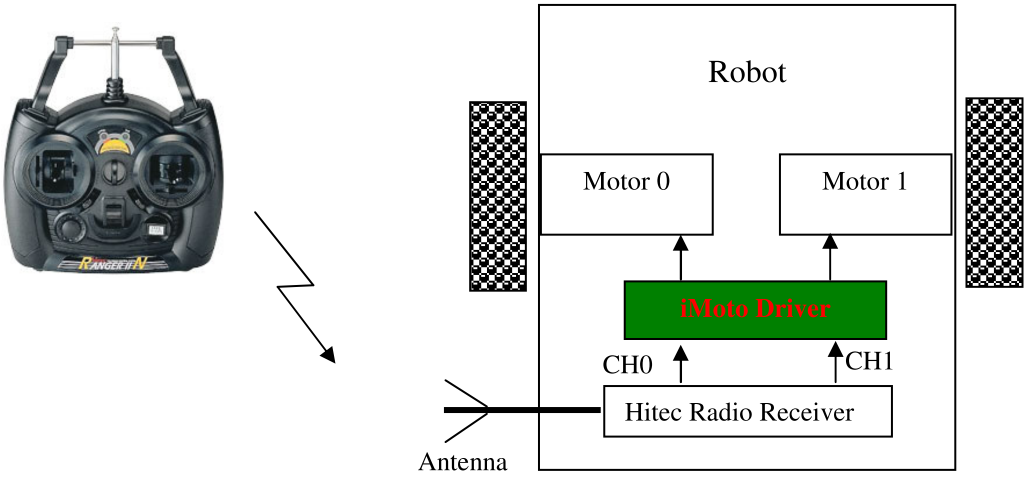
Use iMoto with Radio Controller

Overview

A typical Radio Controller (e.g. Hitec Ranger II) is able to generate to channel of control signals. The control signal is Pulse Width Modulated (PWM). **iMoto High Power Motors Driver board** has built in processor to interpret such signal directly. The diagram below shows a typical application of using iMoto in remote control robots like RC Sumo Robots.



Operating Mode and Parameters

iMoto High Power Motors Driver board can control 2 motors by using RC (Radio Control) PWM signals. Depending on the motors used, it can be set to either **Velocity mode or Voltage mode**. For the motors with encoder, the Velocity mode is used and those without the encoders have to use Voltage mode.

In **Velocity Mode**, the motors are controlled by internal PID controller with encoder feedback and the speed is directly proportional the incoming pulse width. In **Voltage Mode**, the voltage across the motor (PWM duty cycle to be more precise) is directly proportional to the incoming pulse width. Note that **Velocity Mode** is highly recommended for remote control robot like RC Sumo Robot because the robot is much more controllable and responsive than that in **Voltage Mode**.

Then connect the RC signal inputs to J12 and J13. For the remote control application, the output signals of the receiver unit are connected to J12 and J13 accordingly. The detailed pin configuration can be found in Hardware Guide.

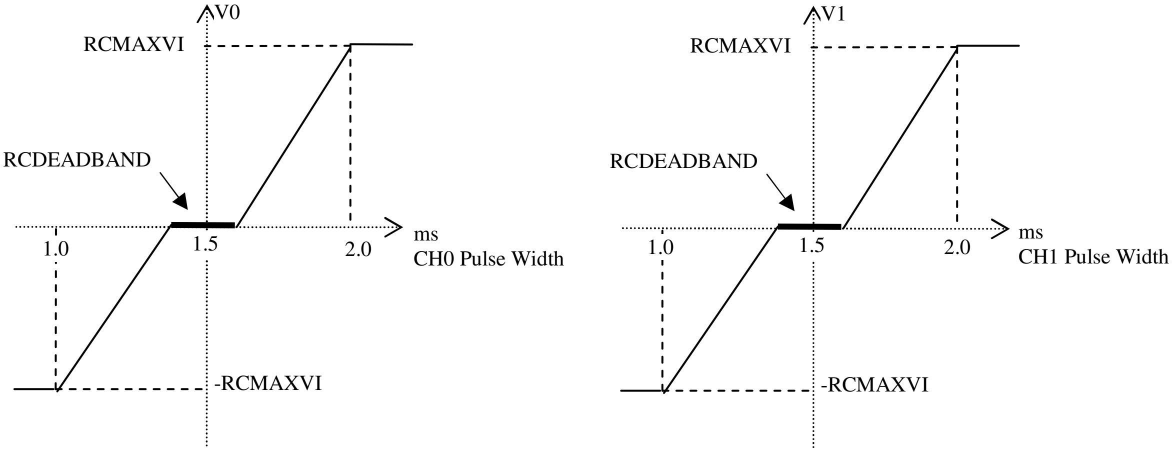
In both Velocity and PWM modes, the motors can be controlled in either coordinated or independent manner. These modes can be set by using the jumpers J7, J8 and J9.

Mode	J7	J8	J9
RC-Velocity-I (Independent)	ON	OFF	OFF
RC-Velocity-C (Coordinated)	OFF	ON	OFF
RC-Voltage-I (Independent)	ON	OFF	ON
RC-Voltage-C (Coordinated)	OFF	ON	ON
UART Mode	OFF	OFF	OFF

Before using the IMOTO in Velocity Mode, the wheel information of the robot must be configured so that the velocity will be in mm/sec. For details, please refer to the next section.

1. RC-Velocity-I (Independent) Mode

In this mode, the speed of motor0 and motor1 are controlled independently by Channel 0 and Channel 1 of the RC receiver respectively. The following diagram shows the relationship between the RC pulse width and the velocity of the two motors, V0 and V1.

