# Tech report on SPAWN

Ankur Handa

Robotics Research Center

IIIT-Hyderabad

A tech report on the robot, SPAWN

# Acknowledgements

I am very thankful to my advisor, Dr. Madhava Krishna for providing me with an opportunity to work at an excellent place, *CAIR*. He has been more than an advisor to me and I must thank him for his patience and the encouragement he has given me over time.

I am greatly indebted by the help and assitance I received from people at the *CAIR*. I owe a huge debt of gratitude to Mr. Sartaj Singh who has been very helpful, supportive, encouraging and caring during my short stay at the *CAIR* and even after. I am also immensely grateful to Mr. VC Ravi and Mr. Nikhil Mahale for the time they made out of their daily schedule to test and debug the code with me. Thanks to Mr. Victor, without whom the system would not have been possible. He has been very kind to me and helped me tirelessly to design the whole system, SPAWN, in a very short period of time: all the cutting of acrylic sheets at any time on my request, the proper alignment of motors, wheels, the stereo head he made at the last moment, the drilling of holes and all that help in making the ornithopter as well. Many thanks to Mr. Michael for all his invaluable contribution in designing the ornithopter. Special thanks to Mr. Supreeth Achar for his help in testing the initial run of the robot at *IIIT*. Thanks also to Mr. Mahesh Mohan for his patience and timely reporting the initial bugs in the robot. Thanks are also due to Mr. Karthikeya Viswanath for lending me his camera.

# Contents

**Nomenclature** **v**

**1 Introduction** **1**

**2 The Anatomy** **2**

2.0.1 ATMEL atMega-16 . . . . . . . . . . . . . . . . . . . . . 3
2.0.2 Caster wheels . . . . . . . . . . . . . . . . . . . . . . . . 4
2.0.3 12V, 7Ah Lead Acid battery pair . . . . . . . . . . . . . . 4
2.0.4 USB cable . . . . . . . . . . . . . . . . . . . . . . . . . . 5
2.0.5 MAX-232 . . . . . . . . . . . . . . . . . . . . . . . . . . 6
2.0.6 Johnson 150x40mm wheels . . . . . . . . . . . . . . . . . 6
2.0.7 DC-to-DC converters . . . . . . . . . . . . . . . . . . . . 7
2.0.8 PITTMAN motors . . . . . . . . . . . . . . . . . . . . . . 7
2.0.9 Main Board . . . . . . . . . . . . . . . . . . . . . . . . . 9

**3 Connecting to the robot** **10**

3.1 Synchronization with the main board . . . . . . . . . . . . . . . 10
3.2 Connection with the Host . . . . . . . . . . . . . . . . . . . . . 11
3.3 Other Commands . . . . . . . . . . . . . . . . . . . . . . . . . . 12
3.3.1 reset odometery . . . . . . . . . . . . . . . . . . . . . . . 12
3.3.2 set Velocity . . . . . . . . . . . . . . . . . . . . . . . . . 12
3.3.3 rotation . . . . . . . . . . . . . . . . . . . . . . . . . . . 13
3.3.4 translation . . . . . . . . . . . . . . . . . . . . . . . . . . 13
3.3.5 get Odometery . . . . . . . . . . . . . . . . . . . . . . . 13
3.4 Serial port link establishment . . . . . . . . . . . . . . . . . . . 13

**4 Algorithms for commands 15**

4.1 Algorithm for Rotation . . . . . . . . . . . . . . . . . . . . . . . . 15

4.2 Algorithm for translation . . . . . . . . . . . . . . . . . . . . . . . . 16

**5 Concluding Remarks 17**

5.1 Reminding the user - Precautionary measures . . . . . . . . . . . 17

**References 18**

# List of Figures

2.1 This is the avr-atMega16-usb board used for serial to USB data transfer. . . . . . . . . . 3
2.2 Front caster wheel. . . . . . . . . . 4
2.3 A typical 12V, 7Ah Lead Acid battery. . . . . . . . . . 5
2.4 The figure shows USB cable and plug A at the end. . . . . . . . . . 5
2.5 Pin-out of MAX232. . . . . . . . . . 6
2.6 Johnson 150x40 wheel used on the robot. . . . . . . . . . 6
2.7 DC to DC converters on acrylic sheet. . . . . . . . . . 7
2.8 The figure shows a sample of PITTMAN GM9234 motor which is used on the robot. These motors belong to the GM9000 series of brushed commutated DC gearmotors. . . . . . . . . . 8
2.9 The main board (placed at the bottom). . . . . . . . . . 9

