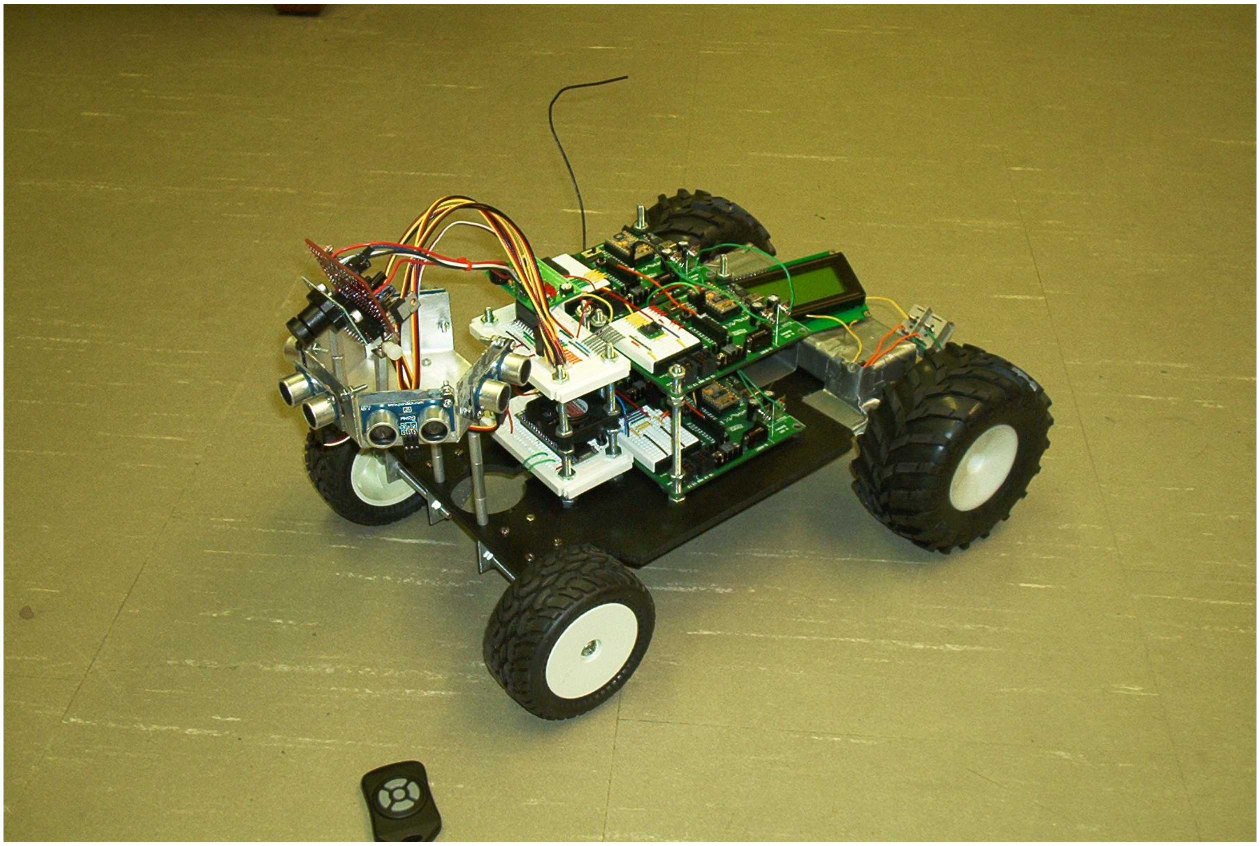
DESIGN OF A FIELD ROBOT FOR WEEDING PURPOSE



Nicolas Francoeur-Leblond

Bioresource Engineering, McGill University

2006

Presentation

This paper was submitted the 21st of November 2006 as a final project report for the course BREE 495 in the Bioresource Engineering department, McGill University. This project was first intended to the 2006 Field Robot competition in Stuttgart Germany but due to time delay, was postponed to future events.

The author would like to thank Dr. G.S.Vijaya Raghavan from the Bioresource Engineering of McGill University for is support throughout the conception of this project. The help and advises provided by Dr. Robert Kok and Dr. Ning Wang, also from the Bioresource Engineering department of McGill University, are acknowledged as well.

Finally, a special thank is extended to M. Samuel Price with whom that project has begun.

Abstract

International agriculture is nowadays governed by free-trade agreements, government subsidies and an endless race to provide the best prices on the global market. This intensification of agriculture has been possible to sustain, partly due to chemicals input in fields and a tradeoff cost of ecosystems degradation, in which human belong. Modern technology must be applied to solve such issues while keeping farmers competitive. A design of a robot is proposed to reduce herbicides input in the field to its minimum while in parallel tries to provide cost, time and fuel economies.

Table of Contents

I. INTRODUCTION

Establishment of Need and Benefit to Agriculture

1.1 Problem Statement

Chemical inputs in agriculture can lead to significant pollution problems. While the issue of fertilizers load effect on the environment is known, pesticide pollution is often forgotten.

A study conducted in Québec has shown that 31% of the monitored wells presented traces of aldicarb and atrazine (Giroux, 1995). In most of cases, concentrations were below the health safety threshold, but studies show resilient chemicals such as organophosphates and their metabolites can cause long-term health effects from bioaccumulation (Une Nation Toxique, Laidlaw Foundation, Nov 2005). In addition, before drinking water contamination, the largest risk to public health will always remain with the direct exposure to the chemicals from drift during application.

Wildlife is most susceptible to pesticide pollution. It is indeed recognized that while subsurface drainage prevents chemicals from reaching deep groundwater and wells, it contributes however to surface water bodies pollution (Berryman et Giroux, 1994). In 1993, the Châteauguay River, a basin intensively farmed, 35% of water samples contained level of pesticides above wildlife protection levels and 8% of fishes were affected with mal-formation. The scenario was even more dramatic in the Yamaska river basin, where bullfrog population were devastated by mal-formation and even decimated from certain tributaries such as the Chibouette River (Découverte, SRC, Oct 2005).

It is to be noted that herbicides such glyphosates (Roundup) are more often used in Québec and considered safer. Nevertheless, high incidence of toxic fungi in cereal crops has been recently attributed to Roundup herbicide (New Scientist, 2003).

Other places in the world may not have same concerns on pesticides threat. For instance, atrazine was formerly banned in the United-States because of the concerns just mentioned. However the U.S.A. Environmental Protection Agency as just (June 2006) re-authorized the use of this herbicide, due to its effectiveness and its cheap costs for farmers. This brings another issue, the dependency of farmers on pesticides and fossil energy input for weed control and other operation of the field. When these expenses are minimized or eliminated, the competitiveness of a farmer can increase significantly.

Fig 1.1 Conventionnal herbicide spraying

1.2 Objectives

There are two main factors to look at when trying to solve the main issue brought up in introduction. The first one is the total area covered by a conventional herbicide sprayer (generally the whole field) versus the actual area covered by weeds. By designing a device that would effectively spray patches of weeds individually hence reducing the area to be covered, a substantial economy can be achieved with less use of herbicide and associated time and energy costs. Not only the load of chemicals reaching water bodies would be reduced, but also expenses to farmers for weed control.

This project aim is to design such a device in the form of a weeding robot. The robot as to be able to navigate through corn rows recognizing and spraying weeds patches along the way. Also, by giving a weeding robot a targeting ability, reducing the distance between nozzles and the ground is a possibility and would have for effect to drift particles.

1.3 Scope of Project

The objective of this project is very similar to the Field Robot Competition, held since 2003 at the University of Wagenigen, Holland and Stuttgart, Germany in 2006. The idea was to compete at the 2006 edition of the event. The guideline on rules and regulations provided by the University of Hoenheim, host of the contest, were carefully observed and accounted in the design. The robots participating at this event had to perform the following tasks:

- Navigate between rows of corn (straight or curved) while keeping the straightest motion as possible. The above had to be accomplished as fast as possible. The velocity of the robot was therefore taken into account.
- Recognize dandy-lions simulated by yellow golf balls.
- Find a designated target, using image processing, bearing or both.
- Any free-style operations routine or implement can be added on the robot.

Fig 1.2 The eye-maize at the Field Robot competition

Further inspired by the ASABE ¼ Scale Pulling Tractor Competition, the scope of the project was enlarged to make the robot a marketable product by considering the following factors:

- Energy, cost and time efficiency
- Serviability and user friendliness
- Manufacturability

II. RESEARCH

2.1 Data Acquisition

- Dr. Ning Wang from the department of Bioresource Engineering (McGill University) was a great help. Firstly through the course Instrumentation & Control where the fundamental of electronics, logic circuits and microcontroller programming were learned.
Secondly, she provided a path to follow in the design process identifying type of sensors that could meet the criteria set above.

- Extensive research was conducted on the web to find similar projects and their features. The field robot competition web site *fieldrobot.com* has a lot of pictures and description of robots from the previous years that were a great source of inspiration.

- Information on most microcontroller, peripherals and their respective datasheets were also available on the web.

- More advanced programming skills for the selected microcontroller language were learned through literature and more specifically with the book, *Basic Stamp 2p Commands, Features and Projects* (Claus Kühnel & Klaus Zahnert, Parallax Inc. Press).

- The surrounding environment and the conditions in which the robot is to perform are constantly updated on the Field Robot web site. From sowing to the actual competition date, emergence of corn is carefully monitored. Corn being scouted can be grown on no till concept or bared crusted soil, and it has an inter-row width of 75cm and plant spacing of 20cm to 25cm. Nevertheless, soil and field surface will be considered for worst case scenario in this design.

Fig 2.1 The BoeBot

2.2 Experimentation

2.2.1 Prototype

During the course Experimentation & Control, the group of students had the opportunity to assemble, during a lab a small robot kit from Parallax™ called the BoeBot. It is a small aluminum chassis with two servomotors for displacement, two infrared sensors and 2 whiskers for location of obstacles. The BoeBot works with a microcontroller Basic Stamp 2 (BS2) and the goal of the lab was programming of this microcontroller to obtain an "intelligent" little machine.
We further developed the BoeBot into a more interesting prototype adding a camera for color detection (*CmuCam1 Appmod)* and a scanning sonar for range finding. This sonar was made of an ultrasonic sensor (*Devantech*) mounted on a servomotor (*Futaba)*. A compass (*Appmod)* was also used to help the robot turn precise angles. Two BS2 microcontrollers connected through serial communication were used to share the load of sensors.

2.2.2 Results

- We have discovered the camera has a very high computational requirement. Thus one BS2 was strictly dedicated to the camera while the other BS2 was hosting all other components. Due to the computing limitations of the BS2 and its ability to perform only one task at a time, the robot's motors were constantly interrupted when other sensors code were executed.

- However the speed at which one sensor is individually controlled seem to be adequate and almost instantaneous.

- The *Appmod* electronic compass had shown serious limitations as its degree of accuracy was only 45º. In addition, even at a descent distance from all other components, magnetic fields inherent to the circuit board would make the compass un-usable. When used alone however, it has shown great utility in doing u-turns.

- Servomotors are very easy to control. Only by varying the period of a 5V pulse with the function PULSOUT, we were able to control the angular speed of the servos.

- Overall, the goal of this experiment was to make the prototype navigate on a red paper lane with two continuous rows of green books on each side. The camera would compute the distance from its focal point to the red lane centroid while the scanning device would measure the distance from the robot to the books on both sides. Both information were compiled to give the servomotors an appropriate direction to follow. Although robot prototype could not move in a continuous fashion, still it was able to accomplish the tasks successfully.

- Both serial communication with the functions SEROUT/SERIN and 1 bit logic communication with the functions HIGH/LOW, between two BS2 were found to be a very effective way to spread all peripheral computing requirements over a larger processing unit.

2.3 Design Approaches

2.3.1 Microcontroller

The first step considered in designing the robot was its "brain", the microcontroller or the computer that would accomplish the required tasks. That is gathering information on the surrounding environment through sensors, analyzing this information and finally taking an adequate decision for navigation and weed detection. Based on Keith A. Prickett (Nov 21st 2005) work on "How to choose a Microcontroller for a design project" the following criteria were set in order to select the appropriate device.

- The most important factor is the power consumption. The microcontroller should have efficient sleep modes during which the chip has very low power consumption. 10hrs of autonomy for the application is suitable.

- Purchase cost is the other main factor, especially for limited budget. The suggested microcontroller price should range within $1 to $100.

- A sound understanding of all peripherals and features, the application will require and the number of input/output (I/O) ports available on the microcontroller for these peripherals.

- Finally, the speed of communication and speed of execution of the microcontroller should be adequate.

