).

Land robots can provide the best exposure to robotics systems at a low cost. Commonly, land robots are less complex in comparison to air and water type robots. The focus of robotic competitions is land type robots. The land type of robots can be further classified into four types: Wheeled, tracked, arm and gripper, and leg (Nocks, 2007).

Wheeled robot.

Wheels in robots are usually included when the robot is made for activities that require constant motion. Unlike people, the centers of gravity for robots remain unchanged even in motion, and therefore the wheels provide them faster movement with more straightforward and eloquent designs that are consequently cheaper to realize. Expert robot designers recommend the customization of wheeled robots to account for the specific situation of the robots that are unique to one's functionality; hence, the kinds of wheeled robots in use today include the fixed wheel, orientable wheel, ball wheel and Omni-wheeled robots (Gauthier, & Regnier, 2010).

To provide mobility to the robot, wheels are the best solution because they are easy to mount, least complex, and low cost. One can use any size of wheels available in the market, according to specifications provided for competition. One can design two, three, four or more wheeled robots according to the need. Four wheeled robots are the most popular, stable, and easy to control. Two wheeled robots can be prepared using a gyroscopic sensor, which provides stability to the robot. Adding such tools makes the robotic function complex. Hence, it is good idea to use four wheels. Use of multiple drive motors can decrease the slip and provide more grip and traction control, which means each motor drives individual wheel. (Colestock, 2005).

Tracked robot.

Also, land robots followed type robots are built for rough and uneven terrains and are often alternatively called thread robots. Unlike wheeled robots that use wheels, track robots apply skid steer drives, a modified concept of the differential drives, which has a track attached at its either side. The attached tracks are driven using separate motors that cause the robot to move by pushing the tracks in either one direction or in opposing directions. Combat military tanks, planetary rovers and desert track are some of the most common examples of tracked robots given that they regularly interact with uneven and untamed terrains (Hsiao, et al., 2015).

Continuous track, also called tank tread or caterpillar track, is a system of vehicle propulsion in which a continuous band of treads or track plates is driven by two or more wheels. In simple words, tracks are what tanks use as wheels. Tracks distribute the weight of the robot on the surface and increase the grip on the slippery surface such as sand, and mud. Tracks can be used to increase ground clearance. Moreover, one can decrease the number of motors used in the tracked robot propulsion system. On the other hand, tracks increase the mechanical complexity of the robot. A tracked robot can provide excellent grip in a robotics competition, especially in robot-war (Perkowitz, 2004).

Legs robot.

Even with great strides in the robotic sector, construction of legged robots is quite a challenge, one whose development is ongoing. The need for legged robots is brought about by the difficulties experienced by wheeled robots in accessing certain surfaces that could only be reached by human-like legs. Engineers have over time created robots with one, two, three, four and six legs, each with its unique posturing offering specific advantages. Hybrid automata could also be created by integrating wheels and legs in a robot to achieve more room for maneuvers within specific terrains and increase the workloads that the robot can be mandated to handle.

The legs robots are complicated in comparison with the wheeled robots and the tracked robots. To make the robot balanced is a difficult task with the leg type robot, but six legs can make the balancing task easy. The advantage of the leg type robot is that it can navigate through large obstacles because the motion of the robot is closer to organic natural motion. The legs type of robot can be more reliable in an obstacle type of robot race. On the other hand, this kind of robot can significantly increase the mechanical, electrical and coding complexity (Tadokoro, 2009).

Arm and gripper robots.

Factories use the arm or gripper type extensively since they can be used where the movement of the robot is limited. However, one can fix an arm on the mobile platform according to the need. One can easily build an arm type robot with less than three or four degrees of freedom (DOF). In simple words, DOF is the number of directions an arm can move. The cost of the robot can go higher with the higher lifting capacity and a higher number of DOF. By adding more DOF to an arm, one can get a complex motion. The design process requires three basic things; a hierarchy (in terms of DOF), number of arms, and a purpose for the end effector of the arm. An arm can have a maximum of six degrees of freedom for each linear, and rotational movement as shown in Figure 1 (Philipe, 1983, pp. 19-34).

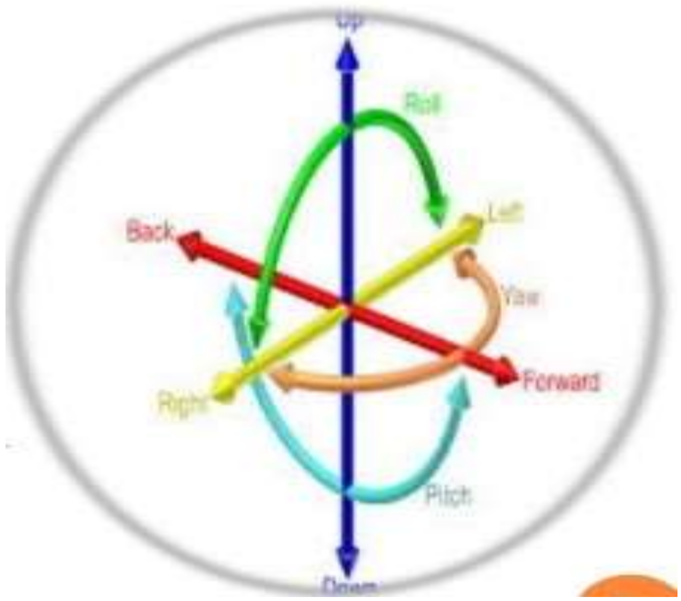


Figure 1. Degrees of freedom (Slideshare, 2018)

Air robots.

Air robots are conventionally known as flying robots and are categorized into four classes, namely: air balloon, rotary-winged, flapping wings and airplane robots. Usually, air type robots are expensive and complicated in design, meaning that large organizations and governments likely to afford bankroll their construction. Also, their functionalities are limited to observational roles, i.e. spying and research. Most air type robots are military creations by various governments or weather-oriented in that they observe the atmospheric conditions and analyze weather patterns. Missile detection and deployment systems are built of air type robots as are weather balloons, and thus they are the most relatable examples of air type robots (Cook, et al., 2018).

Air robots are also called as UAV (Unmanned Aerial Vehicle), and one can choose to build a wing type or copter type. An air robot that can perform a task autonomously is called as AUAV (Autonomous Unmanned Aerial Vehicle). One can build the copter type robot to reduce the cost, complexity and risk. One needs to understand that the entire investment can be lost in just one crash of the robot (Mason, 2001).

Water type robot.

Water type robots are almost always inclined to carry out observational and analytical duties underwater, often going to depths no human would dare reach. Oceanography, the study of oceans, has benefited tremendously from water type robots with notable mentions of robot involvement, including the transatlantic voyage made by Scarlet Knight, an American robot from New Jersey to Spain. Such robots are waterproof, contain high-level technological machines and maintain constant communication with observers on land. Apart from research, water robots are useful in rescue missions, intelligence gathering for militaries and helping people relax in leisure activities involving water bodies (Wilson, 2008).

Water type robots can be divided further into two types: floating type and diving type. The floating type of robots can have very restricted use while the diving type robots may provide more advantages over the floating type robots. The major drawback is that the robot can be lost in many ways such as sinking, internal leakage of the robot, or due to lost communication. The electronic parts must be covered in the watertight case because electronic parts could stop working after contact with water. As a robot gets deep in the water, wireless communication problems can arise. Hence, a powerful and long-range communication system is required for diving type robots. A cable communication and onboard power supply are preferable for diving type robots (Mathia, 2010).

Selection of actuators

An *actuator* can be defined as a device that converts energy into physical motion. In robotics, the energy tends to be electrical. The actuators make a rotational or linear motion. The AC/DC motor is the most commonly used actuator in robotics. Torque is a mechanical parameter to measure the force generated by an actuator, and the speed is a mechanical parameter to measure motion. The rotational speed is measured in RPM (rotation per minute), the linear speed is measured in m/s (meter/second), and torque is measured in N•m (Newton•meter). The encoder is a device that provides feedback to the processor about the position of the rotating shaft of the motor. The encoder can be connected to the motor through the feedback mechanism, or one can buy a motor with internal encoder. A potentiometer is a simple feedback mechanism that can be used instead of encoder when the budget is limited. The mobile robot requires an encoder when the application requires higher accuracy in robot movement. Generally, robots for competitions do not require encoders. The picture below shows the encoder for AC motors (Smith, Mcgrath, & Jason, 2013).

Figure 2. AC motor encoder (Indiamart, 2012).

The following questions can help one choose the right actuator. Is the actuator being used to move a wheeled robot? Is the motor being used to lift or turn a heavy weight? In degrees, how much is the range of motion? What is the specific angle needed to turn the robotic arm? Is the motion linear or rotational? In the subsequent paragraphs, a few of the different types of actuator motor are described (Masterson, Poe, & Fardo, 1985).

AC motors.

The AC motors come in a wide range of torque ratings, RPM ratings, and sizes. AC motors can be used when a higher torque is required and the application is stationary. AC motors can produce higher torque in comparison with DC motors. The motor producing higher torque may produce a limited amount of rotational speed. An AC motor can act as a generator while free spinning. In such a case, one can add flywheel or other electrical equipment that can absorb the extra electricity. One can convert the DC power into the AC power using DC to AC power converter and vice versa for onboard use (Smith et al., 2013).

Figure 3. AC motor (Zoro, 2011).

DC motors.

In most cases, mobile robots use DC motors and microcontrollers. Hence, use of a DC power supply is advisable. Choosing a DC motor is an easy task. DC motors are available in a variety of shapes sizes, torque ratings, and RPM ratings. DC motors provide a very high range of RPM, up to 10,000, and usually produce low torque in comparison with AC motors. One can add gears to increase the torque; or on the other hand, to decrease the speed (Menzel, & D'Aluisio, 2000).

Geared DC motor.

A geared DC motor is a DC motor with internal gearbox. A gearbox controls the speed of the DC motor to increase or decrease the torque. For example, if a gearbox is 200:1 type and the maximum torque and speed of the motor is 0.01 N•m (Newton • Meter) and 1000 RPM, respectively, then, at a speed of 1000 RPM, a motor is producing 0.01 N•m torque. Adding a gear down would reduce the speed 200 times and increase the torque 200 times. Thus, after adding a gear down (200:1):

Speed of a motor: 1000/200 = 5 RPM

Torque of a motor: 0.01•200 = 2 N•m

The most popular types of gearboxes are spur, planetary, and worm. Of these, the spur is the simplest and most popular gearbox (Colestock, 2005). The planetary gearbox allows the higher ratio with minimum gear down at higher efficiency. Worm gearboxes are expensive in comparison with other gearboxes, but provide a higher ratio with single gear stage down.

R/C servo motors.

When precise angular moment is required such as controlling a steering wheel or controlling a radar of an UAV, servo motors can be a good option. A servo motor comes with an internal encoder, which provides feedback of the angular moment. With the use of an internal encoder, a servo motor can provide precise feedback of motion to the microcontroller.

The use of encoder is significant in some robotic applications. The significance of an encoder is explained in the following. One can use different servo motors to drive different wheels. For example, if all the wheels are driven by different servo motors, the robot is four-wheel drive, and the robot is using two batteries. One battery is providing power to two servos on the left side, and another battery is providing power to two servos on the right side. In such a case, it is possible that one battery with the same electrical ratings is more electrically charged than others. Hence, one side of the servos will get more electrical power than the other side. In this condition, the robot will not move in a straight line. Hence, coordination between all four servos is necessary. If the servos are not coordinating with each other in a proper way, there is a chance the robot will not move in a straight line. The feedback of the motion is very important in order to keep the robot moving at the same speed in the desired direction. The encoder will provide the feedback of the servo speed, and the shaft position, which will help all four servos to keep operating at the same speed (McComb, 2011).

The servo motor with an internal break helps in speed control. The servo motor running at high speed can be precisely stopped at the desired set point with the help of the internal encoder and break. Moreover, the internal break can be used to generate the drag, so the wheel will not free spin. To stop a servo motor at desired set point, the break and encoder work together. The price of the servo motor is comparatively higher than other types of motors because of its advantages (Smith et al., 2013). The servos have two wires, one for the power, and a second wire for the encoder. The servo motors are useful in arm or gripper type robots, where a precise angular moment of the motor plays an important part in the movement of the robot. AC powered servo motors are useful for industrial use, and a DC powered servo motors are useful for mobile applications.

Linear actuator.

The linear actuator moves in a straight line. The linear actuator works the same as the stroke of a gasoline engine. The linear actuator has three characteristics: the stroke, the force and the speed. Stroke is measured in millimeters and linear speed in m/s. Use of a linear actuator is a good option to fire a hammer as a weapon, for a robotic-war competition (Moallen, Patel, & Khorasani, 2000). An example of a hammer as a weapon in the mobile robotic application is shown in Figure 4.

Figure 4. Use of a linear actuator to provide movement to hammer (Robot war wiki, 2016)

Computing the motor requirement.

The following questions and answers below are designed to help individuals to understand the motor requirements. One can determine the motor requirements for the robot competition by answering the following questions (Rahimi, & Karwowski, 1992).