3.1 The windows command prompt when the connection is established. 14

# Chapter 1

# Introduction

This report is a modest attempt to provide the reader with all the essential minutiae and the nitty-gritty details associated with this robot. Every possible effort is made to familiarize the reader with the robot through this single piece of documentation. Going with a quick glint of memory, this robot was developed with the colloborative contribution of *IIIT Robotics Lab, Hyderabad* and *CAIR, Bangalore* during the summer of 2007. The author worked as an intern over that short period of time and was the only student (from *IIIT*) involved in the project. During his sojourn at *CAIR*, the author was given complete freedom to select any of the best available parts in the laboratory to design this system.

# Chapter 2

# The Anatomy

The robot as we call it SPAWN, is a result of immense contributions of people who were in anyway closely or remotely involved in the project. All the separate parts put together, are credited with diligent efforts from the individuals involved in shaping them. The following list enumerates the parts that were residing on the robot when the author last saw it running.

1. ATMEL atMega-16.
2. Caster wheels.
3. 12V, 7Ah Lead Acid battery pair.
4. USB cable.
5. MAX-232.
6. Johnson 150 $\times$ 40mm wheels.
7. DC-to-DC converters.
8. PITTMAN motors.
9. Main board.

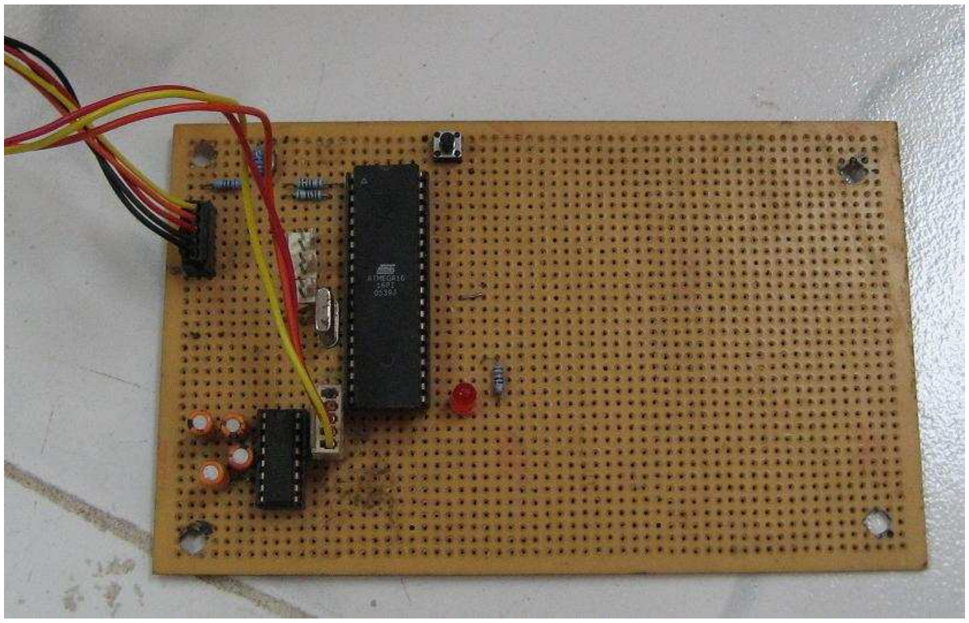


Figure 2.1: This is the avr-atMega16-usb board used for serial to USB data transfer.