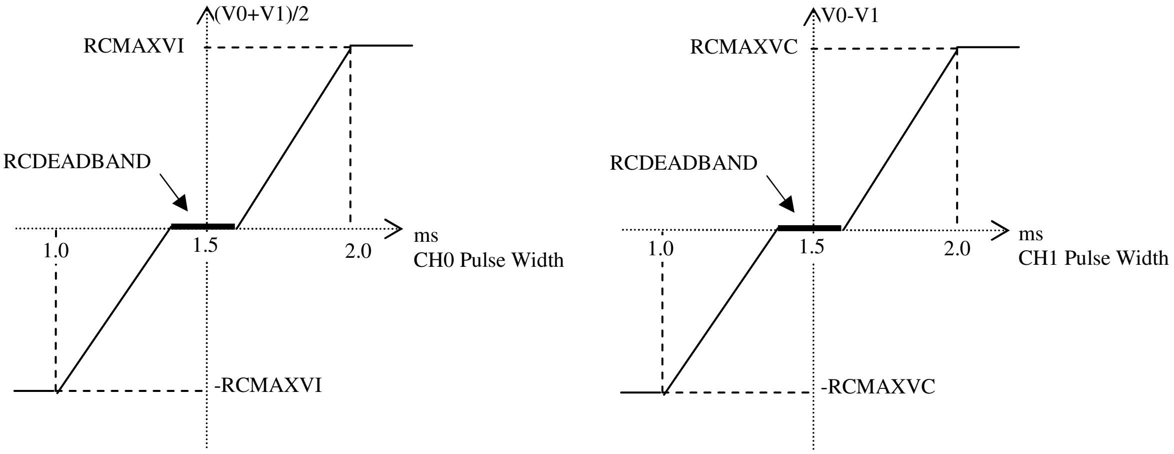


RCMAXVI: The maximum speed of the motor. When the pulse width is 2ms and above, the motor speed is RCMAXVI. When the pulse width is 1ms and below, the motor speed is –RCVMAXI. When pulse width is 1.5ms, the motor speed is zero.

RCDEADBAND: The bandwidth around 1.5ms that is considered neutral, i.e. motor speed is zero.

2. RC-Velocity-C (Coordinated) Mode

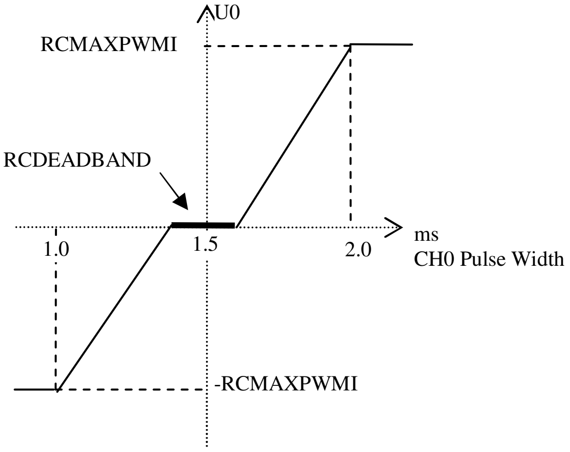
In this mode, the speed of motor0 and motor1 are controlled jointly by Channel 0 and Channel 1 of the RC receiver. The common speed (V0+V1)/2 is controlled by RC Channel 0. The differential speed (V0-V1) is controlled by RC Channel 1. This mode is useful if the operator want to control the linear movement of the robot using one joystick and rotational movement using the other one. The following diagram shows the relationship.

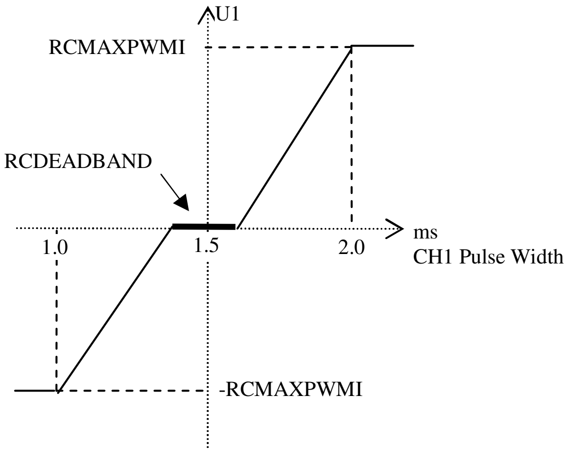


RCMAXVI: The maximum common speed of the two motors.
RCMAXVC: The maximum differential speed of the two motors.
RCDEADBAND: same as 1.

3. RC-Voltage-I (Independent) Mode

In this mode, the voltage (duty cycle) of motor0 and motor1 are controlled jointly by Channel 0 and Channel 1 of the RC receiver. The common voltage (U0+U1)/2 is controlled by RC Channel 0. The differential voltage (U0-U1) is controlled by RC Channel 1.