1. What will be the function (direct use) of the motor?

The motor can be used to do different things, such as to propel the robot, to move the arm, or as a steering mechanism to turn the robot. If the motor is being used to move the wheels or tracks, the motor must be capable of moving the entire mobile robot with maximum weight. Moreover, one should keep in mind that the driving motor should have extra capacity to handle an extended weight or slope. A DC motor is a good option for the propulsion application. The servo motor or linear actuator is a good option for an arm or gripper type robot depending on the specific application.

Providing the power only to the one side of track or wheels can turn the position of the robot. For example, providing the power only to the wheels on the right side of the robot will turn the robot left. This simple steering mechanism called skid steering, which provides good control to turn the robot. The servo motor is a good choice for steering operations. Moreover, by turning one side of wheel pairs at a different speed, the robot turns by skidding. This means one side of wheel pairs do not have to be fully stopped during the steering operation.

2. Will the robot be caring or turning heavy weights?

One should consider using high torque motors when it comes to moving or lifting heavy weight. The robot needs more power in lifting a weight, in comparison with moving the same amount of weight. If one is making a robot that will lift the weight through arm or gripper, one needs to make sure that the arm will not drop the weight in case of a power outage. The arm should stay in the current position when the lifting procedure is paused or stopped. Any motor with an internal break is a good option.

3. What is the range (in degrees) of the motion of the motor shaft, and does the operation require a precise angular moment?

The servo motor can provide good feedback when precise angular or linear moment is required.

4. What is the type of motion, linear or angular?

For an angular moment, one can choose a DC motor, AC motor, servo motor, geared motor, or stepper motor. For a linear moment, one can choose a linear actuator. One should consider looking in the market for the latest motors.

The consideration of the following factors can be good practice to determine the motor requirement: Total mass of robot (g), number of driver motors, radius of drive (in), velocity of robot (m/s), maximum incline (degree), supply voltage (V), desired acceleration (m/s^2), Desired operating time (s), velocity of motor(rpm), torque (N•m), total power (W), maximum current (A), and battery pack (AH) (Hunt, 1985).

Selecting a Microcontroller

The microcontroller is the brain of the robot. A microcontroller is responsible for decision making, communications, and computations. The Arduino Uno R3 USB Microcontroller, Arduino Due 32bit ARM Microcontroller, and Intel 8051 are popular microcontrollers. (Dumouchel, & Damiano, 2017). The microcontroller possesses a set of pins (legs), which can be turned high (1) or low (0), by a set of instructions (program). The microcontroller can be used to control electrical devices such as actuators, storage devices, WiFi or Bluetooth. Home appliances such as a TV, refrigerator, remote controller, and digital watch are few examples that use the microcontroller. Advanced microcontroller comes with onboard storage, RAM, internal BUSs to transfer a data from, clock, contour, timer, and frequency generator.

A microcontroller can be programmed for simple to very complex tasks. A few things to consider about choosing microcontroller are budget, speed of the microcontroller, how many devices the microcontroller will be handling, and how many tasks each device will execute. For example, consider an LED (light emitting diode) connected to the microcontroller. A simple program can be created to toggle an LED. The programmer can set a specific pin of the microcontroller high or low for a specific time that connects to the LED. For the time period the pin is set high, an LED will glow. When the pin is set to low, the LED will not glow. It is important to know how much voltage and current are associated with each pin of the microcontroller before connecting any device directly to the microcontroller. In the above case, the LED is directly is connected to the microcontroller. One should refer to the manufacturer’s manual to know how much current and voltage are associated with each pin of the microcontroller. The task of creating a program for time delay may vary for different microcontrollers, the internal clock of the microcontroller, an operating frequency of the microcontroller, and if the controller is 8-bit, 16-bit or 32-bit type. Some microcontrollers provide a facility to connect an external clock to the controller. Consider that microcontroller has very low output voltage and current, which is not enough to drive the devices, such as a motor. Hence, microcontroller requires an additional system (motor controller) that can drive a motor according to the program or live control (Nehaniv , & Dautenhahn, 2007).

The advanced microcontroller can perform advanced tasks such as communicating with other microcontrollers and computers because of built-in hardware support system called UART. UART is a universal asynchronous receiver-transmitter. An RS-232 cable is an example of a UART serial communication system. The microcontroller understands only binary language also called as binary bits. The microcontroller comes with a built-in compiler that converts the programing language in binary bits. The microcontroller converts the binary bit into the nearest equivalent voltage or current (Launius, & Mccurdy, 2008).

An advanced computer uses a microprocessor as the CPU (central processing unit). A microprocessor is more advanced than the microcontroller in many ways. One can choose to use computers for stationary robotic systems. For mobile robotic applications, computers are not much help because computers are more expensive, heavy, need external peripherals, and need more power in comparison with the microcontroller. Hence, the microcontroller is a better choice to use in mobile robot application in comparison with the computer microprocessor (Kalften, Chmielewski, & Negin, 1989). The comparison of a microcontroller and a computer microprocessor is shown in Table 1.

Table 1

Comparison of a computer and microcontroller

	Computer Microprocessor	**Microcontroller**
Name	Intel core i7	Atmega 328
Price	About $1000	About $100
RAM	8 GB	1 KB
Storage	500 GB	15 KB
Power	600 W	0.1 W

Voltage	12	12
I/O	USB	Pins
Wireless communication	Bluetooth	Bluetooth and RF
Graphics card	2 GB	N/A
Internet	WiFi, Ethernet	WiFi, Ethernet

Selecting the microcontroller is not an easy task. It is wise to keep the combined cost of the microcontroller and development board as low as possible. With the use of *pulse width modulation* (PWM), it is possible to control the speed of a motor. Understanding and answering the following questions will help to choose the appropriate microcontroller (Astolfo, Ferrari, Ferrari, 2007).

1. What is the application of the robot; mobile or stationary? What type of robot is it; land, water, or air?

For the stationary robotic system, one can use the computer or microprocessor instead of microcontroller.

2. Which and how many peripherals will the robot use? e.g., the number of motors, weapons, or sensors.
3. Will a microcontroller support all the possible future upgrades? Is it possible to include more motors in the future, or replace the motor with higher torque or RPM?
4. Are guidelines and related documents easily available? Since programming is an important task, it is necessary to make sure that there are enough documents available from the manufacturer to use the microcontroller efficiently.

5. Is it possible to use a specific microcontroller with the development board? Testing the microcontroller on the development board is a good idea before using it with the robot.

Motor controller

The microcontroller determines the speed and direction of a motor. A motor cannot be connected directly to the microcontroller. A microcontroller operates on low power, which is not enough to drive a motor. It is necessary to connect a device between the microcontroller and motor that can understand the commands of the microcontroller and drive power from the battery and deliver it to the motor. Such a device is the motor controller (Engelberger, 1989).

There are a variety of motors available in the market that operate on a wide voltage range. A motor needs a minimum amount of power to rotate continually at maximum load. It is necessary to select the right motor controller that should be able to supply the minimum amount of power to all the motors to make continuous rotation at maximum load. The important thing to consider is that different motors may operate at different power. It is advised to use a minimum number of motor controllers for the mobile robot. Hence, it is necessary to choose the right motor controller that can power as many motors as possible, according to the operation (Selig, 2000).

There are a few control methods one should be aware of to control the motor speed, such as PWD, voltage control, I^2C, R/C, and UART. Before considering the above control methods, it is necessary to make sure which type of pin is available on the microcontroller. For PWD method, each motor requires a separate channel. A motor controller should be able to control the direction and the speed of a motor. One must consider that not all the motor controllers provides this function (Wilson, 2008).

The human-machine interface

The robot can be controlled by wired or wireless human-machine interface systems. Wired remote control and wired computer-control are two types of wired human-machine interfaces. Wifi, Bluetooth, infrared, and radiofrequency (RF) are examples of wireless human-machine interfaces.

Wired.

Wired remote-control.

Wired remote control is the simplest way to control the mobile robot. By the use of wired remote-control, motors can be connected directly to the remote-control switches and the switches can be connected to a power source. By using such a mechanism, one can avoid the use of a microcontroller and motor controller. Avoiding the use of a microcontroller and the motor controller decreases the complexity of robot design (Hartley, 1983). The wired remote-control interface system is a good option for individuals who want to build the robot for the first time. The moving robot can cause a loose connection of wire. Hence, with the wired remote-control system, it is important to make sure that wires are properly connected with motors, remote-control switches, and power source. With the use of the wired remote-control system, it is possible to use an off-board power supply. The off-board power supply helps to reduce the weight of the robot and provides an option to use the AC power supply. The use of AC power supply opens the way for the use high-power AC motors. The wired remote-control is a good option if the range of the robot operation is within a few meters. (Hall, & Hall, 1985).

Wired computer-control.

Tera Team is the software to control the robot. The website (www.instructables.com/id/DIY-Laptop-Controlled-Robot/) has an example of a computer controlled robot. The website mentioned above provides an example to make a simple robot and write a program to control it. It also provides the guidelines to connect the robot with a computer or laptop. Tera Team also provides an option to connect the robot with a computer by using Bluetooth. The computer-based robot control system may cost higher in comparison with the remote-control system. Tera Team is open source, free software, which provides the C++ programing language platform (Klancar et al., 2017).

Wireless.

Infrared.

An infrared remote control is a low-cost option of the robotics control system. The robot is required to stay in the visible sight of the remote control. An infrared remote cannot transmit or receive data if a receiver is not within visible sight (Cook, 2010).

Radio frequency (RF).

There are many different types of RF remote-controls available in the market. RF remote-control is divided into two parts, a transmitter and a receiver unit. A transmitter unit goes to the remote control and a receiver goes to the robot. A transmitter encodes data and transmits through an RF channel. On the other hand, a receiver receives an encoded RF signal and decodes the signal. A receiver communicates with the microcontroller through I/O port. Some RF controllers are available with the trans-receivers. The trans-receiver can transmit and receive a signal with the same antenna. The trans-receiver provides a feature of receiving feedback from the robot, such as the position of the robot. An RF remote-control system can transmit over the range of a few kilometers (Cook, 2010).

Bluetooth.

A Bluetooth uses microwaves to transmit a signal. A function of a Bluetooth control system is the same as RF control system. A Bluetooth has a very short range (about 25 meters), which is a disadvantag of this type of control system. Moreover, Bluetooth devices use a specific protocol through which a robot is required to be connected with the remote-control (Gerdes, 2014).

WiFi.

With the use of the WiFi control system, one can control the robot from anywhere Internet connectivity is available. The robot is required to stay connected with WiFi all the time if the robot operator is at distance place. With a WiFi control system, the robot can connect with a WiFi router, and a router requires an Internet connection. A WiFi control system will require the computer-based robot control system to control the robot as explained under the section "wired computer-control". With the use of Wifi, this system will work as a wireless computer-control system. (Gerdes, 2014).

Power supply

The onboard rechargeable battery is a good option for a mobile robot power supply. Different peripherals of the robot may operate at different power ratings. Hence, providing a power supply to all the peripherals of a robot with a single battery may not be a good idea. Selecting multiple batteries can solve the above problem. Consider that different batteries may stop working at different times. The microcontrollers and actuators usually operate at the 9V-12V range and most sensors operate at 5V depending upon the brand of the component (Colestock, 2005). One can choose a battery for a motor according to the task of the robot. For robot racing, high RPM motors are useful. For robot wars, high RPM and high torque motors are useful and the robot may have weapons. Hence, high-power batteries may be a good option for robotic war.

Additional tools

Wires, soldering connectors, diodes, resistors, switches, and capacitors, are important additional components. The robot may require more components according to the requirement. The following sections explains more tools. Screwdrivers, wrenches, pliers, wire strippers/cutters, rotatory tools, drills, breadboards, jumper wires, power supply, soldering tools, multimeter, computer and programming software are some additional tools individuals may need to build the robot (Selig, 2005).

Sensors.

Using sensors, the robot can sense the surrounding environment and make decisions accordingly. An ultrasonic sensor, an infrared sensor, a laser sensor, a stretch and bend sensor, and a stereo camera system are some examples of sensors that can help the robot measure distance. GPS can sense the location and the potentiometer, and the gyroscope can sense the rotation. The light sensor, the sound sensor, the thermal sensor, the humidity sensor, the pressure sensor, a gas sensor, and a magnetometer can sense the surrounding environment (Vukobratovic, & Potkonjak, 1982).

Assembly

One can use existing commercial products such as a robot-board. A frame can be made by using traditional tools or by using metal/plastic cutting machines or 3D printer. 3D printing and similar processes may have high cost (Srinivas, Dukkipati, & Ramji, 2009). Assembly process may vary according to the complexity and the design of a robot but can be divided into a few common steps listed below:

1. Putting different parts together inside the frame
2. Connecting motors to the motor controller
3. Connecting power supply
4. Connecting the motor controller, sensors, and communication device to a microcontroller
5. Adding additional parts according to the design and use of a robot

Materials.

Wood, metal, synthetic material, and composite materials are few materials that can be used to build the robot frame. Steel, aluminum, copper, brass, and bronze are basic metals that can be used to make the robot. Polyvinyl chloride (PVC) and plexiglass, foam core and cardboard are other materials that can be used to make a robot (Wright, & Bourne, 1988).