### 2.0.1 ATMEL atMega-16

The laptops (in this case it is *AMD Turion 64*) today in general don't have serial ports in their motherboard. The ATMEL atMega-16 is used as simple serial to USB converter to exchange the data from the laptop with the robot. A seperate soldered board with an atMega-16, a MAX-232, a red LED and a switch, finds its place on the second acrylic sheet placed in order. A white colored USB wire which is attached to the board has one connector at the other end which fits into the USB port. The red LED flashes as the connection is established with the main board. If this doesn't happen in one pass, pressing the reset switch will ensure another attempt to establish a connection. The main board (the green colored) has a 3 PIN-Male connector for serial data transfer (Tx, Rx and Gnd). The atMega-16 utilizes UART to connect to the main board through a TTL-RS232 level shifter IC, MAX-232 which sits on a 16 PIN IC base soldered on the atMega-16 board. There is another 6 PIN connector comprised of 3 separate 2 PIN connectors placed in series. This connector connects with STK-200 programming dongle to download the hex code from PC into atMega-16. The atMega-16 is running at

12Mhz with external oscillator. At this frequency each bit on the USB takes 8 clock cycles and thus the rate of transfer is 1.5M bits/sec. The circuit is designed and soldered as explained in (**?** ). We are using the connection at a low speed data rate.

### 2.0.2 Caster wheels

Two caster wheels, one at the front and other at the rear end, are fixed on the acrylic sheet at the bottom. These caster wheels move freely and the positioning is such that they maintain the center of gravity of the robot at the center of the sheet, unlike the case when only one caster wheel is used. Typically such type of wheels are found on shopping carts and rolling chairs.

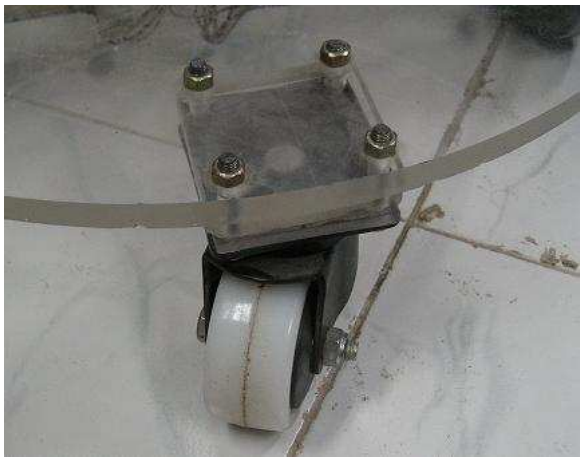


Figure 2.2: Front caster wheel.

### 2.0.3 12V, 7Ah Lead Acid battery pair

Two 12V, 7Ah Lead acid batteries provide the main power supply needed by the robot. These batteries rest on the acrylic sheet at the bottom, above each wheel. The batteries when fully charged can supply 24.5-25.5 Volts. After running the robot for reasonably longer period of time, if the voltage drops to 22V or less, the batteries need charging.

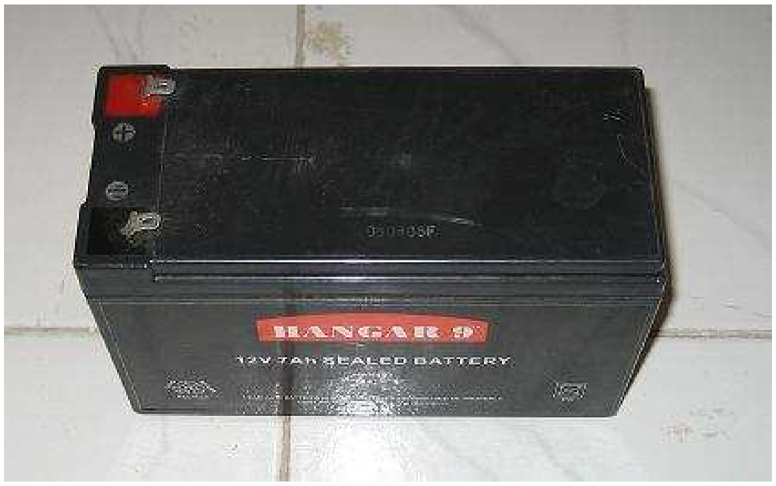


Figure 2.3: A typical 12V, 7Ah Lead Acid battery.

### 2.0.4 USB cable

A white colored cable is attached to the atMega-16 board which has USB Series A plug [2] at the other end. This plug is the most common plug and fits into any of the typical USB ports available at the host. When the robot is switched on, this plug is connected into the USB port of the host. The other end of the cable is a 5 PIN female connector fixed into its male counterpart soldered on the atMega-16 board.