Further, Dr. Ning Wang suggested the following factors to be taken into consideration as well, for the microcontroller's choice: the programming language and an easy connectivity, to avoid tremendous time spent on developing the microcontroller's host board or peripherals compatibility circuits.

2.3.2 Peripherals for obstacle avoidance

- For the obstacle avoidance, the prototype developed in the experimentation stage had shown the standard ultrasound sensors were an excellent choice to sense surrounding obstacles. One unit however, enables the robot to take a reading at only one location at a time.

- Again inspired by the prototype results, a rotating device was imagined, consisting of more than one ultrasound sensor. This device would be mounted on the step motor considered for front steering (Fig 2.4) increasing the action of the sensors. In that fashion, it would have been like the robot had a head and could look around generating a map of the surroundings in polar coordinates.

- Ultimately we realized that a rotating device would be complicated. For each ultrasound sensor readings, it was necessary to have an associated angle relative to the robot. However both distance reading and rotation of the shaft couldn't be accomplished at the same time by the same microcontroller unit. Programming a logic decision for such a situation was found to be too expensive on computation. Overall the idea was good but since a row crop environment is generally uniform and does not offer any "surprise", such complicated mechanism is not practical enough from the cost consideration.

- Infrared range finders were also considered. However, due to their variability with light conditions and their need to be calibrated, that option was discarded.

2.3.3 Peripherals for weed recognition

Many similar projects found on the web uses complex image processing for both row navigation and weed detection. Algorithms are designed to recognize geometrical shape. This idea was discarded immediately because it was too complicated and constituted a whole project in itself.

The camera used in experimentation or a similar color detector was considered as the only option for recognizing yellow golf balls or other types of weeds.

2.3.4 Peripherals for orientation and bearing

The robot was first thought to be equipped with a *Melexis 90217 hall-effect sensor*. This device senses the frequency at which a magnet, installed on a rotating device, passes in front of it computing angular velocity. It would have been used to find the distance traveled by the robot. Conjugated with a compass, the robot could have known its precise location with respect to a reference point. The hall-effect sensor is usually accurate for high-speed rotations. For our application however, the angular speed was so slow that it would take too much time in the microcontroller's program for the sensor to take an accurate reading.

2.3.5 Size of the robot

- The usefulness and efficiency of the project would increase, to a certain extent, with more rows of crop being treated at a time. Most of the weeding robot prototypes that have been published on the web are of considerable size. Time being a crucial aspect in this project, a tight constraint should be put on the size.

Fig 2.2 Diesel powered robot (University of Wagenigen)

- Many small robots working together is a possibility often approached in literature. Upon detection of a patch of weeds, a robot could communicate its coordinates to an un-occupied team-mates. This idea however goes beyond the scope of this project.

2.3.6 Materials

Many materials are available and the following were considered: Aluminum, steel, wood and PVC, also in different shapes such as sheets, or small beams.

- By considering production of robots in hundreds of units reduces the ranges of materials. Beams must be joined together by bolts and more logically by welding adding the cost of a welder-fitter.

- Wood can certainly be an interesting material to work with. Certain prototypes observed in the research were made of wood. However because the robot will operate in outside conditions, wood was discarded because it naturally shrinks and expands with changing humidity levels.

- We found the easiest approach was by cutting and folding sheets of material. Steel was already discarded because of its high cost and its toughness, making it harder to work. Aluminum and PVC sheets were finally selected.

2.3.7 Propulsion & Steering

- Fuel propelled engine is a possibility, one prototype powered by a diesel engine has been developed by the University of Wagenigen (Fig 2.2). Dealing with two different sources of energy brings complexity to the project and a fully electrical system should be adopted.

- From the McGill department of Bioresource Engineering, Dr. Kok suggested the implementation of a walking or crawling robot using in-line servomotors or actuators. The reason for considering this was higher energy efficiency than rolling vehicles as well as an improved ability to deal with harsh terrain. Since speed was evaluated in the design, these were not considered.

- In the rolling type of vehicles, 4-wheel drive propulsion was judged unnecessary considering the energy aspect of the project.

- Tracks seem to be the most adopted locomotion type for robots at the Field Robot competition. A study conducted on the comparison between wheeled and tracked vehicles is compiled in the table below (Paul Hornback, *Armor* March-April 1998). From this, it can be concluded that tracked vehicles are the most suitable for off-road conditions.

However, this report shows that not only wheeled vehicle offer better energy economy but “intrinsically more reliable than tracked vehicles and, therefore requires less maintenance and supply support (spare parts)”.

The approach that was then considered was two independent rear motors propelled vehicle with a conventional steering at the front wheels. A *12V Unipolar Stepper Motor* from Parallax™ connected to tie rods was used to steer the front wheels.

Study Results	*Tracked Vehicles*	*Wheeled Vehicles*
Route Flexibility	✔	
Cross Country Mobility	✔	
Traction on Slopes	✔	
Road Speed		✔
Logistics		✔
O&S Costs		✔
GVW, Volume, & Payload	✔	
Maneuverability/Turning Radius	✔	
Transportability	✔	
Weight Growth Potential	✔	
Gap & Obstacle Crossing	✔	

Fig 2.3 From Armor, March-April 1998

It was later found that the back tires friction with the ground was considerable compared to the front tires traction. Subsequently to support the front wheel steering direction, rotation of the back wheels had to be differentiated. Overall, the direction of the robot was found to be governed mostly by the skid steering effect of the back wheels differentiation, rather than the front steering.

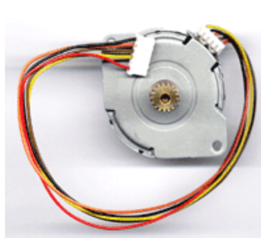


Fig 2.4 Stepper Motor

III. FINAL DESIGN

3.1 General Approach

3.1.1 Overview

- Due to time restriction for completing the physical design, one small size robot was the only possible option. The robotic vehicle will treat one row at a time and will have grossly the same dimension as a standard sheet of paper (8½” x 11”).
- The materials that were ultimately selected were ¼” thick PVC boards for the frame hosting all components. More complicated parts were purchased or manufactured with 1/16” aluminum sheet that was selected for its easiness of bending.
- Propulsion will be achieved with two rear end DC motors. Turning action of the vehicle will be achieved solely by differentiating rotation of the motors (skid steering). However, front wheels are left free to rotate in the vertical axis to offer less friction resistance while the robot turns.
- To answer the free-style criteria, pre-pressurized herbicide tank will be mounted on the robot. Under the control of the camera, a solenoid valve will open the flow from the tank to nozzles located at the front of the vehicle. This project does not focus on the spraying assembly and rather focused on the maximum weight of herbicide that will be loaded on the robot and the valve energy requirements.

3.1.2 Microcontroller

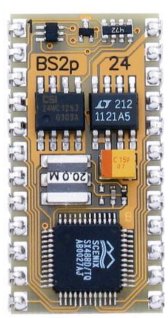
Based on research and experimentation, the Basic Stamp 2p (BS2p) from Parallax™ was selected. It is considered to be the perfect choice for beginner engineering projects over more complicated units such as PIC microcontrollers: “In a development environment, there really is no substitute for the Stamp in terms of the ease in testing concepts and this might be extended to actually implementing the product” (Peter H. Anderson, 98).

Moreover, the PBasic programming language previously learned for the BS2 BoeBot implementation is the same for the BS2p.

Another advantage with the BS2p is its compatibility with many and various type of peripherals, from motor control to pressure transducers. All peripherals in this project will be readily compatible with the BS2p and are found on the Parallax website thus bringing the advantage of dealing with less retailers.

Description

The Basic Stamp 2p (Fig 3.1) is a more advanced controller than the original BS2 used in experimentation.



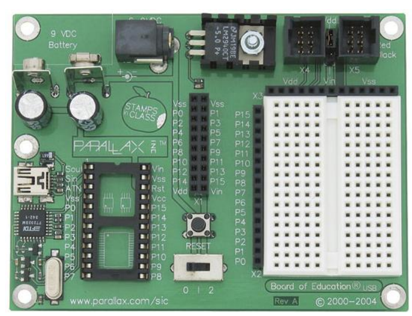


Fig 3.1 BS2p **Fig 3.2 USB Development board**

- It has a processing speed of 20MHz corresponding to a program execution speed of 12 000 instruction per second. Since in experimentation, the speed of execution of the original BS2 (8000instruction/sec) was found adequate when dealing with one sensor only, the BS2p should perform even better. Nevertheless, just like all of the Parallax Basic

Stamps, this version can only execute one task at a time. Therefore a minimum of peripherals connected to the BS2p is required for fluidity of operations.

- It has 38 Bytes of RAM and 16 I/O pins. This will affect the number of program variables. Again, considering the number of sensors used on only one BS2 during experimentation and the corresponding number variables, the BS2p offers plenty enough RAM.
- Running at 5VDC, it draws a current of 40mA and only 350µA while in sleep mode.
- With each BS2p purchased will also come a *USB Development Board* (Fig 3.2). This development board offers many advantages such as a USB port to download programs into the BS2p. It can be supplied with a 7.2 to 9VDC source but is equipped with a voltage regulator to ensure stability of the 5V source for the BS2p. It also offers a breadboard for circuitry and easy connectivity from peripherals to the I/O pins. Overall, this dramatically decreases the time spent on circuitry development.

3.1.3 Computing Approach

Information processing

Encouraged by the results of the experimentation, the selected approach is the multitasking. Since one BS2p can only process one task at a time, connecting all sensors and components on the same chip would result very discrete steps in the robot behavior and a lack of fluidity. To solve this issue, 4 BS2p will be used and synchronized together. Each group and types of sensors, or component, will be controlled by its own microcontroller (Fig 3.3)

The project approach is then divided into each processing unit that we will call “boards” in this report.

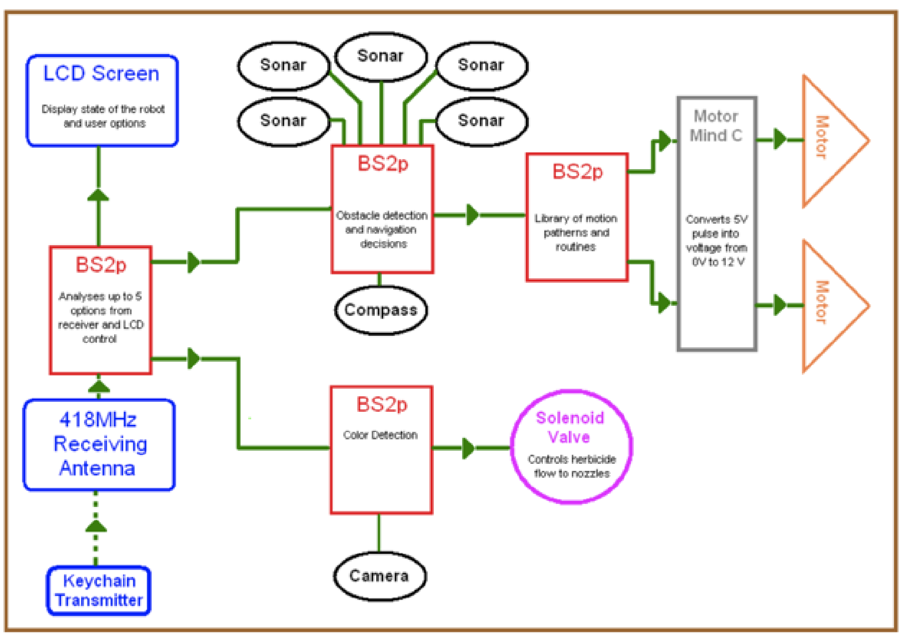


Fig 3.3 Block Diagram of the whole system. Each red rectangle represents 1 microcontroller Basic Stamp 2p. Green lines with arrows correspond to one-way communication while those without arrow represent communication that can occur in both ways. The dashed line represents wireless communication

3.2 Main Board

The user interface board is, for simplicity, called the main board. The purpose of this board is to provide the user with means to control and organize the operations of all other boards. On this board, a serial LCD screen and a receiving antenna are connected. This later is controlled by a remote keychain.

3.2.1 Serial LCD Screen

Purpose and Requirements

The principal purpose of the LCD screen is to display the options available to the user to control the robot. The options displayed on the LCD correspond to a respective condition (If-Then) in the BS2p program. The screen also serves to display the state of the robot that the user may want to know. The screen should be very easy to program since this is far from being an essential feature of the project.

Description

There was many LCD display on the Parallax website. The model chosen (# 27979) was the most convenient because of its serial interface.