Programming

The microcontroller only understands the binary language, which works in the form of *1s* and *0s*. On the other hand, the programmer understands the programming. Many programming software widely recognized as programming language, is available in the market, such as Java, C, C++, Assembly, .NET, Processing, Basic, and Python. These programming languages convert the command of a person into a language the microcontroller can understand (binary). After successful completion of a program, a program can be burned into the microcontroller’s memory. A beginner can focus on learning one programming language, depending upon which microcontroller one is using, and which is the supporting language of a microcontroller by the manufacturer (Srinivas et al., 2009).

One can use a microcontroller development board to program a microcontroller that is easy to use and provides USB interface, so a programmer can connect a microcontroller directly to the computer. A microcontroller development board is a circuit board that is equipped with a voltage regulator, oscillator, resisters and a USB plug that provides support to the microcontroller to be programmed. After connecting a microcontroller to a computer through a USB or RS-232 port, a computer requires drivers to communicate with a microcontroller. The drivers can be found on the manufacturer's website or according to the user manual provided by the manufacturer. The manufacturer of a microcontroller kit usually provides sample codes to program a microcontroller. Documentation of code is a key part, as a programmer develops more codes and knowledge of programming of a microcontroller (Srinivas et al., 2009).

Mobile land robots

As the name states, a mobile land robot is a kind of robot that is locomotive and uses the platform of land robot. (See section "choosing a robotic platform" for more details). "Locomotion is a process of moving an autonomous system from one place to another" (Klancar, Zdesar, Blazic & Skrjanc, 2017, pp. 13). A robot can be autonomous or non-autonomous. A non-autonomous robot is one that is fully or partially controlled by an operator. An autonomous robot can make decisions autonomously. An autonomous robot may require the ability to calculate velocity, angular velocity, travel path, trajectory, obstacles, environmental factors, and targeted mission. An autonomous robot requires sensors and a combination of electrical and technical parts. It is important to have an idea about the relationship between the input parameters of the robot and how a robot will react to those inputs for different conditions. (Klancar et al., 2017).

The actuators, manipulators, control systems, appropriate sensors, adequate power supply, and well-designed software are the necessary parts to build the basic mobile land robot. The right blend of all the systems above needs to fit appropriately together to make the mobile land robot work successfully. Instead of controlling a robot by a remote controller all the time, a robot should have a small amount of artificial intelligence. A robot competition such as an obstacle robot race requires a robot to be artificially intelligent, identify obstacles, pass them and finish the race. For the autonomous mobile land robot, intelligence is a sense of surrounding using sensors and an appropriate automated reaction. (Jones, Flynn & Seiger 1999).

Generally, the competitor is not allowed to control the robot during the obstacle race type of a robot competition. Hence, the robot must be autonomous for an obstacle race. The rules of the competition may vary. It may sound difficult to make artificially intelligent mobile land robot for obstacle race robot competitions, but it is simple. By implementing an electrical circuit explained as following, one can tell the robot when to turn, how far to turn, when to move forward and when to reverse. A robot is equipped with a bump sensor that detects the collision. An electrical circuit with microcontroller moves all the wheels in the forward direction until a bump sensor detects collision. Once the sensor detects the collision, the circuits move the wheels in reverse direction, and then the robot backs up. One can choose to make the robot with four wheel-two drive motors (one on each side) or four wheel-four drive motors (one on each wheel). A resistor-capacitor (RC) circuit is connected with each side of the motors. An RC circuit works on the base of a state of an element or timing that makes a robot turn. If one of the RC circuits is set for the different amount of period then other, one side of wheels will move more than the other side wheels, which will make a robot turn. The robot then resumes forward motion and repeats the same tasks until it crosses all the obstacles (Fjermedal, 1986).

Parts

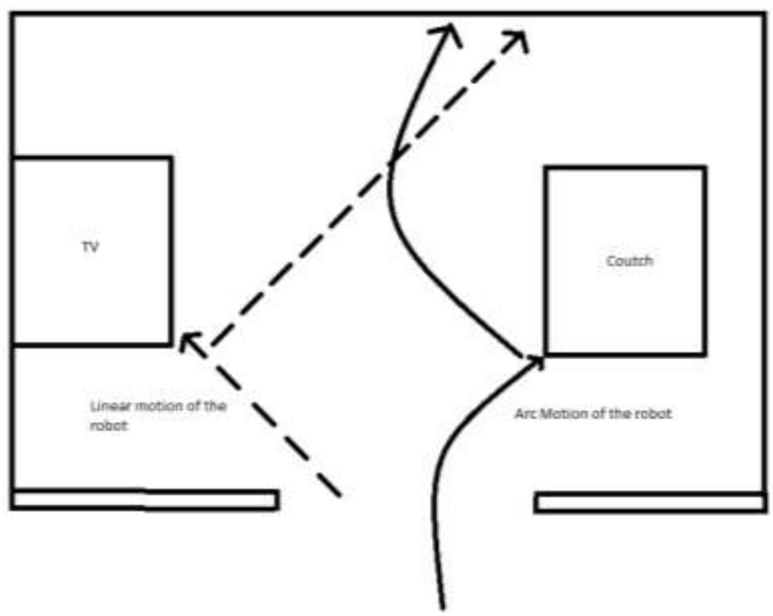


Figure 5. Arc and linear paths of the autonomous robot

As shown in Figure 5, a robot can be a programmed to move forward in linear and arc motion. During the forward motion, if one side of wheel/wheels can move little faster than the other side, one can enhance the arc motion for the robot. In this section, an example to build an obstacle race-type robot is explained. The function of such a robot is to avoid the obstacles and follow along a wall if there is one. There are two wheels used to build the robot. The trailing caster wheel maintains the stability of the robot. One can choose where he or she wants to put all other parts on the robot such as the battery and circuits. This robot is designed with all analog circuits to eliminate the coding part (Christensen, Bunke, & Noltemeier, 1999).

This type of robot requires a motor, gears, axles, wheels, switches, connectors, chassis, and collision sensors. One may require more components according to changes in the design. Figure 6 shows the functional block diagram of the robot, and how information passes from the collision sensor to motors (Butterfield et all, 2016).

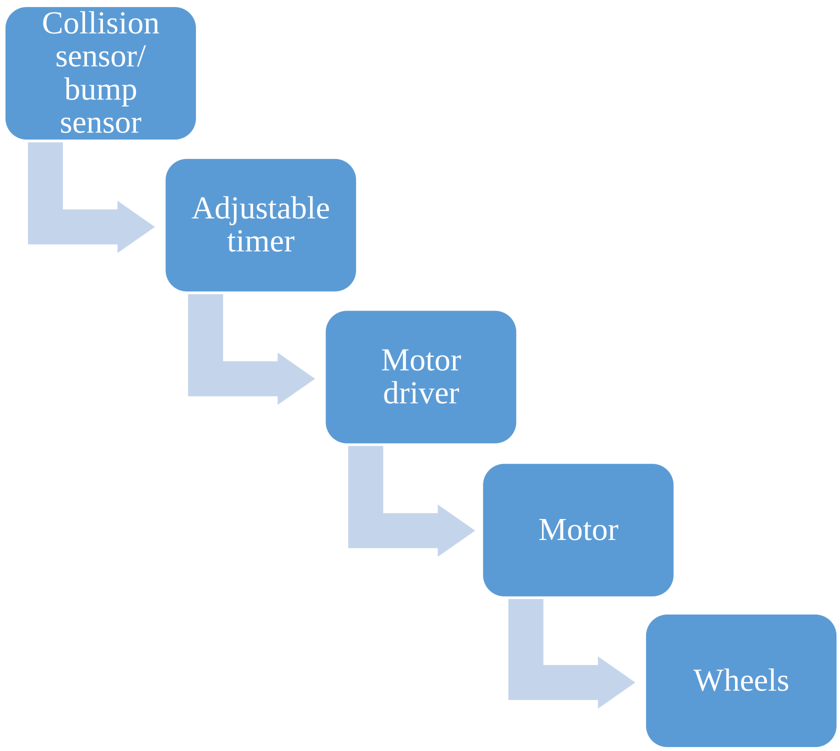


Figure 6. Functional block diagram of the robot (Dutton, Thompson, Barraclough, 1997)

Figure 7 shows the timing diagram of the collision sensor and both motors. According to the timing diagram, the robot moves in the forward direction initially. As soon as the collision sensor senses the collision, both motors change the direction and the robot moves in reverse direction. The right-side motor moves little more in reverse direction then left side motor, which will make a robot turn in right side direction.

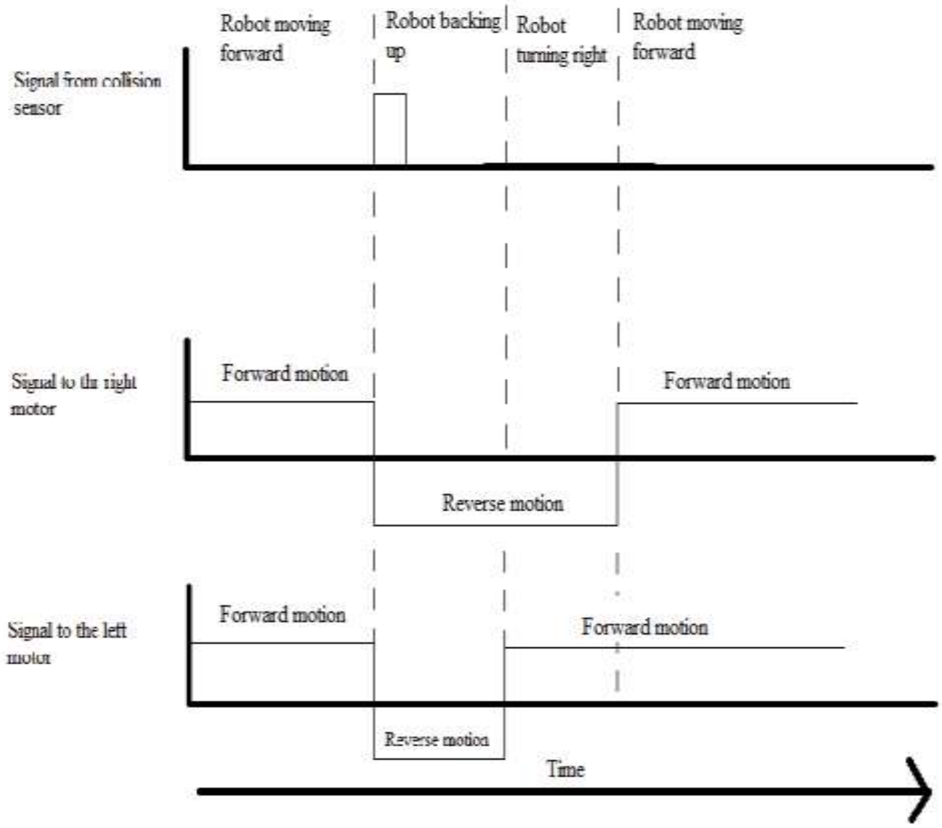


Figure 7. Timing diagram of the robot (Robillard, 1984).

The robot will repeat the sequence until it avoids the obstacle. In the above case the robot moves in straight line in forward direction. As shown in Figure 7, to move the robot in arc, one motor needs to turn slower than other motor. The motion can be enhanced by simply adding a resistor in series with the motor. The robot will turn in the direction of the slower moving motor. If the right-side motor is moving slower than left motor, the robot arcs to the right. If the left motor is turning slower than right motor, the robot arcs to the left (Stoy, Brandt, & Christensen, 2010).