Figure 2.4: The figure shows USB cable and plug A at the end.

### 2.0.5 MAX-232

The basic purpose of this IC is to shift the voltage levels from TTL to RS232 and vice-versa. Generally this IC is used to connect to the serial port of the PC with any other device having a UART. Since the voltage levels may be different at the device and PC, the IC thus shifts the voltage levels to either logic.

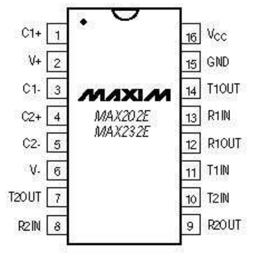


Figure 2.5: Pin-out of MAX232.

### 2.0.6 Johnson 150x40mm wheels

Two 150mm diameter and 40mm thick Johnson wheels are used with which the robot runs over. These wheels are quite tough in strength and provide non-shaky and stable movement. Fig. 2.6 shows a snapshot of wheel.

Figure 2.6: Johnson 150x40 wheel used on the robot.

### 2.0.7 DC-to-DC converters

Three different DC-to-DC converters derive different isolated voltages from the input voltage supplied by the batteries. A 24V to 24V converter supplies power to the SICK laser, one 24V to 12V converter supplies power to the motors. There are two 24V to 5V converters. One of them supplies power to the motor driver circuit and other provides power to the PIC microcontroller. Fig. 2.7 shows the DC-to-DC converters on board.

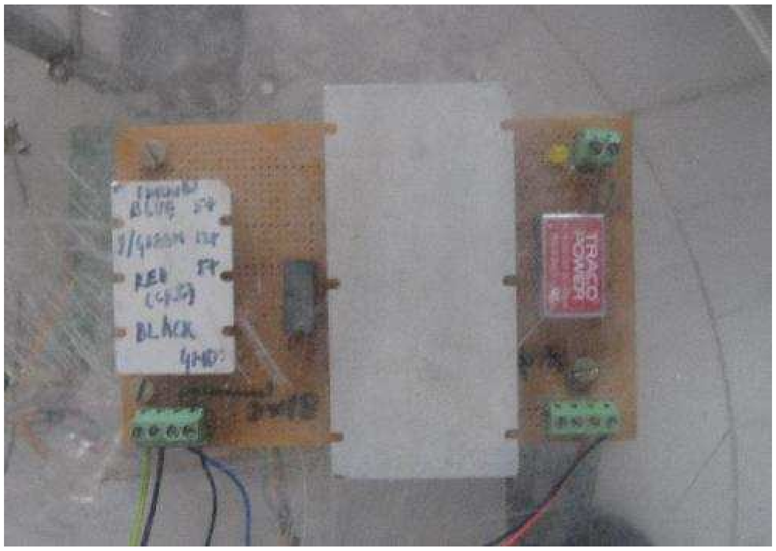


Figure 2.7: DC to DC converters on acrylic sheet.

### 2.0.8 PITTMAN motors

Two PITTMAN motors (as shown in Fig. 2.8) with built-in encoders are used to run the wheels. The various specs mechanical and electrical are listed as:

1. DC Brush Gearmotor.
2. 1.37" Diameter.
3. 218.4:1 ratio.
4. 500 oz-in Maximum Continuous Torque.

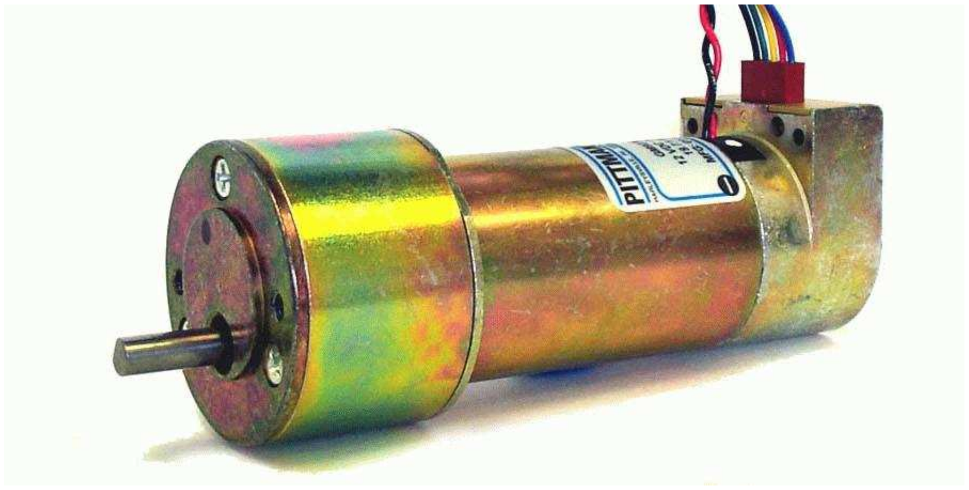


Figure 2.8: The figure shows a sample of PITTMAN GM9234 motor which is used on the robot. These motors belong to the GM9000 series of brushed commutated DC gearmotors.

5. 4199 oz-in Peak Torque (Note: Peak torque is provided for the purpose of performance calculations only. Operation near, or at, a stalled condition will result in motor and/or gearhead damage).
6. 21 rpm No load speed.
7. Torque Constant (Kt) = 3.29 oz-in / amp.
8. Voltage Constant (Ke) = 2.43 v/krpm.
9. Resistance (R) = 1.26 ohms.
10. Inductance = 1.02 mH.
11. Rated voltage: 12 volts.
12. Encoder: 500 CPR.
13. Length (motor) = 3.67".
14. The gearhead will be damaged when operating at the Peak Torque.
15. Unit supplied with ball bearing output shaft.

### 2.0.9 Main Board

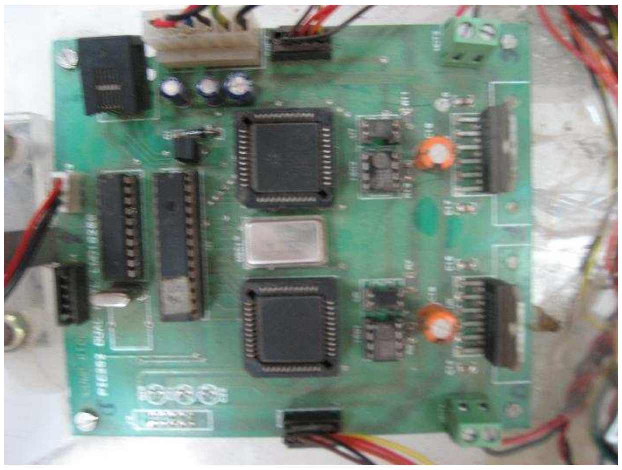


Figure 2.9: The main board (placed at the bottom).

The main board which is lying on the acrylic sheet at the bottom is comprised up of the following ICs:

1. PIC18F452 microcontroller.
2. MAX232 level shifter (TTL to RS232 and vice-versa).
3. LMD18200T motor driver.
4. HCTL-1100 motion controller.

PIC18F452 controls HCTL-1100, LMD18200T and MAX232. Apart from these components, there is one slot for programming PIC18F452, two 5 pin connectors and a 6 pin connector. There are two jumpers just aligned to the placement of micrcontroller on the board. These jumpers have to be taken out when programming is to be done and placed into the same positions after programming. The color coding for the 6 pin connectors is as described below:

| Color coding | |
|---|---|
| Yellowish Green (thick) | +12 Volts |
| Black(thick) | Gnd of 12 Volts |
| Blue | +5V (drive) |
| Black | GND(of 5Volts drive) |
| Red | +5V (ckt) |
| Black | GND(of 5Volts ckt) |

# Chapter 3

# Connecting to the robot

## 3.1 Synchronization with the main board

As mentioned earlier, there is a white colored USB wire coming out of the atMega-16 board which has USB plug A at the end that can be plugged into the USB port of the laptop. There are few steps which need to be followed to establish link with the main board. The main power switch should be turned on before everything. A glowing yellow LED on the main power board should indicate that the whole system is getting power from the batteries. Although it is advised to check the voltage of the batteries before supplying power to the system. Now the USB plug should be plugged into the USB port of the laptop. If the connection establishes, the red LED on the atMega-16 will immediately flash for a moment indicating the synchronization of the atMega-16 board with main board. If this red LED doesn't flash for the first time, the reset switch should be pressed. This will ensure another attempt at establishing a connection with the main board. If this also fails, the main power switch should be turned off and again turned on, repeating the whole process thereafter. It may sometime happen that even after doing all this, the connection is never established. If such a condition occurs, changing the USB port will be another solution to this. Once everything goes fine, the robot is ready to be commanded. The various steps involved thereafter are explained in the following sections.