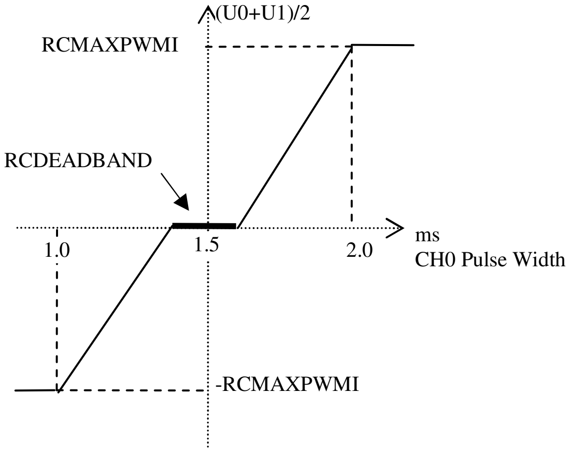


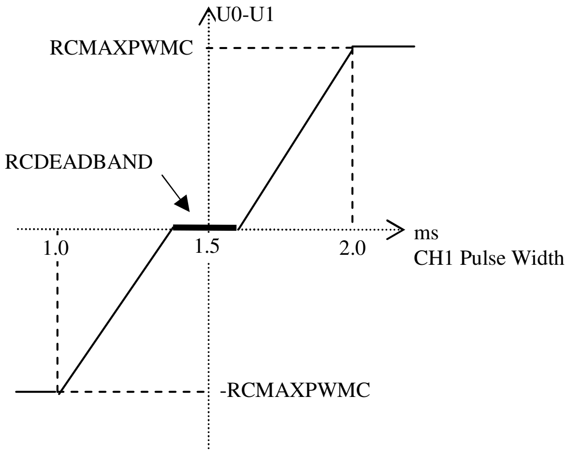
RCMAXPWMI: The maximum PWM duty cycle of the motor driver. When the pulse width is 2ms and above, the duty cycle is RCMAXPWMI. When the pulse width is 1ms and below, the duty cycle is – RCMAXPWMI. When pulse width is 1.5ms, the duty cycle is zero.

RCDEADBAND: The bandwidth around 1.5ms that is considered neutral, i.e. duty cycle is zero.

4. RC-Voltage-C (Coordinated) Mode

For motors without encoder feedback, PID speed controller can not be used. The direct voltage driving can be used. In this mode, the voltage of motor0 and motor1 are controlled independently by Channel 0 and Channel 1 of the RC receiver respectively.
Note that the voltage is represented by PWM duty cycle of the motor driver. The following diagram shows the relationship between the RC pulse width and the velocity of the two motors, U0 and U1.



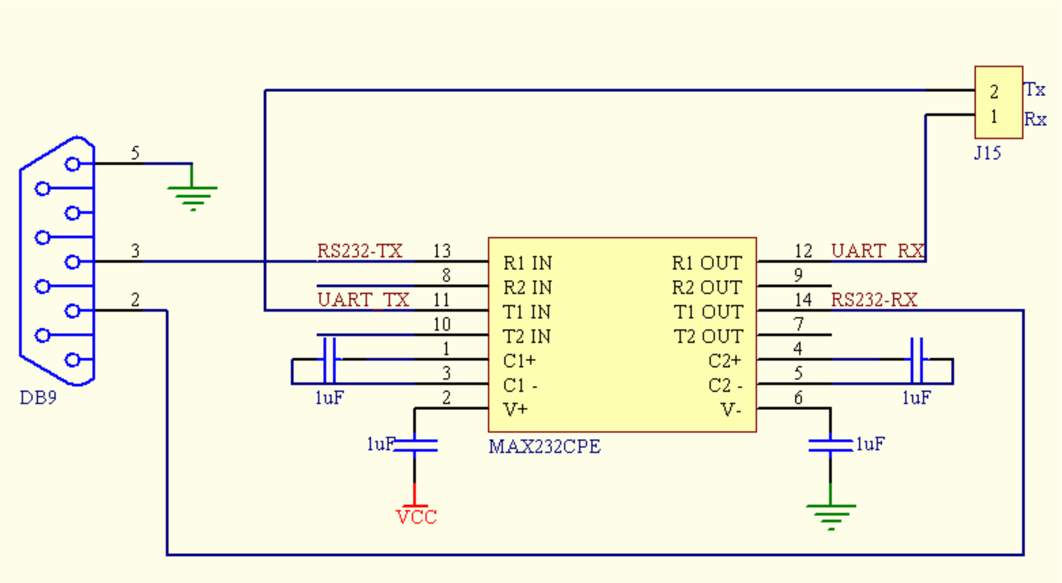


Copyright 2007 iNOVA Microsystems Pte Ltd

RCMAXPWMI: The maximum common voltage of the two motors.
RCMAXPWMC: The maximum differential voltage of the two motors.
RCDEADBAND: same as 3.

Modify the parameters using AT command

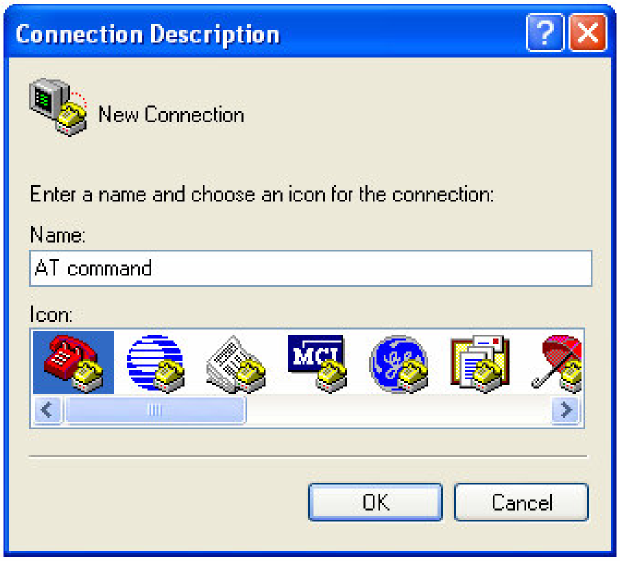
By using UART mode and AT command, the parameters like PID values, wheel information, acceleration and maximum velocity can be modified and saved. In order to use the AT command, the IMOTO needs to be connected to the computer through the MAX232 transceiver as the diagram shown below. So you can make a simple circuit as below and **make sure that the boards have common ground**.