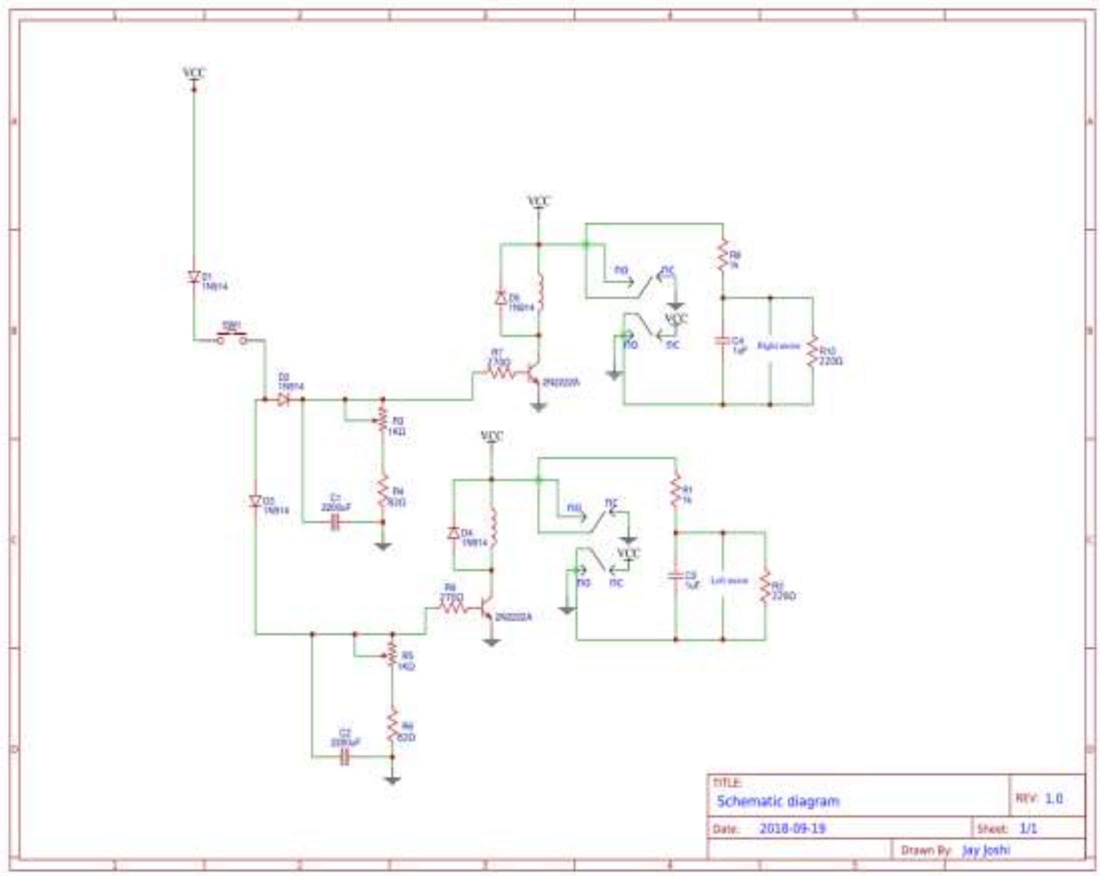


Figure 8. Reference circuit diagram for obstacle type robot

To make an obstacle race-type robot, one can start with building a simple robot. The construction of simple robot is explained in the section “Example of the robot design”. While constructing a simple robot, one needs to understand and follow the circuit design explained Figure 8. It is advisable to prepare a base robot and use the same robot by making required changes for different robot competitions (Talebi, Patel, & Khorasani, 2001).

Appendix G shows all the parts needed for the obstacle-type race robot. One can choose different parts and to build a robot with parts other than are explained in appendix G. DC motors have higher rotational speed and lower torque. To turn DC motors slower and increase the torque, one needs to an install gearbox between motors and wheels. To decrease the rotational speed of the motor and increase the torque of the motor, one needs to shift the gear down. A worm gear, transfer box, and axle-mounted gear together can provide excellent speed reduction (Vukobratovic, 1989).

Robot Competitions

A person who wants take part in robot competition generally requires creating a report about his or her robot. This report may differ for different competitions and organizers. The report may include the structure of the robot, parts used, technical figures, cost of parts, and other details required by the robot competition organizer. This chapter provides few case studies of such reports and detail of these robotic competitions is shown in Table 2.

Table 2

Robot competitions and related details

	Organizer	**University**	**Place**	**Year**	**Team**	**Title of the report**
1	ATMAE	Millersville University	Millersville, PA	2011	MU robotics team	Millersville Automatic Color-Sorter with Dedicated Active Dual-Drive Integration
2	ATMAE	East Carolina University	Greenville, NC	2012	Team-1	2012 ATMAE Robotics

						Technical Report
3	BMCET	Gujarat Technological University	Surat, GJ	2014	Team-3	Accelerometer Based Gesture Controlled Robot

The general expectation for robotic competition technical report sis to provide information about team members, advising faculty, general rules of the competition, robot design, robot construction, robot testing, important diagrams, expenses, and relevant references. The expectation may differ for different competitions and organizers.

Case study of a robotics team from Millersville University, Millersville, PA-2011

The mobile robotics research team from Millersville University, Millersville, PA, prepared a mobile robot in 2011 that detected colors of balls and sorted them by colors. The team focused on simplicity, power and stability for their robot. Figure 9 shows the picture of their robot “MAC DADI” (Millersville Automatic Color-Sorter with Dedicated Active Dual-Drive Integration). They used steel and aluminum tubes to build the chassis with wider wheels to make the robot stable. Use of the aluminum tubes decreased the weight of the robot significantly, which helped to increase power. MAC DADI had two wheelchair type motors that pull ed more current if more power was required. The motors had high torque and high-speed capacity that carried up to 400 lbs. with good speed. The robot used an Allen Bradley 45CLR ColorSight color sensor. Figure 10 shows the color sensor.

Figure 9. MAC DADI (MU robotics team, 2011)

Figure 10. Allen Bradley 45CLR ColorSight (MU robotics team, 2011)

The sensor detects three different colors and the processor will assume the fourth color, which the sensor is not able to detect. The advantage of using this sensor was that it provided accurate sensing of a color over a greater distance. The pickup system (for balls) was equipped with a vacuum pump that generates the enough suction to pick up a ball. The robot was equipped with three BX-24p microcontrollers. Each processor processed 83,000 instructions per second and had 32 KB of storage memory and 400 bytes RAM. The purpose of using three processors was to increase the processing speed. Each processor had its individual task to perform. Three tasks included automation, teleoperation, and color detection. Refer to Appendix B, C, D and E for the schematic diagrams (MU robotics team, 2011).

Case study of a robotics team from East Carolina University, Greenville, NC- 2012

Figure 11. The robot developed by a team of students at East Carolina University (Schmider, et al., 2012)

A team from East Carolina University, Greenville, NC, prepared a mobile robot in 2012. The team developed a mobile robot that worked as a base for future teams to take part in robot competitions. The base mobile robot they developed was an upgraded version of a robot that they developed in 2011. The total cost to build a robot was $1642.55. Figure 11 shows the picture of the robot.

Case study of a robotics team from Gujarat Technological University (GTU), Surat, GJ- 2014

A team of four students from Gujarat Technological University prepared a gesture-controlled robot in 2014. The robot did not have any specific function, but the way to control the robot was different. Most of the robots are controlled by a remote controller. The team of GTU students made the robot control by hand gestures. The person who was controlling the robot wore a specific kind of robot controlling device. Appendix H and I shows the circuit diagram of the transmitter and receiver side of the robot. The robot controlling device was embedded with circuits, sensors, trans-receiver, and a power source. The person who controlled the robot kept his or her arm in a horizontal position, as shown in Figure 12. Keeping the hand in horizontal position kept the robot in standby position. As shown in Figure 13, moving the hand left turned the robot left. Similarly, moving the hand in a forward direction moved robot forward, moving the hand backwards moved robot backwards, moving the hand right turned the robot right (Patel S., Shah H., Golwala U., Joshi J., 2014).

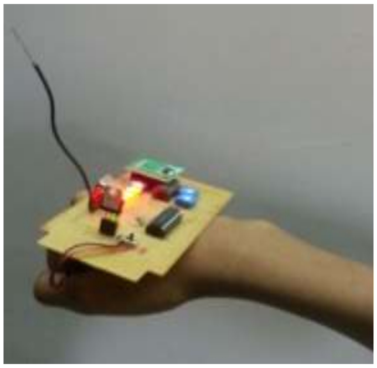


Figure 12. Stand by position (Patel S. et all, 2014)

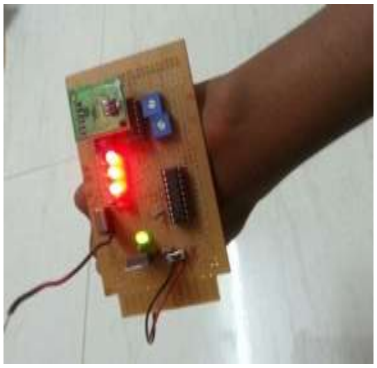


Figure 13. Hand position to turn robot left side (Patel S. et all, 2014)

Figure 14. Hand position to turn robot right side (Patel S. et all, 2014)

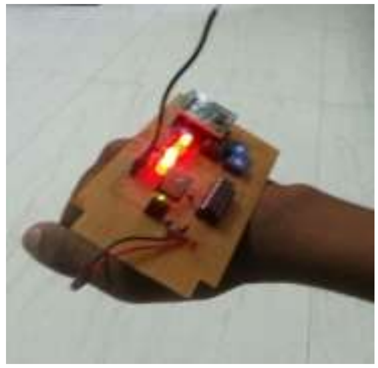


Figure 15. Hand position to move robot reverse (Patel S. et all, 2014)

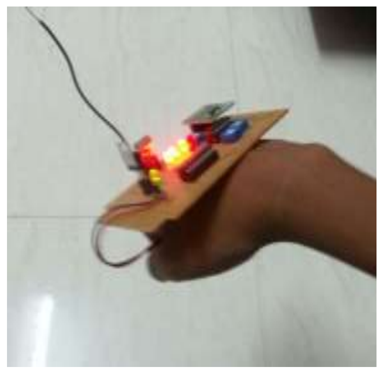


Figure 16. Hand position to move robot forward (Patel S. et all, 2014)

Examples of Robot Design

A robot can be made with or without a microcontroller. A simple robot can be made using a wired remote control. A power switch is an important mechanism in the robot to turn on and off the power supply for entire robot. It is useful mechanism that saves power while the robot is not in use and provides safety when a robot is under maintenance work. Use of a reset button for a robot is a smart idea because if a robot is not functioning according to the instructions, one can simply reset a function of the entire robot. Experts suggest avoiding making the entire robot in one or a few days. Before proceeding to build a robot, one can make a project plan, and gather important information and materials before starting up. Is wise to divide the entire project in small chunks and work on different chunks to integrate all parts together. This will eliminate risk of failure for the entire project (Gauthier, & Regnier, 2010).

Once the project plan is ready, the next step is to purchase components. Generally, shops like Walmart, Lowes, HomeDepot, and Best Buy do not sell robotic components such as microcontrollers, motors, motor drivers, and other complex electronic and mechanical components. The Internet is an excellent source to purchase robotic components. Appendix A shows a few some parts companies and their websites (Wachsmuth, Lenzen, & Knoblich, 2008). The websites shown have components from different manufacturers. Hence, make sure the components are compatible and align with the project plan. Most of the manufacturers provide component guide (catalogs), and instruction sets (for microcontrollers) on the website as an open source. Hence, checking the guides and detailed research about a component before making a purchase is advisable. Moreover, manufacturers provide their phone numbers on each website for customer support (Rosheim, 1994).

Safety

One does not require an engineering degree or any prior experience to build a robot for the first time. However, one needs to work safely. The following can cause safety issues: spark, burn, or heat from soldering machines or batteries, striking sensitive body parts with heavy parts of the robot, nips, cuts, fires, chemical exposure, and electric shock from components. Use appropriate safety precaution for these safety hazards (Robillard, 1983).

It is good to know few facts about electricity to stay safe. Voltage is a measure of force of electricity and ampere is amount of electricity flowing. Voltage doesn't harm the human body, but ampere does. The most important fact about electricity one should know is 500 mA (mili-ampere) of electricity can damage the human body at greater extent, and 1 A (ampere) could kill a person (Popov, & Yurevich, 1987). Hence, one need good awareness while working with electricity.

Numeric system

Numbers play a significant part in preparing the robot. Voltage, current, power, frequency, distance, length, weight, resistance, temperature, torque, mass and many more units related with robotics requires a numeric system. Use of a metric system is advisable because most of the manufacturers of robotic components use this system. Hence, it is easy and compatible to convert one unit into another if the numeric system used is unified over the entire process. The knowledge of measurements, unit of measurements, and abbreviation of units can help build a good robot (Gevarter, 1984). Some of the common measurements, units and abbreviations are shown in Table 3.

Table 3

Measurements, units, and abbreviations (Cook 2002)

Measurement	**Units**	**Abbreviations**	**Unit refers to:**
Length or distance	meter	m	Component dimensions
Resistance	ohm	Ω	Resistors
Voltage	volt	V	Batteries, motors, electric components

Current	ampere or amp	A	Batteries, motors, electric components
Power	watt	W	Electric components, motors
Frequency	hertz	Hz	Crystals, chips, signals
Capacitance	farad	F	Capacitors
Inductance	henry	H	Inductors
Amplification	beta factor	h_{FE} or β	Transistors
Luminous intensity	candela	cd	LEDs, light bulbs
Mass	gram	g	Robotic components
Torque	newton-meter	Nm	Motors
Capacity	amp-hour	Ah	Batteries
Temperature	Celsius	°C	Robotic components
Rotational speed	revolution per minute	rpm	Motors

Line follower robot

Robot race, robot war, line follower robot competition, and obstacle path robot race are a few types of robot competitions. As the names suggest, robot race is race between two or more robots. Robot war is direct fight between two or more robots (under rules, regulations, supervision, and controlled safe environment). Line follower competition are where robots are supposed to follow a black line autonomously and pass all the obstacles in order to finish the race. Among different kind of competitions, line follower robot competition is explained in this section. The reason for choosing line follower robot type competition is to explain autonomous robots (Calabrese, 2003).