Fig 3.4 Serial LCD screen

- It operates at 5V and draws a small 20mA.
- The serial LCD screen displays 4 rows of 20 characters and supports ASCII characters. It communicates at various baud rates but used at 9600 since this is the communication speed of the BS2p.
- Displaying text become very simple as it takes only 1 line of code sent at the correct baud rate over the serial port:

SEROUT pin, Baud_Rate, [“ Text message! “]

3.2.2 Keychain & Emitter/Receiver

Purpose and Requirements

The transmitter receiver is used as wireless push buttons to communicate intentions to the robot in opposition to the LCD screen which is a source for the robot to communicate with the user. The main functions of this assembly should be to start and stop the robot as well as scroll up and down a menu’s options. Such options could be:

- User defined or automatic control of the robot
- Initialization of camera for light conditions or color to be tracked

Description

The transmitter/receiver assembly is an antenna and a keychain from Parallax, communicating with radio frequency at 418MHz. This assembly has 5 (1-bit logic) channels, each connected to an input pin on the BS2p. When a push button is pressed on, is associated channel is set at 5V and a HIGH bit is sent to the respective input pin. This is convenient for user-defined control since 4 channels can used for the “forward”, ”backward”, ”right” and “left” directions and the fifth one for starting or stopping the robot.

Fig 3.5 Keychain remote

3.2.3 Assembly & Program

The main board program is quite simple. A menu containing 3 options is displayed through the LCD (rounded-corner rectangles). Each of the option is associated with one output pin (circles). The program waits for a HIGH bit signal from the receiver (represented by a rhombus), either to select the option or go down to the next option. These 3 options are:

- Initialize the camera, a HIGH bit is sent from pin 8 to pin 0 of the weed recognition board.
- Start the robot, in that case a HIGH bit is sent from pin 9 to pin 12 of the navigation board.
- Remote control mode, a HIGH bit is sent from pin 5 to pin 9 of the propulsion board. At this point, serial communication is established from the main board to the propulsion board so motors can be controlled directly from the keychain.

Each three boards receiving a HIGH bit signal have a condition in their respective program waiting for that signal. When an option is selected, the state of the robot is displayed on the LCD and the program waits again for a HIGH bit signal from the receiver. In that case the current option executed is stopped by sending a LOW bit to its respective pin.

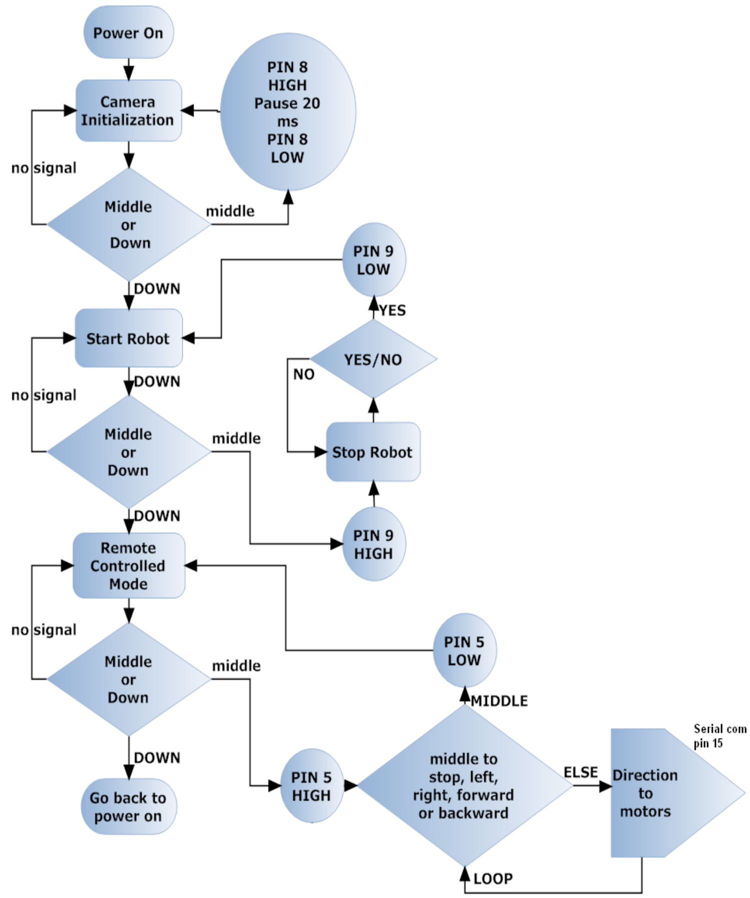


Fig 3.6 Main Board Program

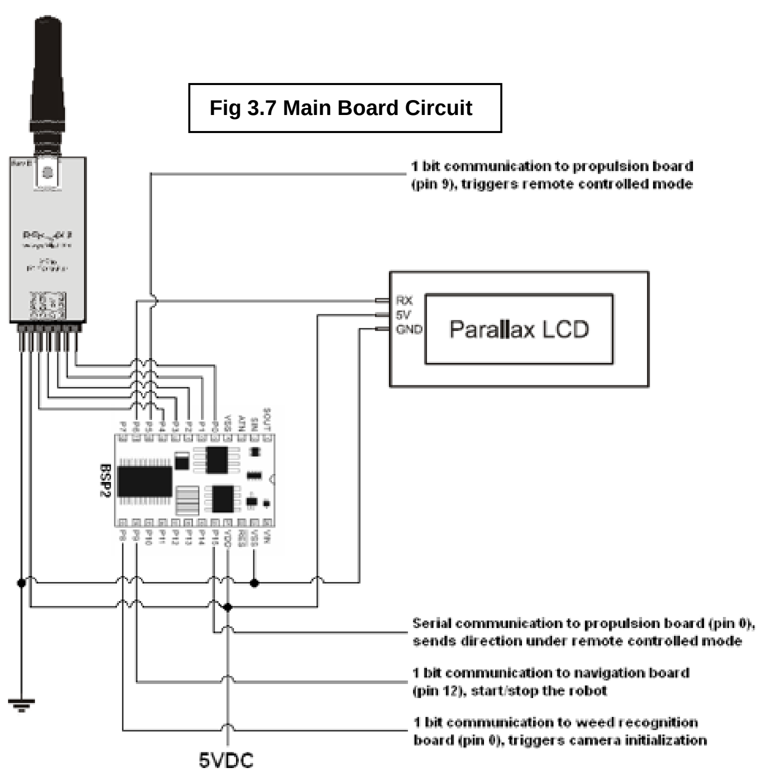


Fig 3.7 Main Board Circuit

3.3 Navigation Board

Based on previous approaches, fixed ultrasonic sensors will be the only way to detect the distance between the robot and the surrounding corn stalks. For special cases such a u-turn rotation at the end of a row, an electronic compass is used to support the navigation. Navigation operations are achieved by one stamp (navigation board). It analyses the data received from ultrasound sensors and the compass and takes a decision regarding the speed and direction of the robotic vehicle.

3.3.1 Ultrasound Sensors

Purpose and Requirements

An ultrasound range finder is used to locate the distance from itself to the nearest obstacle. It woks by emitting a burst frequency of ultrasound and by measuring the time at which the echo pulse comes back (Fig 3.8):

Distance = $344 m.s^{-1}$ / Time_to_Echo

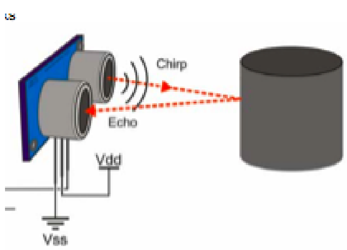


Fig 3.8 Ultrasonic sensor principle

One constraint should be the number of sonar used. Too many of these sensors will add to monetary, computing and energy consumption costs. They should be placed in order to obtain the best peripheral "view" of the rows of corn on both sides of the robot as well as front obstacle.

Description

- At the time of purchase, many of such sensors were available and potentially compatible with the BS2p. Parallax offered 5 units of the *Ping Sensor* for 100$ which was obviously interesting since most of sensors of this type on the market are sold for a minimum of $30 a unit. This represents a minimum economy of $50 and a maximum economy of $200, based on models considered (robotshop.ca).

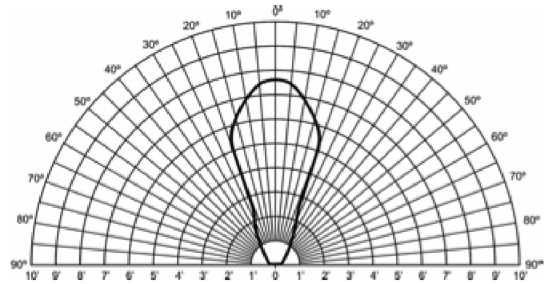


Fig 3.9 Ultrasonic sensor beam angle

- Operating at 5V, it draws a maximum current of 35mA. It has a typical range 3cm to 3m and will take a maximum of 20mS to accomplish 1 reading.
- In the BS2p program, this translates into a PULSOUT function, where the burst of sound is emitted, and a PULSIN function where the echo is measured back.

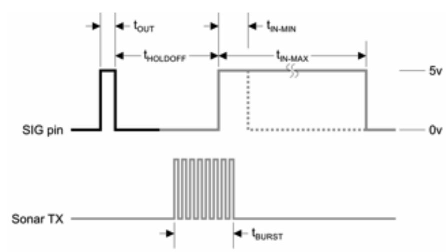


Fig 3.10 Pulse and echo length

- For easiness of manufacturability, it was decided that all ultrasonic sensors should not be scattered all around the robot. A chassis was built to group all ultrasound sensors and the camera together at the front of the robot. It is designed to maximize the width of action of these sensors and to ensure their beams don't overlap. Over five units, one will be placed up front to sense imminent collision with an obstacle. Two units will be placed on each side to sense distance with adjacent corn. If we look at Fig 3.9, we can see that the maximum beam angle of the sonar signal is 40º. The minimum angle set between 2 units should therefore be 40º. Moreover it is better to avoid directing a beam perpendicular to the rows of corn because in this fashion the sensor is most likely to sense a gap between 2 corn stalks. If however, the beam is directed at an angle towards the row, it increases the width of "view" and is less likely to fall on a gap.

3.3.2 Electronic Compass

Purpose and Requirements

The compass indicates the bearing. It is used to assist the robot when doing a u-turn, which is required at the end of each row.

- For the u-turn, the robot has to rotate 180° with a radius of 33.5cm. If the stamp program would pre-set to perform u-turns, it would not account for wheel slippage that may vary considerably from case to case, thus miss-orienting the robot.
- The compass makes sure the full 180° has been accomplished and the robot finds itself aligned with the next beginning inter-row.
- The compass should be isolated from other electrical components and induced magnetic fields to avoid false readings.

Description

Parallax offered a more advanced compass module, the Hitachi HM55B.

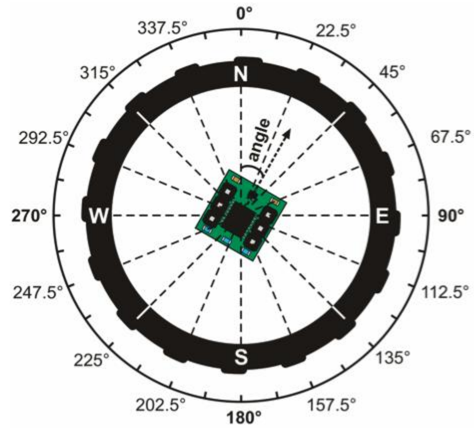


Fig 3.11 Electronic Compass

- It is a dual-axis magnetic field sensor, which has a precision of 1 degree and requires 40mS to provide an output bearing.
- Operating on 3V, a built-in resistor and regulator makes it compatible with the BS2p voltage. It typically draws 7mA of current. However since its peak current approaches 45mA that value only will be retained for battery selection
- The compass module is installed on the breadboard included on the BS2p development board, away from the motors magnetic fields. In addition, a calibration program is provided with this sensor to account for circuitry inherent magnetic fields.

3.3.3 Assembly & Program

Start/Stop Command

The program of the navigation stamp is executed when input pin 12 is set HIGH from output pin 9 of the main board. This situation occurs when, using the correct button on the keychain, the user has answered YES to the condition "Start the Robot " displayed on the LCD. On the contrary, as soon as the user answers YES to the condition "Stop the Robot", pin 12 is set back to LOW from the main board's pin 9 and the program stops.

Readings from the ultrasound Sensors

- The maximum time for which the echo will hit the sensor back is 19ms≈20ms, as specified in the datasheet (Fig 3.10). Since every sensor will operate one after the other it should be observed in the program that a pause of 20m/s is taken between every reading, for one ultrasonic sensor to fully accomplish its task.

