TraxBot
Technical Manual

André Araújo
David Portugal
Micael Couceiro
Rui P. Rocha
Abstract:

This manual describes a mobile robot assembled in the Mobile Robotics Laboratory (MRL) of Institute of Systems and Robotics (ISR - Coimbra). The platform is based in two locomotion tracks, controlled by microcontroller ATmega328 from ATMEL, implemented in an Arduino Uno board that manages process information from the sonars, encoder readings and sends velocity commands through a motor driver that controls the two DC motors. This mobile compact robot is a robust platform able to integrate many more sensors and extended processing power to perform mobile robot routines.

Keyword list: TraxBot, mobile robot, ATmega328, Arduino Uno, motors, ultrasonic sonar range.

Mobile Robotic Laboratory (MRL)

Institute of Systems and Robotics - University of Coimbra (ISR-UC)
Portugal
Contact person: Rui P. Rocha
E-mail address: rprocha@isr.uc.pt
Webpage: http://paloma.isr.uc.pt/mrl/

Author contact: André Araújo
E-mail address: aaraujo@isr.uc.pt

Source webpage
Contents

Chapter 1 - Hardware Specifications ................................................................. 3
  1.1. TraxBot Main Assembly ................................................................. 4
  1.2. Main System Schematic ................................................................. 5
  1.3. Important Notes ........................................................................... 7
    1.3.1. Hardware startup ................................................................. 7
    1.3.2. Hardware maintenance ........................................................... 7
    1.3.3. Activation delay to sonar chain reads ....................................... 7
    1.3.4. Uploading code to Arduino ...................................................... 7

Chapter 2 - Components Specifications ......................................................... 8
  2.1. Sonar Range Finder Maxbotix MB1300 ............................................. 9
    2.1.1. Sonar Calibration ................................................................. 10
    2.1.2. Sonars chain connection ......................................................... 11
    2.3. Arduino UNO connections .......................................................... 13
    2.4. ZigBee Expansion communication module ....................................... 14
    2.5. DC Gearhead Motor Quadrature Encoder ........................................ 15
      2.5.1. Motor Specifications ............................................................ 15
      2.5.2. Wiring Specifications ............................................................ 16
      2.5.3. Incorporated encoder ............................................................. 16
    2.6. Battery Pack ............................................................................. 17

Appendix A - Components Dimensions ......................................................... 18
  TraxBot Robot (mm) ............................................................................. 19
  Extra fixation holes ............................................................................ 20
  Sonar Range Finder Maxbotix MB1300 .................................................. 21
  Motor driver OMNI-3MD ..................................................................... 21
  DC Gearhead Motor with Quadrature Encoder ....................................... 22
  Arduino UNO ..................................................................................... 22
  Battery pack ...................................................................................... 23

Appendix B - Datasheets ............................................................................. 24

Appendix C - Robot Port Configurations ...................................................... 28
Chapter 1 -

Hardware Specifications
1.1. TraxBot Main Assembly

The TraxBot robotic platform is a differential drive system (non-holonomic robot) built upon the Traxster II Robot educational Kit, equipped with 2 DC gearhead motors with quadrature wheel encoders. Since the original platform is endowed with ABS injection molded plastic tracks, rubber bands were attached to the tracks to increase friction and reduce slip during locomotion. The following figure gives a general view of the Traxster Robot.

![Figure 1. Schematic based in 3D model CAD](image)

The TraxBot is equipped with:

- 3 Maxbotix Sonars MB1300
- 1 Arduino Uno with ATmega328
- 1 Motor drive Omni 3MD
- 2 DC Gearhead Motor at 12V with Quadrature Encoder
- 2 Battery Ni-MH pack 12V 2300mAh
- RGB LEDs
- ZigBee module – Xbee Arduino Shield
The processing and control units are the Arduino Uno with a XBee shield module and the motor drive OMNI-3MD, which are located inside the TraxBot chassis. The battery pack is placed under the TraxBot and held together by two strips of Velcro on the outside for easy access and replacement. The circuit switch power is located in the back of the platform. Table 1 presents some hardware specifications.

<table>
<thead>
<tr>
<th>Specification</th>
<th>Value</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Voltage Range</td>
<td>9-12</td>
<td>V</td>
</tr>
<tr>
<td>Electric Current in Operation</td>
<td>1200</td>
<td>mA</td>
</tr>
<tr>
<td>Electric Current in Standby</td>
<td>110</td>
<td>mA</td>
</tr>
<tr>
<td>Maximum Speed</td>
<td>1,00</td>
<td>m/s</td>
</tr>
<tr>
<td>Weight</td>
<td>2045</td>
<td>g</td>
</tr>
<tr>
<td>Weight with notebook</td>
<td>3160</td>
<td>g</td>
</tr>
<tr>
<td>Width</td>
<td>203</td>
<td>mm</td>
</tr>
<tr>
<td>Length</td>
<td>229</td>
<td>mm</td>
</tr>
<tr>
<td>Height</td>
<td>125</td>
<td>mm</td>
</tr>
</tbody>
</table>

Table 1. Hardware Specifications of the TraxBot

1.2. Main System Schematic

This section describes the assembly of the components inside the platform and the control architecture.