It is very important to understand the details about the line the follower robot is going to follow. A line is required to test the line follower robot. For the experimental stage, it is advisable to get solid color line on the solid color floor. If the line color and the floor color are the same color throughout the course, then it will make it very easy for the robot to detect the line. Generally, in the robot competition, the color of a line remains the same throughout the course, but there may be a different line colors in the course to make robot competition tough. Hence, one need to be ready for tough conditions for robot competition. The ideal width of the line is from 1.8 cm to 2.54 cm (Cook 2002, pp. 76). The robot can be designed to follow the darker or lighter line in comparison with floor. One can make a line with any kind of paper that is solid colored. If one is not able to find solid colored paper, one can paint poster boards with marker pens and cut it in appropriate shapes. On the other hand, one need to fix other materials to stick to the floor such as poster board. When fixing the line to the floor, one needs to make sure line is not broken. Hence, tape is excellent resource for making a line because it will stick to the floor and not move. A robot sensor detects the light reflected by the line; thus, a line should not be shiny because it may reflect more light then a robot designed to detect. A dark colored masking tape is good resource to make a line (Derby, 2005).

It is easy to follow a straight solid line for a line follower robot. However, robots can be designed to follow broken lines and handle turns of 45° or more. It is advisable to prepare solid lines and limit the turn angle by 22.5° for testing purposes (Cook 2002, pp. 78). A robot gets confused when two lines are crossing each other. The robot explained in this section provides a solution. The floor is supposed to be obstacle free for line follower robot competition. One should decide the size of the robot according to the robot competition requirements.

One can design a robot to follow both dark colored and light-colored lines. One requires two sensors at the bottom part of robot that will detect the line. Both sensors can be fixed on the robot from each other in accordance with the width of the line. If the line is thick, the sensors needs to be attached far from each other and if the line is thin, the sensors need to be attached near to each other. One can choose to have two wheels or four wheels for a robot. According to the design of the robot, one can choose to have two or four motors to drive the wheels (Ducasse, 2005).

If one decides to have four motors driving individual wheels, both motors on one side should be connected together. Thereafter, a toggle switch is required somewhere on the robot. The toggle switch has three positions, right, left, and center. One is required to connect the sensor, motors, and switch in the following way. When the switch is in the left position, the left sensor is connected with the left-side motors and right sensor is connected with the right-side motors. In the above case, the robot will follow the dark colored line (in comparison with the floor). By toggling the switch to the right position, left sensor is connected with right-side motors and right sensor is connected with left-side motors. In the above case, the robot will follow the light-colored line (in comparison with floor) (Bekey, 2005).

One can choose the battery according to the conditions of the robot competition and components power requirement. A power switch will either connect or disconnect the main power supply to the components. The power switch will turn on the motors, brightness sensors and LEDs. Four brightness sensors and two or more bright LED lights are required for this robot. The brightness sensor sees the light reflected from the line and sends the appropriate command to the robot to follow the line. The LED lights are to provide enough light to sense the line by the brightness sensor. LEDs also allows the robot to operate in darkness. The entire assembly of brightness sensor and LED are attached to the front bottom side of the robot. Two middle sensors are the main sensors which will make the robot to follow the lines. The robot may miss the line on a sharp curve and may go off-track. To avoid this situation, two more sensors are used on each corner. If any of the two sensors on the corner sense the line, that means the robot is going off the track and it will send the appropriate signal to the motor to turn robot in right direction (Holland, 2004). The assembly of the brightness sensors and LEDs are shown in Figure 17.

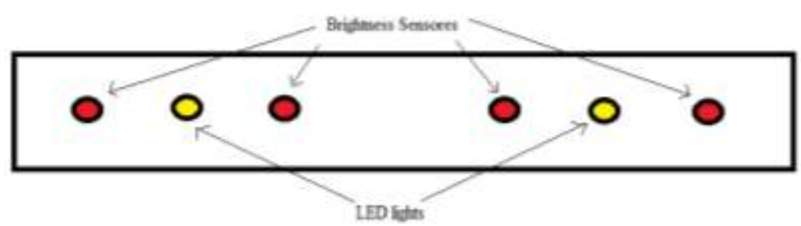


Figure 17. Assembly of LEDs and brightness sensors (Lin, Abney, & Bekey, 2012)

At the beginning of the race the robot must be placed on a line in such a way so the line will pass through the two middle brightness sensors. The brightness sensor measures the amount of the light received. By the change in the brightness reflected from a line, the sensor will know which direction the robot is going. Both sensors will try to maintain the robot in the middle of a line by sensing the equal amount of brightness. If any of the sensor measures more brightness the other one, it will send the signal to the appropriate motor and will turn the robot in right direction and will keep the robot in the middle of a line (Zhelyazova, 2005).

The brightness sensor comes with the adjustment of a brightness sensing capability. The sensors used in the robot may not sense an equal amount of brightness. This tool is very useful to troubleshoot the problem when the robot is not following a line properly. If the robot is not following a line properly, the problem could be with sensors only. If the sensors are not sensing the right amount of brightness, it could mislead the robot. To troubleshoot the problem quickly, one can attach two LEDs on the top of the robot. If one side of the sensor is sensing more light then other side of sensor, that side of the LED will glow (Zhelyazova, 2005).

A chip named *comparator* is a brain of the line follower robot. The function of the comparator is to compare two or more signals and provided an appropriate output signal as the design of the chip. Here, the signal from both brightness sensors is fed to the comparator. The comparator will turn on and off the motors rapidly of the appropriate side according to the input signal generated by brightness sensor. A few components and chips work on the high power and a few on low power. If the component that consumes the higher power such as motor is directly connected with a component that consumes lower power such as comparator or microprocessor, the comparator or microprocessor may not be able to supply enough power to the motor. In such cases, a transistor is required to connect in between (Yang, Xia, Shi, & Zhao, 2012).

Conclusion

This study on the land based mobile robotic systems was designed to help individuals who want to participate in robotic competitions. There are many opportunities to study robotics in depth. As a result, the information gathered about the land based mobile robotic systems related to robotic competitions was condensed for ease of study.

The second section was divided into nine sub-sections to provide the initial idea of the robotic systems to individuals who have minimum or no exposure to robotics. The information in the sub chapters, defining a task of robot, choose a robotic platform, actuators, microcontrollers, motor controllers, the human-machine interface, power supply, tools, and programming have basic information about robotics. As described earlier, the aim was to focus on land-based robotic systems. The third section, mobile land robots, provided the general idea of the land-based mobile robotic system.

Three case studies were presented related to robotic competitions. The first study was the team participated at Millersville University, Millersville, PA in 2011. The second study was the group at East Carolina University, Greenville, NC in 2012. The third study was about the group at Gujarat Technological University, Surat in 2014. The fourth section provided a practical example of how to build a line follower robot.

References

Astolfo, D., Ferrari, M., & Ferrari, G. (2007). *Building robots with lego Mindstorms NXT*. Burlington, MA: Syngress Publishing, Inc.

Bateson, R. N., (1993). *Introduction to control system technology*. New York, NY: Macmillan Publishing Company.

Bekey, G. A. (2005). *Autonomous robots.* London, UK: The MIT Press

Burns, R. S. (2001). *Advance control engineering.* Oxford, UK: Butterworth Heinemann.

Butterfield, A., & Ngondi, G. (2016). *A dictionary of computer science.* Oxford, UK: Oxford University Press.

Calabrese, S. R. (2003). *Practical controls a guide to mechanical systems.* Lilburn, GA: The Fairmont Press, Inc.

Campbell, J., Antoniewcz, M., Gladman, J., Hayes, H., Williams, A., Wooten, Z., Thompson, F., Baxter, K., Nganga, D., & Nobel, A. (2012). 2012 ATMAE Robotics Technical Report. North Carolina Agricultural & Technical State University. Greensboro, NC.

Christensen, H. I., Bunke, H., & Noltemeier, H. (Eds.). (1999). *Sensor based intelligent robots.* Heidelberg, Germany: Springer.

Colestock, H. (2005). *Industrial robotics: selection, design and maintenance*. USA: McGraw-Hill.

Cook, D. (2002). *Robot Building for Beginners.* USA: Apress.

Cook, D. (2010). *Intermediate robot building.* USA: Apress.

Cook, E., Mohan, K., Shome, R., Bekris, K., Shin, J., & Weisel, C. (2018, August). Exploring the Use of Robots for Exposure Studies. In *ISEE Conference Abstracts* (Vol. 2018, No. 1).

D'azzo, J. J., & Houpis, C. H. (1995). *Linear control system analysis and design conventional and modern.* USA: McGraw-Hill.

Derby, S. J. (2005). *Design of automatic machinery.* New York, NY: Marcel Dekker.

Ducasse, S. (2005). *Squeak learn programming with robots.* USA: Apress.

Dumouchel, P., & Damiano, L. (2017). *Living with robots* (M. DeBevoise, Trans.). London, UK: Harvard University Press.

Dutton, K., Thompson, S., & Barraclough, B. (1997). *The art of control engineering.* UK: Addison-Wesley.

Engelberger, J. F. (1989). *Robotics in service.* Cambridge, UK: The MIT Press.

Fjermedal, G. (1986). *The tomorrow makes a brave new world of living-brain machines.* New York, NY: Macmillan Publishing Company.

Gauthier, M., & Regnier, S. (Eds.). (2010). *Robotics microassembly*. Piscataway, NJ: IEEE Press.

Gerdes, L. (Ed.). (2014). *Robotic technology.* Farmington Hills, MI: Greenhaven Press.

Gevarter, W. B., (1984). Robotics: An overview volume 1 robotics and artificial intelligence applications series. Chicago, IL: Manufacturing Productivity Center.

Hall, E. L., & Hall, B. C. (1985). *Robotics a user friendly introduction.* New York, NY: CBS College Publishing.

Hartley, J. (1983). *Robots at work a practical guide for engineers and managers.* Bedford, UK: IFS (Publications) Ltd.

High, G., Schoenenberger, J., McDonnell, B., Henderson, M., Henderson, M., Grewell, D., & Hoffman, R. (2011). Iowa State University ATMAE Robot Competition Technical Report 2011. Iowa State University. Ames, IA.

Hsiao, C. W., Chien, Y. H., Wang, W. Y., Li, I. H., Chen, M. C., & Su, S. F. (2015, April). Wall following and continuously stair climbing systems for a tracked robot. In *2015 IEEE 12th International Conference on Networking, Sensing and Control* (pp. 371-375). IEEE.

Holland, J. M. (2004). *Designing autonomous mobile robot inside the mind of an intelligent machine.* Oxford, UK: Elsevier.

Hunt, V. D. (1985). *Smart robots a handbook of intelligent robotic systems.* New York, NY: Chapman and Hall.

Indiamart (2012). [Graphical explanation and online sell]. SKF motor encoder unit. Retrieved from: https://www.indiamart.com/proddetail/skf-motor-encoder-units-7420207733.html

Jones, J., Flynn, A., & Seiger, B. (1999). *Mobile robots inspiration to implementation.* Natick, MA: A K Peters.

Kalften, R. D., Chmielewski, T. A., & Negin M. (1989). *Robotic engineering an integrated approach.* Englewood Cliffs, NJ: Prentice Hall.

Klancar, G., Zdesar, A., Blazic, S., & Skrjanc, I. (2017). *Wheeled mobile robotics from fundamentals towards autonomous systems.* USA: Joe Hayton.

Launius, R. D., & Mccurdy, H. E. (2008). *Robots in space technology, evolution, and interplanetary travel.* Baltimore, MD: The Johns Hopkins University Press.

Lin, P., Abney, K., & Bekey, G. A. (Eds.). (2012). *Robotics ethics the ethical and social implications of robotics.* London, UK: The MIT Press.

Mason, M. T. (2001). *Mechanics of robotics manipulation.* London, UK: A Bradford Book The MIT Press.

Masterson, J. W., Poe, E. C., & Fardo, S. W. (1985). *Robotics.* Reston, VA: Reston Publishing Company, Inc.

Mathia, K. (2010). *Robotics for electronics manufacturing principles and applications in cleanroom automation.* Cambridge, UK: Cambridge University Press.

McComb, G. (2011). *Robot builder's bonanza.* USA: McGraw-Hill.

Menzel, P., & D'Aluisio, F. (2000). *Robo sapiens evolution of a new species.* C. C. Mann (Ed.). London, UK: The MIT Press.

Mihelj, M., Bajd, T., Ude, A., Lenarčič, J., Stanovnik, A., Munih, M., ... & Šlajpah, S. (2019). *Robotics*. Springer.

Moallen, M., Patel, R. V., & Khorasani, K. (2000). *Flexible-link robot manipulators control techniques and structural design.* London, UK: Springer.

MSU Student Chapter of ATMAE (2011). FNS 2834DK. Mississippi State University. Mississippi State, MS.

MSU Student Chapter of ATMAE (2012). Procrastinator 2012. Mississippi State University. Mississippi State, MS.

MU robotics team (2012). Millersville Automatic Color-Sorter with Dedicated Active Dual-Drive Integration. Millersville University. Millersville, PA.

Nazaretov, V. M., Kim, D. P. (1990). *Engineering artificial intelligence*. I. M. Makarov, & E. I. Rivin (Eds.). Detroit, MI: Hemisphere Publishing Corporation.

Nehaniv, C. L., & Dautenhahn, K. (Eds.). (2007). *Imitation and social learning in robotics, humans and animals.* Cambridge, UK: Cambridge University Press.

Nehmzow, U. (2009). *Robot behaviour design, description, analysis and modelling.* Londonderry, UK: Springer.

Newman, D. J. (2006). *Linux robotics programming smarter robots.* New York, NY: McGraw-Hill Corporation.