1. Set up the Communication between the board and the PC

After the iMoto is connected to the PC as above, HyperTerminal software can be used to change the parameters. The followings are the step by step instructions for how to set up the HyperTerminal to communicate with the iMoto. Note that the baud rate used is 115200 bps.

1. Select the Windows HyperTerminal (Start->All Programs->Accessories->Communications) from your PC and name the connection as below.



Connection Description

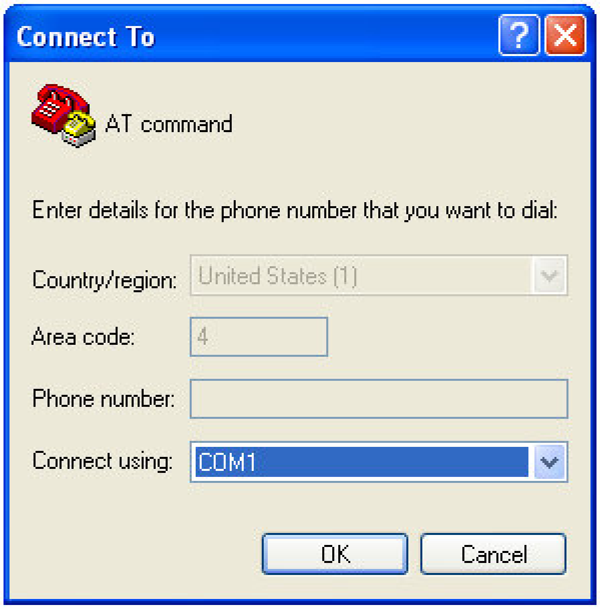
New Connection

Enter a name and choose an icon for the connection:

Name: AT command

Icon:

2. Press OK and choose the COM port that you wished to use.



Connect To

AT command

Enter details for the phone number that you want to dial:

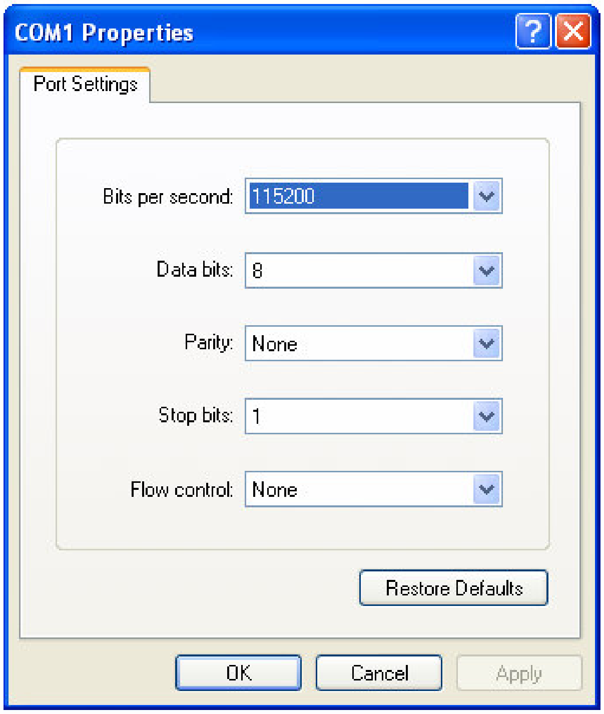
Country/region: United States (1)

Area code: 4

Phone number:

Connect using: COM1

3. Then set all the required fields as below and press ‘Apply’.



COM1 Properties

Bits per second: 115200

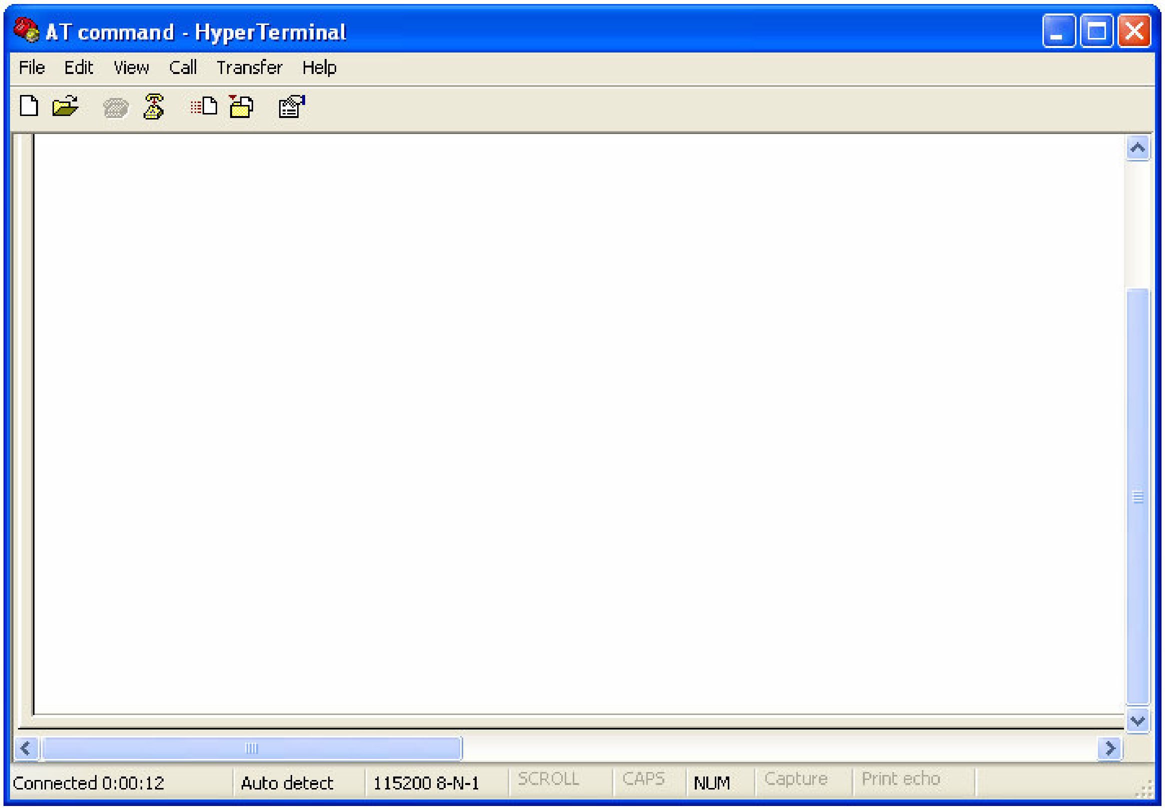
Data bits: 8

Parity: None

Stop bits: 1

Flow control: None

4. Press ‘OK’ and you should be able to see as below and HyperTerminal is ready to use.



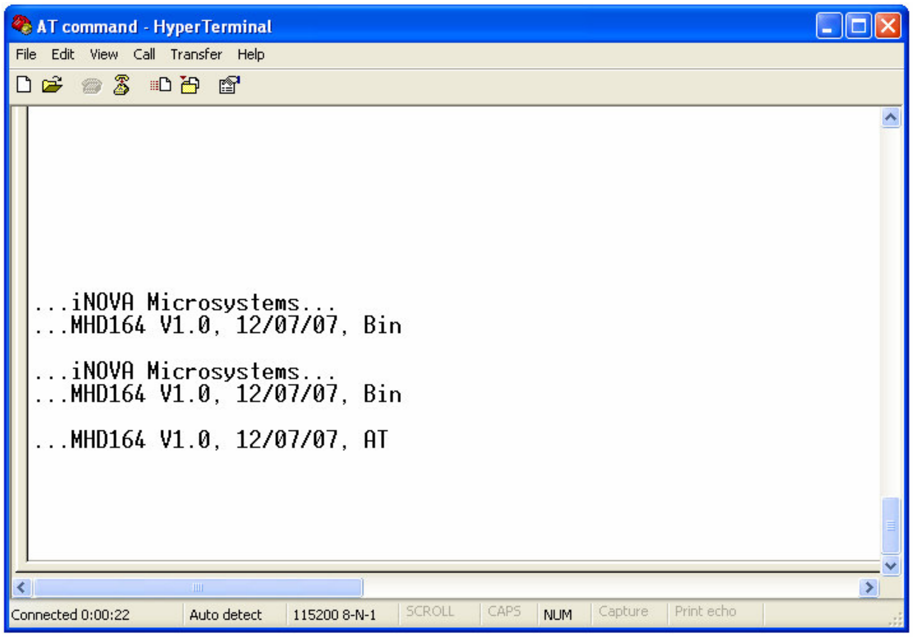
AT command - HyperTerminal

Connected 0:00:12 | Auto detect | 115200 8-N-1 | SCROLL | CAPS | NUM | Capture | Print echo

Copyright 2007 iNOVA Microsystems Pte Ltd

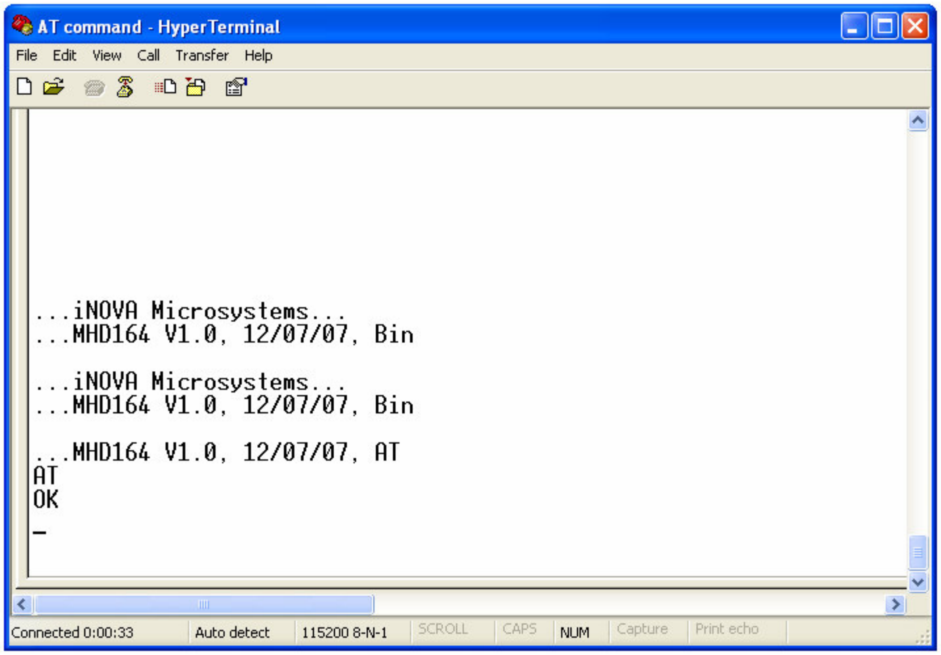
5. After that, set the iMoto to UART mode by changing the jumpers as follow:
J7 = OFF, J8 = OFF, J9 = OFF (i.e. all jumpers taken out)

6. Power off and on the iMoto and you should be able to see as below and the communication is set up successfully. Then the parameters can be changed using the AT commands. For the detailed explanation of AT commands, please refer to the iMoto Software Guide.



```
...iNOVA Microsystems...
...MHD164 V1.0, 12/07/07, Bin

...iNOVA Microsystems...
...MHD164 V1.0, 12/07/07, Bin

...MHD164 V1.0, 12/07/07, AT
```

Just to make sure the AT command interpreter is working properly, type AT then ‘ENTER’ and you should be able to see OK as follows:



```
...iNOVA Microsystems...
...MHD164 V1.0, 12/07/07, Bin

...iNOVA Microsystems...
...MHD164 V1.0, 12/07/07, Bin

...MHD164 V1.0, 12/07/07, AT
AT
OK
_
```

After the communication link is successfully established, it is ready to change the parameters using the AT commands.