- The main challenge is to adequately locate un-continuous barriers on both side of the robot. As opposed to navigation between 2 continuous walls where a distance reading is implicit, there is a possibility that a sensor takes a reading between 2 corn stalks, generating error. Two solution to this issue:

1. First, a tolerance is set at 40cm on each side. Thus if one reading occurs between 2 corn stalks, an implicit obstacle is created at 40cm.
2. Secondly, the values of distance used in the decision logic are actually an average of many readings taken in a short lap of time. In the numerical example below, note that sonar no1 is the front one, the odd numbers at left and the even numbers at right. Let $x_{1,2\ldots5}$ be the distance given from each sensor to the robot, i the number of readings per sensor and X, Y the relative right and left distance of the rows from the robot.

$$X = [\ \textstyle\sum_i (x_{2i} + x_{4i})\ ] / i$$

$$Y = [\ \textstyle\sum_i (x_{3i} + x_{5i})\ ] / i$$

Note that i can be varied in function of the environment. In the case of closely seeded crop a value of 1 may be appropriate while for corn seeded at a spacing of 20-25cm, i will take a higher value.

- X and Y are then compared. If X is greater than Y then the robot is more at the left, should turn right and vice versa. However depending on the magnitude of the difference between X and Y the vehicle has to perform appropriate degrees of turn. Let Z be that magnitude.

$$Z = |\ X - Y\ |$$

To certain ranges of magnitude, we assign a precise degree of turn. Therefore, 4 zones are set, 0 to 5cm being a zone of straight motion while in opposite, more than 30cm being a very hard turn.

- The front sonar works pretty much the same way by doing an average of many readings and setting buffer zones. For instance, any distance less than 20 cm signifies a backward motion whereas a meter or more indicates full speed.

The speed and the magnitude of turn are combined and assign a number. This number is sent through serial communication using the command SEROUT from pin 13 of the navigation board to pin 14 of the propulsion board.

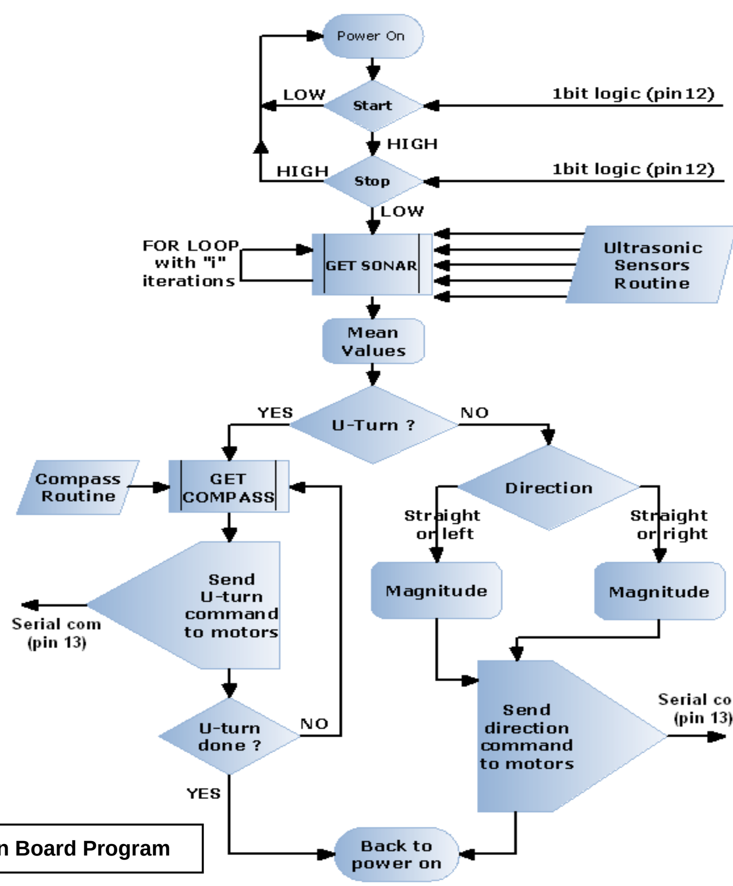


Fig 3.12 Navigation Board Program

Readings from the compass

- Note that the source code for this sensor reports angles in brad, a measure dividing the quadrant in 256 sections. That allows us larger angle increments, easier to locate by the robot. Further, not converting the bearing into degrees avoids rounding error.

- If all sonar should indicate a value of more than 2m, then the routine for u-turns is initiated. A first reading from the sonar is called the reference. The robot is given a motion routine corresponding to a u-turn. At each program increment, the bearing is evaluated and as long as the angle from the reference is no greater than, 180º the u-turn motion is continued.

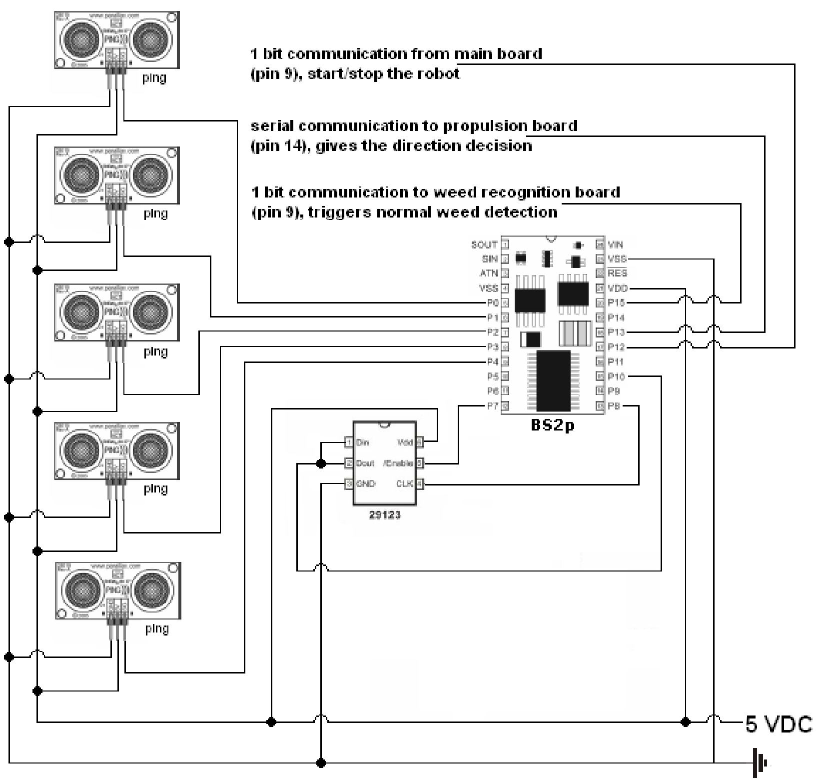


Fig 3.13 Navigation Board Circuit

3.4 Weed Detection Board

One BS2p will control a camera and a solenoid valve driver (weed detection board). This color-tracking device is used to detect any kind of color selected by the user, most logically weed's color. In the Field Robot competition application, yellow golf balls would be tracked as to simulate dandy-lions. Upon recognition of the selected color, the stamp controlling the camera will activate the valve driver, the latter controlling the solenoid valve leading to the flow of herbicide in the nozzle. The camera has a high computational requirement. Connecting other sensors or components to the weed detection board, could considerably slow down the camera's operation thus reducing the amount of land that can be scanned.

3.4.1 Camera

Purpose & Requirements

- The camera should be effective in tracking pre-defined colors. Following the field-robot competition regulations, that color can be yellow to simulate dandy-lions, or the color could be green when tracking general weeds.
- It should also be easy for a user, to re-configure the camera for changing light conditions or a new color to be tracked.
- The camera should have a descent window of view while still having the ability to distinguish small color blobs. However, if the color to be tracked is green, boundary in the field of vision should be set to avoid recognizing corn instead of weeds.

Description

Fig 3.14 CmuCam1

Like any other sensors in this project, the camera has been chosen from the Parallax website as a devise supported by the BS2p. However, being the only choice available in this category, the *CmuCam1 Appmod Vision System* (Fig 3.14) has been chosen for this purpose. This camera, manufactured by Seattle Robotics™ is a SX28 microcontroller interfaced with an OV6620 Omnivision™ CMOS camera.

- The camera has ¼" diameter lens with a vision array of 352 x 288 pixels. It gathers the mean color and variance as well as the centroid's coordinates of the color blob.
- At 9600baud serial communication speed with the BS2p, the camera gathers 17 frame/second of images in 8 bits format.
- It functions at 5V and requires 250mA of active current from the board rather than from the stamp directly.
- The camera is mounted on the same chassis used to hold the ultrasound sensors. It is mounted 30 cm above the ground giving it a window of view of about 100cm x 50cm at best with a minimum size of color blob recognized around 5cm x 5cm in dimension. This is important since the corn on each side doesn't appear on the camera's sight and thus won't recognize the corn as weeds
- Buttons on the mounting board are no longer used in this design.

3.6.3 Solenoid Valve

Purpose & Requirements

- The valve is used to turn on and off the flow of herbicide to the nozzles and it obviously needs to be chemical resistant.
- Information on nozzles for herbicide application was obtained from the Cooperative Extension Service from University of Kentucky. The standard flat fan nozzle has an operating range of 60 to 30psi. The valve must be rated within that range of pressure.

Description

A valve that had the most interesting features was the X-valve from Parker Hannifin Corporation. This miniature 8mm diameter valve operates on 12VDC at only 1W of power consumption and is rated at 50psi.

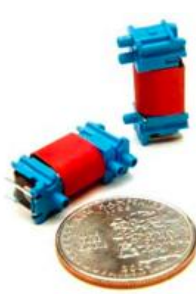


Fig 3.15 Solenoid valve

3.4.2 Valve Driver

Purpose & Requirements

The valve cannot be driven by the BS2p since the voltage from an output pin, as well as in an input pin, may not exceed the supply voltage of the stamp (5V). The valve driver will allow the BS2p to switch on off higher voltage avoiding current to flow back in an I/O pin. Parallax suggests the use of a NPN-transistor or a darlington type transistor. A darlington array ULN 2803, previously purchased for experiments with the stepper motor, is used for this purpose.

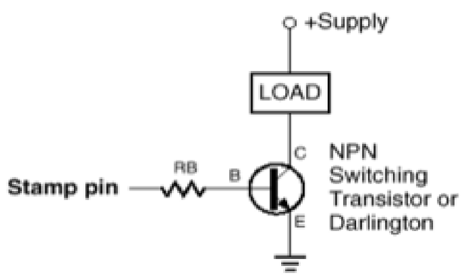


Fig 3.16 Driver circuit as suggested by Parallax

3.4.3 Final Assembly & Program

Initialization

When the main board output pin 8 sends a HIGH bit signal to the weed detection board input pin 0, the camera initializes for ambient light conditions and desired color to track. This HIGH bit signal is sent when the user answers a "YES" to the condition "Camera Initialization" by pressing the proper button on the keychain. Time to execute the initialization program is about 17 sec during which all other operations on the main board are interrupted.

Color Recognition Logic

The second part of the program cannot be executed if the initialization hasn't occurred. However the robot can still navigate without the camera since both systems are independent. Once initialization has been done, if the user answers "YES" to the condition

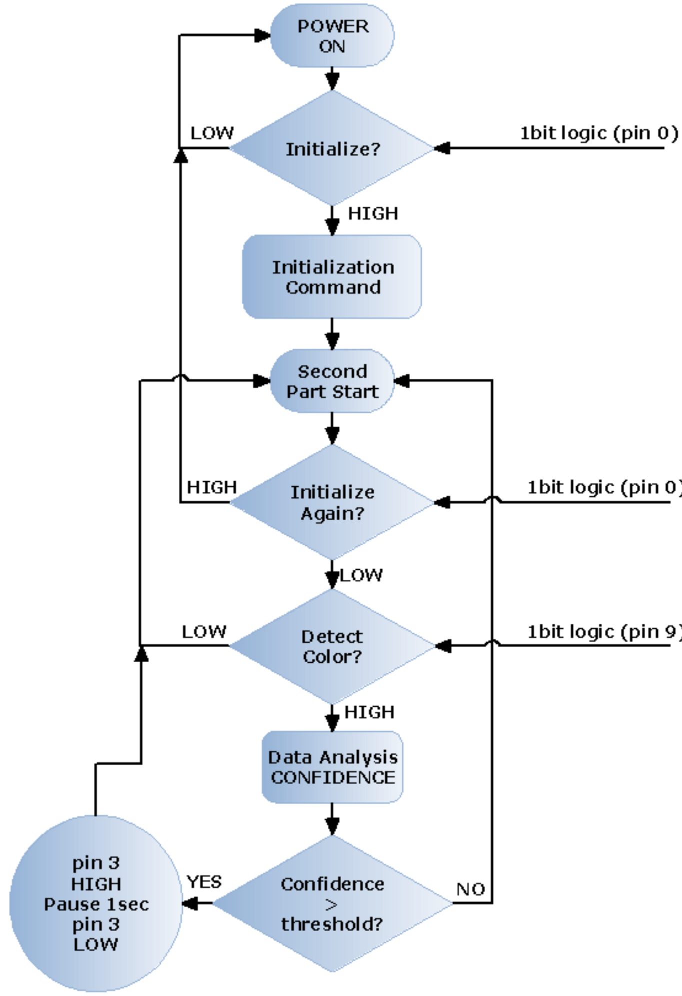


Fig 3.17 Weed detection board program

"Start the Robot", a HIGH bit signal is sent from output pin 15 of the navigation to input pin 9 of the weed detection board and the camera starts tracking colors. This program can be stopped at any time by setting a LOW bit signal between the two last mentioned pins. Again this is achieved by answering "YES" to the condition "Stop the Robot". The camera returns the BS2p, with an array of 10 parameters in a Byte data format. One of this parameter is the confidence at which the camera thinks to see the desired color. Confidence is based on the mean and variance of all pixels compared to the color saved during initialization. A simple condition is set in the program: if the confidence is larger than a threshold, pin 3 connected to the valve driver is set HIGH switching the valve on in the second circuit. Note that this threshold may be changed according to the environment.