Nocks, L. (2007). *The robot the life story off the technology.* Westport, CT: Greenwood Technographies.

Patel, S., Shah, H., Golwala, U., & Joshi, J. (2014). Accelerometer Based Gesture-Controlled Robot. Gujarat Technological University, Surat, GJ, IND.

Perkowitz S. (2004). *Digital people from bionic humans to androids.* Washington, D.C.: Joseph Henry Press.

Philippe, C. (1983). *Robot technology*. Englewood Cliffs, NJ: Prentice-Hall.

Popov, E. P., & Yurevich, E. I. (Eds.). (1987). *Robotics.* Moscow, Russia: Mir Publishers.

Rosheim, M. E. (1994). *Robot evolution the development of anthrobotics.* New York, NY: A Wiley-Interscience Publication.

Robillard, M. J. (1983). *Hero 1: Advance programming and interfacing.* Indianapolis, IN: Howard W. Sams & Co., Inc.

Rahimi, M., & Karwowski, W. (Eds.). (1992). *Human-robot interaction.* Washington, D.C.: Taylor & Francis.

Robillard, M. J. (1984). *Advance robot systems.* Indianapolis, IN: Howard W. Sams & Co., Inc.

Robot war wiki (2016). [Explanation of a robot name “Basher” used in a robotic competition in 2016]. Robot war wiki, part of Fandom TV community. Retrieved from: http://robotwars.wikia.com/wiki/Basher

Robotics Team (2011). Robotics competition technical report 2011. East Carolina University. Greenville, NC.

Robotics Team (2012). Technical Report for “Cube-Bot”. Morehead State University. Morehead, KY.

Rothschild (2012). Robotics. Astrobiology Science Conference (Apr. 2012). Atlanta, GA.

Ruocco, S. R. (1987). *Robot sensors and transducers.* P. G. Davey (Ed.). New York, NY: Halsted Press.

Safford, L. Jr. (1982). *Handbook of advance robotics.* Blue Ridge Summit, PA: Tab Books Inc.

Schmider, P., Burk, R., Middelberg, J., Nagakane, M., Garren, W., Black, G., Shornock, W., & Gookin, P. (2012). 2012 ATMAE Robotics Technical Report. East Carolina University. Greenville, NC

Selig, J. M. (Ed.). (2000). *Geometrical foundation of robotics.* Singapore: World Scientific Publishing Co. Pte. Ltd.

Selig, J. M. (2005). *Geometric fundamentals of robotics.* London, UK: Springer.

Slideshow (2018). [A slide presentation by Vandana Garg]. Humanoid robot. Retrieved from: https://www.slideshare.net/VandanaGarg4/humanoid-robot-57885212

Smith, M., Mcgrath, K., & Jason B. (2013). Application Programming in AWIPS II: NASA Center for AeroSpace Information.

Srinivas, J., Dukkipati, R. V., & Ramji, K. (2009). *Robotics control and programming.* Oxford, UK: Alpha Science International Ltd.

Stoy, K., Brandt, D., & Christensen, D. J. (2010). *Self-reconfigurable robotics.* London, UK: The MIT Press.

Tadokoro, S. (Ed.). (2009). *Rescue robotics DDT project on robots and systems for urban search and rescue.* Mountain View, CA: Springer.

Talebi, H. A., Patel, R. V., & Khorasani, K. (2001). *Control of Flexible-link Manipulators using neural networks.* UK: Springer.

Thompson, S., Thompson, J., Ledbetter, B., Civitello, J., McDonald, M., Montgomery, A., & Civitello T. (2012). 2012 Version “Anita Wynn” (unpublished). Jacksonville State University. Jacksonville, AL.

Thompson, S., Zenanko, A., Cannon, C., Georgeson, D., Acker, S., & Holman, S. (2011). The student etched circuit board. Jacksonville State University. Jacksonville, AL.

Vukobratovic, M., & Potkonjak, V. (1982). *Dynamics of manipulation robots theory and application.* Berlin, Germany: Springer-verlag.

Vukobratovic, M. (1989). *Introduction to robotics.* Berlin, Germany: Springer-Verlag.

Wachsmuth, I., Lenzen, M., & Knoblich, G. (2008). *Embodied communication in human and machines.* Oxford, UK: Oxford University Press.

Wilson, D. H. (2008). *How to build a robot army.* New York, NY: Bloomsburg USA.

Wright, P. K., & Bourne, D. A. (1988). *Manufacturing Intelligence.* USA: Addison-Wesley Publishing Company, Inc.

Yang, H., Xia, Y., Shi, P., & Zhao, L. (2012). *Analysis and synthesis of delta operator systems.* USA: Springer.

Zhelyazova, B. (2005). *Elsevier’s dictionary of automation technics.* San Diego, CA: Elsevier.

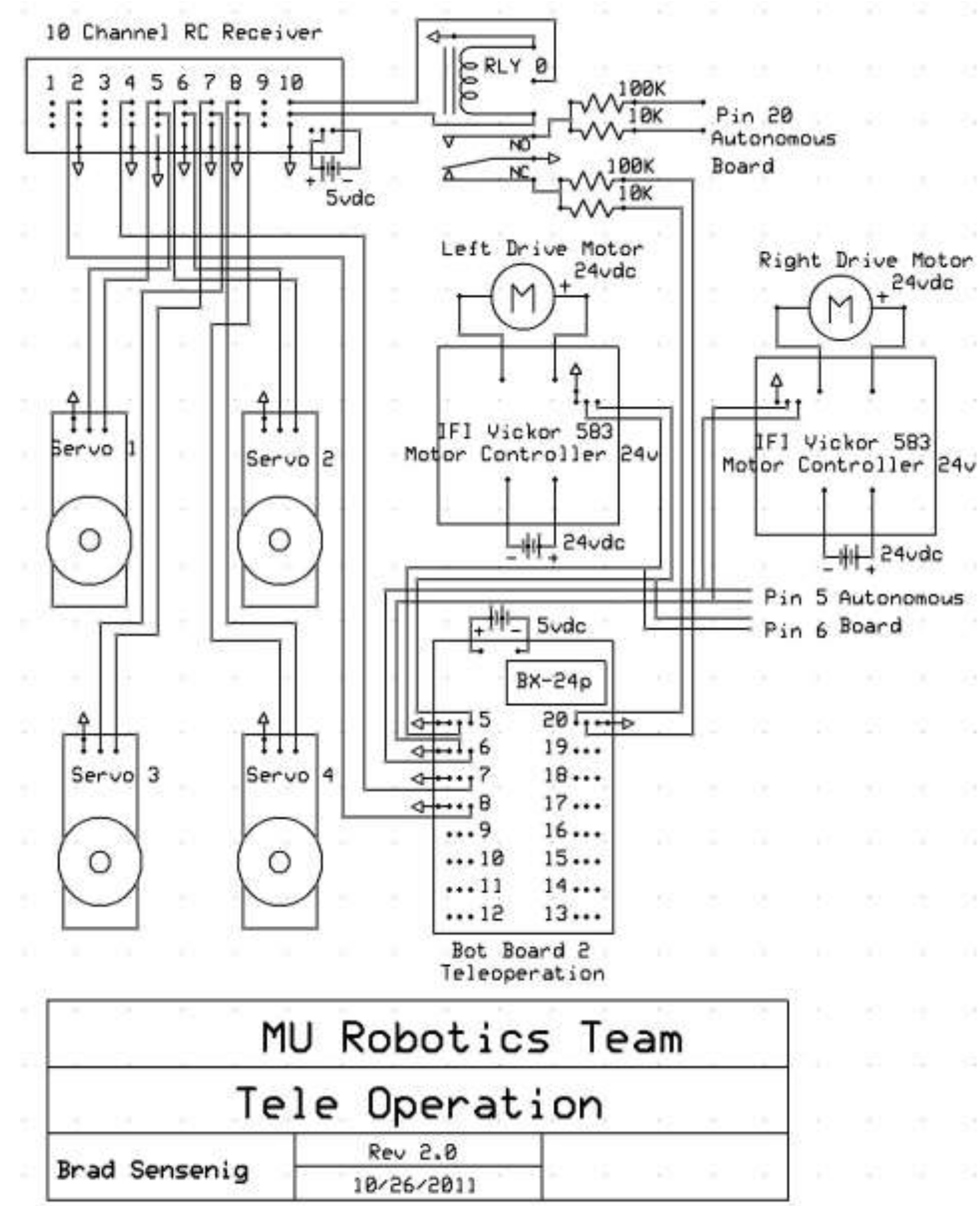
Zoro corporation (2011). [HVAC motor, 1600RPM, 1/16 HP, single phase]. Description of AC motor and online sell. Retrieved from: https://www.zoro.com/fasco-hvac-motor-115-hp-1600-rpm-120v-33-d109/i/G5229472/

Appendix

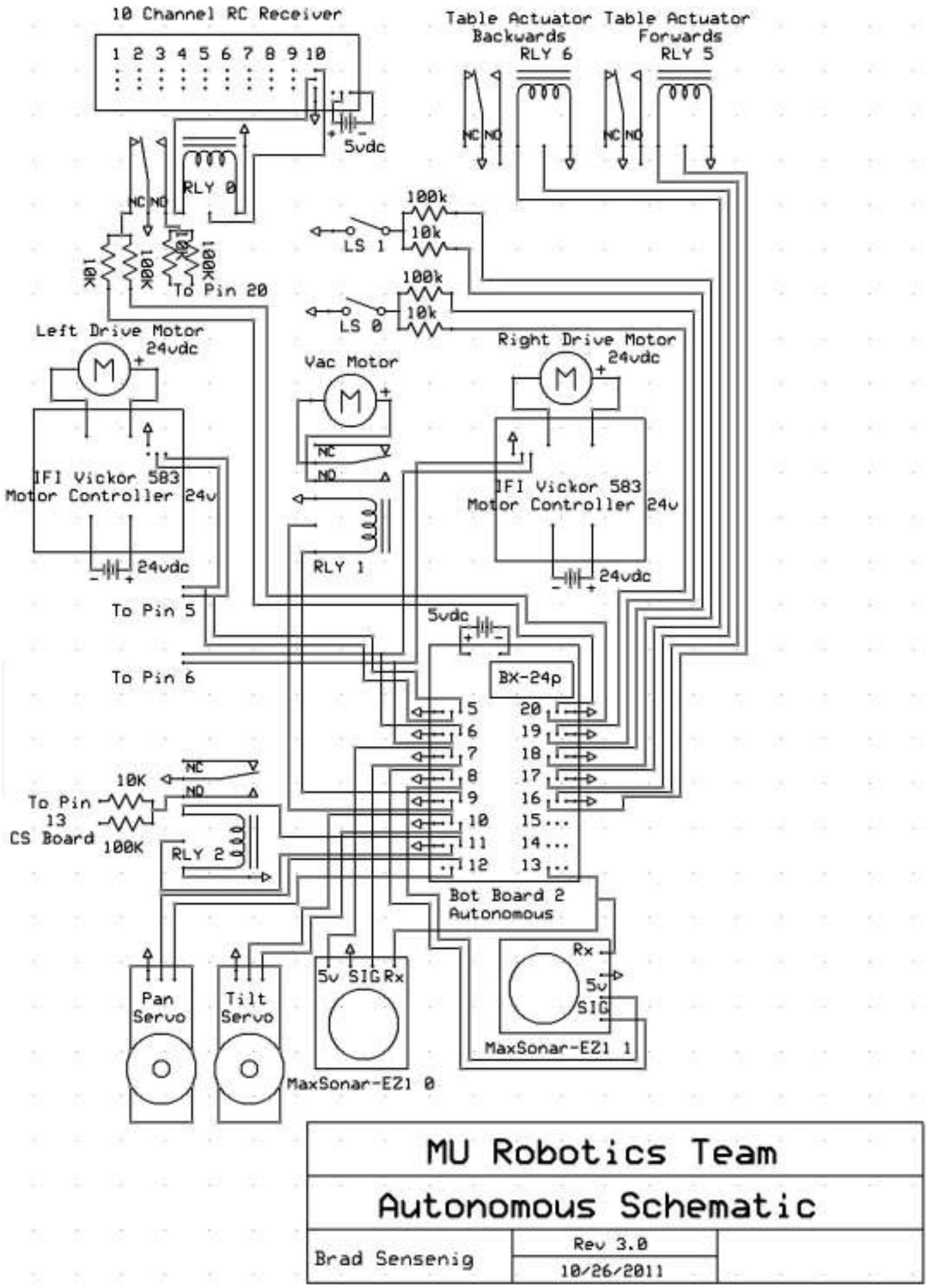
Appendix A: Robotic parts provider companies and websites

Company Name	Website
Robot Shop	www.robotshop.com
All Electronics	www.allelectronics.com
Mouser Electronics	www.mouser.com
Ebay	www.ebay.com
Amazone	www.amazone.com
Digi Key	www.digikey.com
JDR Microdevices	www.jdr.com
DH Gate	www.dhgate.com
Electronics Goldmine	www.goldmine-elec.com
Zagros Robotics	www.zagrosrobotics.com
Allied Electronics	www.alliedelec.com