## 3.2 Connection with the Host

The momentarily flashing red LED just indicates that the atMega-16 is synchronized with the main board. To ensure that the atMega-16 as a USB device is detected, type on the terminal **/sbin/lsusb**. It will show the current state of all the USB ports available on the laptop and the ID of the USB device connected to it as well.

```
Bus 001 Device 001: ID 0000:0000
Bus 002 Device 002: ID 16c0:05dc
```

The non-zero ID here is the ID of the atMega-16 (if it gets detected) and the number besides BUS depends upon the USB port the cable is plugged into. After performing the steps mentioned in Section 3.1 the robot is ready to be commanded from the host over the USB link. There is a particular function named as **connectUsb** in the API of the robot. This function when called upon makes an attempt to connect to the atMega-16 over the USB link. A message **Could not find USB device 'PowerSwitch' with vid=0x16c0 pid=0x5dc** popping up on the terminal indicates that the attempt is unsuccessful. The only way to counter this is to switch the main power off and turn it on and follow the steps mentioned earier. A general advice would be to place this function on top of all the other code fragments. If everything goes fine, the robot is all set

**Algorithm 1** Connection

```
robot spawn;
spawn.connectUsb();
rest of the code fragments...
....
```

to move in any direction at any speed upon user's instructions. There are some other commands for which a user has complete access over. These commands are explained one by one in the next section.

## 3.3 Other Commands

There are more commands which can be given to the robot. These are typical commands which are required for any algorithm or to perform any task with the robot. This may include setting any optional translational and/or rotational velocity, rotating and translating the robot by any angle and distance respectively. These are explained in the following subsections as follows:

### 3.3.1 reset odometery

Before accessing the odometery for the first time, it is best to reset it so that the $x, y$ and $\theta$ of the robot are set to zero. This function should be called just after the **connectUsb()** function.

**Algorithm 2** Reset Odometery

```
robot spawn;
spawn.connectUsb();
spawn.resetOdometery();
....
```

### 3.3.2 set Velocity

In this model of control, there are two velocities viz. translational velocity $v_t$ and rotational velocity $v_r$. The translational velocity, as the name suggests, is nothing but the velocity with which the robot moves forward or backward depending upon sign (+/-) of the velocity and the rotational velocity is the velocity with which the robot makes a cirular turn. These velocities are measured in mm/sec. The function **setVelocity()** takes two arguments (as it is no doubt known that the agruements are $v_t$ and $v_r$) and sets both of the velocities according to arguements. It is generally safe to keep the velocities below 1200mm/sec.

**Algorithm 3** set Velocity

```
spawn.setVelocity($v_t$,$v_r$);
....
```

### 3.3.3 rotation

This is one of the basic commands needed by the user to operate the robot. It has only one agruement which is nothing but the angle in degrees which the user wants to rotate the robot to. The function name is int **rotate(angle)**.

### 3.3.4 translation

This is another basic command which the robot can be controlled with. It takes once arguement, the distance in mm which the robot is to move on user's request. This attribute can be accessed by the function int **translate(distance)**.

### 3.3.5 get Odometery

This function returns the current $x, y$ and $\theta$ of the robot. It takes three arguements (call by reference) and returns the relevant values into them. The function is int **getOdometery(**$x$,$y$,$\theta$**)**. x and y are measured in mm and $\theta$ is measured in degrees.