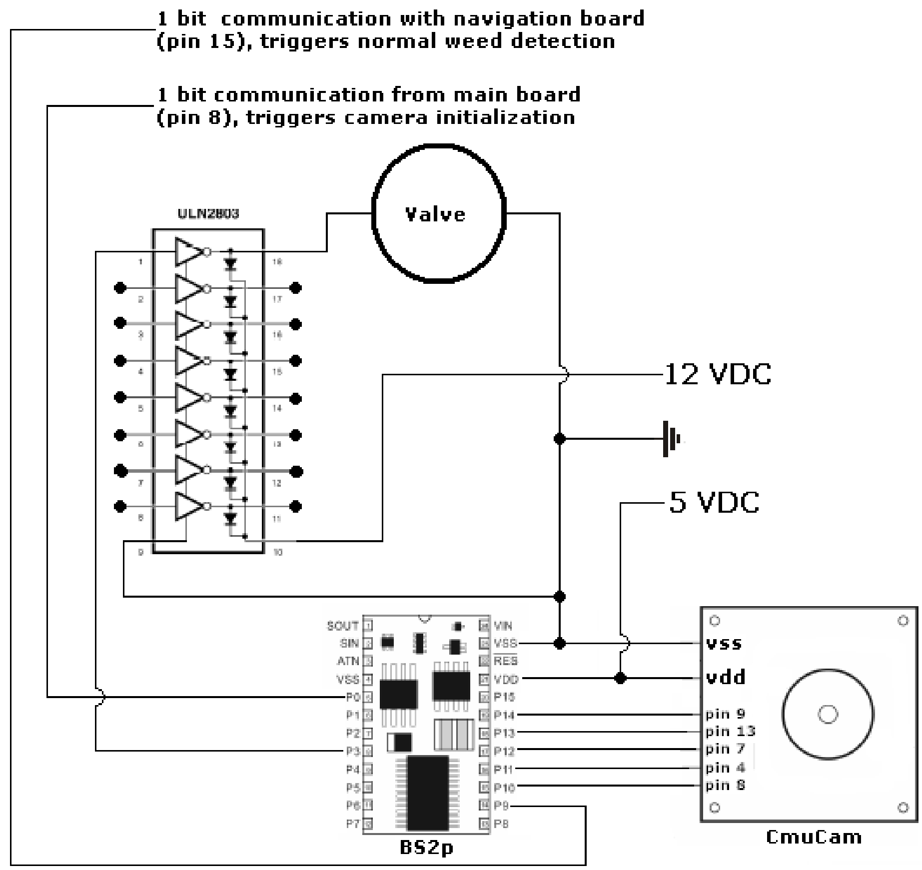


Fig 3.18 Weed detection board circuit

3.5 Propulsion Board

Another BS2p will be responsible for motion (propulsion board). This board contains a library of different locomotion routines. Decision taken by the navigation board is sent to the propulsion board, to be assigned a movement routine. The vehicle will move using DC motors in rear propulsion. The propulsion board will thus be connected with a motor controller, in order to convert the motion sub-routines contained in the BS2p program into rotation. Again, like the weed recognition board, the stamp program should be dedicated to the motors only. While another routine should be performed, no signal would reach the motor controller as a result of un-continuous movement.

3.5.1 Motor Controller

Purpose & Requirements

5V servomotors could be directly connected on the BS2p development board. However these types of servomotors are not suitable due to their relatively low torque and speed. DC motors are perfect for this type of application but require a motor controller. A motor controller is composed of one or more H-Bridges (one H-Bridge per motor) and a controller.

The principle used is the *pulse width modulation.* That is, serial format signal or pulse-signals, normally sent by the BS2p to servos, turns on and off the switches of the H-Bridges allowing the current to flow in the DC motors (Fig 3.19)

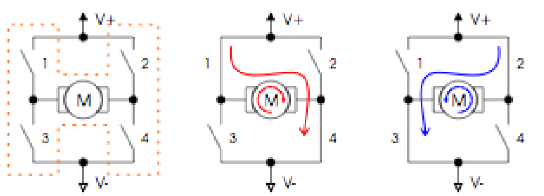


Fig 3.19 H-Bridge

The width and the period of the pulse are converted by the controller into a rate at which these switches open and close hence into an angular speed and direction of the motors (Fig 3.20)

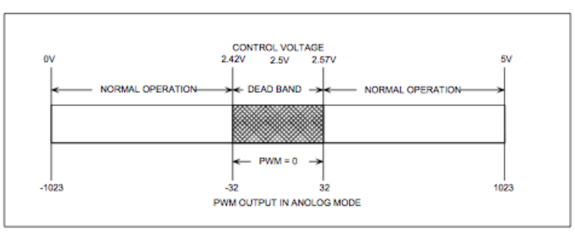


Fig 3.20 Pulse Width Modulation

- The motor controller chosen should be compatible with one of the PBasic functions such as SEROUT/SERIN or PULSOUT for either serial or R/C communication.

- They should sustain a descent current delivered to motors in order to obtain the highest torque possible.

Description

Three controllers were considered: the *Motor Mind C* (MMC), *Motor Mind B* (MMB), both from Solution Cubed™ and the Pololu™ *Micro Dual Serial Motor Controller* (MDSM). All of these controllers can support 2 motors, and their major features are displayed in the chart below (Fig 3.21).

- Two motor controller units could have been used to increase the power delivered to each motor, however we simply opted for one MMC (Fig 3.22) unit since it represents the lowest cost to current ratio and supports an interesting range of voltage (6V to 24V)

- A cooling fan must however be included in the design to protect the MMC from over-heating.

- Three communication modes (analog, R/C and serial) can be experimented with the BS2p. R/C mode was retained since it requires a simple 5V pulse from the BS2p obtained with the PULSOUT function the same way servomotors are controlled.

- It has been found through experimentation that a pulse width of 1700µs corresponds to the dead band and converted into a null speed. A pulse width longer than 1700µs value will lead to a reverse motion while a shorter width will lead to a forward motion.

	Purchase cost	Type of communication	Max current
MMC	$55.00	serial, analog and R/C	6A peak / 4.5A continuous
MMB	$29.00	serial	3,5A peak / 2A continuous
MDSM	$30.00	serial	2A continuous

Fig 3.21 Motor controller comparison

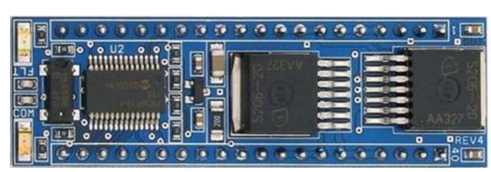


Fig 3.22 The MMC

3.5.2 Motors

Purpose & Requirements

- The motors are to be located at the back of the vehicle and underneath the whole chassis. The later is to avoid, as much as possible, the motors magnetic field from influencing the circuitry.
- Obviously, motors are to propel the vehicle forward. Moreover, by differentiating their rotation, on each side, motors will be used also for steering purpose.
- The motors obviously must be strong enough to carry the robot's weight under loose soil condition without stalling. Note that the driving wheels dimension will also adjusted in consequence.
- The robot should be able to move at a descent speed (rpm) while not biasing the sensors readings.
- Finally both motor used should respect the current limitation of the motor control chip at stall torque.

Description

- It has been found that 12V motors offered the highest torque for the permissible current allowed by the MMC. The following sheet (Fig 3.23) includes some of the motors considered. Note that the diameter of the wheels selected (section 3.81) and used to compute the linear speed and load is 5".
- The type of motor chosen is a *GHM-02* from Lynxmotion™ first because it respects the current limit authorized by the MMC and because it offers the highest torque to current ratio among all motors considered. The consequent linear speed for the robot was judged satisfactory. From the next graph (Fig 3.24), we can observe that the highest efficiency is achieved at a delivered torque around 1.8kg.cm

Motor	No Load Linear speed (cm/s)	Maximum Load (kg)	Maximum Current (A)
GHM-12	1.56	3.15	7.6A @ 12V
GHM-02	0.80	2.79	3A
GHM-01	1.45	1.4	3A
GHM-04	1.16	2.25	7.6A @ 7.2V

Fig 3.23 Motor comparison

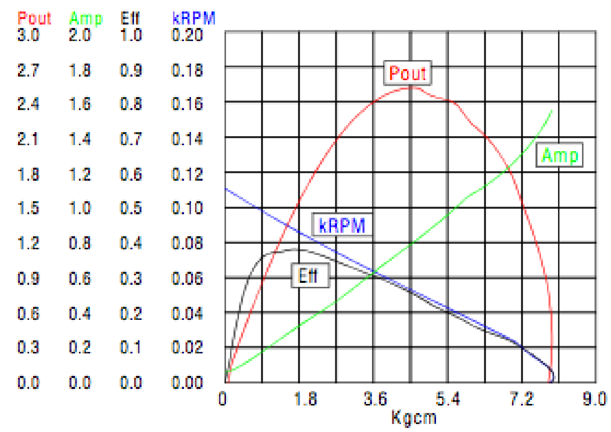


Fig 3.24 GHM-02 motor description chart

3.5.3 Assembly & Program

- The propulsion board constantly checks through the SERIN function at pin 14, the decision of the navigation board. For each decision received, the propulsion board contains a locomotion routine sent to the motor controller with the function PULSOUT through pin 5 and 6. An example of such routine for a hard left-turn would be:

```
For x = 1 to 10
        PULSOUT 5, 1500
        PULSOUT 6, 2000
        PAUSE 5
NEXT
```

At pin 5 and 6 is sent a pulse of 1.5ms and 2ms respectively. These single pulses are embedded into a FOR loop with a PAUSE function of 5ms for two complete pulse signals of period of 6.5ms and 7ms respectively. This would finally translate into a