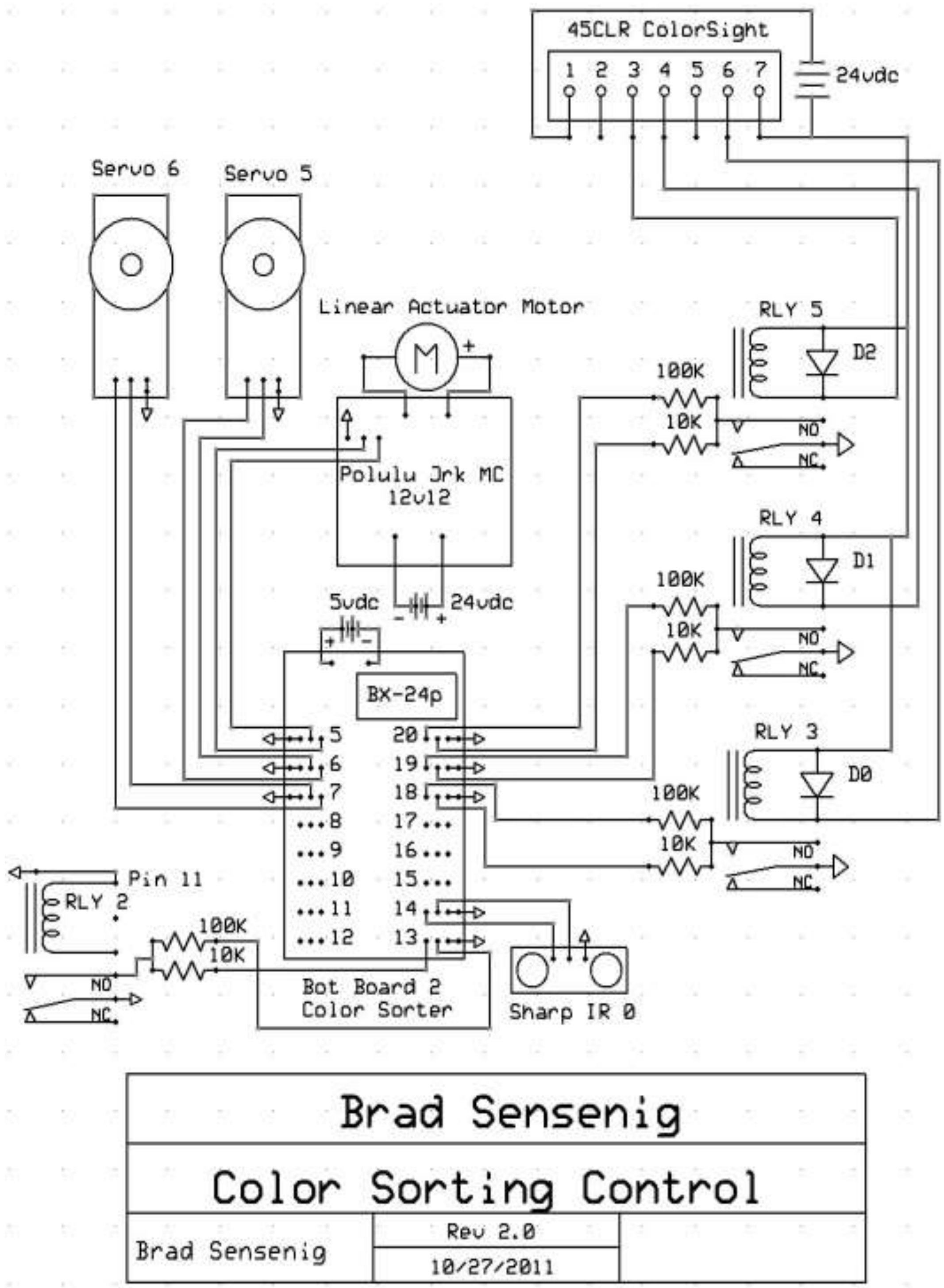
Appendix B: schematic diagram of tele-operation (MU robotics team, 2011)



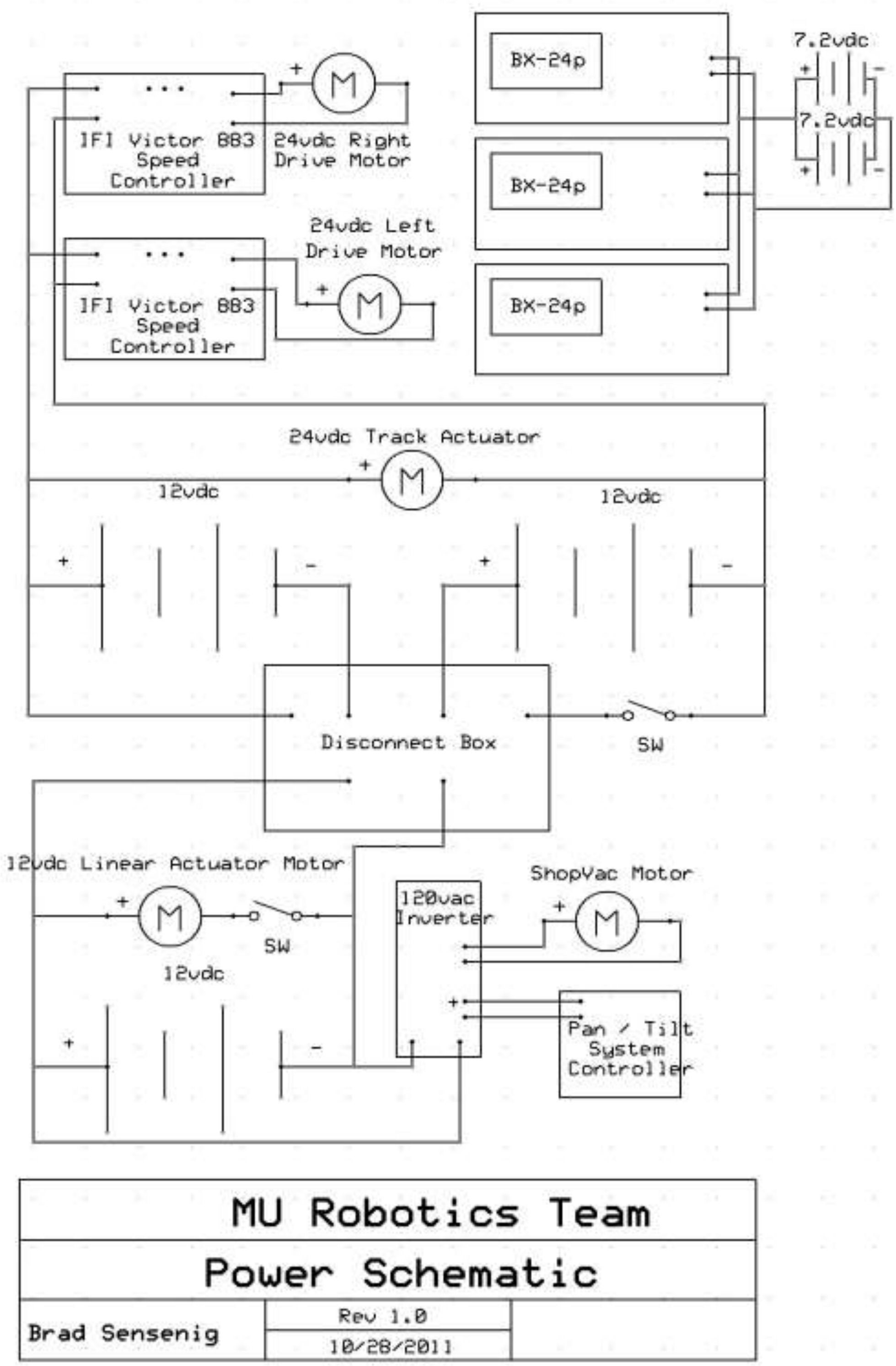
Appendix C: Schematic diagram of Automation (MU robotics team, 2011)



Appendix D: Schematic diagram of color detection (MU robotics team, 2011)



Appendix E: Schematic diagram of power supply (MU robotics team, 2011)



Appendix F: Coding

Here is the coding for the three processors used in MAC DADI. The coding is obtained from MU robotics team technical report for robot competition.

Teleoperation Code

```
' Tele-Operated code for controlling MAC DADI
' Pulsing in PWM signals from RC controller and outputting
' signals to right and left motor controllers
'------ RC Channels ------
' Channel 1 - left stick up/down
' Channel 2 - left stick left/right
' Channel 3 - Right stick up/down
' Channel 4 - Right stick left/right
' Channel 5 - switch E: Servo 1
' Channel 6 - switch C: Servo 2
' Channel 7 - switch G: Servo 3
' Channel 8 - switch D: Servo 4
' Channel 9 - switch B
' Channel 10 - switch A: Mode Switch PWM Relay
'------- Constants -------
'Motor Constants
Const LDMotor as Byte = 5 ' Left Drive Motor - pin 0
Const RDMotor as Byte = 6 ' Right Drive Motor - pin 1
'RC channels
Const RCInM1 as Byte = 7 ' RC input drive motor 1 - pin 2
Const RCInM2 as Byte = 8 ' RC input drive motor 2 - pin 3
'----- Mode Constant ----- ' Relay Input(NO) mode - pin 14
Const ModeSwitch as Byte = 20
'------------------- Sub Main --------------------------------
Public Sub Main()
Dim Width as Single, Width2 as Single
Dim Mode as Byte
Do ' Tele-opperated drive motors control
Mode = GetPin (ModeSwitch)
If (Mode = 0) Then
'Debug.print "tele drive"
Call PulseIn(RCInM1, 1, Width) ' Reads RC
 Call PulseIn(RCInM2, 1, Width2) ' signal
'Debug.Print CStr (Width)
'Debug.Print CStr (Width)
Call PulseOut(LDMotor, Width, 1) ' Outputs rc
Call PulseOut(RDMotor, Width2, 1) ' signals
```

```
Call Delay(0.015)
End If
Loop
End Sub
```