2. How to change the parameters using AT commands

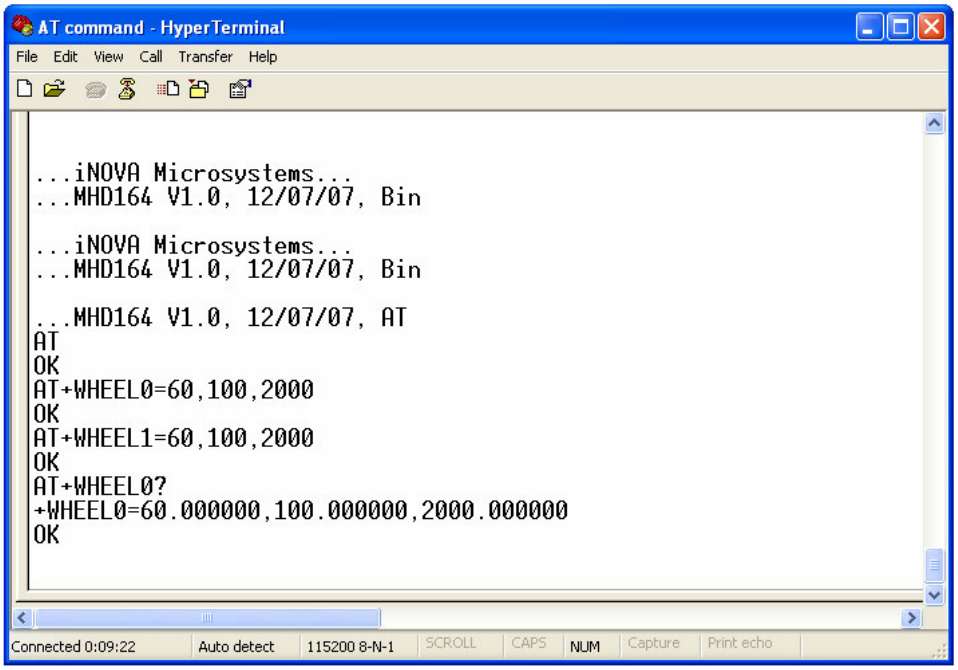
Assuming the robot has the following wheel configuration

Gear ratio = 60:1
Diameter of the wheel = 100mm
CPR (count per revolution) of encoder = 2000
(note that the CPR is 4 times of the encoder's resolution)

To change the wheel information as above, type:
AT+WHEEL0=60,100,2000 <ENTER>
AT+WHEEL1=60,100,2000 <ENTER>

If you key in correctly, you should be able see OK or else it will show error. Just to make sure that the values you just keyed in successfully set, you can read in as follow and you should be able to see the HyperTerminal window as shown below:

AT+WHEEL0? <ENTER>



```
...iNOVA Microsystems...
...MHD164 V1.0, 12/07/07, Bin

...iNOVA Microsystems...
...MHD164 V1.0, 12/07/07, Bin

...MHD164 V1.0, 12/07/07, AT
AT
OK
AT+WHEEL0=60,100,2000
OK
AT+WHEEL1=60,100,2000
OK
AT+WHEEL0?
+WHEEL0=60.000000,100.000000,2000.000000
OK
```

Change maximum acceleration

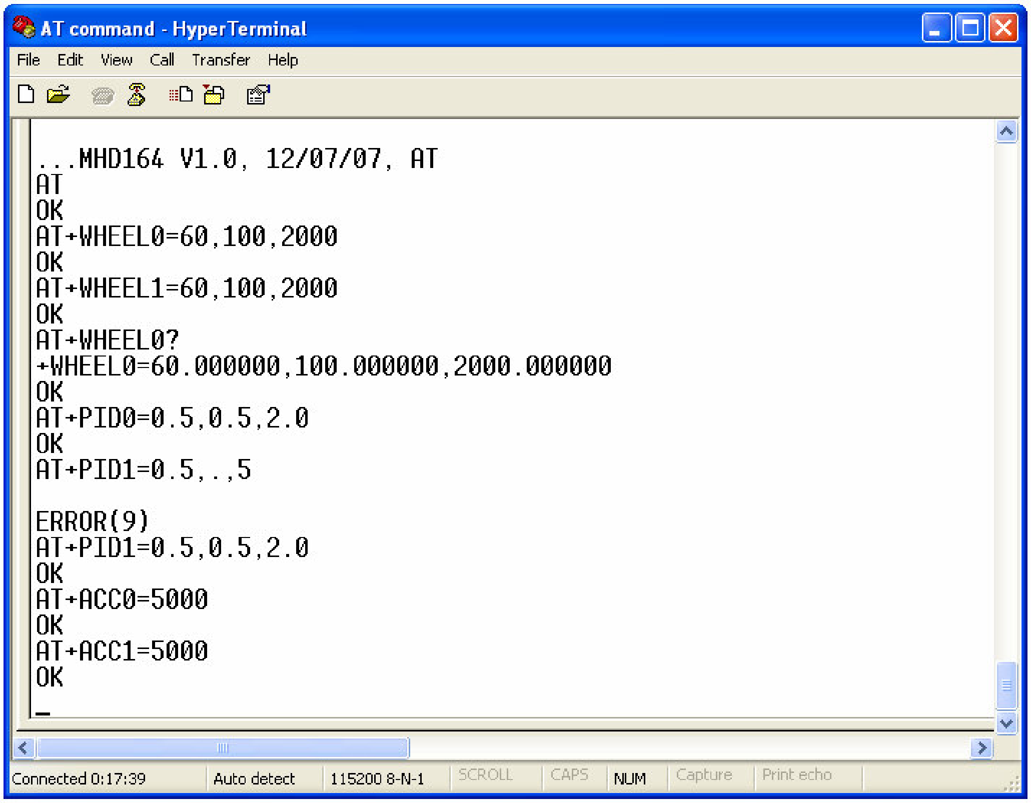
The maximum acceleration determines how fast the motor changes it speed. The default value is 5000mm/sec/sec. If you want to change it to other values like 5000, use the following command