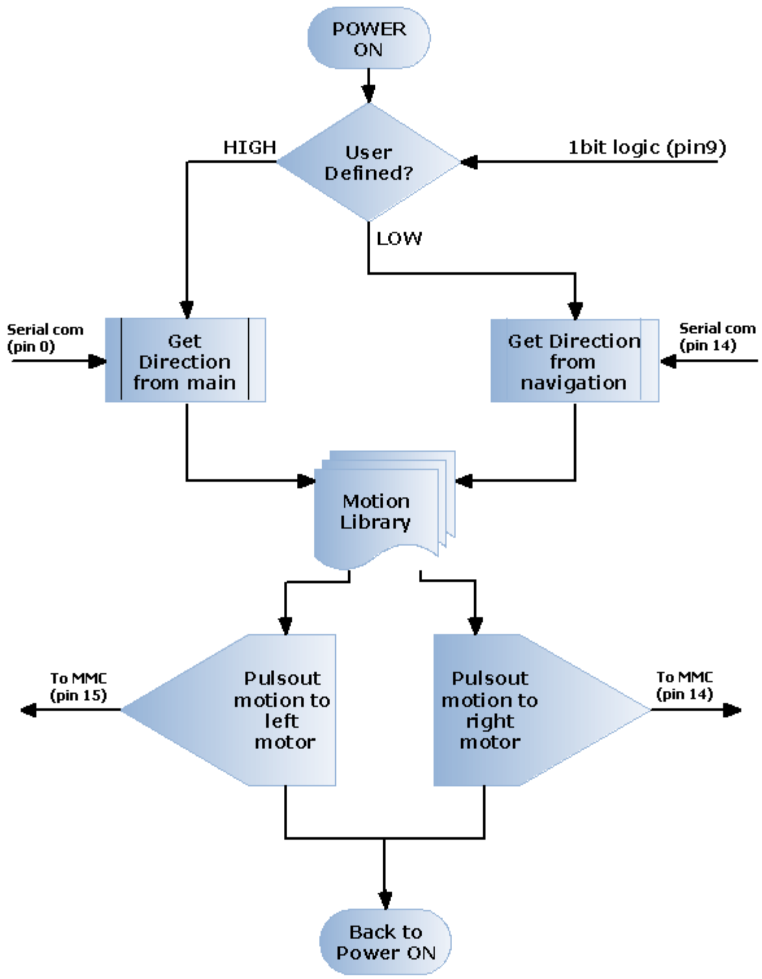
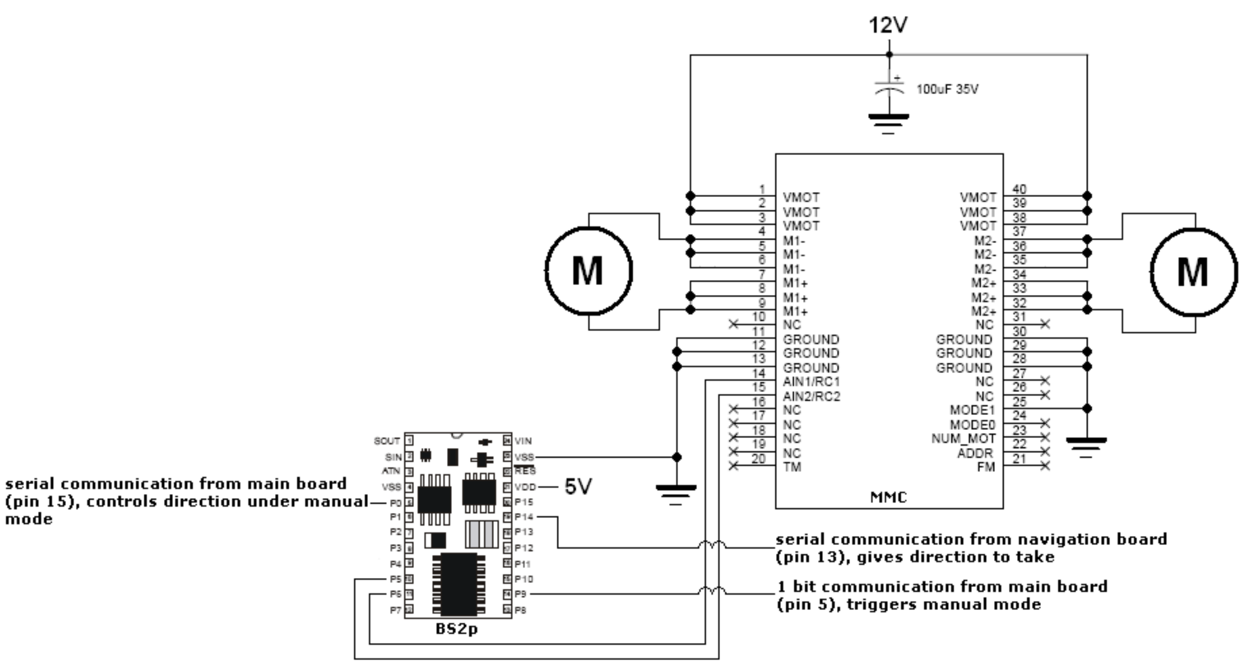


Fig 3.25 Propulsion Board Program

forward rotation of one wheel and a backward rotation of the other wheel for a full turn. As mentioned before, the pulse's width and the period's length will vary the speed of the motor while the amount of iterations done in a FOR loop will vary the duration of the of the rotation. After a FOR loop is completed, a new decision from the navigation board is evaluated.

- If the propulsion board received a HIGH bit on pin 9 from the main board, then all the routines become controlled by the keychain. If pin 9 goes back to a LOW state, the propulsion board falls back under control of the navigation board.

Fig 3.26 Propulsion board circuit



3.7 Power

3.7.1 Battery

Purpose & Requirements

- The battery is obviously meant to power the entire robot. It has to provide enough power to all circuits in order for the robot to work properly and for a descent amount of time.

- Moreover, the robot has to run on 2 different circuits, 12V for the motors, the solenoid valve and the cooling fan as well as 9V for all the BS2p development circuit boards.

Fig 3.27 Battery

Description

The total maximum current that can be drawn from the battery was cumulated as show on the table below.

PING	35mA @ a time
Compass	45mA
Boards	160mA for 4 BS2p
Camera	250mA
Motors	3A for 2 motors
LCD	20mA
Solenoid valve	80mA
TOTAL	3.518A

Fig 3.28 Accessories current consumption

A battery of 12V and 7A/hr was selected (Fig 3.27) . Considering the maximum current drawn, the robot is expected to run for about 2hrs before it needs to be recharged. The weight of the battery is 1.5kg

3.7.2 Voltage regulator

The LM2940CT-9.0 from national semiconductor, found on the Parallax website, provide the 3V drop required between the 12V battery and 9V circuit boards.

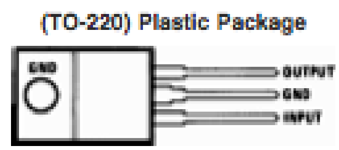


Fig 3.29 Voltage Regulator

3.8 Structural Parts

3.8.1 Wheels

Purpose & Requirements

The back driving wheels were chosen in order to reduce sinkage in loose soil as well as providing the maximum traction.

- A large diameter wheel will induce higher resisting torque on the motors shaft while smaller wheels will result in higher sinkage.

- The wheels should also be selected such it is easy to connect to the motors shaft.

- Note that the front wheels are not subjected to the same requirements since they are free spinning. Nevertheless, their contact area with the ground must be satisfactory enough so they don't dig in the ground as the robot moves forward.

Description

Selected back wheels, from Lynkmotion™, were the largest ones available. These off-road heavy-duty treaded wheels are 5” in diameter and 2,5” wide which provide good contact area with the ground.

Hubs were also purchased in order to directly connect the wheels to the motors shaft.

Fig 3.30 Rear wheels & Hubs

Front wheels however are a bit smaller in size and were selected due to their lower price compared to the back wheels. Hubs weren’t purchased and brackets described in the frame section were made to support them.

Fig 3.31 Front wheels

3.8.2 Traction Analysis

The final weight of the robot will ultimately depend on the weight of the battery and the herbicide tank. A traction analysis is done in order to help in the selection of an optimum battery size and herbicide volume that can be carried by the robot. Following the motors specification, the maximum torque that can be applied to one motor shaft ranges around 8kg.cm That is, with 2 motors and 5” diameter wheels, the maximum rolling resistance that can be applied to the wheels is 2.8kg (Fig%%%). Designing for worst case scenarios, traction analysis was inspired by similar work on lunar regolith (Sines et al. 1990), which is a cohesiveless and loose soil. Soil rolling resistance is a function of wheel sinkage and weight on driving wheel axis:

$$R = [\ (3.m.g) / (\sqrt{2}.r)\ ]^{4/3} / [2^{7/3}.(k_c + b.k_\phi)^{1/3}]$$

Where:

R: is the rolling resistance (N)

m: is mass over the rear wheel axis (kg)

g: is the gravitational constant (9.81 m.s^{-2})

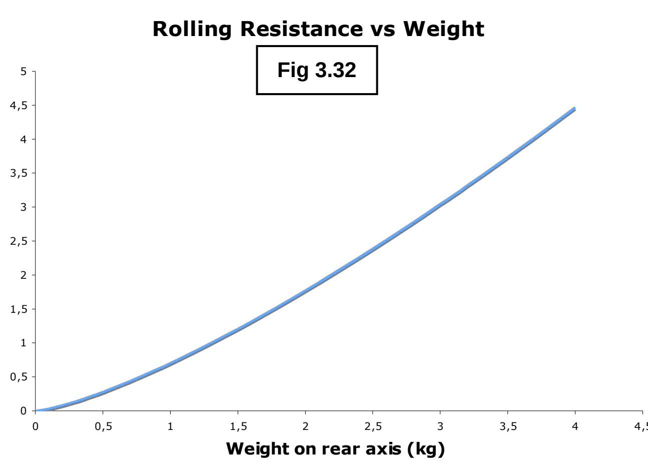
b: is the wheels width (m)

r: is the radius of the driving wheels (m)

k_c: is the cohesive modulus of soil deformation (kN/m^2)

k_ϕ: is the frictional modulus of soil deformation (kN/m^3)

Using a cohesive modulus of soil deformation and a frictional modulus of soil deformation of 3.5kN/m^2 and 8.1kN/m^3 respectively, results for this equation were plotted in excel (Fig 3.32). Based on this, the weight allowed at the back of the robot shouldn’t exceed 2.5kg.



3.8.3 Maximum weight

Based on the traction analysis, the maximum weight that can possibly be transported on the robot rear axis is 2,5kg. With a battery weight of 1,5kg only 1 kg of herbicide can be transported. This value should even be reduced to 500mg for safety measures. Knowing the dilution rate of herbicide and surfactant in water is low, we will assume their respective density negligible compared to water, leading to a liquid volume of 500mL.

3.8.4 Frame

- The frame should support everything from the batteries, the herbicide tanks to the sensors and the circuit board.
- Easiness of manufacturing and dismantling for repair or implementation is required.

Fig 3.33 Motor brackets

Description

- The frame is built out of a ¼" thick PVC sheet, ¼" by 8" in dimension: *Lynxmotion Pillow PVC Panel*.
- The sonar and camera chassis was made of aluminum.
- The motors are supported by brackets. Unlike Fig 3.33, for improved support, two brackets are used per motor.
- The plans were drawn on AUTOCAD (Appendix 2) and the layout printed on scale. This layout is pasted on the PVC and aluminum sheets so manufacturing at the machine shop becomes very easy.
- Only the vertical bend saw and the drill punch were required. 1/8" holes are perforated to fit the motors and the front wheel brackets. All circuit boards are stacked on ¼" thick rods inserted through the PCV board.

IV. COST ANALYSIS

4.1 Purchase Cost

Item	Stock#	Nb required	Price	Sub Total
From Parallax.com				
			$US	
Basic Stamp 2p 24-pin module	BS2P24-IC	4	$79 / Units	316,00
USB Development Board	28850	4	$65,95 / Unit	263,80
Motor Mind C	30001	1	$55,00 / Unit	55,00
Ping Unltrasonic Sensor	28015	5	$100,00 / 5 Units	100,00
Hitachi HM55B Compass Module	29123	1	$29,95 / Unit	29,95
ULN 2803A Darlington Array	500-00005	1	$1,50 / Unit	1,50
Parallax 4x20 Serial LCD (Backlit)	27979	1	$39,95 / Unit	39,95
418 MHz Receiver RX RF (SIP/wire/SW)	28004	1	$69,00 / Unit	69,00
418 MHz Transmitter RF (KC/loop/SW)	28005	1	$20,00 / Unit	20,00
From RobotShop.ca				
			$CAD	
Geared DC Motor GHM-02 Lynxmotion	RB-Lyn-06	2	$31,99 / Unit	63,98
PVC(1) Panel-black-BP-02-1/4" x 8" x 12"	RB-Lyn-92	1	$ 8,99 / Unit	8,99
All-terrain Wheels Lynxmotion 3,5"D x 1,75"L	RB-Lyn-21	1	$27,99 / Pair	55,98
All-terrain Wheels Lynxmotion 5"D x 2,25"L		1	$30 / Pair	30,00
Solution Cubed Cooling Fan 12V	RB-Sol-07	1	$ 20,99	20,99
Miscellaneous				
Valve				500,0
Battery				50,00
Tank, tubing & nozzles				50,00
Wires, voltage regulator, bolts & nuts				30,00
Manufacturing Costs				
5 hours/unit @ $50/hour				250,00

Fig 4.1 Purchase Costs. Note that American to Canadian currency conversion was done on the November 28th 2006

Grand Total: **$2356.3**

4.2 Recovery

- One of the most expensive herbicide on the market, Roundup Ultra, was sold at $21,74 per acre in 2001. For a farm of 300 acres, the price of herbicide climbs to 652$ per application. If we assume 50% of the ground area covered by weeds, the robot represents an economy of $300.
- The quantity of fuel usually required for spraying has been found to be 0.1gal/acre. The cost of diesel at the time the report was written was $2.25/gal. Again, considering a 300 acre farm, the use of a robot leads to fuel savings of 68$ for a total saving of around $400 per application.
- The return on investment for such a robot would consequently be around 6 applications.