Automation Code

```
Option Explicit
'Autonomous Bx-24p
'controls robot approach to bench via sonar sensors, adjust
'drive motors on difference in distance between sensors,
'position end defector with pan / tilt system and linear
'actuator, pick up ping pong balls 'via controlling shop vac and
'performing a search pattern
'------- Constants -------
'Motor Constants
Const LDMotor as Byte = 5 ' Left Drive Motor - pin 0
Const RDMotor as Byte = 6 ' Right Drive Motor - pin 1
Const VacMotor as Byte = 9 ' Shop vac Motor
Const TiltMotor as Byte = 10 ' Motor to drop end effector
Const PanMotor as Byte = 12 ' Motor search bench
'Sensor Constants
Const RSonar as Byte = 7 ' Right Sonar 0
Const LSonar as Byte = 8 ' Left Sonar 1
Const RX as Byte = 13 ' RX pin for sonars
'Mode Switch Constant
Const ModeSwitch as Byte = 20 ' Read nc relay contact
' for tele or auto
'Limit Switch Constants
Const FRLimit as Byte = 19 ' Bench contact switch
Const FLLimit as Byte = 18 ' Bench contact switch
'Track actuator Control
Const ForwardRelay as Byte = 16 ' Relay 5 (NO) to
move Track Actuator forward
Const BackwardRelay as Byte = 17 ' Relay 6 (NO) to move
Track Actuator backward
'Comm Constant
Const CommRelay as Byte = 11 ' Relay to activate
Color Sorter board
'------------------------- Sub Main ----------------------------
Public Sub Main()
Dim Mode as Byte
Do
Mode = GetPin (ModeSwitch) ' pulse motors from this
' bx-24p if true
If (Mode = 1) Then
```

```
Call Navigate()
Call Search()
Call LeaveCircle()
End If
Loop
End Sub
'------------------------- Navigate ---------------------------
'detects bench with sonars, moves forward until front limit
'switches are triggered
Public Sub Navigate ()
Dim LeftLS as Byte, RightLS as Byte ' LS up front
Dim RightSonar as Single, LeftSonar as Single ' Sonars
Dim Compare as Single ' Comparing Sonar
' sensor distance
Do
LeftLS = GetPin (FLLimit) ' limit switch inputs
RightLS = GetPin (FRLimit) ' ---------
If (LeftLS = 1) or (RightLS = 1) Then ' Detect
' bench
Exit Do
' exit loop and sub
End If
RightSonar = GetSonar(RSonar) ' Read Sonar
 LeftSonar = GetSonar(LSonar) ' ----------
Compare = LeftSonar - RightSonar '
Compare Calculation
If (Compare <= -0.1) Then
Call AdjustLeft()
ElseIf (Compare >= 0.1) Then
Call AdjustRight()
Else
Call Forward()
End If
Loop
End Sub
'-------------------- Get Sonar --------------------------------
' Reads the Analog output of the MAXSonar® EZ1™ (PWM Pin) and
' returns the target range as a Byte
Public Function GetSonar(ByVal Pin as Byte) as Single
Dim AValue As Integer
Dim Width As Single
Call PutPin (RX, 0) ' Turn off the EZ1™
Call PutPin (RX, 1) ' Turn on the EZ1™
Call Delay (0.075) ' Wait about 50 ms
Call PulseOut (Pin, 2.0,1)
Call PulseIn (Pin, 1, Width)
```

```
GetSonar = (Width / 0.000147) ' conversion factor
' 147us
End Function
'------------------- Search ------------------------------------
'routine to search top of bench, turn on vac, move actuator
'forward and pan arm until bench is searched - a zigzag pattern,
'retract actuator full back
Public Sub Search()
Dim i as Integer, i2 as Integer, c as Integer
c = 0
Call VacOn()
For i = 1 to 50 ' Determines number of Increments
Call Increment()
If (i = 2) Then ' trigger color sorting board
For i2 = 1 to 20
Call PulseOut (CommRelay, 0.002, 1)
Next
End If
If (c = 1) Then ' ensures every other
' Increment pan right
Call PanRight()
c = 0
Else
Call PanLeft()
c = c + 1
End If
Next
Call VacOff()
End Sub
'------------------------- VacOn -------------------------------
'Turn on VAC by servo-controlled spst switch (electro to
'mechanical)
Public Sub VacOn()
Dim i as Integer
for i = 1 to 5
Call PulseOut (VacMotor, 0.002, 1)
Next
End Sub
'------------------------- PanRight ----------------------------
' pulse the pan motor right
Public Sub PanRight()
Dim i as Integer
For i = 1 to 300
Call PulseOut (PanMotor, 0.002, 1)
Next
End Sub
```

```
'------------------------- PanLeft -----------------------------
' pulse the pan motor left
Public Sub PanLeft()
Dim i as Integer
For i = 1 to 300
Call PulseOut (PanMotor, 0.001, 1)
Next
End Sub
'------------------------- VacOff ------------------------------
'Turn off VAC by servo-controlled spst switch (electro to
mechanical)
Public Sub VacOff()
Dim i as Integer
for i = 1 to 5
 Call PulseOut (VacMotor, 0.0015, 1)
Next
End Sub
'------------------------- Increment ---------------------------
'Sub increments the track actautor a set amount of pulses
Public Sub Increment()
Dim i as Integer
For i = 1 to 100 ' Determines how far
‘actuator travels
Call PulseOut (ForwardRelay, 0.002, 1)
Call Delay (0.02) ' * Necessary delay
Next
End Sub
'------------------------- LeaveCircle -------------------------
Public Sub LeaveCircle()
Dim Mode as Byte
Do
Mode = GetPin (ModeSwitch) ' will exit loop if
modeswitch = 1
If (Mode = 1) Then
Call Backward()
End If
Loop
End Sub
'------------------------- AdjustLeft --------------------------
' pulse right motor greater than left motor navigation routine
Public Sub AdjustLeft()
Dim i as Integer
For i = 1 to 10
Call PulseOut (LDMotor, 0.00125, 1) '
Adjusting PWM signals
Call PulseOut (RDMotor, 0.00175, 1)
```

```
Call Delay (0.05)
Next
End Sub
'------------------------- AdjustRight -------------------------
' pulse left motor greater than right motor navigation routine
Public Sub AdjustRight()
Dim i as Integer
For i = 1 to 10
Call PulseOut (LDMotor, 0.00175, 1) ' Adjusting
Call PulseOut (RDMotor, 0.00125, 1) ' PWM signals
Call Delay (0.05)
Next
End Sub
'------------------------- Forward -----------------------------
' pulse motors forward navigation routine
Public Sub Forward()
Call PulseOut (LDMotor, 0.00175, 1) ' Forward
Call PulseOut (RDMotor, 0.00175, 1)
End Sub
'------------------------- Backward ----------------------------
Public Sub Backward()
Call PulseOut (LDMotor, 0.0011, 1) ' Backward
Call PulseOut (RDMotor, 0.0011, 1)
End Sub
```

Color Sorting Code

```
'Color sorting board that utilizies Allen Bradley Color 'InSight
Sensor with 3 relay outputs, the sharp IR 'sensor is used as a
prox switch for detecting
' ping pong balls in the color sorter, the actuator 'with
feedback moves to position based off of which is 'detected
'Motor Constants
Const Actuator as Byte = 5 ' color sorting actautor
Const TopServo as Byte = 6 ' upper stop servo for CS
Const DropServo as Byte = 7 ' holding servo for color
' detection
'Digital I/0
Const CommPin as Byte = 13 ' Starts CS
Const BallDetect as Byte = 14 ' Sharp sensor pin
'Color Sensor I/Os
Const Color1 as Byte = 18 ' relay 3 output: Insight
Const Color2 as Byte = 19 ' relay 4 output: Insight
Const Color3 as Byte = 20 ' relay 5 output: Insight
Public Sub Main()
Dim Communication as Byte ' Relay Input from
```

```
' Autonomous Board
Dim Relay3 as Byte, Relay4 as Byte, Relay5 as Byte
' InSight Outputs
Dim SharpIR as Integer ' Sharp IO
Dim i as Integer, BallCount as Integer
Top: ' goto after all
' balls are sorted
Do
Communication = GetPin (CommPin)
If (Communication = 1) Then
Exit Do
End If
Loop
Do
Call CloseServo (DropServo) ' Closes the
' bottom Servo in CS
Call OpenServo (TopServo) ' Lets ball into CS
Call CloseServo(TopServo) ' Close top of CS
SharpIR = GetADC (BallDetect)
Relay3 = GetRelay (Color1) ' Reading the
Relay4 = GetRelay (Color2) ' InSight Outputs
Relay5 = GetRelay (Color3)
' ************** Yellow Sort and Store ***************
If (SharpIR <= 8) and (Relay3 = 0) and (Relay4 = 0)
and (Relay5 = 0) Then
For i = 1 to 200
Call PulseOut (Actuator, 0.002, 1)
' Pulse Actuator Foreward
Call Delay (0.05)
Next
Call OpenServo (DropServo)
Call Delay (0.5) ' Pause for
' dropping ball
For i = 1 to 200
Call PulseOut (Actuator, 0.001, 1) ' Pulse
Actuator to Start
Call Delay (0.05)
Next
BallCount = BallCount + 1
' ******************************* Red Sort and Store
*****************************
ElseIf (SharpIR <= 8) and (Relay3 = 1) Then
For i = 1 to 400
Call PulseOut (Actuator, 0.002, 1) ' Pulse
Actuator Foreward
 Call Delay (0.05)
```

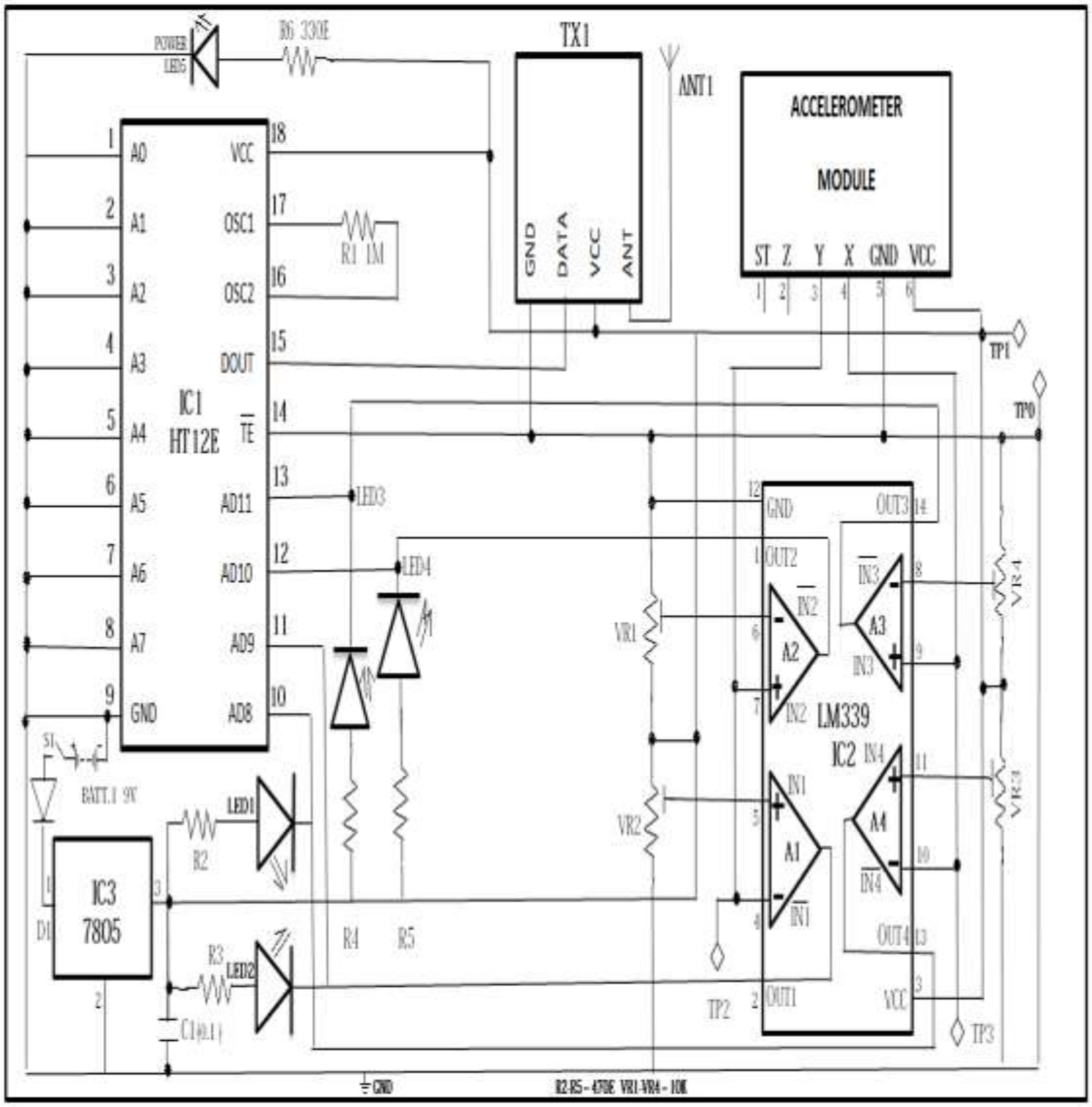
```
Next
Call OpenServo (DropServo)
Call Delay (0.5)
For i = 1 to 400
Call PulseOut (Actuator, 0.001, 1) ' Pulse
Actuator to Start
Call Delay (0.05)
Next
BallCount = BallCount + 1
' ********************************* Blue Sort and Store
*****************************
ElseIf (SharpIR <= 8) and (Relay4 = 1) Then
For i = 1 to 600
Call PulseOut (Actuator, 0.002, 1) ' Pulse
Actuator Foreward
Call Delay (0.05)
Next
Call OpenServo (DropServo)
Call Delay (0.5)
For i = 1 to 600
Call PulseOut (Actuator, 0.001, 1) ' Pulse
Actuator to Start
Call Delay (0.05)
Next
BallCount = BallCount + 1
' ********************************* Green Sort and Store
*****************************
ElseIf (SharpIR <= 8) and (Relay5 = 1) Then
For i = 1 to 800
Call PulseOut (Actuator, 0.002, 1) ' Pulse
Actuator Foreward
Call Delay (0.05)
Next
Call OpenServo (DropServo)
Call Delay (0.5)
For i = 1 to 800
Call PulseOut (Actuator, 0.001, 1) ' Pulse
Actuator to Start
Call Delay (0.05)
Next
BallCount = BallCount + 1
If (BallCount = 36) Then
Goto Top
End If
End If
 Loop
```

```
End Sub
'****************** GetRelay **************************
' Reads Relays from InSight Outputs with GetPin (Digital)
Public Function GetRelay(ByVal Pin as Byte) As Byte
GetRelay = GetPin (Pin)
Call Delay (0.05)
End Function
'****************** OpenServo **************************
' pulses servo open
Public Sub OpenServo (ByVal Pin as Byte)
Dim i as Integer
For i = 1 to 10
Call PulseOut (Pin, 0.001, 1)
Call Delay (0.02)
Next
End Sub
'****************** CloseServo **************************
' pulses servo to Nuetral Position
Public Sub CloseServo (ByVal Pin as Byte)
Dim i as Integer
For i = 1 to 10
Call PulseOut (Pin, 0.0015, 1)
Call Delay (0.02)
Next
End Sub
```

Appendix G: Parts for the obstacle type race robot

Name of the part	Quantity
Hinge	4
4W 2P/F	2
2W 2P/F	1
Wheel Pivot	1
Wheel pivot holder	1
Wheelblock	1
Wheel axle	1
Tires	3
1W P/C	2
1W2L P/C 4C	2
1W1L P/5C	4
Motors	2
Gearbox	2
1W1L P/P +4C	1
2W C/P	4
1W P/P	2
Powerblock top	1
Powerblock bottom	1
Wireholder	1
Angle C/P	2
Switch	1
Gear hub	2
Gear holder	2
Gear	2
Wheel hub holder	2
Wheel hub	2
1W2L 5C	2
Metal shaft	2
Electric plug	8
Wire	1 (10 Meter)

Appendix H: Circuit Diagram for transmitter side of Accelerometer based gesture-controlled robot



Appendix I: Circuit diagram for receiver side of accelerometer based gestured-controlled robot