AT+ACC0=5000 <ENTER>
AT+ACC1=5000 <ENTER>

Change PID parameters

For more advanced users, you are able to tune the PID parameters to optimize the motor performance. Use the following command. The three number is Kp, Ki, Kd respectively.
AT+PID0=0.5,0.5,2.0 <ENTER>
AT+PID1=0.5,0.5,2.0 <ENTER>

After all the above procedures, the HyperTerminal window will be as below:



```
...MHD164 V1.0, 12/07/07, AT
AT
OK
AT+WHEEL0=60,100,2000
OK
AT+WHEEL1=60,100,2000
OK
AT+WHEEL0?
+WHEEL0=60.000000,100.000000,2000.000000
OK
AT+PID0=0.5,0.5,2.0
OK
AT+PID1=0.5,.,5

ERROR(9)
AT+PID1=0.5,0.5,2.0
OK
AT+ACC0=5000
OK
AT+ACC1=5000
OK
_
```

For RC Velocity Independent Mode

To set the maximum velocity of both motors at 2000mm/s,
AT+RCMAXVI=2000,2000 <ENTER>

For RC Velocity Coordinated Mode

To set the maximum common velocity of both motors at 2000mm/s,
AT+RCMAXVI=2000,2000 <ENTER>
To set the maximum differential velocity of both motors at 1000mm/s,
AT+RCMAXVI=1000,1000 <ENTER>

For RC Voltage Independent Mode

To set the maximum PWM of both motors at 0.8 (or 80% duty cycle)
AT+RCMAXPWMI=0.8,0.8 <ENTER>

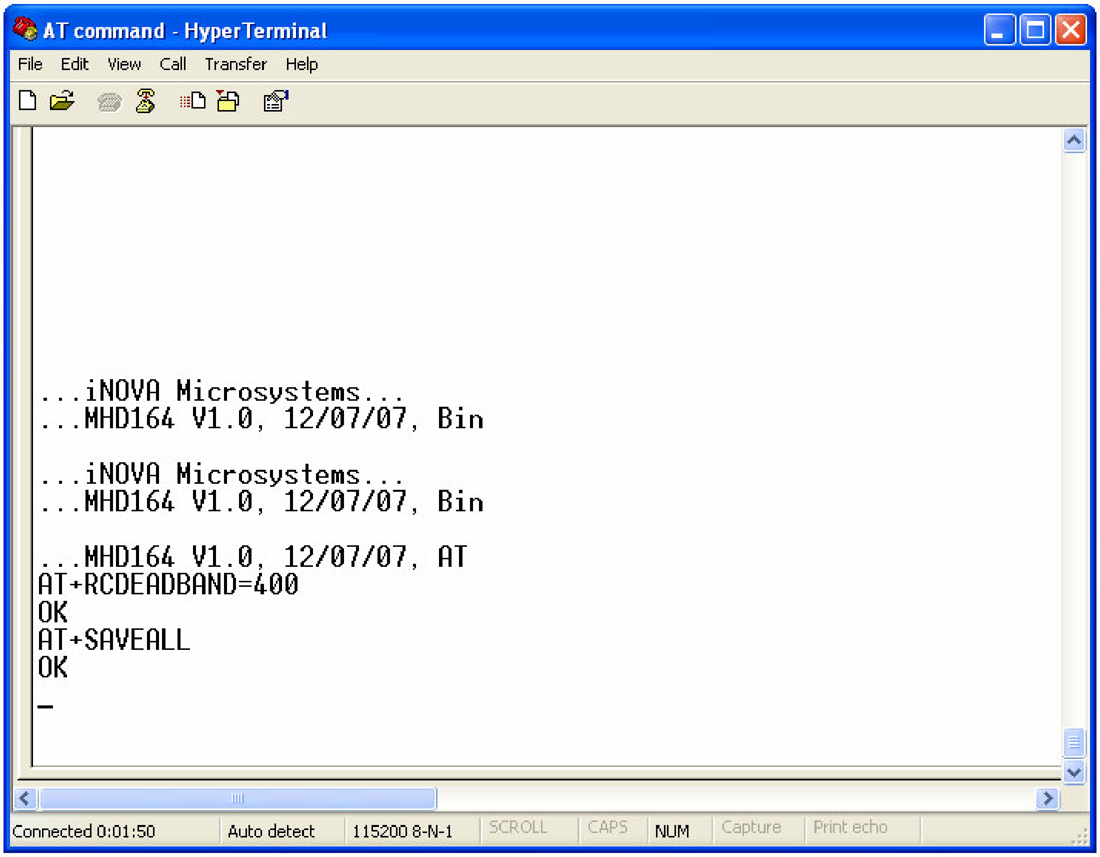
For RC Voltage Coordinated Mode

To set the maximum common voltage of both motors at 0.8 (or 80% duty cycle)
AT+RCMAXPWMI=0.8,0.8 <ENTER>
To set the maximum differential voltage of both motors at 0.4 (or 40% duty cycle)
AT+RCMAXPWMI=0.4,0.4 <ENTER>

To Change the Neutral Bandwidth RCDEADBAND

Neutral bandwidth is the range of the pulse width where the motors are set to be at stationary. For example, if the neutral bandwidth is set at 400µs, the motors will be at stationary for the input RC pulse width range of 1.48ms to 1.52ms since 1.5ms is the zero position for RC signal.