![Main control circuit](image)

Figure 2. Main control circuit
The processing unit consists of an 8 bit microcontroller (µC) ATmega 328p from Atmel, embedded in an Arduino Uno. The CPU runs at 16 MHz and provides 14 MIPS of peak processing power; it also has 14 I/O digital pins (6 of them can be used as PWM analog outputs) and 6 analog inputs, up to 32kB of program memory and 2kB of SRAM. The ATmega8U2 embedded in Arduino Uno board creates a serial communication over USB to virtualize a COM port, to be used in computer. The ATmega 328p µC controls the platform’s motion through the use of the Bot’n Roll OMNI-3MD board.

For range sensing, the robot uses Maxbotix Sonars MB1300 with a maximum range of approximately 6 meters, which can have a configurable disposition with the possibility of using up to 4 sonars in one platform using the analog ports of the Arduino Uno board. Additionally, to enable point-to-point communication between robots, the Xbee Shield, consisting on a ZigBee communication antenna attached on top of the Arduino Uno board as an expansion module, was also incorporated. As for power source, two packs of 12V 2300mAh Ni-MH batteries are deployed under the chassis of each robot to ensure good autonomy. Finally, the platform also has the possibility of including a 10” netbook on top of an acrylic support, which extends the processing power and provides more flexibility. In our case, an ASUS eeePC 1001PXD BLACK N455 is used due to its reduced price and size. Using the netbook has the advantage of enabling communication via Wireless WiFi 802.11 b/g/n. Moreover, the netbook’s battery does not limit the autonomy of the platform, since it can operate autonomously for around 6 hours.

Figure 1 show a general view of the TraxBot assembly, figure 2 presents the main control circuit and diagram 1 presents the control modules.

Diagram 1. Control modules
1.3. Important Notes

1.3.1. Hardware startup
Before plugging the USB cable to the Arduino Uno (e.g., code upload or simply sending commands), make sure that the traxBot's power switch is ON, Fig. 3. Otherwise the robot will not respond, because the Arduino Uno uses the first initialized power source (battery or USB power).

1.3.2. Hardware maintenance
Every once in a while, it is convenient to check if the traction wheels have no gaps, are well lubricated and the track chain is with the appropriate tension, to avoid external errors on navigation.

1.3.3. Activation delay to sonar chain reads
In section 2.1.2, the chain architecture for reading sonar data will be explained in detail. It requires an activation time of up of 250 milliseconds, for proper data acquisition.

1.3.4. Uploading code to Arduino
If the XBee shield is mounted on top of the Arduino Uno board, its switch should be in USB mode in order to upload code, otherwise it will not be possible.

Figure 4. Available switches: a) XBee shield Serie 1  b) XBee shield Serie 2  c) External switch
Chapter 2  - Components Specifications
2.1. Sonar Range Finder Maxbotix MB1300

The Maxbotix MB1300 range sonar provides very short (15.24cm) to long-distance (645.16cm) detection and ranging, with 2.54cm resolution. The interface format used is the analogic output.

Among the advantages of this device, it insures a reliable and stable range data, it is a very low cost sonar, with virtually gone dead zone, which is one of the gold mark of this sensor. It also has very low power range, which is excellent for the battery system implemented in this project, with a fast measurement cycle and readings that occur every 50ms (20-Hz rate), which are reported directly, freeing up user processor.

**Main Features:**

- Continuously variable gain for beam control and side lobe suppression;
- Object detection includes zero range objects;
- 2.5V to 5.5V supply with 2mA typical current draw;
- Sensor operate at 42KHz;
- Free run operation can continually measure and output range information;
- Triggered operation provides the range reading as desire;
Output’s used:

- **AN**: Outputs analog voltage with a scaling factor of (Vcc/512). A supply of 5V yields ~3.9mV/cm and 3.3V yields ~2.5mV/cm. The output is buffered and corresponds to the most recent range data.
- **+5**: Vcc – Operates on 2.5V - 5.5V. Recommended current capability of 3mA for 5V, and 2mA for 3V.
- **GND**: Return for the DC power supply. GND (& Vcc) must be ripple and noise free for best operation.

### 2.1.1. Sonar Calibration

In order to convert the analog output values given by the sonar readings to centimeters, a calibration phase was conducted by measuring sonar readings in a straight line at a distance to a wall between 5 to 200 cm, with an increment of 5 cm. The data was collected and a curve fitting method by means of a power function $f(x) = ax^b + c$ was used, converting the analog values to centimeters, as shown in Figure 4.