V. DISCUSSION

5.1 Critical Analysis

At the time this report is submitted, the final robot has not been tested in field conditions. The robot responds well in lab conditions but both environments can lead to different robot behavior.

For instance, the robot has been tested on a flat and hard surface meaning the sonar has taken readings on a perfectly horizontal plane. The color of the surface was uniform. In field conditions, readings will most probably differ from experiment since the vehicle has do deal with much irregular terrain. Consequently, parameters may have to be changed in the algorithm of all boards for every component.

- For the sonar assembly, the tolerance or the size of the buffer zone may have to be changed. The number of readings per sensors “i” used to find the average may also be changed. The number of conditions that lead to a direction decision may be increased to fit eventual conditions for which the robot hasn’t been trained yet.

- The camera may need to be trained for more complex light conditions or be reinitialized regularly during a run to adjust for the position of the sun and shade. An infra-red cutoff filter is most probably required for sun lit conditions. Parallax suggests the use of neutral density 3 camera filter of simple pair of sunglasses. Weeds can have different shades of green (or yellow for dandy lions) while the camera is trained for only one shade. The confidence for the color tracked, on which the decision is based, may have to be lowered to account for those variations of color shades.

- The number motion routines given to the motor, in the propulsion board, will have to be increase with increasing number of conditions and decision outcome from the navigation board.

- Obviously the main board program needs to be modified to account for change among other boards.

- The power consumption hence the autonomy of the robotic vehicle is subjected to change. Indeed 2 hrs of autonomy is quite short compared to the 10hrs objective suggested during research (2.3.1)

- Photovoltaic cells could have been installed, cheap solar arrays from Robotshop.ca are available for 6 or 12V battery and up to 100mA.

- More energy efficient components could have been used which is a tradeoff for the cost. The Basic Stamp 2p was especially convenient for implementation but its energy efficiency leaves room for further consideration.

- Although the mass of the robot and the consequent rolling friction does not overcome the force applied on the ground by the motors, the later are not used at their rated torque and maximum efficiency. The robot would achieve better energy efficiency with a weight on the rear axis of 0.85kg, which is in turn difficult to attain because a descent battery has an obligated large mass.

- In opposite, weight allowed on the back axis could be increase if the selection of motor was based on their rated torque (best efficiency) rather than their stall torque (max current). However, there would still be a risk of over-heating the MMC motor controller.

- The size of the robot actually makes it un-attractive for farmers. Indeed, with an herbicide capacity of only 500ml this project is certainly not time efficient as the user constantly need to be prepared for re-filling the robot. That induces a greater risk to the farmer of being exposed directly to harmful chemicals.

- The robot does not possess any means (sensors) of knowing the quantity of herbicide left in its tank. The tank could be empty and the vehicle would still roam around in the field.

- However, for agronomic research farms, the weeding robot may be useful. Firstly because while a weed invasion problem might cause significant effects on crop experiments, researchers usually want to avoid stress from pesticides. Secondly because such crop experiments are usually grown in small plots where 500ml of herbicide is plenty enough.

5.2 Future Improvement

This project could be considered as an experimentation stage for a bigger project.

- The size of the robot would be scaled up to treat more rows at a time. One camera should be allocated for each inter-row treated.

- Another design would allow pulling an herbicide tank mounted on wheels rather than supporting the whole weight on one chassis.

- The robot could operate at night while being recharged during the day through a solar array. LEDs would be used with the camera as a constant brightness light source because of their energy efficiency.

- Furthermore, the robot could be larger and higher than the canopy level for solar panel area to be increased.

- The navigation as well as the weed detection could be achieved by image processing only, using geometric shape recognition. That would decreases the number of sensors thus a lower energy consumption.

VI. Conclusion

Multitasking and the use of many microcontrollers, has been found to be a very effective way to control the robot. This method can be used for any type of applications with any type of components if speed and fluidity of operations are required. If a more advanced robot should be designed, the same method would be applied.

The weeding robot technology is definitely full of promises. From this design, although not very efficient, we clearly see from this final prototype that intelligent machines have a future in the field. The steps of the robot design can be considered as a success. Now that small design imperfections appear with the final product, bringing corrections for an improved prototype becomes very easy.

Overall, this project was a great opportunity to learn the basics of object oriented programming, logic circuits and robotics in general. Most of this knowledge can be applicable in the future, to a multitude of other applications.

VII. References

- www.parallax.com

- www.robotshop.ca

- Environnement Canada, l'État de l'environnement au Canada – 1996, les produits agrochimique et le fumier.

- Environnement Canada, Réseau d'évaluation et de Surveillance Écologique. Nathalie La Violette, Yvon Richard, Jacques St-Onge, 1998.

- Stamp v.s PICs in commercial applications, Peter H. Anderson, Baltimore, MD, July 98 http://www.phanderson.com/stamp/commercial.html

- Conceptual Design and analysis of Roads and Road Construction Machinery for Initial Lunar Operations, Joel Banks, Keyanoush Efatpenah, Jeffrey L.Sines, University of Texas at Austin, 1990

- Herbicide price 2001 http://ipcm.wisc.edu/wcm/pdfs/2001/pricefin.pdf

- Weed Control: More Profit by the Patch. Montana State University Communication Services 1996. http://www.montana.edu/wwwpb/maxwell.html

- A Huge Victory for Corn Farmer - by Corn Farmers, July 2006. http://www.mocorn.org/updates/2006/Update072106.pdf

- Une nation toxique 2005, Laidlaw foundation http://www.environmentaldefence.ca/

Appendix 1 (Program code)

A) Main Board

```
'   {$STAMP BS2}
'   {$PBASIC 2.5}

' ----- [Variables]
down         VAR BIT
middle       VAR BIT
up           VAR BIT
left         VAR BIT
right        VAR BIT

TxPin        CON 6
Baud9600     CON 240



INPUT 0
INPUT 1
INPUT 2
INPUT 3
INPUT 4

accueil:
HIGH TxPin ' Set pin high to be a serial port
PAUSE 1000 ' Pause for Serial LCD to initialize
SEROUT TxPin, Baud9600, ["     WELCOME ",CR,
                         "            ",CR,
                         " press key to start "]

  DO
middle=IN0
up=IN1
right=IN2
down=IN3
left=IN4
IF down=1 OR up=1 OR middle=1 OR left=1 OR right=1 THEN menu
  LOOP
  END

Menu:
SEROUT TxPin, Baud9600, [12]
SEROUT TxPin, Baud9600, ["        MENU",CR,
                          " Up and Down: options",CR,
                          " Middle:      select "]
PAUSE 1000

  DO
up=IN1
down=IN3
IF down=1 THEN Option_1
IF up=1   THEN Option_3
  LOOP
  END


'-----------------------------------------------------------------------------
Option_1:
SEROUT TxPin, Baud9600, [12]
SEROUT TxPin, Baud9600, ["   START ROBOT ",CR,
                                      " Automatic Mode "]
PAUSE 1000

  DO
middle=IN0
up=IN1
```

```
down=IN3
IF up=1     THEN menu
IF down=1   THEN Option_2
IF middle=1 THEN Automatic
  LOOP
  END

'---------------------------------------------------------------------------
Option_2:
SEROUT TxPin, Baud9600, [12]
SEROUT TxPin, Baud9600, ["     CAMERA ", CR
                                    " Initialization"]
PAUSE 1000

  DO
middle=IN0
up=IN1
down=IN3
IF down=1   THEN Option_3
IF middle=1 THEN Camera
IF up=1     THEN Option_1
  LOOP
  END

'---------------------------------------------------------------------------
Option_3:
SEROUT TxPin, Baud9600, [12]
SEROUT TxPin, Baud9600, ["Manual mode"]

PAUSE 1000

  DO
middle=IN0
up=IN1
down=IN3
IF up=1     THEN Option_2
IF middle=1 THEN Manual
IF down=1   THEN menu
  LOOP
  END

'---------------------------------------------------------------------------
Automatic:
HIGH 9
SEROUT TxPin, Baud19200, [12]
SEROUT TxPin, Baud19200, [" ROBOT IS MOVING "]

PAUSE 1000

  DO
middle=IN0
IF middle=1 THEN arret_1
  LOOP
  END
'---------------------------------------------------------------------------
Camera:
HIGH 8
SEROUT TxPin, Baud19200, [12]
SEROUT TxPin, Baud19200, [" CAMERA INITIALIZING ",CR,
                          " place desired color " ,CR
                          "  in front of lens "]

PAUSE 1000

  DO
middle=IN0
IF middle=1 THEN arret_2
  LOOP
  END


'---------------------------------------------------------------------------
Manual:
```

```
HIGH 5
SEROUT TxPin, Baud19200, [12]
SEROUT TxPin, Baud19200, [" REMOTE CONTROLED "]

PAUSE 1000

  DO
middle=IN0
up=IN1
right=IN2
down=IN3
left=IN4

IF up = 1 AND left = 1  THEN dir = 5
IF up = 1 AND right = 1 THEN dir = 6
IF up = 1               THEN dir = 1
IF down = 1             THEN dir = 2
IF left = 1             THEN dir = 3
IF right = 1            THEN dir = 4
IF middle = 1           THEN arret_3
GOSUB send
  LOOP
  END

'-------------------------------------------------------------------------------
arret_1:
LOW 9
SEROUT TxPin, Baud19200, [12]
SEROUT TxPin, Baud19200, [" ROBOT STANDS STILL "]

PAUSE 1000

  DO
middle=IN0
down=IN3
up=IN1
IF middle=1       THEN automatic
IF down=1 OR up=1 THEN menu
  LOOP
  END
'-------------------------------------------------------------------------------
Arret_2:
LOW 8

PAUSE 1000

GOTO menu
'-------------------------------------------------------------------------------
Arret_3:
LOW 5
SEROUT TxPin, Baud19200, [12]
SEROUT TxPin, Baud19200, [" ROBOT STANDS STILL "]

PAUSE 1000

  DO
middle=IN0
down=IN3
up=IN1
IF middle=1       THEN manual
IF down=1 OR up=1 THEN menu
  LOOP
  END
'-------------------------------------------------------------------------------
send:
       SEROUT 15, 16624, [dir]
       PAUSE 10
RETURN
```