This neutral bandwidth can be changed using AT command as follow:
AT+RCDEADBAND=400<ENTER>
Note: the bandwidth value is in µs.



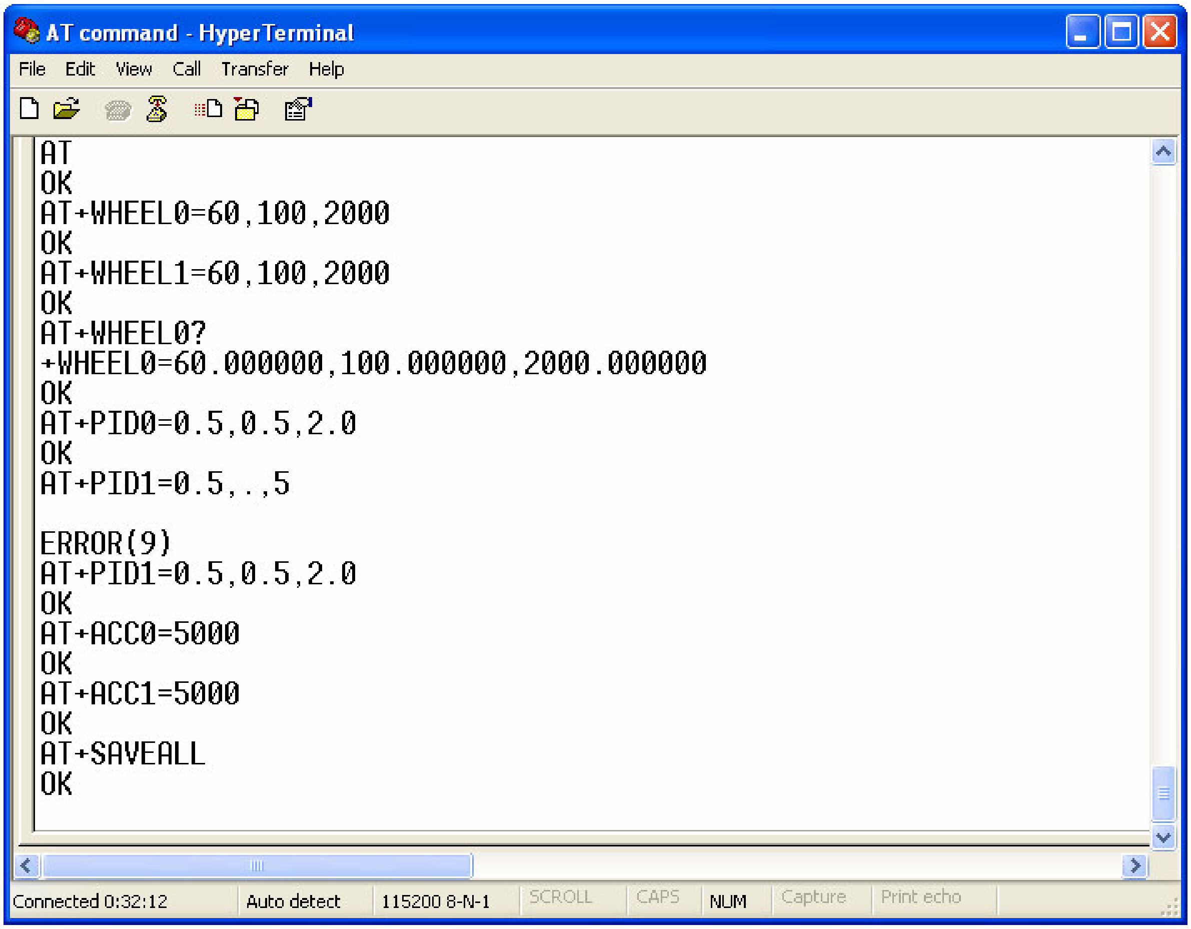
```
...iNOVA Microsystems...
...MHD164 V1.0, 12/07/07, Bin

...iNOVA Microsystems...
...MHD164 V1.0, 12/07/07, Bin

...MHD164 V1.0, 12/07/07, AT
AT+RCDEADBAND=400
OK
AT+SAVEALL
OK
_
```

If you make some typo mistake, the iMoto will prompt an error and all you need to do is just to key in again.

After initializing the parameters, it is important to save all those data in the non-volatile memory so that these data will be there even after the power is reset. The following command is to save all the data in the non-volatile memory.

AT+SAVEALL <ENTER>

IMPORTANT: MAKE SURE THE BOARD IS NOT RESET BEFORE THE DATA ARE SUCCESSFULLY SAVED.



```
AT
OK
AT+WHEEL0=60,100,2000
OK
AT+WHEEL1=60,100,2000
OK
AT+WHEEL0?
+WHEEL0=60.000000,100.000000,2000.000000
OK
AT+PID0=0.5,0.5,2.0
OK
AT+PID1=0.5,.,5

ERROR(9)
AT+PID1=0.5,0.5,2.0
OK
AT+ACC0=5000
OK
AT+ACC1=5000
OK
AT+SAVEALL
OK
```