![Figure 6. Fitting a curve to sonar data](image)

The calibration function obtained, with the sensor readings ($s$) as input, was:

$$f(s) = 1.1767 s^{0.9465} - 0.3759$$
2.1.2. Sonars chain connection

For better range sensing, the TraxBot uses 3 ultrasonic range sonars. It is possible to use up to 4 sonars in one platform through the available analog pins of the Arduino Uno board, since two of the 6 available analog pins are used as I2C communication. These ultrasonic sonars send out a pulse of sound and wait to receive the sound’s echo off of an obstacle. By measuring how long it takes for the sound to bounce back, the µC embedded on the sonar board calculates the distance that the sound traveled, and hence, how far the obstacle is. However, this originates an issue of different sonars “hearing” an echo from other sonars, which result in faulty detections. Hence, to avoid this crosstalk phenomenon, a chain loop was created to have synchronized cadence values from individual sonars, as shown in Fig. 7.

Please follow the instructions on topic 1.3.3 (1.3 Important notes section). In the next table, it is shown the sequence of the sonars chain readings).

<table>
<thead>
<tr>
<th>Reading Sequence</th>
<th>Sonar number (Appendix D)</th>
<th>Pinout Arduino</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Front</td>
<td>A0</td>
</tr>
<tr>
<td>2</td>
<td>Right</td>
<td>A1</td>
</tr>
<tr>
<td>3</td>
<td>Left</td>
<td>A2</td>
</tr>
</tbody>
</table>

*Table 2. Cadence sequence and Arduino pinouts*
2.2. Motor driver OMNI – 3MD

![Figure 8. Motor driver OMNI-3MD Board](image)

The motor driver Bot’n Roll OMNI-3MD operates in I2C slave mode to control 3 DC motors independently. It allows the control of the mobile robot platform, simply sending it the linear velocity, the direction and the rotation velocity, with the possibility to adjust the PID gains, encoders reading and the position control. Also, it is possible to monitor the battery voltage and the board temperature.

The OMNI-3MD has an embedded dsPIC30F6010 µC from Microchip and its CPU provides 30MIPS with a 16 bit processor. It has 144kB of program memory and up to 8kB of RAM memory. This driver has the ability to control three motors in omnidirectional platforms by sending linear velocity, direction and speed commands, performing both velocity and position control. In our case, we only use 2 motors for differential drive.

- Full manual with hardware specification and full describe communication protocol, [here](#).
- Arduino OMNI-3MD libraries, [here](#).
2.3. Arduino UNO connections

![Arduino Uno board with an ATmega 3280 from ATMEL](image)

**Figure 9.** Arduino Uno board with an ATmega 3280 from ATMEL

<table>
<thead>
<tr>
<th>PORTS</th>
<th>Description</th>
<th>From / To</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Range Sonar</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A0..1,2</td>
<td>Analog inputs from range sonars, which reads voltage values with a scaling factor of (Vcc/512).</td>
<td>An</td>
</tr>
<tr>
<td>5V</td>
<td>Power source</td>
<td>Vcc</td>
</tr>
<tr>
<td>GND</td>
<td>Ground</td>
<td>GND</td>
</tr>
<tr>
<td><strong>Motor Drive</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A4</td>
<td>Serial Data connection</td>
<td>SDA</td>
</tr>
<tr>
<td>A5</td>
<td>Serial Clock connection</td>
<td>SCL</td>
</tr>
<tr>
<td>5V</td>
<td>Power source</td>
<td>Vcc</td>
</tr>
<tr>
<td>GND</td>
<td>Ground</td>
<td>GND</td>
</tr>
<tr>
<td><strong>Arduino</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A3</td>
<td>Free input to integrate more sensors.</td>
<td>-</td>
</tr>
<tr>
<td><strong>USB</strong></td>
<td>USB jack</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.</td>
<td>USB PC</td>
</tr>
</tbody>
</table>

*Table 3. Ports description*
2.4. ZigBee Expansion communication module

The TraxBot provides wireless communication, which is fundamental for coordination of robot teams. ZigBee communication is embedded in the TraxBot, which is used for exchanging short messages between robots when operating without a notebook and running simple coordination algorithms.

ZigBee technology provides fast, low-cost wireless communication, featuring low-power consumption, possibility to support a huge number of networks nodes (theoretically up to 65536) as well as to set point-to-point, peer-to-peer, and multicast communication suitable for cooperative robot teams.