B) Navigation Board

```
'   {$STAMP BS2p}
'   {$PBASIC 2.5}

' -----[ I/O Definitions ]-------------------------------------------------
Pang            PIN      0
Peng            PIN      1
Ping            PIN      2
Pong            PIN      3
Pung            PIN      4

DinDout         PIN      9                    ' P9 transceives to/from Din/Dout
Clk             PIN      6                    ' P6 sends pulses to HM55B's Clk
En              PIN      7                    ' P7 controls HM55B's /EN(ABLE)

Reset           CON      %0000                ' Reset command for HM55B
Measure         CON      %1000                ' Start measurement command
Report          CON      %1100                ' Get status/axis values command
Ready           CON      %1100                ' 11 -> Done, 00 -> no errors
NegMask         CON      %1111100000000000    ' For 11-bit negative to 16-bits

x               VAR      WORD                 ' x-axis data
y               VAR      WORD                 ' y-axis data
status          VAR      NIB                  ' Status flags
angle           VAR      WORD                 ' Store angle measurement
reference       VAR      WORD


' -----[ Constants ]-------------------------------------------------------

Trigger         CON     13
Scale           CON     $0CD                    ' raw x 0.80 = uS

RawToCm         CON     2257                    ' 1 / 29.034 (with **)
IsHigh          CON     1                       ' for PULSOUT
IsLow           CON     0
time            CON     2

' -----[ Variables ]-------------------------------------------------------
rawDist         VAR     WORD

cm              VAR     BYTE
cm1             VAR     BYTE
cm2             VAR     BYTE
cm3             VAR     BYTE
cm4             VAR     BYTE
cm5             VAR     BYTE
sum             VAR     BYTE
sum1            VAR     BYTE
sum2            VAR     BYTE

counter         VAR     NIB
idx             VAR     NIB
speed           VAR     BYTE

main_sonar      VAR     BIT


INPUT 14

' -----[ Program Code ]----------------------------------------------------

Beginning:
DO
LOW ...
main_sonar=IN14
IF main_sonar=1 THEN principal
SLEEP time
LOOP
```

```
END
' stamp on low consumption mode
' ------------------------------------------------------------------------

Principal:
HIGH ...
main_sonar=IN14
IF main_sonar=0 THEN Beginning

  cm1 = 0
  cm2 = 0
  cm3 = 0
  cm4 = 0
  cm5 = 0
  counter = 0

FOR idx = 1 TO 3


  GOSUB Get_Sonar1
  IF cm > 150 THEN counter = counter + 1
  IF cm > 30 THEN cm = 35
  cm1 = cm1 + cm


  GOSUB Get_Sonar2
  IF cm > 150 THEN counter = counter + 1
  IF cm > 40 THEN cm = 40
  cm2 = cm2 + cm


  GOSUB Get_Sonar3
  IF cm > 150 THEN counter = counter + 1
  IF cm > 40 THEN cm = 40
  cm3 = cm3 + cm


  GOSUB Get_Sonar4
  IF cm > 150 THEN counter = counter + 1
  IF cm > 40 THEN cm = 40
  cm4 = cm4 + cm


  GOSUB Get_Sonar5
  IF cm > 150 THEN counter = counter + 1
  IF cm > 40 THEN cm = 40
  cm5 = cm5 + cm

NEXT

IF counter = 15 THEN U_Turn

cm1 = cm1 / 3
cm2 = cm2 / 3
cm3 = cm3 / 3
cm4 = cm4 / 3
cm5 = cm5 / 3



sum1 = (cm2 + cm4)/2
sum2 = (cm3 + cm5)/2
sum = ABS(sum1-sum2)

DEBUG "SUM.......",DEC sum,CR
DEBUG "cm1...",DEC cm1, CR
DEBUG "cm2...",DEC cm2, CR
DEBUG "cm3...",DEC cm3, CR
DEBUG "cm4...",DEC cm4, CR
DEBUG "cm5...",DEC cm5, CR

IF sum1>=sum2 THEN left
IF sum1<sum2  THEN right
'-------------------------------------------------------------------------------------------
' Sonar Routine -----------------------------------------------------
```

```
Get_Sonar1:
  Pang = IsLow                                         ' make trigger 0-1-0
  PULSOUT Pang, Trigger                                ' activate sensor
  PULSIN  Pang, IsHigh, rawDist                        ' measure echo pulse
    GOSUB distance
  cm = rawDist ** 2257

    PAUSE 20
  RETURN

Get_Sonar2:
  Peng = IsLow                                         ' make trigger 0-1-0
  PULSOUT Peng, Trigger                                ' activate sensor
  PULSIN  Peng, IsHigh, rawDist                        ' measure echo pulse
    GOSUB distance                                     ' remove return trip
  cm = rawDist ** 2257

    PAUSE 20
  RETURN

Get_Sonar3:
  Ping = IsLow                                         ' make trigger 0-1-0
  PULSOUT Ping, Trigger                                ' activate sensor
  PULSIN  Ping, IsHigh, rawDist                        ' measure echo pulse
    GOSUB distance
  cm = rawDist ** 2257


    PAUSE 20
  RETURN

Get_Sonar4:
  Pong = IsLow                                         ' make trigger 0-1-0
  PULSOUT Pong, Trigger                                ' activate sensor
  PULSIN  Pong, IsHigh, rawDist                        ' measure echo pulse
    GOSUB distance
  cm = rawDist ** 2257

    PAUSE 20
  RETURN

Get_Sonar5:
  Pung = IsLow                                         ' make trigger 0-1-0
  PULSOUT Pung, Trigger                                ' activate sensor
  PULSIN  Pung, IsHigh, rawDist                        ' measure echo pulse
    GOSUB distance
  cm = rawDist ** 2257

    PAUSE 20
  RETURN

Distance:
  rawDist = rawDist */ Scale                           ' convert to uS
  rawDist = rawDist / 2
RETURN


left:
IF sum<5  THEN speed=1
IF sum>25 THEN speed=3
IF sum<25 AND sum>15 THEN speed=5
IF sum<15 AND sum>5  THEN speed=7
IF cm1<20 THEN speed=0
GOTO send

right:
IF sum<5  THEN speed=1
IF sum>25 THEN speed=2
IF sum<25 AND sum>15 THEN speed=4
IF sum<15 AND sum>5  THEN speed=6
IF cm1<20 THEN speed=0
GOTO send

send:
```

```
  SEROUT 15, 16624, [speed]
  PAUSE 50

GOTO principal
'-------------------------------------------------------------------------------
'------[U-turn]----------
' -----[ Main Routine ]-------------------------------------------------------
U_Turn1:

GOSUB Compass_Get_Axes                        ' Get x, and y values

  angle = x ATN -y                            ' Convert x and y to brads
  ref = 0

DO
  speed = ?
  GOSUB send_2
                                                ' Main loop
  GOSUB Compass_Get_Axes                      ' Get x, and y values
  angle = x ATN -y                            ' Convert x and y to brads



  DEBUG HOME, "x-axis N(-S) = ",SDEC x,       ' Display axes and degrees
        CLREOL, CR, "y-axis W(-E) = ",
        SDEC y, CLREOL, CR, CR, "angle = ",
        DEC angle, " degrees", CLREOL

  PAUSE 150                                   ' Debug delay for slower PCs

LOOP                                          ' Repeat main loop


U_turn2:


' -----[ Subroutine ]---------------------------------------------------------

Compass_Get_Axes:                             ' Compass module subroutine

  HIGH En: LOW En                             ' Send reset command to HM55B
  SHIFTOUT DinDout,clk,MSBFIRST,[Reset\4]

  HIGH En: LOW En                             ' HM55B start measurement command
  SHIFTOUT DinDout,clk,MSBFIRST,[Measure\4]
  status = 0                                  ' Clear previous status flags

  DO                                          ' Status flag checking loop
    HIGH En: LOW En                           ' Measurement status command
    SHIFTOUT DinDout,clk,MSBFIRST,[Report\4]
    SHIFTIN  DinDout,clk,MSBPOST,[Status\4]   ' Get Status
  LOOP UNTIL status = Ready                   ' Exit loop when status is ready

  SHIFTIN  DinDout,clk,MSBPOST,[x\11,y\11]    ' Get x & y axis values
  HIGH En                                     ' Disable module

  IF (y.BIT10 = 1) THEN y = y | NegMask       ' Store 11-bits as signed word
  IF (x.BIT10 = 1) THEN x = x | NegMask       ' Repeat for other axis

  RETURN
```

C) Weed Detection Board

```
'   {$STAMP BS2p}
'   {$PBASIC 2.5}


RcvData       VAR   BYTE(10)
main_to_cam   VAR   BIT
sonar_to_cam  VAR   BIT
time          CON   2

INPUT 0
INPUT 9

 DO
SLEEP time
main_to_cam=IN0
IF main_to_cam=1 THEN Initialize
 LOOP
 END

Initialize:
' Pause 1 second for CMUcam1 startup
PAUSE 1000

' Send "reset" to sync CMUcam1 and Stamp
SEROUT 12, 240, ["RS", CR]
PAUSE 1000

' Green LED on
SEROUT 12, 240, ["L1 1",CR]
PAUSE 100

' Turn on auto adjust for 5 seconds
SEROUT 12, 240, ["CR 18 44",CR]
PAUSE 100

' Pause 5 seconds for CMUcam1 to auto adjust to lighting conditions
PAUSE 5000

' Turn off auto adjust
SEROUT 12, 240, ["CR 18 44 19 32",CR]
PAUSE 100

' Green LED auto mode
SEROUT 12, 240, ["L1 2",CR]
PAUSE 100

' Give user time to place color target close in front of camera
PAUSE 5000

' Send command - Set poll mode - only sends one return packet -
' of data after each command - reduces data flow
SEROUT 12, 240, ["PM 1",CR]
PAUSE 100

' Send command - Set raw data mode - also suppress Ack:/Nak: to -
' further reduce serial data
SEROUT 12, 240, ["RM 3",CR]
PAUSE 100

' Track Window command looks at the center of CMUcam1 image -
' grabs the color information and sends to the Track Color function

' Send command - Track window
SEROUT 12, 240, ["TW",CR]

' Display the S Statisics packet from TW command
SERIN 14, 240, [STR RcvData\8]

' Ignore the first M packet from TW
PAUSE 2000

GOTO Beginning

' Wait for navigation board to start or the main board to re-initialize
Beginning:
DO
```

```
SLEEP time
sonar_to_cam=IN9
main_to_cam=IN0
IF sonar_to_cam=1 THEN principal
IF main_to_cam=1 THEN Initialize
LOOP
END

Principal:
SLEEP time
sonar_to_cam=IN9
IF sonar_to_cam=0 THEN beginning
PAUSE 500

' Send command - Track color (with no arguments) -
' will track last color grabbed by TW command
SEROUT 12, 240, ["TC",CR]


SERIN 9, 240, [STR RcvData\10]


  IF RcvData(9)>5 THEN zap
  IF RcvData(9)<5 THEN lost


Zap:
DEBUG "Zap",CR, CR
FREQOUT 13, 150, 2500 ' Beep
SHIFTOUT 11,10,MSBFIRST,[255]
PAUSE 1000
FREQOUT 13, 150, 2500 ' Beep
SHIFTOUT 11,10,MSBFIRST,[0]
GOTO principal

lost:
DEBUG "lost",CR, CR
GOTO principal
```

D) Propulsion Board

```
'{$STAMP BS2p}
```

```
'{$PBASIC 2.5}

' constants for angular velocity of the motors
full            CON  1500   'full speed forward
slow            CON  1750   'slow speed forward
null            CON  0      'no speed
back            CON  2200   'backward

pulse_count     VAR  BYTE   'controls number of iterations in a FOR LOOP
speed           VAR  NIB    'decision sent serially by the navigation stamp
main_to_motor   VAR  BIT    'controls if manual mode or automatic mode

INPUT 9                     'pin 9 is set as an input

Principal:
main_to_motor=IN9           'communication navigation to propulsion
IF main_to_motor=1 THEN manual

GOSUB get_speed

IF speed = 0  THEN Backward
IF speed = 1  THEN Forward
IF speed = 2  THEN H_Right
IF speed = 3  THEN H_Left
IF speed = 4  THEN M_Right
IF speed = 5  THEN M_Left
IF speed = 6  THEN L_Right
IF speed = 7  THEN L_Left

'communication sonar to motors ----------------------------------------------
get_speed:
'Through serial communication, get the decision from navigation stamp
SERIN 14, 16624, [speed]
RETURN

'communication main to motors ----------------------------------------------
get_direction:

SERIN 0,  16624, [speed]
RETURN
'---------------------------------------------------------------------------
Forward:

FOR pulse_count = 1 TO 25
  PULSOUT 5, full
  PULSOUT 4, full
  PAUSE 15
  NEXT
GOTO principal

'---------------------------------------------------------------------------
Backward:

FOR pulse_count = 1 TO 25
  PULSOUT 5, back
  PULSOUT 4, back
PAUSE 15
NEXT

GOTO principal

'---------------------------------------------------------------------------
H_Left:

FOR pulse_count = 1 TO 25

  PULSOUT 5, full
  PULSOUT 4, back
PAUSE 15
NEXT

GOTO principal

'---------------------------------------------------------------------------
```

```
H_Right:

FOR pulse_count = 1 TO 25

  PULSOUT 5, back
  PULSOUT 4, full
   PAUSE 15
NEXT

GOTO principal

'-----------------------------------------------------------------------------
M_Left:

FOR pulse_count = 1 TO 25

  PULSOUT 5, full
  PULSOUT 4, null
   PAUSE 15
NEXT

GOTO principal

'-----------------------------------------------------------------------------
M_Right:
FOR pulse_count = 1 TO 25

  PULSOUT 5, null
  PULSOUT 4, full
   PAUSE 15
NEXT

GOTO principal

'-----------------------------------------------------------------------------
L_Left:
FOR pulse_count = 1 TO 25

  PULSOUT 5, full
  PULSOUT 4, slow
   PAUSE 15
NEXT

GOTO principal

'-----------------------------------------------------------------------------
L_Right:
FOR pulse_count = 1 TO 25

  PULSOUT 5, slow
  PULSOUT 4, full
   PAUSE 15
NEXT

GOTO principal
'-----------------------------------------------------------------------------
Manual:
IF main_to_motor=0 THEN principal
GOSUB get_direction

IF speed=1 THEN Forward
IF speed=2 THEN Backward
IF speed=3 THEN H_Left
IF speed=4 THEN H_Right
IF speed=5 THEN L_Left
IF speed=6 THEN L_Right
```

Appendix 2 (Drawings)