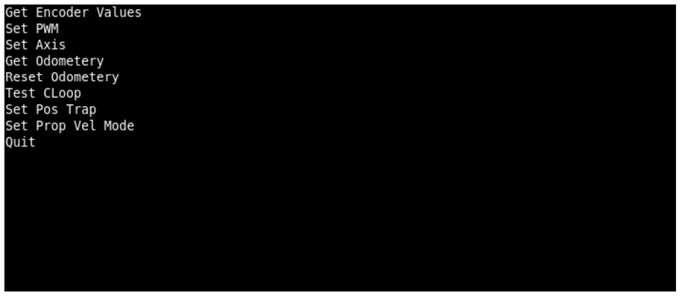
All these functions return 1 or 0 depending upon whether the operation was performed successfully.

## 3.4 Serial port link establishment

This section deals with commanding the robot through only serial port. Since the main board has a 3 PIN connector (Tx, Rx and Gnd) we only need a DB9 PIN connector to connect the robot to the host through its serial port. The 3 PIN female connector with red, brown and black coloring which fits into its male counterpart on the main board, should be connected to the DB9 PIN connector as follows:

| | |
|---|---|
| Red should go to Rx | $3^{rd}$ Pin of DB9 Connector |
| Brown should go to Tx | $2^{nd}$ Pin of DB9 Connector |
| Black should go to Gnd | $5^{th}$ Pin of DB9 Connector |

Just after turning on the main power switch, the serial program running on host side should display the contents as shown in the image below, on the screen. The list of commands shown are the only way to access the robot and run it in any direction and at any speed on user's request.



```
Get Encoder Values
Set PWM
Set Axis
Get Odometery
Reset Odometery
Test CLoop
Set Pos Trap
Set Prop Vel Mode
Quit
```

Figure 3.1: The windows command prompt when the connection is established.

# Chapter 4

# Algorithms for commands

## 4.1 Algorithm for Rotation

The velocities of the robot during rotation are controlled through a PID contol mechanism. The rotation velocity is kept proportional to the difference in the final angle and current angle. The translation velocity is negative proportional to the linear translation being brought upon. The whole cycle; reading the odometery, calculating the velocities and setting them is repeated until the final bearing of robot is close to actual bearing the robot is suppose to move to. The following pseudo code illustrates the algorithm. The letters with subscript $c$ denote the their current value and letters with subscript $f$ denote the final value.

**Algorithm 4** Rotation

```
  while $\theta_c \sim \theta_f$ do
    getOdometery($x_c, y_c, \theta_c$)
    $v_r \propto \theta_f - \theta_c$
    setVelocity($v_t$,$v_r$)
  end while
  setVelocity(0,0)
```

## 4.2 Algorithm for translation

The translation of the robot is also controlled by a PID control mechanism. The translation velocity of robot is kept proportional to the current distance it is from the destination and rotational velocity of robot is negative proportional to the angle it has drifted from the line joining the starting position and the destination position. Similar to the case of rotation, the whole cycle is repeated till a threshold stage.

**Algorithm 5** Translation

```
while $d_{left} \leq d_{thresh}$ do
    getOdometery($x_c, y_c, \theta_c$)
    $d_{left} = \sqrt{(x_c - x_f)^2 + (y_c - y_f)^2}$
    $v_t = \min(k_v d_{left}, v_{tconst})$
    $v_r = \min(-k_\theta \theta_{drift}, v_{rconst})$
    setVelocity($v_t$,$v_r$)
end while
setVelocity(0,0)
```

# Chapter 5

# Concluding Remarks

## 5.1 Reminding the user - Precautionary measures

1. Always check the voltage of the batteries before turning on the main power switch.
2. If the voltage falls below 21V, batteries should be kept on charge.
3. In case SICK laser is to be used make sure the length of USB-SERIAL connector is small so as to avoid it rest on the DC-to-DC converters.
4. DC-to-DC converters get hot very quickly, make sure there are no stray wires or other sort of connectors hanging over them.
5. If batteries are being charged, make sure the current limit is set to less than 700mA. That way the batteries can be kept on charge overnight.
6. Avoid overcharging the batteries. Keep them on charge for around 10 hours or less. Otherwise you may observe sulphuric acid fumes coming out of them.
7. Handle all the connectors delicately.
8. Always keep a battery pair in buffer so as to save the time.

# References

[1] AVR USB. http://www.obdev.at/products/avrusb/index.html.

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