The TraxBot is equipped with a Xbee Shield from Maxstream, consisting on a ZigBee communication module with an antenna attached on top of the Arduino Uno board as an expansion module (Fig. 10), operating on the ZigBee protocol at standard IEEE 802.15.4 and using 2.4 GHz ISM band. This Xbee module is powered at 1mW having a range between 30m to 100m, for indoor and outdoor operation, respectively. The power consumption of this module is extremely low: 10µA in sleep mode and 50mA while sending and receiving data. The advantage of being an expansion of Arduino is the possibility of easily integrating wireless data exchange in the main source code.

![ZigBee Arduino Uno S2 expansion module](image)

**Figure 10.** ZigBee Arduino Uno S2 expansion module
2.5. DC Gearhead Motor Quadrature Encoder

![DC motor with quadrature encoder](image)

This is a Gear Motor w/Encoder, model No.GB37Y3530-12V-251R. It is a powerful 12V motor with a 43.7:1 metal gearbox and an integrated quadrature encoder that provides a resolution of 64 counts per revolution of the motor shaft, which corresponds to 2797 counts per revolution of the gearbox’s output shaft. These units have a 0.61” long, 6 mm-diameter D-shaped output shaft. This motor is intended for use at 12V, though the motor can begin rotating at voltages as low as 1V. The face plate has six mounting holes evenly spaced around the outer edge threaded for M3 screws. These mounting holes form a regular hexagon and the centers of neighboring holes are 15.5 mm apart.

### 2.5.1. Motor Specifications

- **Gear ratio:** 43.7:1
- **Free-run speed @ 12V:** 251 rpm
- **Free-run current @ 12V:** 0.4A
- **Rated torque @ 12V:** 1.5 kg\*cm
- **Encoder Resolution:** 64CPR(motor shaft)/2797CPR(gearbox shaft)
- **Weight:** 205g
2.5.2. Wiring Specifications

<table>
<thead>
<tr>
<th>Cable</th>
<th>Designation</th>
</tr>
</thead>
<tbody>
<tr>
<td>White</td>
<td>Motor + (12V)</td>
</tr>
<tr>
<td>Yellow</td>
<td>Motor -</td>
</tr>
<tr>
<td>Blue</td>
<td>Hall sensor GND</td>
</tr>
<tr>
<td>Green</td>
<td>Hall sensor Vcc (5-24V)</td>
</tr>
<tr>
<td>Red</td>
<td>Hall sensor A Output</td>
</tr>
<tr>
<td>Black</td>
<td>Hall sensor B Output</td>
</tr>
</tbody>
</table>

2.5.3. Incorporated encoder

The motors used in this project have a quadrature encoder incorporated, used to sense position and rotation by converting displacement into digital pulses. Consisting of a disk with coded patterns of opaque and transparent sectors that is attached to a rotating shaft, a quadrature encoder converts rotating patterns into two pulse output signals, A and B. When counted, these pulses determine position. The phase difference between output signal A and output signal B determines the direction of rotation. For example, if pulse output A leads pulse output B, as shown in Fig. 12, the shaft is rotating in the clockwise direction. Conversely, if pulse output B leads pulse output A, the shaft is rotating in the counter-clockwise direction.

![Diagram](image)

**Figure 12.** Direction of rotation given by the quadrature encoder output
2.6. Battery Pack

**Main Features**

- **Battery type:** Ni-MH
- **12V at 2300mAh**
- Used 2 battery pack in parallel, total 12V at 4600mAh

In order to choose appropriate batteries, several considerations were made, like capacity, energy density, power, number of cycles, available sizes, recycling requirements and cost. In Fig. 15, we present the specific energy that represents the capacity that a battery can hold in watt-hours per kilogram (Wh/kg) and the battery’s ability to deliver power in watts per kilogram. The battery type chosen in this project was the Ni-MH batteries, due to the balance between its performance and cost.

Therefore, two packs of 12V 2300mAh Ni-MH batteries, constituted by 8 AA Ni-MH cells, were deployed under the chassis of each robot to ensure good autonomy and the required power. Note that the notebook operates on its own battery without consuming TraxBot batteries.
Appendix A  - Components Dimensions
TraxBot Robot (mm)
Extra fixation holes

Motor driver OMNI 3MD

Acrylic base to support the EeePC

Arduino UNO
Sonar Range Finder Maxbotix MB1300

Motor driver OMNI-3MD
DC Gearhead Motor with Quadrature Encoder

Arduino UNO

Drawn by Matthew Beckler (matthew at mbecker dot org)
Bugs/updates/comments welcome!

All dimensions in mils (0.001")
PCB outside dimensions: 2700 x 2100
Mounting holes: ø=125

Drawn from the official PCB layout from:
2 Battery packs joined:

- Height: 100mm
- Width: 56mm
- Depth: 14mm
Appendix B - Datasheets
Datasheets and links with full details:


Arduino ZigBee - http://www.arduino.cc/en/Main/ArduinoXbeeShield
Appendix C - Robot Port Configurations